

საქართველოს ტექნიკური უნივერსიტეტი

რომან სამხარაძე, ლია გაჩეჩილაძე

ტესტების კრებული საგანში „Transact-SQL ენა“



დამტკიცებულია:
დამხმარე სახელმძღვანელოდ სტუ-ს
„IT-კონსალტინგის“ სამეცნიერო ცენტრის
სარედაქციო-საგამომცემლო კოლეგიის მიერ
ოქმი N4, 5.02.2016

თბილისი
2016

UDC: 004.65

კრებულში წარმოდგენილია T-SQL ენისთვის შემუშავებული ტესტები. ტესტებში ასახულია SQL სერვერთან მუშაობის საფუძვლები, მონაცემთა ბაზებისა და ცხრილების მართვის საკითხები, ფუნქციებთან, შენახულ პროცედურებთან, ინდექსებთან, წარმოდგენებთან, კურსორებთან და ტრიგერებთან მუშაობის პრინციპები. განკუთვნილია „ინფორმატიკა“ საგანმანათლებლო პროგრამის ბაკალავრებისთვის.

რეცენზენტი: ტექნიკის მეცნიერებათა დოქტორი, პროფესორი გ. სურგულაძე
ტექნიკის მეცნიერებათა კანდიდატი, პროფესორი მ. კიკნაძე

პროფ. გ. სურგულაძის რედაქციით

რედკოლეგია:

ა. ფრანგიშვილი (თავმჯდომარე), მ. ახოზაძე, ზ. ბაიაშვილი, ზ. ბოსიკაშვილი,
ზ. გასიტაშვილი, გ. გოგიჩაიშვილი, მ. თევდორაძე, ე. თურქია, ლ. იმნაიშვილი,
თ. კაიშაური, რ. კაკუბავა, ჰ. მელაძე, თ. ლომინაძე, ნ. ლომინაძე, თ. ოზგაძე,
რ. სამხარაძე, გ. სურგულაძე, გ. ჩაჩანიძე, ა. ცინცაძე, გ. ძიძიგური, ზ. წვერაიძე

პროფ. გ. სურგულაძის რედაქციით

© სტუ-ს "IT-კონსალტინგის სამეცნიერო ცენტრი", 2016

ISBN 978-9941-0-8559-8

<http://gtu.ge/publishinghouse/>

ყველა უფლება დაცულია. ამ წიგნის ნებისმიერი ნაწილის (ტექსტი, ფოტო, ილუსტრაცია თუ სხვა) არც ერთი ფორმითა დასაშუალებით (ელექტრონული თუ მექანიკური) არ შეიძლება გამომცემლის წერილობითი ნებართვის გარეშე.

საავტორო უფლების დარღვევა კანონით ისჯება.

სარჩევი

თავი 1. მონაცემთა ბაზების თეორიული საფუძვლები.....	7
საინფორმაციო სისტემები.....	7
მონაცემთა ბაზები.....	7
მონაცემთა ბაზების მოდელები.....	9
მონაცემთა რელაციური ბაზები.....	12
ცხრილების ნორმალიზება.....	12
ცხრილებს შორის არსებული კავშირები.....	14
თავი 2. მონაცემთა ბაზები.....	19
მონაცემთა ბაზების არქიტექტურა და ობიექტები.....	19
სისტემური მონაცემთა ბაზები. ფაილები და ფაილების ჯგუფები.....	29
მონაცემთა ბაზების შექმნა, წაშლა და სახელის შეცვლა.....	33
მონაცემთა ბაზის მიერთება და გამორთვა. ინფორმაციის მიღება მონაცემთა ბაზების შესახებ.....	39
მონაცემთა მომხმარებლის ტიპის შექმნა.....	42
თავი 3. ცხრილები.....	45
ცხრილების დაპროექტება.....	45
ცხრილის შექმნა. ცხრილის წაშლა.....	50
ცხრილისათვის სახელის შეცვლა. ინფორმაციის მიღება ცხრილების შესახებ. დიაგრამები....	69
თავი 4. მონაცემების მართვა.....	74
Transact SQL-ის საფუძვლები.....	74
ცხრილში მონაცემების ჩამატება, შეცვლა და წაშლა.....	84
ცხრილიდან მონაცემების ამორჩევა. SELECT განყოფილება.....	92
WHERE განყოფილება. ლოგიკის ოპერატორები.....	96
LIKE ოპერატორი.....	101
აგრეგირების ფუნქციები.....	104
GROUP BY განყოფილება.....	106
HAVING განყოფილება.....	108
ORDER BY განყოფილება.....	109
COMPUTE განყოფილება.....	111
NULL მნიშვნელობასთან მუშაობა.....	113
მონაცემების მასობრივი გადაწერა.....	115
თავი 5. შეკრებილება.....	118
FROM განყოფილება.....	118
ჯვარედინი შეერთებები.....	121

შიდა შეერთებები. შიდა შეერთების სახეები. შედგენილი შეერთებები. შეერთებები უტოლობის შემთხვევაში. მრავალცხრილიანი შეერთებები.....	122
გარე შეერთებები.....	124
თავი 6. მმართველი კონსტრუქციები.....	127
BEGIN...END.....	127
IF...ELSE.....	128
CASE.....	129
COALESCE.....	129
WHILE...BREAK & CONTINUE.....	130
თავი 7. ქვემოთხოვნი.....	132
დამოუკიდებელი სკალარული ქვემოთხოვნი. დამოუკიდებელი ქვემოთხოვნი მრავლობითი მნიშვნელობებით.....	132
ბმული ქვემოთხოვნი.....	135
ლოგიკის ოპერატორები. ALL და ANY ოპერატორები.....	136
EXISTS ოპერატორი.....	139
თავი 8. ოპერაციები სიმრავლეებზე.....	141
შესავალი. UNION ოპერაცია.....	141
INTERSECT ოპერაცია.....	144
EXCEPT ოპერაცია.....	145
პრიორიტეტი.....	147
თავი 9. ფუნქციები.....	148
მომხმარებლის ფუნქციები.....	148
Scalar ტიპის ფუნქციები.....	149
Inline ტიპის ფუნქციები.....	152
Multi-statement ფუნქციები.....	153
ფუნქციის წაშლა.....	154
ჩადგმული ფუნქციები. მათემატიკის ფუნქციები.....	154
სტრიქონებთან სამუშაო ფუნქციები.....	157
თარიღებთან სამუშაო ფუნქციები.....	161
თავი 10. შენახული პროცედურები.....	164
შესავალი. შენახული პროცედურების ტიპები.....	164
შენახული პროცედურის შექმნა.....	167
შენახული პროცედურების მართვა. შენახული პროცედურის შესახებ ინფორმაციის მიღება. შენახული პროცედურის სახელის შეცვლა. შენახული პროცედურის წაშლა. შენახული პროცედურის ავტომატურად შესრულების მართვა.....	172
თავი 11. ინდექსები.....	174

შესავალი. ინდექსების გამოყენების დაგეგმვა	174
არაკლასტერული ინდექსი. კლასტერული ინდექსი. უნიკალური ინდექსი	175
შევსების ფაქტორი	178
ინდექსის შექმნა	180
ინდექსების მართვა. ინდექსისთვის სახელის შეცვლა. ინდექსის წაშლა. ინდექსების გადაწყობა. ინდექსის შესახებ ინფორმაციის მიღება	184
თავი 12. წარმოდგენები	189
შესავალი.....	189
წარმოდგენის შექმნა.....	192
წარმოდგენების მართვა. წარმოდგენის სახელის შეცვლა. წარმოდგენის წაშლა. წარმოდგენის შესახებ ინფორმაციის მიღება. წარმოდგენის დამოკიდებულებების ნახვა.....	194
თავი 13. კურსორები.....	196
შესავალი. კურსორების რეალიზება.....	196
კურსორის ტიპები. სტატიკური კურსორები. დინამიკური კურსორები. მიმდევრობითი კურსორები. საგასაღებო კურსორები.....	198
კურსორების მართვა	200
კურსორის შექმნა. კურსორის გახსნა	201
მონაცემების წაკითხვა	206
მონაცემების შეცვლა. მონაცემების წაშლა.....	210
კურსორის დახურვა. კურსორის გათავისუფლება.....	210
თავი 14. ტრანზაქციები და დაბლოკვები	212
შესავალი.....	212
ტრანზაქციების მართვა. აშკარა ტრანზაქციები. ავტომატური ტრანზაქციები. არააშკარა ტრანზაქციები. განაწილებული ტრანზაქციები	213
ჩადგმული ტრანზაქციები. ტრანზაქციებში აკრძალული Transact SQL-ის ბრძანებები	217
დაბლოკვების მართვა.....	218
თავი 15. ტრიგერები.....	221
შესავალი.....	221
ტრიგერის შექმნა. ტრიგერის წაშლა.....	223
ტრიგერების შექმნისას გასათვალისწინებელი რეკომენდაციები. ტრიგერების მართვა. ტრიგერის სახელის შეცვლა. ტრიგერის შესახებ ინფორმაციის მიღება.....	230
თავი 16. ცხრილური ბამოსახულებები.....	234
წარმოებული ცხრილები. სვეტებისთვის ფსევდონიმების მინიჭება. არგუმენტების გამოყენება. ჩადგმულობა. მრავლობითი მიმართვები.....	234
საერთო ცხრილური გამოსახულებები. სვეტებისთვის ფსევდონიმების დანიშვნა. არგუმენტების გამოყენება. რამდენიმე საერთო ცხრილური გამოსახულების განსაზღვრა. მრავლობითი მიმართვები	235

APPLY ოპერაცია	236
თავი 17. მონაცემების რეორგანიზაცია და დაჯგუფების ნაკრებები	237
მონაცემების გაშლა. მონაცემების გაშლა T-SQL ენის PIVOT ოპერაციის გამოყენებით	237
მონაცემების შეკვება. მონაცემების შეკვება სტანდარტული SQL ენის საშუალებით. შეკვება T-SQL-ის UNPIVOT ოპერაციის საშუალებით	238
დაჯგუფების ნაკრებები. GROUPING SETS ჩადგმული ელემენტი. CUBE ჩადგმული ელემენტი. ROLLUP ჩადგმული ელემენტი. GROUPING() და GROUPING_ID() ფუნქციები	238
თავი 18. მონაცემთა ბაზისის სარეზერვო ასლები	241
სარეზერვო ასლის ტიპები	241
დამგროვებლის არჩევა. მოწყობილობის განსაზღვრა სარეზერვო ასლის მოსათავსებლად ...	242
მონაცემთა ბაზის სრული და დიფერენცირებული ასლები	244
სრული და დიფერენცირებული სარეზერვო ასლების შექმნა	246
სრული და დიფერენცირებული სარეზერვო ასლიდან აღდგენა	247
თავი 19. უსაფრთხოების სისტემა	249
უფლებამოსილების გამიჯვნის წესები	249
სერვერის უსაფრთხოების სისტემის არქიტექტურა. აუტენტიფიცირების რეჟიმები. Windows NT-ის აუტენტიფიცირების რეჟიმი. სერვერის აუტენტიფიცირების რეჟიმი	251
უსაფრთხოების სისტემის ელემენტები	255
სქემები	256
მომხმარებლები	260
სერვერის როლები. მონაცემთა ბაზის როლები	262
უსაფრთხოების სისტემის ადმინისტრირება. სააღრიცხვო ჩანაწერების შექმნა და მართვა ...	265
მიმართვის უფლებები. მონაცემთა ბაზის ობიექტებთან მიმართვის უფლებები	266
Transact-SQL-ის ბრძანებების შესრულების უფლებები	270
არაცხადი უფლებები	271
მიმართვის აკრძალვა	273
მიმართვის არაცხადი უარყოფა	274
დანართი. ტესტების პასუხები	276
ლიტერატურა	291

თავი 1. მონაცემთა ბაზების თეორიული საფუძვლები

საინფორმაციო სისტემები

1.1.1. საინფორმაციო სისტემა არის:

- ა. სტრუქტურირებული მონაცემებისა და აპარატურულ-პროგრამული საშუალებების ერთობლიობა
- ბ. მხოლოდ აპარატურულ-პროგრამული საშუალებების ერთობლიობა
- გ. მხოლოდ სტრუქტურირებული მონაცემების ერთობლიობა

1.1.2. არსებობს საინფორმაციო სისტემების:

- ა. 3 კლასი
- ბ. 2 კლასი
- გ. 1 კლასი

1.1.3. საინფორმაციო-სამეზნი საინფორმაციო სისტემები ორიენტირებულია:

- ა. მონაცემების ჩამატებაზე
- ბ. მონაცემების ცვლილებაზე
- გ. მონაცემების ძებნასა და მიღებაზე

1.1.4. მონაცემების დამუშავების სისტემები ორიენტირებულია:

- ა. მონაცემების ძებნაზე
- ბ. მონაცემების დამუშავებასა და დამუშავებული მონაცემების მიღებაზე
- გ. მონაცემების მიღებაზე

1.1.5. საინფორმაციო სისტემა შემდეგ ფუნქციებს ასრულებს:

- ა. მხოლოდ ინფორმაციის შეტანა, ცვლილება და შენახვა
- ბ. ინფორმაციის ნახვა და ძებნა
- გ. მხოლოდ ინფორმაციის ამორჩევა გარკვეული კრიტერიუმის მიხედვით

1.1.6. საინფორმაციო სისტემა შემდეგ ფუნქციებს ასრულებს:

- ა. ანგარიშების ფორმირება
- ბ. მხოლოდ ინფორმაციის ნახვა და ძებნა
- გ. მხოლოდ ინფორმაციის სისწორის შემოწმება

მონაცემთა ბაზები

1.2.1. მონაცემთა ბაზა არის გარკვეული საგნობრივი სფეროს ობიექტების შესახებ:

- ა. მონაცემების მოწესრიგებული ერთობლიობა
- ბ. მონაცემების დახარისხებული ერთობლიობა
- გ. მონაცემების მოუწესრიგებელი ერთობლიობა

1.2.2. მონაცემთა ბაზასთან მუშობა შეიძლება:

- ა. ნებისმიერი პროგრამის საშუალებით
- ბ. მონაცემთა ბაზის მართვის სისტემის საშუალებით
- გ. საფაილო სისტემის საშუალებით

1.2.3. მონაცემთა ბაზის მართვის სისტემა არის:

- ა. პროგრამული საშუალებების ერთობლიობა, რომელიც გამოიყენება პროგრამების დასაპროექტებლად
- ბ. პროგრამული საშუალებების ერთობლიობა, რომელიც გამოიყენება მონაცემების გარდასაქმნელად
- გ. პროგრამული საშუალებების ერთობლიობა, რომელიც საჭიროა მონაცემთა ბაზის შესაქმნელად და მასში მოთავსებულ ობიექტებზე მანიპულირებისათვის

1.2.4. მონაცემთა ბაზა შეიძლება იყოს:

- ა. განაწილებული ან ცენტრალიზებული
- ბ. მხოლოდ განაწილებული
- გ. მხოლოდ ცენტრალიზებული

1.2.5. მონაცემთა განაწილებული ბაზის ნაწილები განთავსებულია:

- ა. მხოლოდ ერთ კომპიუტერზე
- ბ. არსად არ არის განლაგებული
- გ. სხვადასხვა კომპიუტერზე

1.2.6. მონაცემთა ცენტრალიზებული ბაზა მოთავსებულია:

- ა. ერთ კომპიუტერზე
- ბ. რამდენიმე კომპიუტერზე
- გ. არსად არ არის მოთავსებული

1.2.7. არქიტექტურის მიხედვით მონაცემთა ცენტრალიზებული ბაზა შეიძლება იყოს:

- ა. მხოლოდ ფაილ-სერვერი
- ბ. ფაილ-სერვერი ან კლიენტ-სერვერი
- გ. მხოლოდ კლიენტ-სერვერი

1.2.8. ფაილ-სერვერული არქიტექტურის შემთხვევაში:

- ა. მონაცემების დამუშავება სერვერზე სრულდება
- ბ. მონაცემების დამუშავება ლოკალურ კომპიუტერზე სრულდება
- გ. მონაცემების დამუშავება არ სრულდება

1.2.9. ფაილ-სერვერული მონაცემთა ბაზის მართვის სისტემებია:

- ა. Microsoft Access და Paradox for Windows
- ბ. Microsoft SQL Server და Microsoft Access
- გ. Microsoft SQL Server და Oracle

1.2.10. კლიენტ-სერვერული არქიტექტურის შემთხვევაში მონაცემთა ბაზა მოთავსებულია:

- ა. ლოკალურ კომპიუტერზე
- ბ. არსად არ არის მოთავსებული
- გ. სერვერი

1.2.11. სისტემა, რომელიც იყენებს კლიენტ-სერვერულ ტექნოლოგიას, შედგება:

- ა. მხოლოდ კლიენტის ნაწილისაგან
- ბ. მხოლოდ სერვერის ნაწილისაგან

გ. კლიენტის ნაწილისაგან და სერვერის ნაწილისაგან

1.2.12. კლიენტ-სერვერულ ტექნოლოგიაში:

ა. კლიენტის ნაწილი არ უზრუნველყოფს გრაფიკულ ინტერფეისს და არ იმყოფება მომხმარებლის კომპიუტერზე

ბ. კლიენტის ნაწილი უზრუნველყოფს გრაფიკულ ინტერფეისს და იმყოფება მომხმარებლის კომპიუტერზე

გ. კლიენტის ნაწილი უზრუნველყოფს გრაფიკულ ინტერფეისს და არ იმყოფება მომხმარებლის კომპიუტერზე

1.2.13. კლიენტ-სერვერულ ტექნოლოგიაში:

ა. სერვერის ნაწილი მოთავსებულია სერვერზე და უზრუნველყოფს მონაცემების მართვას, დანაწილებას, ადმინისტრირებასა და უსაფრთხოებას

ბ. სერვერის ნაწილი მოთავსებული არ არის სერვერზე და უზრუნველყოფს მონაცემების მართვას, დანაწილებას, ადმინისტრირებასა და უსაფრთხოებას

გ. სერვერის ნაწილი მოთავსებული არ არის სერვერზე და არ უზრუნველყოფს მონაცემების მართვას, დანაწილებას, ადმინისტრირებასა და უსაფრთხოებას

1.2.14. კლიენტ-სერვერული მონაცემთა ბაზის მართვის სისტემებია:

ა. Microsoft SQL Server და Microsoft Access

ბ. Microsoft SQL Server და Oracle

გ. Microsoft Access და Microsoft Visual FoxPro

1.2.15. კლიენტ-სერვერული ტექნოლოგიისათვის:

ა. არ არის დამახასიათებელი SQL ენის გამოყენება

ბ. დამახასიათებელია C# ენის გამოყენება

გ. დამახასიათებელია SQL ენის გამოყენება

მონაცემთა ბაზების მოდელები

1.3.1. მონაცემების მოდელი:

ა. არის მონაცემების სტრუქტურისა და მათი დამუშავების ოპერაციების ერთობლიობა

ბ. არ არის მონაცემების სტრუქტურისა და მათი დამუშავების ოპერაციების ერთობლიობა

გ. არის ცხრილებისა და ინდექსების ერთობლიობა

1.3.2. მონაცემების მოდელის მიხედვით:

ა. შეუძლებელია ობიექტების სტრუქტურისა და მათ შორის არსებული კავშირების წარმოდგენა

ბ. შესაძლებელია მხოლოდ ობიექტების სტრუქტურის წარმოდგენა

გ. შესაძლებელია ობიექტების სტრუქტურისა და მათ შორის არსებული კავშირების წარმოდგენა

1.3.3. არსობობს მონაცემების:

ა. მხოლოდ რელაციური მოდელი

ბ. იერარქიული, ქსელური და რელაციური მოდელები

გ. მხოლოდ ქსელური მოდელი

- 1.3.4. მონაცემთა ბაზა შეიძლება იყოს:
- ა. მხოლოდ რელაციური
 - ბ. იერარქიული, ქსელური და რელაციური
 - გ. მხოლოდ ქსელური
- 1.3.5. მონაცემების იერარქიული მოდელის თითოეული ელემენტი:
- ა. სხვა ელემენტებს წარმოქმნის
 - ბ. სხვა ელემენტებს არ წარმოქმნის
 - გ. ინდექსებს წარმოქმნის
- 1.3.6. მონაცემების იერარქიული მოდელის ნებისმიერ წარმოქმნილ ელემენტს აქვს:
- ა. მხოლოდ ერთი წარმომქმნელი ელემენტი
 - ბ. რამდენიმე წარმომქმნელი ელემენტი
 - გ. მხოლოდ ორი წარმომქმნელი ელემენტი
- 1.3.7. მონაცემების იერარქიული მოდელის ნაკლია ის, რომ:
- ა. ხანდახან შესაძლებელია იერარქიის შეცვლა
 - ბ. შესაძლებელია შემუშავებული იერარქიის შეცვლა
 - გ. შეუძლებელია შემუშავებული იერარქიის შეცვლა
- 1.3.8. ქსელური მოდელის წარმოქმნილ ელემენტს:
- ა. არ შეიძლება ჰქონდეს ერთზე მეტი წარმომქმნელი ელემენტი
 - ბ. შეიძლება ჰქონდეს ერთზე მეტი წარმომქმნელი ელემენტი
 - გ. შეიძლება ჰქონდეს ერთი წარმომქმნელი ელემენტი
- 1.3.9. ქსელურ მოდელის ნაკლია ის, რომ:
- ა. იმისათვის, რომ მონაცემები დავამუშავოთ, არ უნდა ვიცოდეთ მონაცემთა ბაზის სტრუქტურა
 - ბ. იმისათვის, რომ მონაცემები დავამუშავოთ, უნდა ვიცოდეთ ტრიგერის სტრუქტურა
 - გ. იმისათვის, რომ მონაცემები დავამუშავოთ, უნდა ვიცოდეთ მონაცემთა ბაზის სტრუქტურა
- 1.3.10. მონაცემების რელაციური მოდელის შემთხვევაში:
- ა. მონაცემების წარმოდგენა ხდება ორგანზომილებიანი ცხრილის სახით
 - ბ. მონაცემების წარმოდგენა ხდება სამგანზომილებიანი მასივის სახით
 - გ. მონაცემების წარმოდგენა ხდება ერთგანზომილებიანი მასივის სახით
- 1.3.11. მარტივ შემთხვევაში, რელაციური მოდელი აღწერს:
- ა. ერთ ერთგანზომილებიან ცხრილს
 - ბ. ერთ ორგანზომილებიან ცხრილს
 - გ. ერთ სამგანზომილებიან ცხრილს
- 1.3.12. ხშირ შემთხვევაში, რელაციური მოდელი:
- ა. არ აღწერს არაფერს
 - ბ. აღწერს ერთი ცხრილის სტრუქტურასა და მის შორის კავშირებს

გ. აღწერს რამდენიმე ცხრილის სტრუქტურასა და მათ შორის კავშირებს

1.3.13. რელაციური არის მონაცემთა ისეთი ბაზა, რომელშიც მონაცემები წარმოდგენილია:

- ა. სწორკუთხა ორგანზომილებიანი ცხრილების სახით
- ბ. სამკუთხა ორგანზომილებიანი ცხრილების სახით
- გ. ორკუთხა ორგანზომილებიანი ცხრილების სახით

1.3.14. მონაცემთა ბაზის შიგნით ცხრილის სახელი:

- ა. შეიძლება გამოვრდეს
- ბ. არ არის უნიკალური
- გ. უნიკალურია

1.3.15. ცხრილი შედგება:

- ა. მხოლოდ სტრიქონებისაგან
- ბ. სტრიქონებისა და სვეტებისაგან
- გ. მხოლოდ სვეტებისაგან

1.3.16. ცხრილი შეიცავს ინფორმაციას:

- ა. ორტიპური ობიექტების შესახებ
- ბ. მრავალტიპური ობიექტების შესახებ
- გ. ერთტიპური ობიექტების შესახებ

1.3.17. ცხრილის სტრიქონი შეიცავს ინფორმაციას:

- ა. კონკრეტული ობიექტის შესახებ
- ბ. არაკონკრეტული ობიექტის შესახებ
- გ. სხვადასხვა ობიექტის შესახებ

1.3.18. ცხრილის თითოეული სვეტი:

- ა. წარმოადგენს ობიექტების კონკრეტული ატრიბუტის მნიშვნელობების ერთობლიობას
- ბ. არ წარმოადგენს ობიექტების კონკრეტული ატრიბუტის მნიშვნელობების ერთობლიობას
- გ. წარმოადგენს ობიექტების არაკონკრეტული ატრიბუტის მნიშვნელობების ერთობლიობას

1.3.19. თითოეულ სვეტს აქვს:

- ა. მხოლოდ სახელი
- ბ. სახელი და ტიპი
- გ. მხოლოდ ტიპი

1.3.20. ერთი ცხრილის სვეტებს:

- ა. ერთნაირი სახელები უნდა ჰქონდეს
- ბ. სხვადასხვა სახელები არ უნდა ჰქონდეს
- გ. სხვადასხვა სახელები უნდა ჰქონდეს

1.3.21. სხვადასხვა ცხრილებს:

- ა. არ შეიძლება ერთნაირი სახელის მქონე სვეტები ჰქონდეთ
- ბ. აუცილებლად უნდა ჰქონდეთ სხვადასხვა სახელები
- გ. შეიძლება ერთნაირი სახელის მქონე სვეტები ჰქონდეთ

1.3.22. ცხრილის სტრიქონებს სახელები:

- ა. არ აქვთ
- ბ. აქვთ
- გ. ხანდახან აქვთ

მონაცემთა რელაციური ბაზები

1.4.1. მონაცემთა რელაციური ბაზა:

- ა. არის ერთმანეთთან ლოგიკურად დაკავშირებული ცხრილების ერთობლიობა
- ბ. არ არის ერთმანეთთან ლოგიკურად დაკავშირებული ცხრილების ერთობლიობა
- გ. არის ერთმანეთთან ლოგიკურად დაუკავშირებელი ცხრილების ერთობლიობა

1.4.2. მონაცემთა ბაზაში თითოეულ ცხრილს:

- ა. არ უნდა ჰქონდეს უნიკალური სახელი და უნდა შედგებოდეს ერთტიპური სტრიქონებისაგან
- ბ. უნდა ჰქონდეს უნიკალური სახელი და უნდა შედგებოდეს ერთტიპური სტრიქონებისაგან
- გ. უნდა ჰქონდეს უნიკალური სახელი და არ უნდა შედგებოდეს ერთტიპური სტრიქონებისაგან

1.4.3. სტრიქონის ერთ სვეტში მოთავსებული უნდა იყოს:

- ა. მხოლოდ ორი მნიშვნელობა
- ბ. მხოლოდ ერთი მნიშვნელობა
- გ. მხოლოდ სამი მნიშვნელობა

1.4.4. ცხრილში:

- ა. უნდა იყოს ზუსტად ერთნაირი ორი სტრიქონი
- ბ. უნდა იყოს ზუსტად ერთნაირი სამი სტრიქონი
- გ. არ უნდა იყოს ზუსტად ერთნაირი ორი სტრიქონი

1.4.5. სტრიქონები უნდა განსხვავდებოდეს:

- ა. სამი სვეტის მნიშვნელობით მაინც
- ბ. ორი სვეტის მნიშვნელობით მაინც
- გ. ერთი სვეტის მნიშვნელობით მაინც

ცხრილების ნორმალიზება

1.5.1. მონაცემთა ბაზის დაპროექტების მიზნებია:

- ა. მონაცემების სიჭარბის არგამორიცხვა
- ბ. ცხრილების რაოდენობის მაქსიმუმამდე დაყვანა
- გ. მონაცემთა ბაზაში საჭირო მონაცემების შენახვის უზრუნველყოფა, მონაცემების სიჭარბის გამორიცხვა და ცხრილების რაოდენობის მინიმუმამდე დაყვანა

1.5.2. ცხრილების ნორმალიზება არის:

- ა. მონაცემების წარმოდგენის პროცესი მარტივი ორგანოზომილებიანი ცხრილების სახით

- ბ. მონაცემების წარმოდგენის პროცესი მარტივი სამგანზომილებიანი ცხრილების სახით
- გ. მონაცემების წარმოდგენის პროცესი მარტივი ერთგანზომილებიანი ცხრილების სახით

1.5.3. ნორმალიზების მიზანია:

- ა. ცხრილების რაოდენობის მაქსიმუმამდე დაყვანა
- ბ. მონაცემების სიჭარბის გამორიცხვა
- გ. ცხრილების გადაწერა

1.5.4. ნორმალიზების მიზანია:

- ა. მონაცემების წინააღმდეგობრიობის გამორიცხვა
- ბ. მონაცემების არაწინააღმდეგობრიობის გამორიცხვა
- გ. მონაცემების წინააღმდეგობრიობის არგამორიცხვა

1.5.5. არსებობს ცხრილის ნორმალიზების:

- ა. სამი ფორმა
- ბ. ექვსი ფორმა
- გ. შვიდი ფორმა

1.5.6. ცხრილი იმყოფება პირველ ნორმალურ ფორმაში მაშინ, როცა ის:

- ა. არ შეიცავს გამეორებად სვეტებს და სვეტების შედგენილ მნიშვნელობებს
- ბ. შეიცავს გამეორებად სვეტებს და სვეტების შედგენილ მნიშვნელობებს
- გ. შეიცავს გამეორებად სვეტებს და სვეტების შეუდგენელ მნიშვნელობებს

1.5.7. პირველ ნორმალურ ფორმაში მყოფ ცხრილს:

- ა. უნივერსალურ ცხრილს არ უწოდებენ
- ბ. ფიზიკურ ცხრილს უწოდებენ
- გ. უნივერსალურ ცხრილს უწოდებენ

1.5.8. ცხრილი იმყოფება მეორე ნორმალურ ფორმაში მაშინ, როცა ის:

- ა. აკმაყოფილებს პირველი ნორმალური ფორმის მოთხოვნებს, შეიცავს ერთ ან მეტ სვეტს, რომლებიც ქმნიან პირველად გასაღებს თითოეული სტრიქონისათვის და ყველა ის სვეტი, რომელიც შედის პირველად გასაღებში, პირველად გასაღებთან დაკავშირებულია სრული ფუნქციონალური დამოკიდებულებით
- ბ. არ აკმაყოფილებს პირველი ნორმალური ფორმის მოთხოვნებს
- გ. აკმაყოფილებს პირველი ნორმალური ფორმის მოთხოვნებს, შეიცავს ერთ ან მეტ სვეტს, რომლებიც ქმნიან პირველად გასაღებს თითოეული სტრიქონისათვის და ყველა ის სვეტი, რომელიც არ შედის პირველად გასაღებში, პირველად გასაღებთან დაკავშირებულია სრული ფუნქციონალური დამოკიდებულებით

1.5.9. თუ ცხრილს აქვს მარტივი პირველადი გასაღები, რომელიც მხოლოდ ერთი სვეტისაგან შედგება, მაშინ ის:

- ა. ავტომატურად იმყოფება პირველ ნორმალურ ფორმაში
- ბ. ავტომატურად იმყოფება მეორე ნორმალურ ფორმაში
- გ. ავტომატურად არ იმყოფება მეორე ნორმალურ ფორმაში

1.5.10. ცხრილი იმყოფება მესამე ნორმალურ ფორმაში მაშინ, როცა ის:

ა. აკმაყოფილებს მეორე ნორმალური ფორმის მოთხოვნებს და მისი არც ერთი არასაგასაღებო სვეტი ფუნქციონალურად არ არის დამოკიდებული სხვა ნებისმიერ არასაგასაღებო სვეტზე

ბ. არ აკმაყოფილებს მეორე ნორმალური ფორმის მოთხოვნებს

გ. აკმაყოფილებს მეორე ნორმალური ფორმის მოთხოვნებს და მისი ყველა არასაგასაღებო სვეტი ფუნქციონალურად არ არის დამოკიდებული სხვა ნებისმიერ არასაგასაღებო სვეტზე

1.5.11. მესამე ნორმალური ფორმის მოთხოვნა დაიყვანება იმაზე, რომ:

ა. ყველა არასაგასაღებო სვეტი დამოკიდებული იყოს მხოლოდ პირველად გასაღებზე და იყოს დამოკიდებული ერთმანეთზე

ბ. ყველა არასაგასაღებო სვეტი დამოკიდებული არ იყოს მხოლოდ პირველად გასაღებზე

გ. ყველა არასაგასაღებო სვეტი დამოკიდებული იყოს მხოლოდ პირველად გასაღებზე და არ იყოს დამოკიდებული ერთმანეთზე

ცხრილებს შორის არსებული კავშირები

1.6.1. ორ ცხრილს შორის კავშირის დამყარების დროს:

ა. ორივე მთავარია

ბ. ერთი არის მთავარი (მშობელი), მეორე კი - დამოკიდებული (შვილობილი)

გ. ორივე დამოკიდებულია

1.6.2. ორ ცხრილს შორის კავშირის დასამყარებლად:

ა. მთავარი ცხრილის პირველადი გასაღები უკავშირდება დამოკიდებული ცხრილის გარე გასაღებს

ბ. დამოკიდებული ცხრილის პირველადი გასაღები უკავშირდება მთავარი ცხრილის გარე გასაღებს

გ. დამოკიდებული ცხრილის პირველადი გასაღები უკავშირდება დამოკიდებული ცხრილის გარე გასაღებს

1.6.3. პირველადი გასაღები:

ა. ცალსახად არ განსაზღვრავს თითოეულ სტრიქონს ცხრილში

ბ. ცალსახად განსაზღვრავს თითოეულ სტრიქონს ცხრილში

გ. ცალსახად განსაზღვრავს თითოეულ სვეტს ცხრილში

1.6.4. პირველადი გასაღები უნდა აკმაყოფილებდეს შემდეგ მოთხოვნებს:

ა. უნიკალურობა და მინიმალურობა

ბ. მხოლოდ უნიკალურობა

გ. მხოლოდ მინიმალურობა

1.6.5. უნიკალურობა იმას ნიშნავს, რომ:

ა. ცხრილში უნდა იყოს ორი ისეთი სტრიქონი, რომლებსაც პირველადი გასაღების ერთნაირი მნიშვნელობები ექნებათ

ბ. ცხრილში არ უნდა იყოს სამი ისეთი სტრიქონი, რომლებსაც პირველადი გასაღების ერთნაირი მნიშვნელობები ექნებათ

გ. ცხრილში არ უნდა იყოს ორი ისეთი სტრიქონი, რომლებსაც პირველადი გასაღების ერთნაირი მნიშვნელობები ექნებათ

1.6.6. მინიმალურობა იმას ნიშნავს, რომ:

- ა. პირველად გასაღებში შემავალი სვეტებიდან ნებისმიერის გამოკლება არ უნდა იწვევდეს გასაღების უნიკალურობის დარღვევას
- ბ. პირველად გასაღებში შემავალი სვეტებიდან ნებისმიერის გამოკლება უნდა იწვევდეს გასაღების უნიკალურობის დარღვევას
- გ. პირველად გასაღებში შემავალი სვეტებიდან ნებისმიერის გამოკლება უნდა იწვევდეს გასაღების უნიკალურობის დაურღვევლობას

1.6.7. პირველადი გასაღები არ უნდა შეიცავდეს ისეთ სვეტს, რომლის მოცილების შემდეგ გასაღები:

- ა. ისევ უნიკალური დარჩება
- ბ. ისევ უნიკალური არ დარჩება
- გ. ისევ არაუნიკალური დარჩება

1.6.8. პირველადი გასაღები:

- ა. უნდა შეიცავდეს ცხრილის მხოლოდ ერთ სვეტს
- ბ. არ შეიძლება შეიცავდეს ცხრილის ერთ ან მეტ სვეტს
- გ. შეიძლება შეიცავდეს ცხრილის ერთ ან მეტ სვეტს

1.6.9. პირველად გასაღებში შემავალმა:

- ა. არც ერთმა სვეტმა არ უნდა მიიღოს განუსაზღვრელი მნიშვნელობა
- ბ. ყველა სვეტმა უნდა მიიღოს განუსაზღვრელი მნიშვნელობა
- გ. არც ერთმა სვეტმა არ უნდა მიიღოს განსაზღვრული მნიშვნელობა

1.6.10. გარე გასაღები (FOREIGN KEY) იქმნება:

- ა. მთავარ ცხრილში
- ბ. დამოკიდებულ ცხრილში
- გ. როგორც მთავარ, ისე დამოკიდებულ ცხრილში

1.6.11. გარე გასაღები შეიძლება შედგებოდეს:

- ა. მხოლოდ ერთი სვეტისგან
- ბ. არც ერთი სვეტისგან
- გ. ერთი ან მეტი სვეტისაგან

1.6.12. გარე გასაღების მნიშვნელობა:

- ა. ყოველთვის უნდა არსებობდეს მთავარ ცხრილში ან არ უნდა იყოს განუსაზღვრელი
- ბ. ყოველთვის არ უნდა არსებობდეს მთავარ ცხრილში ან უნდა იყოს განუსაზღვრელი
- გ. ყოველთვის უნდა არსებობდეს მთავარ ცხრილში ან უნდა იყოს განუსაზღვრელი

1.6.13. მთავარი ცხრილის რომელიმე სტრიქონის პირველადი გასაღების მნიშვნელობა:

- ა. შეიძლება იყოს დაკავშირებული დამოკიდებულებული ცხრილის ყველა სტრიქონთან
- ბ. შეიძლება იყოს დაკავშირებული დამოკიდებულებული ცხრილის არც ერთ სტრიქონთან
- გ. შეიძლება არ იყოს დაკავშირებული დამოკიდებულებული ცხრილის არც ერთ სტრიქონთან

1.6.14. მთავარ ცხრილში პირველადი გასაღების მნიშვნელობის შეცვლისას არსებობს:

- ა. დამოკიდებული ცხრილის ქცევის სამი ვარიანტი
- ბ. დამოკიდებული ცხრილის ქცევის ორი ვარიანტი
- გ. დამოკიდებული ცხრილის ქცევის ერთი ვარიანტი

1.6.15. მთავარ ცხრილში პირველადი გასაღების მნიშვნელობის შეცვლისას არსებობს დამოკიდებული ცხრილის ქცევის შემდეგი ვარიანტები:

- ა. მხოლოდ კასკადირება
- ბ. კასკადირება, შეზღუდვა და დაყენება
- გ. მხოლოდ შეზღუდვა

1.6.16. კასკადირების დროს:

- ა. მთავარ ცხრილში პირველადი გასაღების მნიშვნელობების შეცვლისას სრულდება დამოკიდებული ცხრილის გარე გასაღების შესაბამისი მნიშვნელობების ცვლილება. ხდება ყველა არსებული კავშირების შენარჩუნება
- ბ. მთავარ ცხრილში პირველადი გასაღების მნიშვნელობების შეცვლისას არ სრულდება დამოკიდებული ცხრილის გარე გასაღების შესაბამისი მნიშვნელობების ცვლილება
- გ. არ ხდება ყველა არსებული კავშირების შენარჩუნება

1.6.17. შეზღუდვის დროს:

- ა. დაუშვებელია პირველადი გასაღების იმ მნიშვნელობების შეცვლა, რომლებიც არ არის დაკავშირებული დამოკიდებული ცხრილის სტრიქონებთან
- ბ. დასაშვებია პირველადი გასაღების იმ მნიშვნელობების შეცვლა, რომლებიც დაკავშირებულია დამოკიდებული ცხრილის სტრიქონებთან
- გ. დაუშვებელია პირველადი გასაღების იმ მნიშვნელობების შეცვლა, რომლებიც დაკავშირებულია დამოკიდებული ცხრილის სტრიქონებთან. დასაშვებია პირველადი გასაღების იმ მნიშვნელობების შეცვლა, რომლებიც არ არიან დაკავშირებული დამოკიდებული ცხრილის სტრიქონებთან

1.6.18. დაყენების დროს:

- ა. ინფორმაცია დამოკიდებული ცხრილის სტრიქონების კავშირის შესახებ არ იკარგება
- ბ. პირველადი გასაღების მნიშვნელობის შეცვლის დროს გარე გასაღები იღებს განსაზღვრულ მნიშვნელობას
- გ. პირველადი გასაღების მნიშვნელობის შეცვლის დროს გარე გასაღები იღებს განუსაზღვრელ მნიშვნელობას (NULL). ინფორმაცია დამოკიდებული ცხრილის სტრიქონების კავშირის შესახებ იკარგება

1.6.19. მთავარი ცხრილის სტრიქონის წაშლისას არსებობს დამოკიდებული ცხრილის ქცევის შემდეგი ვარიანტები:

- ა. მხოლოდ კასკადირება
- ბ. კასკადირება, შეზღუდვა და დაყენება
- გ. მხოლოდ დაყენება

1.6.20. კასკადირების დროს:

- ა. დამოკიდებულ ცხრილში სტრიქონების წაშლისას სრულდება მთავარი ცხრილის შესაბამისი სტრიქონების წაშლა
- ბ. მთავარ ცხრილში სტრიქონების წაშლისას არ სრულდება დამოკიდებული ცხრილის

შესაბამისი სტრიქონების წაშლა

გ. მთავარ ცხრილში სტრიქონების წაშლისას სრულდება დამოკიდებული ცხრილის შესაბამისი სტრიქონების წაშლა

1.6.21. შეზღუდვის დროს:

ა. დასაშვებია მთავარი ცხრილის მხოლოდ იმ სტრიქონების წაშლა, რომლებსაც არ აქვთ ბმული სტრიქონები დამოკიდებულ ცხრილში. წინააღმდეგ შემთხვევაში, წაშლის ოპერაცია არ შესრულდება

ბ. დაუშვებელია მთავარი ცხრილის მხოლოდ იმ სტრიქონების წაშლა, რომლებსაც არ აქვთ ბმული სტრიქონები დამოკიდებულ ცხრილში

გ. დასაშვებია მთავარი ცხრილის მხოლოდ იმ სტრიქონების წაშლა, რომლებსაც აქვთ ბმული სტრიქონები დამოკიდებულ ცხრილში

1.6.22. დაყენების დროს:

ა. მთავარი ცხრილიდან სტრიქონის წაშლის დროს დამოკიდებულ ცხრილში შესაბამისი გარე გასაღები არ იღებს განუსაზღვრელ მნიშვნელობას

ბ. მთავარი ცხრილიდან სტრიქონის წაშლის დროს დამოკიდებულ ცხრილში შესაბამისი გარე გასაღები იღებს განსაზღვრულ მნიშვნელობას

გ. მთავარი ცხრილიდან სტრიქონის წაშლის დროს დამოკიდებულ ცხრილში შესაბამისი გარე გასაღები იღებს განუსაზღვრელ მნიშვნელობას (NULL)

1.6.23. ორ ცხრილს შორის დამყარებულია „ერთი-ერთთან“ კავშირი:

ა. თუ პირველი (მთავარი) ცხრილის კონკრეტული სტრიქონი დროის ნებისმიერ მომენტში დაკავშირებული არ არის მეორე (დამოკიდებული) ცხრილის ნულ (არც ერთ) ან ერთ სტრიქონთან

ბ. თუ პირველი (მთავარი) ცხრილის კონკრეტული სტრიქონი დროის ნებისმიერ მომენტში დაკავშირებულია მეორე (დამოკიდებული) ცხრილის ნულ (არც ერთ) ან ერთ სტრიქონთან

გ. თუ პირველი (მთავარი) ცხრილის კონკრეტული სტრიქონი დროის ნებისმიერ მომენტში დაკავშირებულია მეორე (დამოკიდებული) ცხრილის ნულ (არც ერთ) ან რამდენიმე სტრიქონთან

1.6.24. „ერთი-ერთთან“ კავშირი იქმნება იმ შემთხვევაში, როცა:

ა. ორივე სვეტი არის პირველადი გასაღები ან ორივე სვეტს აქვს უნიკალურობის შეზღუდვა

ბ. ორივე სვეტი არ არის პირველადი გასაღები

გ. ორივე სვეტს არ აქვს უნიკალურობის შეზღუდვა

1.6.25. „ერთი-ერთთან“ კავშირი იქმნება იმ შემთხვევაში, როცა:

ა. არ გვინდა ბევრი სვეტის მქონე ცხრილის დაყოფა

ბ. არ გვინდა ცხრილის ნაწილის იზოლირება უსაფრთხოების მოსაზრებიდან გამომდინარე

გ. გვინდა ბევრი სვეტის მქონე ცხრილის დაყოფა, ცხრილის ნაწილის იზოლირება უსაფრთხოების მოსაზრებიდან გამომდინარე და ა.შ.

1.6.26. „ერთი-ბევრთან“ კავშირი იქმნება იმ შემთხვევაში:

ა. თუ პირველი (მთავარი) ცხრილის კონკრეტული სტრიქონი დროის ნებისმიერ მომენტში დაკავშირებულია მეორე (დამოკიდებული) ცხრილის ნულ, ერთ ან მეტ სტრიქონთან და მეორე ცხრილის ნებისმიერი სტრიქონი დაკავშირებულია პირველის ცხრილის მხოლოდ ერთ სტრიქონთან

ბ. თუ პირველი (მთავარი) ცხრილის კონკრეტული სტრიქონი დროის ნებისმიერ მომენტში

დაკავშირებულია მეორე (დამოკიდებული) ცხრილის ნულ, ერთ ან მეტ სტრიქონთან და მეორე ცხრილის ნებისმიერი სტრიქონი დაკავშირებულია პირველის ცხრილის რამდენიმე სტრიქონთან გ. თუ პირველი (მთავარი) ცხრილის კონკრეტული სტრიქონი დროის ნებისმიერ მომენტში დაკავშირებული არ არის მეორე (დამოკიდებული) ცხრილის ნულ, ერთ ან მეტ სტრიქონთან

1.6.27. „ბევრი-ბევრთან“ კავშირი იქმნება იმ შემთხვევაში:

ა. თუ პირველი (მთავარი) ცხრილის კონკრეტული სტრიქონი დროის ნებისმიერ მომენტში დაკავშირებულია მეორე (დამოკიდებული) ცხრილის ნულ, ერთ ან მეტ სტრიქონთან და მეორე ცხრილის ნებისმიერი სტრიქონი დაკავშირებულია პირველის ცხრილის მხოლოდ ერთ სტრიქონთან

ბ. თუ პირველი (მთავარი) ცხრილის კონკრეტული სტრიქონი დროის ნებისმიერ მომენტში დაკავშირებულია მეორე (დამოკიდებული) ცხრილის ნულ, ერთ ან მეტ სტრიქონთან და პირიქით

გ. თუ პირველი (მთავარი) ცხრილის კონკრეტული სტრიქონი დროის ნებისმიერ მომენტში დაკავშირებულია მეორე (დამოკიდებული) ცხრილის ნულ (არც ერთ) ან ერთ სტრიქონთან

თავი 2. მონაცემთა ბაზები

მონაცემთა ბაზების არქიტექტურა და ობიექტები

- 2.1.1. მონაცემთა ბაზის სტრუქტურა შეიძლება იყოს:
- ა. მხოლოდ ლოგიკური
 - ბ. მხოლოდ ფიზიკური
 - გ. ლოგიკური და ფიზიკური
- 2.1.2. მონაცემთა ბაზის ლოგიკური სტრუქტურა მოიცავს:
- ა. მხოლოდ ცხრილების სტრუქტურას
 - ბ. ცხრილების სტრუქტურას, მათ შორის კავშირებს, მომხმარებლებების სიას, შენახულ პროცედურებს და მონაცემთა ბაზის სხვა ობიექტებს
 - გ. მხოლოდ ცხრილებს შორის კავშირებს
- 2.1.3. მონაცემთა ბაზის ფიზიკური სტრუქტურა:
- ა. მოიცავს მონაცემთა ბაზის ფაილებისა და ტრანზაქციების ჟურნალის აღწერას, მათ საწყის ზომას, ნაზარდის ზომას, მაქსიმალურ ზომას, კონფიგურირების პარამეტრებს და ა.შ.
 - ბ. არ მოიცავს მონაცემთა ბაზის ფაილებისა და ტრანზაქციების ჟურნალის აღწერას, მათ საწყის ზომას, ნაზარდის ზომას, მაქსიმალურ ზომას, კონფიგურირების პარამეტრებს და ა.შ.
 - გ. მოიცავს მონაცემთა ბაზის ლოგიკურ სტრუქტურას
- 2.1.4. მონაცემთა ბაზების ობიექტებია:
- ა. მხოლოდ ცხრილები
 - ბ. ცხრილები, შენახული პროცედურები, ტრიგერები, წარმოდგენები, ინდექსები, მომხმარებლების ტიპები, მომხმარებლების ფუნქციები
 - გ. მხოლოდ ცხრილები და ინდექსები
- 2.1.5. მონაცემთა ბაზის ობიექტებიდან საკუთრივ მონაცემებს შეიცავს მხოლოდ:
- ა. ცხრილი
 - ბ. ტრიგერი
 - გ. წარმოდგენა
- 2.1.6. ცხრილის ზოგიერთი სვეტი შეიძლება იყოს:
- ა. გამოთვლადი
 - ბ. წარმოდგენა
 - გ. პირობითი
- 2.1.7. გამოთვლად სვეტებში მიეთითება:
- ა. წარმოდგენა
 - ბ. ფორმულა
 - გ. პირობა
- 2.1.8. სისტემური ცხრილები შეიცავს:
- ა. ინფორმაციას მხოლოდ ტრიგერებისა და ინდექსების შესახებ
 - ბ. სერვერის მუშაობისათვის არასაჭირო ინფორმაციას
 - გ. სერვერის მუშაობისათვის საჭირო ინფორმაციას

2.1.9. დროებითი ცხრილები განკუთვნილია:

- ა. მონაცემების გლობალურად შესანახად
- ბ. მონაცემების ხანგრძლივად შესანახად
- გ. მონაცემების დროებით შესანახად

2.1.10. დროებითი ცხრილები მოთავსებულია:

- ა. model სისტემურ მონაცემთა ბაზაში
- ბ. tempdb სისტემურ მონაცემთა ბაზაში
- გ. master სისტემურ მონაცემთა ბაზაში

2.1.11. თუ დროებითი ცხრილის შექმნის დროს აშკარად მივუთითებთ მონაცემთა ბაზის სახელს, რომელშიც ის უნდა შეიქმნას, მაშინ:

- ა. ცხრილი tempdb სისტემურ მონაცემთა ბაზაში შეიქმნება
- ბ. ცხრილი master სისტემურ მონაცემთა ბაზაში შეიქმნება
- გ. ცხრილი მომხმარებლის მონაცემთა ბაზაში შეიქმნება

2.1.12. დროებითი ცხრილი:

- ა. არ არის ლოკალური და გლობალური
- ბ. არის მხოლოდ ლოკალური
- გ. არის ლოკალური და გლობალური

2.1.13. დროებითი ცხრილები ავტომატურად:

- ა. არ იშლება შეერთების დახურვის დროს
- ბ. იშლება შეერთების დახურვის დროს
- გ. იშლება შეერთების გახსნის დროს

2.1.14. ლოკალური დროებითი ცხრილის სახელი:

- ა. ერთი # სიმბოლოთი იწყება
- ბ. ორი # სიმბოლოთი იწყება
- გ. # სიმბოლოთი არ იწყება

2.1.15. ლოკალური დროებითი ცხრილები:

- ა. ხილულია, მხოლოდ იმ შეერთების შიგნით, რომელშიც ისინი არ შეიქმნა
- ბ. უხილავია, მხოლოდ იმ შეერთების შიგნით, რომელშიც ისინი შეიქმნა
- გ. ხილულია, მხოლოდ იმ შეერთების შიგნით, რომელშიც ისინი შეიქმნა

2.1.16. შეერთების დახურვის დროს:

- ა. ლოკალური დროებითი ცხრილი ავტომატურად იშლება
- ბ. ლოკალური დროებითი ცხრილი ავტომატურად არ იშლება
- გ. ლოკალური დროებითი ცხრილი ხანდახან ავტომატურად არ იშლება

2.1.17. თუ ლოკალური დროებითი ცხრილი შეიქმნა შენახული პროცედურის მუშაობის დროს, მაშინ ამ პროცედურიდან გამოსვლის შემდეგ:

- ა. ცხრილი ავტომატურად არ წაიშლება
- ბ. ცხრილი ავტომატურად წაიშლება

გ. ცხრილი არავტომატურად წაიშლება

2.1.18. გლობალური დროებითი ცხრილის სახელი:

- ა. # სიმბოლოებით არ იწყება
- ბ. ## სიმბოლოებით არ იწყება
- გ. ## სიმბოლოებით იწყება

2.1.19. გლობალური დროებითი ცხრილის შესაქმნელად:

- ა. საკმარისია ცხრილის სახელის წინ ## სიმბოლოების მითითება
- ბ. არ არის საკმარისი ცხრილის სახელის წინ ## სიმბოლოების მითითება
- გ. საკმარისია ცხრილის სახელის წინ # სიმბოლოს მითითება

2.1.20. გლობალურ ცხრილზე მიმართვა:

- ა. არ შეგვიძლია ნებისმიერი შერთებიდან
- ბ. შეგვიძლია ნებისმიერი შერთებიდან
- გ. დაუშვებელია ნებისმიერი შერთებიდან

2.1.21. დროებითი გლობალური ცხრილები:

- ა. არ არის განკუთვნილი მონაცემების გასაცვლელად სხვადასხვა პროგრამას შორის
- ბ. განკუთვნილია მონაცემების გასაცვლელად სხვადასხვა პროგრამას შორის
- გ. განკუთვნილია მხოლოდ ტრიგერებთან სამუშაოდ

2.1.22. გლობალური დროებითი ცხრილი:

- ა. არის იმ შეერთების დამთავრებამდე, რომელშიც ეს ცხრილი შეიქმნა
- ბ. არ არის იმ შეერთების დამთავრებამდე, რომელშიც ეს ცხრილი შეიქმნა
- გ. არის იმ შეერთების დამთავრებამდე, რომელშიც ეს ცხრილი არ შეიქმნა

2.1.23. გლობალური დროებითი ცხრილის წაშლა და შეცვლა:

- ა. არ შეგვიძლია სხვადასხვა შეერთებიდან
- ბ. შეუძლებელია
- გ. შეგვიძლია სხვადასხვა შეერთებიდან

2.1.24. დროებითი ცხრილი ყოველთვის იქმნება:

- ა. მიმდინარე სერვერზე tempdb მონაცემთა ბაზაში
- ბ. სხვა სერვერზე tempdb მონაცემთა ბაზაში
- გ. მიმდინარე სერვერზე model მონაცემთა ბაზაში

2.1.25. სერვერის გაჩერების შემთხვევაში:

- ა. ავტომატურად იშლება მხოლოდ ლოკალური ცხრილი
- ბ. ავტომატურად იშლება მხოლოდ გლობალური ცხრილი
- გ. ლოკალური და გლობალური დროებითი ცხრილები ავტომატურად იშლება

2.1.26. ცხრილის თითოეულ სვეტს განსაზღვრული ტიპი:

- ა. ხანდახან აქვს
- ბ. არ აქვს
- გ. აქვს

2.1.27. მიუთითებს თუ რა მნიშვნელობები იქნება მოთავსებული ცხრილის:

- ა. სვეტებში
- ბ. სტრიქონებში
- გ. ფუნქციებში

2.1.28. CHAR ტიპი:

- ა. აღწერს წილადს
- ბ. არ აღწერს ფიქსირებული სიგრძის სტრიქონს
- გ. აღწერს ფიქსირებული სიგრძის სტრიქონს

2.1.29. DATE ტიპი:

- ა. აღწერს სტრიქონს
- ბ. არ აღწერს თარიღს
- გ. აღწერს თარიღს

2.1.30. DATETIME ტიპი:

- ა. არ აღწერს დროსა და თარიღს
- ბ. აღწერს ერთ სიმბოლოს
- გ. აღწერს დროსა და თარიღს

2.1.31. FLOAT ტიპი:

- ა. აღწერს წილად რიცხვებს
- ბ. არ აღწერს წილად რიცხვებს
- გ. აღწერს მთელ რიცხვებს

2.1.32. IMAGE ტიპი:

- ა. არ აღწერს ორობით მონაცემებს
- ბ. აღწერს ორობით მონაცემებს
- გ. აღწერს მთელ რიცხვებს

2.1.33. INT ტიპი:

- ა. არ აღწერს 4 ბაიტი ზომის მთელ რიცხვებს
- ბ. აღწერს 4 ბაიტი ზომის მთელ რიცხვებს
- გ. აღწერს 2 ბაიტი ზომის მთელ რიცხვებს

2.1.34. NVARCHAR ტიპი:

- ა. აღწერს ცვლადი სიგრძის Unicode სიმბოლოების სტრიქონს ზომით 8000 სიმბოლომდე
- ბ. არ აღწერს ცვლადი სიგრძის Unicode სიმბოლოების სტრიქონს ზომით 8000 სიმბოლომდე
- გ. აღწერს წილად რიცხვებს

2.1.35. TABLE ტიპი:

- ა. გამოიყენება ცხრილების შესანახად
- ბ. არ გამოიყენება ცხრილების შესანახად
- გ. გამოიყენება პროცედურების შესანახად

2.1.36. მონაცემთა მომხმარებლის ტიპები:

- ა. არ იქმნება მონაცემების სისტემური ტიპების საფუძველზე
- ბ. იქმნება მონაცემების სისტემური ტიპების საფუძველზე
- გ. იქმნება მონაცემების არასისტემური ტიპების საფუძველზე

2.1.37. მომხმარებლის ტიპის სახელი უნიკალური:

- ა. არ უნდა იყოს მფლობელის ფარგლებში
- ბ. უნდა იყოს მფლობელის ფარგლებში
- გ. უნდა იყოს არამფლობელის ფარგლებში

2.1.38. სხვადასხვა მფლობელებს:

- ა. შეუძლიათ შექმნან ერთნაირი სახელის მქონე მომხმარებლის ტიპები
- ბ. არ შეუძლიათ შექმნან ერთნაირი სახელის მქონე მომხმარებლის ტიპები
- გ. არ შეუძლიათ შექმნან სხვადასხვა სახელის მქონე მომხმარებლის ტიპები

2.1.39. წარმოდგენა არის ვირტუალური ცხრილი, რომლის შემცველობა განისაზღვრება:

- ა. UPDATE მოთხოვნით
- ბ. DELETE მოთხოვნით
- გ. SELECT მოთხოვნით

2.1.40. წარმოდგენა შედგება:

- ა. მხოლოდ სვეტებისაგან
- ბ. სვეტებისა და სტრიქონებისაგან
- გ. მხოლოდ სტრიქონებისაგან

2.1.41. წარმოდგენა მონაცემებს:

- ა. არ შეიცავს
- ბ. შეიცავს
- გ. ხანდახან შეიცავს

2.1.42. უმარტივესი წარმოდგენა:

- ა. შექმნილია რამდენიმე ცხრილის ბაზაზე და არ შეიცავს ფილტრებს
- ბ. შექმნილია ერთი ცხრილის ბაზაზე და არ შეიცავს ფილტრებს
- გ. შექმნილია ერთი ცხრილის ბაზაზე და შეიცავს ფილტრებს

2.1.43. რთული წარმოდგენა შეიცავს:

- ა. მხოლოდ ერთი ცხრილის გაუფილტრავ ინფორმაციას
- ბ. მხოლოდ ერთ ცხრილს
- გ. რამდენიმე ცხრილის გაფილტრულ ინფორმაციას

2.1.44. შენახული პროცედურა არის:

- ა. ერთ მოდულში გაერთიანებული ბრძანებების ჯგუფი, რომელსაც სახელი აქვს
- ბ. ერთ მოდულში გაერთიანებული ბრძანებების ჯგუფი, რომელსაც სახელი არ აქვს
- გ. რამდენიმე მოდულში გაერთიანებული ბრძანებების ჯგუფი, რომელსაც სახელი არ აქვს

2.1.45. სისტემური შენახული პროცედურების სახელები:

- ა. არ იწყება sp_ პრეფიქსით
- ბ. იწყება sp_ პრეფიქსით
- გ. იწყება sys_ პრეფიქსით

2.1.46. საკუთარი შენახული პროცედურების შექმნა:

- ა. შესაძლებელია
- ბ. შეუძლებელია
- გ. ხანდახან შესაძლებელია

2.1.47. როცა შენახული პროცედურა პირველად სრულდება, მაშინ სერვერი:

- ა. არ ქმნის მისი შესრულების გეგმას
- ბ. ქმნის მისი შესრულების გეგმას
- გ. ქმნის მისი შეუსრულებლობის გეგმას

2.1.48. შენახული პროცედურის განმეორებით შესრულებისას, სერვერი:

- ა. არ იყენებს მეხსიერებაში მოთავსებულ გეგმას
- ბ. იყენებს სხვა სერვერზე მოთავსებულ გეგმას
- გ. იყენებს მეხსიერებაში მოთავსებულ გეგმას

2.1.49. ტრიგერები არის შენახული პროცედურის სპეციალური კლასი, რომლებიც ავტომატურად:

- ა. გაიშვება ცხრილებში მონაცემების დამატების, შეცვლის ან წაშლის დროს
- ბ. არ გაიშვება ცხრილებში მონაცემების დამატების, შეცვლის ან წაშლის დროს
- გ. გაიშვება ცხრილებში მონაცემების მხოლოდ ამორჩევის დროს

2.1.50. ტრიგერები სამ კატეგორიად იყოფა:

- ა. ამორჩევის, ჩამატებისა და წაშლის ტრიგერები
- ბ. შეცვლის, ჩამატებისა და წაშლის ტრიგერები
- გ. შეცვლის, ჩამატებისა და ამორჩევის ტრიგერები

2.1.51. ერთი ცხრილისთვის:

- ა. არ შეიძლება განსაზღვრულ იქნეს თითოეული ტიპის რამდენიმე ტრიგერი
- ბ. შეიძლება განსაზღვრულ იქნეს თითოეული ტიპის რამდენიმე ტრიგერი
- გ. შეიძლება განსაზღვრულ იქნეს თითოეული ტიპის ერთი ტრიგერი

2.1.52. ერთი ტრიგერიდან:

- ა. შეგვიძლია სხვა ტრიგერები გამოვიძახოთ
- ბ. არ შეგვიძლია სხვა ტრიგერები გამოვიძახოთ
- გ. სხვა ტრიგერები აუცილებლად უნდა გამოვიძახოთ

2.1.53. ინდექსი განკუთვნილია შესაბამის ცხრილში ან წარმოდგენაში ინფორმაციის:

- ა. ძებნის შესანელებლად
- ბ. შესაცვლელად
- გ. ძებნის დასაჩქარებლად

2.1.54. ინდექსი განისაზღვრება:

- ა. მხოლოდ ერთი სვეტისთვის
- ბ. ერთი ან მეტი სვეტისთვის
- გ. არც ერთი სვეტისთვის

2.1.55. ინდექსი:

- ა. არ შეიცავს ინდექსირებული სვეტის ან სვეტების დახარისხებულ მნიშვნელობებს საწყისი ცხრილის ან წარმოდგენის შესაბამის სტრიქონზე მიმართვებთან ერთად
- ბ. შეიცავს ინდექსირებული სვეტის ან სვეტების დახარისხებულ მნიშვნელობებს საწყისი ცხრილის ან წარმოდგენის შესაბამის სტრიქონზე მიმართვებთან ერთად
- გ. შეიცავს მხოლოდ წარმოდგენის შესაბამის სტრიქონზე მიმართვებს

2.1.56. მთლიანობის შეზღუდვა:

- ა. გამოიყენება მონაცემების მთლიანობის უზრუნველსაყოფად ლოგიკურ დონეზე
- ბ. გამოიყენება მონაცემების მთლიანობის უზრუნველსაყოფად ფიზიკურ დონეზე
- გ. არ გამოიყენება მონაცემების მთლიანობის უზრუნველსაყოფად ლოგიკურ დონეზე

2.1.57. მთლიანობის შეზღუდვა ესაა მექანიზმი, რომელიც:

- ა. უზრუნველყოფს ჩვენ მიერ განსაზღვრულ პირობებთან (ან შეზღუდვებთან) მონაცემების შესაბამისობის არავტომატურ კონტროლს
- ბ. არ უზრუნველყოფს ჩვენ მიერ განსაზღვრულ პირობებთან (ან შეზღუდვებთან) მონაცემების შესაბამისობის ავტომატურ კონტროლს
- გ. უზრუნველყოფს ჩვენ მიერ განსაზღვრულ პირობებთან (ან შეზღუდვებთან) მონაცემების შესაბამისობის ავტომატურ კონტროლს

2.1.58. მთლიანობის შეზღუდვა:

- ა. შეიძლება გამოყენებულ იქნეს სვეტის ან ცხრილის დონეზე
- ბ. არ შეიძლება გამოყენებულ იქნეს სვეტის ან ცხრილის დონეზე
- გ. შეიძლება გამოყენებულ იქნეს მხოლოდ სვეტის დონეზე

2.1.59. მთლიანობის შეზღუდვა, რომელიც სვეტის დონეზე გამოიყენება:

- ა. მოქმედებს მხოლოდ ამ სვეტში შესატან მონაცემებზე
- ბ. არ მოქმედებს ამ სვეტში შესატან მონაცემებზე
- გ. მოქმედებს სხვა სვეტში შესატან მონაცემებზე

2.1.60. თუ მთლიანობის შეზღუდვა გამოიყენება რამდენიმე სვეტის მიმართ, მაშინ შეზღუდვა:

- ა. მუშაობს მონაცემთა ბაზის დონეზე
- ბ. არ მუშაობს ცხრილის დონეზე
- გ. მუშაობს ცხრილის დონეზე

2.1.61. არსებობს მთლიანობის შეზღუდვის:

- ა. ორი ტიპი
- ბ. სამი ტიპი
- გ. ხუთი ტიპი

2.1.62. NULL მთლიანობის შეზღუდვა მოქმედებს:

- ა. მხოლოდ სვეტის დონეზე

- ბ. სვეტისა და მომხმარებლის ტიპის დონეზე
- გ. მონაცემთა ბაზის დონეზე

2.1.63. სვეტისა და მომხმარებლის ტიპისთვის:

- ა. შეგვიძლია განვსაზღვროთ მხოლოდ NULL შეზღუდვა
- ბ. შეგვიძლია განვსაზღვროთ NULL ან NOT NULL მთლიანობის შეზღუდვა
- გ. არ შეგვიძლია განვსაზღვროთ NULL ან NOT NULL მთლიანობის შეზღუდვა

2.1.64. თუ სვეტისთვის განსაზღვრულია NULL მთლიანობის შეზღუდვა, მაშინ:

- ა. სვეტში ნებადართულია NULL მნიშვნელობის შენახვა
- ბ. სვეტში არ არის ნებადართული NULL მნიშვნელობის შენახვა
- გ. ტრიგერში ნებადართულია NULL მნიშვნელობის შენახვა

2.1.65. თუ სვეტისთვის განსაზღვრულია NOT NULL მთლიანობის შეზღუდვა, მაშინ:

- ა. სვეტში არ არის ნებადართული NULL მნიშვნელობის შენახვა
- ბ. სვეტში ნებადართულია NULL მნიშვნელობის შენახვა
- გ. ტრიგერში ნებადართულია NULL მნიშვნელობის შენახვა

2.1.66. NULL არის სპეციალური მნიშვნელობა, რომელიც აღნიშნავს:

- ა. ტრიგერის არსებობას
- ბ. მნიშვნელობის ყოფნას
- გ. მნიშვნელობის არყოფნას

2.1.67. NULL არ არის იგივე, რაც:

- ა. მნიშვნელობის არყოფნა
- ბ. ინტერვალის სიმბოლო, ნული ან ნულოვანი სიგრძის სტრიქონი
- გ. ტრიგერის არარსებობა

2.1.68. NULL მნიშვნელობა შეიძლება შეფასდეს, როგორც:

- ა. UNKNOWN (უცნობი)
- ბ. TRUE
- გ. FALSE

2.1.69. CHECK მთლიანობის შეზღუდვა მოქმედებს:

- ა. ცხრილის დონეზე და არ ზღუდავს სვეტში შესანახი მნიშვნელობების დიაპაზონს
- ბ. ცხრილის დონეზე
- გ. სვეტის დონეზე და ზღუდავს სვეტში შესანახი მნიშვნელობების დიაპაზონს

2.1.70. თუ CHECK მთლიანობის შეზღუდვის შემოწმების შედეგია TRUE, მაშინ მონაცემების შეცვლა:

- ა. ნებადართული იქნება
- ბ. ნებადართული არ იქნება
- გ. აკრძალება ყველა მომხმარებელს

2.1.71. თუ CHECK მთლიანობის შეზღუდვის შემოწმების შედეგია FALSE, მაშინ ცვლილებები:

- ა. აკრძალება და შეტყობინება გაიცემა შეცდომის შესახებ

- ბ. არ აიკრძალება და შეტყობინება გაიცემა შეცდომის შესახებ
- გ. არ აიკრძალება და შეტყობინება არ გაიცემა შეცდომის შესახებ

2.1.72. ერთი სვეტისთვის შეგვიძლია შევქმნათ:

- ა. არც ერთი CHECK შეზღუდვა
- ბ. ერთი CHECK შეზღუდვა
- გ. რამდენიმე CHECK შეზღუდვა

2.1.73. UNIQUE მთლიანობის შეზღუდვა მოქმედებს:

- ა. ცხრილის დონეზე და იძლევა სვეტში მნიშვნელობების უნიკალურობის გარანტიას
- ბ. სვეტის დონეზე და იძლევა სვეტში მნიშვნელობების უნიკალურობის გარანტიას
- გ. ცხრილის დონეზე და არ იძლევა სვეტში მნიშვნელობების უნიკალურობის გარანტიას

2.1.74. UNIQUE მთლიანობის შეზღუდვა:

- ა. ხანდახან არ უშვებს NULL მნიშვნელობის არსებობას
- ბ. არ უშვებს NULL მნიშვნელობის არსებობას
- გ. უშვებს NULL მნიშვნელობის არსებობას

2.1.75. PRIMARY KEY შეზღუდვა მოქმედებს:

- ა. მხოლოდ სვეტის დონეზე
- ბ. სვეტის ან ცხრილის დონეზე
- გ. მხოლოდ ცხრილის დონეზე

2.1.76. პირველადი გასაღები:

- ა. შეიძლება შედგებოდეს ერთი ან მეტი სვეტისაგან და არის სტრიქონის უნიკალური იდენტიფიკატორი ცხრილის ფარგლებში
- ბ. არ შეიძლება შედგებოდეს ერთი ან მეტი სვეტისაგან და არის სტრიქონის უნიკალური იდენტიფიკატორი ცხრილის ფარგლებში
- გ. არ შეიძლება შედგებოდეს ერთი ან მეტი სვეტისაგან და არ არის სტრიქონის უნიკალური იდენტიფიკატორი ცხრილის ფარგლებში

2.1.77. თუ პირველადი გასაღები ერთი სვეტისაგან შედგება, მაშინ ამ სვეტისთვის:

- ა. უნდა იყოს დაყენებული UNIQUE შეზღუდვა
- ბ. არ უნდა იყოს დაყენებული UNIQUE შეზღუდვა
- გ. უნდა იყოს დაყენებული DEFAULT შეზღუდვა

2.1.78. თუ პირველადი გასაღები რამდენიმე სვეტისაგან შედგება, მაშინ თითოეულ სვეტში მნიშვნელობები:

- ა. არ შეიძლება გამეორდეს
- ბ. შეიძლება გამეორდეს
- გ. არ უნდა გამეორდეს

2.1.79. თუ პირველადი გასაღები რამდენიმე სვეტისაგან შედგება, მაშინ:

- ა. უნიკალური უნდა იყოს ამ სვეტების მნიშვნელობების კომბინაცია თითოეული სვეტისათვის
- ბ. უნიკალური არ უნდა იყოს ამ სვეტების მნიშვნელობების კომბინაცია თითოეული

სტრიქონისათვის

გ. უნიკალური უნდა იყოს ამ სვეტების მნიშვნელობების კომბინაცია თითოეული სტრიქონისათვის

2.1.80. პირველად გასაღებში შემავალი:

- ა. არც ერთი სვეტისათვის არ უნდა იყოს დაყენებული NULL შეზღუდვა
- ბ. ყველა სვეტისათვის არ უნდა იყოს დაყენებული NULL შეზღუდვა
- გ. ყველა სვეტისათვის უნდა იყოს დაყენებული NULL შეზღუდვა

2.1.81. ცხრილში შეგვიძლია შევქმნათ მხოლოდ:

- ა. ერთი პირველადი გასაღები
- ბ. ორი პირველადი გასაღები
- გ. სამი პირველადი გასაღები

2.1.82. FOREIGN KEY შეზღუდვა იქმნება:

- ა. სვეტის დონეზე
- ბ. ცხრილის დონეზე
- გ. სტრიქონის დონეზე

2.1.83. დამოკიდებული ცხრილის გარე გასაღები უკავშირდება:

- ა. დამოკიდებული ცხრილის პირველად გასაღებს
- ბ. მთავარი ცხრილის გარე გასაღებს
- გ. მთავარი ცხრილის პირველად გასაღებს

2.1.84. მთავარი ცხრილიდან შეუძლებელია სტრიქონის წაშლა, თუ მასთან:

- ა. დაკავშირებულია თუნდაც ერთი სტრიქონი დამოკიდებულ ცხრილში
- ბ. არ არის დაკავშირებული თუნდაც ერთი სტრიქონი დამოკიდებულ ცხრილში
- გ. დაკავშირებული არ არის არც ერთი სტრიქონი დამოკიდებულ ცხრილში

2.1.85. მთავარი ცხრილიდან სტრიქონის წაშლის წინ:

- ა. არ არის აუცილებელი წინასწარ წავშალოთ დამოკიდებული ცხრილის ყველა სტრიქონი
- ბ. აუცილებელია წინასწარ წავშალოთ დამოკიდებული ცხრილის ყველა სტრიქონი
- გ. აუცილებელია წინასწარ წავშალოთ მთავარი ცხრილის ყველა სტრიქონი

2.1.86. ნაგულისხმევია მნიშვნელობა, რომელიც:

- ა. ხელით მიენიჭება ცხრილის სვეტს სტრიქონის ჩასმის დროს, თუ ჩასმის ბრძანებაში ამკარად არ არის მითითებული ამ სვეტის მნიშვნელობა
- ბ. ავტომატურად არ მიენიჭება ცხრილის სვეტს სტრიქონის ჩასმის დროს, თუ ჩასმის ბრძანებაში ამკარად არ არის მითითებული ამ სვეტის მნიშვნელობა
- გ. ავტომატურად მიენიჭება ცხრილის სვეტს სტრიქონის ჩასმის დროს, თუ ჩასმის ბრძანებაში ამკარად არ არის მითითებული ამ სვეტის მნიშვნელობა

2.1.87. ნაგულისხმევი მნიშვნელობა:

- ა. შეიძლება იყოს მუდმივა, ჩადგმული ფუნქციის ან მათემატიკური გამოსახულების მიერ გაცემული შედეგი
- ბ. არ შეიძლება იყოს მუდმივა, ჩადგმული ფუნქციის ან მათემატიკური გამოსახულების

მიერ გაცემული შედეგი

გ. შეიძლება არ იყოს მუდმივა, ჩადგმული ფუნქციის ან მათემატიკური გამოსახულების მიერ გაცემული შედეგი

2.1.88. ფუნქცია არის კონსტრუქცია, რომელიც:

- ა. არ შეიცავს ხშირად გამოყენებად კოდს და აქვს სახელი
- ბ. შეიცავს ხშირად გამოყენებად კოდს და აქვს სახელი
- გ. შეიცავს ხშირად გამოყენებად კოდს და არ აქვს სახელი

2.1.89. არსებობს ფუნქციების ორი ტიპი:

- ა. მხოლოდ მომხმარებლის ფუნქციები
- ბ. მხოლოდ ჩადგმული ფუნქციები
- გ. ჩადგმული ფუნქციები და მომხმარებლის ფუნქციები

2.1.90. ჩადგმული ფუნქციები:

- ა. პროგრამირების გარემოს შემადგენელი ნაწილია და წინასწარგანსაზღვრულ მოქმედებებს ასრულებენ
- ბ. პროგრამირების გარემოს შემადგენელი ნაწილია და წინასწარგანსაზღვრულ მოქმედებებს არ ასრულებენ
- გ. არ არის პროგრამირების გარემოს შემადგენელი ნაწილია და წინასწარგანსაზღვრულ მოქმედებებს არ ასრულებენ

სისტემური მონაცემთა ბაზები. ფაილები და ფაილების ჯგუფები

2.2.1. სერვერის სისტემური მონაცემთა ბაზებია:

- ა. master; msdb; model; tempdb
- ბ. msdb; model; tempdb
- გ. master; msdb; model

2.2.2. სისტემურ ცხრილებთან მუშაობა დასაშვებია:

- ა. INSERT და DELETE ბრძანებების გამოყენებით
- ბ. SELECT ბრძანების გამოყენებით
- გ. UPDATE და DELETE ბრძანებების გამოყენებით

2.2.3. სისტემურ ცხრილებში მონაცემების შეცვლა:

- ა. შესაძლებელია ინდექსების გამოყენებით
- ბ. შეუძლებელია სისტემური შენახული პროცედურების გამოყენებით
- გ. შესაძლებელია სისტემური შენახული პროცედურების გამოყენებით

2.2.4. master მონაცემთა ბაზა შეიცავს:

- ა. სერვერის მთელ არასისტემურ ინფორმაციას, აგრეთვე ინფორმაციას სხვა მონაცემთა ბაზებისა და მათი პირველადი ფაილების ადგილმდებარეობის შესახებ
- ბ. სერვერის მთელ სისტემურ ინფორმაციას, აგრეთვე ინფორმაციას სხვა მონაცემთა ბაზებისა და მათი პირველადი ფაილების ადგილმდებარეობის შესახებ
- გ. სერვერის მთელ სისტემურ ინფორმაციას, აგრეთვე ინფორმაციას სხვა მონაცემთა

ბაზებისა და მათი მეორეული ფაილების ადგილმდებარეობის შესახებ

2.2.5. master მონაცემთა ბაზა შემდეგი ფაილებისგან შედგება:

- ა. Master.mdf და Mastlog.ldf
- ბ. msdbdata.mdf და msdblog.ldf
- გ. tempdb.mdf და templogldf

2.2.6. master სისტემურ მონაცემთა ბაზაში მომხმარებლის ობიექტების შექმნა:

- ა. ხანდახან დასაშვებია
- ბ. დასაშვებია
- გ. დაუშვებელია

2.2.7. tempdb მონაცემთა ბაზა:

- ა. არ შეიცავს დროებით ობიექტებს
- ბ. შეიცავს დროებით ობიექტებს
- გ. შეიცავს მუდმივ ობიექტებს

2.2.8. tempdb მონაცემთა ბაზა ორი ფაილისგან შედგება:

- ა. tempdb.mdf და templogldf
- ბ. Master.mdf და Mastlog.ldf
- გ. model.mdf და model.ldf

2.2.9. სერვერზე ახალი მონაცემთა ბაზის შექმნა ხდება მასში:

- ა. master სისტემური მონაცემთა ბაზის ობიექტების გადაწერის გზით
- ბ. tempdb სისტემური მონაცემთა ბაზის ობიექტების გადაწერის გზით
- გ. model სისტემური მონაცემთა ბაზის ობიექტების გადაწერის გზით

2.2.10. model სისტემური მონაცემთა ბაზა ორი ფაილისგან შედგება:

- ა. model.mdf და model.ldf
- ბ. Master.mdf და Mastlog.ldf
- გ. tempdb.mdf და templogldf

2.2.11. msdb მონაცემთა ბაზა ინახავს მთელ ინფორმაციას, რომელიც:

- ა. ეხება ადმინისტრირების ავტომატიზებასა და სერვერის მართვას
- ბ. არ ეხება ადმინისტრირების ავტომატიზებასა და სერვერის მართვას
- გ. ეხება ადმინისტრირების ხელით შესრულებას

2.2.12. msdb სისტემური მონაცემთა ბაზა გამოიყენება:

- ა. SQL Server Agent სამსახურის მიერ მხოლოდ მოვლენების (alerts) დაგეგმვისათვის
- ბ. SQL Server Agent სამსახურის მიერ მოვლენების (alerts), ამოცანების (jobs) და ოპერატორების (operators) რეგისტრირების დაგეგმვისათვის
- გ. SQL Server Agent სამსახურის მიერ მხოლოდ ამოცანების (jobs) დაგეგმვისათვის

2.2.13. msdb ბაზა ორი ფაილისგან შედგება:

- ა. tempdb.mdf და templogldf
- ბ. msdbdata.mdf და msdblog.ldf

გ. Master.mdf და Mastlog.ldf

2.2.14. თითოეული მონაცემთა ბაზა მინიმუმ:

- ა. სამი ფაილისაგან შედგება
- ბ. ორი ფაილისაგან შედგება
- გ. ერთი ფაილისაგან შედგება

2.2.15. თითოეული მონაცემთა ბაზა შემდეგი ფაილებისაგან შედგება:

- ა. მხოლოდ მონაცემთა ბაზის ფაილებისაგან
- ბ. მხოლოდ ტრანზაქციების ჟურნალის ფაილებისაგან
- გ. მონაცემთა ბაზისა და ტრანზაქციების ჟურნალის ფაილებისაგან

2.2.16. მონაცემთა ბაზის შესაძლებელი შემდეგი ტიპის ფაილები გამოიყენება:

- ა. Secondary და Transaction Log
- ბ. Primary, Secondary და Transaction Log
- გ. Primary და Secondary

2.2.17. მონაცემთა ბაზაში გამოყენებულ თითოეულ ფაილს აქვს:

- ა. ფაილის ლოგიკური და ფაილის ფიზიკური სახელები
- ბ. ფაილის ლოგიკური სახელი
- გ. ფაილის ფიზიკური სახელი

2.2.18. მონაცემთა ბაზის:

- ა. ყველა ფაილის ზომა შეიძლება ავტომატურად გაიზარდოს
- ბ. ზოგიერთი ფაილის ზომა შეიძლება ავტომატურად გაიზარდოს
- გ. ყველა ფაილის ზომა შეიძლება ავტომატურად არ გაიზარდოს

2.2.19. არსებობს ფაილების შემდეგი ტიპის ჯგუფები:

- ა. ფაილების ძირითადი ჯგუფი და მომხმარებლების მიერ შექმნილი ფაილების ჯგუფი
- ბ. მომხმარებლების მიერ შექმნილი ფაილების ჯგუფი და ფაილების ნაგულისხმევი ჯგუფი
- გ. ფაილების ძირითადი ჯგუფი, მომხმარებლების მიერ შექმნილი ფაილების ჯგუფი და ფაილების ნაგულისხმევი ჯგუფი

2.2.20. ფაილების ძირითადი ჯგუფი შეიცავს:

- ა. primary ტიპის ფაილსა და ყველა ფაილს, რომლებიც არ არის ჩართული სხვა ჯგუფში
- ბ. მხოლოდ primary ტიპის ფაილს
- გ. primary ტიპის ფაილსა და ყველა ფაილს, რომლებიც ჩართულია სხვა ჯგუფში

2.2.21. მონაცემთა ბაზას:

- ა. შეიძლება ჰქონდეს ფაილების მხოლოდ ერთი ძირითადი ჯგუფი
- ბ. არ შეიძლება ჰქონდეს ფაილების მხოლოდ ერთი ძირითადი ჯგუფი
- გ. შეიძლება ჰქონდეს ფაილების მხოლოდ ორი ძირითადი ჯგუფი

2.2.22. მონაცემთა ბაზაში:

- ა. არ შეიძლება შევქმნათ მომხმარებლების მიერ განსაზღვრული ფაილების რამდენიმე ჯგუფი ფაილების ნებისმიერი შედგენილობით

ბ. შეიძლება შევქმნათ მომხმარებლების მიერ განსაზღვრული ფაილების რამდენიმე ჯგუფი ფაილების ნებისმიერი შედგენილობით

გ. შეიძლება შევქმნათ მომხმარებლების მიერ განსაზღვრული ფაილების მხოლოდ ერთი ჯგუფი ფაილების ნებისმიერი შედგენილობით

2.2.23. მონაცემთა ბაზაში ფაილების:

ა. სამი ჯგუფია ნაგულისხმევი

ბ. ორი ჯგუფია ნაგულისხმევი

გ. ერთ-ერთი ჯგუფია ნაგულისხმევი

2.2.24. თუ მონაცემთა ბაზის შექმნის დროს არ არის მითითებული ფაილების ასეთი ჯგუფი, მაშინ:

ა. ფაილების ძირითადი ჯგუფი არ ხდება ნაგულისხმევი

ბ. ფაილების ძირითადი ჯგუფი ხდება ნაგულისხმევი

გ. ფაილების მეორეული ჯგუფი ხდება ნაგულისხმევი

2.2.25. თუ მონაცემთა ბაზის ობიექტის (ცხრილის ან სვეტის) შექმნის დროს აშკარად არ არის მითითებული თუ ფაილების რომელ ჯგუფს ეკუთვნის ის, მაშინ:

ა. ეს ობიექტი შეიქმნება ფაილების ნაგულისხმევ ჯგუფში

ბ. ეს ობიექტი შეიქმნება ფაილების ძირითად ჯგუფში

გ. ეს ობიექტი არ შეიქმნება ფაილების ნაგულისხმევ ჯგუფში

2.2.26. მონაცემთა ბაზის მფლობელმა:

ა. შეიძლება ფაილების მხოლოდ განსაზღვრული ჯგუფი დანიშნოს ფაილების ნაგულისხმევ ჯგუფად

ბ. შეიძლება ფაილების ნებისმიერი ჯგუფი დანიშნოს ფაილების ნაგულისხმევ ჯგუფად

გ. არ შეიძლება ფაილების ნებისმიერი ჯგუფი დანიშნოს ფაილების ნაგულისხმევ ჯგუფად

2.2.27. ფაილების მხოლოდ:

ა. ერთი ჯგუფი შეიძლება იყოს ნაგულისხმევი

ბ. ორი ჯგუფი შეიძლება იყოს ნაგულისხმევი

გ. სამი ჯგუფი შეიძლება იყოს ნაგულისხმევი

2.2.28. ნებისმიერი ფაილი შეიძლება ჩართული იყოს ფაილების მხოლოდ:

ა. სამ ჯგუფში

ბ. ორ ჯგუფში

გ. ერთ ჯგუფში

2.2.29. ფაილების ჯგუფის განსაზღვრა ტრანზაქციების ჟურნალის ფაილებისთვის:

ა. არ შეიძლება

ბ. შეიძლება

გ. ხანდახან შეიძლება

2.2.30. თუ მონაცემთა ბაზაში არ იქმნება ფაილების ჯგუფები, მაშინ:

ა. ყველა ობიექტი არ მოთავსდება ფაილების ძირითად ჯგუფში

ბ. ყველა ობიექტი მოთავსდება ფაილების ძირითად ჯგუფში

გ. ყველა ობიექტი მოთავსდება ფაილების ნაგულისხმევ ჯგუფში

მონაცემთა ბაზების შექმნა, წაშლა და სახელის შეცვლა

2.3.1. სერვერთან შეერთებისას კავშირი:

- ა. არ მყარდება კონკრეტულ მონაცემთა ბაზასთან
- ბ. მყარდება კონკრეტულ მონაცემთა ბაზასთან
- გ. მყარდება კონკრეტულ ტრიგერთან

2.3.2. სერვერთან შეერთებისას კავშირი მყარდება კონკრეტულ მონაცემთა ბაზასთან, რომელსაც:

- ა. მიმდინარე მონაცემთა ბაზა ეწოდება
- ბ. მიმდინარე მონაცემთა ბაზა არ ეწოდება
- გ. master მონაცემთა ბაზა ეწოდება

2.3.3. სხვა მონაცემთა ბაზაზე გადართვისთვის გამოიყენება:

- ა. SELECT ბრძანება
- ბ. UPDATE ბრძანება
- გ. USE ბრძანება

2.3.4. USE Baza_1 ბრძანების შესრულების შედეგად:

- ა. მიმდინარე გახდება Baza_1 მონაცემთა ბაზა
- ბ. დაიხურება Baza_1 მონაცემთა ბაზა
- გ. გაიფილტრება Baza_1 მონაცემთა ბაზა

2.3.5. მონაცემთა ბაზის შექმნა შესაძლებელია:

- ა. როგორც Transact-SQL-ის ბრძანებების, ისე გრაფიკული ინტერფეისის საშუალებით
- ბ. მხოლოდ Transact-SQL-ის ბრძანებების საშუალებით
- გ. მხოლოდ გრაფიკული ინტერფეისის საშუალებით

2.3.6. მონაცემთა ბაზის შექმნისას master მონაცემთა ბაზის sys.databases სისტემურ წარმოდგენაში:

- ა. შეიტანება მხოლოდ მონაცემთა ბაზის სახელი
- ბ. არ შეიტანება მონაცემთა ბაზის სახელი და შესაბამისი მონაცემები
- გ. შეიტანება მონაცემთა ბაზის სახელი და შესაბამისი მონაცემები

2.3.7. მონაცემთა ბაზის შემქმნელი მომხმარებელი ავტომატურად:

- ა. კარგავს მასთან მუშაობის უფლებას
- ბ. არ ხდება მისი მფლობელი
- გ. ხდება მისი მფლობელი

2.3.8. ჩვეულებრივ, მონაცემთა ბაზის შექმნა:

- ა. არ შეუძლიათ სერვერის sysadmin და dbcreator ფიქსირებული როლების წევრებს
- ბ. შეუძლიათ სერვერის sysadmin და dbcreator ფიქსირებული როლების წევრებს
- გ. შეუძლია ყველას

2.3.9. მომხმარებელს, რომელიც არ არის ადმინისტრატორი:

- ა. ეკრძალება მონაცემთა ბაზის შექმნა
- ბ. არ შეიძლება მიეცეს მონაცემთა ბაზების შექმნის უფლება
- გ. შეიძლება მიეცეს მონაცემთა ბაზების შექმნის უფლება

2.3.10. მონაცემთა ბაზის შექმნისას:

- ა. master მონაცემთა ბაზის sys.databases სისტემურ წარმოდგენაში შეიტანება შესაბამისი მონაცემები
- ბ. tempdb მონაცემთა ბაზის sys.databases სისტემურ წარმოდგენაში შეიტანება შესაბამისი მონაცემები
- გ. model მონაცემთა ბაზის sys.databases სისტემურ წარმოდგენაში შეიტანება შესაბამისი მონაცემები

2.3.11. თუ მონაცემთა ბაზა შეიცავს ინტერვალებს ან სხვა დაუშვებელ სიმბოლოებს, მაშინ მისი სახელი უნდა მოვათავსოთ:

- ა. მრგვალ ფრჩხილებში
- ბ. შემზღუდავ სიმბოლოებში
- გ. ფიგურულ ფრჩხილებში

2.3.12. მონაცემთა ბაზის სახელი:

- ა. შეიძლება იყოს უნიკალური სერვერის ფარგლებში
- ბ. არ უნდა იყოს უნიკალური სერვერის ფარგლებში
- გ. უნდა იყოს უნიკალური სერვერის ფარგლებში

2.3.13. თუ ტრანზაქციების ჟურნალის სახელი მითითებული არ არის, მაშინ სერვერი ტრანზაქციების ჟურნალის სახელად იღებს:

- ა. მონაცემთა ბაზის სახელს და არაფერს არ უმატებს
- ბ. მონაცემთა ბაზის სახელს და უმატებს _Log სიმბოლოებს
- გ. ნებისმიერ სახელს და უმატებს _Log სიმბოლოებს

2.3.14. თუ ტრანზაქციების ჟურნალის ფაილის სახელი არ არის მითითებული, მაშინ:

- ა. სერვერი ავტომატურად ქმნის ტრანზაქციების ჟურნალის სამ ფაილს
- ბ. სერვერი ავტომატურად ქმნის ტრანზაქციების ჟურნალის ორ ფაილს
- გ. სერვერი ავტომატურად ქმნის ტრანზაქციების ჟურნალის ერთ ფაილს

2.3.15. მონაცემთა ბაზის მისაერთებლად:

- ა. საკმარისია მხოლოდ მონაცემთა ბაზის პირველადი ფაილის ადგილმდებარეობის მითითება
- ბ. არ არის საკმარისი მხოლოდ მონაცემთა ბაზის პირველადი ფაილის ადგილმდებარეობის მითითება
- გ. საკმარისია მხოლოდ მონაცემთა ბაზის მეორეული ფაილის ადგილმდებარეობის მითითება

2.3.16. ინფორმაცია მონაცემთა ბაზის ფაილების ადგილმდებარეობის შესახებ:

- ა. ინახება მონაცემთა ბაზის პირველად ფაილში
- ბ. არ ინახება მონაცემთა ბაზის პირველად ფაილში

გ. ინახება მონაცემთა ბაზის ტრანზაქციების ჟურნალის ფაილში

2.3.17. CREATE DATABASE ბრძანების ON არგუმენტი მიუთითებს, რომ:

- ა. არ იწყება მონაცემთა ბაზის ფაილების განსაზღვრა
- ბ. იწყება მონაცემთა ბაზის ფაილების განსაზღვრა
- გ. იწყება მონაცემთა ბაზის ინდექსების განსაზღვრა

2.3.18. CREATE DATABASE ბრძანების PRIMARY არგუმენტი მიუთითებს, რომ:

- ა. არ იწყება მონაცემთა ბაზის პირველადი ფაილის აღწერა
- ბ. მთავრდება მონაცემთა ბაზის პირველადი ფაილის აღწერა
- გ. იწყება მონაცემთა ბაზის პირველადი ფაილის აღწერა

2.3.19. CREATE DATABASE ბრძანების LOG ON არგუმენტი:

- ა. არ განსაზღვრავს ტრანზაქციების ჟურნალის ფაილებს
- ბ. განსაზღვრავს ტრანზაქციების ჟურნალის ფაილებს
- გ. განსაზღვრავს პირველად ფაილს

2.3.20. თუ CREATE DATABASE ბრძანების LOG ON არგუმენტი არ არის მითითებული, მაშინ სერვერი:

- ა. ავტომატურად ქმნის ტრანზაქციების ჟურნალის ერთ ფაილს
- ბ. ავტომატურად ქმნის ტრანზაქციების ჟურნალის ორ ფაილს
- გ. ავტომატურად ქმნის ტრანზაქციების ჟურნალის სამ ფაილს

2.3.21. CREATE DATABASE ბრძანების FOR ATTACH არგუმენტი გამოიყენება მაშინ, როცა:

- ა. უნდა შესრულდეს სერვერთან ტრიგერის მიერთება
- ბ. არ უნდა შესრულდეს სერვერთან მონაცემთა ბაზის მიერთება
- გ. უნდა შესრულდეს სერვერთან მონაცემთა ბაზის მიერთება

2.3.22. მონაცემთა ბაზის მისაერთებლად:

- ა. არასაკმარისია მხოლოდ მონაცემთა ბაზის პირველადი ფაილის ადგილმდებარეობის მითითება
- ბ. საკმარისია მხოლოდ მონაცემთა ბაზის პირველადი ფაილის ადგილმდებარეობის მითითება
- გ. საკმარისია მხოლოდ მონაცემთა ბაზის მეორეული ფაილის ადგილმდებარეობის მითითება

2.3.23. სერვერთან მონაცემთა ბაზის მიერთება შეიძლება შესრულდეს:

- ა. sp_attach_db სისტემური შენახული პროცედურის გამოყენებით
- ბ. sp_attach სისტემური შენახული პროცედურის გამოყენებით
- გ. sp_renamedb შენახული პროცედურის გამოყენებით

2.3.24. CREATE DATABASE ბრძანების NAME არგუმენტი განსაზღვრავს:

- ა. ფაილის ლოგურ სახელს
- ბ. ფაილის ფიზიკურ სახელს
- გ. ფაილის საიდენტიფიკაციო ნომერს

2.3.25. CREATE DATABASE ბრძანების FILENAME არგუმენტი განსაზღვრავს:

- ა. ფაილის საიდენტიფიკაციო ნომერს
- ბ. ფაილის ლოგურ სახელს
- გ. ფაილის ფიზიკურ სახელს

2.3.26. CREATE DATABASE ბრძანების SIZE არგუმენტი განსაზღვრავს:

- ა. ფაილის საწყის ზომას მეგაბაიტებში
- ბ. ფაილის საწყის ზომას ბაიტებში
- გ. კატალოგის ზომას მეგაბაიტებში

2.3.27. CREATE DATABASE ბრძანების FILEGROWTH არგუმენტი განსაზღვრავს:

- ა. ნაზარდის ზომას კატალოგისათვის
- ბ. ნაზარდის ზომას ფაილისათვის
- გ. ნაზარდის ზომას ტრიგერისათვის

2.3.28. ნაზარდის ზომა:

- ა. უნდა მივუთითოთ მხოლოდ გიგაბაიტებში
- ბ. არ შეიძლება მივუთითოთ მეგაბაიტებში ან პროცენტებში
- გ. შეიძლება მივუთითოთ მეგაბაიტებში ან პროცენტებში

2.3.29. ავტომატურად ნაზარდის ზომა აიღება:

- ა. 1 ბაიტი
- ბ. 1 მეგაბაიტი
- გ. 1 კილობაიტი

2.3.30. თუ ნაზარდის ზომა მითითებული არ არის, მაშინ ფაილის ზომა გაიზრდება საწყისი ზომის:

- ა. 10%-ით
- ბ. 20%-ით
- გ. 30%-ით

2.3.31. CREATE DATABASE ბრძანების MAXSIZE არგუმენტი განსაზღვრავს:

- ა. ფაილის მინიმალურ ზომას
- ბ. ფაილის მაქსიმალურ ზომას
- გ. ფაილის საშუალო ზომას

2.3.32. ფაილის მაქსიმალური ზომა ეთითება:

- ა. ბაიტებში
- ბ. კილობაიტებში
- გ. მეგაბაიტებში

2.3.33. ერთი ფაილის ზომის შეზღუდვა:

- ა. არ ზღუდავს მონაცემთა ბაზის ზომის ზრდას
- ბ. ზღუდავს მონაცემთა ბაზის ზომის ზრდას
- გ. ზღუდავს ფუნქციის ზომას

- 2.3.34. ერთი ფაილის ზომის შეზღუდვა არ ზღუდავს მონაცემთა ბაზის ზომის ზრდას, რადგან:
- ა. მონაცემთა ბაზის ზომა არ შეიძლება გაიზარდოს სხვა ფაილების ზომის ზრდის ხარჯზე
 - ბ. მონაცემთა ბაზის ზომა შეიძლება გაიზარდოს სხვა ფაილების ზომის ზრდის ხარჯზე
 - გ. მონაცემთა ბაზის ზომა შეიძლება არ გაიზარდოს სხვა ფაილების ზომის ზრდის ხარჯზე

2.3.35. CREATE DATABASE ბრძანების MAXSIZE არგუმენტის UNLIMITED მნიშვნელობა მუთითებს, რომ:

- ა. კატალოგის ზომა უნდა შეიზღუდოს
- ბ. ფაილის ზომა უნდა შეიზღუდოს
- გ. ფაილის ზომა არ უნდა შეიზღუდოს

2.3.36. თუ ფაილის ზომა არ უნდა შეიზღუდოს, მაშინ:

- ა. უნდა გამოვტოვოთ MAXSIZE არგუმენტი
- ბ. არ უნდა გამოვტოვოთ MAXSIZE არგუმენტი
- გ. აუცილებელია MAXSIZE არგუმენტის მითითება

2.3.37. ფაილები, რომლებიც არ ეკუთვნის არც ერთ ჯგუფს, ჩართული:

- ა. არ იქნება ფაილების ავტომატურ ჯგუფში
- ბ. იქნება ფაილების ავტომატურ ჯგუფში
- გ. იქნება ფაილების მეორეულ ჯგუფში

2.3.38. CREATE DATABASE ბრძანების FILEGROUP არგუმენტი განსაზღვრავს:

- ა. არც ერთ ფაილს
- ბ. ერთ ფაილს
- გ. ფაილების ჯგუფს

2.3.39. მონაცემთა ბაზის წასაშლელად გამოიყენება:

- ა. CREATE DATABASE ბრძანება
- ბ. DROP DATABASE ბრძანება
- გ. ALTER DATABASE ბრძანება

2.3.40. ერთი DROP DATABASE ბრძანებით შესაძლებელია:

- ა. რამდენიმე მონაცემთა ბაზის წაშლა
- ბ. მხოლოდ ერთი მონაცემთა ბაზის წაშლა
- გ. მხოლოდ ორი მონაცემთა ბაზის წაშლა

2.3.41. მონაცემთა ბაზის სახელის შესაცვლელად გამოიყენება:

- ა. sp_renamedb შენახული პროცედურა
- ბ. sp_attach_db სისტემური შენახული პროცედურა
- გ. DROP DATABASE ბრძანება

2.3.42. sp_renamedb შენახული პროცედურის შესრულების უფლება:

- ა. აქვს სერვერის sysadmin ფიქსირებული როლის წევრებს
- ბ. არ აქვს სერვერის sysadmin ფიქსირებული როლის წევრებს
- გ. არავის არ აქვს

2.3.43. CREATE DATABASE Shekveta; ბრძანების შესრულების შედეგად:

- ა. წაიშლება Shekveta მონაცემთა ბაზა
- ბ. არ შეიქმნება Shekveta მონაცემთა ბაზა
- გ. შეიქმნება Shekveta მონაცემთა ბაზა

2.3.44. Shekveta მონაცემთა ბაზის შესაქმნელად უნდა შევასრულოთ ბრძანება:

- ა. DROP DATABASE Shekveta;
- ბ. CREATE DATABASE Shekveta;
- გ. CREATE DATABASE Personali;

2.3.45.

```
CREATE DATABASE Baza2
```

```
ON
```

```
(
```

```
NAME = Baza2,
```

```
FILENAME =
```

```
'C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA\Baza2.mdf',
```

```
SIZE = 15MB,
```

```
MAXSIZE = 36MB,
```

```
FILEGROWTH = 3MB
```

```
)
```

```
LOG ON
```

```
( NAME = Baza2_log,
```

```
FILENAME =
```

```
'C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA\Baza2_Log.ldf',
```

```
SIZE = 3MB,
```

```
MAXSIZE = 25MB,
```

```
FILEGROWTH = 2MB );
```

ბრძანების შესრულების შედეგად:

- ა. შეიქმნება Baza2 მონაცემთა ბაზა
- ბ. არ შეიქმნება Baza2 მონაცემთა ბაზა
- გ. წაიშლება Baza2 მონაცემთა ბაზა

2.3.46.

```
DECLARE @data_path NVARCHAR(256);
```

```
SET @data_path =
```

```
(
```

```
SELECT SUBSTRING(physical_name, 1, CHARINDEX(N'master.mdf', LOWER(physical_name)) - 1)
```

```
FROM master.sys.master_files
```

```
WHERE database_id = 1 AND file_id = 1
```

```
);
```

```
--
```

```
EXEC (
```

```
'CREATE DATABASE Baza1
```

```
ON
```

```
(
```

FILENAME = '"+ @data_path + 'Baza1_dat1.mdf'

)

FOR ATTACH'

); კოდის შესრულების შედეგად:

- ა. შესრულდება Baza1 მონაცემთა ბაზის მიერთება სერვერთან
- ბ. არ შესრულდება Baza1 მონაცემთა ბაზის მიერთება სერვერთან
- გ. შესრულდება Baza1 მონაცემთა ბაზის გამორთვა სერვერიდან

2.3.47. DROP DATABASE Baza1; ბრძანების შესრულების შედეგად:

- ა. შეიქმნება Baza1 მონაცემთა ბაზა
- ბ. წაიშლება Baza1 მონაცემთა ბაზა
- გ. არ წაიშლება Baza1 მონაცემთა ბაზა

2.3.48. DROP DATABASE Baza1, Baza2, Baza3; ბრძანების შესრულების შედეგად:

- ა. წაიშლება Baza1 მონაცემთა ბაზა
- ბ. წაიშლება Baza2 და Baza3 მონაცემთა ბაზები
- გ. წაიშლება Baza1, Baza2 და Baza3 მონაცემთა ბაზები

2.3.49. Baza1 და Baza2 მონაცემთა ბაზების წასაშლელად უნდა შევასრულოთ ბრძანება:

- ა. DROP DATABASE Baza1, Baza2;
- ბ. DROP DATABASE Baza1, Baza2, Baza3;
- გ. DROP DATABASE Baza3;

2.3.50. EXEC sp_renamedb 'Baza1', 'Baza2'; შენახული პროცედურის შესრულების შედეგად:

- ა. Baza2 მონაცემთა ბაზას დაერქმევა ახალი Baza1 სახელი
- ბ. Baza1 მონაცემთა ბაზას დაერქმევა ახალი Baza2 სახელი
- გ. წაიშლება Baza1 მონაცემთა ბაზა

2.3.51. მისათვის, რომ Baza1 მონაცემთა ბაზას დავარქვათ ახალი Baza2 სახელი, უნდა შევასრულოთ შენახული პროცედურა:

- ა. CREATE DATABASE Baza1;
- ბ. DROP DATABASE Baza1, Baza2;
- გ. sp_renamedb 'Baza1', 'Baza2';

მონაცემთა ბაზის მიერთება და გამორთვა. ინფორმაციის მიღება მონაცემთა ბაზების შესახებ

2.4.1. მონაცემთა ბაზის სერვერთან მიერთება და სერვერიდან გამორთვა შეგვიძლია გამოვიყენოთ მონაცემთა ბაზის:

- ა. შესაცვლელად
- ბ. გადასატანად ერთი სერვერიდან მეორეზე
- გ. წასაშლელად

2.4.2. მონაცემთა ბაზების მიერთებისა და გამორთვის ოპერაციების შესრულება შეუძლია:

- ა. არც ერთ მომხმარებელს
- ბ. ნებისმიერ მომხმარებელს

გ. მხოლოდ სერვერის sysadmin როლის წევრებს

2.4.3. მონაცემთა ბაზების მიერთებისა და გამორთვის ოპერაციების შესრულების უფლების გადაცემა:

- ა. შეიძლება სხვა მომხმარებლისათვის
- ბ. არ შეიძლება სხვა მომხმარებლისათვის
- გ. ხანდახან შეიძლება სხვა მომხმარებლისათვის

2.4.4. იმისათვის, რომ მომხმარებელმა შეძლოს მონაცემთა ბაზების მიერთებისა და გამორთვის ოპერაციების შესრულება, ის:

- ა. უნდა წავშალოთ
- ბ. არ უნდა ჩავრთოთ ამ როლში
- გ. უნდა ჩავრთოთ ამ როლში

2.4.5. სერვერიდან გამორთვის შემდეგ:

- ა. ველარ შევძლებთ მონაცემთა ბაზასთან მიმართვას და მასთან მუშაობას
- ბ. შევძლებთ მონაცემთა ბაზასთან მიმართვას და მასთან მუშაობას
- გ. ველარ შევძლებთ მონაცემთა ბაზასთან მიმართვას და მაგრამ შევძლებთ მასთან მუშაობას

2.4.6. მონაცემთა ბაზის გამოსართავად გამოიყენება:

- ა. sp_renamedb შენახული პროცედურა
- ბ. sp_detach_db შენახული პროცედურა
- გ. sp_attach_db სისტემური შენახული პროცედურა

2.4.7. მონაცემთა ბაზის მიერთება არის პროცესი, როცა:

- ა. master მონაცემთა ბაზის sysdatabases წარმოდგენაში არ იქმნება ახალი სტრიქონი, რომელიც შეიცავს მომხმარებლის მონაცემთა ბაზის აღწერას თვით მონაცემთა ბაზის შექმნის გარეშე
- ბ. master მონაცემთა ბაზის sysdatabases წარმოდგენაში იქმნება ახალი სტრიქონი, რომელიც შეიცავს მომხმარებლის მონაცემთა ბაზის აღწერას თვით მონაცემთა ბაზის შექმნის გარეშე
- გ. master მონაცემთა ბაზის sysdatabases წარმოდგენაში იქმნება ახალი სტრიქონი, რომელიც არ შეიცავს მომხმარებლის მონაცემთა ბაზის აღწერას თვით მონაცემთა ბაზის შექმნის გარეშე

2.4.8. სერვერთან მონაცემთა ბაზის მისაერთებლად გამოიყენება:

- ა. sp_detach_db შენახული პროცედურა
- ბ. sp_renamedb შენახული პროცედურა
- გ. sp_attach_db სისტემური შენახული პროცედურა

2.4.9. თუ მისაერთებელია 16-ზე მეტი ფაილი, მაშინ:

- ა. არ უნდა გამოვიყენოთ CREATE DATABASE FOR ATTACH ბრძანება
- ბ. უნდა გამოვიყენოთ CREATE DATABASE FOR ATTACH ბრძანება
- გ. უნდა გამოვიყენოთ sp_renamedb შენახული პროცედურა

2.4.10. სერვერთან მონაცემთა ბაზის მიერთებისას ტრანზაქციების ჟურნალის მიერთება:

- ა. აუცილებელი არ არის
- ბ. აუცილებელია

გ. ზოგჯერ არის აუცილებელი

2.4.11. მიერთების დროს:

- ა. ხდება, მონაცემების მთლიანობის შემოწმება და მოწმდება მონაცემთა ბაზის მომხმარებლების კავშირების კორექტულობა სააღრიცხვო ჩანაწერებთან
- ბ. ხდება, მონაცემების მთლიანობის შემოწმება და არ მოწმდება მონაცემთა ბაზის მომხმარებლების კავშირების კორექტულობა სააღრიცხვო ჩანაწერებთან
- გ. არ ხდება მონაცემების მთლიანობის შემოწმება და არ მოწმდება მონაცემთა ბაზის მომხმარებლების კავშირების კორექტულობა სააღრიცხვო ჩანაწერებთან

2.4.12. მონაცემთა ბაზების შესახებ ინფორმაციის მისაღებად შეგვიძლია გამოვიყენოთ:

- ა. sp_changedb ბრძანება
- ბ. sp_help შენახული პროცედურა
- გ. sp_helpdb შენახული პროცედურა

2.4.13. sp_helpdb ბრძანების სინტაქსია:

- ა. sp_helpdb [[@dbname =] 'მონაცემთა_ბაზის_სახელი']
- ბ. sp_helpdb [@dbname
- გ. sp_helpdb 'მონაცემთა_ბაზის_სახელი']

2.4.14. sp_helpdb ბრძანება გაცემს:

- ა. ცხრილის ზომას, მონაცემთა ბაზის შექმნის თარიღს და მის მიმდინარე სტატუსს
- ბ. მონაცემთა ბაზის მიმდინარე ზომას, მონაცემთა ბაზის მფლობელის სააღრიცხვო ჩანაწერის სახელს, ცხრილის სახელს
- გ. მონაცემთა ბაზის მიმდინარე ზომას, მონაცემთა ბაზის მფლობელის სააღრიცხვო ჩანაწერის სახელს, მონაცემთა ბაზის შექმნის თარიღს და მის მიმდინარე სტატუსს, ინფორმაციას მონაცემთა ბაზის ფაილების შესახებ

2.4.15.

```
EXEC sp_attach_db 'Baza1',  
'C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA\Baza1.mdf',  
'C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA\Baza1_log.ldf';
```

შენახული პროცედურის შესრულების შედეგად:

- ა. Baza1 მონაცემთა ბაზა მიერთდება სერვერს
- ბ. Baza1 მონაცემთა ბაზა არ მიერთდება სერვერს
- გ. Baza1 მონაცემთა ბაზა გამოირთვება სერვერიდან

2.4.16.

```
EXEC sp_attach_db 'Baza2',  
'C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA\Baza2.mdf';
```

შენახული პროცედურის შესრულების შედეგად:

- ა. Baza2 მონაცემთა ბაზა მიერთდება სერვერს
- ბ. Baza2 მონაცემთა ბაზა არ მიერთდება სერვერს
- გ. Baza2 მონაცემთა ბაზა გამოირთვება სერვერიდან

2.4.17. Baza2 მონაცემთა ბაზის მისაერთებლად სერვერთან უნდა შევასრულოთ შენახული

პროცედურა:

- ა. EXEC sp_attach_db 'Baza1',
'C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA\Baza2.mdf';
- ბ. EXEC sp_attach_db 'Baza2',
'C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA\Baza2.mdf';
- გ. EXEC sp_attach_db 'Baza2',
'C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA\Baza1.mdf';

2.4.18. EXEC sp_helpdb 'Shekveta'; შენახული პროცედურის შესრულების შედეგად:

- ა. არ მივიღებთ ინფორმაციას Shekveta მონაცემთა ბაზის შესახებ
- ბ. მივიღებთ ინფორმაციას Shekveta მონაცემთა ბაზის შესახებ
- გ. წაიშლება Shekveta მონაცემთა ბაზა

2.4.19. Shekveta მონაცემთა ბაზის შესახებ ინფორმაციის მისაღებად უნდა შევიტანოთ ბრძანება:

- ა. sp_changedb 'Baza2'
- ბ. EXEC sp_attach_db 'Baza2'
- გ. EXEC sp_helpdb 'Shekveta';

მონაცემთა მომხმარებლის ტიპის შექმნა

2.5.1. მონაცემთა მომხმარებლის ტიპის შესაქმნელად გამოიყენება სისტემური შენახული პროცედურა:

- ა. sp_helpfilegroup
- ბ. sp_spaceused
- გ. sp_addtype

2.5.2. sp_addtype შენახული პროცედურის სინტაქსია:

- ა. sp_addtype [@typename=] ტიპის_სახელი, ,[@nulltype=] 'null_ტიპი'
- ბ. sp_addtype [@typename=] , [@phystype=] სისტემური_ტიპის_სახელი
[,[@nulltype=] 'null_ტიპი']
- გ. sp_addtype [@typename=] ტიპის_სახელი, [@phystype=] სისტემური_ტიპის_სახელი
[,[@nulltype=] 'null_ტიპი']

2.5.3. sp_addtype შენახული პროცედურის 'ტიპის_სახელი' არგუმენტი განსაზღვრავს:

- ა. შესაქმნელი ტიპის იდენტიფიკატორს
- ბ. შესაქმნელი ტიპის სახელს
- გ. შესაქმნელი ტიპის ფაილს

2.5.4. sp_addtype შენახული პროცედურის ტიპის_სახელი არგუმენტი:

- ა. უნდა იყოს უნიკალური მფლობელის ფარგლებში
- ბ. არ უნდა იყოს უნიკალური მფლობელის ფარგლებში
- გ. უნდა იყოს უნიკალური სერვერის ფარგლებში

2.5.5. sp_addtype შენახული პროცედურის სისტემური_ტიპის_სახელი არგუმენტი განსაზღვრავს:

- ა. მონაცემების სისტემურ ტიპს, რომლის საფუძველზეც იქმნება მომხმარებლის ტიპი

- ბ. მონაცემების სისტემურ ტიპს, რომლის საფუძველზეც არ იქმნება მომხმარებლის ტიპი
- გ. მონაცემების არასისტემურ ტიპს, რომლის საფუძველზეც იქმნება მომხმარებლის ტიპი

2.5.6. `sp_addtype` შენახული პროცედურის 'null_ტიპი' არგუმენტი იღებს NULL მნიშვნელობას, მაშინ:

- ა. აკრძალულია NULL მნიშვნელობის შენახვა
- ბ. ნებადართულია NULL მნიშვნელობის შენახვა
- გ. გამოიყენება ავტომატურად განსაზღვრული მნიშვნელობა

2.5.7. `sp_addtype` შენახული პროცედურის 'null_ტიპი' არგუმენტი იღებს NOT NULL მნიშვნელობას, მაშინ:

- ა. ნებადართულია NULL მნიშვნელობის შენახვა
- ბ. გამოიყენება ავტომატურად განსაზღვრული მნიშვნელობა
- გ. აკრძალულია NULL მნიშვნელობის შენახვა

2.5.8. `sp_addtype` შენახული პროცედურის 'null_ტიპი' არგუმენტი იღებს NONULL მნიშვნელობას, მაშინ:

- ა. აკრძალულია NULL მნიშვნელობის შენახვა
- ბ. გამოიყენება ავტომატურად განსაზღვრული მნიშვნელობა
- გ. ნებადართულია NULL მნიშვნელობის შენახვა

2.5.9. `sp_addtype` შენახული პროცედურის 'null_ტიპი' არგუმენტი მითითებული არ არის, მაშინ:

- ა. ნებადართულია NULL მნიშვნელობის შენახვა
- ბ. აკრძალულია NULL მნიშვნელობის შენახვა
- გ. მომხმარებლის ტიპის შექმნისას გამოიყენება ავტომატური მნიშვნელობა

2.5.10. თუ სვეტის შექმნის დროს აშკარადაა მითითებული NULL ან NOT NULL მნიშვნელობები, მაშინ:

- ა. `sp_addtype` შენახული პროცედურის 'null_ტიპი' არგუმენტის მნიშვნელობა იგნორირდება
- ბ.
- გ. `sp_addtype` შენახული პროცედურის 'null_ტიპი' არგუმენტის მნიშვნელობა არ იგნორირდება

2.5.11. მომხმარებლის ტიპის სახელის შესაცვლელად გამოიყენება:

- ა. `sp_rename` შენახული პროცედურა
- ბ. `sp_addtype` შენახული პროცედურა
- გ. `sp_spaceused` შენახული პროცედურა

2.5.12. `sp_rename` შენახული პროცედურის სინტაქსია:

- ა. `sp_rename [@objname =]`
`[@newname=] 'მომხმარებლის_ტიპის_ახალი_სახელი',[@objtype=] 'ობიექტის_ტიპი'`
- ბ. `sp_rename [@objname =]` 'მომხმარებლის_ტიპის_ძველი_სახელი',
`[@newname=] 'მომხმარებლის_ტიპის_ახალი_სახელი',[@objtype=] 'ობიექტის_ტიპი'`
- გ. `sp_rename [@objname =]` 'მომხმარებლის_ტიპის_ძველი_სახელი',
`[@newname=] 'მომხმარებლის_ტიპის_ახალი_სახელი',[@objtype=]`

- 2.5.13. `sp_rename` შენახული პროცედურის 'მომხმარებლის_ტიპის_ძველი_სახელი' არგუმენტი:
- ა. არ არის მომხმარებლის ტიპის მიმდინარე სახელი, რომლის შეცვლაც გვინდა
 - ბ. არის მომხმარებლის ტიპის ახალი სახელი, რომლის შეცვლაც გვინდა
 - გ. არის მომხმარებლის ტიპის მიმდინარე სახელი, რომლის შეცვლაც გვინდა
- 2.5.14. `sp_rename` შენახული პროცედურის 'მომხმარებლის_ტიპის_ახალი_სახელი' არგუმენტი:
- ა. არ არის მომხმარებლის ტიპის ახალი სახელი
 - ბ. არის მომხმარებლის ტიპის მიმდინარე სახელი
 - გ. არის მომხმარებლის ტიპის ახალი სახელი
- 2.5.15. `sp_rename` შენახული პროცედურის 'ობიექტის_ტიპი' არგუმენტს მომხმარებლის ტიპისათვის სახელის შესაცვლელად უნდა მივანიჭოთ:
- ა. 'USERDATATYPE' მნიშვნელობა
 - ბ. 'USERDBTYPE' მნიშვნელობა
 - გ. 'USERDATETYPE' მნიშვნელობა
- 2.5.16. EXEC `sp_addtype mobiluri, 'VARCHAR(10)', 'NOT NULL'`; შენახული პროცედურის შესრულების შედეგად:
- ა. წაიშლება mobiluri მომხმარებლის ტიპი
 - ბ. VARCHAR ტიპის საფუძველზე არ შეიქმნება mobiluri მომხმარებლის ტიპი
 - გ. VARCHAR ტიპის საფუძველზე შეიქმნება mobiluri მომხმარებლის ტიპი
- 2.5.17. DATETIME ტიპის საფუძველზე dabadebis_dge მომხმარებლის ტიპის შესაქმნელად უნდა შევასრულოთ შენახული პროცედურა:
- ა. `sp_spaceused` შენახული პროცედურა
 - ბ. `sp_rename` შენახული პროცედურა
 - გ. EXEC `sp_addtype dabadebis_dge, DATETIME, 'NULL'`;
- 2.5.18. EXEC `sp_rename N'mobiluri', N'Mobiluri_Telefoni', N'USERDATATYPE'`; შენახული პროცედურის შესრულების შედეგად:
- ა. 'mobiluri' მომხმარებლის ტიპს დაერქმევა ახალი 'Mobiluri_Telefoni' სახელი
 - ბ. 'mobiluri' მომხმარებლის ტიპს არ დაერქმევა ახალი 'Mobiluri_Telefoni' სახელი
 - გ. 'Mobiluri_Telefoni' მომხმარებლის ტიპს დაერქმევა ახალი 'mobiluri' სახელი
- 2.5.19. 'mobiluri' მომხმარებლის ტიპისთვის ახალი 'Mobiluri_Telefoni' სახელის დასარქმევად უნდა შევასრულოთ შენახული პროცედურა:
- ა. EXEC `sp_rename N'mobiluri', N'Mobiluri_Telefoni', N'USERDATATYPE'`;
 - ბ. EXEC `sp_rename N'Mobiluri_Telefoni', N'mobiluri', N'USERDATATYPE'`;
 - გ. `sp_addtype`

თავი 3. ცხრილები

ცხრილების დაპროექტება

- 3.1.1. პირველადი გასაღების დანიშნულებაა:
- ა. ცხრილის მთლიანობის უზრუნველყოფა
 - ბ. მონაცემთა ბაზის მთლიანობის უზრუნველყოფა
 - გ. სვეტის მთლიანობის უზრუნველყოფა
- 3.1.2. პირველადი გასაღები შეიძლება შედგებოდეს:
- ა. მხოლოდ ერთი სვეტისგან
 - ბ. ერთი ან მეტი სვეტისაგან
 - გ. არც ერთი სვეტისაგან
- 3.1.3. თუ პირველადი გასაღები ერთი სვეტისაგან შედგება, მაშინ ამ სვეტის მნიშვნელობები:
- ა. უნიკალური უნდა იყოს
 - ბ. უნიკალური არ უნდა იყოს
 - გ. უნიკალური შეიძლება იყოს
- 3.1.4. თუ პირველადი გასაღები რამდენიმე სვეტისაგან შედგება, მაშინ თითოეული სვეტის შიგნით მნიშვნელობები შეიძლება:
- ა. არ გამეორდეს
 - ბ. გამეორდეს
 - გ. არ იყოს დუბლირებული
- 3.1.5. პირველად გასაღებში შემავალი ყველა სვეტის მნიშვნელობების ნებისმიერი კომბინაცია:
- ა. არ უნდა იყოს უნიკალური
 - ბ. უნდა იყოს დუბლირებული
 - გ. უნდა იყოს უნიკალური
- 3.1.6. პირველადი გასაღების შექმნის დროს მასში შემავალი სვეტებისათვის სერვერი ავტომატურად:
- ა. არ ქმნის უნიკალურ ინდექსს
 - ბ. ქმნის უნიკალურ ინდექსს
 - გ. ქმნის უნიკალურ ფაილს
- 3.1.7. მოთხოვნებში პირველადი გასაღების გამოყენების დროს უნიკალური ინდექსი პირველად გასაღებში შემავალი სვეტების მონაცემებთან მიმართებას:
- ა. ამცირებს
 - ბ. არ აჩქარებს
 - გ. აჩქარებს
- 3.1.8. ცხრილს შეიძლება ჰქონდეს:
- ა. მხოლოდ ერთი PRIMARY KEY შეზღუდვა
 - ბ. რამდენიმე PRIMARY KEY შეზღუდვა
 - გ. ორი PRIMARY KEY შეზღუდვა

3.1.9. პირველად გასაღებში შემავალი:

- ა. არც ერთი სვეტი არ უნდა შეიცავდეს NULL მნიშვნელობას
- ბ. ყველა სვეტი უნდა შეიცავდეს NULL მნიშვნელობას
- გ. ზოგიერთი სვეტი არ უნდა შეიცავდეს NULL მნიშვნელობას

3.1.10. პირველადი გასაღების ერთ-ერთი დანიშნულებაა:

- ა. ცხრილის მონაცემების გადაწერა
- ბ. რამდენიმე ცხრილის მონაცემების მიმართებითი მთლიანობის უზრუნველყოფა
- გ. ცხრილის მონაცემების წაკითხვა

3.1.11. გარე გასაღები შეიძლება შეიცავდეს:

- ა. ყველა სვეტს
- ბ. მხოლოდ ორ სვეტს
- გ. ერთ ან მეტ სვეტს

3.1.12. გარე გასაღები გამოიყენება:

- ა. სამ ცხრილს შორის კავშირის დასამყარებლად
- ბ. ორ ცხრილს შორის კავშირის დასამყარებლად
- გ. ერთ ცხრილს შორის კავშირის დასამყარებლად

3.1.13. დამოკიდებული ცხრილის გარე გასაღებში შემავალ სვეტებში მონაცემებმა მხოლოდ ის მნიშვნელობები უნდა მიიღონ, რომლებიც აქვთ:

- ა. მათთან დაკავშირებული მთავარი ცხრილის გარე გასაღების სვეტებს
- ბ. მათთან დაკავშირებული მთავარი ცხრილის პირველადი გასაღების სვეტებს
- გ. მათთან დაკავშირებული მთავარი ცხრილის პირველადი გასაღების სტრიქონებს

3.1.14. FOREIGN KEY შეზღუდვის მთავარი მიზანია:

- ა. დამოკიდებული ცხრილის გარე გასაღებში მოთავსებული მონაცემების მართვა
- ბ. დამოკიდებული ცხრილის პირველად გასაღებში მოთავსებული მონაცემების მართვა
- გ. მთავარი ცხრილის გარე გასაღებში მოთავსებული მონაცემების მართვა

3.1.15. FOREIGN KEY შეზღუდვა საშუალებას გვაძლევს ვმართოთ:

- ა. მთავარი ცხრილის გარე გასაღებში მონაცემების ცვლილება
- ბ. დამოკიდებული ცხრილის პირველად გასაღებში მონაცემების ცვლილება
- გ. მთავარი ცხრილის პირველად გასაღებში მონაცემების ცვლილება

3.1.16. თუ გარე გასაღების სვეტი NULL მნიშვნელობას შეიცავს, მაშინ:

- ა. შემოწმება FOREIGN KEY შეზღუდვაზე შესრულდება
- ბ. შემოწმება FOREIGN KEY შეზღუდვაზე არ შესრულდება
- გ. შემოწმება PRIMARY KEY შეზღუდვაზე არ შესრულდება

3.1.17. გარე გასაღების ინდექსირება:

- ა. ანელებს საჭირო მონაცემების მოძებნას
- ბ. არ აჩქარებს საჭირო მონაცემების მოძებნას
- გ. აჩქარებს საჭირო მონაცემების მოძებნას

3.1.18. UNIQUE შეზღუდვა განკუთვნილია:

- ა. სვეტში მნიშვნელობების უნიკალურობის უზრუნველსაყოფად
- ბ. სტრიქონში მნიშვნელობების უნიკალურობის უზრუნველსაყოფად
- გ. მონაცემთა ბაზაში მნიშვნელობების უნიკალურობის უზრუნველსაყოფად

3.1.19. PRIMARY KEY შეზღუდვისათვის:

- ა. ავტომატურად არ ხდება მნიშვნელობების უნიკალურობის უზრუნველყოფა
- ბ. ავტომატურად ხდება მნიშვნელობების უნიკალურობის უზრუნველყოფა
- გ. ავტომატურად ხდება მნიშვნელობების უნიკალურობის შეცვლა

3.1.20. თუ პირველადი გასაღები განსაზღვრულია და კიდევ ერთ ან მეტ სვეტშია საჭირო უნიკალურობის უზრუნველყოფა, მაშინ:

- ა. უნდა გამოვიყენოთ FOREIGN KEY შეზღუდვა
- ბ. არ უნდა გამოვიყენოთ UNIQUE შეზღუდვა
- გ. უნდა გამოვიყენოთ UNIQUE შეზღუდვა

3.1.21. FOREIGN KEY შეზღუდვისაგან განსხვავებით UNIQUE შეზღუდვა:

- ა. უშვებს სვეტში NULL მნიშვნელობის არსებობას
- ბ. არ უშვებს სვეტში NULL მნიშვნელობის არსებობას
- გ. უშვებს მონაცემთა ბაზაში NULL მნიშვნელობის არსებობას

3.1.22. UNIQUE შეზღუდვის გამოყენებისას შესაბამის სვეტში შეიძლება იყოს:

- ა. სამი NULL მნიშვნელობა
- ბ. ორი NULL მნიშვნელობა
- გ. მხოლოდ ერთი NULL მნიშვნელობა

3.1.23. CHECK შეზღუდვა განსაზღვრავს შესაძლო მნიშვნელობების დიაპაზონს:

- ა. მხოლოდ ერთი სვეტისთვის
- ბ. ერთი ან მეტი სვეტისთვის
- გ. ერთი სტრიქონისთვის

3.1.24. ერთი სვეტისთვის შეიძლება განისაზღვროს:

- ა. რამდენიმე CHECK შეზღუდვა
- ბ. მხოლოდ ერთი CHECK შეზღუდვა
- გ. არც ერთი CHECK შეზღუდვა

3.1.25. თუ რამდენიმე სვეტის მიმართ უნდა გამოვიყენოთ ერთი და იგივე CHECK შეზღუდვა, მაშინ ეს შეზღუდვა:

- ა. ცალ-ცალკე უნდა განისაზღვროს თითოეული მონაცემთა ბაზისთვის
- ბ. ცალ-ცალკე უნდა განისაზღვროს თითოეული სტრიქონისთვის
- გ. ცალ-ცალკე უნდა განისაზღვროს თითოეული სვეტისთვის

3.1.26. CHECK შეზღუდვას საფუძვლად უდევს:

- ა. არითმეტიკული გამოსახულება
- ბ. ლოგიკური გამოსახულება
- გ. ფიზიკური გამოსახულება

- 3.1.27. თუ CHECK შეზღუდვაში ლოგიკური გამოსახულება იღებს true მნიშვნელობას, მაშინ:
- ა. მთლიანობის შეზღუდვა არ სრულდება და მონაცემების შეცვლისა და ჩასმის ოპერაციები ნებადართული იქნება
 - ბ. მთლიანობის შეზღუდვა სრულდება და მონაცემების შეცვლისა და ჩასმის ოპერაციები ნებადართული იქნება
 - გ. მთლიანობის შეზღუდვა სრულდება და მონაცემების შეცვლისა და ჩასმის ოპერაციები ნებადართული არ იქნება
- 3.1.28. თუ მითითებულია რამდენიმე CHECK შეზღუდვა, მაშინ ისინი:
- ა. გამოყენებული იქნება იმ მიმდევრობით, რა მიმდევრობითაც მოხდა მათი შექმნა
 - ბ. გამოყენებული არ იქნება იმ მიმდევრობით, რა მიმდევრობითაც მოხდა მათი შექმნა
 - გ. გამოყენებული იქნება იმ მიმდევრობით, რა მიმდევრობითაც არ მოხდა მათი შექმნა
- 3.1.29. CHECK შეზღუდვებში:
- ა. დასაშვებია სხვა სტრიქონების სახელების მითითებაც
 - ბ. დაუშვებელია სხვა სვეტების სახელების მითითებაც
 - გ. დასაშვებია სხვა სვეტების სახელების მითითებაც
- 3.1.30. თითოეული სტრიქონის სვეტი ყოველთვის შეიცავს:
- ა. კონკრეტულ ან NULL მნიშვნელობას
 - ბ. მხოლოდ NULL მნიშვნელობას
 - გ. მხოლოდ კონკრეტულ მნიშვნელობას
- 3.1.31. თუ სვეტისათვის განსაზღვრულია ნაგულისხმევი მნიშვნელობა, მაშინ:
- ა. ის იქნება სვეტის მნიშვნელობა, თუ აშკარად მივუთითებთ ამ სვეტის მნიშვნელობას
 - ბ. ის არ იქნება სვეტის მნიშვნელობა, თუ აშკარად არ მივუთითებთ ამ სვეტის მნიშვნელობას
 - გ. ის იქნება სვეტის მნიშვნელობა, თუ აშკარად არ მივუთითებთ ამ სვეტის მნიშვნელობას
- 3.1.32. ნაგულისხმევი მნიშვნელობა შეგვიძლია გამოვიყენოთ მაშინ, როცა სვეტს:
- ა. ხშირად ენიჭება ერთი და იგივე მნიშვნელობა
 - ბ. იშვიათად ენიჭება ერთი და იგივე მნიშვნელობა
 - გ. არასოდეს არ ენიჭება ერთი და იგივე მნიშვნელობა
- 3.1.33. ნაგულისხმევი მნიშვნელობის განსაზღვრისას:
- ა. არ შეგვიძლია ჩადგმული ფუნქციების გამოყენებაც
 - ბ. შეგვიძლია ჩადგმული ფუნქციების გამოყენებაც
 - გ. შეუძლებელია ჩადგმული ფუნქციების გამოყენებაც
- 3.1.34. სვეტი-მთვლელისთვის სრულდება უნიკალური მნიშვნელობების:
- ა. ხელით გენერირება
 - ბ. ავტომატურად გენერირება
 - გ. არანაირი გენერირება
- 3.1.35. IDENTITY საკვანძო სიტყვის გამოყენება:
- ა. უზრუნველყოფს მონაცემების უნიკალურობას ორი სვეტის ფარგლებში

- ბ. არ უზრუნველყოფს მონაცემების უნიკალურობას ერთი სვეტის ფარგლებში
 - გ. უზრუნველყოფს მონაცემების უნიკალურობას ერთი სვეტის ფარგლებში
- 3.1.36. სვეტი-მთვლელის განსაზღვრისას უნდა მივუთითოთ:
- ა. მისი საწყისი მნიშვნელობა და ბიჯი
 - ბ. მისი ბიჯი
 - გ. მისი საწყისი მნიშვნელობა
- 3.1.37. თუ სვეტი-მთვლელის საწყისი მნიშვნელობა და ბიჯი მითითებული არ არის, მაშინ:
- ა. ორივეს მნიშვნელობა 1-ის ტოლი იქნება
 - ბ. ორივეს მნიშვნელობა 10-ის ტოლი იქნება
 - გ. ორივეს მნიშვნელობა 0-ის ტოლი იქნება
- 3.1.38. ცხრილს შეიძლება ჰქონდეს:
- ა. ორი სვეტი-მთვლელი
 - ბ. მხოლოდ ერთი სვეტი-მთვლელი
 - გ. სამი სვეტი-მთვლელი
- 3.1.39. თუ სვეტისთვის განსაზღვრულია IDENTITY თვისება, მაშინ ამ სვეტის ტიპი უნდა იყოს:
- ა. DATETIME, DECIMAL
 - ბ. FLOAT, NUMERIC
 - გ. BIGINT, DECIMAL, INT, NUMERIC, SMALLINT, TINYINT
- 3.1.40. თუ სვეტისთვის განსაზღვრულია IDENTITY თვისება, მაშინ ამ სვეტისთვის:
- ა. აკრძალული უნდა იყოს NULL მნიშვნელობის შენახვა
 - ბ. ნებადართული უნდა იყოს NULL მნიშვნელობის შენახვა
 - გ. აკრძალული არ უნდა იყოს NULL მნიშვნელობის შენახვა
- 3.1.41. თუ სვეტისთვის განსაზღვრულია IDENTITY თვისება, მაშინ ამ სვეტისთვის:
- ა. სვეტისთვის უნდა იყოს განსაზღვრული ავტომატური (ნაგულისხმევი) მნიშვნელობა
 - ბ. სვეტისთვის არ უნდა იყოს განსაზღვრული ავტომატური (ნაგულისხმევი) მნიშვნელობა
 - გ. შეიძლება განსაზღვრული იყოს ავტომატური (ნაგულისხმევი) მნიშვნელობა
- 3.1.42. სვეტი-მთვლელი:
- ა. აუცილებლად უნდა შევიდეს ცხრილის პირველადი გასაღების შემადგენლობაში
 - ბ. არ უნდა შევიდეს ცხრილის პირველადი გასაღების შემადგენლობაში
 - გ. შეიძლება შევიდეს ცხრილის პირველადი გასაღების შემადგენლობაში
- 3.1.43. მიმდინარე IDENTITY მნიშვნელობა ინახება:
- ა. master მონაცემთა ბაზის sys.identity_columns სისტემურ წარმოდგენაში
 - ბ. ამავე მონაცემთა ბაზის sys.identity_columns სისტემურ წარმოდგენაში
 - გ. model მონაცემთა ბაზის sys.identity_columns სისტემურ წარმოდგენაში
- 3.1.44. თუ ცხრილიდან ყველა სტრიქონს წავშლით:
- ა. ეს არ გამოიწვევს სვეტი-მთვლელში ნუმერაციის თავიდან დაწყებას
 - ბ. ეს გამოიწვევს სვეტი-მთვლელში ნუმერაციის თავიდან დაწყებას

გ. ეს გამოიწვევს სვეტი-მთვლელობის ნუმერაციის ნულიდან დაწყებას

3.1.45. თუ სტრიქონის ჩასმა არ მოხერხდა, მაშინ სერვერი:

- ა. მაინც გაზრდის სვეტი-მთვლელობის მნიშვნელობას
- ბ. არ გაზრდის სვეტი-მთვლელობის მნიშვნელობას
- გ. შეამცირებს სვეტი-მთვლელობის მნიშვნელობას

3.1.46. თუ ცხრილის უკანასკნელ სტრიქონს წავშლით, რომელშიც სვეტი-მთვლელობის მნიშვნელობა იყო 20 და შემდეგ ერთ სტრიქონს დავუმატებთ, მაშინ დამატებულ სტრიქონში სვეტი-მთვლელობის მნიშვნელობა იქნება:

- ა. 19
- ბ. 20
- გ. 21

ცხრილის შექმნა. ცხრილის წაშლა

3.2.1. ცხრილის შესაქმნელად გამოიყენება ბრძანება:

- ა. CREATE TABLE
- ბ. DBCC CHECKIDENT
- გ. SET IDENTITY_INSERT

3.2.2. თუ CREATE TABLE ბრძანებაში მითითებულია არ არის მონაცემთა ბაზის სახელი, მაშინ:

- ა. ცხრილი model მონაცემთა ბაზაში შეიქმნება
- ბ. ცხრილი მიმდინარე მონაცემთა ბაზაში არ შეიქმნება
- გ. ცხრილი მიმდინარე მონაცემთა ბაზაში შეიქმნება

3.2.3. ცხრილის სახელისა და სქემის სახელის კომბინაცია:

- ა. უნიკალური არ უნდა იყოს მონაცემთა ბაზის ფარგლებში
- ბ. უნიკალური უნდა იყოს მონაცემთა ბაზის ფარგლებში
- გ. უნიკალური უნდა იყოს სერვერის ფარგლებში

3.2.4. თუ ცხრილი არ იქმნება მიმდინარე მონაცემთა ბაზაში, მაშინ მონაცემთა ბაზის სახელი:

- ა. აშკარად უნდა მივუთითოთ
- ბ. აშკარად არ უნდა მივუთითოთ
- გ. საერთოდ არ უნდა მივუთითოთ

3.2.5. თუ ცხრილის სახელის წინ მივუთითებთ # სიმბოლოს, მაშინ შეიქმნება:

- ა. არალოკალური დროებითი ცხრილი
- ბ. ლოკალური დროებითი ცხრილი
- გ. გლობალური დროებითი ცხრილი

3.2.6. თუ ცხრილის სახელის წინ მივუთითებთ ## სიმბოლოებს, მაშინ შეიქმნება:

- ა. გლობალური დროებითი ცხრილი
- ბ. არაგლობალური დროებითი ცხრილი
- გ. ლოკალური დროებითი ცხრილი

3.2.7. CREATE TABLE ბრძანებაში 'გამოსახულება' არგუმენტის გამოთვლის შედეგად:

- ა. მიიღება მნიშვნელობა ჩვეულებრივი სვეტისათვის
- ბ. არ მიიღება მნიშვნელობა გამოთვლადი სვეტისათვის
- გ. მიიღება მნიშვნელობა გამოთვლადი სვეტისათვის

3.2.8. გამოთვლადი სვეტები:

- ა. ვირტუალური სტრიქონებია
- ბ. არაა ვირტუალური სვეტები
- გ. ვირტუალური სვეტებია

3.2.9. გამოთვლადი სვეტები:

- ა. ფიზიკურად არ ინახება ცხრილში
- ბ. ფიზიკურად ინახება ცხრილში
- გ. ფიზიკურად ინახება ცხრილში

3.2.10. გამოთვლადი სვეტის მნიშვნელობა:

- ა. არ გამოითვლება ცხრილის გახსნის დროს ამავე ცხრილის სხვა სვეტების მნიშვნელობების გამოყენებით
- ბ. გამოითვლება ცხრილის გახსნის დროს ამავე ცხრილის სხვა სვეტების მნიშვნელობების გამოყენებით
- გ. გამოითვლება ცხრილის დახურვის დროს ამავე ცხრილის სხვა სვეტების მნიშვნელობების გამოყენებით

3.2.11. CREATE TABLE ბრძანებაში 'გამოსახულება' არგუმენტი:

- ა. შეიძლება შეიცავდეს სტრიქონების სახელებს
- ბ. არ შეიძლება შეიცავდეს სვეტების სახელებს, მუდმივებს, ფუნქციებს
- გ. შეიძლება შეიცავდეს სვეტების სახელებს, მუდმივებს, ფუნქციებს

3.2.12. CREATE TABLE ბრძანების 'გამოსახულება' არგუმენტში ქვემოთხოვანების გამოყენება:

- ა. დასაშვებია
- ბ. დაუშვებელია
- გ. ხანდახან დასაშვებია

3.2.13. გამოთვლადი სვეტები:

- ა. შეიძლება ჩართული იყოს SELECT მოთხოვნაში სვეტების სიის მითითების დროს
- ბ. არ შეიძლება ჩართული იყოს SELECT მოთხოვნაში სვეტების სიის მითითების დროს
- გ. შეიძლება ჩართული იყოს SELECT მოთხოვნაში სტრიქონების სიის მითითების დროს

3.2.14. გამოთვლადი სვეტები:

- ა. შეიძლება გამოვიყენოთ ინდექსებში
- ბ. არ შეიძლება გამოვიყენოთ ინდექსებში
- გ. აუცილებლად უნდა გამოვიყენოთ ინდექსებში

3.2.15. გამოთვლადი სვეტები:

- ა. არ შეიძლება გამოვიყენოთ როგორც პირველადი გასაღების ნაწილი
- ბ. შეიძლება გამოვიყენოთ როგორც პირველადი გასაღების ნაწილი

გ. შეიძლება გამოვიყენოთ როგორც გარე გასაღების ნაწილი

3.2.16. თუ გამოთვლადი სვეტი გამოიყენება ინდექსში ან როგორც პირველადი გასაღების ნაწილი, მაშინ გამოთვლადი სვეტის მნიშვნელობები:

- ა. უნდა განისაზღვროდეს არადეტერმინირებული გამოსახულებით
- ბ. არ უნდა განისაზღვროდეს დეტერმინირებული გამოსახულებით
- გ. უნდა განისაზღვროდეს დეტერმინირებული გამოსახულებით

3.2.17. დავუშვათ ცხრილში გვაქვს ორი მთელი ცხვით სვეტი A და B, ხოლო გამოთვლადი სვეტის მნიშვნელობა გამოითვლება $A + B$ გამოსახულებით. ასეთ შემთხვევაში, ამ სვეტის ინდექსირება:

- ა. ხანდახან არ შეიძლება
- ბ. არ შეიძლება
- გ. შეიძლება

3.2.18. თუ გამოთვლადი სვეტი განისაზღვრება $A + DATEPART(dd, GETDATE())$ გამოსახულებით, მაშინ მისი ინდექსირება:

- ა. არ შეიძლება
- ბ. შეიძლება
- გ. ხანდახან შეიძლება

3.2.19. გამოთვლადი სვეტისათვის:

- ა. არ შეიძლება UNIQUE მნიშვნელობის განსაზღვრა
- ბ. შეიძლება UNIQUE მნიშვნელობის განსაზღვრა
- გ. ხანდახან შეიძლება UNIQUE მნიშვნელობის განსაზღვრა

3.2.20. გამოთვლადი სვეტები:

- ა. ხანდახან უნდა შევიდეს გარე გასაღების შემადგენლობაში
- ბ. უნდა შევიდეს გარე გასაღების შემადგენლობაში
- გ. არ უნდა შევიდეს გარე გასაღების შემადგენლობაში

3.2.21. გამოთვლადი სვეტებისთვის:

- ა. შეიძლება ავტომატური მნიშვნელობების განსაზღვრა
- ბ. არ შეიძლება ავტომატური მნიშვნელობების განსაზღვრა
- გ. ხანდახან შეიძლება ავტომატური მნიშვნელობების განსაზღვრა

3.2.22. გამოთვლადი სვეტების გამოყენება:

- ა. არ შეიძლება INSERT და UPDATE ოპერაციებში
- ბ. შეიძლება INSERT და UPDATE ოპერაციებში
- გ. ხანდახან შეიძლება INSERT და UPDATE ოპერაციებში

3.2.23. სვეტის განსაზღვრისას DEFAULT არგუმენტი:

- ა. მიუთითებს ნაგულისხმევ მნიშვნელობას მოცემული სტრიქონისთვის
- ბ. არ მიუთითებს ნაგულისხმევ მნიშვნელობას მოცემული სვეტისათვის
- გ. მიუთითებს ნაგულისხმევ (ავტომატურ) მნიშვნელობას მოცემული სვეტისათვის

3.2.24. DEFAULT მნიშვნელობა მიენიჭება სვეტს თუ სტრიქონის ჩასმისას:

- ა. აშკარად იქნება მითითებული სვეტის მნიშვნელობა
- ბ. აშკარად არ იქნება მითითებული სვეტის მნიშვნელობა
- გ. აშკარად არ იქნება მითითებული სტრიქონის მნიშვნელობა

3.2.25. ნაგულისხმევი (DEFAULT) მნიშვნელობის განსაზღვრა:

- ა. არ შეიძლება timestamp ტიპის სვეტებისათვის, აგრეთვე, IDENTITY თვისების მქონე სვეტებისათვის
- ბ. შეიძლება timestamp ტიპის სვეტებისათვის, აგრეთვე, IDENTITY თვისების მქონე სვეტებისათვის
- გ. ხანდახან შეიძლება timestamp ტიპის სვეტებისათვის, აგრეთვე, IDENTITY თვისების მქონე სვეტებისათვის

3.2.26. ნაგულისხმევი (DEFAULT) მნიშვნელობა:

- ა. აუცილებლად უნდა იყოს მუდმივა
- ბ. არ შეიძლება იყოს მუდმივა, სისტემური ფუნქციების მნიშვნელობები
- გ. შეიძლება იყოს მუდმივა, სისტემური ფუნქციების მნიშვნელობები

3.2.27. თუ სვეტისათვის არ არის განსაზღვრული DEFAULT მნიშვნელობა და სვეტისათვის არ არის მითითებული მნიშვნელობა სტრიქონის დამატებისას, მაშინ სვეტს:

- ა. NULL მნიშვნელობა მიენიჭება, თუ ეს ნებადართულია
- ბ. NULL მნიშვნელობა მიენიჭება, თუ ეს ნებადართული არაა
- გ. ნული მიენიჭება

3.2.28. სვეტის განსაზღვრისას IDENTITY არგუმენტი მიუთითებს, რომ შესაბამისი სვეტი:

- ა. იქნება სტრიქონი-მთვლელი
- ბ. არ იქნება სვეტი-მთვლელი
- გ. იქნება სვეტი-მთვლელი

3.2.29. თუ სვეტის განსაზღვრისას გამოყენებულია IDENTITY არგუმენტი, მაშინ 'საწყისი_მნიშვნელობა' არის:

- ა. მთვლელის ნაზრდი
- ბ. მთვლელის საწყისი მნიშვნელობა
- გ. მთვლელის საბოლოო მნიშვნელობა

3.2.30. თუ სვეტის განსაზღვრისას გამოყენებულია IDENTITY არგუმენტი, მაშინ 'ნაზრდი' არის:

- ა. მთვლელის საბოლოო მნიშვნელობა
- ბ. მთვლელის ნაზრდი
- გ. მთვლელის საწყისი მნიშვნელობა

3.2.31. სვეტის შეზღუდვის მითითებისას CONSTRAINT პარამეტრის შემდეგ ეთითება:

- ა. სვეტის მნიშვნელობებზე შეზღუდვის სახელი, რომელიც მონაცემთა ბაზის ფარგლებში უნიკალური უნდა იყოს
- ბ. სვეტის მნიშვნელობებზე შეზღუდვის სახელი, რომელიც მონაცემთა ბაზის ფარგლებში უნიკალური არ უნდა იყოს
- გ. სვეტის მნიშვნელობებზე შეზღუდვის სახელი, რომელიც სერვერის ფარგლებში უნიკალური უნდა იყოს

- 3.2.32. სვეტის შეზღუდვის მითითებისას NULL პარამეტრი:
- ა. გვაძლევს სტრიქონში NULL მნიშვნელობის შენახვის უფლებას
 - ბ. არ გვაძლევს სვეტში NULL მნიშვნელობის შენახვის უფლებას
 - გ. გვაძლევს სვეტში NULL მნიშვნელობის შენახვის უფლებას
- 3.2.33. სვეტის შეზღუდვის მითითებისას NOT NULL პარამეტრი:
- ა. არ კრძალავს სვეტში NULL მნიშვნელობის შენახვას
 - ბ. კრძალავს სვეტში NULL მნიშვნელობის შენახვას
 - გ. კრძალავს სვეტში ნულის შენახვას
- 3.2.34. სვეტის შეზღუდვის მითითებისას PRIMARY KEY პარამეტრი მიუთითებს, რომ სვეტის ბაზაზე:
- ა. უნდა შეიქმნას გარე გასაღები
 - ბ. არ უნდა შეიქმნას პირველადი გასაღები
 - გ. უნდა შეიქმნას პირველადი გასაღები
- 3.2.35. თითოეული ცხრილისათვის შეიძლება შეიქმნას მხოლოდ:
- ა. ორი PRIMARY KEY შეზღუდვა
 - ბ. ერთი PRIMARY KEY შეზღუდვა
 - გ. რამდენიმე PRIMARY KEY შეზღუდვა
- 3.2.36. თუ სვეტის შეზღუდვაში მითითებულია PRIMARY KEY პარამეტრი, მაშინ პირველადი გასაღები ასეთი გზით განისაზღვრება:
- ა. ცხრილის დონეზე
 - ბ. სვეტის დონეზე
 - გ. მონაცემთა ბაზის დონეზე
- 3.2.37. თუ სვეტის შეზღუდვაში მითითებულია PRIMARY KEY პარამეტრი, მაშინ პირველადი გასაღები შემდგარი იქნება მხოლოდ:
- ა. ერთი სვეტისაგან
 - ბ. ორი სვეტისაგან
 - გ. ერთმდენიმე სვეტისაგან
- 3.2.38. თუ საჭიროა პირველადი გასაღების ფორმირება ორი და მეტი სვეტის ბაზაზე, მაშინ საჭიროა მისი კონფიგურირება:
- ა. სვეტის დონეზე
 - ბ. ცხრილის დონეზე
 - გ. მონაცემთა ბაზის დონეზე
- 3.2.39. სვეტის შეზღუდვაში UNIQUE პარამეტრი მიუთითებს, რომ სვეტში მოთავსებული მონაცემები:
- ა. უნიკალური არ იქნება, ანუ სვეტში არ მოთავსდება გამეორებადი მნიშვნელობები
 - ბ. უნიკალური იქნება, ანუ სვეტში არ მოთავსდება გამეორებადი მნიშვნელობები
 - გ. უნიკალური იქნება, ანუ სვეტში მოთავსდება გამეორებადი მნიშვნელობები

3.2.40. UNIQUE მთლიანობის შეზღუდვისათვის ავტომატურად:

- ა. ხანდახან იქმნება ინდექსი
- ბ. არ იქმნება ინდექსი
- გ. იქმნება ინდექსი

3.2.41. ერთ ცხრილში სხვადასხვა სვეტისთვის:

- ა. შეგვიძლია შევქმნათ მხოლოდ ორი UNIQUE შეზღუდვა
- ბ. არ შეგვიძლია შევქმნათ რამდენიმე UNIQUE შეზღუდვა
- გ. შეგვიძლია შევქმნათ რამდენიმე UNIQUE შეზღუდვა

3.2.42. CLUSTERED პარამეტრი გამოიყენება სვეტისათვის:

- ა. კლასტერული ინდექსის შესაქმნელად
- ბ. არაკლასტერული ინდექსის შესაქმნელად
- გ. კლასტერული ტრიგერის შესაქმნელად

3.2.43. NOT CLUSTERED პარამეტრი გამოიყენება სვეტისათვის:

- ა. არაკლასტერული ტრიგერის შესაქმნელად
- ბ. არაკლასტერული ინდექსის შესაქმნელად
- გ. კლასტერული ინდექსის შესაქმნელად

3.2.44. CLUSTERED და NOT CLUSTERED პარამეტრები:

- ა. აუცილებლად უნდა გამოვიყენოთ PRIMARY KEY ან UNIQUE საკვანძო სიტყვებთან ერთად
- ბ. არ შეგვიძლია გამოვიყენოთ PRIMARY KEY ან UNIQUE საკვანძო სიტყვებთან ერთად
- გ. შეგვიძლია გამოვიყენოთ PRIMARY KEY ან UNIQUE საკვანძო სიტყვებთან ერთად

3.2.45. PRIMARY KEY შეზღუდვისათვის ავტომატურად:

- ა. იქმნება არაკლასტერული ინდექსი
- ბ. იქმნება კლასტერული ინდექსი
- გ. არ იქმნება კლასტერული ინდექსი

3.2.46. UNIQUE შეზღუდვისათვის ავტომატურად:

- ა. იქმნება არაკლასტერული ინდექსი
- ბ. იქმნება კლასტერული ინდექსი
- გ. არ იქმნება არაკლასტერული ინდექსი

3.2.47. [FOREIGN KEY] REFERENCES მთავარი_ცხრილი [(სვეტის_სახელი [,...n])] პარამეტრი მიუთითებს, რომ 'მთავარი_ცხრილი' პარამეტრით განსაზღვრული ცხრილისათვის სვეტი:

- ა. იქნება პირველადი გასაღები
- ბ. არ იქნება გარე გასაღები
- გ. იქნება გარე გასაღები

3.2.48. გარე გასაღებში შემავალი სვეტები:

- ა. არ შეიძლება მიმართავდეს მხოლოდ 'მთავარი_ცხრილი' პარამეტრით განსაზღვრული ცხრილის PRIMARY KEY ან UNIQUE შეზღუდვის მქონე სვეტებს
- ბ. შეიძლება მიმართავდეს მხოლოდ 'მთავარი_ცხრილი' პარამეტრით განსაზღვრული

ცხრილის PRIMARY KEY ან UNIQUE შეზღუდვის მქონე სვეტებს

გ. შეიძლება მიმართავდეს მხოლოდ 'მთავარი_ცხრილი' პარამეტრით განსაზღვრული ცხრილის FOREIGN KEY შეზღუდვის მქონე სვეტებს

3.2.49. თუ მივუთითებთ ON DELETE CASCADE პარამეტრს, მაშინ მთავარი ცხრილიდან სტრიქონის წაშლის შემთხვევაში დამოკიდებულ ცხრილში:

- ა. შესრულდება სტრიქონების ჩამატება
- ბ. არ წაიშლება შესაბამისი სტრიქონი ან სტრიქონები
- გ. წაიშლება შესაბამისი სტრიქონი ან სტრიქონები

3.2.50. თუ მივუთითებთ ON DELETE NO ACTION პარამეტრს, მაშინ:

- ა. სერვერი გასცემს შეტყობინებას შეცდომის შესახებ და სტრიქონები არ წაიშლება მთავარი ცხრილიდან
- ბ. სერვერი არ გასცემს შეტყობინებას შეცდომის შესახებ და სტრიქონები არ წაიშლება მთავარი ცხრილიდან
- გ. სერვერი გასცემს შეტყობინებას შეცდომის შესახებ და სტრიქონები წაიშლება მთავარი ცხრილიდან

3.2.51. ON DELETE პარამეტრში NO ACTION რეჟიმი:

- ა. ნაწილობრივ იგულისხმება
- ბ. ავტომატურად იგულისხმება
- გ. ავტომატურად არ იგულისხმება

3.2.52. თუ მივუთითებთ ON UPDATE CASCADE პარამეტრს, მაშინ მთავარი ცხრილის პირველადი გასაღების მნიშვნელობის შეცვლისას დამოკიდებული ცხრილის შესაბამის სტრიქონებში:

- ა. შეიცვლება პირველადი გასაღების მნიშვნელობაც
- ბ. არ შეიცვლება გარე გასაღების მნიშვნელობაც
- გ. შეიცვლება გარე გასაღების მნიშვნელობაც

3.2.53. თუ მივუთითებთ ON UPDATE NO ACTION პარამეტრს, მაშინ სერვერი:

- ა. გასცემს შეტყობინებას შეცდომის შესახებ და არ შეასრულებს ცვლილებებს მთავარ ცხრილში
- ბ. არ გასცემს შეტყობინებას შეცდომის შესახებ და არ შეასრულებს ცვლილებებს მთავარ ცხრილში
- გ. გასცემს შეტყობინებას შეცდომის შესახებ და შეასრულებს ცვლილებებს მთავარ ცხრილში

3.2.54. CHECK (ლოგიკური_გამოსახულება) პარამეტრი:

- ა. ახორციელებს მთლიანობის შეზღუდვას იმ შესაძლო მნიშვნელობების შემოწმების გზით, რომლებიც არ შეგვაქვს სვეტში ან სვეტებში
- ბ. არ ახორციელებს მთლიანობის შეზღუდვას
- გ. ახორციელებს მთლიანობის შეზღუდვას იმ შესაძლო მნიშვნელობების შემოწმების გზით, რომლებიც შეგვაქვს სვეტში ან სვეტებში

3.2.55. თუ CHECK (ლოგიკური_გამოსახულება) პარამეტრში ლოგიკური გამოსახულება იღებს true მნიშვნელობას, მაშინ მონაცემების შეცვლის ან ჩასმის ოპერაციები:

- ა. ნებადართული არ იქნება
- ბ. ნებადართული იქნება
- გ. ხანდახან იქნება ნებადართული

3.2.56. თუ CHECK (ლოგიკური_გამოსახულება) პარამეტრში ლოგიკური გამოსახულება იღებს false მნიშვნელობას, მაშინ მონაცემების შეცვლის ან ჩასმის ოპერაციები:

- ა. ნებადართული არ იქნება
- ბ. ხანდახან არ იქნება ნებადართული
- გ. ნებადართული იქნება

3.2.57. <ცხრილის_შეზღუდვა> კონსტრუქცია განსაზღვრავს მთლიანობის შეზღუდვებს:

- ა. სვეტის დონეზე
- ბ. მონაცემთა ბაზის დონეზე
- გ. ცხრილის დონეზე

3.2.58. <ცხრილის_შეზღუდვა> კონსტრუქციაში თუ მითითებულია ASC, მაშინ ინდექსში მონაცემები:

- ა. დახარისხდება კლებადობის მიხედვით
- ბ. არ დახარისხდება
- გ. დახარისხდება ზრდადობის მიხედვით

3.2.59. <ცხრილის_შეზღუდვა> კონსტრუქციაში თუ მითითებულია DESC, მაშინ ინდექსში მონაცემები:

- ა. დახარისხდება ზრდადობის მიხედვით
- ბ. დახარისხდება კლებადობის მიხედვით
- გ. არ დახარისხდება

3.2.60. ცხრილის წასაშლელად გამოიყენება ბრძანება:

- ა. DROP TABLE
- ბ. CREATE TABLE
- გ. DROP DATABASE

3.2.61. იმისათვის, რომ შევძლოთ ცხრილის წაშლა:

- ა. არ უნდა ვიყოთ სერვერის sysadmin სტანდარტული როლის წევრი, ან მონაცემთა ბაზის db_owner ან db_dba სტანდარტული როლის წევრი
- ბ. უნდა ვიყოთ სერვერის sysadmin სტანდარტული როლის წევრი, ან მონაცემთა ბაზის db_owner ან db_dba სტანდარტული როლის წევრი
- გ. არ უნდა ვიყოთ სერვერის sysadmin სტანდარტული როლის წევრი

3.2.62. ჩვენ ვერ წავშლით ცხრილს თუ

- ა. არსებობენ ცხრილები, რომლებიც არ მიმართავენ ამ ცხრილს გარე გასაღების საშუალებით ან თუ ამ ცხრილთან დაკავშირებულია ერთი ან მეტი წარმოდგენა
- ბ. არ არსებობენ ცხრილები, რომლებიც მიმართავენ ამ ცხრილს გარე გასაღების საშუალებით ან თუ ამ ცხრილთან დაკავშირებული არ არის ერთი ან მეტი წარმოდგენა
- გ. არსებობენ ცხრილები, რომლებიც მიმართავენ ამ ცხრილს გარე გასაღების საშუალებით ან თუ ამ ცხრილთან დაკავშირებულია ერთი ან მეტი წარმოდგენა

3.2.63.

```
CREATE TABLE Cxrili
(  
cxriliID INT PRIMARY KEY IDENTITY (1,1),  
sveti1 INT,  
sveti2 NVARCHAR(20),  
sveti3 FLOAT,  
sveti4 DATETIME  
);
```

ბრძანების შესრულების შედეგად:

- ა. შეიქმნება Cxrili ცხრილი, რომელსაც აქვს პირველადი გასაღები, მთელი, სტრიქონული, წილადი და თარიღის ტიპის მქონე სვეტები
- ბ. არ შეიქმნება Cxrili ცხრილი, რომელსაც აქვს პირველადი გასაღები, მთელი, სტრიქონული, წილადი და თარიღის ტიპის მქონე სვეტები
- გ. შეიქმნება Cxrili ცხრილი, რომელსაც არ აქვს პირველადი გასაღები; აქვს მთელი და თარიღის ტიპის მქონე სვეტები

3.2.64. რომელი ბრძანება შექმნის ცხრილს, რომელსაც აქვს პირველადი გასაღები, მთელი, სტრიქონული, წილადი და თარიღის ტიპის მქონე სვეტები:

ა. CREATE TABLE Cxrili
(
cxriliID INT PRIMARY KEY IDENTITY (1,1),
sveti1 INT,
sveti2 NVARCHAR(20),
sveti3 FLOAT,
sveti4 DATETIME
);

ბ. CREATE TABLE Cxrili
(
cxriliID INT,
sveti1 INT,
sveti2 NVARCHAR(20),
sveti3 FLOAT,
sveti4 DATETIME
);

გ. CREATE TABLE Cxrili
(
cxriliID INT PRIMARY KEY IDENTITY (1,1),
sveti1 INT,
sveti4 DATETIME
);

3.2.65. CREATE TABLE Cxrili

```
(  
cxriliID INT CONSTRAINT FK_C1 PRIMARY KEY (cxriliID DESC) IDENTITY (1,1),
```

```
sveti1 INT NULL,
sveti2 INT,
sveti3 NVARCHAR(20),
sveti4 FLOAT,
sveti5 DATETIME
```

); ბრძანების შესრულების შედეგად:

- ა. შეიქმნება ცხრილი, რომლის პირველადი გასაღების მნიშვნელობები დახარისხებული არ იქნება კლებადობით
- ბ. შეიქმნება ცხრილი, რომლის პირველადი გასაღების მნიშვნელობები დახარისხებული იქნება კლებადობით
- გ. შეიქმნება ცხრილი, რომლის პირველადი გასაღების მნიშვნელობები დახარისხებული იქნება ზრდადობით

3.2.66. რომელი ბრძანება შექმნის ცხრილს, რომლის პირველადი გასაღების მნიშვნელობები დახარისხებული იქნება კლებადობით:

ა. CREATE TABLE Cxrili

```
(
cxriliID INT PRIMARY KEY IDENTITY (1,1),
sveti1 INT NULL,
sveti2 INT,
sveti3 NVARCHAR(20),
sveti4 FLOAT,
sveti5 DATETIME
);
```

ბ. CREATE TABLE Cxrili

```
(
cxriliID INT CONSTRAINT FK_C1 PRIMARY KEY (cxriliID ASC) IDENTITY (1,1),
sveti1 INT NULL,
sveti2 INT,
sveti3 NVARCHAR(20),
sveti4 FLOAT,
sveti5 DATETIME
);
```

გ. CREATE TABLE Cxrili

```
(
cxriliID INT CONSTRAINT FK_C1 PRIMARY KEY (cxriliID DESC) IDENTITY (1,1),
sveti1 INT NULL,
sveti2 INT,
sveti3 NVARCHAR(20),
sveti4 FLOAT,
sveti5 DATETIME
);
```

3.2.67. CREATE TABLE Cxrili_2

```
(
cxrili_2ID INT PRIMARY KEY IDENTITY (1,1),
```

```
cxriliID INT NULL FOREIGN KEY (cxriliID) REFERENCES Cxrili(cxriliID),
sveti2 NVARCHAR(20),
sveti3 FLOAT,
sveti4 DATETIME
```

); ბრძანების შესრულების შედეგად:

- ა. შეიქმნება დამოკიდებული ცხრილი, რომლის cxriliID გარე გასაღები მიმართავს Cxrili ცხრილის cxriliID პირველად გასაღებს
- ბ. არ შეიქმნება დამოკიდებული ცხრილი, რომლის cxriliID გარე გასაღები მიმართავს Cxrili ცხრილის cxriliID პირველად გასაღებს
- გ. შეიქმნება დამოკიდებული ცხრილი, რომლის cxrili_2ID გარე გასაღები მიმართავს Cxrili_2 ცხრილის cxriliID პირველად გასაღებს

3.2.68. რომელი ბრძანება შექმნის დამოკიდებულ ცხრილს, რომლის cxriliID გარე გასაღები მიმართავს Cxrili ცხრილის cxriliID პირველად გასაღებს:

ა. CREATE TABLE Cxrili_2

```
(
cxrili_2ID INT PRIMARY KEY IDENTITY (1,1),
cxriliID INT NULL,
sveti2 NVARCHAR(20),
sveti3 FLOAT,
sveti4 DATETIME
```

);

ბ. CREATE TABLE Cxrili_2

```
(
cxrili_2ID INT PRIMARY KEY IDENTITY (1,1),
cxriliID INT NULL FOREIGN KEY (cxriliID) REFERENCES Cxrili(cxriliID),
sveti2 NVARCHAR(20),
sveti3 FLOAT,
sveti4 DATETIME
```

);

გ. CREATE TABLE Cxrili_2

```
(
cxrili_2ID INT PRIMARY KEY IDENTITY (1,1),
sveti3 FLOAT,
sveti4 DATETIME
```

);

3.2.69. CREATE TABLE Cxrili_3

```
(
cxrili_3ID INT PRIMARY KEY IDENTITY (1,1),
cxrili_1ID INT NULL REFERENCES Cxrili(cxriliID) ON DELETE CASCADE,
sveti2 NVARCHAR(20),
sveti3 FLOAT,
sveti4 DATETIME
```

);

ბრძანების შესრულების შედეგად:

ა. არ შეიქმნება Cxrili_3 დამოკიდებული ცხრილი, რომლის cxrili_3ID სვეტი იქნება გარე გასაღები და დაუკავშირდება მთავარი Cxrili_1 ცხრილის cxrili_1ID პირველად გასაღებს. განსაზღვრულია რეჟიმი, როცა მთავარი ცხრილიდან სტრიქონის წაშლისას დამოკიდებულ ცხრილში შესაბამისი სტრიქონები წაიშლება

ბ. შეიქმნება Cxrili_3 დამოკიდებული ცხრილი, რომლის cxrili_3ID სვეტი იქნება გარე გასაღები და დაუკავშირდება მთავარი Cxrili_1 ცხრილის cxrili_1ID პირველად გასაღებს. განსაზღვრულია რეჟიმი, როცა მთავარი ცხრილიდან სტრიქონის წაშლისას დამოკიდებულ ცხრილში შესაბამისი სტრიქონები არ წაიშლება

გ. შეიქმნება Cxrili_3 დამოკიდებული ცხრილი, რომლის cxrili_3ID სვეტი იქნება გარე გასაღები და დაუკავშირდება მთავარი Cxrili_1 ცხრილის cxrili_1ID პირველად გასაღებს. განსაზღვრულია რეჟიმი, როცა მთავარი ცხრილიდან სტრიქონის წაშლისას დამოკიდებულ ცხრილში შესაბამისი სტრიქონები წაიშლება

3.2.70. რომელი ბრძანება შექმნის Cxrili_3 დამოკიდებულ ცხრილს, რომლის cxrili_3ID სვეტი იქნება გარე გასაღები და დაუკავშირდება მთავარი Cxrili_1 ცხრილის cxrili_1ID პირველად გასაღებს. განსაზღვრულია რეჟიმი, როცა მთავარი ცხრილიდან სტრიქონის წაშლისას დამოკიდებულ ცხრილში შესაბამისი სტრიქონები წაიშლება:

ა. CREATE TABLE Cxrili_3

```
(  
cxrili_3ID INT PRIMARY KEY IDENTITY (1,1),  
cxrili_1ID INT NULL REFERENCES Cxrili(cxriliID) ON DELETE CASCADE,  
sveti2 NVARCHAR(20),  
sveti3 FLOAT,  
sveti4 DATETIME  
);
```

ბ. CREATE TABLE Cxrili_3

```
(  
cxrili_3ID INT PRIMARY KEY IDENTITY (1,1),  
cxrili_1ID INT NULL REFERENCES Cxrili(cxriliID),  
sveti2 NVARCHAR(20),  
sveti3 FLOAT,  
sveti4 DATETIME  
);
```

გ. CREATE TABLE Cxrili_3

```
(  
cxrili_3ID INT PRIMARY KEY IDENTITY (1,1),  
cxrili_1ID INT,  
sveti2 NVARCHAR(20),  
sveti3 FLOAT,  
sveti4 DATETIME  
);
```

3.2.71. CREATE TABLE Cxrili_3

```
(  
cxrili_3ID INT PRIMARY KEY IDENTITY (1,1),  
cxrili_1ID INT NULL REFERENCES Cxrili(cxriliID) ON UPDATE CASCADE,
```

```
sveti2 NVARCHAR(20),
sveti3 FLOAT,
sveti4 DATETIME
);
```

ბრძანების შესრულების შედეგად:

ა. არ შეიქმნება Cxrili_3 დამოკიდებული ცხრილი, რომლის cxrili_3ID სვეტი იქნება გარე გასაღები და დაუკავშირდება მთავარი Cxrili_1 ცხრილის cxrili_1ID პირველად გასაღებს. განსაზღვრულია რეჟიმი, როცა მთავარი ცხრილის პირველადი გასაღების მნიშვნელობის შეცვლისას დამოკიდებულ ცხრილში გარე გასაღების მნიშვნელობები შესაბამისად შეიცვლება

ბ. შეიქმნება Cxrili_3 დამოკიდებული ცხრილი, რომლის cxrili_3ID სვეტი იქნება გარე გასაღები და დაუკავშირდება მთავარი Cxrili_1 ცხრილის cxrili_1ID პირველად გასაღებს. განსაზღვრულია რეჟიმი, როცა მთავარი ცხრილის პირველადი გასაღების მნიშვნელობის შეცვლისას დამოკიდებულ ცხრილში გარე გასაღების მნიშვნელობები შესაბამისად შეიცვლება

გ. შეიქმნება Cxrili_3 დამოკიდებული ცხრილი, რომლის cxrili_3ID სვეტი იქნება გარე გასაღები და დაუკავშირდება მთავარი Cxrili_1 ცხრილის cxrili_1ID პირველად გასაღებს. არ არის განსაზღვრული რეჟიმი, როცა მთავარი ცხრილის პირველადი გასაღების მნიშვნელობის შეცვლისას დამოკიდებულ ცხრილში გარე გასაღების მნიშვნელობები შესაბამისად შეიცვლება

3.2.72. რომელი ბრძანება შექმნის Cxrili_3 დამოკიდებულ ცხრილს, რომლის cxrili_3ID სვეტი იქნება გარე გასაღები და დაუკავშირდება მთავარი Cxrili_1 ცხრილის cxrili_1ID პირველად გასაღებს. განსაზღვრება რეჟიმი, როცა მთავარი ცხრილის პირველადი გასაღების მნიშვნელობის შეცვლისას დამოკიდებულ ცხრილში გარე გასაღების მნიშვნელობები შესაბამისად შეიცვლება

ა. CREATE TABLE Cxrili_3

```
(
cxrili_3ID INT PRIMARY KEY IDENTITY (1,1),
cxrili_1ID INT NULL REFERENCES Cxrili(cxriliID),
sveti2 NVARCHAR(20),
sveti3 FLOAT,
sveti4 DATETIME
```

);

ბ. CREATE TABLE Cxrili_3

```
(
cxrili_3ID INT PRIMARY KEY IDENTITY (1,1),
cxrili_1ID INT NULL,
sveti2 NVARCHAR(20),
sveti4 DATETIME
```

);

გ. CREATE TABLE Cxrili_3

```
(
cxrili_3ID INT PRIMARY KEY IDENTITY (1,1),
cxrili_1ID INT NULL REFERENCES Cxrili(cxriliID) ON UPDATE CASCADE,
sveti2 NVARCHAR(20),
sveti3 FLOAT,
sveti4 DATETIME
```

);

3.2.73. CREATE TABLE Cxrili_3

```
(  
cxrili_3ID INT PRIMARY KEY IDENTITY (1,1),  
cxrili_1ID INT NULL REFERENCES Cxrili(cxriliID) ON UPDATE CASCADE  
ON DELETE CASCADE,  
sveti2 NVARCHAR(20),  
sveti3 FLOAT,  
sveti4 DATETIME  
);
```

ბრძანების შესრულების შედეგად:

ა. შეიქმნება Cxrili_3 ცხრილი, რომლის cxrili_3ID სვეტი იქნება გარე გასაღები და დაუკავშირდება მთავარი Cxrili_1 ცხრილის cxrili_1ID პირველად გასაღებს. განისაზღვრება რეჟიმი, როცა მთავარ ცხრილში პირველადი გასაღების მნიშვნელობის შეცვლისას დამოკიდებულ ცხრილში გარე გასაღების მნიშვნელობები შესაბამისად შეიცვლება, ხოლო მთავარი ცხრილიდან სტრიქონის წაშლისას კი - დამოკიდებულ ცხრილში შესაბამისი სტრიქონები წაიშლება

ბ. არ შეიქმნება Cxrili_3 ცხრილი, რომლის cxrili_3ID სვეტი იქნება გარე გასაღები და დაუკავშირდება მთავარი Cxrili_1 ცხრილის cxrili_1ID პირველად გასაღებს. განისაზღვრება რეჟიმი, როცა მთავარ ცხრილში პირველადი გასაღების მნიშვნელობის შეცვლისას დამოკიდებულ ცხრილში გარე გასაღების მნიშვნელობები შესაბამისად შეიცვლება, ხოლო მთავარი ცხრილიდან სტრიქონის წაშლისას კი - დამოკიდებულ ცხრილში შესაბამისი სტრიქონები წაიშლება

გ. შეიქმნება Cxrili_3 ცხრილი, რომლის cxrili_3ID სვეტი იქნება გარე გასაღები და დაუკავშირდება მთავარი Cxrili_1 ცხრილის cxrili_1ID პირველად გასაღებს. არ განისაზღვრება რეჟიმი, როცა მთავარ ცხრილში პირველადი გასაღების მნიშვნელობის შეცვლისას დამოკიდებულ ცხრილში გარე გასაღების მნიშვნელობები შესაბამისად შეიცვლება, ხოლო მთავარი ცხრილიდან სტრიქონის წაშლისას კი - დამოკიდებულ ცხრილში შესაბამისი სტრიქონები წაიშლება

3.2.74. რომელი ბრძანება შექმნის Cxrili_3 ცხრილი, რომლის cxrili_3ID სვეტი იქნება გარე გასაღები და დაუკავშირდება მთავარი Cxrili_1 ცხრილის cxrili_1ID პირველად გასაღებს. განისაზღვრება რეჟიმი, როცა მთავარ ცხრილში პირველადი გასაღების მნიშვნელობის შეცვლისას დამოკიდებულ ცხრილში გარე გასაღების მნიშვნელობები შესაბამისად შეიცვლება, ხოლო მთავარი ცხრილიდან სტრიქონის წაშლისას კი - დამოკიდებულ ცხრილში შესაბამისი სტრიქონები წაიშლება

ა. CREATE TABLE Cxrili_3

```
(  
cxrili_3ID INT PRIMARY KEY IDENTITY (1,1),  
cxrili_1ID INT NULL REFERENCES Cxrili(cxriliID) ON DELETE CASCADE,  
sveti2 NVARCHAR(20),  
sveti3 FLOAT,  
sveti4 DATETIME  
);
```

ბ. CREATE TABLE Cxrili_3

```
(  
cxrili_3ID INT PRIMARY KEY IDENTITY (1,1),
```

```

cxrili_1ID INT NULL REFERENCES Cxrili(cxriliID) ON UPDATE CASCADE
ON DELETE CASCADE,
sveti2 NVARCHAR(20),
sveti3 FLOAT,
sveti4 DATETIME
);
გ. CREATE TABLE Cxrili_3
(
cxrili_3ID INT PRIMARY KEY IDENTITY (1,1),
cxrili_1ID INT NULL REFERENCES Cxrili(cxriliID) ON UPDATE CASCADE,
sveti2 NVARCHAR(20),
sveti3 FLOAT,
sveti4 DATETIME
);

```

3.2.75. CREATE TABLE Cxrili_4

```

(
cxrili_4ID INT PRIMARY KEY IDENTITY (1,1),
cxrili_1ID INT NULL,
sveti1 INT,
sveti2 NVARCHAR(20),
sveti3 FLOAT,
sveti4 DATETIME,
CONSTRAINT FK_C2 FOREIGN KEY (cxrili_1ID, sveti1)
REFERENCES Cxrili_1(cxrili_1ID, sveti1)
ON DELETE CASCADE
);

```

ბრძანების შესრულების შედეგად:

ა. შეიქმნება დამოკიდებული Cxrili_4 ცხრილი, რომლის გარე გასაღები ერთი სვეტისაგან შედგება: cxrili_1ID და sveti1. ისინი მიმართავენ მთავარი Cxrili_1 ცხრილის პირველად გასაღებს, რომელიც ასევე, ორი სვეტისაგან შედგება: cxrili_1ID და sveti1. Cxrili_4 ცხრილისთვის შეიქმნება ცხრილის შეზღუდვა, რომლის სახელია FK_C2. ის არ განსაზღვრავს რეჟიმს, როცა მთავარი ცხრილიდან სტრიქონის წაშლისას დამოკიდებულ ცხრილში შესაბამისი სტრიქონები წაიშლება.

ბ. არ შეიქმნება დამოკიდებული Cxrili_4 ცხრილი, რომლის გარე გასაღები ორი სვეტისაგან შედგება: cxrili_1ID და sveti1. ისინი მიმართავენ მთავარი Cxrili_1 ცხრილის პირველად გასაღებს, რომელიც ასევე, ორი სვეტისაგან შედგება: cxrili_1ID და sveti1. Cxrili_4 ცხრილისთვის შეიქმნება ცხრილის შეზღუდვა, რომლის სახელია FK_C2. ის განსაზღვრავს რეჟიმს, როცა მთავარი ცხრილიდან სტრიქონის წაშლისას დამოკიდებულ ცხრილში შესაბამისი სტრიქონები წაიშლება.

გ. შეიქმნება დამოკიდებული Cxrili_4 ცხრილი, რომლის გარე გასაღები ორი სვეტისაგან შედგება: cxrili_1ID და sveti1. ისინი მიმართავენ მთავარი Cxrili_1 ცხრილის პირველად გასაღებს, რომელიც ასევე, ორი სვეტისაგან შედგება: cxrili_1ID და sveti1. Cxrili_4 ცხრილისთვის შეიქმნება ცხრილის შეზღუდვა, რომლის სახელია FK_C2. ის განსაზღვრავს რეჟიმს, როცა მთავარი ცხრილიდან სტრიქონის წაშლისას დამოკიდებულ ცხრილში შესაბამისი სტრიქონები წაიშლება.

3.2.76. რომელი ბრძანება შექმნის დამოკიდებულ Cxrili_4 ცხრილს, რომლის გარე გასაღები ორი სვეტისაგან შედგება: cxrili_1ID და sveti1. ისინი მიმართავენ მთავარი Cxrili_1 ცხრილის

პირველად გასაღებს, რომელიც ასევე, ორი სვეტისაგან შედგება: cxrili_1ID და sveti1. Cxrili_4 ცხრილისთვის შეიქმნება ცხრილის შეზღუდვა, რომლის სახელია FK_C2. ის განსაზღვრავს რეჟიმს, როცა მთავარი ცხრილიდან სტრიქონის წაშლისას დამოკიდებულ ცხრილში შესაბამისი სტრიქონები წაიშლება:

```
ა. CREATE TABLE Cxrili_4
(
cxrili_4ID INT PRIMARY KEY IDENTITY (1,1),
cxrili_1ID INT NULL,
sveti1 INT,
sveti2 NVARCHAR(20),
sveti3 FLOAT,
sveti4 DATETIME,
CONSTRAINT FK_C2 FOREIGN KEY (cxrili_1ID, sveti1)
REFERENCES Cxrili_1(cxrili_1ID, sveti1)
ON DELETE CASCADE
);
```

```
ბ. CREATE TABLE Cxrili_4
(
cxrili_4ID INT PRIMARY KEY IDENTITY (1,1),
cxrili_1ID INT NULL,
sveti1 INT,
sveti2 NVARCHAR(20),
sveti3 FLOAT,
sveti4 DATETIME
);
```

```
გ. CREATE TABLE Cxrili_4
(
cxrili_4ID INT PRIMARY KEY IDENTITY (1,1),
cxrili_1ID INT NULL,
sveti1 INT,
sveti2 NVARCHAR(20),
sveti3 FLOAT,
sveti4 DATETIME,
CONSTRAINT FK_C2 FOREIGN KEY (cxrili_1ID, sveti1)
REFERENCES Cxrili_1(cxrili_1ID, sveti1)
);
```

3.2.77. CREATE TABLE Cxrili_5

```
(
cxrili_5ID INT PRIMARY KEY IDENTITY (1,1),
sveti1 INT NOT NULL UNIQUE,
sveti2 NVARCHAR(20) NULL UNIQUE,
sveti3 FLOAT,
sveti4 DATETIME
);
```

ბრძანების შესრულების შედეგად:

- ა. არ შეიქმნება Cxrili_5 ცხრილი, რომლის sveti1 და sveti2 სვეტებისთვის განსაზღვრული იქნება UNIQUE შეზღუდვა
- ბ. შეიქმნება Cxrili_5 ცხრილი, რომლის sveti1 და sveti2 სვეტებისთვის განსაზღვრული იქნება UNIQUE შეზღუდვა
- გ. შეიქმნება Cxrili_5 ცხრილი, რომლის sveti1 და sveti2 სვეტებისთვის განსაზღვრული არ იქნება UNIQUE შეზღუდვა

3.2.78. რომელი ბრძანება შექმნის Cxrili_5 ცხრილს, რომლის sveti1 და sveti2 სვეტებისთვის განსაზღვრული იქნება UNIQUE შეზღუდვა:

ა. CREATE TABLE Cxrili_5
(
cxrili_5ID INT PRIMARY KEY IDENTITY (1,1),
sveti1 INT NOT NULL UNIQUE,
sveti2 NVARCHAR(20),
sveti3 FLOAT,
sveti4 DATETIME
);

ბ. CREATE TABLE Cxrili_5
(
cxrili_5ID INT PRIMARY KEY IDENTITY (1,1),
sveti1 INT NOT NULL,
sveti2 NVARCHAR(20) NULL UNIQUE,
sveti3 FLOAT,
sveti4 DATETIME
);

გ. CREATE TABLE Cxrili_5
(
cxrili_5ID INT PRIMARY KEY IDENTITY (1,1),
sveti1 INT NOT NULL UNIQUE,
sveti2 NVARCHAR(20) NULL UNIQUE,
sveti3 FLOAT,
sveti4 DATETIME
);

3.2.79. CREATE TABLE Cxrili_5
(
cxrili_5ID INT PRIMARY KEY IDENTITY (1,1),
sveti1 INT DEFAULT 25,
sveti2 NVARCHAR(20) DEFAULT N'რომან სამხარაძე',
sveti3 FLOAT,
sveti4 DATETIME DEFAULT GETDATE()
);

ბრძანების შესრულების შედეგად:

- ა. შეიქმნება Cxrili_5 ცხრილი, რომლის sveti1, sveti2 და sveti4 სვეტებისთვის განსაზღვრული იქნება ნაგულისხმევი მნიშვნელობები
- ბ. შეიქმნება Cxrili_5 ცხრილი, რომლის sveti2 და sveti4 სვეტებისთვის განსაზღვრული იქნება

ნაგულისხმევი მნიშვნელობები

გ. შეიქმნება Cxrili_5 ცხრილი, რომლის sveti1 სვეტისთვის განსაზღვრული იქნება ნაგულისხმევი მნიშვნელობები

3.2.80. რომელი ბრძანების შესრულების შედეგად შეიქმნება Cxrili_5 ცხრილი, რომლის sveti1, sveti2 და sveti4 სვეტებისთვის განსაზღვრული იქნება ნაგულისხმევი მნიშვნელობები:

ა. CREATE TABLE Cxrili_5

```
(  
cxrili_5ID INT PRIMARY KEY IDENTITY (1,1),  
sveti1 INT DEFAULT 25,  
sveti2 NVARCHAR(20),  
sveti3 FLOAT,  
sveti4 DATETIME DEFAULT GETDATE()  
);
```

ბ. CREATE TABLE Cxrili_5

```
(  
cxrili_5ID INT PRIMARY KEY IDENTITY (1,1),  
sveti1 INT DEFAULT 25,  
sveti2 NVARCHAR(20) DEFAULT N'რომან სამხარაძე',  
sveti3 FLOAT,  
sveti4 DATETIME DEFAULT GETDATE()  
);
```

გ. CREATE TABLE Cxrili_5

```
(  
cxrili_5ID INT PRIMARY KEY IDENTITY (1,1),  
sveti1 INT,  
sveti2 NVARCHAR(20) DEFAULT N'რომან სამხარაძე',  
sveti3 FLOAT,  
sveti4 DATETIME  
);
```

3.2.81. CREATE TABLE Cxrili_5

```
(  
cxrili_5ID INT PRIMARY KEY IDENTITY (1,1),  
sveti1 INT CHECK ( sveti1 > 0 AND sveti1 < 10 ),  
sveti2 NVARCHAR(20) CHECK (sveti2 IN ('1389', '0736', '1756') OR sveti2 LIKE '99[0-9][0-9]'),  
sveti3 FLOAT,  
sveti4 DATETIME CHECK ( sveti4 >= '05.10.2000' AND sveti4 <= '19.03.2005' )  
);
```

ბრძანების შესრულების შედეგად:

ა. შეიქმნება ცხრილი, რომლის sveti2 და sveti4 სვეტებისთვის განსაზღვრული იქნება შეზღუდვები

ბ. შეიქმნება ცხრილი, რომლის sveti1 და sveti4 სვეტებისთვის განსაზღვრული იქნება შეზღუდვები

გ. შეიქმნება ცხრილი, რომლის sveti1, sveti2 და sveti4 სვეტებისთვის განსაზღვრული იქნება შეზღუდვები

3.2.82. რომელი ბრძანებით შეიქმნება ცხრილი, რომლის sveti1, sveti2 და sveti4 სვეტებისთვის განსაზღვრული იქნება შეზღუდვები:

ა. CREATE TABLE Cxrili_5
(
cxrili_5ID INT PRIMARY KEY IDENTITY (1,1),
sveti1 INT CHECK (sveti1 > 0 AND sveti1 < 10),
sveti2 NVARCHAR(20) CHECK (sveti2 IN ('1389', '0736', '1756') OR sveti2 LIKE '99[0-9][0-9]'),
sveti3 FLOAT,
sveti4 DATETIME CHECK (sveti4 >= '05.10.2000' AND sveti4 <= '19.03.2005')
);

ბ. CREATE TABLE Cxrili_5
(
cxrili_5ID INT PRIMARY KEY IDENTITY (1,1),
sveti1 INT,
sveti2 NVARCHAR(20) CHECK (sveti2 IN ('1389', '0736', '1756') OR sveti2 LIKE '99[0-9][0-9]'),
sveti3 FLOAT,
sveti4 DATETIME CHECK (sveti4 >= '05.10.2000' AND sveti4 <= '19.03.2005')
);

გ. CREATE TABLE Cxrili_5
(
cxrili_5ID INT PRIMARY KEY IDENTITY (1,1),
sveti1 INT,
sveti2 NVARCHAR(20) CHECK (sveti2 IN ('1389', '0736', '1756') OR sveti2 LIKE '99[0-9][0-9]'),
sveti3 FLOAT,
sveti4 DATETIME
);

3.2.83. CREATE TABLE Cxrili_5
(
cxrili_5ID INT PRIMARY KEY IDENTITY (1,1),
sveti1 INT,
sveti2 INT,
sveti3 AS sveti1 + sveti2 / 2,
sveti4 AS USER_NAME(),
sveti5 AS GETDATE()
);

ბრძანების შესრულების შედეგად:

- ა. შეიქმნება ცხრილი, რომლის sveti3, sveti4 და sveti5 სვეტები იქნება გამოთვლადი
- ბ. არ შეიქმნება ცხრილი, რომელსაც ექნება გამოთვლადი სვეტები
- გ. შეიქმნება ცხრილი, რომლის sveti4 და sveti5 სვეტები იქნება გამოთვლადი

3.2.84. რომელი ბრძანებით შეიქმნება ცხრილი, რომლის sveti3, sveti4 და sveti5 სვეტები იქნება გამოთვლადი:

ა. CREATE TABLE Cxrili_5
(

```

cxrili_5ID INT PRIMARY KEY IDENTITY (1,1),
sveti1 INT,
sveti2 INT,
sveti3 AS sveti1 + sveti2 / 2,
sveti4 AS USER_NAME(),
sveti5 INT
);

```

```

ბ. CREATE TABLE Cxrili_5
(
cxrili_5ID INT PRIMARY KEY IDENTITY (1,1),
sveti1 INT,
sveti2 INT,
sveti3 AS sveti1 + sveti2 / 2,
sveti4 AS USER_NAME(),
sveti5 AS GETDATE()
);

```

```

გ. CREATE TABLE Cxrili_5
(
cxrili_5ID INT PRIMARY KEY IDENTITY (1,1),
sveti1 INT,
sveti2 INT,
sveti3 INT,
sveti4 AS USER_NAME(),
sveti5 AS GETDATE()
);

```

3.2.85. DROP TABLE Cxrili_1; ბრძანების შესრულების შედეგად:

- ა. შეიქმნება Cxrili_1 ცხრილი
- ბ. არ წაიშლება Cxrili_1 ცხრილი
- გ. წაიშლება Cxrili_1 ცხრილი

3.2.86. რომელი ბრძანება წაშლის Cxrili_1 ცხრილს:

- ა. DROP TABLE Cxrili_1;
- ბ. DROP TABLE Cxrili_2;
- გ. CREATE TABLE Cxrili_1;

ცხრილისათვის სახელის შეცვლა. ინფორმაციის მიღება ცხრილების შესახებ. დიაგრამები

3.3.1. ცხრილისათვის სახელის შესაცვლელად გამოიყენება:

- ა. sp_rename შენახული პროცედურა
- ბ. sp_help შენახული პროცედურა
- გ. sp_depends შენახული პროცედურა

3.3.2. ცხრილის სვეტისთვის სახელის შესაცვლელად გამოიყენება:

- ა. sp_help შენახული პროცედურა

- ბ. sp_rename შენახული პროცედურა
 - გ. sp_depends შენახული პროცედურა
- 3.3.3. sp_rename 'Cxrili10', 'Cxrili20' პროცედურის შესრულების შედეგად:
- ა. Cxrili10 ცხრილს დაერქმევა Cxrili10 სახელი
 - ბ. Cxrili20 ცხრილს დაერქმევა Cxrili10 სახელი
 - გ. Cxrili10 ცხრილს დაერქმევა Cxrili20 სახელი
- 3.3.4. sp_rename 'Personali_1.PersonaliID', 'ID', 'COLUMN' პროცედურის შესრულების შედეგად:
- ა. Personali_1 ცხრილის PersonaliID სვეტს დავარქვათ ახალი COLUMN სახელი
 - ბ. Personali_1 ცხრილის PersonaliID სვეტს დავარქვათ ახალი ID სახელი
 - გ. Personali_1 ცხრილის PersonaliID სვეტს დავარქვათ ახალი Personali_1 სახელი
- 3.3.5. ინფორმაცია მონაცემთა ბაზაში არსებული ცხრილების შესახებ ინახება:
- ა. ამავე მონაცემთა ბაზის sys.objects წარმოდგენაში
 - ბ. master მონაცემთა ბაზის sys.objects წარმოდგენაში
 - გ. ამავე მონაცემთა ბაზის all_columns წარმოდგენაში
- 3.3.6. კონკრეტული ცხრილის შესახებ ინფორმაციის მისაღებად გამოიყენება:
- ა. sp_depends შენახული პროცედურა
 - ბ. OBJECTPROPERTY ფუნქცია
 - გ. sp_rename შენახული პროცედურა
- 3.3.7. SELECT * FROM sys.objects WHERE id = object_id(N'[dbo].[Personali]') AND OBJECTPROPERTY(id, N'IsUserTable') = 1 მოთხოვნის შესრულების შედეგად გაიცემა ინფორმაცია:
- ა. object_id ფუნქციის შესახებ
 - ბ. id სვეტის შესახებ
 - გ. Personali ცხრილის შესახებ
- 3.3.8. ცხრილის თვისებების მისაღებად შეგვიძლია გამოვიყენოთ სისტემური შენახული პროცედურა:
- ა. sp_help
 - ბ. sp_rename
 - გ. sp_depends
- 3.3.9. sp_help Personali პროცედურის შესრულების შედეგად გაიცემა ინფორმაცია:
- ა. sp_help პროცედურის შესახებ
 - ბ. Personali ცხრილის თვისებების შესახებ
 - გ. ორივე ობიექტის შესახებ
- 3.3.10. ინფორმაცია ცხრილზე დამოკიდებული ობიექტების შესახებ სასარგებლოა მაშინ, როცა:
- ა. ვაპირებთ ცხრილის შექმნას
 - ბ. არ ვაპირებთ ცხრილის წაშლას
 - გ. ვაპირებთ ცხრილის წაშლას
- 3.3.11. ინფორმაციის მისაღებად ცხრილზე დამოკიდებული ობიექტების შესახებ უნდა

გამოვიყენოთ სისტემური პროცედურა:

- ა. sp_rename
- ბ. sp_depends
- გ. sp_help

3.3.12. sp_depends 'Personali' პროცედურის შესრულების შედეგად გაიცემა ინფორმაცია:

- ა. Personali ცხრილზე დამოკიდებული ობიექტების შესახებ
- ბ. Personali ცხრილზე დამოუკიდებელი ობიექტების შესახებ
- გ. sp_depends პროცედურაზე დამოკიდებული ობიექტების შესახებ

3.3.13. ინფორმაცია იმის შესახებ, თუ მოცემული ცხრილი რომელ ცხრილებს უკავშირდება PRIMARY KEY და FOREIGN KEY შეზღუდვების საშუალებით, შეგვიძლია მივიღოთ:

- ა. sp_depends შენახული პროცედურის საშუალებით
- ბ. sp_fkeys შენახული პროცედურის საშუალებით
- გ. sp_rename შენახული პროცედურის საშუალებით

3.3.14. თუ sp_fkeys შენახულ პროცედურაში მივუთითებთ მხოლოდ პირველადი გასაღების შემცველი ცხრილის პარამეტრებს, მაშინ გაიცემა ინფორმაცია:

- ა. ამ ცხრილის პირველად გასაღებთან დაუკავშირებული ცხრილის შესახებ
- ბ. ამ ცხრილის გარე გასაღებთან დაკავშირებული ყველა ცხრილის შესახებ
- გ. ამ ცხრილის პირველად გასაღებთან დაკავშირებული ყველა ცხრილის შესახებ

3.3.15. თუ sp_fkeys შენახულ პროცედურაში მივუთითებთ გარე გასაღების შემცველი ცხრილის პარამეტრებს, მაშინ გაიცემა ინფორმაცია:

- ა. ამ ცხრილის პირველად გასაღებთან დაკავშირებული ყველა ცხრილის შესახებ
- ბ. იმ ცხრილების შესახებ, რომლებიც დაკავშირებულია ამ ცხრილის პირველად გასაღებთან
- გ. ამ ცხრილის პირველად გასაღებთან დაუკავშირებული ცხრილის შესახებ

3.3.16. sp_fkeys @pktable_name = N'Personali' პროცედურის შესრულების შედეგად გაიცემა ინფორმაცია:

- ა. Personali ცხრილთან დაკავშირებული ცხრილების შესახებ. ეს ცხრილი დამოკიდებული ცხრილია
- ბ. Personali ცხრილთან დაკავშირებული ცხრილების შესახებ. ეს ცხრილი მთავარი ცხრილია
- გ. @pktable_name ცხრილთან დაკავშირებული ცხრილების შესახებ

3.3.17. sp_fkeys @fktable_name = N'Xelshekruleba' პროცედურის შესრულების შედეგად გაიცემა ინფორმაცია:

- ა. Xelshekruleba ცხრილთან დაკავშირებული ცხრილების შესახებ. ეს ცხრილი დამოკიდებული ცხრილია
- ბ. @fktable_name ცხრილთან დაკავშირებული ცხრილების შესახებ
- გ. Xelshekruleba ცხრილთან დაკავშირებული ცხრილების შესახებ. ეს ცხრილი მთავარი ცხრილია

3.3.18. როცა მონაცემთა ბაზა ერთმანეთთან დაკავშირებულ ბევრ ცხრილს შეიცავს, მისი სტრუქტურის თვალსაჩინოდ წარმოსადგენად უნდა გამოვიყენოთ:

- ა. გრაფიკები

- ბ. დიაგრამები
- გ. სვეტები

3.3.19. დიაგრამების დანიშნულებაა მონაცემთა ბაზის:

- ა. მონაცემთა ბაზის შექმნა
- ბ. ადმინისტრირების გართულება
- გ. ადმინისტრირების გამარტივება

3.3.20. EXEC sp_rename 'Cxrili_4', 'Cxrili_5'; შენახული პროცედურის შესრულების შედეგად:

- ა. Cxrili_4 ცხრილს დაერქმევა ახალი Cxrili_5 სახელი
- ბ. Cxrili_4 ცხრილს არ დაერქმევა ახალი Cxrili_5 სახელი
- გ. Cxrili_5 ცხრილს დაერქმევა ახალი Cxrili_4 სახელი

3.3.21. რომელი ბრძანების შესრულების შედეგად დაერქმევა Cxrili_4 ცხრილს ახალი Cxrili_5 სახელი:

- ა. EXEC sp_rename 'Cxrili_5', 'Cxrili_4';
- ბ. EXEC sp_rename 'Cxrili_4', 'Cxrili_5';
- გ. EXEC sp_renamedb 'Cxrili_4', 'Cxrili_5';

3.3.22. EXEC sp_help Personal; შენახული პროცედურის შესრულების შედეგად:

- ა. გაიცემა Baza_1 ცხრილის თვისებები
- ბ. არ გაიცემა Personal ცხრილის თვისებები
- გ. გაიცემა Personal ცხრილის თვისებები

3.3.23. რომელი ბრძანება გასცემს Personal ცხრილის თვისებებს:

- ა. EXEC sp_help Personal;
- ბ. EXEC sp_help Baza_1;
- გ. EXEC sp_rename Personal;

3.3.24. EXEC sp_depends 'Personal'; შენახული პროცედურის შესრულების შედეგად:

- ა. გაიცემა Personal ცხრილის სვეტების სია
- ბ. არ გაიცემა Personal ცხრილზე დამოკიდებული ობიექტების სახელები და ტიპები
- გ. გაიცემა Personal ცხრილზე დამოკიდებული ობიექტების სახელები და ტიპები

3.3.25. რომელი ბრძანება გასცემს Personal ცხრილზე დამოკიდებულ ობიექტების სახელებს და ტიპებს:

- ა. EXEC sp_depends 'Personal';
- ბ. EXEC sp_help Personal;
- გ. EXEC sp_rename Personal;

3.3.26. EXEC sp_fkeys @pktable_name = N'Personal'; შენახული პროცედურის შესრულების შედეგად:

- ა. გაიცემა ინფორმაცია იმ ცხრილების შესახებ, რომლებიც Personal ცხრილთან არ არის დაკავშირებული
- ბ. გაიცემა ინფორმაცია Personal ცხრილთან დაკავშირებული ცხრილების შესახებ
- გ. არ გაიცემა ინფორმაცია Personal ცხრილთან დაკავშირებული ცხრილების შესახებ

3.3.27. რომელი ბრძანება გასცემს ინფორმაციას Personali ცხრილთან დაკავშირებული ცხრილების შესახებ:

- ა. EXEC sp_rename Personali;
- ბ. EXEC sp_help Personali;
- გ. EXEC sp_fkeys @pktable_name = N'Personali';

თავი 4. მონაცემების მართვა

Transact SQL-ის საფუძვლები

4.1.1. იდენტიფიკატორები გამოიყენება:

- ა. კონკრეტულ ობიექტთან მიმართვისათვის, როგორცაა ცვლადი, ცხრილი, წარმოდგენა, სვეტი, ინდექსი, ტრიგერი და ა.შ.
- ბ. არაკონკრეტულ ობიექტთან მიმართვისათვის
- გ. მხოლოდ ცვლადებთან მიმართვისათვის

4.1.2. იდენტიფიკატორი:

- ა. ხანდახან არის ობიექტის სახელი
- ბ. არის ობიექტის სახელი
- გ. არ არის ობიექტის სახელი

4.1.3. კომენტარი:

- ა. არ არის ტექსტი, რომელიც მოთავსებულია ბრძანების კოდში და შეიცავს მის განმარტებას
- ბ. არის ტექსტი, რომელიც მოთავსებული არ არის ბრძანების კოდში და შეიცავს მის განმარტებას
- გ. არის ტექსტი, რომელიც მოთავსებულია ბრძანების კოდში და შეიცავს მის განმარტებას

4.1.4. Transact-SQL-ის მიერ კომენტარის დამუშავება:

- ა. ხდება
- ბ. არ ხდება
- გ. ხანდახან ხდება

4.1.5. არსებობს:

- ა. ორი ტიპის კომენტარი - სტრიქონული და ბლოკური
- ბ. ერთი ტიპის კომენტარი - სტრიქონული
- გ. ერთი ტიპის კომენტარი - ბლოკური

4.1.6. სტრიქონული კომენტარი:

- ა. „/*” სიმბოლოებით იწყება და ერთ სტრიქონს იკავებს
- ბ. „--“ სიმბოლოებით იწყება და რამდენიმე სტრიქონს იკავებს
- გ. „--“ სიმბოლოებით იწყება და ერთ სტრიქონს იკავებს

4.1.7. ბლოკური კომენტარები:

- ა. „/*” სიმბოლოებით არ იწყება და „*/” სიმბოლოებით არ მთავრდება და რამდენიმე სტრიქონს იკავებს
- ბ. „/*” სიმბოლოებით იწყება და „*/” სიმბოლოებით მთავრდება და რამდენიმე სტრიქონს იკავებს
- გ. „/*” სიმბოლოებით იწყება და „*/” სიმბოლოებით მთავრდება და მხოლოდ ერთ სტრიქონს იკავებს

4.1.8. დარეზერვებული საკვანძო სიტყვები გამოიყენება სერვერის მუშაობის მართვისთვის და:

- ა. არიან დამოკიდებული რეგისტრზე და არ შეიძლება წარმოადგენდნენ იდენტიფიკატორებს

- ბ. არიან დამოკიდებული რეგისტრზე და შეიძლება წარმოადგენდნენ იდენტიფიკატორებს
- გ. არ არიან დამოკიდებული რეგისტრზე და არ შეიძლება წარმოადგენდნენ იდენტიფიკატორებს

4.1.9. გამოსახულება წარმოადგენს:

- ა. იდენტიფიკატორების, ფუნქციების, არითმეტიკისა და ლოგიკის ოპერაციების ნიშნების, მუდმივებისა და სხვა ობიექტების გადაკვეთას
- ბ. იდენტიფიკატორების, ფუნქციების, არითმეტიკისა და ლოგიკის ოპერაციების ნიშნების, მუდმივებისა და სხვა ობიექტების ჯამს
- გ. იდენტიფიკატორების, ფუნქციების, არითმეტიკისა და ლოგიკის ოპერაციების ნიშნების, მუდმივებისა და სხვა ობიექტების კომბინაციას

4.1.10. გამოსახულება:

- ა. შედგება მხოლოდ ოპერანდებისაგან
- ბ. შედგება ოპერანდებისა (მონაცემებისა) და ოპერატორებისაგან (ოპერაციის ნიშნებისაგან)
- გ. არ შედგება ოპერანდებისა (მონაცემებისა) და ოპერატორებისაგან (ოპერაციის ნიშნებისაგან)

4.1.11. მუდმივი სიდიდეებია, რომელთა მნიშვნელობები:

- ა. არ იცვლება
- ბ. იცვლება
- გ. ხანდახან იცვლება

4.1.12. მუდმივა შეიძლება:

- ა. იყოს მხოლოდ სტრიქონი
- ბ. იყოს მთელი რიცხვი, წილადი, სტრიქონი და ა.შ
- გ. არ იყოს მთელი რიცხვი, წილადი, სტრიქონი და ა.შ

4.1.13. ცვლადი არის:

- ა. გარკვეული ზომის მქონე მეხსიერების სახელდებული უბანი, რომელშიც მონაცემები ინახება
- ბ. გარკვეული ზომის არმქონე მეხსიერების სახელდებული უბანი, რომელშიც მონაცემები ინახება
- გ. გარკვეული ზომის მქონე მეხსიერების სახელდებული უბანი, რომელშიც მონაცემები არ ინახება

4.1.14. სვეტის სახელი შეიძლება გამოვიყენოთ როგორც:

- ა. ტრიგერი გამოსახულებაში
- ბ. ოპერანდი გამოსახულებაში
- გ. პროცედურა გამოსახულებაში

4.1.15. ცხრილის ერთი სტრიქონიდან მეორეზე გადასვლის დროს სვეტის სახელს:

- ა. შესაბამისი მნიშვნელობა არ ენიჭება
- ბ. მუდმივას მნიშვნელობა ენიჭება
- გ. შესაბამისი მნიშვნელობა ენიჭება

4.1.16. ფუნქცია არის სახელდებული, მცირე ზომის პროგრამა, რომელიც:

- ა. ახდენს მონაცემების დამუშავებას და შედეგის დაბრუნებას
- ბ. არ ახდენს მონაცემების დამუშავებას და შედეგის დაბრუნებას
- გ. ახდენს მონაცემების დამუშავებას და შედეგს არ აბრუნებს

4.1.17. ფუნქციას შეიძლება:

- ა. ჰქონდეს მხოლოდ ერთი პარამეტრი
- ბ. ჰქონდეს მხოლოდ ორი პარამეტრი
- გ. ჰქონდეს ან არ ჰქონდეს პარამეტრები

4.1.18. სტანდარტული იდენტიფიკატორების განსაზღვრისას:

- ა. დაცულია იდენტიფიკატორების შექმნის წესები
- ბ. არ არის დაცული იდენტიფიკატორების შექმნის წესები
- გ. დაცულია იდენტიფიკატორების წაშლის წესები

4.1.19. შეზღუდული იდენტიფიკატორების განსაზღვრისას:

- ა. არის დაცული იდენტიფიკატორების შექმნის წესები
- ბ. არ არის დაცული იდენტიფიკატორების შექმნის წესები
- გ. დაცულია იდენტიფიკატორების წაშლის წესები

4.1.20. შეზღუდული იდენტიფიკატორები უნდა მოვათავსოთ:

- ა. კუთხურ ფრჩხილებში (< >) ან ბრჭყალებში (“ ”)
- ბ. კვადრატულ ფრჩხილებში ([]) ან აპოსტროფებში (' ')
- გ. კვადრატულ ფრჩხილებში ([]) ან ბრჭყალებში (“ ”)

4.1.21. შეზღუდული იდენტიფიკატორებია:

- ა. [MY TABLE] და [ORDER]
- ბ. [MY TABLE] და [Personali]
- გ. [MYTABLE] და [ORDER]

4.1.22. იდენტიფიკატორის პირველი სიმბოლო შეიძლება იყოს:

- ა. ხაზგასმის სიმბოლო, @, \$ ან # სიმბოლო
- ბ. ათობითი ციფრები, @, \$, _, # სიმბოლოები
- გ. ნებისმიერი ლათინური სიმბოლო a-დან z-მდე, ეროვნული ანბანის სიმბოლო, ხაზგასმის სიმბოლო, @ ან # სიმბოლო

4.1.23. იდენტიფიკატორი, რომელიც # სიმბოლოთი იწყება, აღნიშნავს:

- ა. გლობალურ დროებით ობიექტს
- ბ. ლოკალურ დროებით ობიექტს
- გ. ცვლადს

4.1.24. იდენტიფიკატორი, რომელიც ## სიმბოლოებით იწყება, აღნიშნავს:

- ა. ლოკალურ დროებით ობიექტს
- ბ. გლობალურ დროებით ობიექტს
- გ. ცვლადს

4.1.25. იდენტიფიკატორი, რომელიც @ სიმბოლოთი იწყება, აღნიშნავს:

- ა. ლოკალურ დროებით ობიექტს
- ბ. გლობალურ დროებით ობიექტს
- გ. ცვლადს

4.1.26. იდენტიფიკატორის მომდევნო სიმბოლოები შეიძლება იყოს:

- ა. Unicode Standard 2.0 სტანდარტით განსაზღვრული სიმბოლოები, ათობითი ციფრები, @, \$, _, # სიმბოლოები
- ბ. Unicode Standard 2.0 სტანდარტით განსაზღვრული სიმბოლოები, ინდექსები
- გ. ათობითი ციფრები, @, \$, _, # სიმბოლოები, შენახული პროცედურები

4.1.27. იდენტიფიკატორის შიგნით:

- ა. დაუშვებელია სპეციალური სიმბოლოების გამოყენება: ~, !, %, ^, &, - , (,), {, }, ", ., \, ' და ინტერვალი
- ბ. დასაშვებია სპეციალური სიმბოლოების გამოყენება: ~, !, %, ^, &, - , (,), {, }, ", ., \, ' და ინტერვალი
- გ.

4.1.28. იდენტიფიკატორი:

- ა. უნდა იყოს დარეზერვებული სიტყვა და არ უნდა იყოს უნიკალური
- ბ. უნდა იყოს დარეზერვებული სიტყვა და უნდა იყოს უნიკალური
- გ. არ უნდა იყოს დარეზერვებული სიტყვა და უნდა იყოს უნიკალური

4.1.29. სახელების განსხვავება რეგისტრის მიხედვით:

- ა. ხდება
- ბ. არ ხდება
- გ. ხანდახან ხდება

4.1.30. ზოგიერთი შეზღუდვის გვერდის ასავლელად:

- ა. შეგვიძლია ობიექტების სახელები ბრჭყალებში ან კვადრატულ ფრჩხილებში მოვათავსოთ
- ბ. არ შეგვიძლია ობიექტების სახელები ბრჭყალებში ან კვადრატულ ფრჩხილებში მოვათავსოთ
- გ. შეგვიძლია ობიექტების სახელები ბრჭყალებში ან აპოსტროფებში მოვათავსოთ

4.1.31. ზოგიერთი შეზღუდვის გვერდის ასავლელად:

- ა. შეგვიძლია შეზღუდული იდენტიფიკატორები გამოვიყენოთ
- ბ. არ შეგვიძლია შეზღუდული იდენტიფიკატორები გამოვიყენოთ
- გ.

4.1.32. შეზღუდული იდენტიფიკატორები გამოყენების შემთხვევაში ობიექტის სახელში:

- ა. დასაშვებია ინტერვალები და სპეციალური სიმბოლოები, აგრეთვე, დარეზერვებული სიტყვები
- ბ. დაუშვებელია ინტერვალები და სპეციალური სიმბოლოები, აგრეთვე, დარეზერვებული სიტყვები
- გ. დასაშვებია მხოლოდ ინტერვალები

4.1.33. ობიექტის სრული სახელი იქმნება:

- ა. მონაცემთა ბაზის, სქემისა და ობიექტის სახელის საფუძველზე
- ბ. სქემისა და ობიექტის სახელის საფუძველზე
- გ. სერვერის, მონაცემთა ბაზის, სქემისა და ობიექტის სახელის საფუძველზე

4.1.34. ობიექტის სრულ სახელს შემდეგი სინტაქსი აქვს:

- ა. [[[სერვერის_სახელი.] [სქემის_სახელი].] ობიექტის_სახელი
- ბ. [სქემის_სახელი].] ობიექტის_სახელი
- გ. [[[სერვერის_სახელი.] [მონაცემთა_ბაზის_სახელი].] [სქემის_სახელი].] ობიექტის_სახელი

4.1.35. სერვერის ნებისმიერ ობიექტს:

- ა. არ უნდა ჰქონდეს უნიკალური, სრულად განსაზღვრული სახელი
- ბ. უნდა ჰქონდეს უნიკალური, სრულად განსაზღვრული სახელი
- გ. უნდა ჰქონდეს უნიკალური, სრულად განუსაზღვრელი სახელი

4.1.36. ერთ ბაზაში შეიძლება იყოს ერთნაირი სახელის მქონე ორი ცხრილი, რომლებიც:

- ა. ერთ სქემაში იქნება მოთავსებული
- ბ. სხვადასხვა სქემაში არ იქნება მოთავსებული
- გ. სხვადასხვა სქემაში იქნება მოთავსებული

4.1.37. სვეტების სახელები:

- ა. უნდა იყოს უნიკალური ცხრილის ან წარმოდგენის შიგნით
- ბ. არ უნდა იყოს უნიკალური ცხრილის ან წარმოდგენის შიგნით
- გ. უნდა იყოს უნიკალური ცხრილის ან წარმოდგენის გარეთ

4.1.38. ობიექტთან მიმართვისას არ არის მითითებული სერვერის სახელი, მაშინ:

- ა. არ აიღება მიმდინარე სერვერის სახელი
- ბ. აიღება მიმდინარე სერვერის სახელი
- გ. აიღება მიმდინარე მონაცემთა ბაზის სახელი

4.1.39. თუ მითითებული არ არის მონაცემთა ბაზის სახელი, მაშინ:

- ა. არ აიღება მიმდინარე მონაცემთა ბაზა
- ბ. აიღება მიმდინარე სერვერი
- გ. აიღება მიმდინარე მონაცემთა ბაზა

4.1.40. თუ მითითებული არ არის სქემის სახელი, მაშინ:

- ა. არ აიღება dbi სქემა
- ბ. აიღება dbi სქემა
- გ. აიღება TABLE სქემა

4.1.41. შემზღვევლების (“ “ და [] სიმბოლოები) გამოყენების წესები დამოკიდებულია QUOTED_IDENTIFIER პარამეტრის მნიშვნელობაზე:

- ა. დამოკიდებულია QUOTED_IDENTIFIER პარამეტრის მნიშვნელობაზე
- ბ. არ არის დამოკიდებული QUOTED_IDENTIFIER პარამეტრის მნიშვნელობაზე
- გ. დამოკიდებულია dbi სქემის მნიშვნელობაზე

4.1.42. თუ QUOTED_IDENTIFIER პარამეტრი იმყოფება ON მდგომარეობაში, მაშინ შემზღვევლების გამოყენება შემდეგნაირად ხდება:

ა. 1. კუთხური (< >) და კვადრატული ფრჩხილები ([]) გამოიყენება იდენტიფიკატორების შეზღვევისათვის. ბრჭყალები (" ") გამოიყენება სიმბოლური სტრიქონებისათვის.

2. სიმბოლური სტრიქონები არ უნდა მოთავსდეს აპოსტროფებში (' '). იდენტიფიკატორის შეზღვევისათვის აპოსტროფების გამოყენება არ შეიძლება. თუ სიმბოლური სტრიქონი აპოსტროფს შეიცავს, მაშინ ამ აპოსტროფის გვერდით მეორე აპოსტროფი უნდა მოვათავსოთ

ბ. 1. ბრჭყალები (" ") და კვადრატული ფრჩხილები ([]) არ გამოიყენება იდენტიფიკატორების შეზღვევისათვის. ბრჭყალები (" ") არ გამოიყენება სიმბოლური სტრიქონებისათვის.

2. სიმბოლური სტრიქონები არ უნდა მოთავსდეს აპოსტროფებში (' '). იდენტიფიკატორის შეზღვევისათვის აპოსტროფების გამოყენება არ შეიძლება. თუ სიმბოლური სტრიქონი აპოსტროფს შეიცავს, მაშინ ამ აპოსტროფის გვერდით მეორე აპოსტროფი უნდა მოვათავსოთ

გ. 1. ბრჭყალები (" ") და კვადრატული ფრჩხილები ([]) გამოიყენება იდენტიფიკატორების შეზღვევისათვის. ბრჭყალები (" ") არ გამოიყენება სიმბოლური სტრიქონებისათვის.

2. სიმბოლური სტრიქონები უნდა მოთავსდეს აპოსტროფებში (' '). იდენტიფიკატორის შეზღვევისათვის აპოსტროფების გამოყენება არ შეიძლება. თუ სიმბოლური სტრიქონი აპოსტროფს შეიცავს, მაშინ ამ აპოსტროფის გვერდით მეორე აპოსტროფი უნდა მოვათავსოთ

4.1.43. თუ QUOTED_IDENTIFIER პარამეტრი იმყოფება OFF მდგომარეობაში, მაშინ შემზღვევლების გამოყენება შემდეგნაირად ხდება:

ა. 1. იდენტიფიკატორების შეზღვევისათვის კუთხური ფრჩხილები (< >) გამოიყენება
2. სიმბოლური სტრიქონები უნდა მოთავსდეს ამპერსანდებში (& &) ან ბრჭყალებში (" "). თუ სტრიქონი აპოსტროფს შეიცავს, მაშინ ეს სტრიქონი შეგვიძლია ბრჭყალებში მოვათავსოთ

ბ. 1. იდენტიფიკატორების შეზღვევისათვის კვადრატული ფრჩხილები ([]) გამოიყენება
2. სიმბოლური სტრიქონები უნდა მოთავსდეს აპოსტროფებში (' ') ან ბრჭყალებში (" "). თუ სტრიქონი აპოსტროფს შეიცავს, მაშინ ეს სტრიქონი შეგვიძლია ბრჭყალებში მოვათავსოთ

გ. 1. იდენტიფიკატორების შეზღვევისათვის კვადრატული ფრჩხილები ([]) არ გამოიყენება
2. სიმბოლური სტრიქონები არ უნდა მოთავსდეს აპოსტროფებში (' ') ან ბრჭყალებში (" "). თუ სტრიქონი აპოსტროფს შეიცავს, მაშინ ეს სტრიქონი არ შეგვიძლია ბრჭყალებში მოვათავსოთ

4.1.44. რომელი მოთხოვნაა სწორი:

ა. SET QUOTED_IDENTIFIER ON
SELECT * FROM "MY TABLE" WHERE "Last Name" = N'O'Brien'

ბ. SET QUOTED_IDENTIFIER ON
SELECT * FROM [MY TABLE] WHERE [Last Name] = N'O'Brien'

გ. SET QUOTED_IDENTIFIER OFF
SELECT * FROM "MY TABLE" WHERE "Last Name" = N'O'Brien'

4.1.45. კვადრატული ფრჩხილები, როგორც შემზღვევლები:

ა. გამოიყენება ხანდახან

ბ. არ გამოიყენება ყოველთვის მიუხედავად QUOTED_IDENTIFIER პარამეტრის მნიშვნელობისა

გ. გამოიყენება ყოველთვის მიუხედავად QUOTED_IDENTIFIER პარამეტრის მნიშვნელობისა

4.1.46. ცვლადის გამოყენებამდე საჭიროა მისი გამოცხადება:

- ა. SET ბრძანების საშუალებით
- ბ. DECLARE ბრძანების საშუალებით
- გ. SELECT ბრძანების საშუალებით

4.1.47. ცვლადის სახელი:

- ა. არ უნდა იწყებოდეს @ სიმბოლოთი
- ბ. უნდა იწყებოდეს \$ სიმბოლოთი
- გ. უნდა იწყებოდეს @ სიმბოლოთი

4.1.48. @ სიმბოლო ლოკალური ცვლადების სახელების გარდა გამოიყენება, აგრეთვე,:

- ა. ინდექსების სახელების განსაზღვრისათვის
- ბ. ფუნქციებისა და შენახული პროცედურების არგუმენტების სახელების განსაზღვრისათვის
- გ. ტრიგერების სახელების განსაზღვრისათვის

4.1.49. ერთი DECLARE ბრძანებით:

- ა. არ შეიძლება რამდენიმე ცვლადის გამოცხადება, რომელთა სახელები ერთმანეთისაგან მძიმეებით იქნება გამოყოფილი
- ბ. შეიძლება რამდენიმე ცვლადის გამოცხადება, რომელთა სახელები ერთმანეთისაგან წერტილებით იქნება გამოყოფილი
- გ. შეიძლება რამდენიმე ცვლადის გამოცხადება, რომელთა სახელები ერთმანეთისაგან მძიმეებით იქნება გამოყოფილი

4.1.50. ცვლადებს მნიშვნელობები შეგვიძლია მივანიჭოთ:

- ა. SET და SELECT ბრძანებების გამოყენებით
- ბ. DECLARE და SELECT ბრძანებების გამოყენებით
- გ. SET და DECLARE ბრძანებების გამოყენებით

4.1.51. მინიჭების რომელი ოპერატორი შეიცავს შეცდომას:

- ა. SELECT @1_@ = 5.5
- ბ. SELECT #1_@ = 5.5
- გ. SELECT 1_@ = 5.5

4.1.52. ცვლადს:

- ა. არ შეგვიძლია მივანიჭოთ რომელიმე ფუნქციის მუშაობის შედეგი
- ბ. შეგვიძლია მივანიჭოთ რომელიმე ფუნქციის მუშაობის შედეგი
- გ. შეგვიძლია მივანიჭოთ რომელიმე ინდექსის მუშაობის შედეგი

4.1.53. რომელი მოთხოვნა შეიცავს შეცდომას:

- ა. DECLARE @tanxa float
SELECT @tanxa AS [მაქსიმალური გადასახდელი თანხა]
- ბ. DECLARE @tanxa float
SET @tanxa = (SELECT MAX(gadasaxdeli_1) FROM Xelshekruleba)
- გ. DECLARE @tanxa1 float
SET @tanxa2 = (SELECT MAX(gadasaxdeli_1) FROM Xelshekruleba)

SELECT @tanxa3 AS [მაქსიმალური გადასახდელი თანხა]

4.1.54. ცვლადების მნიშვნელობების გამოსატანად შეგვიძლია:

- ა. SET და SELECT ბრძანებების გამოყენება
- ბ. PRINT და SELECT ბრძანებების გამოყენება
- გ. PRINT და DECLARE ბრძანებების გამოყენება

4.1.55. უმარტივესი ოპერატორებია:

- ა. უნარული ოპერატორები
- ბ. ბინარული ოპერატორები
- გ. ტერნარული ოპერატორები

4.1.56. უნარული ოპერატორები, რომლებიც მხოლოდ:

- ა. ერთ ოპერანდთან მუშაობენ
- ბ. ორ ოპერანდთან მუშაობენ
- გ. სამ ოპერანდთან მუშაობენ

4.1.57. უმარტივესი ოპერატორები გამოიყენება:

- ა. მხოლოდ მთელი რიცხვების მიმართ
- ბ. როგორც მთელი, ისე წილადი რიცხვების მიმართ
- გ. მხოლოდ წილადი რიცხვების მიმართ

4.1.58. უმარტივესი ოპერატორებია:

- ა. „+“, „-“ და SET
- ბ. „~“, „+“ და DECLARE
- გ. „+“, „-“ და „~“

4.1.59. „+“ ოპერატორი გამოიყენება:

- ა. არადადებითი რიცხვის აღნიშვნისათვის
- ბ. უარყოფითი რიცხვის აღნიშვნისათვის
- გ. დადებითი რიცხვის აღნიშვნისათვის

4.1.60. „-“ ოპერატორი გამოიყენება:

- ა. უარყოფითი რიცხვის აღნიშვნისათვის
- ბ. დადებითი რიცხვის აღნიშვნისათვის
- გ. არაუარყოფითი რიცხვის აღნიშვნისათვის

4.1.61. „~“ ოპერატორი გამოიყენება:

- ა. რიცხვის დამატების მისაღებად
- ბ. უარყოფითი რიცხვის მისაღებად
- გ. დადებითი რიცხვის მისაღებად

4.1.62. მინიჭების ოპერატორი:

- ა. „=>“ სიმბოლოთი აღინიშნება და გამოიყენება ცვლადისათვის გარკვეული მნიშვნელობის მისანიჭებლად
- ბ. „=“ სიმბოლოთი აღინიშნება და გამოიყენება ცვლადისათვის გარკვეული მნიშვნელობის

მისანიჭებლად

გ. „<=“ სიმბოლოთი აღინიშნება და გამოიყენება ცვლადისათვის გარკვეული მნიშვნელობის მისანიჭებლად

4.1.63. შედარების ოპერატორი:

ა. „<=“ სიმბოლოთი აღინიშნება და გამოიყენება ცვლადისათვის გარკვეული მნიშვნელობის მისანიჭებლად

ბ. „>“ სიმბოლოთი აღინიშნება და გამოიყენება ცვლადისათვის გარკვეული მნიშვნელობის მისანიჭებლად

გ. „=“ სიმბოლოთი აღინიშნება და გამოიყენება ორი მნიშვნელობის შესადარებლად

4.1.64. ცვლადების გამოცხადების ოპერატორია:

ა. "="

ბ. SET

გ. DECLARE

4.1.65. არითმეტიკის ოპერატორებია:

ა. +, -, *, /, %

ბ. +, -, <, >=

გ. >, <=, *, /

4.1.66. ბინარული ოპერატორებია:

ა. ~, +, %

ბ. +, -, *, /, %

გ. -, *, /, ~

4.1.67. კონკატენაციის ოპერატორია:

ა. ~

ბ. %

გ. +

4.1.68. შედარების ოპერატორებია:

ა. <, !=, !<, =

ბ. >=, !>, +

გ. -, <=, *

4.1.69. თუ პირობა სრულდება, მაშინ შედარების ოპერატორის მუშაობის შედეგია:

ა. NULL

ბ. false

გ. true

4.1.70. თუ პირობა არ სრულდება, მაშინ შედარების ოპერატორის მუშაობის შედეგია:

ა. NULL

ბ. false

გ. true

- 4.1.71. თუ შედარება შეუძლებელია, მაგალითად, ტიპებია განსხვავებული, მაშინ:
- ა. გაიცემა NULL
 - ბ. გაიცემა true
 - გ. გაიცემა false
- 4.1.72. შედარების შედეგების დასამუშავებლად გამოიყენება:
- ა. NULL ოპერატორი
 - ბ. SET ოპერატორი
 - გ. IF ოპერატორი
- 4.1.73. ლოგიკის ოპერატორებია:
- ა. ALL, AND, ANY, BETWEEN, EXISTS, IN, LIKE, NOT, OR და SOME
 - ბ. ALL, AND, ANY, DECLARE
 - გ. BETWEEN, EXISTS, IN, SET
- 4.1.74. ბიტობრივი ოპერატორებია:
- ა. ბიტობრივი AND, ბიტობრივი OR და BETWEEN
 - ბ. ბიტობრივი AND, ბიტობრივი OR და ბიტობრივი XOR
 - გ. ბიტობრივი AND, ბიტობრივი OR და DECLARE
- 4.1.75. SELECT * FROM PersonalI WHERE personaliID = 5; მოთხოვნაში:
- ა. 5 არის სტანდარტული იდენტიფიკატორი
 - ბ. PersonalI არის შეზღუდული იდენტიფიკატორი
 - გ. PersonalI არის სტანდარტული იდენტიფიკატორი
- 4.1.76. SELECT * FROM [MY TABLE] მოთხოვნაში:
- ა. [MY TABLE] არის შეზღუდული იდენტიფიკატორი
 - ბ. [MY TABLE] არის სტანდარტული იდენტიფიკატორი
 - გ. TABLE არის სტანდარტული იდენტიფიკატორი
- 4.1.77. რომელია ცვლადის გამოცხადების ბრძანება:
- ა. SET
 - ბ. DECLARE
 - გ. SELECT
- 4.1.78. SET @Asaki = 7; ბრძანების შესრულების შედეგად:
- ა. @Asaki ცვლადს არ მიენიჭება მნიშვნელობა 7
 - ბ. შესრულდება @Asaki ცვლადის გამოცხადება
 - გ. @Asaki ცვლადს მიენიჭება მნიშვნელობა 7
- 4.1.79. SELECT @# = 150; ბრძანების შესრულების შედეგად:
- ა. @# ცვლადს მიენიჭება მნიშვნელობა 150
 - ბ. @# ცვლადს არ მიენიჭება მნიშვნელობა 150
 - გ. შესრულდება @# ცვლადის გამოცხადება
- 4.1.80. DECLARE @sashualo_asaki FLOAT; ბრძანების შესრულების შედეგად:

- ა. შესრულება @sashualo_asaki ცვლადის გამოცხადება როგორც სტრიქონის
- ბ. შესრულება @sashualo_asaki ცვლადის გამოცხადება როგორც წილადის
- გ. შესრულება @sashualo_asaki ცვლადის გამოცხადება როგორც მთელი რიცხვის

4.1.81. SELECT @sashualo_asaki = AVG(asaki) FROM Personal; მოთხოვნის შესრულების შედეგად:

- ა. @sashualo_asaki ცვლადს მიენიჭება asaki სვეტის მინიმალური მნიშვნელობა
- ბ. @sashualo_asaki ცვლადს არ მიენიჭება asaki სვეტის საშუალო მნიშვნელობა
- გ. @sashualo_asaki ცვლადს მიენიჭება asaki სვეტის საშუალო მნიშვნელობა

4.1.82. SET @tanxa =

```
(
SELECT MAX(gadasaxdeli_1)
FROM Xelshekruleba
);
```

ბრძანების შესრულების შედეგად:

- ა. @tanxa ცვლადს მიენიჭება gadasaxdeli_1 სვეტის მაქსიმალური მნიშვნელობა
- ბ. @tanxa ცვლადს არ მიენიჭება gadasaxdeli_1 სვეტის მაქსიმალური მნიშვნელობა
- გ. @tanxa ცვლადს მიენიჭება gadasaxdeli_1 სვეტის საშუალო მნიშვნელობა

4.1.83. SELECT -9 მოთხოვნას ეკრანზე გამოაქვს:

- ა. 9
- ბ. -9
- გ. 0

4.1.84. SELECT 5 + 6 * 8 მოთხოვნას ეკრანზე გამოაქვს:

- ა. 46
- ბ. 88
- გ. 53

ცხრილში მონაცემების ჩამატება, შეცვლა და წაშლა

4.2.1. ცხრილში მონაცემების ჩასამატებლად გამოიყენება:

- ა. INSERT ბრძანება
- ბ. UPDATE ბრძანება
- გ. DELETE ბრძანება

4.2.2. თუ INSERT ბრძანებაში სვეტების სია მითითებული არ არის, მაშინ VALUES არგუმენტში მითითებული მნიშვნელობების რაოდენობა და მიმდევრობა:

- ა. არ უნდა შეესაბამებოდეს ცხრილის სვეტების რაოდენობას და მიმდევრობას
- ბ. უნდა შეესაბამებოდეს ცხრილის სვეტების რაოდენობას და მიმდევრობას
- გ. უნდა შეესაბამებოდეს ცხრილის სვეტების მხოლოდ რაოდენობას

4.2.3. თუ გვინდა რამდენიმე სვეტის მნიშვნელობის შეტანა ჩვენთვის სასურველი მიმდევრობით, მაშინ მათი სახელები:

- ა. ასეთივე მიმდევრობით არ უნდა მივუთითოთ 'სვეტების_სია' არგუმენტში
- ბ. სხვა მიმდევრობით უნდა მივუთითოთ 'სვეტების_სია' არგუმენტში

გ. ასეთივე მიმდევრობით უნდა მივუთითოთ 'სვეტების_სია' არგუმენტში

4.2.4. თუ გვინდა სვეტების მნიშვნელობების შეტანა ჩვენთვის სასურველი მიმდევრობით, მაშინ მათი სახელები ასეთივე მიმდევრობით უნდა მივუთითოთ 'სვეტების_სია' არგუმენტში. შესაბამისად, VALUE არგუმენტში ვუთითებთ მონაცემებს:

- ა. სხვა მიმდევრობით
- ბ. შესაბამისი მიმდევრობით
- გ. უკუ მიმდევრობით

4.2.5. INSERT ბრძანებაში VALUES არგუმენტი განსაზღვრავს:

- ა. ცხრილში შესაცვლელ მონაცემებს
- ბ. ცხრილიდან წასაშლელ მონაცემებს
- გ. ცხრილში ჩასასმელ მონაცემებს

4.2.6. INSERT ბრძანებაში VALUES არგუმენტში მითითებული სიდიდეების რაოდენობა:

- ა. არ უნდა ემთხვეოდეს ცხრილში ან 'სვეტების_სია' არგუმენტში მითითებული სვეტების რაოდენობას
- ბ. უნდა ემთხვეოდეს ცხრილში ან 'სვეტების_სია' არგუმენტში მითითებული სვეტების რაოდენობას
- გ. უნდა ემთხვეოდეს მხოლოდ 'სვეტების_სია' არგუმენტში მითითებული სვეტების რაოდენობას

4.2.7. INSERT ბრძანების DEFAULT არგუმენტი მიუთითებს, რომ:

- ა. არ მოხდება ავტომატურად განსაზღვრული მნიშვნელობების ჩასმა
- ბ. მოხდება ავტომატურად განსაზღვრული მნიშვნელობების ჩასმა
- გ. მოხდება ხელით განსაზღვრული მნიშვნელობების ჩასმა

4.2.8. თუ სვეტისთვის ავტომატური მნიშვნელობა განსაზღვრული არ არის და ნებადართულია NULL მნიშვნელობის შენახვა, მაშინ INSERT ბრძანების შესრულების დროს სვეტში:

- ა. NULL მნიშვნელობა მოთავსთავდება
- ბ. IDENTITY მნიშვნელობა მოთავსთავდება
- გ. ნული მოთავსთავდება

4.2.9. DEFAULT არგუმენტის გამოყენება:

- ა. არ შეიძლება იმ სტრიქონებისათვის, რომლებისთვისაც განსაზღვრულია IDENTITY თვისება
- ბ. შეიძლება იმ სვეტებისათვის, რომლებისთვისაც განსაზღვრულია IDENTITY თვისება
- გ. არ შეიძლება იმ სვეტებისათვის, რომლებისთვისაც განსაზღვრულია IDENTITY თვისება

4.2.10. NULL არგუმენტი მიუთითებს, რომ INSERT ბრძანების შესრულების დროს სვეტში მოთავსდებოდა NULL მნიშვნელობა თუ ამ სვეტისათვის:

- ა. ნებადართულია ნულის შენახვა
- ბ. ნებადართული არ არის NULL მნიშვნელობის შენახვა
- გ. ნებადართულია NULL მნიშვნელობის შენახვა

4.2.11. 'გამოსახულება' არგუმენტი შეიძლება:

- ა. იყოს მხოლოდ მუდმივა
- ბ. არ იყოს მუდმივა, ცვლადი ან გამოსახულება და განსაზღვრავს სვეტში ჩასასმელ მნიშვნელობას
- გ. იყოს მუდმივა, ცვლადი ან გამოსახულება და განსაზღვრავს სვეტში ჩასასმელ მნიშვნელობას

4.2.12. სვეტში ჩასასმელ მნიშვნელობას:

- ა. უნდა ჰქონდეს ისეთივე ტიპი, როგორც არ აქვს შესაბამის სვეტს და არ უნდა აკმაყოფილებდეს შეზღუდვის პირობებს (თუ შეზღუდვა განსაზღვრულია)
- ბ. არ უნდა ჰქონდეს ისეთივე ტიპი, როგორც აქვს შესაბამის სვეტს და უნდა აკმაყოფილებდეს შეზღუდვის პირობებს (თუ შეზღუდვა განსაზღვრულია)
- გ. უნდა ჰქონდეს ისეთივე ტიპი, როგორც აქვს შესაბამის სვეტს და უნდა აკმაყოფილებდეს შეზღუდვის პირობებს (თუ შეზღუდვა განსაზღვრულია)

4.2.13. INSERT ბრძანების 'მიღებული_ცხრილი' არგუმენტი შეიძლება შეიცავდეს:

- ა. DELETE ბრძანებას
- ბ. UPDATE ბრძანებას
- გ. SELECT ბრძანებას, რომლის საშუალებითაც მოცემულ ცხრილში ჩასასმელ სტრიქონებს სხვა ცხრილიდან მივიღებთ

4.2.14. თუ გამოიყენება INSERT ბრძანების 'მიღებული_ცხრილი' არგუმენტი მაშინ:

- ა. ორივე ცხრილს სხვადასხვა სვეტები უნდა ჰქონდეს
- ბ. ორივე ცხრილს ერთნაირი სვეტები არ უნდა ჰქონდეს
- გ. ორივე ცხრილს ერთნაირი სვეტები უნდა ჰქონდეს

4.2.15. INSERT ბრძანების DEFAULT VALUES არგუმენტის მითითების შემთხვევაში:

- ა. თითოეულ სვეტში მოთავსდება ავტომატურად განუსაზღვრელი მნიშვნელობა
- ბ. თითოეულ სვეტში არ მოთავსდება ავტომატურად განსაზღვრული მნიშვნელობა
- გ. თითოეულ სვეტში მოთავსდება ავტომატურად განსაზღვრული მნიშვნელობა

4.2.16. INSERT ბრძანების DEFAULT VALUES არგუმენტის მითითების შემთხვევაში, თუ სვეტისათვის ასეთი მნიშვნელობა განსაზღვრული არ არის და ნებადართულია NULL მნიშვნელობის შენახვა, მაშინ:

- ა. სვეტში მოთავსდება ნული
- ბ. სვეტში არ მოთავსდება NULL მნიშვნელობა
- გ. სვეტში მოთავსდება NULL მნიშვნელობა

4.2.17. INSERT ბრძანების DEFAULT VALUES არგუმენტის მითითების შემთხვევაში, თუ ავტომატური მნიშვნელობა განსაზღვრული არ არის და არაა ნებადართული NULL მნიშვნელობის შენახვა, მაშინ:

- ა. გაიცემა ნული
- ბ. არ გაიცემა შეტყობინება შეცდომის შესახებ
- გ. გაიცემა შეტყობინება შეცდომის შესახებ

4.2.18. ერთი INSERT ბრძანებით:

- ა. შეიძლება ცხრილში რამდენიმე სტრიქონის ჩამატება

- ბ. არ შეიძლება ცხრილში რამდენიმე სტრიქონის ჩამატება
- გ. შეიძლება ცხრილში მხოლოდ ერთი სტრიქონის ჩამატება

4.2.19. SELECT ... INTO ბრძანების შესრულების შედეგად:

- ა. შეიქმნება ახალი ცხრილი, რომელშიც SELECT მოთხოვნის მიერ ამორჩეული სტრიქონები მოთავსდება
- ბ. არ შეიქმნება ახალი ცხრილი, რომელშიც SELECT მოთხოვნის მიერ ამორჩეული სტრიქონები მოთავსდება
- გ. შეიქმნება ახალი ცხრილი, რომელშიც არ მოთავსდება SELECT მოთხოვნის მიერ ამორჩეული სტრიქონები

4.2.20. SELECT...INTO ბრძანებაში 'სვეტის_სახელი' არგუმენტი განსაზღვრავს იმ სვეტის სახელს, რომელიც:

- ა. ჩართული იქნება ინდექსში
- ბ. ჩართული არ იქნება შედეგში
- გ. ჩართული იქნება შედეგში

4.2.21. თუ სხვადასხვა ცხრილის სვეტებს ერთნაირი სახელები აქვს, მაშინ მათი სახელების:

- ა. უკან უნდა მივუთითოთ ფსევდონიმი
- ბ. წინ უნდა მივუთითოთ ფსევდონიმი
- გ. შუაში უნდა მივუთითოთ ფსევდონიმი

4.2.22. ფსევდონიმის გამოყენება სასარგებლოა იმ შემთხვევაშიც, როცა გვინდა, რომ შესაქმნელი ცხრილის სვეტებს საწყისი ცხრილისაგან განსხვავებული სახელები:

- ა. ჰქონდეს
- ბ. არ ჰქონდეს
- გ. ხანდახან ჰქონდეს

4.2.23. SELECT...INTO ბრძანებაში INTO 'ახალი_ცხრილს_სახელი' არგუმენტი:

- ა. არ მიუთითებს შესაქმნელი ცხრილის სახელს
- ბ. მიუთითებს შესაქმნელი ცხრილის სახელს
- გ. მიუთითებს შესაქმნელი ცხრილის ინდექსს

4.2.24. SELECT...INTO ბრძანებაში INTO 'ახალი_ცხრილს_სახელი' არგუმენტის გამოყენებით თუ დროებით ცხრილს ვქმნით, მაშინ ცხრილის სახელის:

- ა. წინ უნდა მივუთითოთ * ან **
- ბ. უკან უნდა მივუთითოთ # ან ##
- გ. წინ უნდა მივუთითოთ # ან ##

4.2.25. SELECT...INTO ბრძანებაში FROM არგუმენტი:

- ა. შეიცავს საწყისი ცხრილების სახელებს
- ბ. არ შეიცავს საწყისი ცხრილების სახელებს
- გ. შეიცავს საბოლოო ცხრილების სახელებს

4.2.26. SELECT gvari AS [გვარი], ganyofileba AS [განყოფილება], staji AS [სტაჟი]
INTO Pers_1 FROM Personali მოთხოვნის შესრულების შედეგად შეიქმნება:

- ა. Personali ცხრილი
- ბ. Pers_1 ცხრილი
- გ. gvari, ganyofileba და staji სვეტები

4.2.27. ცხრილებში მონაცემების შესაცვლელად გამოიყენება:

- ა. UPDATE ბრძანება
- ბ. DELETE ბრძანება
- გ. INSERT ბრძანება

4.2.28. UPDATE ბრძანებაში 'ცხრილის_სახელი' არგუმენტი იმ ცხრილის სახელია, რომელშიც:

- ა. არ უნდა შესრულდეს მონაცემების ცვლილება
- ბ. უნდა შესრულდეს მონაცემების ცვლილება
- გ. უნდა შესრულდეს ინდექსების ცვლილება

4.2.29. UPDATE ბრძანებაში 'წარმოდგენის_სახელი' იმ წარმოდგენის სახელია, რომელშიც:

- ა. არ უნდა შესრულდეს მონაცემების ცვლილება
- ბ. უნდა შესრულდეს მონაცემების ცვლილება
- გ. უნდა შესრულდეს ტრიგერის ცვლილება

4.2.30. UPDATE ბრძანება მონაცემებს წარმოდგენის:

- ა. მხოლოდ ერთ ცხრილში ცვლის
- ბ. მხოლოდ ორ ცხრილში ცვლის
- გ. მხოლოდ სამ ცხრილში ცვლის

4.2.31. თუ, წარმოდგენა ორი ცხრილისაგან შედგება, მაშინ UPDATE ბრძანება:

- ა. ერთხელ უნდა შევასრულოთ თითოეულ ცხრილში მონაცემების შესაცვლელად
- ბ. სამჯერ უნდა შევასრულოთ თითოეულ ცხრილში მონაცემების შესაცვლელად
- გ. ორჯერ უნდა შევასრულოთ თითოეულ ცხრილში მონაცემების შესაცვლელად

4.2.32. UPDATE ბრძანების SET საკვანძო სიტყვა იწყებს ბლოკს, რომელშიც მითითებულია შესაცვლელი:

- ა. ტრიგერების სია
- ბ. სვეტების ან ცვლადების სია
- გ. ინდექსების სია

4.2.33. UPDATE ბრძანების 'სვეტის_სახელი = { გამოსახულება | DEFAULT | NULL }' არგუმენტში DEFAULT საკვანძო სიტყვა მიუთითებს, რომ სვეტს:

- ა. უნდა მიენიჭოს NULL მნიშვნელობა
- ბ. არ უნდა მიენიჭოს ავტომატურად განსაზღვრული მნიშვნელობა
- გ. უნდა მიენიჭოს ავტომატურად განსაზღვრული მნიშვნელობა

4.2.34. UPDATE ბრძანების 'სვეტის_სახელი = { გამოსახულება | DEFAULT | NULL }' არგუმენტში NULL საკვანძო სიტყვა მიუთითებს, რომ სვეტს:

- ა. უნდა მიენიჭოს NULL მნიშვნელობა
- ბ. არ უნდა მიენიჭოს NULL მნიშვნელობა
- გ. უნდა მიენიჭოს DEFAULT მნიშვნელობა

4.2.35. UPDATE ბრძანების 'სვეტის_სახელი = { გამოსახულება | DEFAULT | NULL }' არგუმენტში 'გამოსახულება' შეიძლება:

- ა. იყოს მუდმივა, ცვლადი ან გამოსახულება
- ბ. არ იყოს მუდმივა, ცვლადი ან გამოსახულება
- გ. იყოს შენახული პროცედურა

4.2.36. თუ UPDATE ბრძანების 'გამოსახულება' არგუმენტში გამოიყენება ის სვეტები, რომლებსაც ცვლილება შეეხო, მაშინ გამოყენებული:

- ა. არ იქნება მათი ძველი მნიშვნელობები
- ბ. იქნება მათი ძველი მნიშვნელობები
- გ. იქნება მათი ახალი მნიშვნელობები

4.2.37. ფილტრის მითითების შემთხვევაში UPDATE ბრძანებით განხორციელებული ცვლილებები შეეხება:

- ა. მთელ ცხრილს
- ბ. მხოლოდ ერთ სტრიქონს
- გ. მხოლოდ ფილტრში ამორჩეულ სტრიქონებს

4.2.38. თუ UPDATE ბრძანებაში ფილტრი არ არის მითითებული, მაშინ:

- ა. ცვლილება შეეხება ცხრილის ერთ სტრიქონის მითითებულ სვეტებს
- ბ. ცვლილება არ შეეხება ცხრილის ყველა სტრიქონის მითითებულ სვეტებს
- გ. ცვლილება შეეხება ცხრილის ყველა სტრიქონის მითითებულ სვეტებს

4.2.39. ცხრილიდან სტრიქონების წასაშლელად გამოიყენება:

- ა. DELETE ბრძანება
- ბ. UPDATE ბრძანება
- გ. INSERT ბრძანება

4.2.40. თუ DELETE ბრძანებაში WHERE განყოფილება არ არის მითითებული, მაშინ ცხრილიდან:

- ა. ერთი სტრიქონი წაიშლება
- ბ. ყველა სტრიქონი წაიშლება
- გ. არც ერთი სტრიქონი არ წაიშლება

4.2.41.

INSERT INTO Personal

VALUES (N'სამხარაძე', N'ბექა', N'სასოფლო', N'თბილისი', NULL, N'ვაკე', 300.55, 27, 5, '07.09.1980', N'კაცი', N'პეტრიაშვილის ქ.1', '221-11-00', '899-012345', N'beqa@geo.net.ge', 9);

მოთხოვნის შესრულების შედეგად:

- ა. Personal ცხრილში ჩაემატება ორი სტრიქონი
- ბ. Personal ცხრილში არ ჩაემატება ერთი სტრიქონი
- გ. Personal ცხრილში ჩაემატება ერთი სტრიქონი

4.2.42. INSERT INTO Personal (saxeli, gvari, qalaqi, raioni, misamarti_saxlis, staji, tarigi_dabadebis, asaki, mobiluri, sqesi, teleponi_saxlis, email, ganyofileba, ierarqia)

VALUES (N'გიორგი', N'სამხარაძე', N'თბილისი', N'ჩუღურეთი', N'ც. დადიანის 2', 3, '14.01.1982',

25, N'877-102332', N'კაცი', N'261-11-12', N'giorgi@geo.net.ge', N'სავაჭრო', 9);

მოთხოვნის შესრულების შედეგად:

- ა. მონაცემების შეტანა განხორციელდება ჩვენთვის საჭირო მიმდევრობით
- ბ. მონაცემების შეტანა არ განხორციელდება ჩვენთვის საჭირო მიმდევრობით
- გ. Personali ცხრილში ჩაემატება ორი სტრიქონი

4.2.43. INSERT INTO Personali VALUES (N'ზვიადი', N'გიორგამე', N'სასპორტო', N'ქობულეთი', N'აჭარა', NULL, NULL, NULL, 10, '02.03.1985', NULL, NULL, NULL, NULL, NULL, NULL); მოთხოვნის შესრულების შედეგად:

- ა. ზოგიერთი სვეტის მნიშვნელობა არ იქნება NULL
- ბ. ზოგიერთი სვეტის მნიშვნელობა იქნება NULL
- გ. ყველა სვეტის მნიშვნელობა იქნება NULL

4.2.44.

INSERT INTO Personali_1

SELECT *

FROM Personali

მოთხოვნის შესრულების შედეგად:

- ა. Personali_1 ცხრილს დაემატება ზოგიერთი სტრიქონი Personali ცხრილიდან
- ბ. Personali_1 ცხრილს არ დაემატება ყველა სტრიქონი Personali ცხრილიდან
- გ. Personali_1 ცხრილს დაემატება ყველა სტრიქონი Personali ცხრილიდან

4.2.45. INSERT INTO Personali DEFAULT VALUES; მოთხოვნის შესრულების შედეგად:

- ა. Personali ცხრილს დაემატება სტრიქონი, რომლის ყველა სვეტში ნაგულისხმევი მნიშვნელობა მოთავსდებოდა
- ბ. Personali ცხრილს დაემატება სტრიქონი, რომლის ყველა სვეტში ნაგულისხმევი მნიშვნელობა არ მოთავსდებოდა
- გ. Personali ცხრილს არ დაემატება სტრიქონი, რომლის ყველა სვეტში ნაგულისხმევი მნიშვნელობა მოთავსდებოდა

4.2.46. INSERT INTO Personali VALUES (N'სამხარაძე', N'ბექა', DEFAULT, N'თბილისი', DEFAULT, DEFAULT, DEFAULT, 27, 5, '07.09.1980', N'კაცი', N'პეტრიაშვილის ქ.1', '221-11-00', '899-012345', DEFAULT, DEFAULT); მოთხოვნის შესრულების შედეგად:

- ა. Personali ცხრილს დაემატება სტრიქონი, რომლის ყველა სვეტში ნაგულისხმევი მნიშვნელობა მოთავსდებოდა
- ბ. Personali ცხრილს დაემატება სტრიქონი, რომლის ზოგიერთ სვეტში ნაგულისხმევი მნიშვნელობა მოთავსდებოდა
- გ. Personali ცხრილს არ დაემატება სტრიქონი, რომლის ზოგიერთ სვეტში ნაგულისხმევი მნიშვნელობა მოთავსდებოდა

4.2.47. INSERT INTO Personali VALUES (N'სამხარაძე', N'რომანი', N'სასოფლო', N'თბილისი', NULL, N'ვაკე', 100.11 * 5, 27, 5 * 3, '07.09.1980', N'კაცი', N'პეტრიაშვილის ქ.1', '221-11-00', '899-012345', N'beqa@geo.net.ge', 9); მოთხოვნის შესრულების შედეგად:

- ა. Personali ცხრილს დაემატება სტრიქონი, რომლის xelfasi და staji სვეტების მნიშვნელობები ფორმულის მიხედვით არ გამოითვლება
- ბ. Personali ცხრილს არ დაემატება სტრიქონი, რომლის xelfasi და staji სვეტების

მნიშვნელობები ფორმულის მიხედვით გამოითვლება

გ. Personalი ცხრილს დაემატება სტრიქონი, რომლის xelfasi და staji სვეტების მნიშვნელობები ფორმულის მიხედვით გამოითვლება

4.2.48.

INSERT INTO Personalი VALUES

(N'სამხარაძე', N'საბა', N'სასოფლო', N'თბილისი', NULL, N'ვაკე', 110.11 * 5, 27, 5 * 3, '07.09.1985', N'კაცი', N'პეტრიაშვილის ქ.1', '221-11-00', '599-012345', N'saba@geo.net.ge', 39),
(N'სამხარაძე', N'ანა', N'სავაჭრო', N'ქუთაისი', N'იმერეთი', NULL, 120.11 * 6, 28, 6 * 3, '07.09.1990', N'ქალი', N'პეტრიაშვილის ქ.1', '221-11-00', '574-012345', N'ana@geo.net.ge', 19); მოთხოვნის შესრულების შედეგად:

- ა. Personalი ცხრილს ორი სტრიქონი დაემატება
- ბ. Personalი ცხრილს სამი სტრიქონი დაემატება
- გ. Personalი ცხრილს ერთი სტრიქონი დაემატება

4.2.49.

SELECT gvari AS [გვარი], ganyofileba AS [განყოფილება], staji AS [სტაჟი]
INTO Pers_1

FROM Personalი;

მოთხოვნის შესრულების შედეგად:

- ა. არ შეიქმნება Pers_1 ცხრილი და მასში Personalი ცხრილიდან ჩაიწერება gvari, ganyofileba და staji სვეტების მნიშვნელობები
- ბ. შეიქმნება Pers_1 ცხრილი და მასში Personalი ცხრილიდან ჩაიწერება gvari, ganyofileba და staji სვეტების მნიშვნელობები
- გ. შეიქმნება Pers_1 ცხრილი და მასში Personalი ცხრილიდან არ ჩაიწერება gvari, ganyofileba და staji სვეტების მნიშვნელობები

4.2.50. UPDATE Xelshekruleba SET @raodenoba = @raodenoba + 1; მოთხოვნის შესრულების შედეგად @raodenoba ცვლადის მნიშვნელობა:

- ა. არ გაიზრდება
- ბ. 1-ით შემცირდება
- გ. 1-ით გაიზრდება

4.2.51. UPDATE Personalი SET xelfasi = xelfasi * 1.1; მოთხოვნის შესრულების შედეგად xelfasi სვეტის მნიშვნელობა:

- ა. 10%-ით გაიზრდება
- ბ. 10%-ით შემცირდება
- გ. 10-ჯერ შემცირდება

4.2.52. თითოეული თანამშრომელის ხელფასის 10%-ით გასაზრდელად უნდა შევასრულოთ მოთხოვნა:

- ა. UPDATE Personalი SET xelfasi = xelfasi * 1.2;
- ბ. UPDATE Personalი SET xelfasi = xelfasi * 1.1;
- გ. UPDATE Personalი SET xelfasi = xelfasi * 1.0;

4.2.53. UPDATE Xelshekruleba SET vali_1 = gadasaxdeli_1 - gadaxdili_1, vali_d = gadasaxdeli_d -

gadaxdili_d; მოთხოვნის შესრულების შედეგად:

- ა. vali_1 სვეტის მნიშვნელობა შეიცვლება
- ბ. vali_1 და vali_d სვეტების მნიშვნელობები არ შეიცვლება
- გ. vali_1 და vali_d სვეტების მნიშვნელობები შეიცვლება

4.2.54. DELETE FROM #Shemkveti; მოთხოვნის შესრულების შედეგად:

- ა. #Shemkveti ცხრილის ყველა სტრიქონი წაიშლება
- ბ. #Shemkveti ცხრილის ყველა სტრიქონი არ წაიშლება
- გ. #Shemkveti ცხრილის ზოგიერთი სტრიქონი წაიშლება

4.2.55. DELETE FROM #Shemkveti WHERE iuridiuli_fizikuri = N'იურიდიული'; მოთხოვნის შესრულების შედეგად:

- ა. #Shemkveti ცხრილის ყველა სტრიქონი წაიშლება
- ბ. #Shemkveti ცხრილიდან წაიშლება იურიდიული პირები
- გ. #Shemkveti ცხრილიდან არ წაიშლება იურიდიული პირები

4.2.56. #Shemkveti ცხრილიდან ყველა სტრიქონის წასაშლელად უნდა შევასრულოთ მოთხოვნა:

- ა. DELETE FROM #Shemkveti WHERE iuridiuli_fizikuri = N'იურიდიული';
- ბ. DELETE FROM #Shemkveti;
- გ. DELETE FROM #Shemkveti WHERE iuridiuli_fizikuri = N'ფიზიკური';

4.2.57. Shemkveti ცხრილიდან იურიდიული პირების წასაშლელად უნდა შევასრულოთ მოთხოვნა:

- ა. DELETE FROM #Shemkveti;
- ბ. DELETE FROM #Shemkveti WHERE iuridiuli_fizikuri = N'ფიზიკური';
- გ. DELETE FROM #Shemkveti WHERE iuridiuli_fizikuri = N'იურიდიული';

ცხრილიდან მონაცემების ამორჩევა. SELECT განყოფილება.

4.3.1. SELECT ბრძანების SELECT განყოფილებაში:

- ა. ეთითება გამოსატანი სვეტების სია
- ბ. არ ეთითება გამოსატანი სვეტების სია
- გ. ეთითება გამოსატანი ცხრილების სია

4.3.2. SELECT ბრძანების SELECT განყოფილებაში ALL არგუმენტი მიუთითებს, რომ:

- ა. მოთხოვნის შესრულების შედეგში დაუშვებელია გამეორებადი (ერთნაირი) სტრიქონების გამოტანა
- ბ. მოთხოვნის შესრულების შედეგში დასაშვებია გამეორებადი (ერთნაირი) სტრიქონების გამოტანა
- გ. მოთხოვნის შესრულების შედეგში დასაშვებია გამეორებადი (ერთნაირი) სვეტების გამოტანა

4.3.3. SELECT ბრძანების SELECT განყოფილებაში ALL არგუმენტი:

- ა. არასოდეს არ იგულისხმება
- ბ. ავტომატურად არ იგულისხმება
- გ. ავტომატურად იგულისხმება

- 4.3.4. SELECT ბრძანების SELECT განყოფილებაში DISTINCT არგუმენტი:
- ა. კრძალავს ერთნაირი სტრიქონების გამოტანას
 - ბ. არ კრძალავს ერთნაირი სტრიქონების გამოტანას
 - გ. კრძალავს ერთნაირი სვეტების გამოტანას
- 4.3.5. SELECT ბრძანების SELECT განყოფილებაში TOP n არგუმენტის გამოყენების შედეგად:
- ა. არ გაიცემა პირველი n სტრიქონი
 - ბ. გაიცემა პირველი n სტრიქონი
 - გ. გაიცემა პირველი n+10 სტრიქონი
- 4.3.6. SELECT ბრძანების SELECT განყოფილების TOP n PERCENT არგუმენტში:
- ა. n არ არის პროცენტი
 - ბ. n არის პროცენტი
 - გ. n არის მთელი რიცხვი
- 4.3.7. SELECT განყოფილების <სია> კონსტრუქციაში * მიუთითებს, რომ გამოტანილი:
- ა. უნდა იყოს ცხრილის ან წარმოდგენის ყველა სვეტის მნიშვნელობა
 - ბ. არ უნდა იყოს ცხრილის ან წარმოდგენის ყველა სვეტის მნიშვნელობა
 - გ. უნდა იყოს ცხრილის ან წარმოდგენის ყველა სტრიქონის მნიშვნელობა
- 4.3.8. SELECT განყოფილების <სია> კონსტრუქციაში 'ცხრილის_სახელი' იმ შემთხვევაში ეთითება, როცა მოთხოვნაში:
- ა. რამდენიმე ინდექსია მითითებული
 - ბ. არც ერთი ცხრილია მითითებული
 - გ. რამდენიმე ცხრილია მითითებული
- 4.3.9. SELECT განყოფილების <სია> კონსტრუქციაში 'წარმოდგენის_სახელი' იმ შემთხვევაში ეთითება, როცა მოთხოვნაში:
- ა. რამდენიმე ინდექსია მითითებული
 - ბ. არც ერთი წარმოდგენაა მითითებული
 - გ. რამდენიმე წარმოდგენაა მითითებული
- 4.3.10. SELECT განყოფილების <სია> კონსტრუქციაში 'ცხრილის_ფსევდონიმი' შეგვიძლია გამოვიყენოთ:
- ა. ტრიგერის სახელის ნაცვლად
 - ბ. მონაცემთა ბაზის სახელის ნაცვლად
 - გ. ცხრილის სახელის ნაცვლად
- 4.3.11. SELECT განყოფილების <სია> კონსტრუქციაში 'სვეტის_სახელი' არგუმენტი განსაზღვრავს:
- ა. იმ ტრიგერის სახელს, რომლის მნიშვნელობაც უნდა გამოვიტანოთ
 - ბ. იმ ცხრილის სახელს, რომლის მნიშვნელობაც უნდა გამოვიტანოთ
 - გ. იმ სვეტის სახელს, რომლის მნიშვნელობაც უნდა გამოვიტანოთ
- 4.3.12. SELECT განყოფილების <სია> კონსტრუქციაში 'სვეტის_სახელი' არგუმენტი განსაზღვრავს

სვეტს, რომელიც:

- ა. უნდა ეკუთვნოდეს FROM განყოფილებაში მითითებული ცხრილებიდან ერთ-ერთს
- ბ. არ უნდა ეკუთვნოდეს FROM განყოფილებაში მითითებული ცხრილებიდან ერთ-ერთს
- გ. არ უნდა ეკუთვნოდეს FROM განყოფილებაში მითითებული ცხრილებიდან არც ერთს

4.3.13. SELECT განყოფილების <სია> კონსტრუქციის 'სვეტის_სახელი' არგუმენტში ფსევდონიმების გამოყენება:

- ა. აუცილებელია იმ შემთხვევაში, როცა გამოსატანი ცხრილების სვეტებს ერთნაირი სახელები არ აქვთ
- ბ. აუცილებელია იმ შემთხვევაში, როცა გამოსატანი ცხრილების სვეტებს ერთნაირი სახელები აქვთ
- გ. არაა აუცილებელი იმ შემთხვევაში, როცა გამოსატანი ცხრილების სვეტებს ერთნაირი სახელები აქვთ

4.3.14. თუ 'სვეტის_ფსევდონიმი' შეიცავს დაუშვებელ სიმბოლოებს, ინტერვალებს ან ეროვნულ სიმბოლოებს, მაშინ ის:

- ა. მრგვალ ფრჩხილებში უნდა მოვათავსოთ
- ბ. კვადრატულ ფრჩხილებში უნდა მოვათავსოთ
- გ. კუთხურ ფრჩხილებში უნდა მოვათავსოთ

4.3.15. SELECT განყოფილების <სია> კონსტრუქციაში 'გამოსახულება' არგუმენტი განსაზღვრავს:

- ა. გამოსახულებას, რომელიც გამოითვლება და მიღებული მნიშვნელობა არ მოთავსდება შედეგში
- ბ. გამოსახულებას, რომელიც არ გამოითვლება და მიღებული მნიშვნელობა არ მოთავსდება შედეგში
- გ. გამოსახულებას, რომელიც გამოითვლება და მიღებული მნიშვნელობა მოთავსდება შედეგში

4.3.16. SELECT განყოფილების <სია> კონსტრუქციაში 'გამოსახულება' არგუმენტი შეიძლება შეიცავდეს:

- ა. მუდმივებს, ცვლადებს, სვეტების სახელებს, ფუნქციებს ან მათ კომბინაციას
- ბ. მხოლოდ მთელ რიცხვებს
- გ. მხოლოდ წილადებს

4.3.17. SELECT განყოფილების <სია> კონსტრუქციაში IDENTITYCOL არგუმენტის გამოყენება:

- ა. არ იწვევს იმ სვეტის გამოტანას, რომელსაც IDENTITY თვისება დაყენებული აქვს YES მდგომარეობაში
- ბ. იწვევს იმ სვეტის გამოტანას, რომელსაც IDENTITY თვისება დაყენებული აქვს YES მდგომარეობაში
- გ. იწვევს იმ სტრიქონის გამოტანას, რომელსაც IDENTITY თვისება დაყენებული აქვს YES მდგომარეობაში

4.3.18. თუ SELECT მოთხოვნაში რამდენიმე ცხრილია მითითებული, რომლებსაც IDENTITYCOL სვეტები აქვთ, მაშინ აშკარად:

- ა. უნდა მივუთითოთ იმ ცხრილის სახელი, რომელსაც ეს სტრიქონი ეკუთვნის
- ბ. არ უნდა მივუთითოთ იმ ცხრილის სახელი, რომელსაც ეს სვეტი ეკუთვნის

გ. უნდა მივუთითოთ იმ ცხრილის სახელი, რომელსაც ეს სვეტი ეკუთვნის

4.3.19. SELECT ბრძანების დამუშავება ხდება შემდეგი მიმდევრობით:

- ა. FROM, WHERE, GROUP BY, HAVING, SELECT, ORDER BY
- ბ. WHERE, GROUP BY, HAVING, SELECT, ORDER BY, FROM
- გ. FROM, WHERE, HAVING, GROUP BY, SELECT, ORDER BY

4.3.20. SELECT ALL personaliID FROM Xelshekruleba; მოთხოვნის შესრულების შედეგად:

- ა. გამოჩნდება Xelshekruleba ცხრილის ყველა სვეტის მნიშვნელობა
- ბ. გამოჩნდება Xelshekruleba ცხრილის personaliID სვეტის გამეორებადი მნიშვნელობები
- გ. არ გამოჩნდება Xelshekruleba ცხრილის personaliID სვეტის გამეორებადი მნიშვნელობები

4.3.21. SELECT DISTINCT personaliID FROM Xelshekruleba; მოთხოვნის შესრულების შედეგად:

- ა. გამოჩნდება Xelshekruleba ცხრილის ყველა სვეტის მნიშვნელობა
- ბ. გამოჩნდება Xelshekruleba ცხრილის personaliID სვეტის გამეორებადი მნიშვნელობები
- გ. გამოჩნდება Xelshekruleba ცხრილის personaliID სვეტის არაგამეორებადი მნიშვნელობები

4.3.22. SELECT TOP 5 * FROM Personali; მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა Personali ცხრილის პირველი 5 სტრიქონი
- ბ. არ გაიცემა Personali ცხრილის პირველი 5 სტრიქონი
- გ. გაიცემა Personali ცხრილის სტრიქონების 5%

4.3.23. SELECT TOP 40 PERCENT * FROM Personali; მოთხოვნის შესრულების შედეგად:

- ა. არ გაიცემა Personali ცხრილის სტრიქონების 40%
- ბ. გაიცემა Personali ცხრილის სტრიქონების 40%
- გ. გაიცემა Personali ცხრილის სტრიქონების 60%

4.3.24. SELECT * FROM Personali; მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა Personali ცხრილის ყველა სვეტი
- ბ. არ გაიცემა Personali ცხრილის ყველა სვეტი და ყველა სტრიქონი
- გ. გაიცემა Personali ცხრილის ყველა სვეტი და ყველა სტრიქონი

4.3.25.

```
SELECT gvari AS [თანამშრომლის გვარი],  
       saxeli AS [თანამშრომლის სახელი],  
       qalaqi, asaki AS [თანამშრომლის ასაკი],  
       tarigi_dabadebis AS [დაბადების თარიღი]  
FROM Personali;
```

მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა gvari, saxeli, qalaqi, asaki და tarigi_dabadebis სვეტების მნიშვნელობები
- ბ. არ გაიცემა gvari, saxeli, qalaqi, asaki და tarigi_dabadebis სვეტების მნიშვნელობები
- გ. გაიცემა gvari, asaki და tarigi_dabadebis სვეტების მნიშვნელობები

4.3.26. SELECT qalaqi, asaki AS [თანამშრომლის ასაკი] FROM Personali; მოთხოვნაში:

- ა. [თანამშრომლის ასაკი] არის asaki სვეტის ფსევდონიმი
- ბ. [თანამშრომლის ასაკი] არის qalaqi სვეტის ფსევდონიმი

გ. [თანამშრომლის ასაკი] არის Personali ცხრილის ფსევდონიმი

4.3.27. SELECT qalaqi, asaki AS [თანამშრომლის ასაკი] FROM Personali; მოთხოვნაში:

- ა. asaki სვეტს აქვს ფსევდონიმი
- ბ. qalaqi სვეტს აქვს ფსევდონიმი
- გ. asaki სვეტს არ აქვს ფსევდონიმი

4.3.28. SELECT gvari, saxeli, YEAR(tarigi_dabadebis) AS [დაბადების წელი] FROM Personali; მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა თანამშრომლების დაბადების თვეები
- ბ. გაიცემა თანამშრომლების დაბადების წლები
- გ. არ გაიცემა თანამშრომლების დაბადების წლები

4.3.29. თანამშრომლების დაბადების წლების გასაგებად უნდა შევასრულოთ მოთხოვნა:

- ა. SELECT gvari AS [დაბადების წელი] FROM Personali;
- ბ. SELECT gvari, saxeli FROM Personali;
- გ. SELECT gvari, saxeli, YEAR(tarigi_dabadebis) AS [დაბადების წელი] FROM Personali;

4.3.30.

```
SELECT personaliID, YEAR(tarigi_dawyebis) AS xelshyear, COUNT(*) AS numxelsh
FROM Xelshekruleba
WHERE shemkvetiID = 6
GROUP BY personaliID, YEAR(tarigi_dawyebis)
ORDER BY personaliID, xelshyear;
```

მოთხოვნაში განყოფილებები შემდეგი თანმიმდევრობით დამუშავდება:

- ა. FROM Xelshekruleba, WHERE shemkvetiID = 6, GROUP BY personaliID, YEAR(tarigi_dawyebis), ORDER BY personaliID, xelshyear
- ბ. ORDER BY personaliID, xelshyear, FROM Xelshekruleba, WHERE shemkvetiID = 6, GROUP BY personaliID, YEAR(tarigi_dawyebis)
- გ. ORDER BY personaliID, xelshyear, FROM Xelshekruleba, GROUP BY personaliID, YEAR(tarigi_dawyebis), WHERE shemkvetiID = 6

4.3.31.

```
SELECT personaliID, YEAR(tarigi_dawyebis) AS xelshyear, COUNT(*) AS numxelsh
FROM Xelshekruleba
GROUP BY personaliID, YEAR(tarigi_dawyebis)
ORDER BY personaliID, xelshyear;
```

მოთხოვნაში განყოფილებები შემდეგი თანმიმდევრობით დამუშავდება:

- ა. FROM Xelshekruleba, GROUP BY personaliID, YEAR(tarigi_dawyebis), ORDER BY personaliID, xelshyear
- ბ. FROM Xelshekruleba, ORDER BY personaliID, xelshyear, GROUP BY personaliID
- გ. ORDER BY personaliID, xelshyear, GROUP BY personaliID, FROM Xelshekruleba

WHERE განყოფილება. ლოგიკის ოპერატორები.

4.4.1. WHERE განყოფილება გამოიყენება ცხრილის:

- ა. სტრიქონების გაფილტვრისათვის
- ბ. სვეტების გაფილტვრისათვის
- გ. სტრიქონების გაერთიანებისათვის

4.4.2. WHERE განყოფილებაში <ძებნის_პირობა> არის:

- ა. ფიზიკური გამოსახულება, რომელიც გასცემს true ან false მნიშვნელობას
- ბ. ნებისმიერი ლოგიკური გამოსახულება, რომელიც გასცემს true ან false მნიშვნელობას
- გ. ნებისმიერი ლოგიკური გამოსახულება, რომელიც გასცემს 5-ს ან 100-ს

4.4.3. თუ მოთხოვნაში WHERE პირობის ნაცვლად გამოვიყენებთ { INNER | { { LEFT | RIGHT | FULL } [OUTER] } JOIN } საკვანძო სიტყვებს, მაშინ ეს მოთხოვნა:

- ა. საშუალო სისწრაფით შესრულდება
- ბ. უფრო ნელა შესრულდება
- გ. უფრო სწრაფად შესრულდება

4.4.4. თუ მოთხოვნაში { INNER | { { LEFT | RIGHT | FULL } [OUTER] } JOIN } საკვანძო სიტყვების ნაცვლად გამოვიყენებთ WHERE პირობას, მაშინ ეს მოთხოვნა:

- ა. უფრო ნელა შესრულდება
- ბ. საშუალო სისწრაფით შესრულდება
- გ. უფრო სწრაფად შესრულდება

4.4.5. თუ დასამუშავებელია ათასობით სტრიქონი, მაშინ უმჯობესია:

- ა. WHERE პირობის გამოყენება
- ბ. { INNER | { { LEFT | RIGHT | FULL } [OUTER] } JOIN } საკვანძო სიტყვების გამოყენება
- გ. FROM განყოფილების გამოყენება

4.4.6. თუ დასამუშავებელია მცირე რაოდენობის სტრიქონი, მაშინ უმჯობესია:

- ა. WHERE პირობის გამოყენება
- ბ. { INNER | { { LEFT | RIGHT | FULL } [OUTER] } JOIN } საკვანძო სიტყვების გამოყენება
- გ. FROM განყოფილების გამოყენება

4.4.7. WHERE განყოფილება გვიბრუნებს მხოლოდ იმ სტრიქონებს, რომლებისთვისაც გამოსათვლელი ლოგიკური გამოსახულების მნიშვნელობა:

- ა. FALSE-ის ტოლია
- ბ. TRUE-ის ტოლია
- გ. UNKNOWN-ის ტოლია

4.4.8. T-SQL ენაში გამოიყენება სამობითი პრედიკატების ლოგიკა, რომელშიც ლოგიკურმა გამოსახულებებმა შეიძლება მიიღოს:

- ა. TRUE, FALSE ან UNKNOWN მნიშვნელობა
- ბ. მხოლოდ TRUE მნიშვნელობა
- გ. მხოლოდ FALSE მნიშვნელობა

4.4.9. სამობით ლოგიკაში მტკიცება „აბრუნებს TRUE-ს“:

- ა. არის იგივე, რაც მტკიცება „არ აბრუნებს UNKNOWN -ს“
- ბ. არის იგივე, რაც მტკიცება „არ აბრუნებს FALSE-ს“

გ. არ არის იგივე, რაც მტკიცება „არ აბრუნებს FALSE-ს“

4.4.10. WHERE სტადიაზე ამოიჩვენა სტრიქონები, რომლებისთვისაც გამოსათვლელი ლოგიკური მნიშვნელობა:

- ა. TRUE-ის ტოლია
- ბ. FALSE-ის ტოლია
- გ. UNKNOWN -ის ტოლია

4.4.11. BETWEEN ოპერატორი საშუალებას:

- ა. გვადლევს მივუთითოთ მნიშვნელობების დასაწყისი
- ბ. არ გვადლევს მივუთითოთ მნიშვნელობების დიაპაზონი
- გ. გვადლევს მივუთითოთ მნიშვნელობების დიაპაზონი

4.4.12. BETWEEN ოპერატორში 'გამოსახულება' არის შესამოწმებელი გამოსახულება, რომელიც:

- ა. არ შეიძლება შეიცავდეს როგორც სვეტის, ისე ცვლადის სახელს
- ბ. შეიძლება შეიცავდეს როგორც სვეტის, ისე ცვლადის სახელს
- გ. შეიძლება შეიცავდეს როგორც სვეტის, ისე პროცედურის სახელს

4.4.13. BETWEEN ოპერატორში 'საწყისი_გამოსახულება' განსაზღვრავს:

- ა. დიაპაზონის დასაწყისს
- ბ. დიაპაზონის დასასრულს
- გ. არაფერს

4.4.14. BETWEEN ოპერატორში 'საბოლოო_გამოსახულება' განსაზღვრავს:

- ა. დიაპაზონის დასასრულს
- ბ. არაფერს
- გ. დიაპაზონის დასაწყისს

4.4.15. IN ოპერატორის საშუალებით:

- ა. შეგვიძლია შევამოწმოთ ემთხვევა თუ არა 'გამოსახულების' მნიშვნელობა 'ქვემოთხოვნის' მიერ დაბრუნებული მნიშვნელობებიდან ან ჩამოთვლილი სიდიდეებიდან ერთ-ერთს
- ბ. არ შეგვიძლია შევამოწმოთ ემთხვევა თუ არა 'გამოსახულების' მნიშვნელობა 'ქვემოთხოვნის' მიერ დაბრუნებული მნიშვნელობებიდან ან ჩამოთვლილი სიდიდეებიდან ერთ-ერთს
- გ. არაფრის გაკეთება არ შეგვიძლია

4.4.16. SELECT * FROM Personal WHERE YEAR(tarigi_dabadebis) = 1966; მოთხოვნის შესრულების შედეგად:

- ა. არ გაიცემა ინფორმაცია 1966 წელს დაბადებული თანამშრომლების შესახებ
- ბ. გაიცემა ინფორმაცია 1966 წელს დაბადებული თანამშრომლების შესახებ
- გ. გაიცემა ინფორმაცია 1968 წელს დაბადებული მამაკაცების შესახებ

4.4.17. 1966 წელს დაბადებული თანამშრომლების შესახებ ინფორმაციის მისაღებად უნდა შევასრულოთ მოთხოვნა:

- ა. SELECT * FROM Personal WHERE YEAR(tarigi_dabadebis) = 1986;

- ბ. SELECT * FROM PersonalI WHERE YEAR(tarigi_dabadebis) = 1968;
- გ. SELECT * FROM PersonalI WHERE YEAR(tarigi_dabadebis) = 1966;

4.4.18. SELECT * FROM PersonalI WHERE staji > 10; მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა ინფორმაცია 10 წელზე მეტი სტაჟის მქონე თანამშრომლების შესახებ
- ბ. არ გაიცემა ინფორმაცია 10 წელზე მეტი სტაჟის მქონე თანამშრომლების შესახებ
- გ. გაიცემა ინფორმაცია 10 წელზე ნაკლები სტაჟის მქონე თანამშრომლების შესახებ

4.4.19. 10 წელზე მეტი სტაჟის მქონე თანამშრომლების შესახებ ინფორმაციის მისაღებად უნდა შევასრულოთ მოთხოვნა:

- ა. SELECT * FROM PersonalI WHERE staji < 10;
- ბ. SELECT * FROM PersonalI WHERE staji > 10;
- გ. SELECT * FROM PersonalI WHERE staji = 10;

4.4.20. SELECT * FROM PersonalI WHERE qalaqi = N'თბილისი' OR qalaqi = N'ბათუმი'; მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა ინფორმაცია მხოლოდ თბილისელი თანამშრომლების შესახებ
- ბ. არ გაიცემა ინფორმაცია თბილისელი და ბათუმელი თანამშრომლების შესახებ
- გ. გაიცემა ინფორმაცია თბილისელი და ბათუმელი თანამშრომლების შესახებ

4.4.21. თბილისელი და ბათუმელი თანამშრომლების შესახებ ინფორმაციის მისაღებად უნდა შევასრულოთ მოთხოვნა:

- ა. SELECT * FROM PersonalI WHERE qalaqi = N'თბილისი' OR qalaqi = N'ბათუმი';
- ბ. SELECT * FROM PersonalI WHERE qalaqi = N'თბილისი' AND qalaqi = N'ბათუმი';
- გ. SELECT * FROM PersonalI WHERE qalaqi = N'თბილისი';

4.4.22. SELECT * FROM PersonalI WHERE qalaqi = N'თბილისი' AND ganyofileba = N'სამედიცინო'; მოთხოვნის შესრულების შედეგად

- ა. არ გაიცემა ინფორმაცია თბილისელი თანამშრომლების შესახებ, რომლებიც სამედიცინო განყოფილებაში მუშაობენ
- ბ. გაიცემა ინფორმაცია თბილისელი თანამშრომლების შესახებ, რომლებიც სამედიცინო განყოფილებაში მუშაობენ
- გ. გაიცემა ინფორმაცია თბილისელი თანამშრომლების შესახებ, რომლებიც სამედიცინო განყოფილებაში არ მუშაობენ

4.4.23. ინფორმაციის მისაღებად თბილისელი თანამშრომლების შესახებ, რომლებიც სამედიცინო განყოფილებაში მუშაობენ უნდა შევასრულოთ მოთხოვნა:

- ა. SELECT * FROM PersonalI WHERE ganyofileba = N'სამედიცინო';
- ბ. SELECT * FROM PersonalI WHERE qalaqi = N'თბილისი';
- გ. SELECT * FROM PersonalI WHERE qalaqi = N'თბილისი' AND ganyofileba = N'სამედიცინო';

4.4.24. SELECT * FROM PersonalI WHERE NOT qalaqi = N'თბილისი'; მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა ინფორმაცია იმ თანამშრომლების შესახებ, რომლებიც თბილისში არ ცხოვრობენ
- ბ. გაიცემა ინფორმაცია იმ თანამშრომლების შესახებ, რომლებიც თბილისში ცხოვრობენ
- გ. არ გაიცემა ინფორმაცია იმ თანამშრომლების შესახებ, რომლებიც თბილისში არ

ცხოვრობენ

4.4.25. ინფორმაციის მისაღებად იმ თანამშრომლების შესახებ, რომლებიც თბილისში არ ცხოვრობენ უნდა შევასრულოთ მოთხოვნა:

- ა. SELECT * FROM Personal WHERE qalaqi = N'თბილისი';
- ბ. SELECT * FROM Personal WHERE NOT qalaqi = N'თბილისი';
- გ. SELECT * FROM Personal WHERE NOT ganyofileba = N'თბილისი';

4.4.26. SELECT * FROM Xelshekruleba WHERE tarigi_dawyebis BETWEEN '1.03.2005' AND '31.03.2005'; მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა 2015 წლის მარტის თვეში დადებული ხელშეკრულებები
- ბ. არ გაიცემა 2005 წლის მარტის თვეში დადებული ხელშეკრულებები
- გ. გაიცემა 2005 წლის მარტის თვეში დადებული ხელშეკრულებები

4.4.27. 2005 წლის მარტის თვეში დადებული ხელშეკრულებების შესახებ ინფორმაციის მისაღებად უნდა შევასრულოთ მოთხოვნა:

- ა. SELECT * FROM Xelshekruleba WHERE tarigi_dawyebis BETWEEN '1.03.2005' AND '31.03.2005';
- ბ. SELECT * FROM Xelshekruleba WHERE tarigi_dawyebis BETWEEN '1.03.2015' AND '31.03.2015';
- გ. SELECT * FROM Xelshekruleba WHERE tarigi_dawyebis BETWEEN '1.04.2005' AND '31.04.2005';

4.4.28. SELECT * FROM Personal WHERE qalaqi IN (N'თბილისი', N'ქუთაისი'); მოთხოვნის შესრულების შედეგად:

- ა. არ გაიცემა ინფორმაცია თბილისელი ან ქუთაისელი თანამშრომლების შესახებ
- ბ. გაიცემა ინფორმაცია თბილისელი ან ქუთაისელი თანამშრომლების შესახებ
- გ. გაიცემა ინფორმაცია მხოლოდ ქუთაისელი თანამშრომლების შესახებ

4.4.29. ინფორმაციის მისაღებად თბილისელი ან ქუთაისელი თანამშრომლების შესახებ უნდა შევასრულოთ მოთხოვნა:

- ა. SELECT * FROM Personal WHERE qalaqi = N'თბილისი' AND qalaqi = N'ქუთაისი';
- ბ. SELECT * FROM Personal WHERE qalaqi NOT IN (N'თბილისი', N'ქუთაისი');
- გ. SELECT * FROM Personal WHERE qalaqi IN (N'თბილისი', N'ქუთაისი');

4.4.30. SELECT gvari, saxeli, qalaqi FROM Personal WHERE qalaqi IN

(
SELECT qalaqi FROM Shemkveti WHERE iuridiuli_fizikuri = N'იურიდიული'
);

მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა ინფორმაცია ფირმის იმ თანამშრომლების შესახებ, რომლებიც ერთ ქალაქში იმყოფებიან იმ შემკვეთებთან ერთად, რომლებიც იურიდიულ პირებს წარმოადგენენ
- ბ. არ გაიცემა ინფორმაცია ფირმის იმ თანამშრომლების შესახებ, რომლებიც ერთ ქალაქში იმყოფებიან იმ შემკვეთებთან ერთად, რომლებიც იურიდიულ პირებს წარმოადგენენ
- გ. გაიცემა ინფორმაცია ფირმის იმ თანამშრომლების შესახებ, რომლებიც ერთ ქალაქში იმყოფებიან იმ შემკვეთებთან ერთად, რომლებიც იურიდიულ პირებს არ წარმოადგენენ

4.4.31. ინფორმაციის მისაღებად ფორმის იმ თანამშრომლების შესახებ, რომლებიც ერთ ქალაქში იმყოფებიან იმ შემკვეთებთან ერთად, რომლებიც იურიდიულ პირებს წარმოადგენენ უნდა შევასრულოთ მოთხოვნა:

- ა. SELECT gvari, saxeli, qalaqi FROM Personal WHERE qalaqi IN
(
SELECT qalaqi FROM Shemkveti WHERE iuridiuli_fizikuri = N'ფიზიკური'
);
- ბ. SELECT gvari, saxeli, qalaqi FROM Personal WHERE qalaqi IN
(
SELECT qalaqi FROM Shemkveti WHERE iuridiuli_fizikuri = N'იურიდიული'
);
- გ. SELECT gvari, saxeli, qalaqi FROM Personal WHERE qalaqi IN
(
SELECT qalaqi FROM Shemkveti WHERE iuridiuli_fizikuri = N'იურიდიული' AND iuridiuli_fizikuri = N'ფიზიკური'
);

LIKE ოპერატორი

4.5.1. LIKE ოპერატორის საშუალებით:

- ა. შეუძლებელია გამოსახულების შედარება მოცემულ შაბლონთან
- ბ. შესაძლებელია გამოსახულების შედარება მოცემულ შაბლონთან
- გ. გამოსახულება გადაიწერება შაბლონში

4.5.2. LIKE ოპერატორში 'გამოსახულება' არგუმენტი:

- ა. არ განსაზღვრავს შესამოწმებელ გამოსახულებას
- ბ. განსაზღვრავს შესამოწმებელ გამოსახულებას
- გ. განსაზღვრავს წილად რიცხვებს

4.5.3. LIKE ოპერატორში 'შაბლონი':

- ა. შეიცავს სიმბოლო_შემცვლელებს
- ბ. არ შეიცავს სიმბოლო_შემცვლელებს
- გ. შეიცავს ფუნქციის სახელებს

4.5.4. LIKE ოპერატორის შაბლონში გამოიყენება დარეზერვებული სიმბოლოები:

- ა. %, _, @ და ^
- ბ. \$, [,] და ^
- გ. %, _, [,], # და ^

4.5.5. LIKE ოპერატორში 'escape_სიმბოლო' არგუმენტით მოიცემა მმართველი სიმბოლო. ამ სიმბოლოს შემდეგ მოთავსებული სიმბოლო-შემცვლელი:

- ა. მოქმედებს და ჩაითვლება ჩვეულებრივ სიმბოლოდ
- ბ. აღარ მოქმედებს და ჩაითვლება ჩვეულებრივ სიმბოლოდ
- გ. აღარ მოქმედებს და არ ჩაითვლება ჩვეულებრივ სიმბოლოდ

4.5.6. LIKE ოპერატორში % სიმბოლო:

- ა. ცვლის სიმბოლოების ნებისმიერ მიმდევრობას
- ბ. არ ცვლის სიმბოლოების ნებისმიერ მიმდევრობას
- გ. ცვლის '%' სიმბოლოს

4.5.7. LIKE ოპერატორში _ (ხაზგასმის სიმბოლო) სიმბოლო:

- ა. ცვლის ნებისმიერ ერთ სიმბოლოს
- ბ. ცვლის სიმბოლოების ნებისმიერ მიმდევრობას
- გ. არ ცვლის ნებისმიერ ერთ სიმბოლოს

4.5.8. SELECT * FROM Shemkveti WHERE firmis_dasaxeleva LIKE N'გ%'; მოთხოვნის შესრულების შედეგად:

- ა. არ გაიცემა ინფორმაცია იმ ფირმების შესახებ, რომელთა სახელები 'ჯ' ასოთი იწყება
- ბ. გაიცემა ინფორმაცია იმ ფირმების შესახებ, რომელთა სახელები 'ჯ' ასოთი იწყება
- გ. გაიცემა ინფორმაცია იმ ფირმების შესახებ, რომელთა სახელები 'ჯ' ასოთი არ იწყება

4.5.9. ფორმაციის მისარებად იმ ფირმების შესახებ, რომელთა სახელები 'ჯ' ასოთი იწყება, უნდა შევასრულოთ მოთხოვნა:

- ა. SELECT * FROM Shemkveti WHERE qalaqi LIKE N'გ%';
- ბ. SELECT * FROM Shemkveti WHERE firmis_dasaxeleva NOT LIKE N'გ%';
- გ. SELECT * FROM Shemkveti WHERE firmis_dasaxeleva LIKE N'გ%';

4.5.10. SELECT gvari, qalaqi FROM Personal WHERE gvari LIKE N'_ა%'; მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა ინფორმაცია იმ თანამშრომლების შესახებ, რომელთა გვარის მეორე სიმბოლოა 'ა'
- ბ. არ გაიცემა ინფორმაცია იმ თანამშრომლების შესახებ, რომელთა გვარის მეორე სიმბოლოა 'ა'
- გ. გაიცემა ინფორმაცია იმ თანამშრომლების შესახებ, რომელთა გვარის მესამე სიმბოლოა 'ა'

4.5.11. ინფორმაციის მისაღებად იმ თანამშრომლების შესახებ, რომელთა გვარის მეორე სიმბოლოა 'ა' უნდა შევასრულოთ მოთხოვნა:

- ა. SELECT gvari, qalaqi FROM Personal WHERE gvari LIKE N'__ა%';
- ბ. SELECT gvari, qalaqi FROM Personal WHERE gvari LIKE N'_ა%';
- გ. SELECT gvari, qalaqi FROM Personal WHERE gvari NOT LIKE N'_ა%';

4.5.12. SELECT * FROM Shemkveti WHERE qalaqi LIKE N'[თბ]>'; მოთხოვნის შესრულების შედეგად:

- ა. არ გაიცემა ის შემკვეთები, რომლებიც იმ ქალაქებში ცხოვრობენ, რომელთა სახელების პირველი ასოებია 'თ' ან 'ბ'
- ბ. გაიცემა ის შემკვეთები, რომლებიც იმ ქალაქებში ცხოვრობენ, რომელთა სახელების პირველი ასოები არ არის 'თ' ან 'ბ'
- გ. გაიცემა ის შემკვეთები, რომლებიც იმ ქალაქებში ცხოვრობენ, რომელთა სახელების პირველი ასოებია 'თ' ან 'ბ'

4.5.13. ინფორმაციის მისაღებად იმ შემკვეთების, რომლებიც იმ ქალაქებში ცხოვრობენ, რომელთა სახელების პირველი ასოებია 'თ' ან 'ბ' უნდა შევასრულოთ მოთხოვნა:

- ა. SELECT * FROM Shemkveti WHERE qalaqi LIKE N'[თბ]%';
- ბ. SELECT * FROM Shemkveti WHERE qalaqi NOT LIKE N'[თბ]%';
- გ. SELECT * FROM Shemkveti WHERE qalaqi LIKE N'[ფრ]%';

4.5.14. SELECT qalaqi, firmis_dasaxeleba, gvari FROM Shemkveti WHERE qalaqi LIKE N'^თბ]%';
 მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა ინფორმაცია იმ ფირმების, რომლებიც იმ ქალაქებში მდებარეობენ, რომელთა დასახელება იწყება 'თ' და 'ბ' ასოებით
- ბ. გაიცემა ინფორმაცია იმ ფირმების, რომლებიც იმ ქალაქებში მდებარეობენ, რომელთა დასახელება არ იწყება 'თ' და 'ბ' ასოებით
- გ. გაიცემა ინფორმაცია იმ ფირმების, რომლებიც იმ ქალაქებში მდებარეობენ, რომელთა დასახელება არ იწყება 'დ' და 'ს' ასოებით

4.5.15. ინფორმაციის მისაღებად იმ ფირმების შესახებ, რომლებიც იმ ქალაქებში მდებარეობენ, რომელთა დასახელება არ იწყება 'თ' და 'ბ' ასოებით, უნდა შევასრულოთ მოთხოვნა:

- ა. SELECT qalaqi, firmis_dasaxeleba, gvari FROM Shemkveti WHERE qalaqi LIKE N'^ტკ]%';
- ბ. SELECT qalaqi, firmis_dasaxeleba, gvari FROM Shemkveti WHERE qalaqi LIKE N'[თბ]%';
- გ. SELECT qalaqi, firmis_dasaxeleba, gvari FROM Shemkveti WHERE qalaqi LIKE N'^თბ]%';

4.5.16. SELECT qalaqi, gvari, saxeli, firmis_dasaxeleba FROM Shemkveti WHERE qalaqi LIKE N'[მ-ჯ]%';
 მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა ის შემკვეთები, რომლებიც ცხოვრობენ იმ ქალაქებში, რომელთა სახელების პირველი სიმბოლო მოთავსებულია 'მ' და 'ჯ' ასოებს შორის
- ბ. გაიცემა ის შემკვეთები, რომლებიც ცხოვრობენ იმ ქალაქებში, რომელთა სახელების პირველი სიმბოლო მოთავსებულია 'ნ' და 'ჟ' ასოებს შორის
- გ. არ გაიცემა ის შემკვეთები, რომლებიც ცხოვრობენ იმ ქალაქებში, რომელთა სახელების პირველი სიმბოლო მოთავსებულია 'მ' და 'ჯ' ასოებს შორის

4.5.17. გაიცემა ის შემკვეთები, რომლებიც ცხოვრობენ იმ ქალაქებში, რომელთა სახელების პირველი სიმბოლო მოთავსებულია 'მ' და 'ჯ' ასოებს შორის

- ა. SELECT qalaqi, gvari, saxeli, firmis_dasaxeleba FROM Shemkveti WHERE qalaqi LIKE N'[ნ-ჟ]%';
- ბ. SELECT qalaqi, gvari, saxeli, firmis_dasaxeleba FROM Shemkveti WHERE qalaqi LIKE N'[მ-ჯ]%';
- გ. SELECT qalaqi, gvari, saxeli, firmis_dasaxeleba FROM Shemkveti WHERE ganyofileba LIKE N'[მ-ჯ]%';

4.5.18. SELECT gvari, iuridiuli_fizikuri, mobiluri_direqtoris FROM Shemkveti WHERE mobiluri_direqtoris LIKE '555%';

მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა ის შემკვეთები, რომელთა მობილური ტელეფონის ნომრები 555-ით არ იწყება
- ბ. არ გაიცემა ის შემკვეთები, რომელთა მობილური ტელეფონის ნომრები 555-ით იწყება
- გ. გაიცემა ის შემკვეთები, რომელთა მობილური ტელეფონის ნომრები 555-ით იწყება

4.5.19. ინფორმაციის მისაღებად იმ შემკვეთების შესახებ, რომელთა მობილური ტელეფონის ნომრები 555-ით იწყება უნდა შევასრულოთ მოთხოვნა:

- ა. SELECT gvari, iuridiuli_fizikuri, mobiluri_direqtoris FROM Shemkveti WHERE mobiluri_direqtoris LIKE '577%';

ბ. SELECT gvari, iuridiuli_fizikuri, mobiluri_direqtoris FROM Shemkveti WHERE mobiluri_direqtoris NOT LIKE '555%';

გ. SELECT gvari, iuridiuli_fizikuri, mobiluri_direqtoris FROM Shemkveti WHERE mobiluri_direqtoris LIKE '555%';

აგრეგირების ფუნქციები

4.6.1. აგრეგირების ფუნქციები მონაცემების:

- ა. სტატისტიკურ დამუშავებას ასრულებენ
- ბ. ფიზიკურ დამუშავებას ასრულებენ
- გ. სტატისტიკურ დამუშავებას არ ასრულებენ

4.6.2. AVG ფუნქცია მითითებული სვეტისთვის გამოთვლის:

- ა. მნიშვნელობების ჯამს
- ბ. მინიმალურ მნიშვნელობას
- გ. საშუალო არითმეტიკულს

4.6.3. თუ მონაცემები დაჯგუფებულია, მაშინ AVG ფუნქცია:

- ა. თითოეული ჯგუფისთვის გამოითვლის მაქსიმუმს
- ბ. თითოეული ჯგუფისთვის არ გამოითვლის საშუალო არითმეტიკულს
- გ. თითოეული ჯგუფისთვის გამოითვლის საშუალო არითმეტიკულს

4.6.4. აგრეგირების ფუნქციების უმრავლესობა:

- ა. ითვალისწინებს NULL მნიშვნელობის შემცველ სვეტებს
- ბ. თითოეული ჯგუფისთვის გამოითვლის მაქსიმუმს
- გ. არ ითვალისწინებს NULL მნიშვნელობის შემცველ სვეტებს

4.6.5. AVG ფუნქციაში ALL არგუმენტი მიუთითებს, რომ აგრეგირება:

- ა. არ ითვალისწინებს NULL მნიშვნელობის შემცველ სვეტებს
- ბ. არ უნდა შეეხოს გამეორებად (ერთნაირ) სტრიქონებს
- გ. უნდა შეეხოს გამეორებად (ერთნაირ) სტრიქონებს

4.6.6. AVG ფუნქციაში ALL არგუმენტი:

- ა. ავტომატურად იგულისხმება
- ბ. ავტომატურად არ იგულისხმება
- გ. ხანდახან იგულისხმება

4.6.7. AVG ფუნქციაში DISTINCT საკვანძო სიტყვა მიუთითებს, რომ აგრეგირება შეეხება:

- ა. მხოლოდ ერთნაირ სტრიქონებს
- ბ. მხოლოდ უნიკალურ სტრიქონებს
- გ. მხოლოდ არაუნიკალურ სტრიქონებს

4.6.8. COUNT ფუნქცია:

- ა. თვლის მნიშვნელობების რაოდენობას მოცემულ სვეტში ან სტრიქონების საერთო რაოდენობას ცხრილში
- ბ. გამოთვლის სვეტის მნიშვნელობების საშუალო არითმეტიკულს

- გ. გამოთვლის სვეტის მნიშვნელობების ჯამს
- 4.6.9. COUNT ფუნქციაში * მიუთითებს, რომ გაიცემა ცხრილში მოთავსებული სტრიქონების:
- ა. საშუალო მნიშვნელობა
 - ბ. ჯამი
 - გ. რაოდენობა
- 4.6.10. SUM ფუნქცია გამოთვლის მითითებული სვეტის მნიშვნელობების:
- ა. საშუალო მნიშვნელობას
 - ბ. ჯამს
 - გ. რაოდენობას
- 4.6.11. MAX ფუნქცია გასცემს მითითებული სვეტის მნიშვნელობებიდან:
- ა. საშუალო მნიშვნელობას
 - ბ. მაქსიმალურს
 - გ. მინიმალურს
- 4.6.12. MIN ფუნქცია გასცემს მითითებული სვეტის მნიშვნელობებიდან მინიმალურს:
- ა. მინიმალურს
 - ბ. ჯამს
 - გ. მაქსიმალურს
- 4.6.13. SELECT AVG(xelfasi) AS [საშუალო ხელფასი] FROM Personali; მოთხოვნის შესრულების შედეგად:
- ა. არ გაიცემა თანამშრომლების საშუალო ხელფასი
 - ბ. გაიცემა თანამშრომლების საშუალო ხელფასი
 - გ. გაიცემა თანამშრომლების მაქსიმალური ხელფასი
- 4.6.14. თანამშრომლების საშუალო ხელფასის მისაღებად უნდა შევასრულოთ მოთხოვნა:
- ა. SELECT AVG(qalaqi) AS [საშუალო ხელფასი] FROM Personali;
 - ბ. SELECT AVG(xelfasi) AS [საშუალო ხელფასი] FROM Personali;
 - გ. SELECT MIN(xelfasi) AS [საშუალო ხელფასი] FROM Personali;
- 4.6.15. SELECT COUNT(DISTINCT personaliID) AS [რაოდენობა] FROM Xelshekruleba; მოთხოვნის შესრულების შედეგად:
- ა. გაიცემა personaliID სვეტის გამეორებადი მნიშვნელობების რაოდენობა
 - ბ. არ გაიცემა personaliID სვეტის არაგამეორებადი მნიშვნელობების რაოდენობა
 - გ. გაიცემა personaliID სვეტის არაგამეორებადი მნიშვნელობების რაოდენობა
- 4.6.16. SELECT COUNT(*) AS [თანამშრომლების რაოდენობა] FROM Personali; მოთხოვნის შესრულების შედეგად:
- ა. გაიცემა Personali ცხრილის სტრიქონების რაოდენობა
 - ბ. არ გაიცემა Personali ცხრილის სტრიქონების რაოდენობა
 - გ. გაიცემა Personali ცხრილის სტრიქონების რაოდენობის ნახევარი
- 4.6.17. SELECT ganyofileba, AVG(asaki) AS [საშუალო ასაკი], COUNT(*) AS [განყოფილება]

FROM Personal GROUP BY ALL ganyofileba; მოთხოვნის შესრულების შედეგად:

- ა. არ გაიცემა თითოეულ განყოფილებაში მომუშავე თანამშრომლების საშუალო ასაკი და მათი რაოდენობა
- ბ. გაიცემა თითოეულ განყოფილებაში მომუშავე თანამშრომლების საშუალო ასაკი და მათი რაოდენობა
- გ. თითოეულ განყოფილებაში მომუშავე თანამშრომლების მინიმალური ასაკი და მათი რაოდენობა

4.6.18. SELECT SUM(gadaxdili_1) AS [გადახდილი თანხა]

FROM Xelshekruleba; მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა gadaxdili_1 სვეტის მნიშვნელობების სხვაობა
- ბ. არ გაიცემა gadaxdili_1 სვეტის მნიშვნელობების ჯამი
- გ. გაიცემა gadaxdili_1 სვეტის მნიშვნელობების ჯამი

4.6.19. SELECT MAX(vali_1) AS [მაქსიმალური ვალი]

FROM Xelshekruleba; მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა vali_1 სვეტის მაქსიმალური მნიშვნელობა
- ბ. გაიცემა vali_1 სვეტის მინიმალური მნიშვნელობა
- გ. გაიცემა vali_1 სვეტის საშუალო მნიშვნელობა

4.6.20. vali_1 სვეტის მაქსიმალური მნიშვნელობის საპოვნელად უნდა შევასრულოთ მოთხოვნა:

- ა. SELECT AVG(vali_1) AS [მაქსიმალური ვალი] FROM Xelshekruleba;
- ბ. SELECT MIN(vali_1) AS [მაქსიმალური ვალი] FROM Xelshekruleba;
- გ. SELECT MAX(vali_1) AS [მაქსიმალური ვალი] FROM Xelshekruleba;

4.6.21. SELECT MIN(vali_1) AS [მინიმალური ვალი]

FROM Xelshekruleba WHERE vali_1 > 0; მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა vali_1 სვეტის მაქსიმალური მნიშვნელობა
- ბ. გაიცემა vali_1 სვეტის მინიმალური მნიშვნელობა
- გ. გაიცემა vali_1 სვეტის საშუალო მნიშვნელობა

4.6.22. vali_1 სვეტის მინიმალური მნიშვნელობის საპოვნელად უნდა შევასრულოთ მოთხოვნა:

- ა. SELECT AVG(vali_1) AS [მინიმალური ვალი] FROM Xelshekruleba;
- ბ. SELECT MAX(vali_1) AS [მინიმალური ვალი] FROM Xelshekruleba;
- გ. SELECT MIN(vali_1) AS [მინიმალური ვალი] FROM Xelshekruleba;

GROUP BY განყოფილება

4.7.1. GROUP BY საკვანძო სიტყვა საშუალებას:

- ა. გვამღევს ცხრილის სტრიქონები გარკვეული კრიტერიუმის მიხედვით დავაჯგუფოთ
- ბ. არ გვამღევს ცხრილის სტრიქონები გარკვეული კრიტერიუმის მიხედვით დავაჯგუფოთ
- გ. გვამღევს ცხრილის სტრიქონები გარკვეული კრიტერიუმის მიხედვით არ დავაჯგუფოთ

4.7.2. თითოეული ჯგუფისათვის:

- ა. შეიძლება აგრეგირების ფუნქციების გამოყენება, რომლებიც არ შესრულდება ჯგუფის ყველა სტრიქონის მიმართ

ბ. არ შეიძლება აგრეგირების ფუნქციების გამოყენება, რომლებიც შესრულდება ჯგუფის ყველა სტრიქონის მიმართ

გ. შეიძლება აგრეგირების ფუნქციების გამოყენება, რომლებიც შესრულდება ჯგუფის ყველა სტრიქონის მიმართ

4.7.3. GROUP BY განყოფილების 'სვეტების_სია' არგუმენტში:

ა. არ ეთითება იმ სვეტების სახელები, რომელთა მიხედვით უნდა შესრულდეს დაჯგუფება

ბ. ეთითება იმ სვეტების სახელები, რომელთა მიხედვით უნდა შესრულდეს დაჯგუფება

გ. ეთითება იმ სვეტების სახელები, რომელთა მიხედვით არ უნდა შესრულდეს დაჯგუფება

4.7.4. თუ მოთხოვნაში განსაზღვრულია პირობა, რომელიც ზღუდავს ამოსარჩევი სტრიქონების დაჯგუფების დიაპაზონს, მაშინ GROUP BY განყოფილებაში ALL სიტყვის გამოყენება:

ა. გამოიწვევს ყველა ჯგუფის გამოტანას

ბ. არ გამოიწვევს ყველა ჯგუფის გამოტანას

გ. გამოიწვევს ყველა ჯგუფის შეტანას

4.7.5. აგრეგირების ფუნქცია:

ა. შესრულდება იმ ჯგუფებისათვის, რომლებიც არ აკმაყოფილებენ მოთხოვნაში მითითებულ პირობას

ბ. არ შესრულდება იმ ჯგუფებისათვის, რომლებიც არ აკმაყოფილებენ მოთხოვნაში მითითებულ პირობას

გ. არ შესრულდება იმ ჯგუფებისათვის, რომლებიც აკმაყოფილებენ მოთხოვნაში მითითებულ პირობას

4.7.6. SELECT personaliID, MAX(gadasaxdeli_1) AS [გადასახდელი თანხა]

FROM Xelshekruleba GROUP BY personaliID; მოთხოვნის შესრულების შედეგად:

ა. გაიცემა შეკვეთის მინიმალური თანხა თითოეული თანამშრომლისათვის

ბ. არ გაიცემა შეკვეთის მაქსიმალური თანხა თითოეული თანამშრომლისათვის

გ. გაიცემა შეკვეთის მაქსიმალური თანხა თითოეული თანამშრომლისათვის

4.7.7. SELECT personaliID, tarigi_dawyebis, MAX(gadasaxdeli_1) AS [გადასახდელი თანხა]

FROM Xelshekruleba

GROUP BY personaliID, tarigi_dawyebis; მოთხოვნის შესრულების შედეგად:

ა. გაიცემა თითოეული თანამშრომლის შეკვეთების მაქსიმალური თანხები თითოეული თარიღისათვის

ბ. არ გაიცემა თითოეული თანამშრომლის შეკვეთების მაქსიმალური თანხები თითოეული თარიღისათვის

გ. გაიცემა თითოეული თანამშრომლის შეკვეთების მინიმალური თანხები თითოეული თარიღისათვის

4.7.8. SELECT personaliID, SUM(gadasaxdeli_1) AS [გადასახდელი თანხა]

FROM Xelshekruleba

WHERE gadasaxdeli_1 < 7000

GROUP BY personaliID; მოთხოვნის შესრულების შედეგად:

ა. გაიცემა თითოეული თანამშრომლის იმ შეკვეთების თანხების ნამრავლი, რომელთა თანხა 7000-ზე ნაკლებია

ბ. გაიცემა თითოეული თანამშრომლის იმ შეკვეთების თანხების ჯამი, რომელთა თანხა 7000-ზე ნაკლებია

გ. არ გაიცემა თითოეული თანამშრომლის იმ შეკვეთების თანხების ჯამი, რომელთა თანხა 7000-ზე ნაკლებია

HAVING განყოფილება

4.8.1. HAVING განყოფილება გამოიყენება ძეზნის პირობის მისათითებლად:

- ა. ინდექსისთვის
- ბ. ტრიგერისთვის
- გ. ჯგუფისთვის

4.8.2. თუ HAVING განყოფილება გამოიყენება GROUP BY განყოფილების გარეშე, მაშინ ის:

- ა. იმუშავებს WHERE განყოფილების მსგავსად
- ბ. არ იმუშავებს WHERE განყოფილების მსგავსად
- გ. იმუშავებს FROM განყოფილების მსგავსად

4.8.3. თუ HAVING განყოფილება გამოიყენება GROUP BY ALL კონსტრუქციასთან ერთად, მაშინ HAVING:

- ა. არ ფარავს ALL სიტყვის მოქმედებას
- ბ. ფარავს ALL სიტყვის მოქმედებას
- გ. ფარავს FROM სიტყვის მოქმედებას

4.8.4. SELECT personaliID, MAX(gadasaxdeli_1) AS [გადასახდელი თანხა]
FROM Xelshekruleba
GROUP BY personaliID
HAVING MAX(gadasaxdeli_1) > 7000;

მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა თითოეული თანამშრომლის შეკვეთების მაქსიმალური თანხა, რომელიც არ აღემატება 7000-ს
- ბ. არ გაიცემა თითოეული თანამშრომლის შეკვეთების მაქსიმალური თანხა, რომელიც აღემატება 7000-ს
- გ. გაიცემა თითოეული თანამშრომლის შეკვეთების მაქსიმალური თანხა, რომელიც აღემატება 7000-ს

4.8.5. ინფორმაციის მისაღებად თითოეული თანამშრომლის შეკვეთების მაქსიმალური თანხის შესახებ, რომელიც აღემატება 7000-ს უნდა შევასრულოთ მოთხოვნა:

- ა. SELECT personaliID, MAX(gadasaxdeli_1) AS [გადასახდელი თანხა]
FROM Xelshekruleba GROUP BY personaliID
HAVING MAX(gadasaxdeli_1) > 7000;
- ბ. SELECT personaliID, MAX(gadasaxdeli_1) AS [გადასახდელი თანხა]
FROM Xelshekruleba GROUP BY personaliID
- გ. SELECT personaliID, MAX(gadasaxdeli_1) AS [გადასახდელი თანხა]
FROM Xelshekruleba

4.8.6. SELECT staji, COUNT(DISTINCT PersonaliID)

```
FROM Personali GROUP BY staji HAVING staji >
(
SELECT AVG(staji) FROM Personali WHERE qalaqi = N'თბილისი'
);
```

მოთხოვნის შესრულების შედეგად:

- ა. არ გაიცემა იმ თანამშრომლების რაოდენობა, რომელთა სტაჟი მეტია თბილისელი თანამშრომლების საშუალო სტაჟზე
- ბ. გაიცემა იმ თანამშრომლების რაოდენობა, რომელთა სტაჟი მეტია თბილისელი თანამშრომლების საშუალო სტაჟზე
- გ. გაიცემა იმ თანამშრომლების რაოდენობა, რომელთა სტაჟი ნაკლებია თბილისელი თანამშრომლების საშუალო სტაჟზე

4.8.7. ინფორმაციის მისაღებად იმ თანამშრომლების რაოდენობის შესახებ, რომელთა სტაჟი მეტია თბილისელი თანამშრომლების საშუალო სტაჟზე უნდა შევასრულოთ მოთხოვნა:

- ა.

```
SELECT staji, COUNT(DISTINCT PersonaliID)
FROM Personali GROUP BY staji HAVING staji >
(
SELECT AVG(staji) FROM Personali
);
```
- ბ.

```
SELECT staji, COUNT(DISTINCT PersonaliID)
FROM Personali GROUP BY staji HAVING staji <
(
SELECT AVG(staji) FROM Personali WHERE qalaqi = N'თბილისი'
);
```
- გ.

```
SELECT staji, COUNT(DISTINCT PersonaliID)
FROM Personali GROUP BY staji HAVING staji >
(
SELECT AVG(staji) FROM Personali WHERE qalaqi = N'თბილისი'
);
```

ORDER BY განყოფილება

4.9.1. ORDER BY განყოფილება გამოიყენება ცხრილის დასახარისხებლად ამა თუ იმ სვეტის მნიშვნელობების:

- ა. მხოლოდ ზრდის მიხედვით
- ბ. ზრდის ან კლების მიხედვით
- გ. მხოლოდ კლების მიხედვით

4.9.2. ცხრილი შეგვიძლია დავახარისხოთ:

- ა. ერთი ან მეტი სვეტის მიხედვით
- ბ. მხოლოდ ერთი სვეტის მიხედვით
- გ. არც ერთი სვეტის მიხედვით

4.9.3. ASC საკვანძო სიტყვა გამოიყენება მითითებული სვეტის მონაცემების დასალაგებლად:

- ა. არც ზრდადობის მიხედვით და არც კლებადობის მიხედვით
- ბ. კლებადობის მიხედვით

გ. ზრდადობის მიხედვით

4.9.4. DESC საკვანძო სიტყვა გამოიყენება მითითებული სვეტის მონაცემების დასალაგებლად:

ა. ზრდადობის მიხედვით

ბ. არც ზრდადობის მიხედვით და არც კლებადობის მიხედვით

გ. კლებადობის მიხედვით

4.9.5. ORDER BY განყოფილებაში თუ მითითებული არ არის ASC და DESC სიტყვები, მაშინ:

ა. დახარისხება ზრდადობის მიხედვით შესრულდება

ბ. დახარისხება კლებადობის მიხედვით შესრულდება

გ. დახარისხება არ შესრულდება არც ზრდადობის მიხედვით და არც კლებადობის მიხედვით

4.9.6. SELECT personaliID, tarigi_dawyebis, gadasaxdeli_1, vali_1

FROM Xelshekruleba ORDER BY tarigi_dawyebis; მოთხოვნის შესრულების შედეგად:

ა. გაიცემა შედეგი, რომელშიც tarigi_dawyebis სვეტის მონაცემები დახარისხებული იქნება კლებადობით

ბ. გაიცემა შედეგი, რომელშიც tarigi_dawyebis სვეტის მონაცემები დახარისხებული იქნება ზრდადობით

გ. არ გაიცემა შედეგი, რომელშიც tarigi_dawyebis სვეტის მონაცემები დახარისხებული იქნება ზრდადობით

4.9.7. შედეგის მისაღებად, რომელშიც tarigi_dawyebis სვეტის მონაცემები დახარისხებული იქნება ზრდადობით უნდა შევასრულოთ მოთხოვნა:

ა. SELECT personaliID, tarigi_dawyebis, gadasaxdeli_1, vali_1
FROM Xelshekruleba

ბ. SELECT personaliID, tarigi_dawyebis, gadasaxdeli_1, vali_1
FROM Xelshekruleba ORDER BY tarigi_dawyebis DESC;

გ. SELECT personaliID, tarigi_dawyebis, gadasaxdeli_1, vali_1
FROM Xelshekruleba ORDER BY tarigi_dawyebis;

4.9.8. SELECT personaliID, tarigi_dawyebis, gadasaxdeli_1, vali_1

FROM Xelshekruleba ORDER BY tarigi_dawyebis DESC; მოთხოვნის შესრულების შედეგად:

ა. გაიცემა შედეგი, რომელშიც tarigi_dawyebis სვეტის მონაცემები დახარისხებული იქნება კლებადობით

ბ. გაიცემა შედეგი, რომელშიც tarigi_dawyebis სვეტის მონაცემები დახარისხებული იქნება ზრდადობით

გ. არ გაიცემა შედეგი, რომელშიც tarigi_dawyebis სვეტის მონაცემები დახარისხებული იქნება კლებადობით

4.9.9. შედეგის მისაღებად, რომელშიც tarigi_dawyebis სვეტის მონაცემები დახარისხებული იქნება კლებადობით უნდა შევასრულოთ მოთხოვნა:

ა. SELECT personaliID, tarigi_dawyebis, gadasaxdeli_1, vali_1
FROM Xelshekruleba ORDER BY tarigi_dawyebis;

ბ. SELECT personaliID, tarigi_dawyebis, gadasaxdeli_1, vali_1
FROM Xelshekruleba ORDER BY tarigi_dawyebis DESC;

გ. SELECT personaliID, tarigi_dawyebis, gadasaxdeli_1, vali_1
FROM Xelshekruleba;

4.9.10. SELECT personaliID, tarigi_dawyebis, gadasaxdeli_1, vali_1
FROM Xelshekruleba ORDER BY tarigi_dawyebis, vali_1 DESC;

მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა შედეგი, რომელშიც tarigi_dawyebis სვეტი დახარისხებული იქნება კლებადობით, vali_1 სვეტი კი - კლებადობით
- ბ. არ გაიცემა შედეგი, რომელშიც tarigi_dawyebis სვეტი დახარისხებული იქნება ზრდადობით, vali_1 სვეტი კი - კლებადობით
- გ. გაიცემა შედეგი, რომელშიც tarigi_dawyebis სვეტი დახარისხებული იქნება ზრდადობით, vali_1 სვეტი კი - კლებადობით

4.9.11. შედეგის მისაღებად, რომელშიც tarigi_dawyebis სვეტი დახარისხებული იქნება ზრდადობით, vali_1 სვეტი კი - კლებადობით უნდა შევასრულოთ მოთხოვნა:

- ა. SELECT personaliID, tarigi_dawyebis, gadasaxdeli_1, vali_1
FROM Xelshekruleba ORDER BY tarigi_dawyebis;
- ბ. SELECT personaliID, tarigi_dawyebis, gadasaxdeli_1, vali_1
FROM Xelshekruleba ORDER BY tarigi_dawyebis, vali_1;
- გ. SELECT personaliID, tarigi_dawyebis, gadasaxdeli_1, vali_1
FROM Xelshekruleba ORDER BY tarigi_dawyebis, vali_1 DESC;

COMPUTE განყოფილება

4.10.1. COMPUTE განყოფილება საშუალებას გვაძლევს არჩეული:

- ა. სვეტების მიმართ გამოვიყენოთ აგრეგირების ფუნქციები
- ბ. სვეტების მიმართ არ გამოვიყენოთ აგრეგირების ფუნქციები
- გ. სტრიქონების მიმართ გამოვიყენოთ აგრეგირების ფუნქციები

4.10.2. COMPUTE განყოფილების გამოყენების შემთხვევაში, აგრეგირების შედეგი:

- ა. არ გაიცემა ცალკე სტრიქონის სახით, მონაცემების შემდეგ
- ბ. გაიცემა ცალკე სტრიქონის სახით, მონაცემების შემდეგ
- გ. გაიცემა ცალკე სვეტის სახით, მონაცემების შემდეგ

4.10.3. COMPUTE განყოფილებაში BY 'გამოსახულება' არგუმენტი:

- ა. უნდა შეიცავდეს სააგრეგირებელი სვეტის სახელს, რომელიც ჩართული არ იქნება შედეგში
- ბ. არ უნდა შეიცავდეს სააგრეგირებელი სვეტის სახელს, რომელიც ჩართული იქნება შედეგში
- გ. უნდა შეიცავდეს სააგრეგირებელი სვეტის სახელს, რომელიც ჩართული იქნება შედეგში

4.10.4. COMPUTE განყოფილებაში BY სიტყვა მიუთითებს, რომ გამოთვლის შედეგი:

- ა. უნდა დაჯგუფდეს
- ბ. არ უნდა დაჯგუფდეს
- გ. უნდა დაჯგუფდეს როგორც სვეტების, ისე სტრიქონების მიხედვით

4.10.5. COMPUTE განყოფილების გამოყენებისას შედეგი წინასწარ უნდა იყოს დახარისხებული იმ სვეტის მიხედვით, რომლის მიხედვითაც:

- ა. უნდა მოხდეს მონაცემთა ბაზის არჩევა
- ბ. არ უნდა შესრულდეს დაჯგუფება
- გ. უნდა შესრულდეს დაჯგუფება

4.10.6. COMPUTE განყოფილებაში:

- ა. დაუშვებელია DISTINCT საკვანძო სიტყვის გამოყენება
- ბ. დასაშვებია DISTINCT საკვანძო სიტყვის გამოყენება
- გ. დასაშვებია CREATE TABLE ბრძანების გამოყენება

4.10.7. SELECT personaliID, gadasaxdeli_1, vali_1, shesruleba

FROM Xelshekruleba

WHERE gadasaxdeli_1 > 2000

ORDER BY personaliID

COMPUTE SUM(gadasaxdeli_1), COUNT(gadasaxdeli_1);

მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა ყველა იმ ხელშეკრულებით გადასახდელი თანხების ჯამი და ხელშეკრულებების რაოდენობა, რომლებშიც გადასახდელი თანხა 2000-ს არ აღემატება
- ბ. გაიცემა ყველა იმ ხელშეკრულებით გადასახდელი თანხების ჯამი და ხელშეკრულებების რაოდენობა, რომლებშიც გადასახდელი თანხა 2000-ს აღემატება
- გ. არ გაიცემა ყველა იმ ხელშეკრულებით გადასახდელი თანხების ჯამი და ხელშეკრულებების რაოდენობა, რომლებშიც გადასახდელი თანხა 2000-ს აღემატება

4.10.8. იმისათვის, რომ მივიღოთ ყველა იმ ხელშეკრულებით გადასახდელი თანხების ჯამი და ხელშეკრულებების რაოდენობა, რომლებშიც გადასახდელი თანხა 2000-ს აღემატება, უნდა შევასრულოთ მოთხოვნა:

ა. SELECT personaliID, gadasaxdeli_1, vali_1, shesruleba

FROM Xelshekruleba

WHERE gadasaxdeli_1 > 2000

COMPUTE SUM(gadasaxdeli_1);

ბ. SELECT personaliID, gadasaxdeli_1, vali_1, shesruleba

FROM Xelshekruleba

WHERE gadasaxdeli_1 < 2000

ORDER BY personaliID

COMPUTE SUM(gadasaxdeli_1), COUNT(gadasaxdeli_1);

გ. SELECT personaliID, gadasaxdeli_1, vali_1, shesruleba

FROM Xelshekruleba

WHERE gadasaxdeli_1 > 2000

ORDER BY personaliID

COMPUTE SUM(gadasaxdeli_1), COUNT(gadasaxdeli_1);

4.10.9. SELECT personaliID, gadasaxdeli_1, vali_1, shesruleba

FROM Xelshekruleba

WHERE gadasaxdeli_1 > 9000

ORDER BY personaliID

COMPUTE SUM(gadasaxdeli_1), COUNT(gadasaxdeli_1) BY personaliID;

მოთხოვნის შესრულების შედეგად:

ა. გაიცემა personaliID სვეტის მიხედვით მიღებული ყველა იმ ხელშეკრულებით გადასახდელი თანხების ჯამი და ხელშეკრულებების რაოდენობა, რომლებშიც გადასახდელი თანხა 9000-ს აღემატება

ბ. არ გაიცემა personaliID სვეტის მიხედვით მიღებული ყველა იმ ხელშეკრულებით გადასახდელი თანხების ჯამი და ხელშეკრულებების რაოდენობა, რომლებშიც გადასახდელი თანხა 9000-ს აღემატება

გ. გაიცემა personaliID სვეტის მიხედვით მიღებული ყველა იმ ხელშეკრულებით გადასახდელი თანხების ჯამი და ხელშეკრულებების რაოდენობა, რომლებშიც გადასახდელი თანხა 9000-ს არ აღემატება

4.10.10. იმისათვის, რომ personaliID სვეტის მიხედვით მივიღოთ ყველა იმ ხელშეკრულებით გადასახდელი თანხების ჯამი და ხელშეკრულებების რაოდენობა, რომლებშიც გადასახდელი თანხა 9000-ს აღემატება, უნდა შევასრულოთ მოთხოვნა:

ა. SELECT personaliID, gadasaxdeli_1, vali_1, shesruleba

FROM Xelshekruleba

WHERE gadasaxdeli_1 < 9000

ORDER BY personaliID

COMPUTE SUM(gadasaxdeli_1), COUNT(gadasaxdeli_1) BY personaliID;

ბ. SELECT personaliID, gadasaxdeli_1, vali_1, shesruleba

FROM Xelshekruleba

WHERE gadasaxdeli_1 > 9000

ORDER BY personaliID

COMPUTE SUM(gadasaxdeli_1), COUNT(gadasaxdeli_1) BY personaliID;

გ. SELECT personaliID, gadasaxdeli_1, vali_1, shesruleba

FROM Xelshekruleba

WHERE gadasaxdeli_1 > 9000

ORDER BY personaliID

COMPUTE SUM(gadasaxdeli_1), COUNT(gadasaxdeli_1);

NULL მნიშვნელობასთან მუშაობა

4.11.1. NULL მნიშვნელობა:

ა. გამოიყენება გამოტოვებული მნიშვნელობებისა და მნიშვნელობების არარსებობის აღსანიშნავად

ბ. არ გამოიყენება გამოტოვებული მნიშვნელობებისა და მნიშვნელობების არარსებობის აღსანიშნავად

გ. გამოიყენება მხოლოდ მნიშვნელობების არარსებობის აღსანიშნავად

4.11.2. სამობით ლოგიკას იმას ნიშნავს, რომ პრედიკატებმა შეიძლება:

ა. ორიდან ერთ-ერთი მნიშვნელობა მიიღოს: TRUE (ჭეშმარიტი) და FALSE (მცდარი)

ბ. სამიდან ერთ-ერთი მნიშვნელობა მიიღოს: TRUE (ჭეშმარიტი), FALSE (მცდარი) და UNKNOWN (უცნობი)

გ. მხოლოდ ერთი მნიშვნელობა მიიღოს: TRUE (ჭეშმარიტი)

4.11.3. თუ ლოგიკური გამოსახულება მოწმდება გამოტოვებული მნიშვნელობისთვის (NULL, მნიშვნელობის არარსებობა), მაშინ ის:

- ა. იღებს TRUE (ჭეშმარიტი) მნიშვნელობას
- ბ. იღებს FALSE (მცდარი) მნიშვნელობას
- გ. იღებს UNKNOWN (უცნობი) მნიშვნელობას

4.11.4. განვიხილოთ პრედიკატი $asaki > 0$. თუ ასაკი 50-ის ტოლია, ლოგიკური გამოსახულება იღებს:

- ა. TRUE მნიშვნელობას
- ბ. FALSE მნიშვნელობას
- გ. UNKNOWN მნიშვნელობას

4.11.5. განვიხილოთ პრედიკატი $asaki > 0$. თუ ასაკი -50-ის ტოლია, ლოგიკური გამოსახულება იღებს:

- ა. TRUE მნიშვნელობას
- ბ. FALSE მნიშვნელობას
- გ. UNKNOWN მნიშვნელობას

4.11.6. განვიხილოთ პრედიკატი $asaki > 0$. თუ ასაკი NULL-ის ტოლია, ლოგიკური გამოსახულება იღებს:

- ა. TRUE მნიშვნელობას
- ბ. FALSE მნიშვნელობას
- გ. UNKNOWN მნიშვნელობას

4.11.7. UNKNOWN მნიშვნელობის მიმართ NOT (არა) ოპერაციის გამოყენება:

- ა. TRUE მნიშვნელობას გვაძლევს
- ბ. FALSE მნიშვნელობას გვაძლევს
- გ. UNKNOWN მნიშვნელობას გვაძლევს

4.11.8. თუ მოცემულია პრედიკატი NOT ($asaki > 0$) და ასაკი NULL-ის ტოლია, მაშინ $asaki > 0$ გამოსახულება გასცემს:

- ა. UNKNOWN მნიშვნელობას
- ბ. FALSE მნიშვნელობას
- გ. TRUE მნიშვნელობას

4.11.9. ორი NULL მნიშვნელობის შემდარებელი გამოსახულება ($NULL = NULL$) გასცემს:

- ა. FALSE მნიშვნელობას
- ბ. UNKNOWN მნიშვნელობას
- გ. TRUE მნიშვნელობას

4.11.10. დაჯგუფებისა და დახარისხების დროს ორი NULL მნიშვნელობა:

- ა. ხანდახან ითვლება ტოლად
- ბ. არ ითვლება ტოლად
- გ. ითვლება ტოლად

4.11.11. UNIQUE შეზღუდვის გამოყენების შემთხვევაში სვეტებში დასაშვებია მხოლოდ:

- ა. ორი NULL მნიშვნელობა
- ბ. ერთი NULL მნიშვნელობა
- გ. სამი NULL მნიშვნელობა

მონაცემების მასობრივი გადაწერა

4.12.1. მონაცემების იმპორტისა და ექსპორტისათვის გამოიყენება:

- ა. BCP უტილიტი
- ბ. DELETE ბრძანება
- გ. UPDATE ბრძანება

4.12.2. BCP უტილიტი ასრულებს მონაცემების გაცვლას სერვერსა და ტექსტურ ფაილებს შორის:

- ა. შეცვლას სერვერსა და ტექსტურ ფაილებს შორის
- ბ. გაცვლას სერვერსა და ტექსტურ ფაილებს შორის
- გ. წაშლას სერვერსა და ტექსტურ ფაილებს შორის

4.12.3. BCP უტილიტი ცხრილის სტრიქონებს წერს:

- ა. Excel-ის ფაილში და პირიქით
- ბ. ტექსტურ ფაილში და პირიქით
- გ. Access-ის ფაილში და პირიქით

4.12.4. BCP უტილიტის მუშაობის შედეგები:

- ა. აისახება შენახულ პროცედურაში
- ბ. აისახება ტრანზაქციების ჟურნალში
- გ. არ აისახება ტრანზაქციების ჟურნალში

4.12.5. თუ BCP უტილიტის მიერ გადაწერის ოპერაცია წარუმატებლად დამთავრდა, მაშინ:

- ა. საწყისი მონაცემების აღდგენას ვერ შევძლებთ
- ბ. საწყისი მონაცემების აღდგენას შევძლებთ
- გ. საწყისი მონაცემების აღდგენას ზოგიერთ შემთხვევაში შევძლებთ

4.12.6. `bcp "Shekveta.dbo.Personali" out "C:\Users\Romani\Documents\BCP\Personali1.txt" -w -T`

ბრძანების შესრულების შედეგად:

- ა. Shekveta მონაცემთა ბაზის Personali ცხრილიდან ტექსტურ ფაილში გადაიწერება ყველა სტრიქონი. მონაცემები არ დამუშავდება Unicode ფორმატში. გამოყენებული იქნება Windows NT-ის სააღრიცხვო ჩანაწერი
- ბ. Shekveta მონაცემთა ბაზის Personali ცხრილიდან ტექსტურ ფაილში გადაიწერება ყველა სტრიქონი. მონაცემები დამუშავდება Unicode ფორმატში. გამოყენებული იქნება Windows NT-ის სააღრიცხვო ჩანაწერი
- გ. Shekveta მონაცემთა ბაზის Personali ცხრილიდან ტექსტურ ფაილში გადაიწერება ყველა სტრიქონი. მონაცემები დამუშავდება Unicode ფორმატში. გამოყენებული არ იქნება Windows NT-ის სააღრიცხვო ჩანაწერი

4.12.7. იმისათვის, რომ Shekveta მონაცემთა ბაზის Personali ცხრილიდან ტექსტურ ფაილში გადაწეროთ ყველა სტრიქონი, მონაცემები დავამუშავოთ Unicode ფორმატში და გამოვიყენოთ Windows NT-ის სააღრიცხვო ჩანაწერი, უნდა შევასრულოთ ბრძანება:

- ა. bcp "Shekveta.dbo.Personali" out "C:\Users\Romani\Documents\ BCP\Personali1.txt" -T
- ბ. bcp "Shekveta.dbo.Personali" out "C:\Users\Romani\Documents\ BCP\Personali1.txt" -w
- გ. bcp "Shekveta.dbo.Personali" out "C:\Users\Romani\Documents\ BCP\Personali1.txt" -w -T

4.12.8. bcp "Shekveta.dbo.Personali" out "C:\Users\Romani\Documents\ BCP\Personali1.txt" -w -S"Serveri_1" -T ბრძანების შესრულების შედეგად:

- ა. Shekveta მონაცემთა ბაზის Personali ცხრილიდან ტექსტურ ფაილში გადაიწერება ყველა სტრიქონი. მონაცემები არ დამუშავდება Unicode ფორმატში. გამოყენებული იქნება Windows NT-ის სააღრიცხვო ჩანაწერი. სერვერის სახელია Serveri_3
- ბ. Shekveta მონაცემთა ბაზის Personali ცხრილიდან ტექსტურ ფაილში გადაიწერება ყველა სტრიქონი. მონაცემები დამუშავდება Unicode ფორმატში. გამოყენებული იქნება Windows NT-ის სააღრიცხვო ჩანაწერი. სერვერის სახელია Serveri_1
- გ. Shekveta მონაცემთა ბაზის Personali ცხრილიდან ტექსტურ ფაილში გადაიწერება ყველა სტრიქონი. მონაცემები დამუშავდება Unicode ფორმატში. გამოყენებული არ იქნება Windows NT-ის სააღრიცხვო ჩანაწერი. სერვერის სახელია Serveri_2

4.12.9. იმისათვის, რომ Serveri_1 სერვერის Shekveta მონაცემთა ბაზის Personali ცხრილიდან ტექსტურ ფაილში გადავწეროთ ყველა სტრიქონი, მონაცემები დავამუშავოთ Unicode ფორმატში და გამოვიყენოთ Windows NT-ის სააღრიცხვო ჩანაწერი, უნდა შევასრულოთ ბრძანება:

- ა. bcp "Shekveta.dbo.Personali" out "C:\Users\Romani\Documents\ BCP\Personali1.txt" -S"Serveri_3" -T
- ბ. bcp "Shekveta.dbo.Personali" out "C:\Users\Romani\Documents\ BCP\Personali1.txt" -w -S"Serveri_2"
- გ. bcp "Shekveta.dbo.Personali" out "C:\Users\Romani\Documents\ BCP\Personali1.txt" -w -S"Serveri_1" -T

4.12.10. bcp "Shekveta.dbo.Personali" out "C:\Users\Romani\Documents\ BCP\Personali1.txt" -w -S"Serveri_1" -U"sa" -P"paroli" ბრძანების შესრულების შედეგად:

- ა. Serveri_1 სერვერის Shekveta მონაცემთა ბაზის Personali ცხრილიდან ტექსტურ ფაილში გადაიწერება ყველა სტრიქონი. მითითებულია მომხმარებლის სახელი და პაროლი. სერვერის სახელია Serveri_1
- ბ. Serveri_1 სერვერის Shekveta მონაცემთა ბაზის Personali ცხრილიდან ტექსტურ ფაილში გადაიწერება ყველა სტრიქონი. მითითებულია მომხმარებლის სახელი და პაროლი. სერვერის სახელია Serveri_2
- გ. Serveri_1 სერვერის Shekveta მონაცემთა ბაზის Personali ცხრილიდან ტექსტურ ფაილში გადაიწერება ყველა სტრიქონი. არაა მითითებული მომხმარებლის სახელი და პაროლი. სერვერის სახელია Serveri_1

4.12.11. იმისათვის, რომ Serveri_1 სერვერის Shekveta მონაცემთა ბაზის Personali ცხრილიდან ტექსტურ ფაილში გადავწეროთ ყველა სტრიქონი, მივუთითოთ სერვერის სახელი - Serveri_1, მომხმარებლის სახელი და პაროლი, უნდა შევასრულოთ ბრძანება:

- ა. bcp "Shekveta.dbo.Personali" out "C:\Users\Romani\Documents\ BCP\Personali1.txt" -w -S"Serveri_2" -U"sa" -P"paroli"
- ბ. bcp "Shekveta.dbo.Personali" out "C:\Users\Romani\Documents\ BCP\Personali1.txt" -w -S"Serveri_1" -U"sa" -P"paroli"
- გ. bcp "Shekveta.dbo.Personali" out "C:\Users\Romani\Documents\ BCP\Personali1.txt"

-w -S"Serveri_1"

4.12.12.bcp "SELECT * FROM Shekveta.dbo.Personali"

queryout "C:\Users\Romani\Documents\BCP\Personali.txt" -w -T ბრძანების შესრულების შედეგად:

- ა. Shekveta მონაცემთა ბაზის Personali ცხრილიდან ტექსტურ ფაილში არ გადაიწერება ყველა სტრიქონი. გამოყენებულია SELECT მოთხოვნა და Windows NT-ის სააღრიცხვო ჩანაწერი
- ბ. Shekveta მონაცემთა ბაზის Personali ცხრილიდან ტექსტურ ფაილში გადაიწერება ყველა სტრიქონი. არაა გამოყენებული SELECT მოთხოვნა და Windows NT-ის სააღრიცხვო ჩანაწერი
- გ. Shekveta მონაცემთა ბაზის Personali ცხრილიდან ტექსტურ ფაილში გადაიწერება ყველა სტრიქონი. გამოყენებულია SELECT მოთხოვნა და Windows NT-ის სააღრიცხვო ჩანაწერი

4.12.13.იმისათვის, რომ Shekveta მონაცემთა ბაზის Personali ცხრილიდან ტექსტურ ფაილში გადაეწეროთ ყველა სტრიქონი, გამოვიყენოთ SELECT მოთხოვნა და Windows NT-ის სააღრიცხვო ჩანაწერი, უნდა შევასრულოთ მოთხოვნა:

- ა. bcp "SELECT * FROM Shekveta.dbo.Shemkveti"
queryout "C:\Users\Romani\Documents\BCP\Personali.txt" -w -T
- ბ. bcp "SELECT * FROM Shekveta.dbo.Personali"
queryout "C:\Users\Romani\Documents\BCP\Personali.txt" -w
- გ. bcp "SELECT * FROM Shekveta.dbo.Personali"
queryout "C:\Users\Romani\Documents\BCP\Personali.txt" -w -T

4.12.14.bcp "Shekveta.dbo.Personali" in "C:\Users\Romani\Documents\ BCP\Personali.txt" -w -T

ბრძანების შესრულების შედეგად:

- ა. Shekveta მონაცემთა ბაზის Personali ცხრილს არ დაემატება სტრიქონები ტექსტური ფაილიდან. მონაცემები დამუშავდება Unicode ფორმატში. გამოყენებული იქნება Windows NT-ის სააღრიცხვო ჩანაწერი
- ბ. Shekveta მონაცემთა ბაზის Personali ცხრილს დაემატება სტრიქონები ტექსტური ფაილიდან. მონაცემები დამუშავდება Unicode ფორმატში. გამოყენებული იქნება Windows NT-ის სააღრიცხვო ჩანაწერი
- გ. Shekveta მონაცემთა ბაზის Personali ცხრილს დაემატება სტრიქონები ტექსტური ფაილიდან. მონაცემები დამუშავდება Unicode ფორმატში. გამოყენებული არ იქნება Windows NT-ის სააღრიცხვო ჩანაწერი

4.12.15.იმისათვის, რომ Shekveta მონაცემთა ბაზის Personali ცხრილს დავუმატოთ სტრიქონები ტექსტური ფაილიდან, მონაცემები დავამუშავოთ Unicode ფორმატში და გამოვიყენოთ Windows NT-ის სააღრიცხვო ჩანაწერი, უნდა შევასრულოთ მოთხოვნა:

- ა. bcp "Shekveta.dbo.Personali" out "C:\Users\Romani\Documents\ BCP\Personali1.txt" -T
- ბ. bcp "Shekveta.dbo.Personali" out "C:\Users\Romani\Documents\ BCP\Personali1.txt" -w
- გ. bcp "Shekveta.dbo.Personali" out "C:\Users\Romani\Documents\ BCP\Personali1.txt" -w -T

თაზო 5. შეერთებები

FROM განყოფილება

- 5.1.1. SELECT მოთხოვნის FROM განყოფილება (ელემენტი) ლოგიკურად:
- ა. პირველი მუშავდება
 - ბ. მეორე მუშავდება
 - გ. მესამე მუშავდება
- 5.1.2. FROM განყოფილებაში შესასვლელი (საწყისი) ცხრილების მიმართ გამოიყენება:
- ა. მხოლოდ PIVOT და UNPIVOT ცხრილური ოპერაციები
 - ბ. JOIN, APPLY, PIVOT და UNPIVOT ცხრილური ოპერაციები
 - გ. მხოლოდ JOIN და UNPIVOT ცხრილური ოპერაციები
- 5.1.3. არსებობს შეერთების:
- ა. ორი ძირითადი ტიპი: ჯვარედინი და გარე
 - ბ. ორი ძირითადი ტიპი: შიდა და გარე
 - გ. სამი ძირითადი ტიპი: ჯვარედინი, შიდა და გარე
- 5.1.4. შეერთების ჯვარედინი, შიდა და გარე ტიპი ერთმანეთისაგან:
- ა. მოთხოვნის ლოგიკური დამუშავების სტადიებით განსხვავდება და სტადიების სხვადასხვა შემადგენლობა აქვს
 - ბ. მოთხოვნის ლოგიკური დამუშავების სტადიებით არ განსხვავდება და სტადიების სხვადასხვა შემადგენლობა აქვს
 - გ. მოთხოვნის ლოგიკური დამუშავების სტადიებით განსხვავდება და სტადიების სხვადასხვა შემადგენლობა არ აქვს
- 5.1.5. ჯვარედინი შეერთება დამუშავების:
- ა. ორ სტადიას მოიცავს - დეკარტულ ნამრავლს და გაფილტვრას
 - ბ. მხოლოდ ერთ სტადიას მოიცავს - დეკარტულ ნამრავლს (წარმოებულს)
 - გ. სამ სტადია მოიცავს - დეკარტულ ნამრავლს, გაფილტვრასა და გარე სტრიქონების დამატებას
- 5.1.6. შიდა შეერთება დამუშავების:
- ა. ორ სტადიას მოიცავს - დეკარტულ ნამრავლს და გაფილტვრას
 - ბ. მხოლოდ ერთ სტადიას მოიცავს - დეკარტულ ნამრავლს (წარმოებულს)
 - გ. სამ სტადია მოიცავს - დეკარტულ ნამრავლს, გაფილტვრასა და გარე სტრიქონების დამატებას
- 5.1.7. გარე შეერთება დამუშავების:
- ა. ორ სტადიას მოიცავს - დეკარტულ ნამრავლს და გაფილტვრას
 - ბ. მხოლოდ ერთ სტადიას მოიცავს - დეკარტულ ნამრავლს (წარმოებულს)
 - გ. სამ სტადია მოიცავს - დეკარტულ ნამრავლს, გაფილტვრასა და გარე სტრიქონების დამატებას
- 5.1.8. SELECT მოთხოვნაში FROM სიტყვის შემდეგ ეთითება იმ:
- ა. ცხრილების ან წარმოდგენების სახელები, საიდანაც ხდება სვეტების ამორჩევა

- ბ. ინდექსების სახელები, საიდანაც ხდება სვეტების ამორჩევა
- გ. პროცედურების სახელები, საიდანაც ხდება სვეტების ამორჩევა

5.1.9. FROM განყოფილებაში <ბმული_ცხრილი> კონსტრუქცია გამოიყენება კავშირის უზრუნველსაყოფად:

- ა. რამდენიმე პროცედურას შორის
- ბ. რამდენიმე ფუნქციას შორის
- გ. რამდენიმე ცხრილს შორის

5.1.10. FROM განყოფილების <ბმული_ცხრილი> კონსტრუქციაში <მარცხენა_ცხრილი> კონსტრუქცია შეიცავს:

- ა. დამოკიდებული ცხრილის სახელს, რომელსაც დაუკავშირდება სხვა ცხრილები
- ბ. მთავარი ცხრილის სახელს, რომელსაც დაუკავშირდება სხვა ცხრილები
- გ. მთავარი ცხრილის სახელს, რომელსაც დაუკავშირდება სხვა ინდექსები

5.1.11. FROM განყოფილების <ბმული_ცხრილი> კონსტრუქციაში <ბმის_ტიპი> კონსტრუქცია განსაზღვრავს:

- ა. ორ ტრიგერს შორის კავშირის ტიპს
- ბ. ორ პროცედურას შორის კავშირის ტიპს
- გ. ორ ცხრილს შორის კავშირის ტიპს

5.1.12. FROM განყოფილების <ბმული_ცხრილი> კონსტრუქციაში <ბმის_ტიპი> კონსტრუქციის:

- ა. მარცხნივ ეთითება მთავარი ცხრილის სახელი
- ბ. მარჯვნივ ეთითება მთავარი ცხრილის სახელი
- გ. მარცხნივ ეთითება დამოკიდებული ცხრილის სახელი

5.1.13. FROM განყოფილების <ბმული_ცხრილი> კონსტრუქციაში <ბმის_ტიპი> კონსტრუქციის:

- ა. მარცხნივ ეთითება დამოკიდებული ცხრილის სახელი
- ბ. მარჯვნივ ეთითება დამოკიდებული ცხრილის სახელი
- გ. მარჯვნივ ეთითება მთავარი ცხრილის სახელი

5.1.14. FROM განყოფილების <ბმული_ცხრილი> კონსტრუქციაში <ბმის_ტიპი> კონსტრუქციის INNER კავშირის ტიპის მითითების შემთხვევაში:

- ა. შედეგში ჩართული იქნება მარჯვენა ცხრილის ყველა სტრიქონი მიუხედავად იმისა, აქვთ თუ არა მათ შესაბამისი სტრიქონები მარცხენა ცხრილში. მარცხენა ცხრილის შესაბამის სვეტებს NULL მნიშვნელობა ექნებათ
- ბ. შედეგში ჩართული იქნება მარცხენა ცხრილის ყველა სტრიქონი, მიუხედავად იმისა, არსებობს თუ არა მათთვის შესაბამისი სტრიქონი მარჯვენა ცხრილში. მარჯვენა ცხრილის შესაბამის სვეტებს NULL მნიშვნელობა ექნებათ
- გ. შედეგში ჩართული იქნება მარცხენა ცხრილის მხოლოდ ის სტრიქონები, რომლებისთვისაც არსებობენ სტრიქონები ბმულ (მარჯვენა) ცხრილში. ჩართული იქნება, აგრეთვე, მარჯვენა ცხრილის მხოლოდ ის სტრიქონები, რომლებისთვისაც არსებობენ შესაბამისი სტრიქონები მარცხენა ცხრილში

5.1.15. FROM განყოფილების <ბმული_ცხრილი> კონსტრუქციაში <ბმის_ტიპი> კონსტრუქციის LEFT კავშირის ტიპის მითითების შემთხვევაში:

ა. შედეგში ჩართული იქნება მარცხენა ცხრილის ყველა სტრიქონი, მიუხედავად იმისა, არსებობს თუ არა მათთვის შესაბამისი სტრიქონი მარჯვენა ცხრილში. მარჯვენა ცხრილის შესაბამის სვეტებს NULL მნიშვნელობა ექნებათ

ბ. შედეგში ჩართული იქნება მარჯვენა ცხრილის ყველა სტრიქონი მიუხედავად იმისა, აქვთ თუ არა მათ შესაბამისი სტრიქონები მარცხენა ცხრილში. მარცხენა ცხრილის შესაბამის სვეტებს NULL მნიშვნელობა ექნებათ

გ. შედეგში ჩართული იქნება მარცხენა ცხრილის მხოლოდ ის სტრიქონები, რომლებისთვისაც არსებობენ სტრიქონები ბმულ (მარჯვენა) ცხრილში. ჩართული იქნება, აგრეთვე, მარჯვენა ცხრილის მხოლოდ ის სტრიქონები, რომლებისთვისაც არსებობენ შესაბამისი სტრიქონები მარცხენა ცხრილში

5.1.16. FROM განყოფილების <ბმული_ცხრილი> კონსტრუქციაში <ბმის_ტიპი> კონსტრუქციის RIGHT კავშირის ტიპის მითითების შემთხვევაში:

ა. შედეგში ჩართული იქნება მარცხენა ცხრილის მხოლოდ ის სტრიქონები, რომლებისთვისაც არსებობენ სტრიქონები ბმულ (მარჯვენა) ცხრილში. ჩართული იქნება, აგრეთვე, მარჯვენა ცხრილის მხოლოდ ის სტრიქონები, რომლებისთვისაც არსებობენ შესაბამისი სტრიქონები მარცხენა ცხრილში

ბ. შედეგში ჩართული იქნება მარჯვენა ცხრილის ყველა სტრიქონი მიუხედავად იმისა, აქვთ თუ არა მათ შესაბამისი სტრიქონები მარცხენა ცხრილში. მარცხენა ცხრილის შესაბამის სვეტებს NULL მნიშვნელობა ექნებათ

გ. შედეგში ჩართული იქნება მარცხენა ცხრილის ყველა სტრიქონი, მიუხედავად იმისა, არსებობს თუ არა მათთვის შესაბამისი სტრიქონი მარჯვენა ცხრილში. მარჯვენა ცხრილის შესაბამის სვეტებს NULL მნიშვნელობა ექნებათ

5.1.17. FROM განყოფილების <ბმული_ცხრილი> კონსტრუქციაში <ბმის_ტიპი> კონსტრუქციის FULL კავშირის ტიპის მითითების შემთხვევაში:

ა. შედეგში ჩართული იქნება მარცხენა და მარჯვენა ცხრილების ყველა სტრიქონი

ბ. შედეგში ჩართული იქნება მარცხენა ცხრილის ყველა სტრიქონი, მიუხედავად იმისა, არსებობს თუ არა მათთვის შესაბამისი სტრიქონი მარჯვენა ცხრილში

გ. შედეგში ჩართული იქნება მარჯვენა ცხრილის ყველა სტრიქონი მიუხედავად იმისა, აქვთ თუ არა მათ შესაბამისი სტრიქონები მარცხენა ცხრილში

5.1.18. FROM განყოფილების <ბმული_ცხრილი> კონსტრუქციაში <ბმის_ტიპი> კონსტრუქციის JOIN არგუმენტის შემდეგ ეთითება:

ა. შუა ცხრილი

ბ. მარცხენა ცხრილი

გ. მარჯვენა ცხრილი

5.1.19. FROM განყოფილების <ბმული_ცხრილი> კონსტრუქციაში ON <ბმის_პირობა> არის:

ა. ორივე ცხრილის დაკავშირების ლოგიკური პირობა

ბ. ორივე ცხრილის დაკავშირების ფიზიკური პირობა

გ. სამივე ცხრილის დაკავშირების ლოგიკური პირობა

5.1.20. FROM განყოფილების <ბმული_ცხრილი> კონსტრუქციაში CROSS JOIN საკვანძო სიტყვის გამოყენებისას სრულდება:

ა. მარჯვენა ცხრილის თითოეული სტრიქონის დაკავშირება მარცხენა ცხრილის თითოეულ

სტრიქონთან

ბ. მარცხენა ცხრილის თითოეული სტრიქონის დაკავშირება მარჯვენა ცხრილის თითოეულ სტრიქონთან

გ. მარცხენა ცხრილის თითოეული სტრიქონის დაკავშირება მარცხენა ცხრილის თითოეულ სტრიქონთან

5.1.21. FROM განყოფილების <ბმული_ცხრილი> კონსტრუქციაში თუ ცხრილებს შორის კავშირი არ არის მითითებული, მაშინ:

ა. ავტომატურად სრულდება მარცხენა ცხრილის თითოეული სტრიქონის დაკავშირება მარცხენა ცხრილის თითოეულ სტრიქონთან

ბ. ავტომატურად არ სრულდება მარცხენა ცხრილის თითოეული სტრიქონის დაკავშირება მარჯვენა ცხრილის თითოეულ სტრიქონთან

გ. ავტომატურად სრულდება მარცხენა ცხრილის თითოეული სტრიქონის დაკავშირება მარჯვენა ცხრილის თითოეულ სტრიქონთან

ჯვარედინი შეერთებები

5.2.1. ჯვარედინი შეერთება:

ა. აფორმირებს ორი ცხრილის სტრიქონების დეკარტულ ნამრავლს

ბ. არ აფორმირებს ორი ცხრილის სტრიქონების დეკარტულ ნამრავლს

ბ. აფორმირებს სამი ცხრილის სტრიქონების დეკარტულ ნამრავლს

5.2.2. ჯვარედინი შეერთების დროს:

ა. ერთი ცხრილის თითოეულ სტრიქონს არ შეეთანადება მეორე ცხრილის ყველა სტრიქონი

ბ. ერთი ცხრილის თითოეულ სტრიქონს შეეთანადება მეორე ცხრილის ყველა სტრიქონი

გ. ორი ცხრილის თითოეულ სტრიქონს შეეთანადება სამი ცხრილის ყველა სტრიქონი

5.2.3. ჯვარედინი შეერთების დროს: თუ ერთ ცხრილში გვაქვს m სტრიქონი და მეორეში - n სტრიქონი, მაშინ სტრიქონების შედეგობრივ ნაკრებში მივიღებთ:

ა. $2 \times m \times n$ სტრიქონს

ბ. $m \times 2 \times n$ სტრიქონს

გ. $m \times n$ სტრიქონს

5.2.4. პრეფიქსი იძლევა სვეტის ცალსახად იდენტიფიცირების საშუალებას მაშინ, როცა ორივე ცხრილში:

ა. გვხვდება სვეტების ერთნაირი სახელები

ბ. გვხვდება სვეტების სხვადასხვა სახელები

გ. არ გვხვდება სვეტების სხვადასხვა სახელები

5.2.5. ცხრილებს ფსევდონიმები:

ა. სახელის სიმოკლისთვის არ ენიჭება

ბ. სახელის სიმოკლისთვის ენიჭება

გ. ყოველთვის ენიჭება

5.2.6. სვეტების პრეფიქსების მითითება აუცილებელია მაშინ, როცა:

ა. სვეტის ერთი და იგივე სახელი არც ერთ ცხრილში არ გვხვდება

- ბ. სვეტის ერთი და იგივე სახელი ორივე ცხრილში არ გვხვდება
- გ. სვეტის ერთი და იგივე სახელი ორივე ცხრილში გვხვდება

5.2.7. თუ სვეტების სახელები ცალსახადაა განსაზღვრული, მაშინ პრეფიქსები:

- ა. შეგვიძლია არ გამოვიყენოთ
- ბ. აუცილებლად უნდა გამოვიყენოთ
- გ. უნდა გამოვიყენოთ სტრიქონების სახელების წინ

5.2.8. თვითშეერთება ეწოდება:

- ა. სხვადასხვა ცხრილის რამდენიმე ეგზემპლარის შეერთების შესაძლებლობას
- ბ. ერთი და იგივე ცხრილის რამდენიმე ეგზემპლარის შეერთების შესაძლებლობას
- გ. ერთი და იგივე ცხრილის რამდენიმე ეგზემპლარის შეერთების შეუძლებლობას

5.2.9. თვითშეერთების დროს:

- ა. შეგვიძლია გამოვიყენოთ მხოლოდ ერთი ტიპის შეერთება
- ბ. არ შეგვიძლია გამოვიყენოთ ყველა ძირითადი ტიპის შეერთები
- გ. შეგვიძლია გამოვიყენოთ ყველა ძირითადი ტიპის შეერთები

შიდა შეერთებები. შიდა შეერთების სახეები. შედგენილი შეერთებები. შეერთებები უტოლობის შემთხვევაში. მრავალცხრილიანი შეერთებები

5.3.1. შიდა შეერთების დროს სრულდება მოთხოვნის ლოგიკური დამუშავების:

- ა. ორი სტადია
- ბ. ერთი სტადია
- გ. სამი სტადია

5.3.2. შიდა შეერთების დროს:

- ა. ჯერ, ამოირჩევა სტრიქონები, შემდეგ კი განისაზღვრება ორი შესასვლელი ცხრილის დეკარტული ნამრავლი
- ბ. ჯერ, განისაზღვრება ორი შესასვლელი ცხრილის დეკარტული ნამრავლი, შემდეგ კი ამოირჩევა სტრიქონები
- გ. მხოლოდ ამოირჩევა სტრიქონები

5.3.3. შიდა შეერთების სახეებია:

- ა. შეერთებები უტოლობის შემთხვევაში და მრავალცხრილური შეერთებები
- ბ. შედგენილი შეერთებები (composite joins) და მრავალცხრილური შეერთებები
- გ. შედგენილი შეერთებები (composite joins), შეერთებები უტოლობის შემთხვევაში და მრავალცხრილური შეერთებები

5.3.4. შედგენილი შეერთება ეფუძნება პრედიკატს, რომელიც:

- ა. მოიცავს რამდენიმე სვეტს თითოეული ცხრილიდან
- ბ. არ მოიცავს რამდენიმე სვეტს თითოეული ცხრილიდან
- გ. მოიცავს მხოლოდ ერთ სვეტს

5.3.5. შედგენილ შეერთებას ადგილი აქვს მაშინ, როცა:

- ა. პირველადი გასაღები შედგება რამდენიმე სვეტისგან, ხოლო გარე გასაღები - ერთი სვეტისაგან
- ბ. პირველადი გასაღები შედგება რამდენიმე სვეტისგან და გარე გასაღებიც შედგება ამდენივე სვეტისგან
- გ. პირველადი გასაღები შედგება ერთი სვეტისგან, ხოლო გარე გასაღები - რამდენიმე სვეტისგან

5.3.6. თუ შეერთების პირობა მხოლოდ ტოლობის ოპერაციას შეიცავს, მაშინ ეს:

- ა. არის შეერთება ტოლობის პირობით ან ეკვივალენტური შეერთება
- ბ. არ არის შეერთება ტოლობის პირობით ან ეკვივალენტური შეერთება
- გ. არის შეერთება უტოლობის პირობით

5.3.7. თუ შეერთების პირობაში გამოიყენება შედარების ნებისმიერი ოპერაცია ტოლობის ოპერაციის გარდა, მაშინ ასეთ შეერთებას ეწოდება:

- ა. შეერთება ტოლობის პირობით
- ბ. შეერთება როგორც ტოლობის, ისე უტოლობის პირობით
- გ. შეერთება უტოლობის პირობით

5.3.8. FROM განყოფილება შეიძლება მოიცავდეს რამდენიმე ცხრილურ ოპერაციას. ეს ოპერაციები ლოგიკურად:

- ა. მუშავდება მარცხნიდან მარჯვნივ
- ბ. მუშავდება მარჯვნიდან მარცხნივ
- გ. არ მუშავდება

5.3.9. FROM განყოფილება შეიძლება მოიცავდეს რამდენიმე ცხრილურ ოპერაციას. ეს ოპერაციები ლოგიკურად მუშავდება მარცხნიდან მარჯვნივ. ეს იმას ნიშნავს, რომ:

- ა. შედეგობრივი ცხრილი, მიღებული პირველ ცხრილურ ოპერაციაში, არ გამოიყენება მარცხენა ცხრილად მეორე ცხრილური ოპერაციისათვის
- ბ. შედეგობრივი ცხრილი, მიღებული პირველ ცხრილურ ოპერაციაში, გამოიყენება მარცხენა ცხრილად მეორე ცხრილური ოპერაციისათვის
- გ. შედეგობრივი ცხრილი, მიღებული პირველ ცხრილურ ოპერაციაში, გამოიყენება მარჯვენა ცხრილად მეორე ცხრილური ოპერაციისათვის

5.3.10. თუ FROM განყოფილებაში არის რამდენიმე შეერთება, პირველი შეერთება ასრულებს ორი საბაზო ცხრილის ლოგიკურ დამუშავებას, ხოლო ყველა დანარჩენი შეერთებები:

- ა. იღებენ წინა შეერთების შედეგს, როგორც მარჯვენა ცხრილს
- ბ. არ იღებენ წინა შეერთების შედეგს, როგორც მარცხენა ცხრილს
- გ. იღებენ წინა შეერთების შედეგს, როგორც მარცხენა ცხრილს

5.3.11. Xelshekruleba.gadasaxdeli_1, Xelshekruleba.vali_1

FROM Personali INNER JOIN Xelshekruleba

ON Personali.personaliID = Xelshekruleba.personaliID

გამოყენებულია:

- ა. შიდა შეერთება
- ბ. გარე შეერთება

გ. ჯვარედინი შეერთება

```
5.3.12. SELECT T1.col1, T1.col2, T2.col1, T2.col2
FROM Table1 AS T1 JOIN Table2 AS T2
ON T1.col1 = T2.col1 AND T1.col2 = T2.col2;
```

მოთხოვნაში:

- ა. გამოყენებულია გარე შეერთება
- ბ. გამოყენებულია შედგენილი შეერთება
- გ. არ არის გამოყენებული შედგენილი შეერთება

```
5.3.13. SELECT P1.personaliID, P1.saxeli, P1.gvari, P2.personaliID, P2.saxeli, P2.gvari
FROM Personali AS P1 JOIN Personali AS P2
ON P1.personaliID < P2.personaliID;
```

მოთხოვნაში:

- ა. გამოყენებულია გარე შეერთება
- ბ. არ არის გამოყენებული შეერთება უტოლობის შემთხვევაში
- გ. გამოყენებულია შეერთება უტოლობის შემთხვევაში

გარე შეერთებები

5.4.1. გარე შეერთებები შიდა შეერთებების დროს გამოყენებული ლოგიკური დამუშავების ორივე სტადიის გარდა:

- ა. მოიცავს მესამე, მხოლოდ მისთვის დამახასიათებელ სტადიას, რომელსაც გარე სტრიქონების დამატება ეწოდება
- ბ. არ მოიცავს მესამე, მხოლოდ მისთვის დამახასიათებელ სტადიას, რომელსაც გარე სტრიქონების დამატება ეწოდება
- გ. მოიცავს მესამე, მხოლოდ მისთვის დამახასიათებელ სტადიას, რომელსაც შიდა სტრიქონების დამატება ეწოდება

5.4.2. გარე შეერთებაში:

- ა. ორივე ცხრილი მოინიშნება როგორც "შესანახი" (მთავარი) LEFT OUTER JOIN, RIGHT OUTER JOIN ან FULL OUTER JOIN საკვანძო სიტყვების საშუალებით
- ბ. ერთი ცხრილი მოინიშნება როგორც "შესანახი" (მთავარი) LEFT OUTER JOIN, RIGHT OUTER JOIN ან FULL OUTER JOIN საკვანძო სიტყვების საშუალებით, მეორე კი - როგორც „არაშესანახი“ (დამხმარე)
- გ. ორივე ცხრილი მოინიშნება როგორც „არაშესანახი“ (დამხმარე)

5.4.3. გარე შეერთებაში:

- ა. OUTER საკვანძო სიტყვის მითითება ხანდახან არის აუცილებელი
- ბ. OUTER საკვანძო სიტყვის მითითება აუცილებელია
- გ. OUTER საკვანძო სიტყვის მითითება არ არის აუცილებელი

5.4.4. გარე შეერთებაში LEFT საკვანძო სიტყვა ნიშნავს, რომ:

- ა. შეინახება მარჯვენა ცხრილის სტრიქონები (შედეგში გამოჩნდება მარჯვენა ცხრილის ყველა სტრიქონი)
- ბ. არ შეინახება მარცხენა ცხრილის სტრიქონები (შედეგში არ გამოჩნდება მარცხენა

ცხრილის ყველა სტრიქონი)

გ. შეინახება მარცხენა ცხრილის სტრიქონები (შედეგში გამოჩნდება მარცხენა ცხრილის ყველა სტრიქონი)

5.4.5. გარე შეერთებაში RIGHT სიტყვა მიუთითებს, რომ:

ა. არ შეინახება მარჯვენა ცხრილის სტრიქონები (შედეგში არ გამოჩნდება მარჯვენა ცხრილის ყველა სტრიქონი)

ბ. შეინახება მარჯვენა ცხრილის სტრიქონები (შედეგში გამოჩნდება მარჯვენა ცხრილის ყველა სტრიქონი)

გ. შეინახება მარცხენა ცხრილის სტრიქონები (შედეგში გამოჩნდება მარცხენა ცხრილის ყველა სტრიქონი)

5.4.6. გარე შეერთებაში FULL საკვანძო სიტყვა მიუთითებს, რომ:

ა. შეინახება ორივე ცხრილის სტრიქონები (შედეგში გამოჩნდება ორივე ცხრილის ყველა სტრიქონი)

ბ. არ შეინახება ორივე ცხრილის სტრიქონები (შედეგში არ გამოჩნდება ორივე ცხრილის ყველა სტრიქონი)

გ. შეინახება მხოლოდ მარცხენა ცხრილის სტრიქონები

5.4.7. გარე შეერთებაში მოთხოვნის ლოგიკური დამუშავების მესამე სტადია:

ა. არ განსაზღვრავს სტრიქონებს შესანახი ცხრილიდან, რომლებსთვისაც არ მოინახა შესაბამისი სტრიქონები არაშესანახ ცხრილში ON ელემენტის პრედიკატის საფუძველზე

ბ. განსაზღვრავს სტრიქონებს შესანახი ცხრილიდან, რომლებსთვისაც არ მოინახა შესაბამისი სტრიქონები არაშესანახ ცხრილში ON ელემენტის პრედიკატის საფუძველზე

გ. განსაზღვრავს სტრიქონებს შესანახი ცხრილიდან, რომლებსთვისაც მოინახა შესაბამისი სტრიქონები არაშესანახ ცხრილში ON ელემენტის პრედიკატის საფუძველზე

5.4.8. SELECT S.shemkvetiID, S.firmis_dasaxeleba, X.xelshekrulebaID

FROM Shemkveti AS S LEFT OUTER JOIN Xelshekruleba AS X

ON S.shemkvetiID = X.shemkvetiID;

მოთხოვნის შესრულების შედეგში:

ა. გამოჩნდება Shemkveti ცხრილის ყველა სტრიქონი

ბ. არ გამოჩნდება Shemkveti ცხრილის ყველა სტრიქონი

გ. გამოჩნდება Xelshekruleba ცხრილის ყველა სტრიქონი

5.4.9. SELECT S.shemkvetiID, S.firmis_dasaxeleba, X.xelshekrulebaID

FROM Shemkveti AS S RIGHT OUTER JOIN Xelshekruleba AS X

ON S.shemkvetiID = X.shemkvetiID;

მოთხოვნის შესრულების შედეგში:

ა. გამოჩნდება Shemkveti ცხრილის ყველა სტრიქონი

ბ. არ გამოჩნდება Xelshekruleba ცხრილის ყველა სტრიქონი

გ. გამოჩნდება Xelshekruleba ცხრილის ყველა სტრიქონი

5.4.10. SELECT S.shemkvetiID, S.firmis_dasaxeleba, X.xelshekrulebaID

FROM Shemkveti AS S FULL OUTER JOIN Xelshekruleba AS X

ON S.shemkvetiID = X.shemkvetiID;

მოთხოვნის შესრულების შედეგში:

- ა. არ გამოჩნდება Shemkveti და Xelshekruleba ცხრილების ყველა სტრიქონი
- ბ. გამოჩნდება Shemkveti და Xelshekruleba ცხრილების ყველა სტრიქონი
- გ. გამოჩნდება მხოლოდ Xelshekruleba ცხრილის ყველა სტრიქონი

თავი 6. მმართველი კონსტრუქციები

BEGIN...END

6.1.1. BEGIN...END კონსტრუქცია:

- ა. იძლევა ერთ ბლოკში ორი და მეტი ბრძანების დაჯგუფების საშუალებას
- ბ. არ იძლევა ერთ ბლოკში ორი და მეტი ბრძანების დაჯგუფების საშუალებას
- გ. იძლევა ერთ ბლოკში მხოლოდ ერთი ბრძანების დაჯგუფების საშუალებას

6.1.2. BEGIN საკვანძო სიტყვა:

- ა. არც იწყებს და არც ამთავრებს ბლოკს
- ბ. იწყებს ბლოკს
- გ. ამთავრებს ბლოკს

6.1.3. END საკვანძო სიტყვა:

- ა. იწყებს ბლოკს
- ბ. არც იწყებს და არც ამთავრებს ბლოკს
- გ. ამთავრებს ბლოკს

6.1.4. BEGIN...END კონსტრუქცია:

- ა. არ შეგვიძლია გამოვიყენოთ განშტოებებში, პირობისა და ციკლის კონსტრუქციებში
- ბ. შეგვიძლია გამოვიყენოთ მხოლოდ განშტოებებში
- გ. შეგვიძლია გამოვიყენოთ განშტოებებში, პირობისა და ციკლის კონსტრუქციებში

6.1.5. BEGIN...END ბლოკი:

- ა. შეიძლება იყოს ჩადგმული
- ბ. შეიძლება იყოს ჩაუდგმელი
- გ. არ შეიძლება იყოს ჩადგმული

6.1.6.

BEGIN

DECLARE @cvladi INT;

SET @cvladi = 5;

END

ბლოკი მოიცავს ბრძანებებს:

- ა. BEGIN, DECLARE @cvladi INT; და SET @cvladi = 5;
- ბ. DECLARE @cvladi INT; და SET @cvladi = 5;
- გ. DECLARE @cvladi INT; , SET @cvladi = 5; და END

6.1.7.

BEGIN

DECLARE @cvladi INT;

SET @cvladi = 5;

END

ბრძანებების მიმდევრობა:

- ა. ხანდახან არის პროგრამული ბლოკი
- ბ. არ არის პროგრამული ბლოკი
- გ. არის პროგრამული ბლოკი

6.1.8.

```
DECLARE @cvladi INT;  
SET @cvladi = 5;  
END
```

ბრძანებების მიმდევრობა:

- ა. ხანდახან არის პროგრამული ბლოკი
- ბ. არ არის პროგრამული ბლოკი
- გ. არის პროგრამული ბლოკი

6.1.9. BEGIN

```
DECLARE @cvladi INT;  
SET @cvladi = 5;
```

ბრძანებების მიმდევრობა:

- ა. ხანდახან არის პროგრამული ბლოკი
- ბ. არ არის პროგრამული ბლოკი
- გ. არის პროგრამული ბლოკი

IF...ELSE

6.2.1. IF...ELSE კონსტრუქციაში 'ლოგიკური_გამოსახულება':

- ა. არ არის ლოგიკური პირობა
- ბ. არის ფიზიკური პირობა
- გ. არის ლოგიკური პირობა

6.2.2. IF...ELSE კონსტრუქციაში 'ლოგიკური_გამოსახულება':

- ა. იღებს true (ჭეშმარიტი) ან false (მცდარი) მნიშვნელობას
- ბ. იღებს მნიშვნელობებს 10 ან 20
- გ. არ იღებს true (ჭეშმარიტი) ან false (მცდარი) მნიშვნელობას

6.2.3. თუ IF...ELSE კონსტრუქციაში 'ლოგიკური_გამოსახულება' იღებს true მნიშვნელობას, მაშინ:

- ა. შესრულდება პირველი 'sql_ბრძანება' ან 'ბრძანებების_ბლოკი'
- ბ. შესრულდება პირველი 'sql_ბრძანება' ან 'ბრძანებების_ბლოკი'
- გ. შესრულდება ELSE სიტყვის შემდეგ მოთავსებული 'sql_ბრძანება' ან 'ბრძანებების_ბლოკი'

6.2.4. თუ IF...ELSE კონსტრუქციაში 'ლოგიკური_გამოსახულება' იღებს false მნიშვნელობას, მაშინ:

- ა. შესრულდება პირველი 'sql_ბრძანება' ან 'ბრძანებების_ბლოკი'
- ბ. არ შესრულდება ELSE სიტყვის შემდეგ მოთავსებული 'sql_ბრძანება' ან 'ბრძანებების_ბლოკი'
- გ. შესრულდება ELSE სიტყვის შემდეგ მოთავსებული 'sql_ბრძანება' ან 'ბრძანებების_ბლოკი'

6.2.5. DECLARE @kenti_luwi INT;

```
SET @kenti_luwi = 10;
```

```
IF @kenti_luwi % 2 = 0
```

```
SELECT N'რიცხვი ლუწია';
```


ELSE

SELECT N'რიცხვი კენტია';

პროგრამული კოდის შესრულების შედეგად გაიცემა:

- ა. 'რიცხვი ლუწია'
- ბ. 'რიცხვი კენტია'
- გ. არაფერი არ გაიცემა

CASE

6.3.1. CASE:

- ა. არის ერთგანშტოებიანი კონსტრუქცია
- ბ. არის მრავალგანშტოებიანი კონსტრუქცია
- გ. არ არის მრავალგანშტოებიანი კონსტრუქცია

6.3.2. 'case_გამოსახულება':

- ა. შეიძლება შეიცავდეს ცვლადის სახელს ან ფუნქციას
- ბ. შეიძლება შეიცავდეს ტრიგერის სახელს ან შენახულ პროცედურას
- გ. არ შეიძლება შეიცავდეს ცვლადის სახელს ან ფუნქციას

6.3.3. CASE კონსტრუქციაში 'when_გამოსახულება':

- ა. არ იღებს 'case_გამოსახულება' არგუმენტის მნიშვნელობებიდან ერთ-ერთს
- ბ. იღებს მნიშვნელობებს 0 ან 1
- გ. იღებს 'case_გამოსახულება' არგუმენტის მნიშვნელობებიდან ერთ-ერთს

6.3.4. თუ CASE კონსტრუქციაში 'when_გამოსახულება' და 'case_გამოსახულება' არგუმენტების მნიშვნელობა ერთმანეთს დაემთხვევა, მაშინ:

- ა. გაიცემა 'then_გამოსახულება' არგუმენტის მნიშვნელობა
- ბ. გაიცემა 'case_გამოსახულება' არგუმენტის მნიშვნელობა
- გ. არ გაიცემა 'then_გამოსახულება' არგუმენტის მნიშვნელობა

6.3.5. თუ CASE კონსტრუქციაში 'case_გამოსახულება' არგუმენტების მნიშვნელობები არ ემთხვევა 'when_გამოსახულება' არგუმენტის არც ერთ მნიშვნელობას, მაშინ:

- ა. არ გაიცემა 'else_გამოსახულება' არგუმენტის მნიშვნელობა
- ბ. გაიცემა 'else_გამოსახულება' არგუმენტის მნიშვნელობა
- გ. გაიცემა 'case_გამოსახულება' არგუმენტის მნიშვნელობა

COALESCE

6.4.1. COALESCE კონსტრუქცია:

- ა. გასცემს მეორე არანულოვან მნიშვნელობას
- ბ. არ გასცემს პირველ არანულოვან მნიშვნელობას
- გ. გასცემს პირველ არანულოვან მნიშვნელობას

6.4.2. DECLARE @ricxvi_1 INT, @ricxvi_2 FLOAT, @ricxvi_3 INT;

SET @ricxvi_1 = 33;

SET @ricxvi_2 = 20.55;

PRINT COALESCE(@ricxvi_3, @ricxvi_1, @ricxvi_2);

პროგრამული კოდის შესრულების შედეგად გაიცემა:

- ა. 33
- ბ. 20.55
- გ. არაფერი არ გაიცემა

WHILE...BREAK & CONTINUE

6.5.1. WHILE...BREAK & CONTINUE კონსტრუქცია:

- ა. არ გამოიყენება ციკლების ორგანიზებისათვის
- ბ. გამოიყენება ციკლების ორგანიზებისათვის
- გ. გამოიყენება ციკლების წაშლისთვის

6.5.2. თუ WHILE...BREAK & CONTINUE კონსტრუქციაში 'ლოგიკური_გამოსახულება' იღებს true მნიშვნელობას, მაშინ:

- ა. შესრულდება ციკლის ტანი (კოდი)
- ბ. შესრულდება ციკლის შემდეგ მოთავსებული ოპერატორი
- გ. არ შესრულდება ციკლის ტანი (კოდი)

6.5.3. თუ WHILE...BREAK & CONTINUE კონსტრუქციაში 'ლოგიკური_გამოსახულება' false მნიშვნელობას მიიღებს, მაშინ:

- ა. ციკლის შესრულება გაგრძელდება
- ბ. ციკლის შესრულება არ შეწყდება
- გ. ციკლის შესრულება შეწყდება

6.5.4. BREAK ბრძანების შესრულება:

- ა. იწვევს ციკლის შესრულების იძულებით შეწყვეტას და ციკლიდან გამოსვლას
- ბ. იწვევს ლოგიკური პირობის შემოწმებაზე გადასვლას ბრძანების შესრულება
- გ. არ იწვევს ციკლის შესრულების იძულებით შეწყვეტას და ციკლიდან გამოსვლას

6.5.5. CONTINUE ბრძანების შესრულება:

- ა. არ იწვევს ლოგიკური პირობის შემოწმებაზე გადასვლას ბრძანების შესრულება
- ბ. იწვევს ლოგიკური პირობის შემოწმებაზე გადასვლას ბრძანების შესრულება
- გ. იწვევს ციკლის შესრულების იძულებით შეწყვეტას და ციკლიდან გამოსვლას

6.5.6. CONTINUE ბრძანების შემდეგ მოთავსებული ბრძანებები:

- ა. არ შესრულდება
- ბ. შესრულდება
- გ. ხანდახან შესრულდება

6.5.7. DECLARE @ricxvi_1 INT;

SET @ricxvi_1 = 1;

WHILE @ricxvi_1 <= 10

BEGIN

PRINT STR(@ricxvi_1) + N'-ის კვადრატი =' + STR(SQUARE(@ricxvi_1));

SET @ricxvi_1 = @ricxvi_1 + 1;

END

პროგრამული კოდის შესრულების შედეგად:

- ა. არ გაიცემა 1-დან 10-მდე რიცხვების კვადრატი

- ბ. გაიცემა 1-დან 10-მდე რიცხვების კვადრატი
- გ. არაფერი არ გაიცემა

```
6.5.8. DECLARE @ricxvi_1 INT;
SET @ricxvi_1 = 1;
WHILE 10 = 10
BEGIN
PRINT STR(@ricxvi_1) + N'-ის კვადრატი =' + STR(SQUARE(@ricxvi_1));
SET @ricxvi_1 = @ricxvi_1 + 1;
IF @ricxvi_1 <= 10 CONTINUE;
BREAK;
END
```

პროგრამული კოდის შესრულების შედეგად:

- ა. არ გაიცემა 1-დან 10-მდე რიცხვების კვადრატი
- ბ. გაიცემა 1-დან 10-მდე რიცხვების კვადრატი
- გ. არაფერი არ გაიცემა

თავი 7. ქვემოთხოვნები

დამოუკიდებელი სკალარული ქვემოთხოვნები. დამოუკიდებელი ქვემოთხოვნები მრავლობითი მნიშვნელობებით

- 7.1.1. ჩადგმული მოთხოვნა გვაქვს მაშინ, როცა ერთი მოთხოვნის შიგნით:
- ა. ოთხი მოთხოვნაა მოთავსებული
 - ბ. სამი მოთხოვნაა მოთავსებული
 - გ. მეორეა მოთავსებული
- 7.1.2. მოთხოვნას, რომელიც მეორე მოთხოვნას მოიცავს:
- ა. გარე ეწოდება
 - ბ. შიდა ეწოდება
 - გ. არფერი არ ეწოდება
- 7.1.3. შიდა მოთხოვნა (ქვემოთხოვნა) - ესაა მოთხოვნა, რომლის მიერ გაცემული სტრიქონები (შედეგობრივი ნაკრები):
- ა. გამოიყენება შიდა მოთხოვნის მიერ
 - ბ. გამოიყენება გარე მოთხოვნის მიერ
 - გ. არ გამოიყენება გარე მოთხოვნის მიერ
- 7.1.4. ქვემოთხოვნის მიერ გაცემული შედეგი::
- ა. შეიძლება არ იცვლებოდეს მასში მითითებულ ცხრილებში შეტანილი ცვლილებების გამო
 - ბ. არ შეიძლება იცვლებოდეს მასში მითითებულ ცხრილებში შეტანილი ცვლილებების გამო
 - გ. შეიძლება იცვლებოდეს მასში მითითებულ ცხრილებში შეტანილი ცვლილებების გამო
- 7.1.5. ქვემოთხოვნების გამოყენება საშუალებას გვაძლევს თავი:
- ა. ავარიდოთ ისეთ მოქმედებებს, რომლებიც ითხოვენ მოთხოვნის შუალედური შედეგების შენახვას ცვლადებში
 - ბ. არ ავარიდოთ ისეთ მოქმედებებს, რომლებიც ითხოვენ მოთხოვნის შუალედური შედეგების შენახვას ცვლადებში
 - გ. ავარიდოთ ისეთ მოქმედებებს, რომლებიც არ ითხოვენ მოთხოვნის შუალედური შედეგების შენახვას ცვლადებში
- 7.1.6. ქვემოთხოვნა შეიძლება:
- ა. არ იყოს დამოუკიდებელი (მარტივი) ან ბმული (კორელირებული)
 - ბ. იყოს დამოუკიდებელი (მარტივი) ან ბმული (კორელირებული)
 - გ. იყოს მხოლოდ დამოუკიდებელი (მარტივი)
- 7.1.7. დამოუკიდებელი ქვემოთხოვნა:
- ა. არის დამოკიდებული გარე მოთხოვნაზე, რომელსაც ის არ ეკუთვნის
 - ბ. არის დამოკიდებული გარე მოთხოვნაზე, რომელსაც ის ეკუთვნის
 - გ. არ არის დამოკიდებული გარე მოთხოვნაზე, რომელსაც ის ეკუთვნის
- 7.1.8. ბმული ქვემოთხოვნა:
- ა. არ არის დამოკიდებული გარე მოთხოვნაზე, რომელსაც ის ეკუთვნის
 - ბ. დამოკიდებულია გარე მოთხოვნაზე, რომელსაც ის არ ეკუთვნის

- გ. დამოკიდებულია გარე მოთხოვნაზე, რომელსაც ის ეკუთვნის
- 7.1.9. როგორც დამოუკიდებელ, ისე ბმულ ქვემოთხოვნებს:
- ა. შეუძლია გასცეს მხოლოდ ერთი (სკალარული) მნიშვნელობა
 - ბ. შეუძლია გასცეს ერთი (სკალარული) მნიშვნელობა, ბევრი (მრავლობითი) მნიშვნელობა ან ცხრილი
 - გ. შეუძლია გასცეს მხოლოდ ცხრილი
- 7.1.10. დამოუკიდებელი ქვემოთხოვნის შემთხვევაში:
- ა. ჯერ სრულდება შიდა მოთხოვნა, შემდეგ კი გარე
 - ბ. ჯერ სრულდება გარე მოთხოვნა, შემდეგ კი შიდა
 - გ. სრულდება მხოლოდ გარე მოთხოვნა
- 7.1.11. სკალარულია ქვემოთხოვნა, რომელიც:
- ა. გასცემს მხოლოდ ერთ მნიშვნელობას, მიუხედავად იმისა ის ბმულია თუ არა
 - ბ. გასცემს რამდენიმე მნიშვნელობას, მიუხედავად იმისა ის ბმულია თუ არა
 - გ. არ გასცემს მხოლოდ ერთ მნიშვნელობას, მიუხედავად იმისა ის ბმულია თუ არა
- 7.1.12. სკალარული ქვემოთხოვნა:
- ა. არ შეიძლება ჩავრთოთ გარე მოთხოვნის იმ ელემენტებში (WHERE, SELECT და ა.შ.), რომლებიც შეიძლება სკალარულ გამოსახულებებს შეიცავდეს
 - ბ. შეიძლება ჩავრთოთ გარე მოთხოვნის იმ ელემენტებში (WHERE, SELECT და ა.შ.), რომლებიც შეიძლება სკალარულ გამოსახულებებს შეიცავდეს
 - გ. შეიძლება ჩავრთოთ გარე მოთხოვნის იმ ელემენტებში (WHERE, SELECT და ა.შ.), რომლებიც შეიძლება სკალარულ გამოსახულებებს არ შეიცავდეს
- 7.1.13. ქვემოთხოვნა მრავლობითი მნიშვნელობებით არის ქვემოთხოვნა, რომელიც:
- ა. რამდენიმე (მრავლობით) მნიშვნელობას გასცემს ერთი სვეტის სახით მხოლოდ იმ შემთხვევაში თუ ის ბმული ქვემოთხოვნა არ არის
 - ბ. რამდენიმე (მრავლობით) მნიშვნელობას გასცემს ერთი სვეტის სახით მხოლოდ იმ შემთხვევაში თუ ის ბმული ქვემოთხოვნაა
 - გ. რამდენიმე (მრავლობით) მნიშვნელობას გასცემს ერთი სვეტის სახით იმისგან დამოუკიდებლად, ბმულია თუ არა ეს ქვემოთხოვნა
- 7.1.14. IN პრედიკატში:
- ა. შეგვიძლია გამოვიყენოთ ქვემოთხოვნები მხოლოდ ერთი მნიშვნელობით
 - ბ. არ შეგვიძლია გამოვიყენოთ ქვემოთხოვნები მრავლობითი მნიშვნელობებით
 - გ. შეგვიძლია გამოვიყენოთ ქვემოთხოვნები მრავლობითი მნიშვნელობებით
- 7.1.15. თუ IN პრედიკატში 'სკალარული_გამოსახულება' ერთ-ერთი მნიშვნელობის ტოლია, რომელიც ქვემოთხოვნამ გასცა, მაშინ პრედიკატი იღებს:
- ა. TRUE მნიშვნელობას
 - ბ. FALSE მნიშვნელობას
 - გ. UNKNOWN მნიშვნელობას

7.1.16. თუ IN პრედიკატში 'სკალარული_გამოსახულება' არ არის არც ერთი მნიშვნელობის ტოლი, რომელიც ქვემოთხოვნამ გასცა, მაშინ პრედიკატი იღებს:

- ა. TRUE მნიშვნელობას
- ბ. FALSE მნიშვნელობას
- გ. UNKNOWN მნიშვნელობას

7.1.17. SELECT xelshekrulebaID, personaliID, shemkvetiID, tarigi_dawyebis, vali_1
FROM Xelshekruleba
WHERE vali_1 =

(
SELECT MAX(X.vali_1)
FROM Xelshekruleba AS X
); მოთხოვნაში ჩადგმული მოთხოვნა გასცემს:

- ა. ერთ მნიშვნელობას
- ბ. ორ მნიშვნელობას
- გ. სამ მნიშვნელობას

7.1.18. SELECT xelshekrulebaID, personaliID, shemkvetiID, tarigi_dawyebis, vali_1
FROM Xelshekruleba
WHERE vali_1 =

(
SELECT MAX(X.vali_1)
FROM Xelshekruleba AS X
); მოთხოვნის შესრულების შედეგად გაიცემა:

- ა. მაქსიმალური ვალის მნიშვნელობა
- ბ. მინიმალური ვალის მნიშვნელობა
- გ. ვალის საშუალო მნიშვნელობა

7.1.19. SELECT shemkvetiID, xelshekrulebaID, tarigi_damtavrebis, personaliID
FROM Xelshekruleba
WHERE shemkvetiID IN

(
SELECT S.shemkvetiID
FROM Shemkveti AS S
WHERE S.qalaqi = N'თბილისი'

); მოთხოვნის შესრულების შედეგად:

- ა. არ გაიცემა ხელშეკრულებები, რომლებიც გააფორმეს თბილისელმა შემკვეთებმა
- ბ. გაიცემა ხელშეკრულებები, რომლებიც გააფორმეს თბილისელმა შემკვეთებმა
- გ. გაიცემა ხელშეკრულებები, რომლებიც არ გააფორმეს თბილისელმა შემკვეთებმა

7.1.20. იმისათვის, რომ მივიღოთ ინფორმაცია თბილისელი შემკვეთების მიერ გაფორმებული ხელშეკრულებების შესახებ უნდა შევასრულოთ მოთხოვნა:

ა. SELECT shemkvetiID, xelshekrulebaID, tarigi_damtavrebis, personaliID
FROM Xelshekruleba
WHERE shemkvetiID IN

(

```

SELECT S.shemkvetiID
FROM Shemkveti AS S
WHERE S.qalaqi = N'ბათუმი'
);
ბ. SELECT shemkvetiID, xelshekrulebaID, tarigi_damtavrebis, personaliID
FROM Xelshekruleba
WHERE shemkvetiID IN
(
SELECT S.shemkvetiID
FROM Shemkveti AS S
WHERE S.qalaqi = N'თბილისი'
);
გ. SELECT shemkvetiID, xelshekrulebaID, tarigi_damtavrebis, personaliID
FROM Xelshekruleba
WHERE shemkvetiID NOT IN
(
SELECT S.shemkvetiID
FROM Shemkveti AS S
WHERE S.qalaqi = N'თბილისი'
);

```

ბმული ქვემოთხოვნები

7.2.1. ბმული ქვემოთხოვნა არის ისეთი ქვემოთხოვნა, რომელიც:

- ა. გარე მოთხოვნაში გამოყენებული ცხრილის სვეტებს არ მიმართავს
- ბ. გარე მოთხოვნაში გამოყენებული ცხრილის სვეტებს მიმართავს
- გ. შიდა მოთხოვნაში გამოყენებული ცხრილის სვეტებს მიმართავს

7.2.2. ბმული მოთხოვნების გამოყენების დროს:

- ა. შიდა მოთხოვნა სრულდება გარე მოთხოვნაში მითითებული ცხრილის თითოეული სტრიქონისათვის
- ბ. გარე მოთხოვნა სრულდება შიდა მოთხოვნაში მითითებული ცხრილის თითოეული სტრიქონისათვის
- გ. შიდა მოთხოვნა სრულდება შიდა მოთხოვნაში მითითებული ცხრილის თითოეული სტრიქონისათვის

7.2.3. ბმული მოთხოვნების გამოყენების დროს:

- ა. ჯერ ამოიჩვენა შიდა მოთხოვნის სტრიქონი, შემდეგ კი მისთვის შესრულდება გარე მოთხოვნა
- ბ. ჯერ ამოიჩვენა გარე მოთხოვნის სტრიქონი, შემდეგ კი მისთვის შესრულდება შიდა მოთხოვნა
- გ. ჯერ ამოიჩვენა შიდა მოთხოვნის სტრიქონი, შემდეგ კი მისთვის შესრულდება შიდა მოთხოვნა

```

7.2.4. SELECT P.personaliID, P.gvari, P.saxeli
FROM Personali P

```

```
WHERE 1 <
(
SELECT COUNT(*)
FROM Xelshekruleba X
WHERE P.personaliID = X.personaliID
); მოთხოვნის შესრულების შედეგად:
```

- ა. გაიცემა იმ თანამშრომლების გვარები და ნომრები, რომლებსაც ერთზე მეტი შემკვეთი არ ჰყავთ
- ბ. არ გაიცემა იმ თანამშრომლების გვარები და ნომრები, რომლებსაც ერთზე მეტი შემკვეთი ჰყავთ
- გ. გაიცემა იმ თანამშრომლების გვარები და ნომრები, რომლებსაც ერთზე მეტი შემკვეთი ჰყავთ

7.2.5. იმისათვის, რომ მივიღოთ იმ თანამშრომლების გვარები და ნომრები, რომლებსაც ერთზე მეტი შემკვეთი ჰყავთ უნდა შევასრულოთ მოთხოვნა:

```
ა. SELECT P.personaliID, P.gvari, P.saxeli
FROM Personali P
WHERE 1 >
(
SELECT COUNT(*)
FROM Xelshekruleba X
WHERE P.personaliID = X.personaliID
);
```

```
ბ. SELECT P.personaliID, P.gvari, P.saxeli
FROM Personali P
WHERE 1 =
(
SELECT COUNT(*)
FROM Xelshekruleba X
WHERE P.personaliID = X.personaliID
);
```

```
გ. SELECT P.personaliID, P.gvari, P.saxeli
FROM Personali P
WHERE 1 <
(
SELECT COUNT(*)
FROM Xelshekruleba X
WHERE P.personaliID = X.personaliID
);
```

ლოგიკის ოპერატორები. ALL და ANY ოპერატორები.

7.3.1. ALL ოპერატორი სკალარულ სიდიდეს:

- ა. ადარებს ქვემოთხოვნის მიერ დაბრუნებულ თითოეულ მნიშვნელობას
- ბ. გასცემს ქვემოთხოვნის შედეგს
- გ. არ ადარებს ქვემოთხოვნის მიერ დაბრუნებულ თითოეულ მნიშვნელობას

7.3.2. თუ ALL ოპერატორში ლოგიკური პირობა სრულდება ყველა დაბრუნებული მნიშვნელობისათვის, მაშინ პირობა:

- ა. ჩაითვლება ნახევრად შესრულებულად
- ბ. ჩაითვლება შესრულებულად
- გ. არ ჩაითვლება შესრულებულად

7.3.3. ALL ოპერატორში ქვემოთხოვანა:

- ა. არ გასცემს ერთი სვეტის მნიშვნელობებს
- ბ. გასცემს ერთი სვეტის მნიშვნელობებს
- გ. გასცემს რამდენიმე სვეტის მნიშვნელობებს

7.3.4. ALL ოპერატორში ქვემოთხოვანა გასცემს ერთი სვეტის მნიშვნელობებს, რომლის მნიშვნელობების ტიპი:

- ა. არ უნდა ემთხვეოდეს 'გამოსახულების' ტიპს
- ბ. ხანდახან უნდა ემთხვეოდეს 'გამოსახულების' ტიპს
- გ. უნდა ემთხვეოდეს 'გამოსახულების' ტიპს

7.3.5. SOME და ANY ოპერატორები სკალარულ სიდიდეს:

- ა. ადარებენ ქვემოთხოვანის მიერ გაცემულ თითოეულ მნიშვნელობას
- ბ. ადარებენ პროცედურის მიერ გაცემულ თითოეულ მნიშვნელობას
- გ. არ ადარებენ ქვემოთხოვანის მიერ გაცემულ თითოეულ მნიშვნელობას

7.3.6. თუ SOME და ANY ოპერატორებში ლოგიკური პირობა სრულდება ერთ-ერთი მათგანისათვის, მაშინ პირობა:

- ა. არ ჩაითვლება შესრულებულად
- ბ. ჩაითვლება შესრულებულად
- გ. ჩაითვლება ნახევრად შესრულებულად

7.3.7. IF N'კი' = ALL

```
(  
SELECT shesruleba  
FROM Xelshekruleba  
)  
SELECT N'ყველა ხელშეკრულება შესრულდა';  
ELSE  
SELECT N'ყველა ხელშეკრულება არ შესრულდა';
```

მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა ინფორმაცია იმის შესახებ ყველა ხელშეკრულება შესრულდა თუ არა
- ბ. არ გაიცემა ინფორმაცია იმის შესახებ ყველა ხელშეკრულება შესრულდა თუ არა
- გ. გაიცემა ინფორმაცია იმის შესახებ ხოგიერთი ხელშეკრულება შესრულდა თუ არა

7.3.8. ინფორმაციის მისაღებად იმის შესახებ ყველა ხელშეკრულება შესრულდა თუ არა უნდა შევასრულოთ მოთხოვნა:

ა. IF N'კი' = ANY
(

```

SELECT shesruleba
FROM Xelshekruleba
)
  SELECT N'ყველა ხელშეკრულება შესრულდა';
ELSE
  SELECT N'ყველა ხელშეკრულება არ შესრულდა';
ბ.      IF N'კი' = ALL
(
SELECT shesruleba
FROM Xelshekruleba
)
  SELECT N'ყველა ხელშეკრულება შესრულდა';
ELSE
  SELECT N'ყველა ხელშეკრულება არ შესრულდა';
გ.      IF N'კი' != ALL
(
SELECT shesruleba
FROM Xelshekruleba
)
  SELECT N'ყველა ხელშეკრულება შესრულდა';
ELSE
  SELECT N'ყველა ხელშეკრულება არ შესრულდა';

```

7.3.9. IF N'არა' = ANY

```

(
SELECT shesruleba
FROM Xelshekruleba
)
  SELECT N'ზოგიერთი ხელშეკრულება არ შესრულდა';
ELSE
SELECT N'ყველა ხელშეკრულება შესრულდა';
მოთხოვნის შესრულების შედეგად:

```

- ა. გაიცემა ინფორმაცია იმის შესახებ, რომელიმე ხელშეკრულება შესრულდა თუ არა
- ბ. არ გაიცემა ინფორმაცია იმის შესახებ, რომელიმე ხელშეკრულება შესრულდა თუ არა
- გ. გაიცემა ინფორმაცია იმის შესახებ, ყველა ხელშეკრულება შესრულდა თუ არა

7.3.10. თუ გვაინტერესებს, რომელიმე ხელშეკრულება შესრულდა თუ არა, უნდა შევასრულოთ მოთხოვნა:

```

ა.      IF N'არა' != ANY
(
SELECT shesruleba
FROM Xelshekruleba
)
  SELECT N'ზოგიერთი ხელშეკრულება არ შესრულდა';
ELSE
SELECT N'ყველა ხელშეკრულება შესრულდა';

```

```

ბ. IF N'არა' = ANY
(
SELECT shesruleba
FROM Xelshekruleba
)
SELECT N'ზოგიერთი ხელშეკრულება არ შესრულდა';
ELSE
SELECT N'ყველა ხელშეკრულება შესრულდა';
გ. IF N'არა' = ALL
(
SELECT shesruleba
FROM Xelshekruleba
)
SELECT N'ზოგიერთი ხელშეკრულება არ შესრულდა';
ELSE
SELECT N'ყველა ხელშეკრულება შესრულდა';

```

EXISTS ოპერატორი

7.4.1. EXISTS ოპერატორი გასცემს true მნიშვნელობას, თუ ქვემოთხოვნა:

- ა. არ გასცემს ორ სტრიქონს
- ბ. არ გასცემს თუნდაც ერთ სტრიქონს
- გ. გასცემს თუნდაც ერთ სტრიქონს

7.4.2. EXISTS ოპერატორი გასცემს false მნიშვნელობას, თუ ქვემოთხოვნა:

- ა. გასცემს თუნდაც ერთ სტრიქონს
- ბ. არ გასცემს თუნდაც ერთ სტრიქონს
- გ. გასცემს ორ სტრიქონს

```

7.4.3. SELECT shemkvetiID, firmis_dasaxeleba
FROM Shemkveti AS S
WHERE qalaqi = N'თბილისი' AND
EXISTS

```

```

(
SELECT *
FROM Xelshekruleba AS X
WHERE S.shemkvetiID = X.shemkvetiID
); მოთხოვნის შესრულების შედეგად:

```

- ა. გაიცემა ქუთაისელი შემკვეთები, რომლებმაც შეკვეთა გააფორმეს
- ბ. არ გაიცემა თბილისელი შემკვეთები, რომლებმაც შეკვეთა გააფორმეს
- გ. გაიცემა თბილისელი შემკვეთები, რომლებმაც შეკვეთა გააფორმეს

7.4.4. ინფორმაციის მისაღებად თბილისელი შემკვეთების შესახებ, რომლებმაც შეკვეთა გააფორმეს უნდა შევასრულოთ მოთხოვნა:

```

ა. SELECT shemkvetiID, firmis_dasaxeleba
FROM Shemkveti AS S

```

```

WHERE qalaqi != N'თბილისი' AND
EXISTS
(
SELECT *
FROM Xelshekruleba AS X
WHERE S.shemkvetiID = X.shemkvetiID
);
ბ.      SELECT shemkvetiID, firmis_dasaxeleba
FROM Shemkveti AS S
WHERE qalaqi = N'თელავი' AND
EXISTS
(
SELECT *
FROM Xelshekruleba AS X
WHERE S.shemkvetiID = X.shemkvetiID
);
გ.      SELECT shemkvetiID, firmis_dasaxeleba
FROM Shemkveti AS S
WHERE qalaqi = N'თბილისი' AND
EXISTS
(
SELECT *
FROM Xelshekruleba AS X
WHERE S.shemkvetiID = X.shemkvetiID
);

```

თავი 8. ოპერაციები სიმრავლეებზე

შესავალი. UNION ოპერაცია

8.1.1. სტრიქონების სიმრავლეებზე ოპერაციები:

- ა. სტრიქონების ორ შესასვლელ (საწყის) სიმრავლეზე ან სტრიქონების ორ მულტისიმრავლეზე სრულდება
- ბ. სვეტების ორ შესასვლელ (საწყის) სიმრავლეზე ან სვეტების ორ მულტისიმრავლეზე სრულდება
- გ. სტრიქონების ერთ შესასვლელ (საწყის) სიმრავლეზე ან სტრიქონების ერთ მულტისიმრავლეზე სრულდება

8.1.2. სტრიქონების სიმრავლეებზე ოპერაციების შედეგი:

- ა. არ არის მულტისიმრავლე
- ბ. მულტისიმრავლეა
- გ. სვეტების სიმრავლეა

8.1.3. სტრიქონების მულტისიმრავლე:

- ა. მხოლოდ განსხვავებულ სტრიქონებს შეიცავს
- ბ. არ შეიძლება ერთნაირ სტრიქონებს შეიცავდეს
- გ. შეიძლება ერთნაირ სტრიქონებს შეიცავდეს

8.1.4. T-SQL ენაში სტრიქონების სიმრავლეებზე განსაზღვრულია:

- ა. სამი ოპერაცია: UNION, INTERSECT და EXCEPT
- ბ. ორი ოპერაცია: UNION და EXCEPT
- გ. ერთი ოპერაცია - UNION

8.1.5. სტრიქონების სიმრავლეებზე ოპერაციებში შემავალი მოთხოვნები:

- ა. არ უნდა შეიცავდეს ORDER BY განყოფილებას
- ბ. უნდა შეიცავდეს ORDER BY განყოფილებას
- გ. შეიძლება შეიცავდეს ORDER BY განყოფილებას

8.1.6. ORDER BY განყოფილების გამოყენება:

- ა. ხანდახან დასაშვებია საბოლოო შედეგის მიმართ
- ბ. დაუშვებელია საბოლოო შედეგის მიმართ
- გ. დასაშვებია მხოლოდ საბოლოო შედეგის მიმართ

8.1.7. სტრიქონების სიმრავლეებზე ოპერაციებში შემავალი ორივე მოთხოვნის მიერ გაცემულ თითოეულ შედეგობრივ ნაკრებში:

- ა. სვეტების რაოდენობა და მიმდევრობა ერთნაირი არ უნდა იყოს
- ბ. სვეტების რაოდენობა და მიმდევრობა ერთნაირი უნდა იყოს
- გ. მხოლოდ სვეტების რაოდენობა უნდა იყოს ერთნაირი

8.1.8. სტრიქონების სიმრავლეებზე ოპერაციებში შემავალი ორივე მოთხოვნის მიერ გაცემულ თითოეულ შედეგობრივ ნაკრებში:

- ა. შესაბამის სვეტებს მონაცემთა თავსებადი ტიპები არ უნდა ჰქონდეს
- ბ. შესაბამის სვეტებს მონაცემთა თავსებადი ტიპები უნდა ჰქონდეს

- გ. შესაბამის სვეტებს მონაცემთა თავსებადი ტიპები ხანდახან უნდა ჰქონდეს
- 8.1.9. სიმრავლეებზე ოპერაციის მიერ გაცემულ შედეგობრივ ნაკრებში:
- ა. სტრიქონების სახელები პირველი მოთხოვნით განისაზღვრება
 - ბ. სვეტების სახელები პირველი მოთხოვნით არ განისაზღვრება
 - გ. სვეტების სახელები პირველი მოთხოვნით განისაზღვრება
- 8.1.10. სიმრავლეებზე ოპერაციის მიერ გაცემულ შედეგობრივ ნაკრებში თუ შედეგობრივ სვეტებს უნდა მივანიჭოთ ფსევდონიმები, მაშინ ეს:
- ა. პირველ მოთხოვნაში უნდა გავაკეთოთ
 - ბ. მეორე მოთხოვნაში უნდა გავაკეთოთ
 - გ. პირველ მოთხოვნაში არ უნდა გავაკეთოთ
- 8.1.11. სტრიქონების შედარებისას სიმრავლეებზე ოპერაცია:
- ა. ორ NULL მნიშვნელობას ერთნაირად არ თვლის
 - ბ. ორ NULL მნიშვნელობას ერთნაირად თვლის
 - გ. ერთ NULL მნიშვნელობას ერთნაირად თვლის
- 8.1.12. DISTINCT ელემენტი ორი შესასვლელი მულტისიმრავლიდან ერთნაირ სტრიქონებს:
- ა. შლის და შედეგად მულტისიმრავლე გაიცემა
 - ბ. არ შლის და შედეგად სიმრავლე გაიცემა
 - გ. შლის და შედეგად სიმრავლე გაიცემა
- 8.1.13. ALL ელემენტი ორი შესასვლელი მულტისიმრავლიდან ერთნაირ სტრიქონებს:
- ა. არ შლის და შედეგად მულტისიმრავლე გაიცემა
 - ბ. შლის და შედეგად მულტისიმრავლე გაიცემა
 - გ. არ შლის და შედეგად სიმრავლე გაიცემა
- 8.1.14. DISTINCT ელემენტი უზრუნველყოფილია:
- ა. მხოლოდ UNION და EXCEPT ოპერაციებისათვის
 - ბ. UNION, INTERSECT და EXCEPT ოპერაციებისათვის
 - გ. მხოლოდ INTERSECT და EXCEPT ოპერაციებისათვის
- 8.1.15. ALL ელემენტი უზრუნველყოფილია:
- ა. მხოლოდ UNION ოპერაციისთვის
 - ბ. მხოლოდ EXCEPT ოპერაციისთვის
 - გ. მხოლოდ INTERSECT ოპერაციისთვის
- 8.1.16. DISTINCT ელემენტს სინტაქსურად:
- ა. მივუთითებთ, რადგან ის იგულისხმება მაშინ, როცა არ არის მითითებული ALL ელემენტი
 - ბ. ვერ მივუთითებთ, რადგან ის იგულისხმება მაშინ, როცა არ არის მითითებული ALL ელემენტი
 - გ. ვერ მივუთითებთ, რადგან ის არ იგულისხმება მაშინ, როცა არის მითითებული ALL ელემენტი

8.1.17. UNION ოპერაცია:

- ა. არ აერთიანებს რამდენიმე მოთხოვნის შესრულების შედეგს
- ბ. აერთიანებს რამდენიმე მოთხოვნის შესრულების შედეგს
- გ. აერთიანებს ერთი მოთხოვნის შესრულების შედეგს

8.1.18. UNION ოპერაციის შესრულების დროს:

- ა. ყველა მოთხოვნაში სვეტების რაოდენობა და მიმდევრობა ერთნაირი უნდა იყოს
- ბ. ყველა მოთხოვნაში სვეტების რაოდენობა და მიმდევრობა ერთნაირი არ უნდა იყოს
- გ. ყველა მოთხოვნაში სტრიქონების რაოდენობა და მიმდევრობა ერთნაირი უნდა იყოს

8.1.19. UNION ოპერაციის შესრულების დროს:

- ა. სტრიქონების ტიპები უნდა ემთხვეოდეს
- ბ. სვეტების ტიპები არ უნდა ემთხვეოდეს
- გ. სვეტების ტიპები უნდა ემთხვეოდეს

8.1.20. თუ ALL არგუმენტი არ არის მითითებული, მაშინ UNION ოპერაცია ავტომატურად:

- ა. არ გამორიცხავს ერთნაირ სტრიქონებს
- ბ. გამორიცხავს ერთნაირ სტრიქონებს
- გ. გამორიცხავს ერთნაირ სვეტებს

8.1.21. UNION ოპერაციის შესრულების დროს თუ გვინდა ყველა, მათ შორის ერთნაირი, სტრიქონის მიღება, მაშინ:

- ა. ALL არგუმენტი უნდა გამოვიყენოთ
- ბ. ALL არგუმენტი არ უნდა გამოვიყენოთ
- გ. ORDER BY არგუმენტი უნდა გამოვიყენოთ

8.1.22. თუ UNION ოპერაციის შესრულების დროს პირველი მოთხოვნა გასცემს 10 სტრიქონს, მეორე კი - 20-ს, მაშინ შედეგობრივ ნაკრებში აღმოჩნდება:

- ა. 10 სტრიქონი
- ბ. 200 სტრიქონი
- გ. 30 სტრიქონი

8.1.23. SELECT qalaqi, raioni FROM Personal

UNION ALL

SELECT qalaqi, raioni FROM Shemkveti;

მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა Personal და Shemkveti ცხრილების ყველა სტრიქონი
- ბ. არ გაიცემა Personal და Shemkveti ცხრილების ყველა სტრიქონი
- გ. გაიცემა მხოლოდ Personal ცხრილის ყველა სტრიქონი

8.1.24. SELECT qalaqi, raioni FROM Personal

UNION ALL

SELECT qalaqi, raioni FROM Shemkveti;

მოთხოვნის შესრულების შედეგად:

- ა. ჯერ გამოჩნდება Shemkveti ცხრილის სტრიქონები, შემდეგ კი - Personal ცხრილის

სტრიქონები

ბ. ჯერ გამოჩნდება Personali ცხრილის სტრიქონები, შემდეგ კი - Shemkveti ცხრილის სტრიქონები

გ. გაიცემა მხოლოდ Personali ცხრილის ყველა სტრიქონი

8.1.25. SELECT qalaqi, raioni FROM Personali

UNION

SELECT qalaqi, raioni FROM Shemkveti;

მოთხოვნის შესრულების შედეგად:

ა. გაიცემა თანამშრომლებისა და შემკვეთების ერთნაირი ადგილმდებარეობები

ბ. არ გაიცემა თანამშრომლებისა და შემკვეთების ერთმანეთისაგან განსხვავებული ადგილმდებარეობები

გ. გაიცემა თანამშრომლებისა და შემკვეთების ერთმანეთისაგან განსხვავებული ადგილმდებარეობები

8.1.26. იმისათვის, რომ მივიღოთ თანამშრომლებისა და შემკვეთების ერთმანეთისაგან განსხვავებული ადგილმდებარეობები, უნდა შევასრულოთ მოთხოვნა:

ა. SELECT qalaqi, raioni FROM Personali

INTERSECT

SELECT qalaqi, raioni FROM Shemkveti;

ბ. SELECT qalaqi, raioni FROM Personali

UNION ALL

SELECT qalaqi, raioni FROM Shemkveti;

გ. SELECT qalaqi, raioni FROM Personali

UNION

SELECT qalaqi, raioni FROM Shemkveti;

INTERSECT ოპერაცია

8.2.1. ორი სიმრავლის გადაკვეთა:

ა. არის ყველა იმ ელემენტების ერთობლიობა, რომლებიც ორივე სიმრავლეს ეკუთვნის

ბ. არ არის ყველა იმ ელემენტების ერთობლიობა, რომლებიც ორივე სიმრავლეს ეკუთვნის

გ. არის ყველა იმ ელემენტების ერთობლიობა, რომლებიც ორივე სიმრავლეს არ ეკუთვნის

8.2.2. INTERSECT ოპერაცია გასცემს მხოლოდ იმ სტრიქონებს, რომლებიც:

ა. გვხვდება ორივე შესასვლელ ნაკრებში

ბ. არ გვხვდება ორივე შესასვლელ ნაკრებში

გ. გვხვდება მხოლოდ პირველ შესასვლელ ნაკრებში

8.2.3. INTERSECT ოპერაცია:

ა. ჯერ ერთნაირ სტრიქონებს წაშლის ორივე შესასვლელი მულტისიმრავლიდან, შემდეგ კი გასცემს მხოლოდ იმ სტრიქონებს, რომლებიც არ გვხვდება ორივე სიმრავლეში

ბ. ჯერ ერთნაირ სტრიქონებს წაშლის ორივე შესასვლელი მულტისიმრავლიდან, შემდეგ კი გასცემს მხოლოდ იმ სტრიქონებს, რომლებიც გვხვდება ორივე სიმრავლეში

გ. ჯერ ერთნაირ სტრიქონებს არ წაშლის ორივე შესასვლელი მულტისიმრავლიდან, შემდეგ კი გასცემს მხოლოდ იმ სტრიქონებს, რომლებიც გვხვდება ორივე სიმრავლეში

8.2.4. INTERSECT ოპერაციის შესრულების დროს სტრიქონი გაიცემა იმ შემთხვევაში, თუ ის:

- ა. სამჯერ მაინც გვხვდება ორივე მულტისიმრავლეში
- ბ. ორჯერ მაინც გვხვდება ორივე მულტისიმრავლეში
- გ. ერთხელ მაინც გვხვდება ორივე მულტისიმრავლეში

8.2.5. სტრიქონების სიმრავლეებზე ოპერაცია ორ NULL მნიშვნელობას:

- ა. განიხილავს როგორც ტოლს და არ გასცემს შესაბამის სტრიქონს
- ბ. არ განიხილავს როგორც ტოლს და გასცემს შესაბამის სტრიქონს
- გ. განიხილავს როგორც ტოლს და გასცემს შესაბამის სტრიქონს

8.2.6. SELECT qalaqi, raioni FROM Personal

INTERSECT

SELECT qalaqi, raioni FROM Shemkveti;

მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა თანამშრომლებისა და შემკვეთების ერთმანეთისაგან განსხვავებულ საერთო ადგილმდებარეობა
- ბ. არ გაიცემა თანამშრომლებისა და შემკვეთების ერთმანეთისაგან განსხვავებულ საერთო ადგილმდებარეობა
- გ. გაიცემა თანამშრომლებისა და შემკვეთების ერთმანეთისაგან განსხვავებული ადგილმდებარეობა

8.2.7. იმისათვის, რომ მივიღოთ თანამშრომლებისა და შემკვეთების ერთმანეთისაგან განსხვავებული საერთო ადგილმდებარეობა, უნდა შევასრულოთ მოთხოვნა:

ა. SELECT qalaqi, raioni FROM Personal

UNION

SELECT qalaqi, raioni FROM Shemkveti;

ბ. SELECT qalaqi, raioni FROM Personal

INTERSECT

SELECT qalaqi, raioni FROM Shemkveti;

გ. SELECT qalaqi, raioni FROM Personal

UNION ALL

SELECT qalaqi, raioni FROM Shemkveti;

EXCEPT ოპერაცია

8.3.1. A და B სიმრავლეების სხვაობა არის იმ ელემენტების სიმრავლე, რომლებიც:

- ა. არ ეკუთვნის A-ს და ეკუთვნის B-ს
- ბ. ეკუთვნის A-ს და არ ეკუთვნის B-ს
- გ. არ ეკუთვნის A-ს და არ ეკუთვნის B-ს

8.3.2. სიმრავლეების სხვაობა შეგვიძლია განვიხილოთ როგორც:

- ა. B სიმრავლე A სიმრავლის ელემენტების გარეშე
- ბ. A სიმრავლე B სიმრავლის ელემენტების გარეშე
- გ. B სიმრავლე B სიმრავლის ელემენტების გარეშე

8.3.3. EXCEPT ოპერაცია ზემოქმედებს ორი მოთხოვნის მიერ გაცემული სტრიქონების შედეგობრივ ნაკრებზე და:

- ა. გასცემს იმ სტრიქონებს, რომლებიც გვხვდება პირველ ნაკრებში, მაგრამ არ გვხვდება მეორეში
- ბ. არ გასცემს იმ სტრიქონებს, რომლებიც გვხვდება პირველ ნაკრებში, მაგრამ არ გვხვდება მეორეში
- გ. გასცემს იმ სტრიქონებს, რომლებიც არ გვხვდება პირველ ნაკრებში, მაგრამ გვხვდება მეორეში

8.3.4. EXCEPT ოპერაცია:

- ა. სიმეტრიულია, ე.ი. ამ ოპერაციის შესრულების დროს მნიშვნელობა აქვს თუ რომელი მოთხოვნაა პირველი და რომელი მეორე
- ბ. ასიმეტრიულია, ე.ი. ამ ოპერაციის შესრულების დროს მნიშვნელობა არ აქვს თუ რომელი მოთხოვნაა პირველი და რომელი მეორე
- გ. ასიმეტრიულია, ე.ი. ამ ოპერაციის შესრულების დროს მნიშვნელობა აქვს თუ რომელი მოთხოვნაა პირველი და რომელი მეორე

8.3.5. UNION და INTERSECT ოპერაციები:

- ა. სიმეტრიულია, ე.ი. ამ ოპერაციის შესრულების დროს მნიშვნელობა არ აქვს თუ რომელი მოთხოვნაა პირველი და რომელი მეორე
- ბ. სიმეტრიულია, ე.ი. ამ ოპერაციის შესრულების დროს მნიშვნელობა აქვს თუ რომელი მოთხოვნაა პირველი და რომელი მეორე
- გ. ასიმეტრიულია, ე.ი. ამ ოპერაციის შესრულების დროს მნიშვნელობა არ აქვს თუ რომელი მოთხოვნაა პირველი და რომელი მეორე

8.3.6. SELECT qalaqi, raioni FROM Personal
EXCEPT

SELECT qalaqi, raioni FROM Shemkveti;

მოთხოვნის შესრულების შედეგად:

- ა. არ გაიცემა თანამშრომლების სხვადასხვა ადგილმდებარეობა, რომელიც არ ემთხვევა შემკვეთების ადგილმდებარეობას
- ბ. გაიცემა თანამშრომლების სხვადასხვა ადგილმდებარეობა, რომელიც არ ემთხვევა შემკვეთების ადგილმდებარეობას
- გ. გაიცემა თანამშრომლების სხვადასხვა ადგილმდებარეობა, რომელიც ემთხვევა შემკვეთების ადგილმდებარეობას

8.3.7. თანამშრომლების სხვადასხვა ადგილმდებარეობის სანახავად, რომელიც არ ემთხვევა შემკვეთების ადგილმდებარეობას, უნდა შევასრლოთ მოთხოვნა:

ა. SELECT qalaqi, raioni FROM Personal

UNION ALL

SELECT qalaqi, raioni FROM Shemkveti;

ბ. SELECT qalaqi, raioni FROM Personal

UNION

SELECT qalaqi, raioni FROM Shemkveti;

გ. SELECT qalaqi, raioni FROM Personal

EXCEPT

SELECT qalaqi, raioni FROM Shemkveti;

პრიორიტეტი

8.4.1. სიმრავლეებზე ოპერაციებისთვის:

- ა. განუსაზღვრელია პრიორიტეტები
- ბ. განსაზღვრულია პრიორიტეტები
- გ. ნახევრადაა განსაზღვრული პრიორიტეტები

8.4.2. INTERSECT ოპერაციას:

- ა. ისეთივე პრიორიტეტი აქვს, როგორც UNION და EXCEPT ოპერაციებს
- ბ. უფრო დაბალი პრიორიტეტი აქვს ვიდრე UNION და EXCEPT ოპერაციებს
- გ. უფრო მაღალი პრიორიტეტი აქვს ვიდრე UNION და EXCEPT ოპერაციებს

8.4.3. UNION და EXCEPT ოპერაციებს:

- ა. ერთნაირი პრიორიტეტი ხანდახან აქვს
- ბ. ერთნაირი პრიორიტეტი არ აქვს
- გ. ერთნაირი პრიორიტეტი აქვს

8.4.4. სიმრავლეებზე ოპერაციების შესრულების მართვისთვის, მაგალითად პრიორიტეტის შესაცვლელად, გამოიყენება:

- ა. მრგვალი ფრჩხილები
- ბ. კვადრატული ფრჩხილები
- გ. ფიგურული ფრჩხილები

თავი 9. ფუნქციები

მომხმარებლის ფუნქციები

9.1.1. მომხმარებლის მიერ შემუშავებული ფუნქციები:

- ა. წარმოადგენენ მონაცემთა ბაზის დამოუკიდებელ ობიექტებს და მოთავსებული არიან შესაბამის მონაცემთა ბაზაში
- ბ. არ წარმოადგენენ მონაცემთა ბაზის დამოუკიდებელ ობიექტებს და მოთავსებული არიან შესაბამის მონაცემთა ბაზაში
- გ. წარმოადგენენ მონაცემთა ბაზის დამოუკიდებელ ობიექტებს და მოთავსებული არ არიან შესაბამის მონაცემთა ბაზაში

9.1.2. არსებობს მომხმარებლების მიერ შემუშავებული ფუნქციების:

- ა. ორი კლასი
- ბ. სამი კლასი
- გ. ერთი კლასი

9.1.3. არსებობს მომხმარებლების მიერ შემუშავებული ფუნქციების შემდეგი ტიპები:

- ა. Scalar, Multi-statement და table
- ბ. Scalar, Inline და cursor
- გ. Scalar, Inline და Multi-statement

9.1.4. Scalar ტიპის ფუნქციები გასცემენ ნებისმიერი ტიპის:

- ა. არასკალარულ მნიშვნელობას, გარდა timestamp (rowversion), text, ntext, image, table და cursor ტიპებისა
- ბ. სკალარულ მნიშვნელობას, გარდა timestamp (rowversion), text, ntext, image, table და cursor ტიპებისა
- გ. სკალარულ მნიშვნელობას, timestamp (rowversion), text, ntext, image, table და cursor ტიპების ჩათვლით

9.1.5. Scalar ტიპის ფუნქცია შეიძლება შეიცავდეს:

- ა. ერთ ან მეტ ბრძანებას, რომელიც BEGIN...END ბლოკში უნდა იყოს მოთავსებული
- ბ. ერთ ან მეტ ბრძანებას, რომელიც BEGIN...END ბლოკში არ უნდა იყოს მოთავსებული
- გ. მხოლოდ ორ ბრძანებას, რომელიც BEGIN...END ბლოკში არ უნდა იყოს მოთავსებული

9.1.6. Inline ტიპის ფუნქციები შეიცავენ მხოლოდ:

- ა. ერთ UPDATE ბრძანებას
- ბ. ორ SELECT ბრძანებას
- გ. ერთ SELECT ბრძანებას

9.1.7. მისი საშუალებით ფორმირდება მონაცემთა ნაკრები, რომელიც გაიცემა:

- ა. SELECT ტიპის მნიშვნელობის სახით
- ბ. table ტიპის მნიშვნელობის სახით
- გ. UPDATE ტიპის მნიშვნელობის სახით

9.1.8. Multi-statement ტიპის ფუნქციები გასცემენ:

- ა. UPDATE ტიპის მნიშვნელობას

- ბ. table ტიპის მნიშვნელობას
- გ. SELECT ტიპის მნიშვნელობას

9.1.9. Multi-statement ფუნქციის ტანი შეიძლება:

- ა. რამდენიმე ბრძანებას შეიცავდეს
- ბ. არ შეიცავდეს არც ერთ ბრძანებას
- გ. შეიცავდეს მხოლოდ ერთ ბრძანებას

9.1.10. ფუნქციებს შეიძლება ჰქონდეთ:

- ა. მხოლოდ ერთი პარამეტრი
- ბ. ერთი ან მეტი პარამეტრი ან არც ერთი
- გ. არც ერთი პარამეტრი

9.1.11. ფუნქციის პარამეტრის ტიპი:

- ა. არ უნდა იყოს timestamp (rowversion), cursor და table
- ბ. უნდა იყოს timestamp (rowversion), cursor და table
- გ. არ უნდა იყოს int და table

9.1.12. ფუნქციის ტანში:

- ა. დაუშვებელია ლოკალური ცვლადების გამოცხადება, სხვა ფუნქციებისა და შენახული პროცედურების გამოძახება და კურსორების შექმნა
- ბ. დასაშვებია მხოლოდ ლოკალური ცვლადების გამოცხადება
- გ. დასაშვებია ლოკალური ცვლადების გამოცხადება, სხვა ფუნქციებისა და შენახული პროცედურების გამოძახება და კურსორების შექმნა

9.1.13. ფუნქციის კოდში INSERT, UPDATE და DELETE ბრძანებების გამოყენებისას:

- ა. დასაშვებია მუშაობა მხოლოდ იმ მონაცემებთან, რომელიც ინახება ფუნქციის ტანში შექმნილი int ტიპის ცვლადში
- ბ. დასაშვებია მუშაობა მხოლოდ იმ მონაცემებთან, რომელიც ინახება ფუნქციის ტანში შექმნილი table ტიპის ცვლადში
- გ. დაუშვებელია მუშაობა მხოლოდ იმ მონაცემებთან, რომელიც ინახება ფუნქციის ტანში შექმნილი table ტიპის ცვლადში

9.1.14. ფუნქციის ტანში:

- ა. დაუშვებელია PRINT და SELECT ბრძანებების გამოყენება მონაცემების უშუალოდ დაბრუნების მიზნით
- ბ. დასაშვებია PRINT და SELECT ბრძანებების გამოყენება მონაცემების უშუალოდ დაბრუნების მიზნით
- გ. დაუშვებელია RETURN ბრძანებების გამოყენება მონაცემების უშუალოდ დაბრუნების მიზნით

Scalar ტიპის ფუნქციები

9.2.1. CREATE FUNCTION ბრძანებაში "@პარამეტრის_სახელი მონაცემთა_ტიპი [=DEFAULT]" აგრუმენტი:

- ა. მხოლოდ მთელ რიცხვებს განსაზღვრავს

- ბ. ფუნქციის პარამეტრებს არ განსაზღვრავს
 - გ. ფუნქციის პარამეტრებს განსაზღვრავს
- 9.2.2. CREATE FUNCTION ბრძანებაში თითოეული პარამეტრის სახელი:
- ა. არ უნდა იყოს უნიკალური და იწყებოდეს @ სიმბოლოთი
 - ბ. უნდა იყოს უნიკალური და იწყებოდეს @ სიმბოლოთი
 - გ. უნდა იყოს უნიკალური და არ უნდა იწყებოდეს @ სიმბოლოთი
- 9.2.3. CREATE FUNCTION ბრძანებაში:
- ა. ჯერ ეთითება პარამეტრის ტიპი, შემდეგ კი პარამეტრის სახელი
 - ბ. ჯერ ეთითება პარამეტრის სახელი, შემდეგ კი პარამეტრის ტიპი
 - გ. არ ეთითება პარამეტრის სახელი და არც პარამეტრის ტიპი
- 9.2.4. CREATE FUNCTION ბრძანებაში პარამეტრისთვის:
- ა. დასაშვებია ყველა ტიპი, გარდა timestamp (rowversion), cursor და table ტიპებისა
 - ბ. დაუშვებელია ყველა ტიპი, გარდა timestamp (rowversion), cursor და table ტიპებისა
 - გ. დასაშვებია timestamp (rowversion), cursor და table ტიპები
- 9.2.5. DEFAULT აგრუმენტი მიუთითებს იმ მნიშვნელობას, რომელიც:
- ა. ავტომატურად არ მიენიჭება პარამეტრს, თუ მისი მნიშვნელობა აშკარად არ იქნება მითითებული
 - ბ. ავტომატურად მიენიჭება პარამეტრს, თუ მისი მნიშვნელობა აშკარად არ იქნება მითითებული
 - გ. ავტომატურად მიენიჭება პარამეტრს, თუ მისი მნიშვნელობა აშკარად იქნება მითითებული
- 9.2.6. პარამეტრები ერთმანეთისაგან:
- ა. წერტილებით გამოიყოფა
 - ბ. მძიმეებით გამოიყოფა
 - გ. წერტილ-მძიმეებით გამოიყოფა
- 9.2.7. ENCRYPTION საკვანძო სიტყვის გამოყენება:
- ა. არ იწვევს CREATE FUNCTION ბრძანების კოდის დაშიფვრას
 - ბ. იწვევს CREATE FUNCTION ბრძანების კოდის დაშიფვრას
 - გ. იწვევს CREATE FUNCTION ბრძანების კოდის გაშიფვრას
- 9.2.8. CREATE FUNCTION ბრძანების კოდი:
- ა. ინახება მიმდინარე მონაცემთა ბაზის syscomments სისტემური წარმოდგენის text სვეტში
 - ბ. არ ინახება მიმდინარე მონაცემთა ბაზის syscomments სისტემური წარმოდგენის text სვეტში
 - გ. ინახება მიმდინარე მონაცემთა ბაზის systables სისტემური წარმოდგენის text სვეტში
- 9.2.9. თუ ფუნქციის კოდი დაშიფრული არ არის, მაშინ მისი ნახვა:
- ა. UPDATE ბრძანებით შეიძლება
 - ბ. INSERT ბრძანებით შეიძლება
 - გ. SELECT ბრძანებით შეიძლება

9.2.10. ფუნქცია შეიძლება მიმართავდეს მონაცემთა ბაზის სხვადასხვა ობიექტებს. ამ ობიექტების შეცვლა ან წაშლა გამოიწვევს ფუნქციის შესრულების შეფერხებას. ამის თავიდან ასაცილებლად:

- ა. ENCRYPTION საკვანძო სიტყვა გამოიყენება
- ბ. SCHEMABINDING საკვანძო სიტყვა გამოიყენება
- გ. UPDATE ბრძანება გამოიყენება

9.2.11. ფუნქცია შეიძლება მიმართავდეს მონაცემთა ბაზის სხვადასხვა ობიექტებს. ამ ობიექტების შეცვლა ან წაშლა:

- ა. გამოიწვევს ფუნქციის შესრულების შეფერხებას
- ბ. გამოიწვევს ფუნქციის შესრულების დაწყებას
- გ. არ გამოიწვევს ფუნქციის შესრულების შეფერხებას

9.2.12. BEGIN...END საკვანძო სიტყვებს შორის:

- ა. არ არის მოთავსებული ფუნქციის ტანი (კოდი)
- ბ. მოთავსებულია ფუნქციის ტანი (კოდი)
- გ. მოთავსებულია UPDATE ბრძანების კოდი

9.2.13. 'RETURN სკალარული_გამოსახულება' ფუნქციის მუშაობას:

- ა. არ ამთავრებს და შედეგს გასცემს
- ბ. ამთავრებს და შედეგს არ გასცემს
- გ. ამთავრებს და შედეგს გასცემს

9.2.14. CREATE FUNCTION Maqsimaluri_Xelfasi(@ganyofileba NVARCHAR(30))

RETURNS FLOAT

AS

BEGIN

DECLARE @Max_xelfasi FLOAT;

SET @Max_xelfasi =

(

SELECT MAX(xelfasi)

FROM Personali

WHERE ganyofileba = @ganyofileba

);

RETURN @Max_xelfasi;

END

პროგრამული კოდის შესრულების შედეგად:

- ა. შეიქმნება Scalar ტიპის ფუნქცია, რომელიც გამოთვლის და გასცემს მითითებული განყოფილების მაქსიმალურ ხელფასს
- ბ. არ შეიქმნება Scalar ტიპის ფუნქცია, რომელიც გამოთვლის და გასცემს მითითებული განყოფილების მაქსიმალურ ხელფასს
- გ. შეიქმნება Scalar ტიპის ფუნქცია, რომელიც არ გამოთვლის და არ გასცემს მითითებული განყოფილების მაქსიმალურ ხელფასს

9.2.15. CREATE FUNCTION Maqsimaluri_Xelfasi(@ganyofileba NVARCHAR(30)) ბრძანებით იქმნება Scalar ტიპის ფუნქცია. ის შემდეგნაირად უნდა გამოვიძახოთ:

- ა. SELECT dbo.Maqsimaluri_Xelfasi();
- ბ. SELECT dbo.Maqsimaluri_Xelfasi(N'სამედიცინო');
- გ. SELECT dbo.Maqsimaluri_Xelfasi(N'სამედიცინო') FROM Personali;

Inline ტიპის ფუნქციები

9.3.1. Inline ტიპის ფუნქცია:

- ა. აბრუნებს table ტიპის მნიშვნელობას
- ბ. არ აბრუნებს table ტიპის მნიშვნელობას
- გ. აბრუნებს int ტიპის მნიშვნელობას

9.3.2. Inline ტიპის ფუნქცია აბრუნებს table ტიპის მნიშვნელობას, რომელსაც:

- ა. არ უნდა ჰქონდეს SELECT მოთხოვნის მიერ გაცემული შედეგის სტრუქტურა
- ბ. უნდა ჰქონდეს UPDATE ბრძანების მიერ გაცემული შედეგის სტრუქტურა
- გ. უნდა ჰქონდეს SELECT მოთხოვნის მიერ გაცემული შედეგის სტრუქტურა

9.3.3. Inline ფუნქციის თავისებურებაა ის, რომ:

- ა. view მნიშვნელობის სტრუქტურა სტატიკურად იქმნება მოთხოვნის შესრულების დროს
- ბ. table მნიშვნელობის სტრუქტურა დინამიურად არ იქმნება მოთხოვნის შესრულების დროს
- გ. table მნიშვნელობის სტრუქტურა დინამიურად იქმნება მოთხოვნის შესრულების დროს

9.3.4. გაცემული table მნიშვნელობა შეგვიძლია უშუალოდ გამოვიყენოთ SELECT მოთხოვნის:

- ა. WHERE განყოფილებაში ფუნქციის სახელის მითითების გზით
- ბ. FROM განყოფილებაში ფუნქციის სახელის მითითების გზით
- გ. GROUP BY განყოფილებაში ფუნქციის სახელის მითითების გზით

```
9.3.5. CREATE FUNCTION Iuridiuli_Pirebi1(@qalaqi NVARCHAR(30), @iuridiuli_fizikuri
NVARCHAR(30))
RETURNS TABLE
AS
RETURN
SELECT *
```

FROM Shemkveti

WHERE iuridiuli_fizikuri = @iuridiuli_fizikuri AND qalaqi = @qalaqi;

პროგრამული კოდის შესრულების შედეგად:

- ა. შეიქმნება Inline ტიპის ფუნქცია. ის არ გასცემს ინფორმაციას მითითებულ ქალაქში მყოფი შემკვეთების შესახებ
- ბ. არ შეიქმნება Inline ტიპის ფუნქცია. ის გასცემს ინფორმაციას მითითებულ ქალაქში მყოფი შემკვეთების შესახებ
- გ. შეიქმნება Inline ტიპის ფუნქცია. ის გასცემს ინფორმაციას მითითებულ ქალაქში მყოფი შემკვეთების შესახებ

```
9.3.6. CREATE FUNCTION Iuridiuli_Pirebi1(@qalaqi NVARCHAR(30), @iuridiuli_fizikuri
NVARCHAR(30)) ბრძანება ქმნის Inline ტიპის ფუნქციას. მისი გამოძახება ხდება შემდეგნაირად:
```

```
ა. SELECT iuridiuli_fizikuri, gvari, qalaqi
```


- FROM Iuridiuli_Pirebi1(N'თბილისი', N'იურიდიული');
- ბ. SELECT Iuridiuli_Pirebi1(N'თბილისი', N'იურიდიული');
- გ. SELECT iuridiuli_fizikuri, gvari, qalaqi
FROM Iuridiuli_Pirebi1(N'თბილისი');

Multi-statement ფუნქციები

9.4.1. Multi-statement ფუნქციაში TABLE სიტყვის შემდეგ:

- ა. აშკარად განისაზღვრება დასაბრუნებელი მნიშვნელობის სტრუქტურა
- ბ. აშკარად არ განისაზღვრება დასაბრუნებელი მნიშვნელობის სტრუქტურა
- გ. არააშკარად განისაზღვრება დასაბრუნებელი მნიშვნელობის სტრუქტურა

9.4.2. Multi-statement ფუნქციის ტანში:

- ა. დაუმეგბელია Transact_SQL-ის ნებისმიერი კონსტრუქციის გამოყენება
- ბ. დასაშვებია Transact_SQL-ის ნებისმიერი კონსტრუქციის გამოყენება
- გ. დასაშვებია მხოლოდ BEGIN...END კონსტრუქციის გამოყენება

9.4.3. Multi-statement ფუნქციის ტანში მონაცემთა დასაბრუნებელი ნაკრები:

- ა. უნდა შეივსოს INSERT ბრძანებით
- ბ. არ უნდა შეივსოს INSERT ბრძანებით
- გ. უნდა შეივსოს UPDATE ბრძანებით

9.4.4. Multi-statement ფუნქციის განსაზღვრისას table ტიპის მნიშვნელობას:

- ა. არ უნდა მიენიჭოს სახელი
- ბ. აუცილებლად უნდა მიენიჭოს სახელი
- გ. ხანდახან შეიძლება მიენიჭოს სახელი

9.4.5. Multi_statement ფუნქციაში RETURN სიტყვის შემდეგ:

- ა. არ არის აუცილებელი საწყისი მნიშვნელობის მითითება
- ბ. არის აუცილებელი დასაბრუნებელი მნიშვნელობის მითითება
- გ. არ არის აუცილებელი დასაბრუნებელი მნიშვნელობის მითითება

9.4.6. Multi_statement ფუნქციაში RETURN სიტყვის შემდეგ არ არის აუცილებელი დასაბრუნებელი მნიშვნელობის მითითება. ამ შემთხვევაში სერვერი:

- ა. ავტომატურად დააბრუნებს int ტიპის მონაცემს
- ბ. ავტომატურად არ დააბრუნებს table ტიპის მონაცემთა ნაკრებს
- გ. ავტომატურად დააბრუნებს table ტიპის მონაცემთა ნაკრებს

9.4.7. Multi_statement ფუნქცია გასცემს table ტიპის მონაცემთა ნაკრებს, რომლის სახელი და სტრუქტურა მითითებული იყო:

- ა. BEGIN სიტყვის შემდეგ
- ბ. RETURNS სიტყვის შემდეგ
- გ. RETURN სიტყვის შემდეგ

ფუნქციის წაშლა

9.5.1. ფუნქციის წასაშლელად გამოიყენება ბრძანება:

- ა. DROP FUNCTION
- ბ. ALTER FUNCTION
- გ. CREATE FUNCTION

9.5.2. DROP FUNCTION ბრძანებით:

- ა. შეუძლებელია სამივე ტიპის ფუნქციის წაშლა
- ბ. შესაძლებელია მხოლოდ ერთი ტიპის ფუნქციის წაშლა
- გ. შესაძლებელია სამივე ტიპის ფუნქციის წაშლა

9.5.3. თუ ვშლით ფუნქციას, რომელიც სხვა მომხმარებელს ან მონაცემთა ბაზის სხვა მფლობელს ეკუთვნის, მაშინ:

- ა. არ უნდა მივუთითოთ ფუნქციის მფლობელის სახელი
- ბ. უნდა მივუთითოთ ფუნქციის მფლობელის სახელი
- გ. უნდა მივუთითოთ ფუნქციის მფლობელის იდენტიფიკატორი

9.5.4. ერთი DROP FUNCTION ბრძანებით:

- ა. არ შეიძლება რამდენიმე ფუნქციის წაშლა
- ბ. შეიძლება რამდენიმე ფუნქციის წაშლა
- გ. შეიძლება მხოლოდ ერთი ფუნქციის წაშლა

ჩადგმული ფუნქციები. მათემატიკის ფუნქციები

9.6.1. ABS(რიცხვითი_გამოსახულება) ფუნქცია:

- ა. გასცემს რიცხვის აბსოლუტურ მნიშვნელობას
- ბ. გასცემს 1-ს, თუ რიცხვი დადებითია, 0-ს თუ რიცხვი 0-ის ტოლია და -1-ს თუ რიცხვი უარყოფითია
- გ. მითითებული არგუმენტისთვის გასცემს უახლოეს მინიმალურ მთელ რიცხვს, ანუ ასრულებს დამრგვალებას ქვევით

9.6.2. ISNUMERIC(გამოსახულება) ფუნქცია:

- ა. გასცემს 1-ს, თუ რიცხვი დადებითია, 0-ს თუ რიცხვი 0-ის ტოლია და -1-ს თუ რიცხვი უარყოფითია
- ბ. გასცემს რიცხვის აბსოლუტურ მნიშვნელობას
- გ. გასცემს 1-ს, თუ არგუმენტს რიცხვითი ტიპი აქვს, წინააღმდეგ შემთხვევაში - 0-ს

9.6.3. SIGN(რიცხვითი_გამოსახულება) ფუნქცია:

- ა. მითითებული არგუმენტისთვის გასცემს უახლოეს მინიმალურ მთელ რიცხვს, ანუ ასრულებს დამრგვალებას ქვევით
- ბ. გასცემს 1-ს, თუ რიცხვი დადებითია, 0-ს თუ რიცხვი 0-ის ტოლია და -1-ს თუ რიცხვი უარყოფითია
- გ. გასცემს 1-ს, თუ არგუმენტს რიცხვითი ტიპი აქვს, წინააღმდეგ შემთხვევაში - 0-ს

9.6.4. RAND([რიცხვი]) ფუნქცია:

- ა. მითითებული არგუმენტისთვის გასცემს უახლოეს მინიმალურ მთელ რიცხვს, ანუ ასრულებს დამრგვალებას ქვევით
- ბ. გასცემს შემთხვევით წილად რიცხვს 0÷1 დიაპაზონში. არგუმენტს უნდა ჰქონდეს tinyint, int ან smallint ტიპი. თუ არგუმენტი არ არის მითითებული, მაშინ შემთხვევითი რიცხვის გენერირება მოხდება სისტემური დროის საფუძველზე
- გ. გასცემს უახლოეს მთელ რიცხვს, რომელიც არგუმენტზე მეტი ან ტოლია, ანუ ასრულებს დამრგვალებას ზევით

9.6.5. FLOOR(რიცხვითი_გამოსახულება) ფუნქცია:

- ა. მითითებული არგუმენტისთვის გასცემს უახლოეს მინიმალურ მთელ რიცხვს, ანუ ასრულებს დამრგვალებას ქვევით
- ბ. გასცემს უახლოეს მთელ რიცხვს, რომელიც არგუმენტზე მეტი ან ტოლია, ანუ ასრულებს დამრგვალებას ზევით
- გ. ასრულებს 'რიცხვითი_გამოსახულება' დამრგვალებას მითითებული სიზუსტით. დამრგვალება შეიძლება შესრულდეს როგორც ათობითი წერტილის შემდეგ (სიგრძე > 0), ისე ათობით წერტილამდე (სიგრძე < 0)

9.6.6. CEILING(რიცხვითი_გამოსახულება) ფუნქცია:

- ა. ასრულებს 'რიცხვითი_გამოსახულება' დამრგვალებას მითითებული სიზუსტით. დამრგვალება შეიძლება შესრულდეს როგორც ათობითი წერტილის შემდეგ (სიგრძე > 0), ისე ათობით წერტილამდე (სიგრძე < 0)
- ბ. გასცემს უახლოეს მთელ რიცხვს, რომელიც არგუმენტზე მეტი ან ტოლია, ანუ ასრულებს დამრგვალებას ზევით
- გ. მითითებული არგუმენტისთვის გასცემს უახლოეს მინიმალურ მთელ რიცხვს, ანუ ასრულებს დამრგვალებას ქვევით

9.6.7. ROUND(რიცხვითი_გამოსახულება, სიგრძე [, ფუნქცია]) ფუნქცია:

- ა. გასცემს უახლოეს მთელ რიცხვს, რომელიც არგუმენტზე მეტი ან ტოლია, ანუ ასრულებს დამრგვალებას ზევით
- ბ. მითითებული არგუმენტისთვის გასცემს უახლოეს მინიმალურ მთელ რიცხვს, ანუ ასრულებს დამრგვალებას ქვევით
- გ. ასრულებს 'რიცხვითი_გამოსახულება' დამრგვალებას მითითებული სიზუსტით. დამრგვალება შეიძლება შესრულდეს როგორც ათობითი წერტილის შემდეგ (სიგრძე > 0), ისე ათობით წერტილამდე (სიგრძე < 0)

9.6.8. POWER(რიცხვითი_გამოსახულება, ხარისხი) ფუნქცია:

- ა. გასცემს რიცხვის ხარისხს
- ბ. გასცემს არგუმენტის კვადრატს
- გ. არგუმენტიდან იღებს კვადრატულ ფესვს

9.6.9. SQUARE(წილადი_გამოსახულება) ფუნქცია:

- ა. გასცემს არგუმენტის კვადრატს
- ბ. გასცემს π მნიშვნელობას (3.14)
- გ. გასცემს რიცხვის ხარისხს

9.6.10. SQRT(წილადი_გამოსახულება) ფუნქცია:

- ა. გასცემს რიცხვის ხარისხს
- ბ. არგუმენტიდან იღებს კვადრატულ ფესვს
- გ. გასცემს π მნიშვნელობას (3.14)

9.6.11. PI():

- ა. გასცემს π მნიშვნელობას (3.14)
- ბ. გასცემს არგუმენტის კვადრატს
- გ. ასრულებს კუთხის გარდაქმნას რადიანიდან გრადუსში

9.6.12. DEGREES(რიცხვითი_გამოსახულება) ფუნქცია:

- ა. ასრულებს კუთხის გარდაქმნას გრადუსიდან რადიანში
- ბ. რადიანებში გასცემს მითითებული კუთხის სინუსს
- გ. ასრულებს კუთხის გარდაქმნას რადიანიდან გრადუსში

9.6.13. RADIANS(რიცხვითი_გამოსახულება) ფუნქცია:

- ა. რადიანებში გასცემს მითითებული კუთხის სინუსს
- ბ. ასრულებს კუთხის გარდაქმნას რადიანიდან გრადუსში
- გ. ასრულებს კუთხის გარდაქმნას გრადუსიდან რადიანში

9.6.14. SIN(წილადი_გამოსახულება) ფუნქცია:

- ა. გასცემს არგუმენტის ექსპონენტას
- ბ. რადიანებში გასცემს მითითებული კუთხის სინუსს
- გ. რადიანებში გასცემს მითითებული კუთხის ტანგენსს

9.6.15. COS(წილადი_გამოსახულება) ფუნქცია:

- ა. რადიანებში გასცემს მითითებული კუთხის ტანგენსს
- ბ. რადიანებში გასცემს მითითებული კუთხის კოსინუსს
- გ. რადიანებში გასცემს მითითებული კუთხის კოტანგენსს

9.6.16. TAN(წილადი_გამოსახულება) ფუნქცია:

- ა. რადიანებში გასცემს მითითებული კუთხის ტანგენსს
- ბ. გასცემს არგუმენტის ექსპონენტას
- გ. რადიანებში გასცემს მითითებული კუთხის კოტანგენსს

9.6.17. COT(წილადი_გამოსახულება) ფუნქცია:

- ა. გასცემს არგუმენტის ექსპონენტას
- ბ. გასცემს არგუმენტის ათობით ლოგარითმს
- გ. რადიანებში გასცემს მითითებული კუთხის კოტანგენსს

9.6.18. EXP(წილადი_გამოსახულება) ფუნქცია:

- ა. გასცემს არგუმენტის ნატურალურ ლოგარითმს
- ბ. გასცემს არგუმენტის ათობით ლოგარითმს
- გ. გასცემს არგუმენტის ექსპონენტას

9.6.19. LOG(წილადი_გამოსახულება) ფუნქცია:

- ა. რადიანებში გასცემს მითითებული კუთხის ტანგენსს

- ბ. გასცემს არგუმენტის ნატურალურ ლოგარითმს
- გ. გასცემს არგუმენტის ათობით ლოგარითმს

9.6.20. LOG10(წილადი_გამოსახულება) ფუნქცია:

- ა. გასცემს არგუმენტის ათობით ლოგარითმს
- ბ. რადიანებში გასცემს მითითებული კუთხის ტანგენსს
- გ. გასცემს არგუმენტის ნატურალურ ლოგარითმს

9.6.21. SELECT POWER(5,2) AS [რიცხვის ხარისხი],
SQRT(25) AS [კვადრატული ფესვი რიცხვიდან];

მოთხოვნის შესრულების შედეგად:

- ა. არ გაიცემა 25 და 5
- ბ. ჯერ გაიცემა 25 შემდეგ კი - 5
- გ. ჯერ გაიცემა 5 შემდეგ კი - 25

9.6.22. SELECT SIN(0) AS [სინუსი], COS(0) AS [კოსინუსი],
ABS(-10) AS [აბსოლუტური მნიშვნელობა];

მოთხოვნის შესრულების შედეგად:

- ა. გამოითვლება SIN(0) და COS(0)
- ბ. ჯერ გამოითვლება COS(0), შემდეგ SIN(0), ბოლოს კი - ABS(-10)
- გ. ჯერ გამოითვლება SIN(0), შემდეგ COS(0), ბოლოს კი - ABS(-10)

9.6.23. SELECT RAND(); მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა შემთხვევითი რიცხვი
- ბ. არ გაიცემა შემთხვევითი რიცხვი
- გ. გაიცემა კვადრატული ფესვის მნიშვნელობა

9.6.24. SELECT FLOOR(3.7); მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა 4
- ბ. გაიცემა 3
- გ. გაიცემა 3.7

9.6.25. SELECT CEILING(3.4); მოთხოვნის შესრულების შედეგად:

- ა. გაიცემა 3
- ბ. გაიცემა 4
- გ. გაიცემა 3.7

სტრიქონებთან სამუშაო ფუნქციები

9.7.1. ASCII(სტრიქონი) ფუნქცია:

- ა. გასცემს სტრიქონის მარცხენა სიმბოლოს ASCII კოდს
- ბ. გასცემს სტრიქონის მარცხენა სიმბოლოს UNICODE კოდს
- გ. გასცემს არგუმენტის შესაბამის სიმბოლოს

9.7.2. UNICODE('უნიკოდ_სტრიქონი') ფუნქცია:

- ა. გასცემს სტრიქონის მარცხენა სიმბოლოს UNICODE კოდს
- ბ. გასცემს სტრიქონის მარცხენა სიმბოლოს ASCII კოდს

გ. გასცემს არგუმენტის შესაბამის Unicode სიმბოლოს

9.7.3. CHAR(მთელრიცხვა_გამოსახულება) ფუნქცია:

ა. გასცემს არგუმენტის ზომას ბაიტებში

ბ. გასცემს არგუმენტის შესაბამის სიმბოლოს

გ. გასცემს სტრიქონის მარცხენა სიმბოლოს ASCII კოდს

9.7.4. NCHAR(მთელრიცხვა_გამოსახულება) ფუნქცია:

ა. გასცემს არგუმენტის შესაბამის სიმბოლოს

ბ. გასცემს არგუმენტის შესაბამის Unicode სიმბოლოს

გ. გასცემს არგუმენტის ზომას ბაიტებში

9.7.5. LEN(სტრიქონი) ფუნქცია:

ა. გასცემს არგუმენტის შესაბამის Unicode სიმბოლოს

ბ. შლის ინტერვალებს სტრიქონის დასაწყისიდან

გ. გასცემს არგუმენტის ზომას ბაიტებში

9.7.6. LTRIM(სტრიქონი) ფუნქცია:

ა. გასცემს მითითებული რაოდენობის სიმბოლოს სტრიქონის დასაწყისიდან

ბ. შლის ინტერვალებს სტრიქონის ბოლოდან

გ. შლის ინტერვალებს სტრიქონის დასაწყისიდან

9.7.7. RTRIM(სტრიქონი) ფუნქცია:

ა. გასცემს მითითებული რაოდენობის სიმბოლოს სტრიქონის დასაწყისიდან

ბ. შლის ინტერვალებს სტრიქონის დასაწყისიდან

გ. შლის ინტერვალებს სტრიქონის ბოლოდან

9.7.8. LEFT(სტრიქონი, მთელრიცხვა_გამოსახულება) ფუნქცია:

ა. გასცემს მითითებული რაოდენობის სიმბოლოს სტრიქონის დასაწყისიდან

ბ. სტრიქონის სიმბოლოები ქვედა რეგისტრში გადაჰყავს

გ. გასცემს მითითებული რაოდენობის სიმბოლოს სტრიქონის ბოლოდან

9.7.9. RIGHT(სტრიქონი, მთელრიცხვა_გამოსახულება) ფუნქცია:

ა. სტრიქონის სიმბოლოები ქვედა რეგისტრში გადაჰყავს

ბ. გასცემს მითითებული რაოდენობის სიმბოლოს სტრიქონის ბოლოდან

გ. სტრიქონის სიმბოლოები ზედა რეგისტრში გადაჰყავს

9.7.10. LOWER(სტრიქონი) ფუნქცია:

ა. გასცემს მითითებული რაოდენობის სიმბოლოს სტრიქონის ბოლოდან

ბ. სტრიქონის სიმბოლოები ზედა რეგისტრში გადაჰყავს

გ. სტრიქონის სიმბოლოები ქვედა რეგისტრში გადაჰყავს

9.7.11. UPPER(სტრიქონი) ფუნქცია:

ა. სტრიქონის სიმბოლოები ზედა რეგისტრში გადაჰყავს

ბ. მითითებული სტრიქონისათვის გასცემს მითითებული სიგრძის ქვესტრიქონს დაწყებული მითითებული პოზიციიდან

გ. 'სტრიქონი' არგუმენტში ეძებს შაბლონის საშუალებით მითითებულ ქვესტრიქონს

9.7.12. STR(წილადი_გამოსახულება,[,თანრიგების_საერთო_რაოდენობა[,
თანრიგების_რაოდენობა_წერტილის_შემდეგ]]) ფუნქცია:

ა. მითითებული სტრიქონისათვის გასცემს მითითებული სიგრძის ქვესტრიქონს დაწყებული მითითებული პოზიციიდან

ბ. 'სტრიქონი' არგუმენტში ეძებს შაბლონის საშუალებით მითითებულ ქვესტრიქონს

გ. რიცხვით მნიშვნელობას სიმბოლურ მნიშვნელობად გარდაქმნის. პირველი არგუმენტია გარდასაქმნელი რიცხვი, მეორეა - თანრიგების საერთო რაოდენობა წერტილის ჩათვლით, მესამე კი - ათობითი წერტილის შემდეგ თანრიგების რაოდენობა

9.7.13. SUBSTRING(სტრიქონი, საწყისი_პოზიცია, სიგრძე) ფუნქცია:

ა. გასცემს მითითებული რაოდენობის ინტერვალს

ბ. მითითებული სტრიქონისათვის გასცემს მითითებული სიგრძის ქვესტრიქონს დაწყებული მითითებული პოზიციიდან

გ. 'სტრიქონი' არგუმენტში ეძებს შაბლონის საშუალებით მითითებულ ქვესტრიქონს

9.7.14. CHARINDEX(სტრიქონი1, სტრიქონი2 [, საწყისი_პოზიცია]) ფუნქცია:

ა. გასცემს მითითებული რაოდენობის ინტერვალს

ბ. სტრიქონი2 სტრიქონში ეძებს სტრიქონი1 ქვესტრიქონს. პოზიციის შემთხვევაში გაიცემა იმ პოზიციის ნომერი სადაც პირველად მოიძებნა ქვესტრიქონი. თუ სტრიქონი1 ქვესტრიქონი რამდენიმეჯერ მეორდება სტრიქონი2 სტრიქონში, მაშინ რომელიმე მათგანის მოსაძებნად უნდა მივუთითოთ ძებნის დაწყების პოზიცია

გ. ახდენს სტრიქონის ტირაჟირებას იმდენჯერ, რა რიცხვიც მითითებულია მეორე პარამეტრით

9.7.15. PATINDEX(“%შაბლონი%”, სტრიქონი) ფუნქცია:

ა. 'სტრიქონი' არგუმენტში ეძებს შაბლონის საშუალებით მითითებულ ქვესტრიქონს

ბ. ახდენს სტრიქონის ტირაჟირებას იმდენჯერ, რა რიცხვიც მითითებულია მეორე პარამეტრით

გ. გასცემს მითითებული რაოდენობის ინტერვალს

9.7.16. SPACE(მთელრიცხვა_გამოსახულება) ფუნქცია:

ა. ახდენს სტრიქონის ტირაჟირებას იმდენჯერ, რა რიცხვიც მითითებულია მეორე პარამეტრით

ბ. 'სტრიქონი1' სტრიქონიდან შლის მითითებული რაოდენობის სიმბოლოს დაწყებული მითითებული პოზიციიდან და მათ ნაცვლად 'სტრიქონი1' სტრიქონში ათავსებს სტრიქონი2 სტრიქონს

გ. გასცემს მითითებული რაოდენობის ინტერვალს

9.7.17. REPLICATE(სტრიქონი, მთელრიცხვა_გამოსახულება) ფუნქცია:

ა. პირველი პარამეტრით მითითებულ სტრიქონში მეორე პარამეტრით მითითებულ ქვესტრიქონს შეცვლის მესამე პარამეტრით მითითებული ქვესტრიქონით

ბ. 'სტრიქონი1' სტრიქონიდან შლის მითითებული რაოდენობის სიმბოლოს დაწყებული მითითებული პოზიციიდან და მათ ნაცვლად 'სტრიქონი1' სტრიქონში ათავსებს სტრიქონი2 სტრიქონს

გ. ახდენს სტრიქონის ტირაჟირებას იმდენჯერ, რა რიცხვიც მითითებულია მეორე პარამეტრით

9.7.18. STUFF(სტრიქონი1, საწყისი_პოზიცია, რაოდენობა, სტრიქონი2) ფუნქცია:

- ა. ფუნქცია ახდენს მითითებული სტრიქონის შებრუნებას
- ბ. 'სტრიქონი1' სტრიქონიდან შლის მითითებული რაოდენობის სიმბოლოს დაწყებული მითითებული პოზიციიდან და მათ ნაცვლად 'სტრიქონი1' სტრიქონში ათავსებს სტრიქონი2 სტრიქონს
- გ. პირველი პარამეტრით მითითებულ სტრიქონში მეორე პარამეტრით მითითებულ ქვესტრიქონს შეცვლის მესამე პარამეტრით მითითებული ქვესტრიქონით

9.7.19. REPLACE('სტრიქონი1', 'სტრიქონი2', 'სტრიქონი3') ფუნქცია:

- ა. ფუნქცია ახდენს მითითებული სტრიქონის შებრუნებას
- ბ. პირველი პარამეტრით მითითებულ სტრიქონში მეორე პარამეტრით მითითებულ ქვესტრიქონს შეცვლის მესამე პარამეტრით მითითებული ქვესტრიქონით
- გ. მითითებულ სტრიქონს ათავსებს კვადრატულ ფრჩხილებში, თუ შემზღუდავი სიმბოლო მითითებული არ არის. მეორე პარამეტრი განსაზღვრავს შემზღუდავ სიმბოლოს(('{','}')

9.7.20. REVERSE(სტრიქონი) ფუნქცია:

- ა. 'სტრიქონი1' სტრიქონიდან შლის მითითებული რაოდენობის სიმბოლოს დაწყებული მითითებული პოზიციიდან და მათ ნაცვლად 'სტრიქონი1' სტრიქონში ათავსებს სტრიქონი2 სტრიქონს
- ბ. მითითებულ სტრიქონს ათავსებს კვადრატულ ფრჩხილებში, თუ შემზღუდავი სიმბოლო მითითებული არ არის. მეორე პარამეტრი განსაზღვრავს შემზღუდავ სიმბოლოს(('{','}')
- გ. ფუნქცია ახდენს მითითებული სტრიქონის შებრუნებას

9.7.21. QUOTENAME('სტრიქონი' [, 'შემზღუდავი_სიმბოლო']) ფუნქცია:

- ა. მითითებულ სტრიქონს ათავსებს კვადრატულ ფრჩხილებში, თუ შემზღუდავი სიმბოლო მითითებული არ არის. მეორე პარამეტრი განსაზღვრავს შემზღუდავ სიმბოლოს(('{','}')
- ბ. სტრიქონის სიმბოლოები ზედა რეგისტრში გადაჰყავს
- გ. 'სტრიქონი1' სტრიქონიდან შლის მითითებული რაოდენობის სიმბოლოს დაწყებული მითითებული პოზიციიდან და მათ ნაცვლად 'სტრიქონი1' სტრიქონში ათავსებს სტრიქონი2 სტრიქონს

9.7.22. SELECT N'რომან' + SPACE(5) + N'სამხარაძე'; მოთხოვნის შესრულების შედეგად გაიცემა:

- ა. 'სამხარაძე რომან'
- ბ. 'რომან სამხარაძე'
- გ. 'რომანსამხარაძე'

9.7.23. SELECT STUFF(N'რომან სამხარაძე', 1, 5, N'საბა'); მოთხოვნის შესრულების შედეგად გაიცემა:

- ა. 'საბასამხარაძე'
- ბ. 'რომან სამხარაძე'
- გ. 'საბა სამხარაძე'

9.7.24. SELECT REPLACE(N'სამხარაძე, გაჩევილაძე, კაპანაძე', N'ძე', N'შვილი'); მოთხოვნის

შესრულების შედეგად გაიცემა:

- ა. 'სამხარაშვილი, გაჩეჩილაშვილი, კაპანაშვილი'
- ბ. 'სამხარაძე, გაჩეჩილაძე, კაპანაძე'
- გ. 'სამხარაშვილი, გაჩეჩილაძე, კაპანაძე'

9.7.25. SELECT REPLICATE(N'საბა სამხარაძე, ', 3); მოთხოვნის შესრულების შედეგად გაიცემა:

- ა. 'საბა სამხარაძე, '
- ბ. 'საბა სამხარაძე, საბა სამხარაძე, საბა სამხარაძე, '
- გ. 'საბა სამხარაძე, საბა სამხარაძე, '

9.7.26. SELECT CHARINDEX(N'სა', N'საბა სამხარაძე'), CHARINDEX(N'სა', N'საბა სამხარაძე', 2);
მოთხოვნის შესრულების შედეგად გაიცემა:

- ა. 1 1
- ბ. 6 1
- გ. 1 6

9.7.27. SELECT LEN(N'რომან სამხარაძე'); მოთხოვნის შესრულების შედეგად გაიცემა:

- ა. 15
- ბ. 14
- გ. 16

9.7.28. SELECT LEFT(N'რომან სამხარაძე', 5), RIGHT(N'რომან სამხარაძე', 10); მოთხოვნის
შესრულების შედეგად გაიცემა:

- ა. 'სამხარაძე' და 'რომან'
- ბ. 'რომან' და 'სამხარაძე'
- გ. 'სამხარაძე' და 'სამხარაძე'

თარიღებთან სამუშაო ფუნქციები

9.8.1. GETDATE() ფუნქცია:

- ა. გასცემს მიმდინარე სისტემურ დროს
- ბ. გასცემს დღეს მითითებული თარიღიდან
- გ. გასცემს 1-ს, თუ მითითებული გამოსახულება შეიძლება გარდაიქმნას datetime ან smalldatetime ტიპად, წინააღმდეგ შემთხვევაში გაიცემს 0

9.8.2. ISDATE(გამოსახულება) ფუნქცია:

- ა. გასცემს დღეს მითითებული თარიღიდან
- ბ. გასცემს 1-ს, თუ მითითებული გამოსახულება შეიძლება გარდაიქმნას datetime ან smalldatetime ტიპად, წინააღმდეგ შემთხვევაში გაიცემს 0
- გ. გასცემს თვეს მითითებული თარიღიდან

9.8.3. DAY(თარიღი) ფუნქცია:

- ა. გასცემს დღეს მითითებული თარიღიდან
- ბ. გასცემს თვეს მითითებული თარიღიდან
- გ. გასცემს წელს მითითებული თარიღიდან

9.8.4. MONTH(თარიღი) ფუნქცია:

- ა. გასცემს წელს მითითებული თარიღიდან
- ბ. გასცემს თვეს მითითებული თარიღიდან
- გ. მითითებულ თარიღს უმატებს დღეებს, თვეებს, წლებს, საათებს, წუთებს და ა.შ.

9.8.5. YEAR(თარიღი) ფუნქცია:

- ა. მითითებულ თარიღს უმატებს დღეებს, თვეებს, წლებს, საათებს, წუთებს და ა.შ.
- ბ. გამოთვლის სხვაობას ორი თარიღის მითითებულ ნაწილებს შორის
- გ. გასცემს წელს მითითებული თარიღიდან

9.8.6. DATEADD(datepart, რიცხვი, თარიღი) ფუნქცია:

- ა. მითითებულ თარიღს უმატებს დღეებს, თვეებს, წლებს, საათებს, წუთებს და ა.შ.
- ბ. თარიღიდან გამოყოფს მითითებულ ნაწილს და გასცემს მას სიმბოლურ ფორმატში
- გ. გამოთვლის სხვაობას ორი თარიღის მითითებულ ნაწილებს შორის

9.8.7. DATEDIFF(თარიღის_ნაწილი, საწყისი_თარიღი, საბოლოო_თარიღი) ფუნქცია:

- ა. თარიღიდან გამოყოფს მითითებულ ნაწილს და გასცემს მას რიცხვით ფორმატში
- ბ. გამოთვლის სხვაობას ორი თარიღის მითითებულ ნაწილებს შორის
- გ. თარიღიდან გამოყოფს მითითებულ ნაწილს და გასცემს მას სიმბოლურ ფორმატში

9.8.8. DATENAME(თარიღის_ნაწილი, თარიღი) ფუნქცია:

- ა. გასცემს წელს მითითებული თარიღიდან
- ბ. თარიღიდან გამოყოფს მითითებულ ნაწილს და გასცემს მას სიმბოლურ ფორმატში
- გ. თარიღიდან გამოყოფს მითითებულ ნაწილს და გასცემს მას რიცხვით ფორმატში

9.8.9. DATEPART(თარიღის_ნაწილი, თარიღი) ფუნქცია:

- ა. გასცემს დღეს მითითებული თარიღიდან
- ბ. გასცემს წელს მითითებული თარიღიდან
- გ. თარიღიდან გამოყოფს მითითებულ ნაწილს და გასცემს მას რიცხვით ფორმატში

9.8.10. SET DATEFORMAT DMY;

SELECT DATEADD(dd, 5, '19.03.2005'), DATEADD(yy, -3, '05 Oct 2000'),
DATEADD(hh, 10, '19.03.2005 4:30:45'); მოთხოვნის შესრულების შედეგად გაიცემა:

- ა. 2005-03-24 00:00:00, 1997-10-05 00:00:00, 2005-03-19 14:30:45
- ბ. 2005-03-19 00:00:00, 2000-10-05 00:00:00, 2005-03-19 14:30:45
- გ. 2005-03-24 00:00:00, 1997-10-05 00:00:00, 2005-03-19 4:30:45

9.8.11. SET DATEFORMAT DMY;

SELECT DATEDIFF(dd, '19.03.2005', '10.03.2005'), DATEDIFF(yy, '05 Oct 2000', '05 Oct 2005'),
DATEDIFF(hh, '19.03.2005 4:30:45', '19.03.2005 14:40:55'); მოთხოვნის შესრულების

შედეგად გაიცემა:

- ა. 9 -5 10
- ბ. -9 5 10
- გ. 9 5 -10

9.8.12. SET DATEFORMAT DMY;

SELECT MONTH('19/03/2005'), MONTH('05 Oct 2000'); მოთხოვნის შესრულების შედეგად

გაცემა:

ა.	3	3
ბ.	10	3
გ.	3	10

თავი 10. შენახული პროცედურები

შესავალი. შენახული პროცედურების ტიპები

10.1.1. შენახული პროცედურა:

- ა. არის Transact-SQL-ის ბრძანებების სახელდებული ნაკრები
- ბ. არის Transact-SQL-ის ტრიგერების სახელდებული ნაკრები
- გ. არ არის Transact-SQL-ის ბრძანებების სახელდებული ნაკრები

10.1.2. შენახული პროცედურა:

- ა. უშუალოდ სერვერზე არ ინახება
- ბ. უშუალოდ სერვერზე ინახება
- გ. უშუალოდ წარმოდგენაში ინახება

10.1.3. შენახული პროცედურა:

- ა. არ წარმოადგენს მონაცემთა ბაზის დამოუკიდებელ ობიექტს
- ბ. წარმოადგენს მონაცემთა ბაზის დამოუკიდებელ ობიექტს
- გ. წარმოადგენს მონაცემთა ბაზის დამოუკიდებელ ობიექტს

10.1.4. შენახული პროცედურების გამოძახება:

- ა. შეუძლებელია კლიენტის პროგრამის, სხვა შენახული პროცედურის ან ტრიგერის მიერ
- ბ. შესაძლებელია კლიენტის პროგრამის, სხვა შენახული პროცედურის ან ტრიგერის მიერ
- გ. შესაძლებელია კლიენტის ინდექსის მიერ

10.1.5. შენახული პროცედურის კოდის შეცვლა:

- ა. შეუძლია მხოლოდ მის მფლობელს ან მონაცემთა ბაზის db_owner ფიქსირებული როლის წევრს
- ბ. არ შეუძლია მხოლოდ მის მფლობელს ან მონაცემთა ბაზის db_owner ფიქსირებული როლის წევრს
- გ. შეუძლია მხოლოდ მონაცემთა ბაზის db_owner არაფიქსირებული როლის წევრს

10.1.6. შენახული პროცედურის ფლობის უფლება:

- ა. საერთოდ არ გადაიცემა
- ბ. შეგვიძლია ერთი მომხმარებლიდან მეორეს გადავცეთ
- გ. არ შეგვიძლია ერთი მომხმარებლიდან მეორეს გადავცეთ

10.1.7. შენახული პროცედურის შესრულების წინ სერვერი მისთვის:

- ა. არ ადგენს შესრულების გეგმას, ასრულებს მის ოპტიმიზებას და კომპილირებას
- ბ. ადგენს შესრულების გეგმას, ასრულებს მის ოპტიმიზებას და კომპილირებას
- გ. ადგენს შესრულების გეგმას, არ ასრულებს მის ოპტიმიზებას და კომპილირებას

10.1.8. მწარმოებლურობის ასამაღლებლად სერვერი:

- ა. არ ასრულებს შენახული პროცედურის შესრულების გეგმისა და კომპილირებული კოდის ქეშირებას
- ბ. ასრულებს შენახული პროცედურის მხოლოდ შესრულების გეგმის ქეშირებას
- გ. ასრულებს შენახული პროცედურის შესრულების გეგმისა და კომპილირებული კოდის ქეშირებას

10.1.9. შენახული პროცედურის განმეორებით გამოძახების შემთხვევაში სერვერი მაშინვე:

- ა. არ დაიწყებს ბრძანებების შესრულებას
- ბ. დაიწყებს ბრძანებების შესრულებას
- გ. დაიწყებს მის მომზადებას შესასრულებლად

10.1.10. შენახული პროცედურის შესასრულებლად:

- ა. საკმარისია მისი სახელის მითითება
- ბ. საკმარისი არ არის მისი სახელის მითითება
- გ. საკმარისია მისი პარამეტრების მითითება

10.1.11. შენახული პროცედურის შესასრულებლად საკმარისია მისი სახელის მითითება. ეს:

- ა. არ ამცირებს მოთხოვნის ზომას, რომელიც ქსელში იგზავნება კლიენტიდან სერვერისკენ
- ბ. ამცირებს მოთხოვნის ზომას, რომელიც ქსელში იგზავნება კლიენტიდან სერვერისკენ
- გ. ამცირებს მოთხოვნის ზომას, რომელიც ქსელში იგზავნება სერვერიდან კლიენტისკენ

10.1.12. შენახული პროცედურების გამოყენება:

- ა. ზრდის ქსელზე დატვირთვებს
- ბ. არ ამცირებს ქსელზე დატვირთვებს
- გ. ამცირებს ქსელზე დატვირთვებს

10.1.13. სისტემური შენახული პროცედურები:

- ა. შედიან სერვერის შემადგენლობაში და განკუთვნილი არიან ადმინისტრაციული მოქმედებების შესასრულებლად
- ბ. არ შედიან სერვერის შემადგენლობაში და განკუთვნილი არიან ადმინისტრაციული მოქმედებების შესასრულებლად
- გ. შედიან სერვერის შემადგენლობაში და განკუთვნილი არ არიან ადმინისტრაციული მოქმედებების შესასრულებლად

10.1.14. სისტემურ შენახულ პროცედურებს:

- ა. არ აქვთ sp_ პრეფიქსი
- ბ. აქვთ sp_ პრეფიქსი
- გ. აქვთ _sp პრეფიქსი

10.1.15. სისტემური შენახული პროცედურები:

- ა. არ ინახება master სისტემურ მონაცემთა ბაზაში და შეგვიძლია ნებისმიერი მონაცემთა ბაზიდან გამოვიძახოთ
- ბ. ინახება master სისტემურ მონაცემთა ბაზაში და არ შეგვიძლია ნებისმიერი მონაცემთა ბაზიდან გამოვიძახოთ
- გ. ინახება master სისტემურ მონაცემთა ბაზაში და შეგვიძლია ნებისმიერი მონაცემთა ბაზიდან გამოვიძახოთ

10.1.16. მომხმარებლის მიერ შექმნილი შენახული პროცედურები:

- ა. არ ინახება შესაბამის მონაცემთა ბაზაში და წარმოადგენს ამ მონაცემთა ბაზის ობიექტს
- ბ. ინახება შესაბამის მონაცემთა ბაზაში და წარმოადგენს ამ მონაცემთა ბაზის ობიექტს
- გ. ინახება შესაბამის მონაცემთა ბაზაში და არ წარმოადგენს ამ მონაცემთა ბაზის ობიექტს

10.1.17.დროებითი შენახული პროცედურები არსებობენ:

- ა. მხოლოდ გარკვეული დროის განმავლობაში, რის შემდეგ ავტომატურად იშლება სერვერის მიერ
- ბ. ხანგრძლივად
- გ. მხოლოდ გარკვეული დროის განმავლობაში, რის შემდეგ ავტომატურად არ იშლება სერვერის მიერ

10.1.18.დროებითი პროცედურები არსებობს:

- ა. მხოლოდ გლობალური
- ბ. მხოლოდ ლოკალური
- გ. ლოკალური და გლობალური

10.1.19.ლოკალური დროებითი შენახული პროცედურები:

- ა. არ შეიძლება გამოძახებული იყოს მხოლოდ იმ შეერთებიდან, რომლებშიც ისინი შეიქმნა
- ბ. შეიძლება გამოძახებული იყოს მხოლოდ იმ შეერთებიდან, რომლებშიც ისინი შეიქმნა
- გ. შეიძლება გამოძახებული იყოს მხოლოდ იმ შეერთებიდან, რომლებშიც ისინი არ შეიქმნა

10.1.20.პროცედურების სახელები:

- ა. # სიმბოლოთი უნდა იწყებოდეს
- ბ. * სიმბოლოთი უნდა იწყებოდეს
- გ. # სიმბოლოთი არ უნდა იწყებოდეს

10.1.21.დროებითი შენახული პროცედურები:

- ა. არ ინახება tempdb მონაცემთა ბაზაში და ავტომატურად იშლება მომხმარებლის გამორთვისას, სერვერის გაჩერების ან ხელახალი გაშვების დროს
- ბ. ინახება tempdb მონაცემთა ბაზაში და არ იშლება მომხმარებლის გამორთვისას, სერვერის გაჩერების ან ხელახალი გაშვების დროს
- გ. ინახება tempdb მონაცემთა ბაზაში და ავტომატურად იშლება მომხმარებლის გამორთვისას, სერვერის გაჩერების ან ხელახალი გაშვების დროს

10.1.22.გლობალური დროებითი შენახული პროცედურები:

- ა. შეიძლება გამოძახებული იყოს ნებისმიერი შეერთებიდან
- ბ. საერთოდ არ შეიძლება გამოძახებული იყოს
- გ. არ შეიძლება გამოძახებული იყოს ნებისმიერი შეერთებიდან

10.1.23.გლობალური დროებითი პროცედურის განსაზღვრისას მისი სახელის წინ:

- ა. უნდა მოვათავსოთ ** სიმბოლოები
- ბ. უნდა მოვათავსოთ ## სიმბოლოები
- გ. არ უნდა მოვათავსოთ ## სიმბოლოები

10.1.24.გლობალური დროებითი პროცედურები:

- ა. არ ინახება tempdb მონაცემთა ბაზაში და იშლება სერვერის გაჩერების ან ხელახალი გაშვების დროს, აგრეთვე, იმ შეერთების დახურვისას, რომელშიც ისინი შეიქმნა
- ბ. ინახება tempdb მონაცემთა ბაზაში და არ იშლება სერვერის გაჩერების ან ხელახალი გაშვების დროს, აგრეთვე, იმ შეერთების დახურვისას, რომელშიც ისინი შეიქმნა

გ. ინახება tempdb მონაცემთა ბაზაში და იშლება სერვერის გაჩერების ან ხელახალი გაშვების დროს, აგრეთვე, იმ შეერთების დახურვისას, რომელშიც ისინი შეიქმნა

შენახული პროცედურის შექმნა

10.2.1. შენახულ პროცედურას ექნება:

- ა. მისი შემქმნელი მომხმარებლის უფლებები
- ბ. მისი შემსრულებელი მომხმარებლის უფლებები
- გ. მისი წამშლელი მომხმარებლის უფლებები

10.2.2. თუ პროცედურის შექმნისას ნებადართული იყო სისტემურ ცხრილებთან მიმართვა, მაშინ პროცედურა:

- ა. ყოველთვის ვერ შეძლებს სისტემურ ცხრილებთან მიმართვას
- ბ. ყოველთვის შეძლებს სისტემურ ცხრილებთან მიმართვას
- გ. ხანდახან შეძლებს სისტემურ ცხრილებთან მიმართვას

10.2.3. შენახულ პროცედურას:

- ა. აუცილებლად უნდა ჰქონდეს შესასვლელი და გამოსასვლელი პარამეტრები
- ბ. აუცილებლად არ უნდა ჰქონდეს შესასვლელი და გამოსასვლელი პარამეტრები
- გ. შეიძლება ჰქონდეს შესასვლელი და გამოსასვლელი პარამეტრები

10.2.4. პროცედურის პარამეტრები:

- ა. შეგვიძლია გამოვიყენოთ როგორც ჩვეულებრივი ცვლადები
- ბ. შეგვიძლია გამოვიყენოთ როგორც არაჩვეულებრივი ცვლადები
- გ. არ შეგვიძლია გამოვიყენოთ როგორც ჩვეულებრივი ცვლადები

10.2.5. პროცედურის კოდი:

- ა. შეიძლება შეიცავდეს ინდექსების ნებისმიერ მიმდევრობას
- ბ. არ შეიძლება შეიცავდეს ნებისმიერი ბრძანებების მიმდევრობას, სხვა შენახული პროცედურების გამოძახებების ჩათვლით
- გ. შეიძლება შეიცავდეს ნებისმიერი ბრძანებების მიმდევრობას, სხვა შენახული პროცედურების გამოძახებების ჩათვლით

10.2.6. შენახული პროცედურის შესაქმნელად გამოიყენება:

- ა. CREATE DATABASE ბრძანება
- ბ. CREATE PROCEDURE ბრძანება
- გ. CREATE INDEX ბრძანება

10.2.7. შენახული პროცედურის შესაქმნისას:

- ა. დაუშვებელია მფლობელის სახელისა და დასაშვებია მონაცემთა ბაზის სახელის მითითება
- ბ. დასაშვებია მფლობელის სახელისა და მონაცემთა ბაზის სახელის მითითება
- გ. დაუშვებელია მფლობელის სახელისა და მონაცემთა ბაზის სახელის მითითება

10.2.8. შენახული პროცედურა ყოველთვის:

- ა. არ იქმნება მიმდინარე მონაცემთა ბაზაში

- ბ. იქმნება მიმდინარე მონაცემთა ბაზაში
- გ. იქმნება master მონაცემთა ბაზაში

10.2.9. შენახული პროცედურის პარამეტრების სახელები:

- ა. @ სიმბოლოთი უნდა იწყებოდეს
- ბ. # სიმბოლოთი უნდა იწყებოდეს
- გ. * სიმბოლოთი უნდა იწყებოდეს

10.2.10. შენახული პროცედურის პარამეტრები ერთმანეთისაგან გამოიყოფა:

- ა. წერტილებით
- ბ. წერტილ-მძიმეებით
- გ. მძიმეებით

10.2.11. შენახული პროცედურის პარამეტრები წარმოადგენენ:

- ა. გლობალურ პარამეტრებს, ამიტომ სხვადასხვა შენახულ პროცედურებს შეიძლება ერთნაირი პარამეტრები ჰქონდეს
- ბ. ლოკალურ პარამეტრებს, ამიტომ სხვადასხვა შენახულ პროცედურებს შეიძლება ერთნაირი პარამეტრები ჰქონდეს
- გ. ლოკალურ პარამეტრებს, ამიტომ სხვადასხვა შენახულ პროცედურებს არ შეიძლება ერთნაირი პარამეტრები ჰქონდეს

10.2.12. შენახულ პროცედურაში პარამეტრის სახელი ცვლადის სახელს:

- ა. არ უნდა ემთხვეოდეს
- ბ. უნდა ემთხვეოდეს
- გ. ხანდახან უნდა ემთხვეოდეს

10.2.13. პარამეტრს:

- ა. არ შეიძლება ნებისმიერი ტიპი ჰქონდეს, მათ შორის მომხმარებლის მიერ განსაზღვრული ტიპები
- ბ.
- გ. შეიძლება ნებისმიერი ტიპი ჰქონდეს, მათ შორის მომხმარებლის მიერ განსაზღვრული ტიპები

10.2.14. რაც შეეხება cursor ტიპს, ის შეგვიძლია გამოვიყენოთ:

- ა. მხოლოდ შესასვლელი პარამეტრებისათვის
- ბ. როგორც გამოსასვლელი, ისე შესასვლელი პარამეტრებისთვის
- გ. მხოლოდ გამოსასვლელი პარამეტრებისათვის

10.2.15. OUTPUT არგუმენტი მიუთითებს, რომ პარამეტრი:

- ა. არის გამოსასვლელი, ანუ შეგვიძლია გამოვიყენოთ მონაცემების დასაბრუნებლად შენახული პროცედურიდან
- ბ. არის გამოსასვლელი, ანუ არ შეგვიძლია გამოვიყენოთ მონაცემების დასაბრუნებლად შენახული პროცედურიდან
- გ. არც შესასვლელია და არც გამოსასვლელი

10.2.16.შენახული პროცედურიდან გამოსვლისას გამოსასვლელი პარამეტრის მიმდინარე მნიშვნელობა:

- ა. ენიჭება იმ ლოკალურ ცვლადს, რომელიც მითითებული არ იყო შენახული პროცედურის გამოძახების დროს
- ბ. ენიჭება იმ ლოკალურ ცვლადს, რომელიც მითითებული იყო შენახული პროცედურის გამოძახების დროს
- გ. არ ენიჭება იმ ლოკალურ ცვლადს, რომელიც მითითებული იყო შენახული პროცედურის გამოძახების დროს

10.2.17.CREATE PROCEDURE ბრძანების VARYING არგუმენტს:

- ა. არ აქვს cursor ტიპი და არ გამოიყენება OUTPUT არგუმენტთან ერთად
- ბ. აქვს cursor ტიპი და გამოიყენება OUTPUT არგუმენტთან ერთად
- გ. აქვს int ტიპი და არ გამოიყენება OUTPUT არგუმენტთან ერთად

10.2.18.CREATE PROCEDURE ბრძანების VARYING არგუმენტი მიუთითებს, რომ გამოსასვლელი პარამეტრი:

- ა. იქნება სიმრავლე
- ბ. არ იქნება სიმრავლე
- გ. იქნება მთელი რიცხვი

10.2.19.CREATE PROCEDURE ბრძანების DEFAULT არგუმენტი მიუთითებს იმ მნიშვნელობას, რომელსაც პარამეტრი ავტომატურად:

- ა. არ მიიღებს თუ მისი მნიშვნელობა არ იქნება მითითებული
- ბ. მიიღებს თუ მისი მნიშვნელობა არ იქნება მითითებული
- გ. მიიღებს თუ მისი მნიშვნელობა იქნება მითითებული

10.2.20.CREATE PROCEDURE ბრძანების RECOMPILE არგუმენტი მიუთითებს, რომ შენახული პროცედურის ყოველი გამოძახებისას:

- ა. უნდა შედგეს მისი შესრულების გეგმა და არ უნდა შესრულდეს შენახული პროცედურის კოდის ხელახალი კომპილირება
- ბ. არ უნდა შედგეს მისი შესრულების გეგმა და შესრულდეს შენახული პროცედურის კოდის ხელახალი კომპილირება
- გ. უნდა შედგეს მისი შესრულების გეგმა და შესრულდეს შენახული პროცედურის კოდის ხელახალი კომპილირება

10.2.21.CREATE PROCEDURE ბრძანების ENCRYPTION არგუმენტი მიუთითებს, რომ:

- ა. უნდა შესრულდეს შენახული პროცედურის კოდის დაშიფვრა
- ბ. არ უნდა შესრულდეს შენახული პროცედურის კოდის დაშიფვრა
- გ. უნდა შესრულდეს შენახული პროცედურის კოდის გაშიფვრა

10.2.22.CREATE PROCEDURE ბრძანების AS არგუმენტი:

- ა. ხანდახან იწყებს შენახული პროცედურის ტანს
- ბ. არ იწყებს შენახული პროცედურის ტანს
- გ. იწყებს შენახული პროცედურის ტანს

10.2.23.შენახული პროცედურიდან გამოსასვლელად შეგვიძლია გამოვიყენოთ:

- ა. CREATE PROCEDURE ბრძანება
- ბ. RETURN ბრძანება
- გ. UPDATE ბრძანება

10.2.24. ერთი შენახული პროცედურის ტანიდან მეორე შენახული პროცედურის გამოძახებისას:

- ა. იქმნება ჩადგმული პროცედურები
- ბ. იქმნება ჩაუდგმელი პროცედურები
- გ. არ იქმნება ჩადგმული პროცედურები

10.2.25. შენახული პროცედურების ჩადგმულობის მიმდინარე დონის მისაღებად შეგვიძლია გამოვიყენოთ:

- ა. @@CPU_BUSY ფუნქცია
- ბ. @@TOTAL_READ ფუნქცია
- გ. @@NESTLEVEL ცვლადი

10.2.26. შექმნილი შენახული პროცედურის სახელი შეიტანება ამავე მონაცემთა ბაზის:

- ა. sys.all_views სისტემურ წარმოდგენაში
- ბ. sys.objects სისტემურ წარმოდგენაში
- გ. sys.indexes სისტემურ წარმოდგენაში

10.2.27. შენახული პროცედურის შექმნის მომენტში:

- ა. არ სრულდება ბრძანებების მხოლოდ სინტაქსური შემოწმება და მოწმდება მასში მითითებული ობიექტების არსებობა
- ბ. სრულდება ბრძანებების მხოლოდ სინტაქსური შემოწმება და არ მოწმდება მასში მითითებული ობიექტების არსებობა
- გ. სრულდება ბრძანებების მხოლოდ სინტაქსური შემოწმება და მოწმდება მასში მითითებული ობიექტების არსებობა

10.2.28. მიმართვების სისწორის შემოწმება ხდება შენახული პროცედურის:

- ა. სინტაქსური შემოწმების დროს
- ბ. შექმნის მომენტში
- გ. კომპილირების დროს

10.2.29. თუ შენახული პროცედურის გამოძახება არის ერთადერთი ბრძანება შესასრულებლად გადასაცემ პაკეტში, მაშინ:

- ა. აუცილებელია EXECUTE ბრძანების გამოყენება
- ბ. შენახული პროცედურა არ შესრულდება მხოლოდ მისი სახელის მითითების გზით
- გ. შენახული პროცედურა შესრულდება მხოლოდ მისი სახელის მითითების გზით

10.2.30. თუ შენახული პროცედურის გამოძახება არ არის ერთადერთი ბრძანება პაკეტში, მაშინ:

- ა. არ უნდა გამოვიყენოთ EXECUTE ბრძანება
- ბ. უნდა გამოვიყენოთ EXECUTE ბრძანება
- გ. შენახული პროცედურა შესრულდება მხოლოდ მისი სახელის მითითების გზით

10.2.31. EXECUTE ბრძანება უნდა გამოვიყენოთ, აგრეთვე, შენახული პროცედურის გამოძახებისას:

- ა. სხვა შენახული პროცედურიდან ან ტრიგერიდან
- ბ. სხვა ინდექსიდან
- გ. სხვა ცხრილიდან

10.2.32.EXECUTE ბრძანების OUTPUT არგუმენტის გამოყენება ნებადართულია იმ შემთხვევაში, როცა შესაბამისი პარამეტრი გამოცხადებული:

- ა. არ იყო OUTPUT სიტყვის გამოყენებით შენახული პროცედურის შექმნის დროს
- ბ. იყო OUTPUT სიტყვის გამოყენებით შენახული პროცედურის გამოძახების დროს
- გ. იყო OUTPUT სიტყვის გამოყენებით შენახული პროცედურის შექმნის დროს

10.2.33.EXECUTE ბრძანების DEFAULT არგუმენტის გამოყენება ნებადართულია იმ შემთხვევაში, როცა შესაბამისი პარამეტრისათვის განსაზღვრული:

- ა. იყო ნაგულისხმევი მნიშვნელობა შენახული პროცედურის გამოძახების დროს
- ბ. იყო ნაგულისხმევი მნიშვნელობა შენახული პროცედურის შექმნის დროს
- გ. არ იყო ნაგულისხმევი მნიშვნელობა შენახული პროცედურის შექმნის დროს

10.2.34.შენახული პროცედურის გამოძახებისას პარამეტრების სახელები შეგვიძლია:

- ა. არ მივუთითოთ
- ბ. მივუთითოთ
- გ. წავშალოთ

10.2.35.შენახული პროცედურის გამოძახებისას პარამეტრების სახელები შეგვიძლია არ მივუთითოთ. ასეთ შემთხვევაში, პარამეტრების მნიშვნელობები ისეთივე მიმდევრობით:

- ა. უნდა მივუთითოთ, როგორც იყვნენ ისინი ჩამოთვლილი შენახული პროცედურის შექმნის დროს
- ბ. არ უნდა მივუთითოთ, როგორც იყვნენ ისინი ჩამოთვლილი შენახული პროცედურის შექმნის დროს
- გ. უნდა მივუთითოთ, როგორც არ იყვნენ ისინი ჩამოთვლილი შენახული პროცედურის შექმნის დროს

10.2.36.შენახული პროცედურის პარამეტრისათვის ავტომატური მნიშვნელობის მინიჭება, ჩამონათვალში მისი სახელის გამოტოვებით:

- ა. არ შეიძლება
- ბ. შეიძლება
- გ. ხანდახან შეიძლება

10.2.37.შენახული პროცედურის ბოლო პარამეტრები შეგვიძლია გამოვტოვოთ, თუ მათთვის:

- ა. განსაზღვრულია ავტომატური მნიშვნელობა
- ბ. განსაზღვრული არ არის ავტომატური მნიშვნელობა
- გ. განსაზღვრულია არავტომატური მნიშვნელობა

10.2.38.შენახული პროცედურის გამოძახებისას:

- ა. შეგვიძლია მივუთითოთ ან პარამეტრის სახელი მნიშვნელობასთან ერთად, ან მხოლოდ მნიშვნელობა სახელის გარეშე
- ბ. არ შეგვიძლია მივუთითოთ ან პარამეტრის სახელი მნიშვნელობასთან ერთად, ან მხოლოდ მნიშვნელობა სახელის გარეშე

გ. შეგვიძლია მივუთითოთ ან პარამეტრის სახელი მნიშვნელობის გარეშე, ან მხოლოდ მნიშვნელობა სახელთან ერთად

10.2.39.CREATE PROCEDURE MyProc @ganyofileba NVARCHAR(30)

AS

SELECT *

FROM Personali

WHERE ganyofileba = @ganyofileba

პროგრამული კოდის შესრულების შედეგად შეიქმნება შენახული პროცედურა, რომელიც:

- ა. გასცემს ინფორმაციას მითითებულ ქალაქში მომუშავე თანამშრომლების შესახებ
- ბ. გასცემს ინფორმაციას მითითებულ განყოფილებაში მომუშავე თანამშრომლების შესახებ
- გ. არ გასცემს ინფორმაციას მითითებულ განყოფილებაში მომუშავე თანამშრომლების შესახებ

10.2.40.CREATE PROCEDURE MyProc @ganyofileba NVARCHAR(30) ბრძანების შესრულების

შედეგად შექმნილია შენახული პროცედურა. მისი გამოძახება შეიძლება შემდეგნაირად:

- ა. EXEC MyProc 100;
- ბ. EXEC MyProc;
- გ. EXEC MyProc N'სავაჭრო';

10.2.41.რომელი შენახული პროცედურა გასცემს თანამშრომლების შესახებ ინფორმაციას:

ა. CREATE PROCEDURE Chemi_Procedura

AS

SELECT * FROM Personali;

ბ. CREATE PROCEDURE Chemi_Procedura

AS

SELECT * FROM Shemkveti;

გ. CREATE PROCEDURE Chemi_Procedura

AS

SELECT * FROM Xelshekruleba;

შენახული პროცედურების მართვა. შენახული პროცედურის შესახებ ინფორმაციის მიღება. შენახული პროცედურის სახელის შეცვლა. შენახული პროცედურის წაშლა. შენახული პროცედურის ავტომატურად შესრულების მართვა

10.3.1. შენახული პროცედურის შესახებ ინფორმაციის მისაღებად:

- ა. უნდა გამოვიყენოთ sp_help პროცედურა
- ბ. არ უნდა გამოვიყენოთ sp_help პროცედურა
- გ. უნდა გამოვიყენოთ sp_rename პროცედურა

10.3.2. შენახული პროცედურის სახელის შესაცვლელად:

- ა. გამოვიყენებთ sp_rename ბრძანებას
- ბ. გამოვიყენებთ sp_help პროცედურას

გ. გამოიყენება sp_procoption პროცედურა

10.3.3. შენახული პროცედურის წასაშლელად:

- ა. გამოიყენება DROP PROCEDURE ბრძანება
- ბ. გამოიყენება sp_help პროცედურა
- გ. გამოიყენება sp_rename ბრძანება

10.3.4. შენახული პროცედურის ავტომატურად შესრულებისათვის:

- ა. საჭიროა მისი კონფიგურირება
- ბ. საჭირო არაა მისი კონფიგურირება
- გ. საჭიროა ტრიგერის კონფიგურირება

10.3.5. შენახული პროცედურის კონფიგურირება:

- ა. არ სრულდება მისი თვისებების ცვლილების გზით
- ბ. სრულდება მისი თვისებების ცვლილების გზით
- გ. სრულდება მისი თვისებების უცვლელობის გზით

10.3.6. შენახული პროცედურის კონფიგურირებისათვის:

- ა. გამოიყენება sp_rename ბრძანება
- ბ. გამოიყენება DROP PROCEDURE ბრძანება
- გ. გამოიყენება sp_procoption პროცედურა

10.3.7. EXEC sp_help MyProc; შენახული პროცედურის შესრულების შედეგად:

- ა. გაიცემა ინფორმაცია MyProc შენახული პროცედურის შესახებ
- ბ. არ გაიცემა ინფორმაცია MyProc შენახული პროცედურის შესახებ
- გ. გაიცემა ინფორმაცია sp_help შენახული პროცედურის შესახებ

10.3.8. EXEC sp_rename 'Chemi_Procedura3', 'Chemi_Procedura'; შენახული პროცედურის შესრულების შედეგად:

- ა. 'Chemi_Procedura' შენახულ პროცედურას დაერქმევა 'Chemi_Procedura3' ახალი სახელი
- ბ. 'Chemi_Procedura3' შენახულ პროცედურას დაერქმევა 'Chemi_Procedura' ახალი სახელი
- გ. 'Chemi_Procedura3' შენახულ პროცედურას არ დაერქმევა 'Chemi_Procedura' ახალი სახელი

10.3.9. იმისათვის, რომ 'Chemi_Procedura3' შენახულ პროცედურას დავარქვათ 'Chemi_Procedura' ახალი სახელი, უნდა შევასრულოთ ბრძანება:

- ა. EXEC sp_rename 'Chemi_Procedura', 'Chemi_Procedura3';
- ბ. EXEC sp_rename 'Chemi_Procedura3', 'Chemi_Procedura';
- გ. EXEC sp_rename 'Chemi_Procedura3', 'Chemi_Procedura';

10.3.10. DROP PROC A1; ბრძანების შესრულების შედეგად:

- ა. წაიშლება A1 შენახული პროცედურა
- ბ. არ წაიშლება A1 შენახული პროცედურა
- გ. შეიქმნება A1 შენახული პროცედურა

თავი 11. ინდექსები

შესავალი. ინდექსების გამოყენების დაგეგმვა

11.1.1. არაინდექსირებულ ცხრილში, რომელშიც ათასობით სტრიქონია, მონაცემების ძებნა:

- ა. დიდ დროს იკავებს
- ბ. მცირე დროს იკავებს
- გ. დროს არ იკავებს

11.1.2. მონაცემების ძებნის დაჩქარების მიზნით გამოიყენება:

- ა. ფუნქციები
- ბ. ინდექსები
- გ. ტრიგერები

11.1.3. ინდექსი წარმოადგენს მიმართვების ნაკრებს, რომლებიც:

- ა. მოწესრიგებულია გარკვეული სტრიქონის მნიშვნელობების მიხედვით
- ბ. მოწესრიგებული არ არის გარკვეული სვეტის მნიშვნელობების მიხედვით
- გ. მოწესრიგებულია გარკვეული სვეტის მნიშვნელობების მიხედვით

11.1.4. ინდექსი მონაცემთა ბაზის:

- ა. დამოკიდებულ ობიექტს წარმოადგენს
- ბ. დამოუკიდებელ ობიექტს წარმოადგენს
- გ. არანაირ ობიექტს არ წარმოადგენს

11.1.5. ინდექსი შეიძლება მოიცავდეს ცხრილის:

- ა. ერთ ან მეტ სვეტს
- ბ. არც ერთ სვეტს
- გ. მხოლოდ ერთ სვეტს

11.1.6. ფიზიკურად ინდექსი წარმოადგენს:

- ა. ინდექსირებული სვეტის მონაცემების მოწესრიგებულ ნაკრებს ცხრილის სტრიქონების ლოგიკური განლაგების ადგილმდებარეობაზე მიმთითებლებით
- ბ. ინდექსირებული სვეტის მონაცემების მოწესრიგებულ ნაკრებს ცხრილის სტრიქონების ფიზიკური განლაგების ადგილმდებარეობაზე მიმთითებლებით
- გ. არაინდექსირებული სვეტის მონაცემების მოწესრიგებულ ნაკრებს ცხრილის სტრიქონების ფიზიკური განლაგების ადგილმდებარეობაზე მიმთითებლებით

11.1.7. როცა სრულდება მოთხოვნა, რომელიც ინდექსირებულ სვეტს მიმართავს, სერვერი:

- ა. არავტომატურად აანალიზებს ინდექსს საჭირო მნიშვნელობის მოძებნის მიზნით
- ბ. ავტომატურად არ აანალიზებს ინდექსს საჭირო მნიშვნელობის მოძებნის მიზნით
- გ. ავტომატურად აანალიზებს ინდექსს საჭირო მნიშვნელობის მოძებნის მიზნით

11.1.8. როცა სრულდება სტრიქონების ცვლილება, მაშინ:

- ა. სრულდება მხოლოდ ინდექსების გაახლება
- ბ. სრულდება მხოლოდ მონაცემების გაახლება
- გ. საკუთრივ მონაცემების გაახლების გარდა, სრულდება ყველა ინდექსის გაახლება

11.1.9. ინდექსების გამოყენების უპირატესობაა:

- ა. მონაცემების ამორჩევის მნიშვნელოვანი დაჩქარება
- ბ. მონაცემების გაახლების (დამატების, შეცვლის და წაშლის) პროცესის შენელება
- გ. მონაცემების ამორჩევის მნიშვნელოვანი შენელება

11.1.10. ინდექსების გამოყენების ნაკლია:

- ა. მონაცემების ამორჩევის არც თუ მნიშვნელოვანი შენელება
- ბ. მონაცემების გაახლების (დამატების, შეცვლის და წაშლის) პროცესის შენელება
- გ. მონაცემების ამორჩევის მნიშვნელოვანი დაჩქარება

11.1.11. არ უნდა მოვახდინოთ იმ სვეტების ინდექსირება, რომლებიც:

- ა. არ თამაშობენ არანაირ როლს მოთხოვნის შესრულების დროს
- ბ. თამაშობენ არანაირ როლს მოთხოვნის შესრულების დროს
- გ. არ თამაშობენ უმნიშვნელო როლს მოთხოვნის შესრულების დროს

11.1.12. ძალიან გრძელი სვეტების ინდექსირება, რომელთა სიგრძეა რამდენიმე ათეული სიმბოლო:

- ა. არის სასურველი
- ბ. არ არის სასურველი
- გ. ხანდახან არის სასურველი

11.1.13. არსებობს ინდექსების:

- ა. ორი ტიპი: არაკლასტერული და უნიკალური
- ბ. ერთი ტიპი: უნიკალური
- გ. სამი ტიპი: კლასტერული, არაკლასტერული და უნიკალური

არაკლასტერული ინდექსი. კლასტერული ინდექსი. უნიკალური ინდექსი

11.2.1. არაკლასტერული ინდექსი:

- ა. შეიცავს ერთი ან მეტი სვეტის მნიშვნელობებს დალაგებულს ზრდადობით ან კლებადობით
- ბ. არ შეიცავს ერთი ან მეტი სვეტის მნიშვნელობებს დალაგებულს ზრდადობით ან კლებადობით
- გ. შეიცავს ერთი ან მეტი სვეტის მნიშვნელობებს დალაგებულს მხოლოდ ზრდადობით

11.2.2. არაკლასტერული ინდექსი:

- ა. არ შეიცავს სტრიქონებზე მიმართვებს
- ბ. შეიცავს სტრიქონებზე მიმართვებს
- გ. შეიცავს სვეტებზე მიმართვებს

11.2.3. არაკლასტერული ინდექსში მნიშვნელობების ძებნა::

- ა. სრულდება მხოლოდ ზრდადობით დალაგებულ უბანში
- ბ. არ სრულდება ზრდადობით ან კლებადობით დალაგებულ უბანში
- გ. სრულდება ზრდადობით ან კლებადობით დალაგებულ უბანში

11.2.4. არაკლასტერულ ინდექსში ნაპოვნი მნიშვნელობის გასწვრივ:

- ა. მოთავსებულია შესაბამის სტრიქონზე მიმართვა, რომელიც შეიცავს სტრიქონის რეალურ მისამართს ცხრილში
- ბ. არაა მოთავსებული შესაბამის სტრიქონზე მიმართვა, რომელიც შეიცავს სტრიქონის რეალურ მისამართს ცხრილში
- გ. მოთავსებულია შესაბამის სტრიქონზე მიმართვა, რომელიც არ შეიცავს სტრიქონის რეალურ მისამართს ცხრილში

11.2.5. არაკლასტერული ინდექსის გამოყენების შემთხვევაში, სერვერი:

- ა. არ მიმართავს ინდექსს და ასრულებს საჭირო სტრიქონის ძებნას
- ბ. ჯერ მიმართავს ინდექსს და შემდეგ ასრულებს საჭირო სტრიქონის ძებნას
- გ. მიმართავს ინდექსს და არ ასრულებს საჭირო სტრიქონის ძებნას

11.2.6. არაკლასტერული ინდექსის შექმნა:

- ა. უმჯობესია იშვიათად ცვალებადი სვეტებისთვის
- ბ. არაა უმჯობესი ხშირადცვალებადი სვეტებისთვის
- გ. უმჯობესია ხშირადცვალებადი სვეტებისთვის

11.2.7. ერთ ცხრილში:

- ა. შეიძლება განისაზღვროს 999-მდე არაკლასტერული ინდექსი
- ბ. არ შეიძლება განისაზღვროს 999-მდე არაკლასტერული ინდექსი
- გ. შეიძლება განისაზღვროს 100-მდე არაკლასტერული ინდექსი

11.2.8. ერთი არაკლასტერული ინდექსი:

- ა. არ შეიძლება შეიცავდეს 16-მდე სვეტს, რომლებიც ერთ ცხრილს უნდა ეკუთვნოდეს
- ბ. შეიძლება შეიცავდეს 16-მდე სვეტს, რომლებიც ერთ ცხრილს უნდა ეკუთვნოდეს
- გ. შეიძლება შეიცავდეს 16-მდე სვეტს, რომლებიც ერთ ცხრილს არ უნდა ეკუთვნოდეს

11.2.9. შედგენილ ინდექსში შემავალი სვეტების:

- ა. მაქსიმალურად დასაშვები საერთო ზომაა 500 ბაიტი
- ბ. მაქსიმალურად დასაშვები საერთო ზომაა 100 ბაიტი
- გ. მაქსიმალურად დასაშვები საერთო ზომაა 900 ბაიტი

11.2.10. შედგენილი ინდექსი შედგება:

- ა. რამდენიმე სვეტისგან
- ბ. ერთი სვეტისგან
- გ. ერთი სტრიქონისგან

11.2.11. ცხრილში კლასტერული შეიძლება იყოს:

- ა. მხოლოდ ერთი ინდექსი
- ბ. მხოლოდ ორი ინდექსი
- გ. მხოლოდ სამი ინდექსი

11.2.12. კლასტერული ინდექსის (clustered index) გამოყენების დროს ცხრილში მონაცემების:

- ა. ფიზიკური განლაგება არ გადაეწყობა ინდექსის სტრუქტურის შესაბამისად

- ბ. ფიზიკური განლაგება გადაეწყობა ინდექსის სტრუქტურის შესაბამისად
- გ. ლოგიკური განლაგება გადაეწყობა ინდექსის სტრუქტურის შესაბამისად

11.2.13. კლასტერული ინდექსები მნიშვნელოვნად:

- ა. ზრდიან მონაცემების ძებნის მწარმოებლურობას
- ბ. ამცირებენ მონაცემების ძებნის მწარმოებლურობას
- გ. ვერ ზრდიან მონაცემების ძებნის მწარმოებლურობას

11.2.14. კლასტერული ინდექსის გამოყენების შემთხვევაში, მონაცემების მომდევნო პორცია განლაგდება უშუალოდ მოძებნილი მონაცემების შემდეგ. შედეგად:

- ა. შესრულდება ინდექსთან მიმართვის და სტრიქონის ძებნის ზედმეტი ოპერაციები
- ბ. აღარ შესრულდება ინდექსთან მიმართვის და სტრიქონის ძებნის ზედმეტი ოპერაციები
- გ. სერვერი თავდაპირველად მიმართავს ინდექსს, შემდეგ კი პოულობს საჭირო სტრიქონს ცხრილში

11.2.15. კლასტერულ ინდექსად უნდა ავირჩიოთ ყველაზე:

- ა. ხშირად გამოყენებადი სვეტები
- ბ. იშვიათად გამოყენებადი სვეტები
- გ. ხშირად გამოყენებადი სტრიქონები

11.2.16. კლასტერული ინდექსი:

- ა. არ შეიძლება შეიცავდეს რამდენიმე სვეტს
- ბ. შეიძლება შეიცავდეს რამდენიმე სვეტს
- გ. შეიძლება შეიცავდეს მხოლოდ ერთ სვეტს

11.2.17. კლასტერული ინდექსში შემავალი სვეტების რაოდენობა შეძლებისდაგვარად:

- ა. ძალიან ბევრი უნდა იყოს
- ბ. ბევრი უნდა იყოს
- გ. მცირე უნდა იყოს

11.2.18. კლასტერული ინდექსი ხშირად ცვალებადი სვეტებისათვის:

- ა. უნდა გამოვიყენოთ
- ბ. არ უნდა გამოვიყენოთ
- გ. ხანდახან უნდა გამოვიყენოთ

11.2.19. ცხრილში პირველადი გასაღების შექმნისას სერვერი მისთვის ავტომატურად ქმნის კლასტერულ ინდექსს:

- ა. თუ ის იყო ადრე შექმნილი ან გასაღების განსაზღვრისას იყო აშკარად მითითებული ინდექსის სხვა ტიპი
- ბ. თუ ის არ იყო ადრე შექმნილი ან გასაღების განსაზღვრისას არ იყო აშკარად მითითებული ინდექსის სხვა ტიპი
- გ. თუ ის იყო ადრე შექმნილი ან გასაღების განსაზღვრისას იყო ავტომატურად მითითებული ინდექსის სხვა ტიპი

11.2.20. თუ ცხრილში განსაზღვრულია კლასტერული და არაკლასტერული ინდექსები, მაშინ:

- ა. კლასტერული ინდექსის მიმთითებელი მიმართავს სტრიქონის ლოგიკურ მდებარეობას,

რომელიც ამ სტრიქონს აღწერს

- ბ. კლასტერული ინდექსის მიმთითებელი მიმართავს სტრიქონის არა ფიზიკურ მდებარეობას, არამედ არაკლასტერული ინდექსის შესაბამის ელემენტს, რომელიც ამ სტრიქონს აღწერს
- გ. არაკლასტერული ინდექსის მიმთითებელი მიმართავს სტრიქონის არა ფიზიკურ მდებარეობას, არამედ კლასტერული ინდექსის შესაბამის ელემენტს, რომელიც ამ სტრიქონს აღწერს

11.2.21. თუ არაკლასტერული ინდექსის აგებისას კლასტერული ინდექსი არ არის უნიკალური, მაშინ სერვერი:

- ა. ავტომატურად უმატებს მას დამატებით მნიშვნელობებს, რომლებიც მას უნიკალურს ხდის
- ბ. ავტომატურად არ უმატებს მას დამატებით მნიშვნელობებს, რომლებიც მას უნიკალურს არ ხდის
- გ. არ უმატებს მას დამატებით მნიშვნელობებს

11.2.22. უნიკალური ინდექსები ინდექსირებულ ცხრილში მნიშვნელობების უნიკალურობის გარანტიას:

- ა. ხანდახან იძლევიან
- ბ. არ იძლევიან
- გ. იძლევიან

11.2.23. უნიკალური ინდექსი:

- ა. შეიძლება რეალიზებული იყოს როგორც კლასტერული, ისე არაკლასტერული ინდექსისათვის
- ბ. არ შეიძლება რეალიზებული იყოს როგორც კლასტერული, ისე არაკლასტერული ინდექსისათვის
- გ. შეიძლება რეალიზებული იყოს მხოლოდ კლასტერული ინდექსისათვის

11.2.24. ერთ ცხრილში შეიძლება არსებობდეს:

- ა. ერთი უნიკალური არაკლასტერული ინდექსი და რამდენიმე უნიკალური კლასტერული ინდექსი
- ბ. ერთი უნიკალური კლასტერული ინდექსი და რამდენიმე უნიკალური არაკლასტერული ინდექსი
- გ. რამდენიმე უნიკალური კლასტერული ინდექსი და მხოლოდ ერთი უნიკალური არაკლასტერული ინდექსი

11.2.25. მონაცემების მთლიანობის უზრუნველსაყოფად უმჯობესია:

- ა. UNIQUE ან PRIMARY KEY შეზღუდვების გამოყენება
- ბ. უნიკალური ინდექსის
- გ. არ გამოვიყენოთ ინდექსი

შევსების ფაქტორი

11.3.1. შევსების ფაქტორი განსაზღვრავს გვერდზე მონაცემების ჩაწერის:

- ა. სიმჭიდროვეს

- ბ. სიჩქარეს
- გ. სიხშირეს

11.3.2. შევსების ფაქტორი განსაზღვრავს:

- ა. მონაცემების ჩაწერის სიჩქარეს
- ბ. საინდექსო გვერდების თავისუფალი სივრცის რამდენი პროცენტი შეივსება მონაცემებით
- გ. საინდექსო გვერდების თავისუფალი სივრცის რამდენი პროცენტი არ შეივსება მონაცემებით

11.3.3. რაც მეტია შევსების ფაქტორით ნაკლები იქნება გვერდზე თავისუფალი სივრცე და მით უფრო კომპაქტურად იქნება განთავსებული ინფორმაცია ინდექსების შესახებ:

- ა. მით მეტი იქნება გვერდზე თავისუფალი სივრცე და მით უფრო ნაკლებ კომპაქტურად იქნება განთავსებული ინფორმაცია ინდექსების შესახებ
- ბ. მით მეტი იქნება გვერდზე თავისუფალი სივრცე და მით უფრო კომპაქტურად იქნება განთავსებული ინფორმაცია ინდექსების შესახებ
- გ. მით ნაკლები იქნება გვერდზე თავისუფალი სივრცე და მით უფრო კომპაქტურად იქნება განთავსებული ინფორმაცია ინდექსების შესახებ

11.3.4. შევსების ფაქტორის არჩევისას:

- ა. არ უნდა შევავსოთ რამდენად ინტენსიურად შესრულდება ცხრილში მონაცემების ცვლილება (დამატება, წაშლა და შეცვლა)
- ბ. უნდა შევავსოთ რამდენად ინტენსიურად შესრულდება ცხრილში მონაცემების ცვლილება (დამატება, წაშლა და შეცვლა)
- გ. უნდა შევავსოთ რამდენად ინტენსიურად შესრულდება ცხრილში მონაცემების მხოლოდ დამატება

11.3.5. თუ ცხრილი ძირითადად გამოიყენება მხოლოდ წაკითხვისათვის, მაშინ უმჯობესია შევსების ფაქტორი დავაყენოთ:

- ა. 100%-თან ახლოს
- ბ. 50%-თან ახლოს
- გ. 10%-თან ახლოს

11.3.6. თუ ცხრილში მონაცემები ხშირად იცვლება, მაშინ შევსების ფაქტორი:

- ა. უნდა იყოს ძალიან დიდი
- ბ. უნდა იყოს დიდი როგორც მონაცემებისათვის, ისე ინდექსებისათვის
- გ. არ უნდა იყოს დიდი როგორც მონაცემებისათვის, ისე ინდექსებისათვის

11.3.7. თუ ინტენსიურად გამოიყენება ცხრილში დავაყენებთ მაღალ შეფასების ფაქტორს, მაშინ სერვერი იძულებული იქნება:

- ა. ხშირად შეასრულოს გვერდების დახლეჩა ახალი მონაცემების ჩასმის მიზნით
- ბ. იშვიათად შეასრულოს გვერდების დახლეჩა ახალი მონაცემების ჩასმის მიზნით
- გ. არ შეასრულოს გვერდების დახლეჩა ახალი მონაცემების ჩასმის მიზნით

11.3.8. ცხრილში მონაცემების ცვლილებასა და დამატებასთან ერთად:

- ა. არ იცვლება გვერდების შევსების ხარისხი
- ბ. იცვლება გვერდების შევსების ხარისხი

გ. იცვლება გვერდების შევსების სისწრაფე

11.3.9. ინდექსის ზომის შემცირების მიზნით, პერიოდულად საჭირო:

- ა. ხდება ინდექსის წაშლა
- ბ. არ ხდება ინდექსის გადაწყობა
- გ. ხდება ინდექსის გადაწყობა

ინდექსის შექმნა

11.4.1. ინდექსის შექმნის უფლება აქვს:

- ა. მხოლოდ ფუნქციის მფლობელს
- ბ. მხოლოდ ცხრილის მფლობელს
- გ. მხოლოდ შენახული პროცედურის მფლობელს

11.4.2. ინდექსის შექმნის უფლების გადაცემა სხვა მომხმარებლისათვის:

- ა. არ შეიძლება
- ბ. შეიძლება
- გ. ხანდახან შეიძლება

11.4.3. პირველადი გასაღების შექმნისას:

- ა. ინდექსი ხანდახან იქმნება
- ბ. ინდექსი ავტომატურად არ იქმნება
- გ. ინდექსი ავტომატურად იქმნება

11.4.4. ინდექსის ავტომატურად იქმნება:

- ა. CREATE DATABASE ბრძანების შესრულებისას
- ბ. UNIQUE შეზღუდვის განსაზღვრისას
- გ. DROP TABLE ბრძანების შესრულებისას

11.4.5. ინდექსის შესაქმნელად გამოიყენება:

- ა. DROP TABLE ბრძანება
- ბ. CREATE DATABASE ბრძანება
- გ. CREATE INDEX ბრძანება

11.4.6. CREATE INDEX ბრძანებაში UNIQUE არგუმენტის მითითების შემთხვევაში:

- ა. არ შეიქმნება უნიკალური ინდექსი
- ბ. შეიქმნება უნიკალური ინდექსი
- გ. შეიქმნება არაუნიკალური ინდექსი

11.4.7. უნიკალური ინდექსის შექმნისას სერვერი:

- ა. ასრულებს სვეტის წინასწარ შემოწმებას მნიშვნელობების უნიკალურობაზე
- ბ. არ ასრულებს სვეტის წინასწარ შემოწმებას მნიშვნელობების უნიკალურობაზე
- გ. არაფერს არ აკეთებს

11.4.8. თუ უნიკალური ინდექსის შექმნისას სვეტში არის ორი ერთნაირი მნიშვნელობა, მაშინ:

- ა. ინდექსი არ შეიქმნება და არ გაიცემა შეტყობინება შეცდომის შესახებ

- ბ. ინდექსი შეიქმნება და გაიცემა შეტყობინება შეცდომის შესახებ
- გ. ინდექსი არ შეიქმნება და გაიცემა შეტყობინება შეცდომის შესახებ

11.4.9. თუ ინდექსი შედგენილია, ანუ შეიცავს ორ ან მეტ სვეტს, მაშინ უნიკალური:

- ა. არ უნდა იყოს ყველა საინდექსირებელი სვეტის მნიშვნელობების ერთობლიობა ცხრილის ან წარმოდგენის თითოეულ სტრიქონში
- ბ. უნდა იყოს ყველა საინდექსირებელი სვეტის მნიშვნელობების ერთობლიობა ცხრილის ან წარმოდგენის თითოეულ სტრიქონში
- გ. უნდა იყოს ყველა საინდექსირებელი სვეტის მნიშვნელობების ერთობლიობა ცხრილის ან წარმოდგენის თითოეულ სვეტში

11.4.10. საინდექსებელ სვეტში სასურველია, აგრეთვე, NULL მნიშვნელობის შენახვის აკრძალვა, რათა:

- ა. ავიცილოთ სტრიქონებში მონაცემების გამეორება
- ბ. ავიცილოთ სვეტებში მონაცემების გამეორება
- გ. ავიცილოთ მნიშვნელობების უნიკალურობასთან დაკავშირებული პრობლემები

11.4.11. თუ სვეტში გვხვდება ორი NULL მნიშვნელობა, მაშინ სერვერი მას ორ ერთნაირ მნიშვნელობად ჩათვლის:

- ა. ორ ერთნაირ მნიშვნელობად ჩათვლის
- ბ. ორ სხვადასხვა მნიშვნელობად ჩათვლის
- გ. სამ ერთნაირ მნიშვნელობად ჩათვლის

11.4.12. უნიკალური ინდექსის განსაზღვრის შემდეგ სერვერი არ შეგვასრულებინებს ისეთ INSERT ან UPDATE ბრძანებას, რომელიც გამოიწვევს:

- ა. ორი ერთნაირი მნიშვნელობის არსებობას
- ბ. ორი სხვადასხვა მნიშვნელობის არსებობას
- გ. სამი ერთნაირი მნიშვნელობის არსებობას

11.4.13. CREATE INDEX ბრძანებაში CLUSTERED არგუმენტი მიუთითებს, რომ შესაქმნელი ინდექსი იქნება:

- ა. არაკლასტერული
- ბ. კლასტერული
- გ. როგორც კლასტერული, ისე არაკლასტერული

11.4.14. CREATE INDEX ბრძანებაში NONCLUSTERED მიუთითებს, რომ შესაქმნელი ინდექსი იქნება:

- ა. არაკლასტერული
- ბ. კლასტერული
- გ. როგორც კლასტერული, ისე არაკლასტერული

11.4.15. თუ CREATE INDEX ბრძანებაში მითითებულია ერთზე მეტი სვეტი, მაშინ ინდექსი:

- ა. იქნება შედგენილი
- ბ. არ იქნება შედგენილი
- გ. იქნება შეუდგენელი

11.4.16.ერთი შედგენილი ინდექსი შეიძლება შეიცავდეს:

- ა. 64-მდე სვეტს
- ბ. 16-მდე სვეტს
- გ. 8-მდე სვეტს

11.4.17.ინდექსის აგება:

- ა. დასაშვებია text, ntext ან image ტიპის მქონე სვეტების ბაზაზე
- ბ. დაუშვებელია text, ntext ან image ტიპის მქონე სვეტების ბაზაზე
- გ. დაუშვებელია int და nvarchar ტიპის მქონე სვეტების ბაზაზე

11.4.18.ინდექსში სვეტების სახელების მითითების მიმდევრობა გავლენას:

- ა. ახდენს მოთხოვნების შესრულების სხშირეზე
- ბ. არ ახდენს მოთხოვნების შესრულების მწარმოებლურობაზე
- გ. ახდენს მოთხოვნების შესრულების მწარმოებლურობაზე

11.4.19.მაქსიმალური მწარმოებლურობის უზრუნველსაყოფად რეკომენდებულია:

- ა. მხოლოდ იმ სვეტების მითითება რომლებსაც მაქსიმალური სიგრძე აქვთ
- ბ. თავდაპირველად იმ სვეტების სახელების მითითება, რომლებსაც მაქსიმალური სიგრძე აქვთ, შემდეგ კი იმ სვეტების სახელების მითითება, რომლებსაც მინიმალური სიგრძე აქვთ
- გ. თავდაპირველად იმ სვეტების სახელების მითითება, რომლებსაც მინიმალური სიგრძე აქვთ, შემდეგ კი იმ სვეტების სახელების მითითება, რომლებსაც მაქსიმალური სიგრძე აქვთ

11.4.20.თუ CREATE INDEX ბრძანებაში [ASC | DESC] არგუმენტი ინდექსში:

- ა. არ განსაზღვრავს საკვანძო ელემენტების დახარისხების მეთოდს
- ბ. განსაზღვრავს საკვანძო ელემენტების დახარისხების მეთოდს
- გ. ინდექსის შექმნის მეთოდს

11.4.21.CREATE INDEX ბრძანებაში თუ მითითებულია ASC არგუმენტი, მაშინ მონაცემები დალაგდება:

- ა. კლებადობის მიხედვით
- ბ. ზრდადობის მიხედვით
- გ. არც ზრდადობით და არც კლებადობით

11.4.22.CREATE INDEX ბრძანებაში თუ მითითებულია DESC არგუმენტი, მაშინ მონაცემები დალაგდება:

- ა. კლებადობის მიხედვით
- ბ. ზრდადობის მიხედვით
- გ. არც ზრდადობით და არც კლებადობით

11.4.23.CREATE INDEX ბრძანებაში ავტომატურად:

- ა. იგულისხმება როგორც ASC, ისე DESC
- ბ. იგულისხმება DESC
- გ. იგულისხმება ASC

11.4.24.CREATE INDEX ბრძანებაში 'FILLFACTOR = შევსების_ფაქტორი' აგრუმენტი:

- ა. განსაზღვრავს საინდექსო გვერდების შევსების ხარისხს და აზრი არ აქვს მხოლოდ

ინდექსის შექმნის დროს

ბ. განსაზღვრავს საინდექსო გვერდების შევსების ხარისხს და აზრი აქვს მხოლოდ ინდექსის შექმნის დროს

გ. არ განსაზღვრავს საინდექსო გვერდების შევსების ხარისხს და აზრი აქვს მხოლოდ ინდექსის შექმნის დროს

11.4.25.თუ უნიკალური ინდექსის შექმნისას (როგორც კლასტერული, ისე არაკლასტერული) მითითებული იყო IGNORE_DUP_KEY არგუმენტი, მაშინ იმ ოპერაციის შესრულებისას, რომელიც იწვევს დუბლირებული მნიშვნელობების შექმნას, სერვერი:

ა. გასცემს შეტყობინებას შეცდომის შესახებ და გააუქმებს მხოლოდ იმ სტრიქონების შეცვლას, რომლებმაც დუბლირება გამოიწვიეს

ბ. არ გასცემს შეტყობინებას შეცდომის შესახებ და გააუქმებს მხოლოდ იმ სტრიქონების შეცვლას, რომლებმაც დუბლირება გამოიწვიეს

გ. გასცემს შეტყობინებას შეცდომის შესახებ და არ გააუქმებს მხოლოდ იმ სტრიქონების შეცვლას, რომლებმაც დუბლირება გამოიწვიეს

11.4.26.თუ უნიკალური ინდექსის შექმნისას (როგორც კლასტერული, ისე არაკლასტერული) IGNORE_DUP_KEY არგუმენტი არ იყო მითითებული, მაშინ იმ ოპერაციის შესრულებისას, რომელიც იწვევს დუბლირებული მნიშვნელობების შექმნას, სერვერი:

ა. არ გასცემს შეტყობინებას შეცდომის შესახებ, მაგრამ აუქმებს ყველა ცვლილებას

ბ. გასცემს შეტყობინებას შეცდომის შესახებ, მაგრამ აუქმებს ყველა ცვლილებას

გ. გასცემს შეტყობინებას შეცდომის შესახებ, მაგრამ არ აუქმებს ყველა ცვლილებას

11.4.27.თუ ცხრილში განსაზღვრული იყო ინდექსი იმავე სახელით, როგორც შესაქმნელ ინდექსს ექნება, მაშინ DROP_EXISTING არგუმენტის გამოყენება:

ა. გამოიწვევს არსებული ინდექსის ახლით შეცვლას და არ შეასრულებს მის გადაწყობას

ბ. არ გამოიწვევს არსებული ინდექსის ახლით შეცვლას და შეასრულებს მის გადაწყობას

გ. გამოიწვევს არსებული ინდექსის ახლით შეცვლას და შეასრულებს მის გადაწყობას

11.4.28.CREATE INDEX ბრძანებაში DROP_EXISTING არგუმენტის გამოყენება განსაკუთრებით ეფექტურია კლასტერული ინდექსების გადაწყობის დროს მაშინ, როცა ცხრილში:

ა. არაკლასტერული ინდექსების დიდი რაოდენობაა

ბ. კლასტერული ინდექსების დიდი რაოდენობაა

გ. კლასტერული და არაკლასტერული ინდექსების დიდი რაოდენობაა

11.4.29.თუ ჯერ წავშლით კლასტერულ ინდექსს, სერვერს მოუწევს ყველა:

ა. არაკლასტერული ინდექსის სტრუქტურის გადაწყობა, რათა შეცვალოს ცხრილის სტრიქონებზე მიმართვები

ბ. კლასტერული ინდექსის სტრუქტურის გადაწყობა, რათა შეცვალოს ცხრილის სტრიქონებზე მიმართვები

გ. კლასტერული და არაკლასტერული ინდექსის სტრუქტურის გადაწყობა, რათა შეცვალოს ცხრილის სტრიქონებზე მიმართვები

11.4.30.CREATE INDEX ბრძანებაში DROP_EXISTING არგუმენტის მითითების შემთხვევაში სერვერი:

ა. არ შეასრულებს არაკლასტერული ინდექსების შუალედურ გადაწყობას, რაც მკვეთრად

ზრდის ინდექსის შექმნის სიჩქარეს

ბ. შეასრულებს არაკლასტერული ინდექსების შუალედურ გადაწყობას, რაც მკვეთრად ზრდის ინდექსის შექმნის სიჩქარეს

გ. არ შეასრულებს არაკლასტერული ინდექსების შუალედურ გადაწყობას, რაც მკვეთრად ამცირებს ინდექსის შექმნის სიჩქარეს

11.4.31.CREATE INDEX ბრძანებაში DROP_EXISTING არგუმენტის გამოყენება ნებადართულია მაშინ, როცა მონაცემთა ბაზაში ცხრილისთვის ან წარმოდგენისთვის:

ა. არ არსებობს ამავე სახელის მქონე ინდექსი

ბ. უკვე არსებობს ამავე სახელის მქონე ინდექსი

გ. საერთოდ არ არსებობს ინდექსი

11.4.32.ინდექსი ყოველთვის იქმნება:

ა. მხოლოდ არამიმდინარე მონაცემთა ბაზის ცხრილის ან წარმოდგენისათვის

ბ. master მონაცემთა ბაზის ცხრილის ან წარმოდგენისათვის

გ. მხოლოდ მიმდინარე მონაცემთა ბაზის ცხრილის ან წარმოდგენისათვის

11.4.33.CREATE UNIQUE INDEX Ind_mobiluri ON Personal (mobiluri); ბრძანების შესრულების შედეგად:

ა. Personal ცხრილის mobiluri სვეტისთვის შეიქმნება უნიკალური Ind_mobiluri ინდექსი

ბ. Personal ცხრილის mobiluri სვეტისთვის არ შეიქმნება უნიკალური Ind_mobiluri ინდექსი

გ. Personal ცხრილის mobiluri სვეტისთვის შეიქმნება არაუნიკალური Ind_mobiluri ინდექსი

11.4.34.იმისათვის, რომ Personal ცხრილის mobiluri სვეტისთვის შევქმნათ უნიკალური Ind_mobiluri ინდექსი, უნდა შევასრულოთ ბრძანება:

ა. CREATE INDEX Ind_mobiluri ON Personal (mobiluri);

ბ. CREATE UNIQUE INDEX Ind_mobiluri ON Personal (mobiluri);

გ. CREATE UNIQUE INDEX Ind_mobiluri ON Personal (qalaqi);

11.4.35.CREATE NONCLUSTERED INDEX Ind_qalaqi_gvari

ON Personal ([qalaqi] DESC, [gvari] ASC) WITH FILLFACTOR = 40; ბრძანების შესრულების შედეგად:

ა. Personal ცხრილის qalaqi და gvari სვეტებისთვის შეიქმნება Ind_qalaqi_gvari არაკლასტერული, შედგენილი ინდექსი. შევსების ფაქტორია 60%. qalaqi სვეტი დალაგებულია ზრდადობით, gvari სვეტი კი - კლებადობით

ბ. Personal ცხრილის qalaqi და gvari სვეტებისთვის შეიქმნება Ind_qalaqi_gvari უნიკალური, კლასტერული, შედგენილი ინდექსი. შევსების ფაქტორია 40%. qalaqi სვეტი დალაგებულია კლებადობით, gvari სვეტი კი - ზრდადობით

გ. Personal ცხრილის qalaqi და gvari სვეტებისთვის შეიქმნება Ind_qalaqi_gvari არაკლასტერული, შედგენილი ინდექსი. შევსების ფაქტორია 40%. qalaqi სვეტი დალაგებულია კლებადობით, gvari სვეტი კი - ზრდადობით

**ინდექსების მართვა. ინდექსისთვის სახელის შეცვლა. ინდექსის წაშლა.
ინდექსების გადაწყობა. ინდექსის შესახებ ინფორმაციის მიღება**

11.5.1. ინდექსისთვის სახელის შესაცვლელად გამოიყენება:

- ა. sp_rename შენახული პროცედურა
- ბ. sp_help შენახული პროცედურა
- გ. DROP INDEX ბრძანება

11.5.2. ინდექსის წასაშლელად გამოიყენება:

- ა. sp_rename შენახული პროცედურა
- ბ. sp_help შენახული პროცედურა
- გ. DROP INDEX ბრძანება

11.5.3. ერთი DROP INDEX ბრძანებით:

- ა. შეიძლება რამდენიმე ინდექსის წაშლა ნებისმიერ ცხრილში
- ბ. არ შეიძლება რამდენიმე ინდექსის წაშლა ნებისმიერ ცხრილში
- გ. შეიძლება მხოლოდ ერთი ინდექსის წაშლა ნებისმიერ ცხრილში

11.5.4. დროთა განმავლობაში საინდექსო ცხრილების შევსების ხარისხი მნიშვნელოვნად იცვლება, რასაც:

- ა. არ მივყავართ ცხრილში მონაცემების დამატებისა და შეცვლის ოპერაციების მწარმოებლურობის დაქვეითებასთან
- ბ. მივყავართ ცხრილში მონაცემების დამატებისა და შეცვლის ოპერაციების მწარმოებლურობის ამაღლებასთან
- გ. მივყავართ ცხრილში მონაცემების დამატებისა და შეცვლის ოპერაციების მწარმოებლურობის დაქვეითებასთან

11.5.5. დროთა განმავლობაში საინდექსო ცხრილების შევსების ხარისხი მნიშვნელოვნად იცვლება, რასაც მივყავართ ცხრილში მონაცემების დამატებისა და შეცვლის ოპერაციების მწარმოებლურობის დაქვეითებასთან. ასეთ შემთხვევებში:

- ა. უნდა შევასრულოთ ინდექსის გადაწყობა საინდექსო გვერდებზე თავისუფალი ადგილის გაუქმების მიზნით
- ბ. უნდა შევასრულოთ ინდექსის გადაწყობა საინდექსო გვერდებზე თავისუფალი ადგილის მოწესრიგების მიზნით
- გ. არ უნდა შევასრულოთ ინდექსის გადაწყობა საინდექსო გვერდებზე თავისუფალი ადგილის მოწესრიგების მიზნით

11.5.6. ინდექსის გადაწყობის გზით:

- ა. შესაძლებელია ინდექსში შემავალი სვეტების შემადგენლობის შეცვლა
- ბ. შესაძლებელია ინდექსში შემავალი სვეტების შემადგენლობის გაუქმება
- გ. შეუძლებელია ინდექსში შემავალი სვეტების შემადგენლობის შეცვლა

11.5.7. ინდექსის გადაწყობის ერთ-ერთი გზაა:

- ა. ინდექსისთვის სახელის შეცვლა
- ბ. ინდექსის წაშლა
- გ. ინდექსის წაშლა და მისი ხელახლა შექმნა

11.5.8. ინდექსის გადაწყობის ეფექტური გზაა:

- ა. ინდექსის წაშლა და მისი ხელახლა შექმნა

- ბ. CREATE INDEX ბრძანებაში DROP_EXISTING არგუმენტის მითითება
- გ. ინდექსისთვის სახელის შეცვლა

11.5.9. ინდექსის გადაწყობის მესამე გზაა:

- ა. DBCC DBREINDEX ბრძანების გამოყენება
- ბ. sp_rename შენახული პროცედურის გამოყენება
- გ. sp_help შენახული პროცედურის გამოყენება

11.5.10. ინდექსების შესახებ ინფორმაციის მისაღებად გამოიყენება:

- ა. sp_helpindex შენახული პროცედურა
- ბ. sp_help შენახული პროცედურის გამოყენება
- გ. sp_rename შენახული პროცედურის გამოყენება

11.5.11. ინდექსის თვისებების სანახავად გამოიყენება:

- ა. sp_rename შენახული პროცედურის გამოყენება
- ბ. INDEXPROPERTY ბრძანება
- გ. DBCC DBREINDEX ბრძანების გამოყენება

11.5.12. თუ ინდექსის თვისება დაყენებულია, მაშინ გაიცემა:

- ა. NULL
- ბ. 0
- გ. 1

11.5.13. თუ ინდექსის თვისება დაყენებული არაა, მაშინ გაიცემა:

- ა. NULL
- ბ. 0
- გ. 1

11.5.14. თუ ინდექსის თვისების მდგომარეობის განსაზღვრა შეუძლებელია, მაშინ გაიცემა:

- ა. NULL
- ბ. 0
- გ. 1

11.5.15. თუ ინდექსის Is Clustered თვისება 1 მდგომარეობაშია, მაშინ ინდექსი:

- ა. კლასტერულია
- ბ. არაკლასტერულია
- გ. განუსაზღვრელია

11.5.16. თუ ინდექსის Is Clustered თვისება 0 მდგომარეობაშია, მაშინ ინდექსი:

- ა. კლასტერულია
- ბ. არაკლასტერულია
- გ. განუსაზღვრელია

11.5.17. თუ ინდექსის Is Unique თვისება 1 მდგომარეობაშია, მაშინ ინდექსი:

- ა. არის უნიკალურიც და არაუნიკალურიც
- ბ. არ არის უნიკალური

გ. უნიკალურია

11.5.18.თუ ინდექსის IsUnique თვისება 0 მდგომარეობაშია, მაშინ ინდექსი:

- ა. არ არის უნიკალური
- ბ. უნიკალურია
- გ. არის უნიკალურიც და არაუნიკალურიც

11.5.19.EXEC sp_rename 'Personali.Ind_ganyofileba', 'IndGanyofileba', 'INDEX'; შენახული პროცედურის შესრულების შედეგად:

- ა. Personali ცხრილში Ind_ganyofileba ინდექსს არ დაერქმევა ახალი IndGanyofileba სახელი
- ბ. Personali ცხრილში Ind_ganyofileba ინდექსს დაერქმევა ახალი IndGanyofileba სახელი
- გ. Personali ცხრილში IndGanyofileba ინდექსს დაერქმევა ახალი Ind_ganyofileba სახელი

11.5.20.იმისათვის, რომ Personali ცხრილში Ind_ganyofileba ინდექსს დაერქმევა ახალი IndGanyofileba სახელი, უნდა შევასრულოთ ბრძანება:

- ა. sp_rename 'Personali.Ind_ganyofileba', 'IndGanyofileba';
- ბ. sp_rename 'Personali.Ind_ganyofileba', 'IndGanyofileba', 'COLUMN';
- გ. sp_rename 'Personali.Ind_ganyofileba', 'IndGanyofileba', 'INDEX';

11.5.21.DROP INDEX Personali.Ind_qalaqi_gvari; ბრძანების შესრულების შედეგად:

- ა. Personali ცხრილიდან წაიშლება Ind_qalaqi_gvari ინდექსი
- ბ. Personali ცხრილიდან არ წაიშლება Ind_qalaqi_gvari ინდექსი
- გ. Personali ცხრილიდან წაიშლება Ind_qalaqi_gvari სვეტი

11.5.22.იმისათვის, რომ Personali ცხრილიდან წავშალოთ Ind_qalaqi_gvari ინდექსი, უნდა შევასრულოთ ბრძანება:

- ა. CREATE INDEX Personali.Ind_qalaqi_gvari;
- ბ. DROP Personali.Ind_qalaqi_gvari;
- გ. DROP INDEX Personali.Ind_qalaqi_gvari;

11.5.23.DBCC DBREINDEX ('Personali', PK_Personali, 80); ბრძანების შესრულების შედეგად:

- ა. შესრულდება Personali ცხრილის PK_Personali ინდექსის გადაწყობა. შევსების ფაქტორია 80
- ბ. არ შესრულდება Personali ცხრილის PK_Personali ინდექსის გადაწყობა. შევსების ფაქტორია 80
- გ. შესრულდება Personali ცხრილის PK_Personali10 ინდექსის გადაწყობა. შევსების ფაქტორია 60

11.5.24.იმისათვის, რომ გადავაწყოთ Personali ცხრილის PK_Personali ინდექსი, უნდა შევასრულოთ ბრძანება:

- ა. DBCC DBREINDEX ('Personali', Ind_Personali);
- ბ. DBCC DBREINDEX ('Personali', PK_Personali);
- გ. DBREINDEX ('Personali', PK_Personali);

11.5.25.EXEC sp_helpindex 'Personali'; ბრძანების შესრულების შედეგად:

- ა. გაიცემა ინფორმაცია Shemkveti ცხრილის ინდექსების შესახებ

- ბ. არ გაიცემა ინფორმაცია Personali ცხრილის ინდექსების შესახებ
- გ. გაიცემა ინფორმაცია Personali ცხრილის ინდექსების შესახებ

თავი 12. წარმოღვენები

შესავალი

12.1.1. წარმოდგენა არის:

- ა. ლოგიკური ცხრილი
- ბ. ფიზიკური ცხრილი
- გ. ვირტუალური ცხრილი

12.1.2. წარმოდგენა შედგება სვეტებისა და სტრიქონებისაგან, რომლებიც:

- ა. დინამიურად ამოირჩევა ერთი ან მეტი ცხრილიდან და/ან წარმოდგენიდან
- ბ. სტატიკურად ამოირჩევა ერთი ან მეტი ცხრილიდან ან წარმოდგენიდან
- გ. დინამიურად ამოირჩევა მხოლოდ ორი ცხრილიდან

12.1.3. ფიზიკურად წარმოდგენას აქვს:

- ა. INSERT მოთხოვნის სახე, რომლის საფუძველზეც სრულდება მონაცემების ამორჩევა
- ბ. SELECT მოთხოვნის სახე, რომლის საფუძველზეც სრულდება მონაცემების ამორჩევა
- გ. UPDATE მოთხოვნის სახე, რომლის საფუძველზეც სრულდება მონაცემების ამორჩევა

12.1.4. თუ წარმოდგენაში ჩართული არ არის ცხრილის რომელიმე სვეტი, მაშინ ცხრილზე დადებულია:

- ა. ვერტიკალური ფილტრი
- ბ. ჰორიზონტალური ფილტრი
- გ. დიაგონალური ფილტრი

12.1.5. თუ მოთხოვნა შეიცავს სტრიქონების ამორჩევის პირობებს, მაშინ ცხრილზე დადებულია:

- ა. დიაგონალური ფილტრი
- ბ. ჰორიზონტალური ფილტრი
- გ. ვერტიკალური ფილტრი

12.1.6. წარმოდგენა ბმული ცხრილების სვეტებს:

- ა. შეიძლება შეიცავდეს
- ბ. არ შეიძლება შეიცავდეს
- გ. არ უნდა შეიცავდეს

12.1.7. წარმოდგენასთან მიმართვის დროს სერვერი:

- ა. არ ამოწმებს იმ ობიექტების არსებობას, რომლებიც საჭიროა წარმოდგენის განმსაზღვრელი SELECT მოთხოვნის შესასრულებლად
- ბ. ამოწმებს იმ ობიექტების არსებობას, რომლებიც საჭიროა წარმოდგენის განმსაზღვრელი UPDATE მოთხოვნის შესასრულებლად
- გ. ამოწმებს იმ ობიექტების არსებობას, რომლებიც საჭიროა წარმოდგენის განმსაზღვრელი SELECT მოთხოვნის შესასრულებლად

12.1.8. თუ SELECT მოთხოვნაში მითითებული რომელიმე ცხრილი წაშლილია, მაშინ წარმოდგენა:

- ა. ვერ იმუშავებს და არ გაიცემა შეტყობინება შეცდომის შესახებ
- ბ. მაინც იმუშავებს და გაიცემა შეტყობინება შეცდომის შესახებ

გ. ვერ იმუშავებს და გაიცემა შეტყობინება შეცდომის შესახებ

12.1.9. წარმოდგენა ყოველთვის იქმნება მიმდინარე მონაცემთა ბაზაში:

- ა. model მონაცემთა ბაზაში
- ბ. master მონაცემთა ბაზაში
- გ. მიმდინარე მონაცემთა ბაზაში

12.1.10. განაწილებული მოთხოვნების გამოყენებით:

- ა. შეგვიძლია მივმართოთ მიმდინარე სერვერის სხვა მონაცემთა ბაზებში შექმნილ ცხრილებს და არ შეგვიძლია მივმართოთ წარმოდგენებს
- ბ. არ შეგვიძლია მივმართოთ მიმდინარე სერვერის სხვა მონაცემთა ბაზებში შექმნილ ცხრილებსა და წარმოდგენებს
- გ. შეგვიძლია მივმართოთ მიმდინარე სერვერის სხვა მონაცემთა ბაზებში შექმნილ ცხრილებსა და წარმოდგენებს

12.1.11. ერთი მომხმარებლის წარმოდგენებს:

- ა. ხანდახან შეიძლება ერთნაირი სახელები ჰქონდეს
- ბ. ერთნაირი სახელები უნდა ჰქონდეს
- გ. სხვადასხვა სახელები უნდა ჰქონდეს

12.1.12. წარმოდგენისა და ცხრილის სახელი ერთმანეთს:

- ა. ხანდახან შეიძლება ემთხვეოდეს
- ბ. არ უნდა ემთხვეოდეს
- გ. უნდა ემთხვეოდეს

12.1.13. თუ წარმოდგენაში სვეტი მიღებულია არითმეტიკული გამოსახულების გამოთვლის გზით, მაშინ:

- ა. წარმოდგენის სვეტს სახელი უნდა მივანიჭოთ
- ბ. წარმოდგენის სტრიქონს სახელი უნდა მივანიჭოთ
- გ. წარმოდგენის სვეტს სახელი არ უნდა მივანიჭოთ

12.1.14. თუ წარმოდგენაში ორ ან მეტ სვეტს შეიძლება ჰქონდეს ერთნაირი სახელები, მაშინ:

- ა. წარმოდგენის სვეტს სახელი უნდა მივანიჭოთ
- ბ. წარმოდგენის სვეტს სახელი არ უნდა მივანიჭოთ
- გ. წარმოდგენის სტრიქონს სახელი უნდა მივანიჭოთ

12.1.15. წარმოდგენა დროებით ცხრილებს:

- ა. შეიძლება მიმართავდეს
- ბ. არ შეიძლება მიმართავდეს
- გ. ხანდახან შეიძლება მიმართავდეს

12.1.16. დროებითი წარმოდგენის შექმნა:

- ა. შეიძლება
- ბ. ხანდახან შეიძლება
- გ. არ შეიძლება

12.1.17.რადგან წარმოდგენა მონაცემებს სხვადასხვა ცხრილებიდან და წარმოდგენებიდან ირჩევს, ამიტომ მისთვის:

- ა. არ შეიძლება განისაზღვროს მთლიანობაზე შეზღუდვები, ტრიგერები და ავტომატური მნიშვნელობები
- ბ. შეიძლება განისაზღვროს მთლიანობაზე შეზღუდვები, ტრიგერები და ავტომატური მნიშვნელობები
- გ. არ შეიძლება განისაზღვროს მხოლოდ მთლიანობაზე შეზღუდვები

12.1.18.წარმოდგენისთვის ჩვეულებრივი ან სრულტექსტოვანი ინდექსის შექმნა:

- ა. შეიძლება
- ბ. არ შეიძლება
- გ. ხანდახან შეიძლება

12.1.19.წარმოდგენის სვეტებისათვის ნაგულისხმევი მნიშვნელობები:

- ა. ხანდახან შეიძლება განვსაზღვროთ
- ბ. შეიძლება განვსაზღვროთ
- გ. არ შეიძლება განვსაზღვროთ

12.1.20.წარმოდგენისთვის SELECT მოთხოვნის განსაზღვრისას:

- ა. არ შეიძლება მხოლოდ OPTION, ORDER BY განყოფილებების გამოყენება
- ბ. შეიძლება OPTION, ORDER BY, COMPUTE BY, COMPUTE და INTO განყოფილებების გამოყენება
- გ. არ შეიძლება OPTION, ORDER BY, COMPUTE BY, COMPUTE და INTO განყოფილებების გამოყენება

12.1.21.წარმოდგენისთვის SELECT მოთხოვნის განსაზღვრისას ORDER BY განყოფილების გამოყენება:

- ა. შეიძლება იმ შემთხვევაში, თუ მითითებულია OPTION სიტყვა
- ბ. არ შეიძლება იმ შემთხვევაში, თუ მითითებულია TOP სიტყვა
- გ. შეიძლება იმ შემთხვევაში, თუ მითითებულია TOP სიტყვა

12.1.22.წარმოდგენის საშუალებით:

- ა. არ შეიძლება მონაცემების ცვლილება იმ ცხრილებში, რომლებიც წარმოდგენაში მონაწილეობენ
- ბ. შეიძლება მონაცემების ცვლილება იმ ცხრილებში, რომლებიც წარმოდგენაში მონაწილეობენ
- გ. შეიძლება მონაცემების ცვლილება იმ ცხრილებში, რომლებიც წარმოდგენაში არ მონაწილეობენ

12.1.23.წარმოდგენის საშუალებით რამდენიმე ცხრილში მონაცემების შეცვლა:

- ა. არ შეიძლება
- ბ. შეიძლება
- გ. ხანდახან შეიძლება

12.1.24.ერთმა UPDATE ან INSERT ბრძანებამ მონაცემები წარმოდგენის:

- ა. მხოლოდ ერთ ცხრილში უნდა შეცვალოს

- ბ. მხოლოდ ორ ცხრილში უნდა შეცვალოს
- გ. მხოლოდ სამ ცხრილში უნდა შეცვალოს

წარმოდგენის შექმნა

12.2.1. წარმოდგენის შესაქმნელად გამოიყენება:

- ა. CREATE TABLE ბრძანება
- ბ. CREATE INDEX ბრძანება
- გ. CREATE VIEW ბრძანება

12.2.2. ავტომატურად, წარმოდგენაში სვეტების სახელები:

- ა. ემთხვევა საწყისი ცხრილების სვეტების სახელებს
- ბ. არ ემთხვევა საწყისი ცხრილების სვეტების სახელებს
- გ. ემთხვევა საწყისი ცხრილების სტრიქონების სახელებს

12.2.3. წარმოდგენაში სვეტის სახელი აშკარად უნდა მივუთითოთ შემდეგ შემთხვევებში:

- ა. 1. როცა სვეტის მნიშვნელობა არ ფორმირდება არითმეტიკული გამოსახულების, ფუნქციის ან მუდმივას საფუძველზე; 2. როცა წარმოდგენის სვეტს არ უნდა ჰქონდეს იმ სვეტისაგან განსხვავებული სახელი, რომლისგანაც ის მიიღება
- ბ. 1. როცა სვეტის მნიშვნელობა ფორმირდება არითმეტიკული გამოსახულების, ფუნქციის ან მუდმივას საფუძველზე; 2. როცა ორ ან მეტ სვეტს აქვს ერთნაირი სახელი; 3. როცა წარმოდგენის სვეტს უნდა ჰქონდეს იმ სვეტისაგან განსხვავებული სახელი, რომლისგანაც ის მიიღება
- გ. 1. როცა ორ ან მეტ სვეტს არ აქვს ერთნაირი სახელი; 2. როცა წარმოდგენის სვეტს უნდა ჰქონდეს იმ სვეტისაგან განსხვავებული სახელი, რომლისგანაც ის მიიღება

12.2.4. CREATE VIEW ბრძანების ENCRYPTION არგუმენტი მიუთითებს, რომ:

- ა. არ უნდა მოხდეს წარმოდგენის კოდის დაშიფვრა
- ბ. უნდა მოხდეს წარმოდგენის კოდის გაშიფვრა
- გ. უნდა მოხდეს წარმოდგენის კოდის დაშიფვრა

12.2.5. CREATE VIEW ბრძანების SCHEMABINDING არგუმენტი თუ მითითებულია წარმოდგენის შექმნის დროს, მაშინ სერვერი წარმოდგენის სტრუქტურას:

- ა. დააკავშირებს იმ ობიექტების სტრუქტურასთან, რომლებსაც UPDATE მოთხოვნა მიმართავს
- ბ. არ დააკავშირებს იმ ობიექტების სტრუქტურასთან, რომლებსაც SELECT მოთხოვნა მიმართავს
- გ. დააკავშირებს იმ ობიექტების სტრუქტურასთან, რომლებსაც SELECT მოთხოვნა მიმართავს

12.2.6. CREATE VIEW ბრძანების WITH CHECK OPTION არგუმენტი მიუთითებს, რომ:

- ა. არ უნდა შესრულდეს იმ ცვლილებების შემოწმება, რომლებიც წარმოდგენის საშუალებით შესრულდება
- ბ. უნდა შესრულდეს იმ ცვლილებების შემოწმება, რომლებიც წარმოდგენის საშუალებით შესრულდება
- გ. უნდა შესრულდეს იმ ცვლილებების შემოწმება, რომლებიც წარმოდგენის საშუალებით

არ შესრულდება

12.2.7. წარმოდგენის სახელი ინახება:

- ა. მიმდინარე მონაცემთა ბაზის sys.objects სისტემურ წარმოდგენაში
- ბ. მიმდინარე მონაცემთა ბაზის syscomments სისტემურ წარმოდგენაში
- გ. master მონაცემთა ბაზის sys.objects სისტემურ წარმოდგენაში

12.2.8. წარმოდგენის კოდი ინახება:

- ა. master მონაცემთა ბაზის sys.objects სისტემურ წარმოდგენაში
- ბ. მიმდინარე მონაცემთა ბაზის syscomments სისტემურ წარმოდგენაში
- გ. model მონაცემთა ბაზის sys.objects სისტემურ წარმოდგენაში

12.2.9. ინფორმაცია წარმოდგენის საწყის ცხრილებზე დამოკიდებულების შესახებ ინახება:

- ა. მიმდინარე მონაცემთა ბაზის syscomments სისტემურ წარმოდგენაში
- ბ. მიმდინარე მონაცემთა ბაზის sys.objects სისტემურ წარმოდგენაში
- გ. მიმდინარე მონაცემთა ბაზის sysdepends სისტემურ წარმოდგენაში

12.2.10.CREATE VIEW View_1

```
AS  
SELECT *  
FROM Xelshekruleba  
WHERE vali_1 > 0;
```

ბრძანების შესრულების შედეგად:

- ა. შეიქმნება წარმოდგენა, რომელშიც გამოჩნდება Xelshekruleba ცხრილის ყველა სვეტი და ინფორმაცია მევალების შესახებ
- ბ. არ შეიქმნება წარმოდგენა, რომელშიც გამოჩნდება Xelshekruleba ცხრილის ყველა სვეტი და ინფორმაცია მევალების შესახებ
- გ. შეიქმნება წარმოდგენა, რომელშიც არ გამოჩნდება Xelshekruleba ცხრილის ყველა სვეტი და ინფორმაცია მევალების შესახებ

12.2.11. იმისათვის, რომ შევქმნათ წარმოდგენა, რომელშიც გამოჩნდება Xelshekruleba ცხრილის ყველა სვეტი და ინფორმაცია მევალების შესახებ, უნდა შევასრულოთ ბრძანება:

ა. CREATE VIEW View_1
AS

```
SELECT *  
FROM Xelshekruleba  
WHERE vali_1 < 0;
```

ბ. CREATE VIEW View_1
AS

```
SELECT *  
FROM Xelshekruleba  
WHERE vali_1 > 0;
```

გ. CREATE VIEW View_1
AS

```
SELECT xelshekrulebaID, kursi, tarigi_dawyebis  
FROM Xelshekruleba
```

WHERE vali_1 > 0;

წარმოდგენების მართვა. წარმოდგენის სახელის შეცვლა. წარმოდგენის წაშლა. წარმოდგენის შესახებ ინფორმაციის მიღება. წარმოდგენის დამოკიდებულებების ნახვა

12.3.1. წარმოდგენის სახელის შესაცვლელად გამოიყენება:

- ა. sp_rename ბრძანება
- ბ. DBCC DBREINDEX ბრძანება
- გ. CREATE TABLE ბრძანება

12.3.2. წარმოდგენის წასაშლელად გამოიყენება:

- ა. CREATE VIEW ბრძანება
- ბ. DROP TABLE ბრძანება
- გ. DROP VIEW ბრძანება

12.3.3. წარმოდგენის წაშლის უფლება:

- ა. არ აქვს მხოლოდ მის მფლობელს და ეს უფლება არ შეიძლება სხვას გადაეცეს
- ბ. აქვს მხოლოდ მის მფლობელს და ეს უფლება არ შეიძლება სხვას გადაეცეს
- გ. აქვს მხოლოდ მის მფლობელს და ეს უფლება შეიძლება სხვას გადაეცეს

12.3.4. თუ საჭიროა, რომ სხვა მომხმარებელმა წაშალოს წარმოდგენა, მაშინ მას:

- ა. უნდა გადავცეთ წარმოდგენის ფლობის უფლება
- ბ. არ უნდა გადავცეთ წარმოდგენის ფლობის უფლება
- გ. უნდა ჩამოვართვათ წარმოდგენის ფლობის უფლება

12.3.5. წარმოდგენის შესახებ ინფორმაციის მისაღებად გამოიყენება:

- ა. CREATE VIEW ბრძანება
- ბ. sp_help შენახული პროცედურა
- გ. sp_rename ბრძანება

12.3.6. კოდის მისაღებად, რომლის საშუალებითაც შეიქმნა წარმოდგენა, შეგვიძლია გამოვიყენოთ:

- ა. sp_rename ბრძანება
- ბ. CREATE VIEW ბრძანება
- გ. sp_helptext შენახული პროცედურა

12.3.7. იმ ობიექტების სანახავად, რომლებზეც წარმოდგენაა დამოკიდებული, შეგვიძლია:

- ა. sp_helptext შენახული პროცედურა
- ბ. sp_depends შენახული პროცედურის გამოყენება
- გ. sp_rename ბრძანება

12.3.8. DROP VIEW View_1; ბრძანების შესრულების შედეგად:

- ა. არ წაიშლება View_1 წარმოდგენა
- ბ. შეიქმნება View_1 წარმოდგენა

გ. წაიშლება View_1 წარმოდგენა

12.3.9. View_1 წარმოდგენის წასაშლელად უნდა შევასრულოთ ბრძანება:

- ა. DROP VIEW View_1;
- ბ. sp_rename View_1;
- გ. CREATE VIEW View_1;

12.3.10. EXEC sp_help 'View_1'; ბრძანების შესრულების შედეგად:

- ა. არ გაიცემა ცხრილი, რომელიც შეიცავს View_1 წარმოდგენის სვეტებს მათი თვისებების აღწერით
- ბ. გაიცემა ცხრილი, რომელიც შეიცავს View_1 წარმოდგენის სვეტებს მათი თვისებების აღწერით
- გ. გაიცემა ცხრილი, რომელიც არ შეიცავს View_1 წარმოდგენის სვეტებს მათი თვისებების აღწერით

12.3.11. ცხრილის მისაღებად, რომელიც შეიცავს View_1 წარმოდგენის სვეტებს მათი თვისებების აღწერით, უნდა შევასრულოთ მოთხოვნა:

- ა. EXEC sp_rename 'View_1';
- ბ. DROP 'View_1';
- გ. EXEC sp_help 'View_1';

12.3.12. EXEC sp_helptext 'View_1'; ბრძანების შესრულების შედეგად:

- ა. გაიცემა View_1 წარმოდგენის შესაბამისი პროგრამული კოდი
- ბ. არ გაიცემა View_1 წარმოდგენის შესაბამისი პროგრამული კოდი
- გ. შეიქმნება View_1 წარმოდგენა

12.3.13. View_1 წარმოდგენის შესაბამისი პროგრამული კოდის მისაღებად, უნდა შევასრულოთ ბრძანება:

- ა. EXEC sp_rename 'View_1';
- ბ. EXEC sp_helptext 'View_1';
- გ. EXEC sp_help 'View_1';

12.3.14. EXEC sp_depends 'View_1'; ბრძანების შესრულების შედეგად:

- ა. გაიცემა იმ ობიექტების სია, რომლებზეც View_1 წარმოდგენა არ არის დამოკიდებული
- ბ. არ გაიცემა იმ ობიექტების სია, რომლებზეც View_1 წარმოდგენაა დამოკიდებული
- გ. გაიცემა იმ ობიექტების სია, რომლებზეც View_1 წარმოდგენაა დამოკიდებული

12.3.15. იმ ობიექტების სიის მისაღებად, რომლებზეც View_1 წარმოდგენაა დამოკიდებული, უნდა შევასრულოთ ბრძანება:

- ა. EXEC sp_depends 'View_1';
- ბ. EXEC sp_helptext 'View_1';
- გ. EXEC sp_help 'View_1';

თავი 13. კურსორები

შესავალი. კურსორების რეალიზება

13.1.1. მოთხოვნის შესრულების შედეგად სერვერს:

- ა. არ შეუძლია კლიენტის პროგრამას ასობით ათასი სტრიქონი დაუბრუნოს
- ბ. შეუძლია კლიენტის პროგრამას მხოლოდ ერთი სტრიქონი დაუბრუნოს
- გ. შეუძლია კლიენტის პროგრამას ასობით ათასი სტრიქონი დაუბრუნოს

13.1.2. კურსორი კლიენტის პროგრამას საშუალებას აძლევს იმუშაოს:

- ა. არა ასობით ან ათასობით სტრიქონთან, არამედ ერთ სტრიქონთან ან სტრიქონების მცირე ბლოკთან
- ბ. ასობით ან ათასობით სტრიქონთან და არა ერთ სტრიქონთან ან სტრიქონების მცირე ბლოკთან
- გ. ასობით ან ათასობით სტრიქონთან

13.1.3. სტრიქონების შედეგობრივი ნაკრები:

- ა. არის მონაცემთა სრული ნაკრების ნაწილი, გაფილტრული ჰორიზონტალურად WHERE განყოფილების გამოყენებით
- ბ. არ არის მონაცემთა სრული ნაკრების ნაწილი, გაფილტრული ჰორიზონტალურად WHERE განყოფილების გამოყენებით
- გ. არის მონაცემთა სრული ნაკრების ნაწილი, გაფილტრული ვერტიკალურად ORDER BY განყოფილების გამოყენებით

13.1.4. კურსორები:

- ა. არ მუშაობენ მონაცემების შედეგობრივ ნაკრებთან და ჩვენ საშუალებას არ გვაძლევენ ვიმუშაოთ ამ ნაკრების სხვადასხვა ნაწილებთან
- ბ. მუშაობენ მონაცემების შედეგობრივ ნაკრებთან და ჩვენ საშუალებას გვაძლევენ ვიმუშაოთ ამ ნაკრების სხვადასხვა ნაწილებთან
- გ. მუშაობენ მონაცემების შედეგობრივ ნაკრებთან და ჩვენ საშუალებას არ გვაძლევენ ვიმუშაოთ ამ ნაკრების სხვადასხვა ნაწილებთან

13.1.5. კურსორების საშუალებით მონაცემების დამუშავების სიჩქარე გაცილებით:

- ა. მაღალია მონაცემთა ბაზის სტანდარტულ საშუალებებთან შედარებით
- ბ. მაღალია სერვერის სტანდარტულ საშუალებებთან შედარებით
- გ. დაბალია სერვერის სტანდარტულ საშუალებებთან შედარებით

13.1.6. არსებობს სამი სახის კურსორი:

- ა. ფუნქციების კურსორები და Transact-SQL-ის კურსორები
- ბ. Transact-SQL-ის კურსორები, სერვერის კურსორები და კლიენტის კურსორები
- გ. ტრიგერის კურსორები, სერვერის კურსორები და კლიენტის კურსორები

13.1.7. Transact-SQL-ის კურსორები ძირითადად:

- ა. მხოლოდ წარმოდგენების შიგნით გამოიყენება
- ბ. ინდექსების, შენახული პროცედურებისა და Transact-SQL-ის კოდის შიგნით გამოიყენება
- გ. ტრიგერების, შენახული პროცედურებისა და Transact-SQL-ის კოდის შიგნით გამოიყენება

13.1.8. სერვერის კურსორები ახდენენ პროგრამა-დანართების პროგრამული ინტერფეისის რეალიზებას:

- ა. ODBC, OLE DB და DB-Library სისტემებისთვის
- ბ. ტრიგერებისთვის
- გ. ინდექსებისთვის

13.1.9. კლიენტის კურსორების რეალიზებას:

- ა. არ ახდენს კლიენტი
- ბ. ახდენს კლიენტი
- გ. ახდენს სერვერი

13.1.10. კლიენტის კურსორი:

- ა. იღებს სერვერის მიერ გაცემულ შედეგობრივ სტრიქონებს და მათ ლოკალურად არ ინახავს
- ბ. არ იღებს სერვერის მიერ გაცემულ შედეგობრივ სტრიქონებს და მათ ლოკალურად ინახავს
- გ. იღებს სერვერის მიერ გაცემულ შედეგობრივ სტრიქონებს და მათ ლოკალურად ინახავს

13.1.11. ერთი კურსორი:

- ა. შეიძლება მუშაობდეს სხვადასხვა ბაზაში მოთავსებულ რამდენიმე ცხრილთან
- ბ. არ შეიძლება მუშაობდეს სხვადასხვა ბაზაში მოთავსებულ რამდენიმე ცხრილთან
- გ. შეიძლება მუშაობდეს მხოლოდ ერთ ცხრილთან

13.1.12. კურსორში მოთავსებული სტრიქონების წაკითხვის ოპერაციას ეწოდება:

- ა. გაფილტვრა
- ბ. ამორჩევა (fetch)
- გ. დახარისხება

13.1.13. თუ ერთი ოპერაციის შესრულების შედეგად კურსორი საშუალებას გვაძლევს ამოვარჩიოთ:

- ა. ერთი სტრიქონი, მაშინ ასეთ კურსორს ბლოკური ეწოდება
- ბ. რამდენიმე სტრიქონი, მაშინ ასეთ კურსორს ბლოკური ეწოდება
- გ. ორი სტრიქონი, მაშინ ასეთ კურსორს ბლოკური ეწოდება

13.1.14. კურსორები შეიძლება ორ კატეგორიად დაიყოს:

- ა. მიმდევრობითი და გადახვევადი
- ბ. მხოლოდ მიმდევრობითი
- გ. მხოლოდ გადახვევადი

13.1.15. მიმდევრობითი კურსორები საშუალებას გვაძლევენ მონაცემები:

- ა. ორივე მიმართულებით და მივმართოთ ნებისმიერ სტრიქონს
- ბ. მიმდევრობით ამოვარჩიოთ მხოლოდ ერთი მიმართულებით - ბოლოდან დასაწყისისკენ
- გ. მიმდევრობით ამოვარჩიოთ მხოლოდ ერთი მიმართულებით - დასაწყისიდან ბოლოსკენ

13.1.16. გადახვევადი კურსორები საშუალებას გვაძლევენ მონაცემები ამოვარჩიოთ:

- ა. მიმდევრობით ამოვარჩიოთ მხოლოდ ერთი მიმართულებით - დასაწყისიდან ბოლოსკენ

- ბ. ორივე მიმართულებით და მივმართოთ ნებისმიერ სტრიქონს
- გ. მიმდევრობით ამოვარჩიოთ მხოლოდ ერთი მიმართულებით - ბოლოდან დასაწყისისკენ

კურსორის ტიპები. სტატიკური კურსორები. დინამიკური კურსორები. მიმდევრობითი კურსორები. საგასაღებო კურსორები

13.2.1. სტატიკური კურსორის გახსნისას სერვერი სტრიქონების მთელი შედეგობრივი ნაკრების ასლს ინახავს:

- ა. tempdb სისტემურ მონაცემთა ბაზაში
- ბ. model სისტემურ მონაცემთა ბაზაში
- გ. master სისტემურ მონაცემთა ბაზაში

13.2.2. სტატიკური კურსორი:

- ა. იცვლება შექმნის შემდეგ და ყოველთვის ასახავს მონაცემების იმ ნაკრებს, რომელიც არსებობდა კურსორის გახსნის მომენტში
- ბ. არ იცვლება შექმნის შემდეგ და ყოველთვის ასახავს მონაცემების იმ ნაკრებს, რომელიც არსებობდა კურსორის გახსნის მომენტში
- გ. არ იცვლება შექმნის შემდეგ და ყოველთვის არ ასახავს მონაცემების იმ ნაკრებს, რომელიც არსებობდა კურსორის გახსნის მომენტში

13.2.3. სტატიკური კურსორი ყოველთვის:

- ა. იხსნება „მხოლოდ წაკითხვის“ რეჟიმში
- ბ. არ იხსნება „მხოლოდ წაკითხვის“ რეჟიმში
- გ. იხსნება „მხოლოდ ჩაწერის“ რეჟიმში

13.2.4. თუ სხვა მომხმარებლები საწყის ცხრილში ამატებენ, ცვლიან ან შლიან კურსორში შესულ სტრიქონებს, მაშინ ეს ცვლილებები სტატიკურ კურსორში:

- ა. ხანდახან აისახება
- ბ. არ აისახება
- გ. აისახება

13.2.5. დინამიური კურსორის გამოყენების დროს სრულდება სტრიქონების დინამიურად ამორჩევა საწყისი ცხრილიდან მხოლოდ მაშინ, როცა:

- ა. ხდება მომხმარებლის მიერ ამ სტრიქონებთან მიმართვა
- ბ. არ ხდება მომხმარებლის მიერ ამ სტრიქონებთან მიმართვა
- გ. ხდება მომხმარებლის მიერ ამ ტრიგერებთან მიმართვა

13.2.6. დინამიური კურსორში ამორჩევის მომენტში სერვერი სტრიქონებს:

- ა. ხანდახან ბლოკავს
- ბ. არ ბლოკავს
- გ. ბლოკავს

13.2.7. კურსორის შედეგობრივ ნაკრებში მომხმარებლების მიერ შეტანილი ცვლილებები დინამიურ კურსორში:

- ა. არ გამოჩნდება ამორჩევის დროს
- ბ. გამოჩნდება ჩაწერის დროს

გ. გამოჩნდება ამორჩევის დროს

13.2.8. თუ სხვა მომხმარებელმა სტრიქონი შეცვალა დინამიურ კურსორში მისი ამორჩევის შემდეგ, მაშინ ეს ცვლილებები კურსორში:

- ა. აისახება
- ბ. არ აისახება
- გ. ხანდახან აისახება

13.2.9. კურსორში შეტანილი ცვლილებები უხილავია სხვა მომხმარებლებისთვის მანამ, სანამ კურსორი:

- ა. არ იქნება დადასტურებული
- ბ. იქნება დადასტურებული
- გ. ხანდახან იქნება დადასტურებული

13.2.10. თუ კურსორის იზოლირების დონე უშვებს „ჭუჭყიან“ კითხვას, მაშინ სხვა მომხმარებლები:

- ა. შეძლებენ კურსორიდან მონაცემების წაკითხვას საწყის მდგომარეობებში
- ბ. ვერ შეძლებენ კურსორიდან მონაცემების წაკითხვას შუალედურ მდგომარეობებში
- გ. შეძლებენ კურსორიდან მონაცემების წაკითხვას შუალედურ მდგომარეობებში

13.2.11. მიმდევრობითი კურსორები საშუალებას გვაძლევენ სტრიქონები ამოვირჩიოთ:

- ა. მხოლოდ ბოლოდან დასაწყისისკენ
- ბ. მხოლოდ დასაწყისიდან ბოლოსკენ
- გ. მხოლოდ დასაწყისიდან დასაწყისისკენ

13.2.12. მიმდევრობითი კურსორი სტრიქონების მთელ ნაკრებს:

- ა. ინახავს
- ბ. არ ინახავს
- გ. ხანდახან ინახავს

13.2.13. ცხრილიდან მაშინ სრულდება სტრიქონების წაკითხვა, როცა ისინი კურსორში:

- ა. არ ამოირჩევა
- ბ. ხანდახან ამოირჩევა
- გ. ამოირჩევა

13.2.14. მიმდევრობითი კურსორი იძლევა მომხმარებლის მიერ ცხრილში:

- ა. INSERT, UPDATE და DELETE ბრძანებებით შეტანილი ყველა ცვლილების დინამიური ასახვის შესაძლებლობას
- ბ. INSERT, UPDATE და DELETE ბრძანებებით შეტანილი ყველა ცვლილების სტატიკური ასახვის შესაძლებლობას
- გ. SELECT ბრძანებებით შეტანილი ყველა ცვლილების სტატიკური ასახვის შესაძლებლობას

13.2.15. მიმდევრობით კურსორში გამოჩნდება მონაცემების:

- ა. შუალედური მდგომარეობა
- ბ. საწყისი მდგომარეობა
- გ. უკანასკნელი მდგომარეობა

13.2.16.საგასაღებო კურსორები ან გასაღებების ნაკრებზე დამოკიდებული კურსორები:

- ა. აგებულია უნიკალური იდენტიფიკატორების ბაზაზე
- ბ. არ არის აგებული უნიკალური იდენტიფიკატორების ბაზაზე
- გ. აგებულია არაუნიკალური იდენტიფიკატორების ბაზაზე

13.2.17.ცხრილის სტრიქონების უნიკალური იდენტიფიკატორების (გასაღებების) სიმრავლეს:

- ა. ტრიგერების სიმრავლე ეწოდება
- ბ. გასაღებების სიმრავლე ეწოდება
- გ. პროცედურების სიმრავლე ეწოდება

13.2.18.საგასაღებო კურსორი წარმოადგენს გასაღებების ნაკრებს, რომლებიც:

- ა. არ ახდენენ კურსორის მთელი შედეგობრივი ნაკრების სტრიქონების იდენტიფიცირებას
- ბ. ახდენენ კურსორის მთელი შედეგობრივი ნაკრების სტრიქონების იდენტიფიცირებას
- გ. ახდენენ კურსორის საწყისი ნაკრების სვეტების იდენტიფიცირებას

13.2.19.გასაღებების ნაკრები ინახება:

- ა. tempdb სისტემურ მონაცემთა ბაზაში
- ბ. model სისტემურ მონაცემთა ბაზაში
- გ. master სისტემურ მონაცემთა ბაზაში

13.2.20.რადგან ინფორმაცია ინახება მხოლოდ კურსორის მთელ შედეგობრივ ნაკრებში ჩართული სტრიქონების საგასაღებო სვეტების შესახებ, ამიტომ საგასაღებო კურსორები:

- ა. არ ასახავენ სხვა მომხმარებლების მიერ შეტანილ ყველა ცვლილებას
- ბ. არ ასახავენ სხვა მომხმარებლების მიერ შეტანილ არც ერთ ცვლილებას
- გ. ასახავენ სხვა მომხმარებლების მიერ შეტანილ ყველა ცვლილებას

13.2.21.კურსორის გახსნის შემდეგ დამატებული სტრიქონები კურსორში:

- ა. ხანდახან გამოჩნდება
- ბ. გამოჩნდება
- გ. არ გამოჩნდება

13.2.22.კურსორში ჩართული სტრიქონები, მაგრამ წაშლილი სხვა მომხმარებლების მიერ, გამოჩნდება როგორც:

- ა. ხანდახან დაუზიანებელი
- ბ. დაუზიანებელი
- გ. დაზიანებული

13.2.23.გასაღებების ცხრილის შედგენის მომენტში სერვერი საწყისი ცხრილის სტრიქონებს:

- ა. ბლოკავს
- ბ. არ ბლოკავს
- გ. ხანდახან ბლოკავს

კურსორების მართვა

13.3.1. კურსორთან მუშაობის დროს შეგვიძლია შემდეგი ძირითადი ოპერაციები შევასრულოთ:

- ა. მხოლოდ კურსორის გახსნა და კურსორის დახურვა
- ბ. კურსორის შექმნა, კურსორის გახსნა, სტრიქონების ამორჩევა კურსორიდან და მათი შეცვლა კურსორის საშუალებით, კურსორის დახურვა და კურსორის გათავისუფლება
- გ. მხოლოდ კურსორის გახსნა და სტრიქონების ამორჩევა კურსორიდან და მათი შეცვლა კურსორის საშუალებით

13.3.2. შექმნილი, მაგრამ გაუხსნელი კურსორი მონაცემებს:

- ა. შეიცავს
- ბ. არ შეიცავს
- გ. ხანდახან შეიცავს

13.3.3. გახსნის ოპერაცია კურსორს მონაცემებით:

- ა. ხანდახან ავსებს
- ბ. ავსებს
- გ. არ ავსებს

13.3.4. კურსორთან მუშაობის დამთავრების შემდეგ ის უნდა დავხუროთ. ამ დროს სერვერი ათავისუფლებს სივრცეს:

- ა. tempdb მონაცემთა ბაზაში, რომელიც კურსორს გამოეყო შექმნის დროს
- ბ. model მონაცემთა ბაზაში, რომელიც კურსორს გამოეყო შექმნის დროს
- გ. master მონაცემთა ბაზაში, რომელიც კურსორს გამოეყო შექმნის დროს

13.3.5. კურსორის გათავისუფლების დროს კურსორი:

- ა. იცვლის სახელს
- ბ. არ იშლება
- გ. იშლება

კურსორის შექმნა. კურსორის გახსნა

13.4.1. კურსორის შესაქმნელად გამოიყენება ბრძანება:

- ა. CREATE DATABASE
- ბ. CREATE TABLE
- გ. DECLARE CURSOR

13.4.2. არსებობს DECLARE CURSOR ბრძანების:

- ა. ერთი ფორმატი - განსაზღვრული Transact-SQL სტანდარტით
- ბ. ერთი ფორმატი - განსაზღვრული SQL-92 სტანდარტით
- გ. ორი ფორმატი: განსაზღვრული SQL-92 სტანდარტით და Transact-SQL-ით

13.4.3. SQL-92 სტანდარტით განსაზღვრული DECLARE CURSOR ბრძანების INSENSITIVE არგუმენტი მიუთითებს, რომ:

- ა. არ უნდა შეიქმნას სტატიკური კურსორი
- ბ. უნდა შეიქმნას სტატიკური კურსორი
- გ. უნდა შეიქმნას დინამიური კურსორი

13.4.4. თუ SQL-92 სტანდარტით განსაზღვრული DECLARE CURSOR ბრძანების INSENSITIVE

არგუმენტი მითითებული არ არის, მაშინ:

- ა. შეიქმნება დინამიური კურსორი
- ბ. შეიქმნება სტატიკური კურსორი
- გ. არ შეიქმნება დინამიური კურსორი

13.4.5. SQL-92 სტანდარტით განსაზღვრული DECLARE CURSOR ბრძანების SCROLL არგუმენტი მითითებს, რომ:

- ა. არ შეიქმნება გადახვევადი კურსორი
- ბ. შეიქმნება მიმდევრობითი კურსორი
- გ. შეიქმნება გადახვევადი კურსორი

13.4.6. თუ SQL-92 სტანდარტით განსაზღვრული DECLARE CURSOR ბრძანების SCROLL არგუმენტი მითითებული არ არის, მაშინ

- ა. შეიქმნება გადახვევადი კურსორი
- ბ. შეიქმნება მიმდევრობითი კურსორი
- გ. არ შეიქმნება მიმდევრობითი კურსორი

13.4.7. SQL-92 სტანდარტით განსაზღვრული DECLARE CURSOR ბრძანების 'FOR select_ბრძანება' არგუმენტი შეიცავს:

- ა. SELECT მოთხოვნას, რომელიც არ გასცემს კურსორის სტრიქონების შედეგობრივ ნაკრებს
- ბ. SELECT მოთხოვნას, რომელიც გასცემს კურსორის სტრიქონების შედეგობრივ ნაკრებს
- გ. UPDATE მოთხოვნას, რომელიც გასცემს კურსორის სტრიქონების შედეგობრივ ნაკრებს

13.4.8. SQL-92 სტანდარტით განსაზღვრული DECLARE CURSOR ბრძანების 'FOR select_ბრძანება' არგუმენტში SELECT მოთხოვნა:

- ა. არ უნდა შეიცავდეს INTO, FOR BROWSE და COMPUTE BY განყოფილებებს
- ბ. უნდა შეიცავდეს INTO, FOR BROWSE და COMPUTE BY განყოფილებებს
- გ. არ უნდა შეიცავდეს INTO, FOR BROWSE და FROM განყოფილებებს

13.4.9. SQL-92 სტანდარტით განსაზღვრული DECLARE CURSOR ბრძანების 'FOR select_ბრძანება' არგუმენტში SELECT მოთხოვნა კონფლიქტშია კურსორის ტიპთან, მაშინ სერვერი:

- ა. არ შეასრულებს კურსორის არაცხად გარდაქმნას თავსებად ტიპში
- ბ. შეასრულებს კურსორის ცხად გარდაქმნას არათავსებად ტიპში
- გ. შეასრულებს კურსორის არაცხად გარდაქმნას თავსებად ტიპში

13.4.10. SQL-92 სტანდარტით განსაზღვრული DECLARE CURSOR ბრძანებაში თუ მითითებულია FOR READ ONLY არგუმენტი, მაშინ:

- ა. არ შეიქმნება „მხოლოდ წაკითხვადი“ კურსორი
- ბ. შეიქმნება „მხოლოდ წაკითხვადი“ კურსორი
- გ. შეიქმნება „მხოლოდ ჩაწერადი“ კურსორი

13.4.11. „მხოლოდ წაკითხვადი“ კურსორი:

- ა. შეიძლება იყოს დინამიური
- ბ. შეიძლება იყოს სტატიკური
- გ. არ შეიძლება იყოს დინამიურიც

13.4.12. SQL-92 სტანდარტით განსაზღვრული DECLARE CURSOR ბრძანებაში თუ მითითებულია FOR UPDATE [OF სვეტის_სახელი [,...n]] არგუმენტი, მაშინ კურსორში:

- ა. შეუძლებელი იქნება სტრიქონების ცვლილება
- ბ. შესაძლებელი არ იქნება სტრიქონების ცვლილება
- გ. შესაძლებელი იქნება სტრიქონების ცვლილება

13.4.13. SQL-92 სტანდარტით განსაზღვრული DECLARE CURSOR ბრძანებაში თუ მითითებულია FOR UPDATE [OF სვეტის_სახელი [,...n]] არგუმენტი, მაშინ კურსორში:

- ა. დაუშვებელი იქნება მხოლოდ 'სვეტის_სახელი' სიაში მითითებული სვეტების შეცვლა
- ბ. დასაშვებია იქნება მხოლოდ 'სვეტის_სახელი' სიაში მითითებული სვეტების შეცვლა
- გ. დასაშვებია იქნება მხოლოდ 'სვეტის_სახელი' სიაში მითითებული სტრიქონების შეცვლა

13.4.14. თუ SQL-92 სტანდარტით განსაზღვრული DECLARE CURSOR ბრძანების FOR UPDATE [OF სვეტის_სახელი [,...n]] არგუმენტში მითითებული არ არის 'სვეტის_სახელი', მაშინ კურსორში:

- ა. შესაძლებელი იქნება კურსორის ყველა სვეტის შეცვლა
- ბ. შეუძლებელი იქნება კურსორის ყველა სვეტის შეცვლა
- გ. შესაძლებელი არ იქნება კურსორის ყველა სვეტის შეცვლა

13.4.15. Transact-SQL სტანდარტით განსაზღვრული DECLARE CURSOR ბრძანებაში LOCAL არგუმენტი მიუთითებს, რომ შეიქმნება

- ა. გლობალური კურსორი, რომელიც ხილული იქნება მხოლოდ ამ კურსორის შემქმნელი პაკეტის, ტრიგერის, შენახული პროცედურის ან მომხმარებლის მიერ შექმნილი ფუნქციის შიგნით
- ბ. ლოკალური კურსორი, რომელიც ხილული იქნება მხოლოდ ამ კურსორის შემქმნელი პაკეტის, ტრიგერის, შენახული პროცედურის ან მომხმარებლის მიერ შექმნილი ფუნქციის შიგნით
- გ. ლოკალური კურსორი, რომელიც ხილული არ იქნება მხოლოდ ამ კურსორის შემქმნელი პაკეტის, ტრიგერის, შენახული პროცედურის ან მომხმარებლის მიერ შექმნილი ფუნქციის შიგნით

13.4.16. პაკეტის, ტრიგერის ან შენახული პროცედურის დამთავრებისთანავე ლოკალური კურსორი:

- ა. წაიშლება
- ბ. არ წაიშლება
- გ. შეიქმნება

13.4.17. Transact-SQL სტანდარტით განსაზღვრული DECLARE CURSOR ბრძანებაში GLOBAL არგუმენტი მიუთითებს, რომ შეიქმნება გლობალური კურსორი, რომელიც:

- ა. არ იარსებებს მიმდინარე შეერთების დახურვამდე
- ბ. იარსებებს მიმდინარე შეერთების დახურვამდე
- გ. იარსებებს მიმდინარე შეერთების გახსნამდე

13.4.18. Transact-SQL სტანდარტით განსაზღვრული DECLARE CURSOR ბრძანებაში FORWARD_ONLY არგუმენტი მიუთითებს, რომ:

- ა. არ შეიქმნება მიმდევრობითი კურსორი

- ბ. შეიქმნება გადახვევადი კურსორი
- გ. შეიქმნება მიმდევრობითი კურსორი

13.4.19. Transact-SQL სტანდარტით განსაზღვრული DECLARE CURSOR ბრძანებაში SCROLL მიუთითებს, რომ:

- ა. არ შეიქმნება გადახვევადი კურსორი
- ბ. შეიქმნება გადახვევადი კურსორი
- გ. შეიქმნება მიმდევრობითი კურსორი

13.4.20. Transact-SQL სტანდარტით განსაზღვრული DECLARE CURSOR ბრძანებაში STATIC მიუთითებს, რომ:

- ა. შეიქმნება სტატიკური კურსორი
- ბ. არ შეიქმნება სტატიკური კურსორი
- გ. შეიქმნება დინამიური კურსორი

13.4.21. Transact-SQL სტანდარტით განსაზღვრული DECLARE CURSOR ბრძანებაში KEYSSET არგუმენტი მიუთითებს, რომ:

- ა. შეიქმნება საგასაღებო კურსორი
- ბ. არ შეიქმნება საგასაღებო კურსორი
- გ.

13.4.22. Transact-SQL სტანდარტით განსაზღვრული DECLARE CURSOR ბრძანებაში DYNAMIC მიუთითებს, რომ:

- ა. არ შეიქმნება დინამიური კურსორი
- ბ. შეიქმნება დინამიური კურსორი
- გ. შეიქმნება სტატიკური კურსორი

13.4.23. Transact-SQL სტანდარტით განსაზღვრული DECLARE CURSOR ბრძანებაში თუ FAST_FORWARD არგუმენტი მითითებულია READ_ONLY არგუმენტთან ერთად, მაშინ:

- ა. მოხდება შექმნილი კურსორის ოპტიმიზება მონაცემებთან წელი მიმართვის მიზნით
- ბ. არ მოხდება შექმნილი კურსორის ოპტიმიზება მონაცემებთან სწრაფი მიმართვის მიზნით
- გ. მოხდება შექმნილი კურსორის ოპტიმიზება მონაცემებთან სწრაფი მიმართვის მიზნით

13.4.24. Transact-SQL სტანდარტით განსაზღვრული DECLARE CURSOR ბრძანებაში FAST_FORWARD არგუმენტი:

- ა. არ შეიძლება გამოყენებული იყოს FORWARD_ONLY და OPTIMISTIC არგუმენტებთან ერთად
- ბ. შეიძლება გამოყენებული იყოს FORWARD_ONLY და OPTIMISTIC არგუმენტებთან ერთად
- გ. არ შეიძლება გამოყენებული იყოს READ_ONLY, FORWARD_ONLY და OPTIMISTIC არგუმენტებთან ერთად

13.4.25. Transact-SQL სტანდარტით განსაზღვრული DECLARE CURSOR ბრძანებაში OPTIMISTIC არგუმენტი მიუთითებს, რომ კურსორში:

- ა. იკრძალება იმ სტრიქონების შეცვლა ან წაშლა, რომლებიც ცხრილში შეიცვალა კურსორის გახსნის შემდეგ

ბ. არ იკრძალება იმ სტრიქონების შეცვლა ან წაშლა, რომლებიც ცხრილში შეიცვალა კურსორის გახსნის შემდეგ

გ. იკრძალება იმ სტრიქონების შეცვლა ან წაშლა, რომლებიც ცხრილში შეიცვალა კურსორის გახსნამდე

13.4.26. Transact-SQL სტანდარტით განსაზღვრული DECLARE CURSOR ბრძანებაში TYPE_WARNING არგუმენტი თუ მითითებულია, მაშინ სერვერი მომხმარებელს:

ა. არ შეატყობინებს კურსორის ტიპის შეცვლის შესახებ, თუ ის არათავსებადია SELECT მოთხოვნის ტიპთან

ბ. შეატყობინებს კურსორის ტიპის შეცვლის შესახებ, თუ ის არათავსებადია SELECT მოთხოვნის ტიპთან

გ. შეატყობინებს კურსორის ტიპის შეცვლის შესახებ, თუ ის არათავსებადია INSERT მოთხოვნის ტიპთან

13.4.27. კურსორის გასახსნელად და მონაცემებით შესავსებად გამოიყენება:

ა. CLOSE ბრძანება

ბ. OPEN ბრძანება

გ. INSERT ბრძანება

13.4.28. თუ საჭიროა გლობალური კურსორის გახსნა, მაშინ OPEN ბრძანებაში:

ა. არ უნდა მივუთითოთ GLOBAL არგუმენტი

ბ. უნდა მივუთითოთ LOCAL არგუმენტი

გ. უნდა მივუთითოთ GLOBAL არგუმენტი

13.4.29. DECLARE Xelshekruleba_Cursori CURSOR FOR

```
SELECT * FROM Xelshekruleba WHERE vali_1 > 0;
```

```
OPEN Xelshekruleba_Cursori;
```

```
FETCH NEXT FROM Xelshekruleba_Cursori;
```

```
CLOSE Xelshekruleba_Cursori;
```

```
DEALLOCATE Xelshekruleba_Cursori;
```

პროგრამული კოდის შესრულების შედეგად:

ა. შეიქმნება კურსორი, რომელშიც მოთავსდება მევალებების შემცველი სტრიქონები. შესრულდება კურსორიდან ერთი სტრიქონის წაკითხვა, შემდეგ კურსორის დახურვა, ბოლოს კი - წაშლა

ბ. შეიქმნება კურსორი, რომელშიც არ მოთავსდება მევალებების შემცველი სტრიქონები. შესრულდება კურსორიდან ერთი სტრიქონის წაკითხვა, შემდეგ კურსორის დახურვა, ბოლოს კი - წაშლა

გ. შეიქმნება კურსორი, რომელშიც მოთავსდება მევალებების შემცველი სტრიქონები. შესრულდება კურსორის წაშლა, შემდეგ კურსორიდან ერთი სტრიქონის წაკითხვა, ბოლოს კი - დახურვა

13.4.30. იმისათვის, რომ შევქმნათ კურსორი, რომელშიც მოთავსდება მევალებების შემცველი სტრიქონები, შევასრულოთ კურსორიდან ერთი სტრიქონის წაკითხვა, შემდეგ კურსორის დახურვა, ბოლოს კი - წაშლა, უნდა შევასრულოთ პროგრამული კოდი:

ა. DECLARE Xelshekruleba_Cursori CURSOR FOR

```
SELECT * FROM Xelshekruleba WHERE vali_1 > 0;
```

```

OPEN Xelshekruleba_Cursori;
FETCH NEXT FROM Xelshekruleba_Cursori;
DEALLOCATE Xelshekruleba_Cursori;
CLOSE Xelshekruleba_Cursori;
ბ. DECLARE Xelshekruleba_Cursori CURSOR FOR
   SELECT * FROM Xelshekruleba WHERE vali_1 > 0;
OPEN Xelshekruleba_Cursori;
FETCH NEXT FROM Xelshekruleba_Cursori;
CLOSE Xelshekruleba_Cursori;
DEALLOCATE Xelshekruleba_Cursori;
გ. DECLARE Xelshekruleba_Cursori CURSOR FOR
   SELECT * FROM Xelshekruleba WHERE vali_1 > 0;
FETCH NEXT FROM Xelshekruleba_Cursori;
CLOSE Xelshekruleba_Cursori;
OPEN Xelshekruleba_Cursori;
DEALLOCATE Xelshekruleba_Cursori;

```

მონაცემების წაკითხვა

13.5.1. კურსორიდან მონაცემების წასაკითხად გამოიყენება:

- ა. FETCH ბრძანება
- ბ. INSERT ბრძანება
- გ. OPEN ბრძანება

13.5.2. FETCH ბრძანების FIRST არგუმენტი მიუთითებს, რომ:

- ა. არ გაიცემა კურსორის შედეგობრივი ნაკრების პირველი სტრიქონი, რომელიც მიმდინარე გახდება
- ბ. გაიცემა კურსორის შედეგობრივი ნაკრების პირველი სტრიქონი, რომელიც მიმდინარე გახდება
- გ. გაიცემა კურსორის შედეგობრივი ნაკრების მეორე სტრიქონი, რომელიც მიმდინარე გახდება

13.5.3. FETCH ბრძანების LAST არგუმენტი მიუთითებს, რომ:

- ა. არ გაიცემა კურსორის შედეგობრივი ნაკრების უკანასკნელი სტრიქონი, რომელიც მიმდინარე გახდება
- ბ. გაიცემა კურსორის შედეგობრივი ნაკრების პირველი სტრიქონი, რომელიც მიმდინარე გახდება
- გ. გაიცემა კურსორის შედეგობრივი ნაკრების უკანასკნელი სტრიქონი, რომელიც მიმდინარე გახდება

13.5.4. FETCH ბრძანების NEXT არგუმენტი მიუთითებს, რომ:

- ა. გაიცემა კურსორის შედეგობრივი ნაკრების მიმდინარე სტრიქონის შემდეგ მოთავსებული სტრიქონი, რომელიც მიმდინარე გახდება
- ბ. არ გაიცემა კურსორის შედეგობრივი ნაკრების მიმდინარე სტრიქონის შემდეგ მოთავსებული სტრიქონი, რომელიც მიმდინარე გახდება
- გ. გაიცემა კურსორის შედეგობრივი ნაკრების მიმდინარე სტრიქონის წინ მოთავსებული

სტრიქონი, რომელიც მიმდინარე გახდება

13.5.5. FETCH ბრძანების PRIOR არგუმენტი მიუთითებს, რომ:

- ა. გაიცემა კურსორის შედეგობრივი ნაკრების მიმდინარე სტრიქონის უკან მოთავსებული სტრიქონი, რომელიც მიმდინარე გახდება
- ბ. გაიცემა კურსორის შედეგობრივი ნაკრების მიმდინარე სტრიქონის წინ მოთავსებული სტრიქონი, რომელიც მიმდინარე გახდება
- გ. არ გაიცემა კურსორის შედეგობრივი ნაკრების მიმდინარე სტრიქონის წინ მოთავსებული სტრიქონი, რომელიც მიმდინარე გახდება

13.5.6. FETCH ბრძანების ABSOLUTE { n | @nvar } არგუმენტი გასცემს სტრიქონს კურსორის მთლიან შედეგობრივ ნაკრებში მისი:

- ა. ლოგიკური რიგითი ნომრის მიხედვით
- ბ. ფარდობითი რიგითი ნომრის მიხედვით
- გ. აბსოლუტური რიგითი ნომრის მიხედვით

13.5.7. FETCH ბრძანების ABSOLUTE { n | @nvar } არგუმენტში n ან @nvar შეიძლება იყოს:

- ა. მხოლოდ უარყოფითი
- ბ. როგორც დადებითი, ისე უარყოფითი
- გ. მხოლოდ დადებითი

13.5.8. FETCH ბრძანების ABSOLUTE { n | @nvar } არგუმენტში თუ მითითებულია დადებითი მნიშვნელობა, მაშინ სტრიქონი:

- ა. გადაითვლება ნაკრების დასაწყისიდან
- ბ. გადაითვლება ნაკრების ბოლოდან
- გ. გადაითვლება ნაკრების შუიდან

13.5.9. FETCH ბრძანების ABSOLUTE { n | @nvar } არგუმენტში თუ მითითებულია უარყოფითი მნიშვნელობა, მაშინ სტრიქონი:

- ა. გადაითვლება ნაკრების შუიდან
- ბ. გადაითვლება ნაკრების ბოლოდან
- გ. გადაითვლება ნაკრების დასაწყისიდან

13.5.10. FETCH ბრძანების ABSOLUTE { n | @nvar } არგუმენტში თუ მითითებულია ნულოვანი მნიშვნელობა, მაშინ სტრიქონი:

- ა. ხანდახან გაიცემა
- ბ. გაიცემა
- გ. არ გაიცემა

13.5.11. FETCH ბრძანების RELATIVE { n | @nvar } არგუმენტი გასცემს სტრიქონს, რომელიც მდებარეობს მიმდინარე სტრიქონიდან n სტრიქონის შემდეგ, თუ n:

- ა. უარყოფითია
- ბ. დადებითია
- გ. ნულოვანია

13.5.12. FETCH ბრძანების RELATIVE { n | @nvar } არგუმენტი გასცემს სტრიქონს, რომელიც

მდებარეობს მიმდინარე სტრიქონიდან n სტრიქონით წინ, თუ n:

- ა. უარყოფითია
- ბ. დადებითია
- გ. ნულოვანია

13.5.13.FETCH ბრძანების RELATIVE { n | @nvar } არგუმენტში თუ მითითებულია ნულოვანი მნიშვნელობა, მაშინ გაიკემა:

- ა. წინა სტრიქონი
- ბ. მიმდინარე სტრიქონი
- გ. მომდევნო სტრიქონი

13.5.14.FETCH ბრძანების INTO @variable_name [,...n] არგუმენტი მიუთითებს იმ ცვლადების სიას, რომლებიც:

- ა. შეიცავენ დასაბრუნებელი სტრიქონის სვეტების არაშესაბამის მნიშვნელობებს
- ბ. არ შეიცავენ დასაბრუნებელი სტრიქონის სვეტების შესაბამის მნიშვნელობებს
- გ. შეიცავენ დასაბრუნებელი სტრიქონის სვეტების შესაბამის მნიშვნელობებს

13.5.15.FETCH ბრძანების INTO @variable_name [,...n] არგუმენტში ცვლადების მითითების მიმდევრობა:

- ა. არ უნდა შეესაბამებოდეს კურსორში სვეტების მიმდევრობას
- ბ. უნდა შეესაბამებოდეს კურსორში სვეტების მიმდევრობას
- გ. უნდა შეესაბამებოდეს ინდექსში სვეტების მიმდევრობას

13.5.16.FETCH ბრძანების INTO @variable_name [,...n] არგუმენტში ცვლადის ტიპი:

- ა. უნდა ემთხვეოდეს სტრიქონის ტიპს
- ბ. არ უნდა ემთხვეოდეს სვეტის ტიპს
- გ. უნდა ემთხვეოდეს სვეტის ტიპს

13.5.17.FETCH ბრძანების INTO @variable_name [,...n] არგუმენტი თუ მითითებული არ არის, მაშინ მონაცემები:

- ა. ეკრანზე გამოიკემა
- ბ. ფაილში ჩაიწერება
- გ. ბაზაში ჩაიწერება

13.5.18.FETCH NEXT FROM Xelshekruleba_Cursori; ბრძანების შესრულების შედეგად:

- ა. Xelshekruleba_Cursori კურსორიდან შესრულდება წინა სტრიქონის წაკითხვა
- ბ. Xelshekruleba_Cursori კურსორიდან შესრულდება მომდევნო სტრიქონის წაკითხვა
- გ. Xelshekruleba_Cursori კურსორიდან შესრულდება პირველი სტრიქონის წაკითხვა

13.5.19.FETCH PRIOR FROM Xelshekruleba_Cursori; ბრძანების შესრულების შედეგად:

- ა. Xelshekruleba_Cursori კურსორიდან შესრულდება წინა სტრიქონის წაკითხვა
- ბ. Xelshekruleba_Cursori კურსორიდან შესრულდება მომდევნო სტრიქონის წაკითხვა
- გ. Xelshekruleba_Cursori კურსორიდან შესრულდება პირველი სტრიქონის წაკითხვა

13.5.20.FETCH LAST FROM Xelshekruleba_Cursori; ბრძანების შესრულების შედეგად:

- ა. Xelshekruleba_Cursori კურსორიდან შესრულდება წინა სტრიქონის წაკითხვა

- ბ. Xelshekruleba_Cursori კურსორიდან შესრულდება პირველი სტრიქონის წაკითხვა
- გ. Xelshekruleba_Cursori კურსორიდან შესრულდება უკანასკნელი სტრიქონის წაკითხვა

13.5.21.FETCH FIRST FROM Xelshekruleba_Cursori; ბრძანების შესრულების შედეგად:

- ა. Xelshekruleba_Cursori კურსორიდან შესრულდება წინა სტრიქონის წაკითხვა
- ბ. Xelshekruleba_Cursori კურსორიდან შესრულდება პირველი სტრიქონის წაკითხვა
- გ. Xelshekruleba_Cursori კურსორიდან შესრულდება უკანასკნელი სტრიქონის წაკითხვა

13.5.22.FETCH ABSOLUTE 3 FROM Xelshekruleba_Cursori; ბრძანების შესრულების შედეგად:

- ა. Xelshekruleba_Cursori კურსორიდან შესრულდება დასაწყისიდან მესამე სტრიქონის წაკითხვა
- ბ. Xelshekruleba_Cursori კურსორიდან შესრულდება პირველი სტრიქონის წაკითხვა
- გ. Xelshekruleba_Cursori კურსორიდან შესრულდება უკანასკნელი სტრიქონის წაკითხვა

13.5.23.FETCH ABSOLUTE -3 FROM Xelshekruleba_Cursori; ბრძანების შესრულების შედეგად:

- ა. Xelshekruleba_Cursori კურსორიდან შესრულდება ბოლოდან მესამე სტრიქონის წაკითხვა
- ბ. Xelshekruleba_Cursori კურსორიდან შესრულდება პირველი სტრიქონის წაკითხვა
- გ. Xelshekruleba_Cursori კურსორიდან შესრულდება უკანასკნელი სტრიქონის წაკითხვა

13.5.24.FETCH RELATIVE 2 FROM Xelshekruleba_Cursori; ბრძანების შესრულების შედეგად:

- ა. Xelshekruleba_Cursori კურსორიდან შესრულდება ბოლოდან მესამე სტრიქონის წაკითხვა
- ბ. Xelshekruleba_Cursori კურსორიდან შესრულდება სტრიქონის წაკითხვა, რომელიც მდებარეობს მიმდინარე სტრიქონიდან 2 სტრიქონით წინ
- გ. Xelshekruleba_Cursori კურსორიდან შესრულდება უკანასკნელი სტრიქონის წაკითხვა

13.5.25.FETCH RELATIVE -2 FROM Xelshekruleba_Cursori; ბრძანების შესრულების შედეგად:

- ა. Xelshekruleba_Cursori კურსორიდან შესრულდება ბოლოდან მესამე სტრიქონის წაკითხვა
- ბ. Xelshekruleba_Cursori კურსორიდან შესრულდება სტრიქონის წაკითხვა, რომელიც მდებარეობს მიმდინარე სტრიქონიდან 2 სტრიქონით უკან
- გ. Xelshekruleba_Cursori კურსორიდან შესრულდება უკანასკნელი სტრიქონის წაკითხვა

13.5.26.DECLARE Xelshekruleba_Cursori CURSOR SCROLL FOR

SELECT vali_1 FROM Xelshekruleba WHERE vali_1 > 0;

OPEN Xelshekruleba_Cursori;

DECLARE @vali_1 FLOAT;

FETCH NEXT FROM Xelshekruleba_Cursori INTO @vali_1;

SELECT @vali_1;

CLOSE Xelshekruleba_Cursori;

DEALLOCATE Xelshekruleba_Cursori;

პროგრამული კოდის შესრულების შედეგად:

- ა. @vali_1 ცვლადიდან წაიშლება vali_1 სვეტის მნიშვნელობა
- ბ. @vali_1 ცვლადში არ ჩაიწერება vali_1 სვეტის მნიშვნელობა
- გ. @vali_1 ცვლადში ჩაიწერება vali_1 სვეტის მნიშვნელობა

მონაცემების შეცვლა. მონაცემების წაშლა

13.6.1. კურსორის საშუალებით მონაცემების შესაცვლელად გამოიყენება:

- ა. DELETE ბრძანება
- ბ. UPDATE ბრძანება
- გ. INSERT ბრძანება

13.6.2. კურსორის საშუალებით მონაცემების შეცვლის ერთი ოპერაციის შედეგად:

- ა. არ შეიძლება რამდენიმე სვეტის შეცვლა, მაგრამ ყველა შესაცვლელი სვეტი უნდა ეკუთვნოდეს ერთ ცხრილს
- ბ. შეიძლება რამდენიმე სვეტის შეცვლა, მაგრამ ყველა შესაცვლელი სვეტი არ უნდა ეკუთვნოდეს ერთ ცხრილს
- გ. შეიძლება რამდენიმე სვეტის შეცვლა, მაგრამ ყველა შესაცვლელი სვეტი უნდა ეკუთვნოდეს ერთ ცხრილს

13.6.3. კურსორის საშუალებით მონაცემების შეცვლისას სვეტს:

- ა. არ მიენიჭება ავტომატური მნიშვნელობა, თუ მითითებულია DEFAULT საკვანძო სიტყვა
- ბ. მიენიჭება ავტომატური მნიშვნელობა, თუ მითითებულია DEFAULT საკვანძო სიტყვა
- გ. მიენიჭება ავტომატური მნიშვნელობა, თუ მითითებულია NULL საკვანძო სიტყვა

13.6.4. კურსორის საშუალებით მონაცემების შეცვლისას სვეტს:

- ა. მიენიჭება NULL მნიშვნელობა, თუ მითითებულია NULL საკვანძო სიტყვა
- ბ. არ მიენიჭება NULL მნიშვნელობა, თუ მითითებულია NULL საკვანძო სიტყვა
- გ. მიენიჭება NULL მნიშვნელობა, თუ მითითებულია DEFAULT საკვანძო სიტყვა

13.6.5. კურსორიდან მონაცემების წასაშლელად გამოიყენება:

- ა. UPDATE ბრძანება
- ბ. INSERT ბრძანება
- გ. DELETE ბრძანება

კურსორის დახურვა. კურსორის გათავისუფლება

13.7.1. კურსორის დახურვა:

- ა. ათავისუფლებს მისთვის გამოყოფილ რესურსებს და შლის კურსორში მოთავსებულ სტრიქონებს
- ბ. არ ათავისუფლებს მისთვის გამოყოფილ რესურსებს და შლის კურსორში მოთავსებულ სტრიქონებს
- გ. ათავისუფლებს მისთვის გამოყოფილ რესურსებს და არ შლის კურსორში მოთავსებულ სტრიქონებს

13.7.2. კურსორის დახურვის დროს:

- ა. არ მოიხსნება კურსორის მუშაობის დროს დაყენებული ბლოკირებები
- ბ. მოიხსნება კურსორის მუშაობის დროს დაყენებული ბლოკირებები
- გ. მოიხსნება კურსორის მუშაობის დროს მოხსნილი ბლოკირებები

13.7.3. კურსორი, რომელიც დაიხურა, მაგრამ არ გათავისუფლდა:

- ა. არ შეიძლება განმეორებით გაიხსნას
- ბ. შეიძლება მხოლოდ ერთხელ გაიხსნას
- გ. შეიძლება განმეორებით გაიხსნას

13.7.4. კურსორის დასახურად გამოიყენება:

- ა. DELETE ბრძანება
- ბ. CLOSE ბრძანება
- გ. UPDATE ბრძანება

13.7.5. კურსორის გათავისუფლება:

- ა. იწვევს მის წაშლას მონაცემთა ბაზიდან
- ბ. არ იწვევს მის წაშლას მონაცემთა ბაზიდან
- გ. იწვევს მის შექმნას მონაცემთა ბაზიდან

13.7.6. კურსორის გასათავისუფლებლად გამოიყენება:

- ა. CLOSE ბრძანება
- ბ. DELETE ბრძანება
- გ. DEALLOCATE ბრძანება

თავი 14. ტრანზაქციები და დაბლოკვები

შესავალი

14.1.1. ერთ მონაცემთა ბაზასთან დროის ნებისმიერ მომენტში ერთდროულად:

- ა. შეიძლება რამდენიმე მომხმარებელი მუშაობდეს
- ბ. არ შეიძლება რამდენიმე მომხმარებელი მუშაობდეს
- გ. შეიძლება მხოლოდ ერთი მომხმარებელი მუშაობდეს

14.1.2. ტრანზაქცია არის პროცესი, როცა:

- ა. არ სრულდება მასში შემავალი არც ერთი ბრძანება
- ბ. სრულდება მასში შემავალი ბრძანებების ნაწილი
- გ. სრულდება მასში შემავალი ყველა ბრძანება ან არც ერთი:

14.1.3. ტრანზაქცია შეიძლება შედგებოდეს:

- ა. როგორც ერთი, ისე ასობით ბრძანებისაგან
- ბ. მხოლოდ ერთი ბრძანებისაგან
- გ. მხოლოდ ათი ბრძანებისაგან

14.1.4. თუ ყველა ბრძანება წარმატებით შესრულდა, მაშინ:

- ა. მოხდება ტრანზაქციის უკუქცევა
- ბ. მოხდება ტრანზაქციის ფიქსირება
- გ. არ მოხდება ტრანზაქციის ფიქსირება

14.1.5. თუ ტრანზაქციის რომელიმე ბრძანება ვერ შესრულდა, მაშინ:

- ა. არ მოხდება ტრანზაქციის უკუქცევა, ანუ გაუქმდება ყველა ცვლილება
- ბ. მოხდება ტრანზაქციის ფიქსირება
- გ. მოხდება ტრანზაქციის უკუქცევა, ანუ გაუქმდება ყველა ცვლილება

14.1.6. ტრანზაქციის უკუქცევის შემთხვევაში სისტემა:

- ა. აღდგება საწყის მდგომარეობაში, რომელიც მას ჰქონდა ტრანზაქციის დაწყებამდე
- ბ. არ აღდგება საწყის მდგომარეობაში, რომელიც მას ჰქონდა ტრანზაქციის დაწყებამდე
- გ. აღდგება საწყის მდგომარეობაში, რომელიც მას ჰქონდა ტრანზაქციის დამთავრების მომენტში

14.1.7. ინფორმაცია სისტემის საწყისი მდგომარეობის შესახებ:

- ა. ინახება ტრიგერების ჟურნალში
- ბ. ინახება ტრანზაქციების ჟურნალში
- გ. არ ინახება ტრანზაქციების ჟურნალში

14.1.8. ჩვეულებრივ, სერვერი:

- ა. ავტომატურად ბლოკავს იმ მონაცემებს, რომლებთანაც ტრანზაქცია მუშაობს
- ბ. ავტომატურად არ ბლოკავს იმ მონაცემებს, რომლებთანაც ტრანზაქცია მუშაობს
- გ. ავტომატურად ბლოკავს იმ მონაცემებს, რომლებთანაც ტრანზაქცია არ მუშაობს

14.1.9. მცირე ზომის ტრანზაქციების გამოყენების შემთხვევაში სერვერის მუშაობის ეფექტურობა იზრდება დიდი ზომის ტრანზაქციების გამოყენებასთან შედარებით, რადგან:

- ა. სერვერი ავტომატურად არ ბლოკავს იმ მონაცემებს, რომლებთანაც ტრანზაქცია მუშაობს
- ბ. სერვერი ავტომატურად ბლოკავს იმ მონაცემებს, რომლებთანაც ტრანზაქცია არ მუშაობს
- გ. სერვერი ავტომატურად ბლოკავს იმ მონაცემებს, რომლებთანაც ტრანზაქცია მუშაობს

14.1.10. დიდი ზომის ტრანზაქციების გამოყენების შემთხვევაში:

- ა. მცირე დროით ხდება მონაცემების დაბლოკვა და შესაბამისად, ადგილი აქვს ხანგრძლივ მოცდენებს
- ბ. ხანგრძლივი დროით ხდება მონაცემების დაბლოკვა და შესაბამისად, ადგილი აქვს ხანგრძლივ მოცდენებს
- გ. ხანგრძლივი დროით ხდება მონაცემების დაბლოკვა და შესაბამისად, ადგილი აქვს ხანმოკლე მოცდენებს

14.1.11. დაბლოკვა არის მონაცემების დამუშავების ზოგიერთ ოპერაციაზე:

- ა. იშვიათად დადებული შეზღუდვა
- ბ. ხანგრძლივად დადებული შეზღუდვა
- გ. დროებით დადებული შეზღუდვა

14.1.12. რომელი მსჯელობაა სწორი:

- ა. შეიძლება დაიბლოკოს მხოლოდ ერთი სტრიქონი
- ბ. შეიძლება დაიბლოკოს როგორც ცხრილის ერთი სტრიქონი, ისე მთელი მონაცემთა ბაზა
- გ. არ შეიძლება დაიბლოკოს როგორც ცხრილის ერთი სტრიქონი, ისე მთელი მონაცემთა ბაზა

14.1.13. რომელი მსჯელობაა სწორი:

- ა. არსებობს სხვადასხვა სახის დაბლოკვები, რომლებსაც დაბლოკვების მენეჯერი მართავს
- ბ. არ არსებობს სხვადასხვა სახის დაბლოკვები, რომლებსაც დაბლოკვების მენეჯერი მართავს
- გ. არსებობს სხვადასხვა სახის დაბლოკვები, რომლებსაც დაბლოკვების მენეჯერი არ მართავს

ტრანზაქციების მართვა. აშკარა ტრანზაქციები. ავტომატური ტრანზაქციები. არააშკარა ტრანზაქციები. განაწილებული ტრანზაქციები

14.2.1. ტრანზაქცია:

- ა. ხანდახან განისაზღვრება შეერთების დონეზე
- ბ. არ განისაზღვრება შეერთების დონეზე
- გ. განისაზღვრება შეერთების დონეზე

14.2.2. შეერთების დახურვის დროს:

- ა. ავტომატურად იხურება ტრანზაქცია
- ბ. ავტომატურად არ იხურება ტრანზაქცია
- გ. ხელით იხურება ტრანზაქცია

14.2.3. რომელი მსჯელობაა სწორი:

- ა. არსებობს ორი სახის ტრანზაქცია: ავტომატური და არააშკარა (ნაგულისხმევი)
- ბ. არსებობს სამი სახის ტრანზაქცია: აშკარა, ავტომატური და არააშკარა (ნაგულისხმევი)
- გ. არსებობს ერთი სახის ტრანზაქცია: აშკარა

14.2.4. ტრანზაქცია არის ლოკალური თუ ის სრულდება:

- ა. რამდენიმე მონაცემთა ბაზის შიგნით
- ბ. ორი მონაცემთა ბაზის შიგნით
- გ. ერთი მონაცემთა ბაზის შიგნით

14.2.5. განწილებული არის ტრანზაქცია, რომელიც მიმართავს:

- ა. ერთ მონაცემთა ბაზას
- ბ. რამდენიმე მონაცემთა ბაზას, რომლებიც შეიძლება სხვადასხვა სერვერებზე იყოს განთავსებული
- გ. რამდენიმე მონაცემთა ბაზას, რომლებიც არ შეიძლება სხვადასხვა სერვერებზე იყოს განთავსებული

14.2.6. აშკარა ტრანზაქციების გამოყენების შემთხვევაში:

- ა. აშკარად უნდა მიუთითოთ ტრანზაქციის მხოლოდ დაწყება
- ბ. აშკარად არ უნდა მიუთითოთ ტრანზაქციის დაწყება და დამთავრება
- გ. აშკარად უნდა მიუთითოთ ტრანზაქციის დაწყება და დამთავრება

14.2.7. BEGIN TRANSACTION ბრძანება განსაზღვრავს ტრანზაქციის:

- ა. დასაწყისს
- ბ. დასასრულს
- გ. შუა ნაწილს

14.2.8. ტრანზაქციის დაწყების დროს ტრანზაქციების ჟურნალში:

- ა. არ ფიქსირდება შესაცვლელი მონაცემების საწყისი მნიშვნელობები და ტრანზაქციის დაწყების მომენტი
- ბ. ფიქსირდება შესაცვლელი მონაცემების საწყისი მნიშვნელობები და ტრანზაქციის დაწყების მომენტი
- გ. ფიქსირდება შესაცვლელი მონაცემების საწყისი მნიშვნელობები და არ ფიქსირდება ტრანზაქციის დაწყების მომენტი

14.2.9. BEGIN TRANSACTION ბრძანების 'ტრანზაქციის_სახელი' არგუმენტი, ჩვეულებრივ, გამოიყენება მხოლოდ:

- ა. ჩადგმულ ტრანზაქციებთან სამუშაოდ ყველაზე მაღალი დონის ტრანზაქციის სახელდებისთვის
- ბ. ჩადგმულ ტრანზაქციებთან სამუშაოდ ყველაზე დაბალი დონის ტრანზაქციის სახელდებისთვის
- გ. ტრანზაქციის სახელის უარყოფისთვის

114.2.10. COMMIT TRANSACTION ბრძანებება განსაზღვრავს ტრანზაქციის:

- ა. დასაწყისს
- ბ. დასასრულს
- გ. შუა ნაწილს

14.2.11.თუ ტრანზაქციის შესრულების დროს ადგილი:

- ა. ჰქონდა შეცდომებს, მაშინ შესრულებული ცვლილებები დაფიქსირდება
- ბ. არ ჰქონდა შეცდომებს, მაშინ შესრულებული ცვლილებები დაფიქსირდება
- გ. არ ჰქონდა შეცდომებს, მაშინ შესრულებული ცვლილებები არ დაფიქსირდება

14.2.12.ROLLBACK TRANSACTION ბრძანებების შესრულება იწვევს ტრანზაქციის:

- ა. შეწყვეტას და უკუქცევას
- ბ. დაფიქსირებას
- გ. მხოლოდ შეწყვეტას

14.2.13.ტრანზაქციის უკუქცევის დროს, შესრულებული ცვლილებები:

- ა. უქმდება და არ აღდგება სისტემის პირვანდელი მდგომარეობა
- ბ. არ უქმდება და აღდგება სისტემის პირვანდელი მდგომარეობა
- გ. უქმდება და აღდგება სისტემის პირვანდელი მდგომარეობა

14.2.14.ჩვეულებრივ, სერვერი:

- ა. არ მუშაობს ტრანზაქციის ავტომატურად დაწყების რეჟიმში
- ბ. მუშაობს ტრანზაქციის ავტომატურად დაწყების რეჟიმში
- გ. მუშაობს ტრანზაქციის ხელით დაწყების რეჟიმში

14.2.15.ავტომატურად დაწყების რეჟიმში თითოეული ბრძანება:

- ა. განიხილება როგორც ცალკეული ტრანზაქცია
- ბ. არ განიხილება როგორც ცალკეული ტრანზაქცია
- გ. განიხილება როგორც მრავლობითი ტრანზაქცია

14.2.16.თუ საჭიროა ტრანზაქციის შექმნა, რომელიც რამდენიმე ბრძანებას შეიცავს, მაშინ ტრანზაქცია:

- ა. აშკარად არ უნდა მივუთითოთ
- ბ. ავტომატურად უნდა მივუთითოთ
- გ. აშკარად უნდა მივუთითოთ

14.2.17.ტრანზაქციის ავტომატური განსაზღვრის რეჟიმის დასაყენებლად გამოიყენება ბრძანება:

- ა. SET IMPLICIT_TRANSACTION ON
- ბ. SET IMPLICIT_TRANSACTION OFF
- გ. SELECT

14.2.18.ტრანზაქციის არააშკარა (ნაგულისხმევი) დაწყების რეჟიმში მუშაობის დროს სერვერი ავტომატურად იწყებს ახალ ტრანზაქციას, როგორც კი:

- ა. დაიწყება წინა ტრანზაქცია
- ბ. დამთავრდება წინა ტრანზაქცია
- გ. დამთავრდება მომდევნო ტრანზაქცია

14.2.19.მომხმარებელი არააშკარა ტრანზაქციის დასაწყებად:

- ა. არაფერს არ უთითებს
- ბ. სპეციალურ ბრძანებას უთითებს

გ. SELECT ბრძანებას უთითებს

14.2.20. არააშკარა ტრანზაქცია გრძელდება მანამ, სანამ მომხმარებელი:

- ა. აშკარად არ მიუთითებს ტრანზაქციის მხოლოდ უკუქცევის ბრძანებას
- ბ. აშკარად არ მიუთითებს ტრანზაქციის უკუქცევის ან ფიქსირების ბრძანებას
- გ. აშკარად არ მიუთითებს ტრანზაქციის მხოლოდ ფიქსირების ბრძანებას

14.2.21. როცა შეერთებისთვის დაყენებულია ტრანზაქციის არააშკარა დაწყების რეჟიმი, მაშინ სერვერი ავტომატურად ამთავრებს მიმდინარე ტრანზაქციას და იწყებს ახალს, თუ გვხვდება შემდეგი ბრძანებებიდან ერთ-ერთი

- ა. DROP, OPEN და CLOSE
- ბ. CREATE, DELETE, SELECT და UPDATE
- გ. TRUNCATE TABLE და COMMIT

14.2.22. ტრანზაქციის არააშკარა განსაზღვრის რეჟიმის დასაყენებლად გამოიყენება ბრძანება:

- ა. SELECT
- ბ. SET IMPLICIT_TRANSACTION OFF
- გ. SET IMPLICIT_TRANSACTION ON

14.2.23. როცა საჭიროა სხვადასხვა მონაცემთა ბაზაში მოთავსებულ რესურსებთან მიმართვა:

- ა. აუცილებელი არაა განაწილებული ტრანზაქციის გამოყენება
- ბ. აუცილებელია განაწილებული ტრანზაქციის გამოყენება
- გ. აუცილებელია არაგანაწილებული ტრანზაქციის გამოყენება

14.2.24. განაწილებული ტრანზაქცია წარმოადგენს რამდენიმე ცალკეულ ტრანზაქციას, რომლებიც:

- ა. ლოკალურად სრულდება მხოლოდ ერთ მონაცემთა ბაზაში
- ბ. გლობალურად სრულდება ცალკეულ მონაცემთა ბაზაში
- გ. ლოკალურად სრულდება ცალკეულ მონაცემთა ბაზაში

14.2.25. სერვერი განაწილებული ტრანზაქციის შესრულებას იწყებს ბრძანებით:

- ა. COMMIT DISTRIBUTED TRANSACTION
- ბ. BEGIN DISTRIBUTED TRANSACTION
- გ. ROLLBACK DISTRIBUTED TRANSACTION

14.2.26. BEGIN TRANSACTION;

INSERT INTO #Droebiti_Cxrili_1 VALUES (N'რომანი');

INSERT INTO #Droebiti_Cxrili_1 VALUES (N'ლიკა');

COMMIT TRANSACTION;

ტრანზაქციის შესრულების შედეგად:

- ა. #Droebiti_Cxrili_1 ცხრილს არ დაემატება არც ერთი სტრიქონი
- ბ. #Droebiti_Cxrili_1 ცხრილს დაემატება ერთი სტრიქონი
- გ. #Droebiti_Cxrili_1 ცხრილს დაემატება ორი სტრიქონი

14.2.27. BEGIN TRANSACTION;

INSERT INTO #Droebiti_Cxrili_1 VALUES (N'რომანი');


```
INSERT INTO #Droebiti_Cxrili_1 VALUES (N'ლიკა');  
ROLLBACK TRANSACTION;
```

ტრანზაქციის შესრულების შედეგად:

- ა. #Droebiti_Cxrili_1 ცხრილს არ დაემატება არც ერთი სტრიქონი
- ბ. #Droebiti_Cxrili_1 ცხრილს დაემატება ორი სტრიქონი
- გ. #Droebiti_Cxrili_1 ცხრილს დაემატება ერთი სტრიქონი

ჩადგმული ტრანზაქციები. ტრანზაქციებში აკრძალული Transact SQL-ის ბრძანებები

14.3.1. ჩადგმული ტრანზაქციები ეწოდება ტრანზაქციებს, რომელთა შესრულება:

- ა. არ ინიცირდება (იწყება) აქტიური ტრანზაქციის კოდიდან
- ბ. ინიცირდება (იწყება) აქტიური ტრანზაქციის კოდიდან
- გ. ინიცირდება (იწყება) არააქტიური ტრანზაქციის კოდიდან

14.3.2. ჩადგმული ტრანზაქციების გამოყენება:

- ა. მისაწვდომია მხოლოდ აშკარა ტრანზაქციებში
- ბ. მისაწვდომი არაა მხოლოდ აშკარა ტრანზაქციებში
- გ. მისაწვდომია მხოლოდ არააშკარა ტრანზაქციებში

14.3.3. ავტომატური ან არააშკარა ტრანზაქციების შესრულებისას, ახალი ტრანზაქციის დაწყებამდე, წინა ტრანზაქცია:

- ა. არ უნდა იყოს დამთავრებული
- ბ. უნდა იყოს დამთავრებული
- გ. უნდა იყოს დამთავრებული სანახევროდ

14.3.4. ჩადგმული ტრანზაქციების ძირითადი დანიშნულებაა იმ ტრანზაქციების უზრუნველყოფა, რომლებიც:

- ა. სრულდებიან ჩადგმული ინდექსების მიერ
- ბ. არ სრულდებიან ჩადგმული პროცედურების მიერ
- გ. სრულდებიან ჩადგმული პროცედურების მიერ

14.3.5. ჩადგმული ტრანზაქციის შესაქმნელად:

- ა. უნდა დავიწყოთ ახალი ტრანზაქცია და დაიხუროს წინა ტრანზაქცია
- ბ. უნდა დავიწყოთ ახალი ტრანზაქცია წინა ტრანზაქციის დახურვის გარეშე
- გ. არ უნდა დავიწყოთ ახალი ტრანზაქცია წინა ტრანზაქციის დახურვის გარეშე

14.3.6. ზედა დონის ტრანზაქციის დამთავრება:

- ა. გადაიდება ჩადგმული ტრანზაქციების დამთავრებამდე
- ბ. არ გადაიდება ჩადგმული ტრანზაქციების დამთავრებამდე
- გ. გადაიდება ჩადგმული ტრანზაქციების დაწყებამდე

14.3.7. თუ ყველაზე დაბალი დონის ტრანზაქცია წარუმატებლად დამთავრდა და უკუქცეულია, მაშინ:

- ა. ქვედა დონის ყველა ტრანზაქცია უკუიქცევა პირველი დონის ტრანზაქციის ჩათვლით
- ბ. ზედა დონის ყველა ტრანზაქცია უკუიქცევა პირველი დონის ტრანზაქციის ჩათვლით

გ. ზედა დონის ყველა ტრანზაქცია არ უკუიქცევა პირველი დონის ტრანზაქციის ჩათვლით

14.3.8. თუ ქვედა დონის რამდენიმე ტრანზაქცია წარმატებით დამთავრდა (მაგრამ არ დაფიქსირდა), მაგრამ საშუალო დონეზე ტრანზაქცია წარუმატებლად დამთავრდა, მაშინ:

ა. უკუიქცევა ყველა დონის ტრანზაქცია იმ ტრანზაქციების ჩათვლით, რომლებიც წარმატებით არ დამთავრდა

ბ. არ უკუიქცევა ყველა დონის ტრანზაქცია იმ ტრანზაქციების ჩათვლით, რომლებიც წარმატებით დამთავრდა

გ. უკუიქცევა ყველა დონის ტრანზაქცია იმ ტრანზაქციების ჩათვლით, რომლებიც წარმატებით დამთავრდა

14.3.9. როცა ყველა დონის ტრანზაქცია წარმატებით დამთავრდება, მაშინ ყველა ცვლილება:

ა. დაფიქსირდება იმ შემთხვევაში, როცა წარმატებით დამთავრდება პირველი დონის ტრანზაქცია

ბ. არ დაფიქსირდება იმ შემთხვევაში, როცა წარმატებით დამთავრდება პირველი დონის ტრანზაქცია

გ. დაფიქსირდება იმ შემთხვევაში, როცა წარმატებით არ დამთავრდება პირველი დონის ტრანზაქცია

14.3.10. ტრანზაქციებში აკრძალულია იმ ოპერაციების შესრულება, რომლებისთვისაც:

ა. არაა რეალიზებული შესრულებული მოქმედებების უკუიქცევა

ბ. რეალიზებულია შესრულებული მოქმედებების უკუიქცევა

გ. არაა რეალიზებული გაუქმებული მოქმედებების უკუიქცევა

14.3.11. ტრანზაქციის შიგნით აკრძალულია შემდეგი ბრძანებების გამოყენება:

ა. RESTORE DATABASE და CREATE TABLE

ბ. ALTER DATABASE, LOAD TRANSACTION და LOAD DATABASE

გ. BACKUP LOG და CREATE VIEW

დაბლოკვების მართვა

14.4.1. დაბლოკვების დაყენებისა და მოხსნის, კონფლიქტების გადაწყვეტის სამუშაოს:

ა. დაბლოკვების მენეჯერი (lock manager) ასრულებს

ბ. დაბლოკვების მენეჯერი (lock manager) არ ასრულებს

გ. არადაბლოკვების მენეჯერი (lock manager) ასრულებს

14.4.2. მანამ, სანამ ტრანზაქცია შეძლებს მონაცემების შეცვლას, ის დაბლოკვების მენეჯერისაგან:

ა. არ ითხოვს შესაბამისი დაბლოკვის დაწყებას

ბ. ითხოვს შესაბამისი დაბლოკვის დაწყებას

გ. ითხოვს შესაბამისი დაბლოკვის დამთავრებას

14.4.3. თუ მოთხოვნილი მონაცემები მოცემულ მომენტში გამოიყენება სხვა პროცესის (შესრულების სტადიაში მყოფი პროგრამა) მიერ, მაშინ დაბლოკვების მენეჯერი:

ა. მას რიგში აყენებს და აღარ შეასრულებს, როცა მიმდინარე პროცესი მუშაობას დაამთავრებს და მონაცემებს გაათავისუფლებს

ბ. არც უარყოფს დაბლოკვის მოთხოვნას და არც რიგში აყენებს მას

გ. ან უარყოფს დაბლოკვის მოთხოვნას, ან მას რიგში აყენებს და მაშინ შეასრულებს, როცა მიმდინარე პროცესი მუშაობას დაამთავრებს და მონაცემებს გაათავისუფლებს

14.4.4. თუ მონაცემები (რესურსი) მოთხოვნის მომენტში თავისუფალია, მაშინ დაბლოკვების მენეჯერი:

- ა. აკმაყოფილებს მოთხოვნას და ბლოკავს მოთხოვნილ მონაცემებს
- ბ. არ აკმაყოფილებს მოთხოვნას და ბლოკავს მოთხოვნილ მონაცემებს
- გ. აკმაყოფილებს მოთხოვნას და არ ბლოკავს მოთხოვნილ მონაცემებს

14.4.5. დაბლოკვის შემდეგ ტრანზაქციას:

- ა. არ შეუძლია მონაცემების დამუშავება
- ბ. შეუძლია მონაცემების დამუშავება
- გ. ხანდახან შეუძლია მონაცემების დამუშავება

14.4.6. რესურსის დაბლოკვის მოხსნის ლოდინის დრო:

- ა. ხანდახან შეგვიძლია ვარეგულიროთ
- ბ. არ შეგვიძლია ვარეგულიროთ
- გ. შეგვიძლია ვარეგულიროთ

14.4.7. SET LOCK_TIMEOUT 'პერიოდი' ბრძანებაში 'პერიოდი':

- ა. მიუთითებს მილიწამებს, რომლის განმავლობაშიც ტრანზაქცია დაელოდება რესურსის გათავისუფლებას
- ბ. მიუთითებს წამებს, რომლის განმავლობაშიც ტრანზაქცია დაელოდება რესურსის გათავისუფლებას
- გ. მიუთითებს წუთებს, რომლის განმავლობაშიც ტრანზაქცია დაელოდება რესურსის გათავისუფლებას

14.4.8. SET LOCK_TIMEOUT 'პერიოდი' ბრძანებაში 'პერიოდი' არგუმენტის ნაგულისხმევი მნიშვნელობაა:

- ა. 10, რაც შეესაბამება 10 მილიწამიან ლოდინს
- ბ. -1, რაც შეესაბამება უსასრულო ლოდინს
- გ. 100, რაც შეესაბამება 100 მილიწამიან ლოდინს

14.4.9. SET LOCK_TIMEOUT 'პერიოდი' ბრძანებაში 'პერიოდი' არგუმენტის მიმდინარე მნიშვნელობას გასცემს:

- ა. @@CPU_BUSY
- ბ. @@ERROR ფუნქცია
- გ. @@LOCK_TIMEOUT ფუნქცია

14.4.10. დაბლოკვის სისტემა:

- ა. უზრუნველყოფს მონაცემების დამუშავების მრავალმომხმარებლურ გარემოს
- ბ. არ უზრუნველყოფს მონაცემების დამუშავების მრავალმომხმარებლურ გარემოს
- გ. უზრუნველყოფს მონაცემების დამუშავების ერთმომხმარებლურ გარემოს

14.4.11. დაბლოკვის სისტემა გამოირთვება მხოლოდ მაშინ, როცა სერვერი მუშაობს ერთი მომხმარებლის რეჟიმში „მხოლოდ წაკითხვისათვის“:

- ა. ორი მომხმარებლის რეჟიმში „მხოლოდ წაკითხვისათვის”
- ბ. ერთი მომხმარებლის რეჟიმში „მხოლოდ წაკითხვისათვის”
- გ. სამი მომხმარებლის რეჟიმში „მხოლოდ წაკითხვისათვის”

14.4.12. დაბლოკვების მაქსიმალური რაოდენობის განსაზღვრისათვის გამოიყენება:

- ა. sp_rename შენახული პროცედურა
- ბ. sp_lock შენახული პროცედურა
- გ. sp_configure შენახული პროცედურა

14.4.13. კონკრეტული პროცესის მიერ დაყენებული დაბლოკვების შესახებ ინფორმაცია შეგვიძლია მივიღოთ:

- ა. sp_lock შენახული პროცედურის საშუალებით
- ბ. sp_configure შენახული პროცედურის საშუალებით
- გ. sp_rename შენახული პროცედურის საშუალებით

14.4.14. sp_lock შენახული პროცედურის ‘პროცესის_იდენტიფიკატორი_1’ არგუმენტი:

- ა. არ შეიცავს პროცესის საიდენტიფიკაციო ნომერს, რომლის შესახებაც გვინდა ინფორმაციის მიღება
- ბ. შეიცავს პროცესის საიდენტიფიკაციო ნომერს, რომლის შესახებაც გვინდა ინფორმაციის მიღება
- გ. შეიცავს პროცესის საიდენტიფიკაციო ნომერს, რომლის შესახებაც არ გვინდა ინფორმაციის მიღება

14.4.15. sp_lock შენახული პროცედურაში არგუმენტები არ არის მითითებული, მაშინ:

- ა. არ გაიცემა ინფორმაცია ყველა პროცესის მიერ დაყენებული დაბლოკვების შესახებ
- ბ. არ გაიცემა ინფორმაცია არც ერთი პროცესის მიერ დაყენებული დაბლოკვების შესახებ
- გ. გაიცემა ინფორმაცია ყველა პროცესის მიერ დაყენებული დაბლოკვების შესახებ

14.4.16. პროცესის შესაწყვეტად გამოიყენება:

- ა. KILL ბრძანება
- ბ. COMMIT ბრძანება
- გ. ROLLBACK ბრძანება

14.4.17. სერვერზე აქტიური პროცესების სია ინახება:

- ა. model მონაცემთა ბაზის sysprocesses სისტემურ წარმოდგენაში
- ბ. master მონაცემთა ბაზის sysprocesses სისტემურ წარმოდგენაში
- გ. master მონაცემთა ბაზის sys.all_objects სისტემურ წარმოდგენაში

14.4.18. საერთო ინფორმაციის მისაღებად სისტემური პროცესის შესახებ გამოიყენება:

- ა. sp_configure შენახული პროცედურა
- ბ. sp_lock შენახული პროცედურა
- გ. sp_who შენახული პროცედურა

თავი 15. ტრიგერები

შესავალი

- 15.1.1. ტრიგერი არის შენახული პროცედურის სპეციალური ტიპი, რომელიც სერვერის მიერ:
- ა. ავტომატურად გაიშვება ამა თუ იმ ცხრილზე მოქმედებების შესრულების დროს
 - ბ. ავტომატურად არ გაიშვება ამა თუ იმ ცხრილზე მოქმედებების შესრულების დროს
 - გ. ხელით გაიშვება ამა თუ იმ ცხრილზე მოქმედებების შესრულების დროს
- 15.1.2. თითოეული ტრიგერი დაკავშირებული:
- ა. არ არის კონკრეტულ ცხრილთან
 - ბ. არის კონკრეტულ ცხრილთან
 - გ. არის კონკრეტულ ინდექსთან
- 15.1.3. ცხრილში მონაცემების შეცვლის წინ სერვერი ავტომატურად გაუშვებს ტრიგერს, და თუ მისი შესრულება:
- ა. წარმატებით დამთავრდა, მაშინ არ შესრულდება ცხრილში მონაცემების ცვლილება
 - ბ. წარუმატებლად დამთავრდა, მაშინ შესრულდება ცხრილში მონაცემების ცვლილება
 - გ. წარმატებით დამთავრდა, მაშინ შესრულდება ცხრილში მონაცემების ცვლილება
- 15.1.4. ტრიგერის მიერ შესრულებული მოქმედებები:
- ა. არ განიხილება როგორც ერთი ტრანზაქცია
 - *ბ. განიხილება როგორც ერთი ტრანზაქცია
 - გ. განიხილება როგორც ორი ტრანზაქცია
- 15.1.5. ტრიგერები არ უნდა გამოვიყენოთ მარტივი შემოწმებების შესასრულებლად, რისთვისაც:
- ა. შეიძლება UNIQUE მთლიანობის შეზღუდვის გამოყენება
 - ბ. არ შეიძლება CHECK მთლიანობის შეზღუდვის გამოყენება
 - გ. შეიძლება CHECK მთლიანობის შეზღუდვის გამოყენება
- 15.1.6. ტრიგერები არ უნდა გამოვიყენოთ იმ მოქმედებების შესასრულებლად, რომლებიც შეიძლება შესრულდეს:
- ა. შენახული პროცედურებით ან Transact-SQL-ის პაკეტით
 - ბ. მხოლოდ შენახული პროცედურებით
 - გ. მხოლოდ შენახული Transact-SQL-ის პაკეტით
- 15.1.7. რომელი მსჯელობაა სწორი:
- ა. არსებობს ტრიგერების ორი ტიპი: INSERT TRIGGER და SELECT TRIGGER
 - ბ. არსებობს ტრიგერების სამი ტიპი: INSERT TRIGGER, UPDATE TRIGGER და DELETE TRIGGER
 - გ. არსებობს ტრიგერების ოთხი ტიპი: INSERT TRIGGER, UPDATE TRIGGER, DELETE TRIGGER და SELECT TRIGGER
- 15.1.8. INSERT TRIGGER ტიპის ტრიგერები გაიშვება:
- ა. UPDATE ბრძანებით მონაცემების ჩასმის მცდელობის დროს
 - ბ. SELECT ბრძანებით მონაცემების ჩასმის მცდელობის დროს
 - გ. INSERT ბრძანებით მონაცემების ჩასმის მცდელობის დროს

15.1.9. UPDATE TRIGGER ტიპის ტრიგერები გაიშვება:

- ა. DELETE ბრძანებით მონაცემების წაშლის მცდელობის დროს
- ბ. UPDATE ბრძანებით მონაცემების შეცვლის მცდელობის დროს
- გ. INSERT ბრძანებით მონაცემების ჩასმის მცდელობის დროს

15.1.10. DELETE TRIGGER ტიპის ტრიგერები გაიშვება:

- ა. UPDATE ბრძანებით მონაცემების შეცვლის მცდელობის დროს
- ბ. INSERT ბრძანებით მონაცემების ჩასმის მცდელობის დროს
- გ. DELETE ბრძანებით მონაცემების წაშლის მცდელობის დროს

15.1.11. ტექსტური ბლოკების დამუშავების ბრძანებების შესრულებისას, როგორცაა WRITETXT, READTXT და UPDATETXT ტრიგერი:

- ა. არ გაიშვება
- ბ. გაიშვება
- გ. აიკრძალება

15.1.12. რომელი მსჯელობაა სწორი:

- ა. ტრიგერის ქცევას არც ერთი პარამეტრი არ განსაზღვრავს:
- ბ. ტრიგერის ქცევას ერთი პარამეტრი განსაზღვრავს: AFTER
- გ. ტრიგერის ქცევას ორი პარამეტრი განსაზღვრავს: AFTER და INSTEAD OF

15.1.13. AFTER ტრიგერი სრულდება მისი გამომწვევი ბრძანებების წარმატებით შესრულების:

- ა. შუაში
- ბ. წინ
- გ. შემდეგ

15.1.14. თუ რაიმე მიზეზის გამო ბრძანებების შესრულება წარუმატებლად დამთავრდა, მაშინ ტრიგერი:

- ა. შესრულდება
- ბ. არ შესრულდება
- გ. ხანდახან შესრულდება

15.1.15. მონაცემების ცვლილება მოთხოვნის შესრულების შედეგად და ტრიგერის შესრულება სრულდება:

- ა. ორი ტრანზაქციის ტანში
- ბ. ერთი ტრანზაქციის ტანში
- გ. სამი ტრანზაქციის ტანში

15.1.16. თუ მოხდება ტრანზაქციის უკუქცევა, მაშინ:

- ა. უკუიქცევა მოთხოვნის შესრულების შედეგად გაკეთებული ცვლილებებიც
- ბ. არ უკუიქცევა მოთხოვნის შესრულების შედეგად გაკეთებული ცვლილებებიც
- გ. უკუიქცევა მოთხოვნის შესრულების შედეგად გაკეთებული ცვლილებებიც

15.1.17. AFTER-ტრიგერები განისაზღვრება მხოლოდ:

- ა. ცხრილებისთვის და არ განისაზღვრება წარმოდგენებისთვის

- ბ. წარმოდგენებისთვის და არ განისაზღვრება ცხრილებისთვის
- გ. ცხრილებისთვის და განისაზღვრება წარმოდგენებისთვის

15.1.18. თითოეული INSERT, UPDATE და DELETE ბრძანებისათვის შეგვიძლია განვსაზღვროთ:

- ა. მხოლოდ ერთი ტრიგერი
- ბ. რამდენიმე AFTER-ტრიგერი
- გ. მხოლოდ ორი ტრიგერი

15.1.19. INSTEAD OF ტრიგერი გამოიძახება ბრძანების შესრულების:

- ა. უკან
- ბ. წინ
- გ. ნაცვლად

15.1.20. INSTEAD OF-ტრიგერები შეგვიძლია განვსაზღვროთ:

- ა. როგორც ცხრილებისთვის, ისე წარმოდგენებისთვის
- ბ. მხოლოდ ცხრილებისთვის
- გ. მხოლოდ წარმოდგენებისთვის

15.1.21. თითოეული INSERT, UPDATE და DELETE ბრძანებისათვის:

- ა. არ შეგვიძლია განვსაზღვროთ მხოლოდ თითო INSTEAD OF-ტრიგერი
- ბ. შეგვიძლია განვსაზღვროთ მხოლოდ ორი INSTEAD OF-ტრიგერი
- გ. შეგვიძლია განვსაზღვროთ მხოლოდ თითო INSTEAD OF-ტრიგერი

15.1.22. ავტომატურად ყველა ტრიგერი:

- ა. AFTER-ტრიგერია
- ბ. DELETE-ტრიგერია
- გ. INSTEAD OF-ტრიგერია

15.1.23. ერთი ცხრილისთვის შეგვიძლია შევქმნათ:

- ა. მხოლოდ ერთი ერთტიპური ტრიგერი
- ბ. რამდენიმე ერთტიპური ტრიგერი
- გ. მხოლოდ ორი ერთტიპური ტრიგერი

ტრიგერის შექმნა. ტრიგერის წაშლა

15.2.1. ტრიგერების შექმნა დროებითი ან სისტემური ცხრილებისათვის:

- ა. არ შეიძლება
- ბ. შეიძლება
- გ. ხანდახან შეიძლება

15.2.2. ტრიგერს დროებით ცხრილს:

- ა. არ შეუძლია მიმართოს
- ბ. ხანდახან შეუძლია მიმართოს
- გ. შეუძლია მიმართოს

15.2.3. თუ ტრიგერს სჭირდება სისტემურ ცხრილთან მიმართვა, მაშინ უნდა გამოვიყენოთ

წარმოდგენები:

- ა. არ უნდა გამოვიყენოთ წარმოდგენები
- ბ. უნდა გამოვიყენოთ წარმოდგენები
- გ. უნდა გამოვიყენოთ ცხრილები

15.2.4. პაკეტში CREATE TRIGGER ბრძანება:

- ა. უნდა იყოს პირველი და არ უნდა გამოვიყენებოდეს მხოლოდ ერთი ცხრილის მიმართ
- ბ. არ უნდა იყოს პირველი და უნდა გამოვიყენებოდეს მხოლოდ ერთი ცხრილის მიმართ
- გ. უნდა იყოს პირველი და უნდა გამოვიყენებოდეს მხოლოდ ერთი ცხრილის მიმართ

15.2.5. რომელი მსჯელობაა სწორი:

- ა. ტრიგერები არ შეიძლება შეიქმნას მხოლოდ მიმდინარე მონაცემთა ბაზაში
- ბ. ტრიგერები შეიძლება შეიქმნას მხოლოდ მიმდინარე მონაცემთა ბაზაში
- გ. ტრიგერები შეიძლება შეიქმნას მხოლოდ სხვა მონაცემთა ბაზაში

15.2.6. რომელი მსჯელობაა სწორი:

- ა. ტრიგერის შიგნით დასაშვებია მხოლოდ სხვა მონაცემთა ბაზებთან მიმართვა
- ბ. ტრიგერის შიგნით დაუშვებელია სხვა მონაცემთა ბაზებთან მიმართვა, მათ შორის დაშორებულ სერვერზე მოთავსებულ მონაცემთა ბაზასთან
- გ. ტრიგერის შიგნით დასაშვებია სხვა მონაცემთა ბაზებთან მიმართვა, მათ შორის დაშორებულ სერვერზე მოთავსებულ მონაცემთა ბაზასთან

15.2.7. ტრიგერის შესაქმნელად გამოვიყენება:

- ა. CREATE TRIGGER ბრძანება
- ბ. CREATE INDEX ბრძანება
- გ. CREATE DATABASE ბრძანება

15.2.8. ტრიგერის სახელი მონაცემთა ბაზის ფარგლებში:

- ა. ხანდახან უნდა იყოს უნიკალური
- ბ. არ უნდა იყოს უნიკალური
- გ. უნდა იყოს უნიკალური

15.2.9. CREATE TRIGGER ბრძანებაში თუ მითითებულია AFTER არგუმენტი, მაშინ:

- ა. ტრიგერი მხოლოდ იმ შემთხვევაში გაიშვება, როცა მისი გამომძახებელი ბრძანებები წარმატებით შესრულდება
- ბ. ტრიგერი მხოლოდ იმ შემთხვევაში არ გაიშვება, როცა მისი გამომძახებელი ბრძანებები წარმატებით შესრულდება
- გ. ტრიგერი მხოლოდ იმ შემთხვევაში გაიშვება, როცა მისი გამომძახებელი ბრძანებები წარუმატებლად შესრულდება

15.2.10. CREATE TRIGGER ბრძანებაში თუ მითითებულია INSTEAD OF არგუმენტი, მაშინ:

- ა. ტრიგერი შესრულდება მისი გამომძახებელი მოთხოვნის წინ
- ბ. ტრიგერი შესრულდება მისი გამომძახებელი მოთხოვნის ნაცვლად
- გ. ტრიგერი არ შესრულდება მისი გამომძახებელი მოთხოვნის ნაცვლად

15.2.11. CREATE TRIGGER ბრძანების 'AS sql_ბრძანება' არგუმენტი:

- ა. შეიცავს Transact_SQL-ის ბრძანებებს, რომლებიც შესრულდება ტრიგერის გაშვებისას
- ბ. შეიცავს Transact_SQL-ის ბრძანებებს, რომლებიც არ შესრულდება ტრიგერის გაშვებისას
- გ. არ შეიცავს Transact_SQL-ის ბრძანებებს, რომლებიც შესრულდება ტრიგერის გაშვებისას

15.2.12.რომელი მსჯელობაა სწორი:

- ა. შეუძლებელია ტრიგერის დაკავშირება რამდენიმე ბრძანებასთან
- ბ. შესაძლებელია ტრიგერის დაკავშირება რამდენიმე ბრძანებასთან
- გ. შესაძლებელია ტრიგერის დაკავშირება მხოლოდ ერთ ბრძანებასთან

15.2.13.CREATE TRIGGER ბრძანების 'IF UPDATE (სვეტის_სახელი)' არგუმენტის გამოყენება საშუალებას:

- ა. გვადლევს ტრიგერი შევასრულოთ ცხრილის კონკრეტული სტრიქონის მოდიფიცირების დროს
- ბ. არ გვადლევს ტრიგერი შევასრულოთ ცხრილის კონკრეტული სვეტის მოდიფიცირების დროს
- გ. გვადლევს ტრიგერი შევასრულოთ ცხრილის კონკრეტული სვეტის მოდიფიცირების დროს

15.2.14.CREATE TRIGGER ბრძანების 'IF UPDATE (სვეტის_სახელი)' არგუმენტი გამოიყენება მხოლოდ:

- ა. INSERT და DELETE ბრძანებებისთვის
- ბ. INSERT და UPDATE ბრძანებებისთვის
- გ. DELETE და UPDATE ბრძანებებისთვის

15.2.15.CREATE TRIGGER ბრძანების '{ AND | OR } UPDATE (სვეტის_სახელი)' არგუმენტი გამოიყენება 'IF UPDATE (სვეტის_სახელი)' არგუმენტთან ერთად, თუ:

- ა. საჭირო არ არის ტრიგერის გაშვება რამდენიმე სვეტის მოდიფიცირების შემთხვევაში
- ბ. საჭიროა ტრიგერის გაშვება რამდენიმე სვეტის მოდიფიცირების შემთხვევაში
- გ. საჭიროა ტრიგერის გაშვება მხოლოდ ერთი სვეტის მოდიფიცირების შემთხვევაში

15.2.16.CREATE TRIGGER ბრძანების IF (COLUMNS_UPDATED()) არგუმენტი გვატყობინებს თუ რომელი:

- ა. სვეტები შეიცვალა ან დაემატა
- ბ. სტრიქონები შეიცვალა ან დაემატა
- გ. ცხრილები შეიცვალა ან დაემატა

15.2.17.CREATE TRIGGER ბრძანების IF (COLUMNS_UPDATED()) არგუმენტი გამოიყენება მხოლოდ:

- ა. DELETE და UPDATE ბრძანებებისთვის
- ბ. UPDATE და INSERT ბრძანებებისათვის
- გ. INSERT და DELETE ბრძანებებისთვის

15.2.18.COLUMNS_UPDATED() ფუნქცია გასცემს ორობით რიცხვს, რომლის თითოეული:

- ა. ბაიტი შეესაბამება კონკრეტულ სვეტს
- ბ. ბიტი შეესაბამება კონკრეტულ სტრიქონს
- გ. ბიტი შეესაბამება კონკრეტულ სვეტს

15.2.19.COLUMNS_UPDATED() ფუნქციის მიერ გაცემულ ორობით რიცხვში თუ:

- ა. ბიტი 0 მდგომარეობაშია, ეს ნიშნავს, რომ შესაბამისი სვეტი შეიცვალა
- ბ. ბიტი 1 მდგომარეობაშია, ეს ნიშნავს, რომ შესაბამისი სვეტი შეიცვალა
- გ. ბიტი 1 მდგომარეობაშია, ეს ნიშნავს, რომ შესაბამისი სტრიქონი შეიცვალა

15.2.20.COLUMNS_UPDATED() ფუნქციის მიერ გაცემულ ორობით რიცხვში:

- ა. უმცროსი ბიტი შეესაბამება მეოთხე სვეტს, მეორე ბიტი მარჯვნიდან შეესაბამება მეხუთე სვეტს და ა.შ.
- ბ. უფროსი ბიტი შეესაბამება პირველ სვეტს, მეორე ბიტი მარჯვნიდან შეესაბამება მეორე სვეტს და ა.შ.
- გ. უმცროსი ბიტი შეესაბამება პირველ სვეტს, მეორე ბიტი მარჯვნიდან შეესაბამება მეორე სვეტს და ა.შ.

15.2.21.CREATE TRIGGER ბრძანების 'ბიტების_ნილაბი' არგუმენტი:

- ა. განსაზღვრავს ბიტების ნილაბს ერთ ან მეტ სვეტში ცვლილებების განსაზღვრისათვის
- ბ. არ განსაზღვრავს ბიტების ნილაბს ერთ ან მეტ სვეტში ცვლილებების განსაზღვრისათვის
- გ. განსაზღვრავს ბიტების ნილაბს ერთ ან მეტ სტრიქონში ცვლილებების განსაზღვრისათვის

15.2.22.ინფორმაცია ტრიგერების შესახებ მოთავსებულია:

- ა. model მონაცემთა ბაზის sys.triggers სისტემურ წარმოდგენაში
- ბ. master მონაცემთა ბაზის sys.triggers სისტემურ წარმოდგენაში
- გ. მიმდინარე მონაცემთა ბაზის sys.triggers სისტემურ წარმოდგენაში

15.2.23.INSERT და UPDATE ბრძანებებით შეცვლილი სვეტების სიის მისაღებად გამოიყენება ფუნქცია:

- ა. COLUMN_UPDATE()
- ბ. TRIGGER_NESTLEVEL
- გ. SET NOCOUNT

15.2.24.სვეტის ცვლილების ფაქტის დასადასტურებლად გამოიყენება ფუნქცია:

- ა. UPDATE(სვეტის_სახელი)
- ბ. TRIGGER_NESTLEVEL
- გ. COLUMN_UPDATE()

15.2.25.ინფორმაციის მისაღებად სტრიქონების რაოდენობის შესახებ, რომლებიც შეიცვლება ტრიგერის წარმატებით დამთავრების შემთხვევაში, შეგვიძლია გამოვიყენოთ ფუნქცია:

- ა. TRIGGER_NESTLEVEL
- ბ. COLUMN_UPDATE()
- გ. @@ROWCOUNT

15.2.26.რომელი მსჯელობაა სწორი:

- ა. ტრიგერი არ სრულდება როგორც არაცხადად განსაზღვრული ტრანზაქცია, ამიტომ ტრიგერის შიგნით დასაშვებია ტრანზაქციების მართვის ბრძანებების გამოყენება
- ბ. ტრიგერი სრულდება როგორც არაცხადად განსაზღვრული ტრანზაქცია, ამიტომ ტრიგერის შიგნით დასაშვებია ტრანზაქციების მართვის ბრძანებების გამოყენება

გ. ტრიგერი სრულდება როგორც არაცხადად განსაზღვრული ტრანზაქცია, ამიტომ ტრიგერის შიგნით დაუშვებელია ტრანზაქციების მართვის ბრძანებების გამოყენება

15.2.27. შეცდომის შემთხვევაში ტრიგერის მუშაობის შესაწყვეტად და შესრულებული ცვლილებების გასაუქმებლად უნდა გამოვიყენოთ:

- ა. TRIGGER_NESTLEVEL ბრძანება
- ბ. COMMIT TRANSACTION ბრძანება
- გ. ROLLBACK TRANSACTION ბრძანება

15.2.28. ტრიგერის წარმატებით დამთავრების შემთხვევაში უნდა გამოვიყენოთ COMMIT TRANSACTION ბრძანება:

- ა. COMMIT TRANSACTION ბრძანება
- ბ. TRIGGER_NESTLEVEL ბრძანება
- გ. ROLLBACK TRANSACTION ბრძანება

15.2.29. როცა სერვერი იწყებს ტრიგერის შესრულებას, ის ქმნის ორ სპეციალურ ცხრილს:

- ა. inserted და updated
- ბ. updated და deleted
- გ. inserted და deleted

15.2.30. ტრიგერის მუშაობისას inserted და deleted ცხრილებში მოთავსებულია სტრიქონები, რომლებიც შესაბამისად:

- ა. ჩასმული და წაშლილი იქნება ტრანზაქციის დამთავრებისას
- ბ. ჩასმული და წაშლილი იქნება ტრანზაქციის დაწყებისას
- გ. შეცვლილი იქნება ტრანზაქციის დამთავრებისას

15.2.31. inserted და deleted ცხრილების სტრუქტურა:

- ა. იდენტური არაა იმ ცხრილის სტრუქტურისა, რომლისთვისაც ტრიგერია განკუთვნილი
- ბ. იდენტურია იმ ცხრილის სტრუქტურისა, რომლისთვისაც ტრიგერია განკუთვნილი
- გ. იდენტურია იმ ცხრილის სტრუქტურისა, რომლისთვისაც ტრიგერი არაა განკუთვნილი

15.2.32. თუ ტრიგერი გამოიწვია INSERT ბრძანებამ, მაშინ inserted ცხრილი:

- ა. არ შეიცავს ყველა სვეტს, რომელთა ჩასმასაც ვცდილობთ ცხრილში
- ბ. შეიცავს ყველა სვეტს, რომელთა ჩასმასაც ვცდილობთ ცხრილში
- გ. შეიცავს ყველა სვეტს, რომელთა ჩასმას არ ვცდილობთ ცხრილში

15.2.33. თუ ტრიგერი გამოიწვია INSERT ბრძანებამ, მაშინ deleted ცხრილი:

- ა. არც ერთ სვეტს არ შეიცავს
- ბ. ყველა სვეტს არ შეიცავს
- გ. ყველა სტრიქონს შეიცავს

15.2.34. თუ ტრიგერი გამოიწვია INSERT ბრძანებამ, მაშინ ტრიგერის:

- ა. დამთავრების შემდეგ inserted ცხრილიდან ყველა სტრიქონი ჩაემატება ცხრილს
- ბ. დაწყებამდე inserted ცხრილიდან ყველა სტრიქონი ჩაემატება ცხრილს
- გ. დამთავრების შემდეგ inserted ცხრილიდან არც ერთი სტრიქონი არ ჩაემატება ცხრილს

15.2.35.თუ ტრიგერი გამოიწვია DELETE ბრძანებამ, მაშინ:

- ა. inserted ცხრილში მოთავსდება ის სტრიქონები, რომელთა წაშლასაც ვცდილობთ
- ბ. deleted ცხრილში მოთავსდება ის სტრიქონები, რომელთა წაშლასაც არ ვცდილობთ
- გ. deleted ცხრილში მოთავსდება ის სტრიქონები, რომელთა წაშლასაც ვცდილობთ

15.2.36.თუ ტრიგერი გამოიწვია DELETE ბრძანებამ, მაშინ:

- ა. deleted ცხრილში არ იქნება მოთავსებული არც ერთი სტრიქონი
- ბ. inserted ცხრილში იქნება მოთავსებული ყველა სტრიქონი
- გ. inserted ცხრილში არ იქნება მოთავსებული არც ერთი სტრიქონი

15.2.37.თუ ტრიგერი გამოიწვია UPDATE ბრძანებამ, მაშინ ამ ბრძანების შესრულების დროს deleted ცხრილში:

- ა. მოთავსებული არ იქნება სტრიქონების ძველი მნიშვნელობები, რომლებიც წაიშლება ტრიგერის წარმატებით დამთავრების შემთხვევაში
- ბ. მოთავსებული იქნება სტრიქონების ძველი მნიშვნელობები, რომლებიც წაიშლება ტრიგერის წარმატებით დამთავრების შემთხვევაში
- გ. მოთავსებული იქნება სტრიქონების ძველი მნიშვნელობები, რომლებიც არ წაიშლება ტრიგერის წარმატებით დამთავრების შემთხვევაში

15.2.38.თუ ტრიგერი გამოიწვია UPDATE ბრძანებამ, მაშინ სტრიქონების:

- ა. ახალი მნიშვნელობები მოთავსებული იქნება deleted ცხრილში
- ბ. ძველი მნიშვნელობები მოთავსებული იქნება inserted ცხრილში
- გ. ახალი მნიშვნელობები მოთავსებული იქნება inserted ცხრილში

15.2.39.თუ ტრიგერი გამოიწვია UPDATE ბრძანებამ, მაშინ სტრიქონების ახალი მნიშვნელობები მოთავსებული იქნება inserted ცხრილში. ეს სტრიქონები:

- ა. დაემატება საწყის ცხრილს ტრიგერის წარმატებით დამთავრების შემთხვევაში
- ბ. არ დაემატება საწყის ცხრილს ტრიგერის წარმატებით დამთავრების შემთხვევაში
- გ. დაემატება საწყის ცხრილს ტრიგერის წარუმატებლად დამთავრების შემთხვევაში

15.2.40.ტრიგერის წასაშლელად გამოიყენება:

- ა. DROP TRIGGER ბრძანება
- ბ. CREATE TRIGGER ბრძანება
- გ. ALTER TRIGGER ბრძანება

15.2.41.რომელი მსჯელობაა სწორი:

- ა. შეუძლებელია რამდენიმე ტრიგერის ერთდროულად წაშლა
- ბ. შესაძლებელია რამდენიმე ტრიგერის ერთდროულად წაშლა
- გ. შესაძლებელია მხოლოდ ორი ტრიგერის ერთდროულად წაშლა

15.2.42.CREATE TRIGGER Trigeri_4 ON Shemkveti_1 FOR DELETE AS
IF EXISTS

```
(  
SELECT * FROM deleted WHERE iuridiuli_fizikuri = N'იურიდიული'  
)  
BEGIN
```

```

PRINT N'იურიდიული პირის წაშლის მცდელობა'
PRINT N'წაშლა აკრძალულია'
ROLLBACK TRANSACTION
END

```

ბრძანების შესრულების შედეგად:

- ა. არ შეიქმნება DELETE ტრიგერი, რომელიც Shemkveti_1 ცხრილიდან არ წაგვამლევინებს იურიდიულ პირებს
- ბ. შეიქმნება DELETE ტრიგერი, რომელიც Shemkveti_1 ცხრილიდან წაგვამლევინებს იურიდიულ პირებს
- გ. შეიქმნება DELETE ტრიგერი, რომელიც Shemkveti_1 ცხრილიდან არ წაგვამლევინებს იურიდიულ პირებს

15.2.43. იმისათვის, რომ შევქმნათ DELETE ტრიგერი, რომელიც Shemkveti_1 ცხრილიდან არ წაგვამლევინებს იურიდიულ პირებს, უნდა შევასრულოთ ბრძანება:

```

ა. CREATE TRIGGER Trigeri_4 ON Shemkveti_1 FOR DELETE AS
IF EXISTS
(
SELECT * FROM deleted WHERE iuridiuli_fizikuri = N'ფიზიკური'
)
BEGIN
PRINT N'ფიზიკური პირის წაშლის მცდელობა'
PRINT N'წაშლა აკრძალულია'
ROLLBACK TRANSACTION
END
ბ. CREATE TRIGGER Trigeri_4 ON Shemkveti_1 FOR DELETE AS
IF EXISTS
(
SELECT * FROM deleted WHERE iuridiuli_fizikuri = N'იურიდიული'
)
BEGIN
PRINT N'იურიდიული პირის წაშლის მცდელობა'
PRINT N'წაშლა აკრძალულია'
ROLLBACK TRANSACTION
END
გ. CREATE TRIGGER Trigeri_4 ON Shemkveti_1 FOR DELETE AS
IF EXISTS
(
SELECT * FROM deleted WHERE iuridiuli_fizikuri = N'იურიდიული'
)
BEGIN
PRINT N'იურიდიული პირის წაშლის მცდელობა'
PRINT N'წაშლა აკრძალულია'
COMMIT TRANSACTION
END

```

15.2.44. თუ ტრიგერის შექმნა იწყება CREATE TRIGGER Trigeri_4 ON Shemkveti_1 FOR DELETE AS

სტრიქონით, მაშინ შეიქმნება:

- ა. DELETE ტრიგერი
- ბ. INSERT ტრიგერი
- გ. UPDATE ტრიგერი

15.2.45.თუ ტრიგერის შექმნა იწყება CREATE TRIGGER Trigeri_4 ON Shemkveti_1 FOR INSERT AS სტრიქონით, მაშინ შეიქმნება:

- ა. DELETE ტრიგერი
- ბ. INSERT ტრიგერი
- გ. UPDATE ტრიგერი

15.2.46.თუ ტრიგერის შექმნა იწყება CREATE TRIGGER Trigeri_4 ON Shemkveti_1 FOR UPDATE AS სტრიქონით, მაშინ შეიქმნება:

- ა. DELETE ტრიგერი
- ბ. INSERT ტრიგერი
- გ. UPDATE ტრიგერი

15.2.47.DROP TRIGGER Trigeri_1; ბრძანების შესრულების შედეგად:

- ა. წაიშლება Trigeri_1 ტრიგერი
- ბ. შეიქმნება Trigeri_1 ტრიგერი
- გ. არ წაიშლება Trigeri_1 ტრიგერი

15.2.48. Trigeri_1 ტრიგერის წასაშლელად უნდა შევასრულოთ ბრძანება:

- ა. CREATE TRIGGER Trigeri_1;
- ბ. DROP TRIGGER Trigeri_1;
- გ. DROP TRIGGER Trigeri_55;

ტრიგერების შექმნისას გასათვალისწინებელი რეკომენდაციები. ტრიგერების მართვა. ტრიგერის სახელის შეცვლა. ტრიგერის შესახებ ინფორმაციის მიღება

15.3.1. რომელი მსჯელობაა სწორი:

- ა. ტრიგერის ტანში დასაშვებია Transact_SQL-ის ბრძანებების გამოყენება, აგრეთვე, სისტემური და მომხმარებლის მიერ განსაზღვრული შენახული პროცედურების გამოძახება
- ბ. ტრიგერის ტანში დაუშვებელია Transact_SQL-ის ბრძანებების გამოყენება, აგრეთვე, სისტემური და მომხმარებლის მიერ განსაზღვრული შენახული პროცედურების გამოძახება
- გ. ტრიგერის ტანში დასაშვებია Transact_SQL-ის ბრძანებების გამოყენება და დაუსვებელია სისტემური და მომხმარებლის მიერ განსაზღვრული შენახული პროცედურების გამოძახება

15.3.2. ტრიგერმა უმჯობესია:

- ა. ყოველთვის დააბრუნოს შედეგი
- ბ. ხშირად დააბრუნოს შედეგი
- გ. არ დააბრუნოს შედეგი

15.3.3. რომელი მსჯელობაა სწორი:

- ა. ტრიგერს ხანდახან შეუძლია შედეგის დაბრუნება
- ბ. ტრიგერს შეუძლია შედეგის დაბრუნება
- გ. ტრიგერს არ შეუძლია შედეგის დაბრუნება

15.3.4. ტრიგერის შესრულების შემდეგ სერვერი გასცემს შეტყობინებას ბრძანებების ჯგუფის წარმატებით შესრულების შესახებ. ამ შეტყობინების აკრძალვისათვის:

- ა. უნდა შევასრულოთ DROP TRIGGER ბრძანება
- ბ. არ უნდა შევასრულოთ SET NOCOUNT ბრძანება
- გ. უნდა შევასრულოთ SET NOCOUNT ბრძანება

15.3.5. რომელი მსჯელობაა სწორი:

- ა. არ არის რეკომენდებული ტრიგერიდან დროებით ცხრილებთან მიმართვა თუ არ არის დროებითი ცხრილების სტრუქტურის მუდმივობის გარანტია
- ბ. რეკომენდებულია ტრიგერიდან დროებით ცხრილებთან მიმართვა თუ არ არის დროებითი ცხრილების სტრუქტურის მუდმივობის გარანტია
- გ. არ არის რეკომენდებული ტრიგერიდან დროებით ცხრილებთან მიმართვა თუ არის დროებითი ცხრილების სტრუქტურის მუდმივობის გარანტია

15.3.6. ტრიგერის ტანში SET ბრძანებების გამოყენება:

- ა. ხანდახან დაუშვებელია
- ბ. დაუშვებელია
- გ. დასაშვებია

15.3.7. SET ბრძანებებით დაყენებული მნიშვნელობები ძალაშია:

- ა. ტრიგერის მუშაობის დამთავრების შემდეგ
- ბ. მხოლოდ ტრიგერის მუშაობის პერიოდში
- გ. ტრიგერის მუშაობის დაწყებამდე

15.3.8. ტრიგერის სახელის შესაცვლელად გამოიყენება:

- ა. sp_rename სისტემური შენახული პროცედურა
- ბ. sp_help სისტემური შენახული პროცედურა
- გ. sp_delete სისტემური შენახული პროცედურა

15.3.9. ტრიგერის გამოძახების დროს შესრულებული კოდის მისაღებად უნდა შევასრულოთ:

- ა. sp_delete სისტემური შენახული პროცედურა
- ბ. sp_rename სისტემური შენახული პროცედურა
- გ. sp_helptext სისტემური შენახული პროცედურა

15.3.10. კონკრეტული ცხრილისთვის განსაზღვრული ტრიგერების სიის მისაღებად გამოიყენება:

- ა. sp_helptext სისტემური შენახული პროცედურა
- ბ. sp_helptrigger სისტემური შენახული პროცედურა
- გ. sp_delete სისტემური შენახული პროცედურა

15.3.11. იმ ობიექტების სახეხავად, რომლებზეც ტრიგერია დამოკიდებული, უნდა შევასრულოთ:

- ა. sp_depends სისტემური შენახული პროცედურა
- ბ. sp_helptrigger სისტემური შენახული პროცედურა

გ. sp_helptext სისტემური შენახული პროცედურა

15.3.12.EXEC sp_rename 'Trigeri_1', 'Trigeri_2', 'OBJECT'; ბრძანების შესრულების შედეგად:

- ა. Trigeri_1 ტრიგერს არ დაერქმევა ახალი სახელი - Trigeri_2
- ბ. Trigeri_1 ტრიგერს დაერქმევა ახალი სახელი - Trigeri_2
- გ. Trigeri_2 ტრიგერს დაერქმევა ახალი სახელი - Trigeri_1

15.3.13.იმისათვის, რომ Trigeri_1 ტრიგერს დავარქვათ ახალი სახელი - Trigeri_2, უნდა შევასრულოთ ბრძანება:

- ა. EXEC sp_rename 'Trigeri_2', 'Trigeri_1', 'OBJECT';
- ბ. EXEC sp_rename 'Trigeri_1', 'Trigeri_2', 'OBJECT';
- გ. EXEC sp_rename 'Trigeri_1', 'Trigeri_2', 'TABLE';

15.3.14.EXEC sp_helptext 'Trigeri_3'; ბრძანების შესრულების შედეგად:

- ა. შეიქმნება Trigeri_3 ტრიგერი
- ბ. არ გაიცემა ინფორმაცია Trigeri_3 ტრიგერის შესაბამისი კოდის შესახებ
- გ. გაიცემა ინფორმაცია Trigeri_3 ტრიგერის შესაბამისი კოდის შესახებ

15.3.15.ინფორმაციის Trigeri_3 ტრიგერის შესაბამისი კოდის შესახებ, უნდა შევასრულოთ ბრძანება:

- ა. EXEC sp_rename 'Trigeri_3';
- ბ. EXEC sp_helptrigger 'Shemkveti';
- გ. EXEC sp_helptext 'Trigeri_3';

15.3.16.EXEC sp_helptrigger 'Shemkveti'; ბრძანების შესრულების შედეგად:

- ა. გაიცემა ინფორმაცია Shemkveti ცხრილის ტრიგერების შესახებ
- ბ. არ გაიცემა ინფორმაცია Shemkveti ცხრილის ტრიგერების შესახებ
- გ. გაიცემა ინფორმაცია Shemkveti ცხრილის სვეტების შესახებ

15.3.17.ინფორმაციის მისაღებად Shemkveti ცხრილის ტრიგერების შესახებ, უნდა შევასრულოთ ბრძანება:

- ა. EXEC sp_helptrigger 'Shemkveti';
- ბ. EXEC sp_helptext 'Trigeri_3';
- გ. EXEC sp_rename 'Trigeri_3';

15.3.18.EXEC sp_helptrigger 'Shemkveti', 'DELETE'; ბრძანების შესრულების შედეგად:

- ა. გაიცემა ინფორმაცია Shemkveti ცხრილის INSERT-ტრიგერების შესახებ
- ბ. გაიცემა ინფორმაცია Shemkveti ცხრილის DELETE-ტრიგერების შესახებ
- გ. გაიცემა ინფორმაცია Shemkveti ცხრილის UPDATE-ტრიგერების შესახებ

15.3.19.ინფორმაციის მისაღებად Shemkveti ცხრილის DELETE-ტრიგერების შესახებ, უნდა შევასრულოთ ბრძანება:

- ა. EXEC sp_helptrigger 'Shemkveti', 'INSERT';
- ბ. EXEC sp_helptrigger 'Shemkveti', 'DELETE';
- გ. EXEC sp_helptrigger 'Shemkveti', 'UPDATE';

15.3.20.EXEC sp_helptrigger 'Shemkveti', 'INSERT'; ბრძანების შესრულების შედეგად:

- ა. გაიცემა ინფორმაცია Shemkveti ცხრილის INSERT-ტრიგერების შესახებ
- ბ. გაიცემა ინფორმაცია Shemkveti ცხრილის DELETE-ტრიგერების შესახებ
- გ. გაიცემა ინფორმაცია Shemkveti ცხრილის UPDATE-ტრიგერების შესახებ

15.3.21.ინფორმაციის მისაღებად Shemkveti ცხრილის INSERT-ტრიგერების შესახებ, უნდა შევასრულოთ ბრძანება:

- ა. EXEC sp_helptrigger 'Shemkveti', 'INSERT';
- ბ. EXEC sp_helptrigger 'Shemkveti', 'DELETE';
- გ. EXEC sp_helptrigger 'Shemkveti', 'UPDATE';

15.3.22.EXEC sp_helptrigger 'Shemkveti', 'UPDATE'; ბრძანების შესრულების შედეგად:

- ა. გაიცემა ინფორმაცია Shemkveti ცხრილის INSERT-ტრიგერების შესახებ
- ბ. გაიცემა ინფორმაცია Shemkveti ცხრილის DELETE-ტრიგერების შესახებ
- გ. გაიცემა ინფორმაცია Shemkveti ცხრილის UPDATE-ტრიგერების შესახებ

15.3.23.ინფორმაციის მისაღებად Shemkveti ცხრილის UPDATE-ტრიგერების შესახებ, უნდა შევასრულოთ ბრძანება:

- ა. EXEC sp_helptrigger 'Shemkveti', 'INSERT';
- ბ. EXEC sp_helptrigger 'Shemkveti', 'DELETE';
- გ. EXEC sp_helptrigger 'Shemkveti', 'UPDATE';

15.3.24.EXEC sp_depends 'Personali2_ganyofileba'; ბრძანების შესრულების შედეგად:

- ა. გაიცემა ინფორმაცია იმ ობიექტების შესახებ, რომლებზეც Personali2_ganyofileba ტრიგერია დამოკიდებული
- ბ. არ გაიცემა ინფორმაცია იმ ობიექტების შესახებ, რომლებზეც Personali2_ganyofileba ტრიგერია დამოკიდებული
- გ. გაიცემა ინფორმაცია იმ ობიექტების შესახებ, რომლებზეც Personali2_ganyofileba ტრიგერი არ არის დამოკიდებული

15.3.25.ინფორმაციის მისაღებად იმ ობიექტების შესახებ, რომლებზეც Personali2_ganyofileba ტრიგერია დამოკიდებული, უნდა შევასრულოთ ბრძანება:

- ა. EXEC sp_helptrigger 'Shemkveti', 'UPDATE';
- ბ. EXEC sp_depends 'Personali2_ganyofileba';
- გ. EXEC sp_rename 'Trigeri_3';

თავი 16. ცხრილური გამოსახულებები

წარმოებული ცხრილები. სვეტებისთვის ფსევდონიმების მინიჭება. არგუმენტების გამოყენება. ჩადგმულობა. მრავლობითი მიმართვები

16.1.1. ცხრილური გამოსახულება არის მოთხოვნა, რომელიც ვირტუალურ ცხრილს:

- ა. გასცემს
- ბ. არ გასცემს
- გ. შლის

16.1.2. ცხრილური გამოსახულება:

- ა. არ შეგვიძლია მონაცემების დამუშავების ინსტრუქციებში ცხრილების მსგავსად გამოვიყენოთ
- ბ. შეგვიძლია მონაცემების დამუშავების ინსტრუქციებში ცხრილების მსგავსად გამოვიყენოთ
- გ. შეგვიძლია მონაცემების დამუშავების ინსტრუქციებში ცხრილების მსგავსად არ გამოვიყენოთ

16.1.3. SQL სერვერი ცხრილური გამოსახულებების:

- ა. ორ ტიპს უზრუნველყოფს
- ბ. სამ ტიპს უზრუნველყოფს
- გ. ოთხ ტიპს უზრუნველყოფს

16.1.4. SQL სერვერი ცხრილური გამოსახულებების შემდეგ ტიპებს უზრუნველყოფს:

- ა. წარმოებული ცხრილები, საერთო ცხრილური გამოსახულებები, წარმოდგენები და შემცვლელი ცხრილური ფუნქციები
- ბ. მხოლოდ წარმოებული ცხრილები და საერთო ცხრილური გამოსახულებები
- გ. მხოლოდ წარმოდგენები

16.1.5. წარმოებულ ცხრილებს სხვანაირად:

- ა. ცხრილური ქვემოთხოვნები ეწოდება
- ბ. არაცხრილური ქვემოთხოვნები ეწოდება
- გ. სტრიქონული მოთხოვნები ეწოდება

16.1.6. წარმოებული ცხრილები:

- ა. შიდა SELECT მოთხოვნის FROM განყოფილებაში განისაზღვრება
- ბ. გარე SELECT მოთხოვნის FROM განყოფილებაში განისაზღვრება
- გ. გარე SELECT მოთხოვნის WHERE განყოფილებაში განისაზღვრება

16.1.7. როგორც კი გარე მოთხოვნა მთავრდება, წარმოებული ცხრილი:

- ა. ინახება
- ბ. არ იშლება
- გ. იშლება

16.1.8. ნებისმიერი ტიპის ცხრილური გამოსახულების განსაზღვრა შეიძლება მოთხოვნაში, რომელიც:

- ა. სამ პირობას აკმაყოფილებს: სტრიქონების მიმდევრობა ნებისმიერი უნდა იყოს, ყველა

სვეტს უნდა ჰქონდეს სახელი და სვეტების სახელები უნიკალური უნდა იყოს

ბ. ორ პირობას აკმაყოფილებს: სტრიქონების მიმდევრობა ნებისმიერი უნდა იყოს, ყველა სვეტს უნდა ჰქონდეს სახელი

გ. ერთ პირობას აკმაყოფილებს: სტრიქონების მიმდევრობა ნებისმიერი უნდა იყოს

16.1.9. ANSI SQL სტანდარტი:

ა. იძლევა ORDER BY განყოფილების გამოყენების უფლებას იმ მოთხოვნებში, რომლებიც ცხრილურ გამოსახულებებს განსაზღვრავენ

ბ. არ იძლევა ORDER BY განყოფილების გამოყენების უფლებას იმ მოთხოვნებში, რომლებიც ცხრილურ გამოსახულებებს განსაზღვრავენ

გ. არ იძლევა ORDER BY განყოფილების გამოყენების უფლებას იმ მოთხოვნებში, რომლებიც ცხრილურ გამოსახულებებს არ განსაზღვრავენ

16.1.10.თუ TOP ელემენტისა და ORDER BY განყოფილების შემცველ მოთხოვნაში მოცემულია ცხრილური გამოსახულება, მაშინ ORDER BY განყოფილების გამოყენება:

ა. აკრძალულია

ბ. დაუშვებელია

გ. დასაშვებია

16.1.11.SELECT მოთხოვნაში სვეტის ფსევდონიმი:

ა. უნდა მივანიჭოთ ყველა გამოსახულებას, რომელიც გამოიყენება ცხრილური გამოსახულების განსაზღვრისთვის

ბ. არ უნდა მივანიჭოთ ყველა გამოსახულებას, რომელიც გამოიყენება ცხრილური გამოსახულების განსაზღვრისთვის

გ. უნდა მივანიჭოთ ყველა გამოსახულებას, რომელიც არ გამოიყენება ცხრილური გამოსახულების განსაზღვრისთვის

16.1.12.თუ ცხრილური გამოსახულების განმსაზღვრელ მოთხოვნაში ორი ცხრილი ერთიანდება, რომლებშიც ერთი და იგივე სახელის მქონე სვეტია, მაშინ მათ:

ა. სხვადასხვა ფსევდონიმი არ უნდა მივანიჭოთ

ბ. სხვადასხვა ფსევდონიმი უნდა მივანიჭოთ

გ. ერთნაირი ფსევდონიმი უნდა მივანიჭოთ

საერთო ცხრილური გამოსახულებები. სვეტებისთვის ფსევდონიმების დანიშვნა. არგუმენტების გამოყენება. რამდენიმე საერთო ცხრილური გამოსახულების განსაზღვრა. მრავლობითი მიმართვები

16.2.1. საერთო ცხრილურ გამოსახულებებში:

ა. შეუძლებელია სვეტებისთვის ფსევდონიმების დანიშვნის ორი ფორმის გამოყენება: ჩადგმული (შიდა) და გარე

ბ. შესაძლებელია სვეტებისთვის ფსევდონიმების დანიშვნის ორი ფორმის გამოყენება: ჩადგმული (შიდა) და გარე

გ. შესაძლებელია სვეტებისთვის ფსევდონიმების დანიშვნის მხოლოდ ერთი ფორმის გამოყენება - ჩადგმული (შიდა)

APPLY ოპერაცია

16.3.1. APPLY არის ცხრილური ოპერაცია და მოთხოვნის:

- ა. FROM განყოფილებაში გამოიყენება
- ბ. WHERE განყოფილებაში გამოიყენება
- გ. HAVING განყოფილებაში გამოიყენება

16.3.2. APPLY ოპერაციის:

- ა. ერთი ტიპი არსებობს: CROSS APPLY
- ბ. ორი ტიპი არსებობს: CROSS APPLY და OUTER APPLY
- გ. ტიპი არსებობს: OUTER APPLY

16.3.3. APPLY ოპერაცია:

- ა. ორი შესასვლელი ცხრილის (მარცხენა და მარჯვენა) მიმართ გამოიყენება, რომელთაგან მეორე (მარჯვენა) არ შეიძლება ცხრილური გამოსახულება იყოს
- ბ. ორი შესასვლელი ცხრილის (მარცხენა და მარჯვენა) მიმართ გამოიყენება, რომელთაგან მეორე (მარჯვენა) შეიძლება ცხრილური გამოსახულება იყოს
- გ. ორი შესასვლელი ცხრილის (მარცხენა და მარჯვენა) მიმართ გამოიყენება, რომელთაგან მეორე (მარცხენა) შეიძლება ცხრილური გამოსახულება იყოს

16.3.4. APPLY ოპერაციაში მარჯვენა ცხრილი:

- ა. წარმოებული ცხრილია ან ჩამნაცვლებელი ცხრილური ფუნქცია
- ბ. მხოლოდ წარმოებული ცხრილია
- გ. მხოლოდ ჩამნაცვლებელი ცხრილური ფუნქციაა

თავი 17. მონაცემების რეორგანიზება და დაჯგუფების ნაკრებები

მონაცემების გაშლა. მონაცემების გაშლა T-SQL ენის PIVOT ოპერაციის გამოყენებით

17.1.1. მონაცემების გაშლა:

- ა. ნიშნავს სვეტების გარდაქმნას სტრიქონების
- ბ. არ ნიშნავს სტრიქონების გარდაქმნას სვეტებად
- გ. ნიშნავს სტრიქონების გარდაქმნას სვეტებად

17.1.2. მონაცემების შეკვცა:

- ა. ნიშნავს სვეტების გარდაქმნას სტრიქონებად
- ბ. არ ნიშნავს სვეტების გარდაქმნას სტრიქონებად
- გ. ნიშნავს სტრიქონების გარდაქმნას სვეტების

17.1.3. დამჯგუფებელი ნაკრები არის:

- ა. სვეტების ერთობლიობა, რომლის მიხედვით მონაცემების დაჯგუფება არ სრულდება
- ბ. სვეტების ერთობლიობა, რომლის მიხედვით მონაცემების დაჯგუფება სრულდება
- გ. სტრიქონების ერთობლიობა, რომლის მიხედვით მონაცემების დაჯგუფება სრულდება

17.1.4. მონაცემების გაშლა არის მონაცემების მობრუნების მეთოდი, როცა:

- ა. სტრიქონები სვეტებად გარდაიქმნება შემაჯამებელი მონაცემების შესაძლო გამოთვლით
- ბ. სტრიქონები სვეტებად არ გარდაიქმნება შემაჯამებელი მონაცემების შესაძლო გამოთვლით
- გ. სვეტები სტრიქონებად გარდაიქმნება შემაჯამებელი მონაცემების შესაძლო გამოთვლით

17.1.5. მონაცემების გაშლის თითოეული მოთხოვნა მოიცავს ლოგიკური დამუშავების:

- ა. სამ სტადიას: ა) დაჯგუფების სტადია, დაკავშირებული დამაჯგუფებელ ან სტრიქონულ ელემენტთან; ბ) გახსნის სტადია, დაკავშირებული გაშლასთან ან სტრიქონულ ელემენტთან; გ) შედეგის მიღების სტადია, დაკავშირებული შედეგობრივ ელემენტთან და შეჯამების ან დაგროვების ფუნქციასთან
- ბ. ორ სტადიას: ა) დაჯგუფების სტადია, დაკავშირებული დამაჯგუფებელ ან სტრიქონულ ელემენტთან; ბ) გახსნის სტადია, დაკავშირებული გაშლასთან ან სტრიქონულ ელემენტთან
- გ. ერთ სტადიას - დაჯგუფების სტადია, დაკავშირებული დამაჯგუფებელ ან სტრიქონულ ელემენტთან

17.1.6. მონაცემების გაშლა:

- ა. ერთი გზით შეიძლება შევასრულოთ - სტანდარტული SQL ენის საშუალებების გამოყენებით
- ბ. ორი გზით შეიძლება შევასრულოთ: სტანდარტული SQL ენის საშუალებებისა და T-SQL ენის PIVOT ოპერაციის გამოყენებით
- გ. ერთი გზით შეიძლება შევასრულოთ - T-SQL ენის PIVOT ოპერაციის გამოყენებით

17.1.7. PIVOT ოპერაცია გამოიყენება SELECT მოთხოვნის:

- ა. WHERE განყოფილებაში

- ბ. HAVING განყოფილებაში
- გ. FROM განყოფილებაში

17.1.8. PIVOT ოპერაცია ამუშავებს:

- ა. საწყის ცხრილს ან ცხრილურ გამოსახულებას, გამოთვლის ჯამურ მნიშვნელობას და გასცემს შედეგობრივ ცხრილს
- ბ. საწყის ცხრილს ან ცხრილურ გამოსახულებას და გამოთვლის ჯამურ მნიშვნელობას
- გ. საწყის ცხრილს ან ცხრილურ გამოსახულებას და გასცემს შედეგობრივ ცხრილს

მონაცემების შეკვეცა. მონაცემების შეკვეცა სტანდარტული SQL ენის საშუალებით. შეკვეცა T-SQL-ის UNPIVOT ოპერაციის საშუალებით

17.2.1. მონაცემების შეკვეცისას შეჯამებული (გაშლილი) ცხრილის თითოეული საწყისი სტრიქონი:

- ა. შეიძლება სვეტების სიმრავლედ გარდაიქმნას - თითო სვეტი საწყისი სტრიქონის ყოველი მოცემული მნიშვნელობისთვის
- ბ. არ შეიძლება სტრიქონების სიმრავლედ გარდაიქმნას - თითო სტრიქონი საწყისი სვეტის ყოველი მოცემული მნიშვნელობისთვის
- გ. შეიძლება სტრიქონების სიმრავლედ გარდაიქმნას - თითო სტრიქონი საწყისი სვეტის ყოველი მოცემული მნიშვნელობისთვის

17.2.2. მონაცემების შეკვეცა:

- ა. ორი გზით შეიძლება შევასრულოთ: SQL ენის სტანდარტული საშუალებებისა და T-SQL ენის UNPIVOT ოპერაციის გამოყენებით
- ბ. ერთი გზით შეიძლება შევასრულოთ - SQL ენის სტანდარტული საშუალებებით
- გ. ერთი გზით შეიძლება შევასრულოთ - T-SQL ენის UNPIVOT ოპერაციის გამოყენებით

17.2.3. მონაცემების შეკვეცის სტანდარტული გადაწყვეტა ლოგიკური დამუშავების:

- ა. ორი სტადიის რეალიზებას მოიცავს: ასლის ფორმირება და ელემენტების ამოღება
- ბ. სამი სტადიის რეალიზებას მოიცავს: ასლის ფორმირება, ელემენტების ამოღება და არარსებული გადაკვეთების გამორიცხვა
- გ. ერთი სტადიის რეალიზებას მოიცავს - არარსებული გადაკვეთების გამორიცხვა

17.2.4. UNPIVOT ოპერაცია რეალიზებულია როგორც ცხრილური ოპერაცია და გამოიყენება:

- ა. WHERE განყოფილებაში
- ბ. GROUP BY განყოფილებაში
- გ. FROM განყოფილებაში

დაჯგუფების ნაკრებები. GROUPING SETS ჩადგმული ელემენტი. CUBE ჩადგმული ელემენტი. ROLLUP ჩადგმული ელემენტი. GROUPING() და GROUPING_ID() ფუნქციები

17.3.1. დაჯგუფების ნაკრები - ესაა:

- ა. სვეტების ერთობლიობა, რომელთა მიხედვით მონაცემების დაჯგუფება სრულდება
- ბ. სვეტების ერთობლიობა, რომელთა მიხედვით მონაცემების დაჯგუფება არ სრულდება

- გ. სტრიქონების ერთობლიობა, რომელთა მიხედვით მონაცემების დაჯგუფება სრულდება
- 17.3.2. ტრადიციულად SQL ენაში ერთი მოთხოვნა, განკუთვნილი შეჯამებული მონაცემების მისაღებად, დაჯგუფების:
- ა. ორ ნაკრებს განსაზღვრავს
 - ბ. ერთ ნაკრებს განსაზღვრავს
 - გ. სამ ნაკრებს განსაზღვრავს
- 17.3.3. GROUPING SETS ჩადგმული ელემენტის გამოყენებით:
- ა. შესაძლებელია ერთ მოთხოვნაში დაჯგუფების მხოლოდ ერთი ნაკრები განისაზღვროს
 - ბ. შეუძლებელია ერთ მოთხოვნაში დაჯგუფების რამდენიმე ნაკრები განისაზღვროს
 - გ. შესაძლებელია ერთ მოთხოვნაში დაჯგუფების რამდენიმე ნაკრები განისაზღვროს
- 17.3.4. GROUP BY განყოფილების CUBE ჩადგმული ელემენტი დაჯგუფების:
- ა. მხოლოდ ერთი ნაკრების განსაზღვრის შემოკლებულ საშუალებას წარმოადგენს
 - ბ. რამდენიმე ნაკრების განსაზღვრის შემოკლებულ საშუალებას წარმოადგენს
 - გ. რამდენიმე ნაკრების განსაზღვრის შემოკლებულ საშუალებას არ წარმოადგენს
- 17.3.5. ROLLUP ჩადგმული ელემენტი, რომელიც შედის GROUP BY განყოფილების შემადგენლობაში:
- ა. წარმოგვიდგენს დაჯგუფების რამდენიმე ნაკრების განსაზღვრის შემოკლებულ საშუალებას
 - ბ. არ წარმოგვიდგენს დაჯგუფების რამდენიმე ნაკრების განსაზღვრის შემოკლებულ საშუალებას
 - გ. წარმოგვიდგენს დაჯგუფების მხოლოდ ერთი ნაკრების განსაზღვრის შემოკლებულ საშუალებას
- 17.3.6. ROLLUP ელემენტი გულისხმობს იერარქიის:
- ა. არარსებობას შესასვლელ სვეტებში და დაჯგუფების იმ ნაკრებებს ქმნის, რომლებსაც აზრი იერარქიის გათვალისწინებით აქვს
 - ბ. არსებობას შესასვლელ სვეტებში და დაჯგუფების იმ ნაკრებებს ქმნის, რომლებსაც აზრი იერარქიის გათვალისწინებით არ აქვს
 - გ. არსებობას შესასვლელ სვეტებში და დაჯგუფების იმ ნაკრებებს ქმნის, რომლებსაც აზრი იერარქიის გათვალისწინებით აქვს
- 17.3.7. ერთ-ერთი საშუალება ცალსახად განვსაზღვროთ დაჯგუფების ნაკრებთან კავშირები, მაშინაც კი როცა დასაჯგუფებელ სვეტებში დაშვებულია NULL მნიშვნელობები, არის:
- ა. GETDATE() ფუნქციის გამოყენება
 - ბ. GROUPING() ფუნქციის გამოყენება
 - გ. YEAR() ფუნქციის გამოყენება
- 17.3.8. GROUPING() ფუნქცია იღებს სვეტის სახელს და გასცემს:
- ა. 1-ს, თუ ის არის მიმდინარე დაჯგუფების ნაკრების წევრი, და 0-ს წინააღმდეგ შემთხვევაში
 - ბ. 0-ს, თუ ის არის მიმდინარე დაჯგუფების ნაკრების წევრი, და 1-ს წინააღმდეგ შემთხვევაში
 - გ. -1-ს, თუ ის არის მიმდინარე დაჯგუფების ნაკრების წევრი, და 1-ს წინააღმდეგ შემთხვევაში

17.3.9. GROUPING_ID() ფუნქცია დაგვიბრუნებს ორობითი ასახვის მთელრიცხვა მნიშვნელობას, რომელშიც თითოეული:

ა. ბიტი ერთ-ერთ შესასვლელ ელემენტს წარმოადგენს. უკიდურესი მარჯვენა ბიტი წარმოადგენს უკიდურეს მარცხენა ელემენტს სიაში

ბ. ბიტი ერთ-ერთ შესასვლელ ელემენტს წარმოადგენს. უკიდურესი მარცხენა ბიტი წარმოადგენს უკიდურეს მარჯვენა ელემენტს სიაში

გ. ბიტი ერთ-ერთ შესასვლელ ელემენტს წარმოადგენს. უკიდურესი მარჯვენა ბიტი წარმოადგენს უკიდურეს მარჯვენა ელემენტს სიაში

თავი 18. მონაცემთა ბაზების სარეზერვო ასლები

სარეზერვო ასლის ტიპები

18.1.1. სარეზერვო ასლი წარმოადგენს:

- ა. ერთ ან მეტ ფაილს, რომელშიც მონაცემთა ბაზაა მოთავსებული მთლიანად ან ნაწილობრივ
- ბ. ორ ფაილს, რომელშიც მონაცემთა ბაზაა მოთავსებული მთლიანად
- გ. სამ ფაილს, რომელშიც მონაცემთა ბაზაა მოთავსებული ნაწილობრივ

18.1.2. სარეზერვო ასლის შექმნა საჭიროა:

- ა. მონაცემების წასაშლელად მონაცემების დაზიანების შემთხვევაში და ა.შ.
- ბ. მონაცემების აღსადგენად კომპიუტერის, დისკის ან მონაცემების დაზიანების შემთხვევაში და ა.შ.
- გ. მონაცემების წასაშლელად კომპიუტერისა და დისკის დაზიანების შემთხვევაში და ა.შ.

18.1.3. სარეზერვო ასლის შესანახად:

- ა. გამოიყენება მხოლოდ მაგნიტური ლენტები
- ბ. არ გამოიყენება ინფორმაციის მატარებლები, როგორცაა მაგნიტური ლენტები, მაგნიტური დისკები, ZIP დისკები და ა.შ.
- გ. გამოიყენება ინფორმაციის მატარებლები, როგორცაა მაგნიტური ლენტები, მაგნიტური დისკები, ZIP დისკები და ა.შ.

18.1.4. მონაცემთა ბაზების სარეზერვო ასლები:

- ა. უნდა შეიქმნას გარკვეული პერიოდულობით
- ბ. არ უნდა შეიქმნას გარკვეული პერიოდულობით
- გ. უნდა წაიშალოს გარკვეული პერიოდულობით

18.1.5. სარეზერვო ასლის შექმნა:

- ა. არ იკავებს ოპერაციული სისტემის რესურსებს
- ბ. იკავებს ოპერაციული სისტემის რესურსებს
- გ. ხანდახან იკავებს ოპერაციული სისტემის რესურსებს

18.1.6. არსებობს სარეზერვო ასლის:

- ა. ერთი ტიპი
- ბ. ორი ტიპი
- გ. ოთხი ტიპი

18.1.7. არსებობს სარეზერვო ასლის შემდეგი ტიპები:

- ა. მონაცემთა ბაზის სრული სარეზერვო ასლი, ტრანზაქციების ჟურნალის სარეზერვო ასლი, დიფერენცირებული სარეზერვო ასლი, ფაილებისა და ფაილების ჯგუფის სარეზერვო ასლი
- ბ. მონაცემთა ბაზის სრული სარეზერვო ასლი, ტრანზაქციების ჟურნალის სარეზერვო ასლი, დიფერენცირებული სარეზერვო ასლი
- გ. ტრანზაქციების ჟურნალის სარეზერვო ასლი, დიფერენცირებული სარეზერვო ასლი, ფაილებისა და ფაილების ჯგუფის სარეზერვო ასლი

18.1.8. ნებისმიერი ტიპის სარეზერვო ასლი:

- ა. ყოველთვის არ იქმნება ერთი მონაცემთა ბაზისთვის და წარმოადგენს ერთ ფაილს
- ბ. ყოველთვის იქმნება ერთი მონაცემთა ბაზისთვის და წარმოადგენს ერთ ფაილს
- გ. ყოველთვის იქმნება ერთი მონაცემთა ბაზისთვის და არ წარმოადგენს ერთ ფაილს

18.1.9. სარეზერვო ასლის ფაილი:

- ა. არის ოპერაციული სისტემის ჩვეულებრივი ფაილი, რომელიც არ შეგვიძლია სხვა დისკზე ან კატალოგში გადავწეროთ, ან, გადავიტანოთ სხვა სერვერზე და იქ აღვადგინოთ
- ბ. არ არის ოპერაციული სისტემის ჩვეულებრივი ფაილი, რომელიც შეგვიძლია სხვა დისკზე ან კატალოგში გადავწეროთ, აგრეთვე, გადავიტანოთ სხვა სერვერზე და იქ აღვადგინოთ
- გ. არის ოპერაციული სისტემის ჩვეულებრივი ფაილი, რომელიც შეგვიძლია სხვა დისკზე ან კატალოგში გადავწეროთ, აგრეთვე, გადავიტანოთ სხვა სერვერზე და იქ აღვადგინოთ

18.1.10.სარეზერვო ასლი:

- ა. შეიძლება აღვადგინოთ იმავე ან სხვა სახელით
- ბ. არ შეიძლება აღვადგინოთ იმავე ან სხვა სახელით
- გ. შეიძლება აღვადგინოთ მხოლოდ იმავე სახელით

18.1.11.მონაცემთა ბაზის სარეზერვო ასლის შექმნის დროს სერვერი:

- ა. ხანდახან ასრულებს მის არქივირებას
- ბ. ასრულებს მის არქივირებას
- გ. არ ასრულებს მის არქივირებას

18.1.12.მიღებული სარეზერვო ასლი:

- ა. არ შეიძლება დამატებით შევკუმშოთ რომელიმე არქივატორის გამოყენებით
- ბ. არ შეგვიძლია დამატებით შევკუმშოთ რომელიმე არქივატორის გამოყენებით
- გ. შეგვიძლია დამატებით შევკუმშოთ რომელიმე არქივატორის გამოყენებით

18.1.13.რაც უფრო დიდია მონაცემთა ბაზის ზომა, მით:

- ა. მეტ დროს დაიკავებს მისი სარეზერვო ასლის შექმნის პროცესი
- ბ. ნაკლებ დროს დაიკავებს მისი სარეზერვო ასლის შექმნის პროცესი
- გ. მეტ დროს არ დაიკავებს მისი სარეზერვო ასლის შექმნის პროცესი

18.1.14.რაც უფრო ინტენსიურად სრულდება ცვლილებები მონაცემთა ბაზაში, მით უფრო:

- ა. იშვიათად უნდა შევასრულოთ სარეზერვო ასლის შექმნა
- ბ. ხშირად უნდა შევასრულოთ სარეზერვო ასლის შექმნა
- გ. ხშირად არ უნდა შევასრულოთ სარეზერვო ასლის შექმნა

18.1.15.თუ მონაცემთა ბაზა მისაწვდომია მხოლოდ წაკითხვის რეჟიმში, მაშინ:

- ა. აუცილებელია ორი სარეზერვო ასლის შექმნა
- ბ. საკმარისი არაა ერთი სარეზერვო ასლის შექმნა
- გ. საკმარისია ერთი სარეზერვო ასლის შექმნა

დამგროვებლის არჩევა. მოწყობილობის განსაზღვრა სარეზერვო ასლის

მოსათავსებლად

- 18.2.1. სარეზერვო ასლის შექმნისათვის სერვერზე შექმნილი ლოგიკური მოწყობილობების სია ინახება:
- ა. master მონაცემთა ბაზის sysdevices სისტემურ წარმოდგენაში
 - ბ. tempdb მონაცემთა ბაზის sysdevices სისტემურ წარმოდგენაში
 - გ. model მონაცემთა ბაზის sysdevices სისტემურ წარმოდგენაში
- 18.2.2. ახალი მოწყობილობის დასამატებლად გამოიყენება:
- ა. sp_add სისტემური შენახული პროცედურა
 - ბ. sp_addumpdevice სისტემური შენახული პროცედურა
 - გ. sp_help სისტემური შენახული პროცედურა
- 18.2.3. sp_addumpdevice სისტემური შენახული პროცედურის სინტაქსია:
- ა. sp_addumpdevice [@devtype =] 'მოწყობილობის_ტიპი' ,
[@physicalname =] 'ფიზიკური_სახელი'
 - ბ. sp_addumpdevice [@devtype =] 'მოწყობილობის_ტიპი'
 - გ. sp_addumpdevice [@devtype =] 'მოწყობილობის_ტიპი' ,
[@logicalname =] 'ლოგიკური_სახელი' , [@physicalname =] 'ფიზიკური_სახელი'
- 18.2.4. sp_addumpdevice სისტემური პროცედურის 'მოწყობილობის_ტიპი' პარამეტრი:
- ა. მოწყობილობის ტიპია
 - ბ. მოწყობილობის ლოგიკური სახელია
 - გ. მოწყობილობის ფიზიკური სახელია
- 18.2.5. sp_addumpdevice სისტემური პროცედურის 'ლოგიკური_სახელი' პარამეტრი:
- ა. მოწყობილობის ტიპია
 - ბ. მოწყობილობის ლოგიკური სახელია
 - გ. მოწყობილობის ფიზიკური სახელია
- 18.2.6. sp_addumpdevice სისტემური პროცედურის 'ფიზიკური_სახელი' პარამეტრი:
- ა. მოწყობილობის ტიპია
 - ბ. მოწყობილობის ლოგიკური სახელია
 - გ. მოწყობილობის ფიზიკური სახელია
- 18.2.7. sp_addumpdevice სისტემური პროცედურის 'მოწყობილობის_ტიპი' პარამეტრი იღებს შემდეგ მნიშვნელობებს:
- ა. მხოლოდ DISK (დისკი)
 - ბ. TAPE (ლენტა), DISK (დისკი) და PIPE (სახელდებული არხი)
 - გ. მხოლოდ TAPE (ლენტა)
- 18.2.8. ლოგიკური მოწყობილობის წასაშლელად გამოიყენება:
- ა. sp_device სისტემური შენახული პროცედურა
 - ბ. sp_drop სისტემური შენახული პროცედურა
 - გ. sp_dropdevice სისტემური შენახული პროცედურა
- 18.2.9. sp_dropdevice სისტემური პროცედურის სინტაქსია:
- ა. sp_dropdevice [@logicalname =] 'მოწყობილობის_ლოგიკური_სახელი'

[, [@delfile =] 'DELFILE']

ბ. sp_dropdevice 'DELFILE'

გ. sp_dropdevice [@logicalname =] [, [@delfile =] 'DELFILE']

18.2.10.EXEC sp_addumpdevice 'DISK', 'ChemiDiski',

'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Backup\Asli_1.bak'; ბრძანების შესრულების შედეგად:

ა. სარეზერვო ასლის მოსათავსებლად არ შეიქმნება ChemiDiski სახელის მქონე დისკური მოწყობილობა, რომლის ფიზიკური სახელია C:\dump\Asli_1.bak

ბ. სარეზერვო ასლის მოსათავსებლად შეიქმნება ChemiDiski სახელის მქონე დისკური მოწყობილობა, რომლის ფიზიკური სახელია C:\dump\Asli_1.bak

გ. სარეზერვო ასლის მოსათავსებლად შეიქმნება C:\dump\Asli_1.bak სახელის მქონე დისკური მოწყობილობა, რომლის ფიზიკური სახელია ChemiDiski

18.2.11.sp_dropdevice 'ChemiDiski'; ბრძანების შესრულების შედეგად:

ა. გაიცემა ინფორმაცია სარეზერვო ასლის შენახვის მიზნით შექმნილი 'ChemiDiski' დისკური მოწყობილობის შესახებ

ბ. არ წაიშლება სარეზერვო ასლის შენახვის მიზნით შექმნილი დისკური მოწყობილობა 'ChemiDiski'

გ. წაიშლება სარეზერვო ასლის შენახვის მიზნით შექმნილი დისკური მოწყობილობა 'ChemiDiski'

18.2.12.sp_helpdevice 'ChemiDiski'; ბრძანების შესრულების შედეგად:

ა. გაიცემა ინფორმაცია სარეზერვო ასლის შენახვის მიზნით შექმნილი 'ChemiDiski' დისკური მოწყობილობის შესახებ

ბ. არ გაიცემა ინფორმაცია სარეზერვო ასლის შენახვის მიზნით შექმნილი 'ChemiDiski' დისკური მოწყობილობის შესახებ

გ. წაიშლება სარეზერვო ასლის შენახვის მიზნით შექმნილი დისკური მოწყობილობა 'ChemiDiski'

მონაცემთა ბაზის სრული და დიფერენცირებული ასლები

18.3.1. მონაცემთა ბაზის სრული სარეზერვო ასლის შექმნა:

ა. გულისხმობს მთელი მონაცემთა ბაზის ასლის შექმნას

ბ. არ გულისხმობს მთელი მონაცემთა ბაზის ასლის შექმნას

გ. ხანდახან გულისხმობს მთელი მონაცემთა ბაზის ასლის შექმნას

18.3.2. მონაცემთა ბაზის სრული სარეზერვო ასლის დადებითი მხარეა ის, რომ:

ა. არაა საკმარისი მხოლოდ ერთი არქივის აღდგენა, რადგან მთელი ინფორმაცია ერთ ფაილშია მოთავსებული

ბ. საკმარისია მხოლოდ ერთი არქივის აღდგენა, რადგან მთელი ინფორმაცია ერთ ფაილშია მოთავსებული

გ. საკმარისია მხოლოდ ერთი არქივის აღდგენა, რადგან მთელი ინფორმაცია ორ ფაილშია მოთავსებული

18.3.3. სრული სარეზერვო ასლი უნდა შექმნათ მაშინ, როცა მონაცემთა ბაზა:

- ა. იშვიათად იცვლება და მონაცემების დაკარგვას დიდი მნიშვნელობა ექნება
- ბ. ხშირად იცვლება და მონაცემების დაკარგვას დიდი მნიშვნელობა არ ექნება
- გ. იშვიათად იცვლება და მონაცემების დაკარგვას დიდი მნიშვნელობა არ ექნება

18.3.4. მონაცემთა ბაზის სრული სარეზერვო ასლის:

- ა. ნაკლია არქივის შექმნის ხანგრძლივი დრო, მასში მცირეოდენი ცვლილებების შეტანის შემთხვევაშიც კი
- ბ. დადებითი მხარეა არქივის შექმნის ხანგრძლივი დრო, მასში მცირეოდენი ცვლილებების შეტანის შემთხვევაშიც კი
- გ. ნაკლია არქივის შექმნის მცირე დრო, მასში მცირეოდენი ცვლილებების შეტანის შემთხვევაშიც კი

18.3.5. სრული სარეზერვო ასლი:

- ა. შეიცავს მონაცემთა ბაზის ყველა მონაცემს სარეზერვო ასლის შექმნის ოპერაციის დამთავრების მომენტისათვის
- ბ. არ შეიცავს მონაცემთა ბაზის ყველა მონაცემს სარეზერვო ასლის შექმნის ოპერაციის დამთავრების მომენტისათვის
- გ. შეიცავს მონაცემთა ბაზის ყველა მონაცემს სარეზერვო ასლის შექმნის ოპერაციის დაწყების მომენტისათვის

18.3.6. სრული სარეზერვო ასლი:

- ა. არ არის საბაზო, რადგან ის გამოიყენება დიფერენცირებული და ტრანზაქციების ჟურნალის სარეზერვო ასლის შექმნის
- ბ. არის საბაზო, რადგან ის გამოიყენება დიფერენცირებული და ტრანზაქციების ჟურნალის სარეზერვო ასლის შექმნის
- გ. არის საბაზო, რადგან ის არ გამოიყენება დიფერენცირებული და ტრანზაქციების ჟურნალის სარეზერვო ასლის შექმნის

18.3.7. სრული და დიფერენცირებული სარეზერვო ასლის შექმნის დროს:

- ა. არ ხდება ტრანზაქციების ჟურნალის ჩამოჭრა (ზომის შემცირება) და დროთა განმავლობაში ამან არ შეიძლება გამოიწვიოს მისი გადავსება
- ბ. ხდება ტრანზაქციების ჟურნალის ჩამოჭრა (ზომის შემცირება) და დროთა განმავლობაში ამან შეიძლება გამოიწვიოს მისი გადავსება
- გ. არ ხდება ტრანზაქციების ჟურნალის ჩამოჭრა (ზომის შემცირება) და დროთა განმავლობაში ამან შეიძლება გამოიწვიოს მისი გადავსება

18.3.8. მონაცემთა ბაზის დიფერენცირებული სარეზერვო ასლი:

- ა. შეიცავს ინფორმაციას იმ ცვლილებების შესახებ, რომლებიც შესრულდა მონაცემთა ბაზაში მისი სრული სარეზერვო ასლის უკანასკნელი შექმნის მომენტიდან
- ბ. არ შეიცავს ინფორმაციას იმ ცვლილებების შესახებ, რომლებიც შესრულდა მონაცემთა ბაზაში მისი სრული სარეზერვო ასლის უკანასკნელი შექმნის მომენტიდან
- გ. შეიცავს ინფორმაციას იმ ცვლილებების შესახებ, რომლებიც შესრულდა მონაცემთა ბაზაში მისი სრული სარეზერვო ასლის პირველი შექმნის მომენტიდან

18.3.9. დიფერენცირებული ასლი იკავებს:

- ა. დიდ ადგილს და მოითხოვს ნაკლებ დროს შექმნისათვის

- ბ. ნაკლებ ადგილს და მოითხოვს ნაკლებ დროს შექმნისათვის
- გ. ნაკლებ ადგილს და მოითხოვს დიდ დროს შექმნისათვის

18.3.10. დიფერენცირებული ასლი შესაძლებლობას:

- ა. გვადლევს პერიოდულად შევქმნათ დიდი ზომის მონაცემთა ბაზების სარეზერვო ასლი გაცილებით დიდი დროის განმავლობაში
- ბ. არ გვადლევს პერიოდულად შევქმნათ დიდი ზომის მონაცემთა ბაზების სარეზერვო ასლი შედარებით მცირე დროის განმავლობაში
- გ. გვადლევს პერიოდულად შევქმნათ დიდი ზომის მონაცემთა ბაზების სარეზერვო ასლი შედარებით მცირე დროის განმავლობაში

18.3.11. დიფერენცირებული სარეზერვო ასლის შექმნის:

- ა. წინ უნდა შევქმნათ მონაცემთა ბაზის სრული სარეზერვო ასლი
- ბ. შემდეგ უნდა შევქმნათ მონაცემთა ბაზის სრული სარეზერვო ასლი
- გ. წინ არ უნდა შევქმნათ მონაცემთა ბაზის სრული სარეზერვო ასლი

18.3.12. დიდი ზომის მონაცემთა ბაზებში, რომლებშიც ცვლილებების რაოდენობა:

- ა. დიდია, დიფერენცირებული სარეზერვო ასლის შექმნა ყველაზე ოპტიმალური მეთოდია
- ბ. მცირეა, დიფერენცირებული სარეზერვო ასლის შექმნა ყველაზე ოპტიმალური მეთოდია
- გ. მცირეა, დიფერენცირებული სარეზერვო ასლის შექმნა არაოპტიმალური მეთოდია

სრული და დიფერენცირებული სარეზერვო ასლების შექმნა

18.4.1. სრული და დიფერენცირებული სარეზერვო ასლების შესაქმნელად გამოიყენება:

- ა. BACKUP BASE ბრძანება
- ბ. RESTORE DATABASE ბრძანება
- გ. BACKUP DATABASE ბრძანება

18.4.2. BACKUP DATABASE ბრძანების სინტაქსია:

- ა. BACKUP DATABASE მონაცემთა_ბაზის_სახელი TO <ინფორმაციის_მატარებელი>
- ბ. BACKUP DATABASE მონაცემთა_ბაზის_სახელი
- გ. BACKUP DATABASE ინფორმაციის_მატარებელი TO <მონაცემთა_ბაზის_სახელი>

18.4.3. BACKUP DATABASE ბრძანების DIFFERENTIAL პარამეტრი:

- ა. მიუთითებს რომ არ უნდა შეიქმნას მონაცემთა ბაზის დიფერენცირებული ასლი
- ბ. მიუთითებს რომ უნდა შეიქმნას მონაცემთა ბაზის დიფერენცირებული ასლი
- გ. მიუთითებს რომ უნდა შეიქმნას მონაცემთა ბაზის სრული ასლი

18.4.4. BACKUP DATABASE ბრძანების DIFFERENTIAL პარამეტრი თუ არ არის მითითებული, მაშინ:

- ა. შეიქმნება მონაცემთა ბაზის დიფერენცირებული ასლი
- ბ. არ შეიქმნება მონაცემთა ბაზის სრული ასლი
- გ. შეიქმნება მონაცემთა ბაზის სრული ასლი

18.4.5. BACKUP DATABASE ბრძანების NAME პარამეტრი:

- ა. არქივის სახელია
- ბ. არქივის პაროლია
- გ. არქივის ზომაა

18.4.6. BACKUP DATABASE ბრძანების NAME პარამეტრი:

- ა. არ არის მითითებული, მაშინ არქივს ცარიელი სახელი არ მიენიჭება
- ბ. არ არის მითითებული, მაშინ არქივს ცარიელი სახელი მიენიჭება
- გ. არ არის მითითებული, მაშინ არქივს არაცარიელი სახელი მიენიჭება

18.4.7. BACKUP DATABASE Shekveta TO Shekveta_Akli WITH NAME = 'Shekvata_Akli'; ბრძანების შესრულების შედეგად:

- ა. არ შეიქმნება Shekveta მონაცემთა ბაზის სრული ასლი
- ბ. შეიქმნება Shekveta მონაცემთა ბაზის სრული ასლი
- გ. შეიქმნება Shekveta მონაცემთა ბაზის დიფერენცირებული ასლი

18.4.8. BACKUP DATABASE Shekveta TO Shekveta_Akli

WITH DIFFERENTIAL, NAME = 'Shekvata_Akli_Dif'; ბრძანების შესრულების შედეგად:

- ა. შეიქმნება Shekveta მონაცემთა ბაზის სრული ასლი
- ბ. არ შეიქმნება Shekveta მონაცემთა ბაზის დიფერენცირებული ასლი
- გ. შეიქმნება Shekveta მონაცემთა ბაზის დიფერენცირებული ასლი

სრული და დიფერენცირებული სარეზერვო ასლიდან აღდგენა

18.5.1. სრული და დიფერენცირებული სარეზერვო ასლიდან აღსადგენად:

- ა. გამოიყენება BACKUP DATABASE ბრძანება
- ბ. არ გამოიყენება RESTORE DATABASE ბრძანება
- გ. გამოიყენება RESTORE DATABASE ბრძანება

18.5.2. RESTORE DATABASE ბრძანების სინტაქსია:

- ა. RESTORE DATABASE მონაცემთა_ბაზის_სახელი
- ბ. BACKUP DATABASE მონაცემთა_ბაზის_სახელი
- გ. RESTORE DATABASE ცხრილის_ბაზის_სახელი

18.5.3. RESTORE DATABASE ბრძანების FROM პარამეტრი:

- ა. იმ მოწყობილობის სახელია, საიდანაც არ უნდა შესულდეს მონაცემთა ბაზის აღდგენა
- ბ. იმ მოწყობილობის სახელია, საიდანაც უნდა შესულდეს მონაცემთა ბაზის აღდგენა
- გ. იმ მოწყობილობის სახელია, რომელშიც უნდა შესულდეს მონაცემთა ბაზის ჩაწერა

18.5.4. RESTORE DATABASE ბრძანების WITH REPLACE პარამეტრი თუ მითითებულია და:

- ა. სარეზერვო ასლიდან აღდგენის დროს სერვერზე არ არსებობს მონაცემთა ბაზა ამავე სახელით, მაშინ შესრულდება მასზე გადაწერა
- ბ. სარეზერვო ასლიდან აღდგენის დროს სერვერზე არსებობს მონაცემთა ბაზა ამავე სახელით, მაშინ არ შესრულდება მასზე გადაწერა
- გ. სარეზერვო ასლიდან აღდგენის დროს სერვერზე არსებობს მონაცემთა ბაზა ამავე სახელით, მაშინ შესრულდება მასზე გადაწერა

18.5.5. RESTORE DATABASE Shekveta FROM Shekveta_Aсли; ბრძანების შესრულების შედეგად:

- ა. შესრულდება აღდგენა სრული სარეზერვო ასლიდან. Shekveta მონაცემთა ბაზა არ არსებობს და ის შეიქმნება
- ბ. შესრულდება აღდგენა სრული სარეზერვო ასლიდან. Shekveta მონაცემთა ბაზა არსებობს
- გ. შესრულდება აღდგენა სრული სარეზერვო ასლიდან. Shekveta მონაცემთა ბაზა არ არსებობს და ის არ შეიქმნება

18.5.6. RESTORE DATABASE Shekveta FROM Shekveta_Aсли WITH REPLACE; ბრძანების შესრულების შედეგად:

- ა. არ შესრულდება აღდგენა სრული სარეზერვო ასლიდან. Shekveta მონაცემთა ბაზა უკვე არსებობს და ხდება მასზე გადაწერა ასლიდან
- ბ. შესრულდება აღდგენა სრული სარეზერვო ასლიდან. Shekveta მონაცემთა ბაზა უკვე არსებობს და ხდება მასზე გადაწერა ასლიდან
- გ. შესრულდება აღდგენა სრული სარეზერვო ასლიდან. Shekveta მონაცემთა ბაზა არ არსებობს

თავი 19. უსაფრთხოების სისტემა

უფლებამოსილების გამიჯვნის წესები

- 19.1.1. ხშირად, მონაცემთა ბაზასთან რამდენიმე მომხმარებელი მუშაობს. ამ დროს:
- ა. აუცილებელია მასთან მიმართვის უფლებების გამიჯვნა
 - ბ. აუცილებელი არაა მასთან მიმართვის უფლებების გამიჯვნა
 - გ. აუცილებელია მასთან მიმართვის უფლებების უარყოფა
- 19.1.2. ხშირად, მონაცემთა ბაზასთან რამდენიმე მომხმარებელი მუშაობს. ეს:
- ა. მოითხოვს უსაფრთხოების სისტემის არასწორად დაგეგმვას
 - ბ. მოითხოვს უსაფრთხოების სისტემის სწორად დაგეგმვას
 - გ. არ მოითხოვს უსაფრთხოების სისტემის სწორად დაგეგმვას
- 19.1.3. უსაფრთხოების სისტემის დაგეგმვის პროცესში:
- ა. არ უნდა განისაზღვროს ის მოქმედებები, რომელთა შესრულების უფლება ექნებათ მათ
 - ბ. არ უნდა განისაზღვროს ის მომხმარებლები, რომლებსაც ექნებათ მონაცემთა ბაზასთან მიმართვის უფლება
 - გ. უნდა განისაზღვროს ის მომხმარებლები, რომლებსაც ექნებათ მონაცემთა ბაზასთან მიმართვის უფლება და ის მოქმედებები, რომელთა შესრულების უფლება ექნებათ მათ
- 19.1.4. უსაფრთხოების სისტემის პირველი ეტაპი მდგომარეობს იმ მომხმარებლების განსაზღვრაში, რომლებსაც:
- ა. ექნებათ მონაცემთა ბაზასთან მიმართვის უფლება
 - ბ. არ ექნებათ მონაცემთა ბაზასთან მიმართვის უფლება
 - გ. ექნებათ და ამავე დროს არ ექნებათ მონაცემთა ბაზასთან მიმართვის უფლება
- 19.1.5. უსაფრთხოების სისტემის პირველი ეტაპი მდგომარეობს იმ მომხმარებლების განსაზღვრაში, რომლებსაც ექნებათ მონაცემთა ბაზასთან მიმართვის უფლება. მაგრამ, ჯერ მომხმარებლებს:
- ა. არ უნდა ჰქონდეთ სერვერთან მიმართვის უფლება
 - ბ. უნდა ჰქონდეთ სერვერთან მიმართვის უფლება
 - გ. ერთდროულად უნდა ჰქონდეთ სერვერთან მიმართვის და არმიმართვის უფლება
- 19.1.6. სერვერთან მიმართვისათვის გამოიყენება:
- ა. მხოლოდ Windows NT-ის უსაფრთხოების სისტემა
 - ბ. მხოლოდ სერვერის უსაფრთხოების სისტემა
 - გ. სერვერის ან Windows NT-ის უსაფრთხოების სისტემა
- 19.1.7. თუ სერვერთან მიმართვისათვის გამოიყენება სერვერის უსაფრთხოების სისტემა, მაშინ მომხმარებელი:
- ა. უნდა დავარეგისტრირდეს სერვერზე, ანუ მისთვის უნდა შევქმნათ სააღრიცხვო ჩანაწერი
 - ბ. არ უნდა დავარეგისტრირდეს სერვერზე, ანუ მისთვის უნდა შევქმნათ სააღრიცხვო ჩანაწერი
 - გ. უნდა დავარეგისტრირდეს Windows NT სისტემაში

19.1.8. თუ სერვერთან მიმართვისათვის გამოიყენება Windows NT-ის უსაფრთხოების სისტემა, მაშინ მომხმარებელი:

- ა. უნდა დავარეგისტრიროთ სერვერზე, ანუ მისთვის უნდა შევქმნათ სააღრიცხვო ჩანაწერი
- ბ. უნდა დავარეგისტრიროთ Windows NT სისტემაში, ანუ მისთვის შევქმნათ სააღრიცხვო ჩანაწერი დომენში
- გ. უნდა დავარეგისტრიროთ Windows NT სისტემაში, ანუ მისთვის არ უნდა შევქმნათ სააღრიცხვო ჩანაწერი დომენში

19.1.9. თუ მომხმარებელს აქვს სერვერთან მიმართვის უფლება, ეს:

- ა. ნიშნავს იმას, რომ მას არ აქვს არც ერთ ბაზასთან მიმართვის უფლება
- ბ. ნიშნავს იმას, რომ მას აქვს ყველა ბაზასთან მიმართვის უფლება
- გ. არ ნიშნავს იმას, რომ მას აქვს რომელიმე მონაცემთა ბაზასთან ან მის ობიექტთან მიმართვის უფლება

19.1.10. უსაფრთხოების სისტემის დაგეგმვის მეორე ეტაპი მდგომარეობს იმ მოქმედებების განსაზღვრაში, რომელთა შესრულების უფლებაც:

- ა. აქვს კონკრეტულ მომხმარებელს მონაცემთა ბაზაში
- ბ. არ აქვს კონკრეტულ მომხმარებელს მონაცემთა ბაზაში
- გ. ერთდროულად აქვს და არ აქვს მომხმარებელს მონაცემთა ბაზაში

19.1.11. მონაცემთა ბაზასთან და მის ობიექტებთან სრული მიმართვის უფლება:

- ა. არ აქვს ადმინისტრატორს
- ბ. აქვს ადმინისტრატორს
- გ. აქვს რიგით მომხმარებელს

19.1.12. ადმინისტრატორის შემდეგ მეორე ადამიანი არის ობიექტის მფლობელი:

- ა. მეორე ადამიანი არის ადმინისტრატორი
- ბ. პირველი ადამიანი არის ობიექტის მფლობელი
- გ. მეორე ადამიანი არის ობიექტის მფლობელი

19.1.13. მონაცემთა ბაზაში ობიექტს შექმნის დროს:

- ა. ენიშნება მფლობელი, რომელსაც შეუძლია განსაზღვროს ამ ობიექტთან მიმართვის უფლებები
- ბ. არ ენიშნება მფლობელი, რომელსაც შეუძლია განსაზღვროს ამ ობიექტთან მიმართვის უფლებები
- გ. ენიშნება მფლობელი, რომელსაც არ შეუძლია განსაზღვროს ამ ობიექტთან მიმართვის უფლებები

19.1.14. მომხმარებლების მესამე კატეგორიას აქვს მიმართვის ის უფლებები, რომლებიც მათ:

- ა. არ მისცა ადმინისტრატორმა ან ობიექტის მფლობელმა
- ბ. მისცა ადმინისტრატორმა ან ობიექტის მფლობელმა
- გ. მისცა მომხმარებელმა, რომელიც არ არის არც ადმინისტრატორი და არც ობიექტის მფლობელი

19.1.15. სწორად დაგეგმილმა უსაფრთხოების სისტემამ მომხმარებელს:

- ა. არ უნდა მისცეს უფლება შეასრულოს ის მოქმედებები, რომლებიც არ სცილდება მის უფლებამოსილებას
- ბ. უნდა მისცეს უფლება შეასრულოს ის მოქმედებები, რომლებიც სცილდება მის უფლებამოსილებას
- გ. არ უნდა მისცეს უფლება შეასრულოს ის მოქმედებები, რომლებიც სცილდება მის უფლებამოსილებას

19.1.16. პაროლი:

- ა. სასურველია, რომ სიმბოლოების გარდა ციფრებსაც შეიცავდეს
- ბ. არაა სასურველი, რომ სიმბოლოების გარდა ციფრებსაც შეიცავდეს
- გ. სასურველია, რომ სიმბოლოებს არ შეიცავდეს

19.1.17. პაროლი:

- ა. უნდა იყოს ოჯახის წევრის სახელი, დაბადების წელი
- ბ. არ უნდა იყოს ოჯახის წევრის სახელი, დაბადების წელი, პასპორტის ნომერი და ა.შ.
- გ. უნდა იყოს პასპორტის ნომერი და ა.შ.

19.1.18. პაროლი:

- ა. უნდა ჰქონდეს მოქმედების შეზღუდული ვადა, რომლის გასვლის შემდეგ მომხმარებელმა ის არ უნდა შეცვალოს
- ბ. არ უნდა ჰქონდეს მოქმედების შეზღუდული ვადა, რომლის გასვლის შემდეგ მომხმარებელმა ის უნდა შეცვალოს
- გ. უნდა ჰქონდეს მოქმედების შეზღუდული ვადა, რომლის გასვლის შემდეგ მომხმარებელმა ის უნდა შეცვალოს

19.1.19. მაქსიმალური უსაფრთხოების მისაღწევად:

- ა. უნდა გამოვიყენოთ მომხმარებლების აუტენტიფიცირება Windows NT-ის საშუალებებით
- ბ. არ უნდა გამოვიყენოთ მომხმარებლების აუტენტიფიცირება Windows NT-ის საშუალებებით
- გ. უნდა გამოვიყენოთ მომხმარებლების აუტენტიფიცირება სერვერის საშუალებებით

სერვერის უსაფრთხოების სისტემის არქიტექტურა. აუტენტიფიცირების რეჟიმები. Windows NT-ის აუტენტიფიცირების რეჟიმი. სერვერის აუტენტიფიცირების რეჟიმი

19.2.1. სერვერის უსაფრთხოების სისტემა ემყარება:

- ა. მომხმარებლებსა და საადრიცხვო ჩანაწერებს
- ბ. მხოლოდ მომხმარებლებს
- გ. მხოლოდ საადრიცხვო ჩანაწერებს

19.2.2. მომხმარებლები გადიან შემოწმების:

- ა. ერთ ეტაპს
- ბ. ორ ეტაპს
- გ. სამ ეტაპს

19.2.3. პირველ ეტაპზე ხდება მომხმარებლის იდენტიფიცირება:

- ა. არ ხდება
- ბ. მხოლოდ პაროლის მიხედვით
- გ. სააღრიცხვო ჩანაწერის სახელისა და პაროლის მიხედვით, ე.ი. აუტენტიფიცირება

19.2.4. თუ სახელი და პაროლი სწორადაა შეტანილი, მაშინ მომხმარებელი:

- ა. მიუერთდება სერვერს
- ბ. არ მიუერთდება სერვერს
- გ. ხან მიუერთდება, ხან არა სერვერს

19.2.5. სერვერთან მიერთება ანუ დარეგისტრირება მომხმარებელს:

- ა. აძლევს მონაცემთა ბაზასთან ავტომატურად მიმართვის უფლებას
- ბ. არ აძლევს მონაცემთა ბაზასთან ავტომატურად მიმართვის უფლებას
- გ. არ აძლევს მონაცემთა ბაზასთან არა ავტომატურად მიმართვის უფლებას

19.2.6. თითოეული მონაცემთა ბაზისთვის სარეგისტრაციო სახელი (სააღრიცხვო ჩანაწერი, login):

- ა. უნდა აისახოს მონაცემთა ბაზის მომხმარებლის პაროლში (user)
- ბ. არ უნდა აისახოს მონაცემთა ბაზის მომხმარებლის სახელში (user)
- გ. უნდა აისახოს მონაცემთა ბაზის მომხმარებლის სახელში (user)

19.2.7. მეორე ეტაპზე, მომხმარებლისთვის გაცემული უფლებების საფუძველზე, მისი:

- ა. სარეგისტრაციო სახელი შესაბამის მონაცემთა ბაზასთან მიმართვის უფლებას იღებს
- ბ. სარეგისტრაციო სახელი შესაბამის მონაცემთა ბაზასთან მიმართვის უფლებას არ იღებს
- გ. სარეგისტრაციო პაროლი შესაბამის მონაცემთა ბაზასთან მიმართვის უფლებას იღებს

19.2.8. სხვადასხვა მონაცემთა ბაზაში ერთი და იგივე მომხმარებლის სარეგისტრაციო სახელს:

- ა. არ შეიძლება ჰქონდეს ერთნაირი ან სხვადასხვა სახელები მიმართვის სხვადასხვა უფლებებით
- ბ. შეიძლება ჰქონდეს ერთნაირი ან სხვადასხვა სახელები მიმართვის სხვადასხვა უფლებებით
- გ. შეიძლება ჰქონდეს მხოლოდ ერთნაირი მიმართვის ერთნაირი უფლებებით

19.2.9. მონაცემთა ბაზებთან მიმართვისათვის პროგრამა-დანართებს:

- ა. უნდა ჰქონდეთ არანაირი უფლებები
- ბ. არ უნდა ჰქონდეთ შესაბამისი უფლებები
- გ. უნდა ჰქონდეთ შესაბამისი უფლებები

19.2.10. ჩვეულებრივ, პროგრამა-დანართებს ისეთივე უფლებები:

- ა. ეძლევათ, როგორც აქვთ მათ გამშვებ მომხმარებლებს
- ბ. არ ეძლევათ, როგორც აქვთ მათ გამშვებ მომხმარებლებს
- გ. ეძლევათ, როგორც არ აქვთ მათ გამშვებ მომხმარებლებს

19.2.11. ზოგიერთი პროგრამა-დანართის მუშაობისთვის:

- ა. არაა საჭირო მიმართვის უფლებების ისეთი ნაკრები, რომელიც არ არის დამოკიდებული მომხმარებლის მიმართვის უფლებებზე

ბ. საჭიროა მიმართვის უფლებების ისეთი ნაკრები, რომელიც არ არის დამოკიდებული მომხმარებლის მიმართვის უფლებებზე

გ. საჭიროა მიმართვის უფლებების ისეთი ნაკრები, რომელიც დამოკიდებულია მომხმარებლის მიმართვის უფლებებზე

19.2.12.პროგრამა-დანართის მუშაობისთვის საჭირო უფლებების მინიჭება:

ა. შესაძლებელია როლების (ჯგუფების) გამოყენებლად

ბ. შეუძლებელია როლების (ჯგუფების) გამოყენებით

გ. შესაძლებელია როლების (ჯგუფების) გამოყენებით

19.2.13.მომხმარებლების აუტენტიფიცირების:

ა. ორი რეჟიმი არსებობს

ბ. სამი რეჟიმი არსებობს

გ. ერთი რეჟიმი არსებობს

19.2.14.მომხმარებლების აუტენტიფიცირების შემდეგი რეჟიმი არსებობს:

ა. აუტენტიფიცირების რეჟიმი მხოლოდ Windows NT-ის საშუალებებით (Windows NT Authentication)

ბ. აუტენტიფიცირების რეჟიმი Windows NT-ის საშუალებებით (Windows NT Authentication) და აუტენტიფიცირების შერეული რეჟიმი (Windows NT Authentication and SQL Server Authentication)

გ. აუტენტიფიცირების მხოლოდ შერეული რეჟიმი (Windows NT Authentication and SQL Server Authentication)

19.2.15.შერეული რეჟიმი საშუალებას:

ა. გვადლევს დავრეგისტრირდეთ მხოლოდ Windows NT-ის საშუალებებით

ბ. არ გვადლევს დავრეგისტრირდეთ როგორც სერვერის, ისე Windows NT-ის საშუალებებით

გ. გვადლევს დავრეგისტრირდეთ როგორც სერვერის, ისე Windows NT-ის საშუალებებით

19.2.16.როცა მომხმარებელი სერვერს უერთდება Windows NT-ის საადრიცხვო ჩანაწერის გამოყენებით, ამ დროს:

ა. მყარდება სანდო შეერთება

ბ. არ მყარდება სანდო შეერთება

გ. მყარდება არასანდო შეერთება

19.2.17.ოპერაციული სისტემის საადრიცხვო ჩანაწერები (logins):

ა. არ შეიცავენ ყველა მონაცემს მომხმარებლების შესახებ: სახელს, პაროლს, ჯგუფებში წევრობას და ა.შ.

ბ. შეიცავენ ყველა მონაცემს მომხმარებლების შესახებ: სახელს, პაროლს, ჯგუფებში წევრობას და ა.შ.

გ. შეიცავენ მხოლოდ მომხმარებლების პაროლს

19.2.18.თითოეულ საადრიცხვო ჩანაწერს აქვს უნიკალური იდენტიფიკატორი (login ID) ანუ უსაფრთხოების იდენტიფიკატორი (Security IDentification, SID), რომლის საშუალებით:

ა. მომხმარებელი აიდენტიფიცირებს მხოლოდ პაროლს ქსელში

- ბ. მომხმარებელი არ აიდენტიფიცირებს თავის თავს ქსელში
- გ. მომხმარებელი აიდენტიფიცირებს თავის თავს ქსელში

19.2.19. იდენტიფიკატორი არის გრძელი თექვსმეტობითი რიცხვი, რომელიც სააღრიცხვო ჩანაწერის შექმნის დროს:

- ა. ოპერაციული სისტემის მიერ შემთხვევითი გზით გენერირდება
- ბ. ოპერაციული სისტემის მიერ შემთხვევითი გზით არ გენერირდება
- გ. სერვერის მიერ შემთხვევითი გზით გენერირდება

19.2.20. Windows NT-ის საშუალებებით აუტენტიფიცირების შემთხვევაში დომენის მომხმარებლის სააღრიცხვო ჩანაწერი (login ID) ინახება სერვერის:

- ა. model მონაცემთა ბაზაში
- ბ. master მონაცემთა ბაზაში
- გ. tempdb მონაცემთა ბაზაში

19.2.21. Windows NT-ის საშუალებებით აუტენტიფიცირების შემთხვევაში:

- ა. ინფორმაცია მომხმარებლის სახელის, პაროლის და ა.შ. შესახებ ინახება model მონაცემთა ბაზაში
- ბ. ინფორმაცია მომხმარებლის სახელის, პაროლის და ა.შ. შესახებ არ ინახება დომენის მონაცემთა ბაზაში
- გ. ინფორმაცია მომხმარებლის სახელის, პაროლის და ა.შ. შესახებ ინახება დომენის მონაცემთა ბაზაში

19.2.22. Windows NT-ის საშუალებებით აუტენტიფიცირების დროს მომხმარებელს:

- ა. არ სჭირდება სახელისა და პაროლის მითითება სერვერთან მიმართვის მიზნით
- ბ. სჭირდება სახელისა და პაროლის მითითება სერვერთან მიმართვის მიზნით
- გ. სჭირდება მხოლოდ პაროლის მითითება სერვერთან მიმართვის მიზნით

19.2.23. Windows NT-ის საშუალებებით აუტენტიფიცირების დროს მომხმარებელს არ სჭირდება სახელისა და პაროლის მითითება სერვერთან მიმართვის მიზნით, რადგან მომხმარებლის სახელისა და პაროლის შემოწმებას:

- ა. Windows NT-ის დომენის კონტროლერი არ ასრულებს
- ბ. Windows NT-ის დომენის კონტროლერი ასრულებს
- გ. master დომენის კონტროლერი ასრულებს

19.2.24. სერვერი იმ მომხმარებლის სახელს, რომელიც შეერთების დამყარებას ცდილობს,:

- ა. Model მონაცემთა ბაზის syslogins სისტემურ წარმოდგენაში ადარებს იმ სახელს, რომლებსთვისაც მიმართვა ნებადართულია
- ბ. Master მონაცემთა ბაზის syslogins სისტემურ წარმოდგენაში ადარებს იმ სახელს, რომლებსთვისაც მიმართვა ნებადართული არაა
- გ. Master მონაცემთა ბაზის syslogins სისტემურ წარმოდგენაში ადარებს იმ სახელს, რომლებსთვისაც მიმართვა ნებადართულია

19.2.25. სერვერთან შეერთების დასამყარებლად, რომელიც იმყოფება დომენში, რომელთანაც:

- ა. არ არის დამყარებული სანდო ურთიერთობები, შეგვიძლია გამოვიყენოთ სერვერის აუტენტიფიცირების რეჟიმი

- ბ. არის დამყარებული სანდო ურთიეთობები, შეგვიძლია გამოვიყენოთ სერვერის აუტენტიფიცირების რეჟიმი
- გ. არ არის დამყარებული სანდო ურთიეთობები, არ შეგვიძლია გამოვიყენოთ სერვერის აუტენტიფიცირების რეჟიმი

19.2.26. სერვერის აუტენტიფიცირების რეჟიმი გამოიყენება მაშინ, როცა:

- ა. არის დომენში დარეგისტრირების საშუალება
- ბ. არ არის დომენში დარეგისტრირების საშუალება
- გ. ხან არის და ხან არ არის დომენში დარეგისტრირების საშუალება

19.2.27. სერვერის აუტენტიფიცირების გამოყენების შემთხვევაში სერვერთან მიმართვა:

- ა. ხორციელდება სერვერის პაროლის საფუძველზე
- ბ. არ ხორციელდება სერვერის სააღრიცხვო ჩანაწერების საფუძველზე
- გ. ხორციელდება სერვერის სააღრიცხვო ჩანაწერების საფუძველზე

19.2.28. სერვერის საშუალებებით აუტენტიფიცირებისათვის სერვერის sysadmin ან securityadmin სტანდარტული როლის წევრმა მომხმარებლისთვის:

- ა. უნდა შექმნას სააღრიცხვო ჩანაწერი და შეასრულოს მისი კონფიგურირება
- ბ. არ უნდა შექმნას სააღრიცხვო ჩანაწერი და შეასრულოს მისი კონფიგურირება
- გ. უნდა შექმნას სააღრიცხვო ჩანაწერი და არ შეასრულოს მისი კონფიგურირება

19.2.29. სააღრიცხვო ჩანაწერი:

- ა. არ შეიცავს სააღრიცხვო ჩანაწერის სახელს, სერვერის უნიკალური იდენტიფიკატორსა და პაროლს
- ბ. შეიცავს სააღრიცხვო ჩანაწერის სახელს, სერვერის უნიკალური იდენტიფიკატორსა და პაროლს
- გ. შეიცავს სააღრიცხვო ჩანაწერის პაროლს

19.2.30. სააღრიცხვო ჩანაწერის სახელი, სერვერის უნიკალური იდენტიფიკატორი და პაროლი:

- ა. ინახება model მონაცემთა ბაზაში
- ბ. არ ინახება master მონაცემთა ბაზაში
- გ. ინახება master მონაცემთა ბაზაში

19.2.31. სერვერის აუტენტიფიცირების გამოყენების შემთხვევაში:

- ა. სერვერი თვითონ ამოწმებს სახელისა და პაროლის სისწორეს
- ბ. სერვერი თვითონ არ ამოწმებს სახელისა და პაროლის სისწორეს
- გ. სერვერი თვითონ ამოწმებს მხოლოდ პაროლის სისწორეს

უსაფრთხოების სისტემის ელემენტები

19.3.1. სერვერის უსაფრთხოების სისტემის ელემენტებია:

- ა. სააღრიცხვო ჩანაწერები (login), მომხმარებლები (user)
- ბ. სააღრიცხვო ჩანაწერები (login), მომხმარებლები (user), როლები (role) და ჯგუფები (group)
- გ. როლები (role) და ჯგუფები (group)

19.3.2. მომხმარებელმა, რომელიც სერვერს უერთდება:

- ა. უნდა მოახდინოს თავისი თავის აუტენტიფიცირება სააღრიცხვო ჩანაწერის გამოყენებით
- ბ. არ უნდა მოახდინოს თავისი თავის აუტენტიფიცირება სააღრიცხვო ჩანაწერის გამოყენებით
- გ. უნდა მოახდინოს თავისი თავის აუტენტიფიცირება სააღრიცხვო ჩანაწერის გამოყენებლად

19.3.3. აუტენტიფიცირების წარმატებით გავლის შემდეგ მომხმარებელს:

- ა. არ შეეძლება სერვერთან მიმართვა
- ბ. შეეძლება სერვერთან მიმართვა
- გ. ხანდახან შეეძლება სერვერთან მიმართვა

19.3.4. მონაცემთა ბაზასთან მიმართვისათვის მომხმარებლის სააღრიცხვო ჩანაწერი (login):

- ა. აისახება მონაცემთა ბაზის პაროლში
- ბ. არ აისახება მონაცემთა ბაზის მომხმარებელში (user)
- გ. აისახება მონაცემთა ბაზის მომხმარებელში (user)

19.3.5. მონაცემთა ბაზის მომხმარებელს:

- ა. შეუძლია მიმართოს ამ მონაცემთა ბაზის ნებისმიერ ობიექტს: ცხრილს, წარმოდგენას, შენახულ პროცედურას და ა.შ.
- ბ. არ შეუძლია მიმართოს ამ მონაცემთა ბაზის ნებისმიერ ობიექტს: ცხრილს, წარმოდგენას, შენახულ პროცედურას და ა.შ.
- გ. შეუძლია მიმართოს ამ მონაცემთა ბაზის მხოლოდ ცხრილს

19.3.6. მონაცემთა ბაზის მომხმარებელში შეიძლება აისახოს:

- ა. Windows NT-ის სააღრიცხვო ჩანაწერი; Windows NT-ის ჯგუფი;
- ბ. Windows NT-ის სააღრიცხვო ჩანაწერი; Windows NT-ის ჯგუფი; სერვერის სააღრიცხვო ჩანაწერი
- გ. Windows NT-ის ჯგუფი; სერვერის სააღრიცხვო ჩანაწერი

19.3.7. ასახვები ინახება მიმდინარე მონაცემთა ბაზის:

- ა. sysusers ფუნქციაში
- ბ. sysusers ცხრილში
- გ. sysusers სისტემურ წარმოდგენაში

19.3.8. guest მომხმარებელს:

- ა. ეძლევა მინიმალური უფლებები მხოლოდ წაკითხვის რეჟიმში
- ბ. არ ეძლევა მინიმალური უფლებები მხოლოდ წაკითხვის რეჟიმში
- გ. ეძლევა მაქსიმალური უფლებები მხოლოდ ჩაწერის რეჟიმში

სქემები

19.4.1. სქემა:

- ა. არ არის მასში შემავალი ობიექტების კოლექცია, რომელიც ერთ სახელების სივრცეს ქმნის
- ბ. არის მასში შემავალი ობიექტების კოლექცია, რომელიც ერთ სახელების სივრცეს ქმნის

გ. არის მასში შემავალი ობიექტების კოლექცია, რომელიც ერთ სახელების სივრცეს არ ქმნის

19.4.2. სახელების სივრცე არის სიმრავლე, რომლის თითოეულ ელემენტს უნიკალური სახელი:

ა. ხანდახან აქვს

ბ. არ აქვს

გ. აქვს

19.4.3. სქემა:

ა. შეიძლება შეიცავდეს ცხრილებს, წარმოდგენებს, ფუნქციებს, პროცედურებს, ტრიგერებს, ინდექსებსა და მომხმარებლებს

ბ. არ შეიძლება შეიცავდეს ცხრილებს, წარმოდგენებს, ფუნქციებს, პროცედურებს, ტრიგერებს, ინდექსებსა და მომხმარებლებს

გ. შეიძლება შეიცავდეს მხოლოდ ცხრილებს და მომხმარებლებს

19.4.4. ერთი სქემის შიგნით ობიექტებს სხვადასხვა სახელები:

ა. არ უნდა ჰქონდეს

ბ. უნდა ჰქონდეს

გ. ხანდახან უნდა ჰქონდეს

19.4.5. სხვადასხვა სქემაში:

ა. არ უნდა იყოს ერთნაირი სახელის მქონე ობიექტები

ბ. უნდა იყოს ერთნაირი სახელის მქონე ობიექტები

გ. შეიძლება იყოს ერთნაირი სახელის მქონე ობიექტები

19.4.6. სქემის შესაქმნელად გამოიყენება:

ა. CREATE SCHEMA ბრძანება

ბ. CREATE TABLE ბრძანება

გ. CREATE INDEX ბრძანება

19.4.7. CREATE SCHEMA ბრძანების სინტაქსია:

ა. CREATE SCHEMA სქემის_სახელი.ცხრილის_სახელი

ბ. CREATE SCHEMA სქემის_სახელი

გ. CREATE SCHEMA სქემის_სახელი.სვეტის_სახელი

19.4.8. სქემის წასაშლელად გამოიყენება:

ა. DROP TABLE ბრძანება

ბ. DROP DATABASE ბრძანება

გ. DROP SCHEMA ბრძანება

19.4.9. DROP SCHEMA ბრძანების სინტაქსია:

ა. DROP SCHEMA სქემის_სახელი

ბ. DROP SCHEMA სქემის_სახელი.სვეტის_სახელი

გ. DROP SCHEMA სქემის_სახელი.ცხრილის_სახელი

19.4.10. CREATE SCHEMA Sqema1 ბრძანების შესრულების შედეგად:

- ა. წაიშლება Sqema1 სქემა
- ბ. შეიქმნება Sqema1 სქემა
- გ. არ შეიქმნება Sqema1 სქემა

19.4.11.DROP SCHEMA Sqema1 ბრძანების შესრულების შედეგად:

- ა. შეიქმნება Sqema1 სქემა
- ბ. არ წაიშლება Sqema1 სქემა
- გ. წაიშლება Sqema1 სქემა

19.4.12.CREATE TABLE Sqema1.Cxrili1 (svet1 int, sveti2 int) ბრძანების შესრულების შედეგად:

- ა. Cxrili1 ცხრილი შეიქმნება Sqema1 სქემაში
- ბ. Sqema1 ცხრილი შეიქმნება Cxrili1 სქემაში
- გ. Cxrili1 ცხრილი წაიშლება Sqema1 სქემიდან

19.4.13.DROP TABLE Sqema1.Cxrili1 ბრძანების შესრულების შედეგად:

- ა. Sqema1 ცხრილი შეიქმნება Cxrili1 სქემიდან
- ბ. Cxrili1 ცხრილი წაიშლება Sqema1 სქემიდან
- გ. Cxrili1 ცხრილი შეიქმნება Sqema1 სქემაში

19.4.14.მონაცემთა ბაზის შექმნის დროს:

- ა. იქმნება არასტანდარტული სქემები
- ბ. არ იქმნება სტანდარტული სქემები
- გ. იქმნება სტანდარტული სქემები

19.4.15.მონაცემთა ბაზის შექმნის დროს იქმნება სტანდარტული სქემები. ერთ-ერთი მათგანია:

- ა. dbo
- ბ. DROP SCHEMA
- გ. SQL

19.4.16.თუ ობიექტის შექმნის დროს ან ობიექტთან მიმართვის დროს სქემის სახელი არ არის მითითებული, მაშინ:

- ა. ავტომატურად არ იგულისხმება dbo სქემა
- ბ. ავტომატურად იგულისხმება dbi სქემა
- გ. არაავტომატურად იგულისხმება dbi სქემა

19.4.17.ჩვენ შეგვიძლია:

- ა. შევქმნათ სქემები და მასში არ მოვათავსოთ სხვადასხვა ობიექტები
- ბ. არ შევქმნათ სქემები და მასში მოვათავსოთ სხვადასხვა ობიექტები
- გ. შევქმნათ სქემები და მასში მოვათავსოთ სხვადასხვა ობიექტები

19.4.18.სახელების გადაწყვეტა არის პროცესი, როცა ობიექტის სახელიდან:

- ა. ხდება მისი სრულად განსაზღვრული სახელის ფორმირება და მისი უნიკალურობის შემოწმება
- ბ. არ ხდება მისი სრულად განსაზღვრული სახელის ფორმირება და მისი უნიკალურობის შემოწმება

გ. ხდება მისი არასრულად განსაზღვრული სახელის ფორმირება და მისი უნიკალურობის შემოწმება

19.4.19.თუ სქემას არ მივუთითებთ, მაშინ ცხრილი:

- ა. არ მოიძებნება dbi სქემაში
- ბ. მოიძებნება dbi სქემაში
- გ. მოიძებნება dbi პაროლში

19.4.20.თითოეულ მომხმარებელს:

- ა. აქვს არანაგულისხმევი სქემა
- ბ. არ აქვს ნაგულისხმევი სქემა
- გ. აქვს ნაგულისხმევი სქემა

19.4.21.თითოეულ მომხმარებელს აქვს ნაგულისხმევი სქემა. ის:

- ა. არის პირველი სქემა, რომელიც მოიძებნება სერვერის მიერ, როცა ის წყვეტს ობიექტების სახელებს
- ბ. არ არის პირველი სქემა, რომელიც მოიძებნება სერვერის მიერ, როცა ის წყვეტს ობიექტების სახელებს
- გ. არის პირველი სქემა, რომელიც არ მოიძებნება სერვერის მიერ, როცა ის წყვეტს ობიექტების სახელებს

19.4.22.ნაგულისხმევი სქემა:

- ა. არ შეიძლება განისაზღვროს CREATE USER ბრძანების DEFAULT_SCHEMA რეჟიმის გამოყენებით
- ბ. შეიძლება განისაზღვროს CREATE USER ბრძანების DEFAULT_SCHEMA რეჟიმის გამოყენებით
- გ. შეიძლება განისაზღვროს DROP USER ბრძანების DEFAULT_SCHEMA რეჟიმის გამოყენებით

19.4.23.თუ DEFAULT_SCHEMA განუსაზღვრელია, მაშინ ნაგულისხმევი სქემა:

- ა. იქნება SQL
- ბ. არ იქნება dbo
- გ. იქნება dbo

19.4.24.CREATE SCHEMA Sqema1; ბრძანების შესრულების შედეგად:

- ა. შეიქმნება Sqema1 სქემა
- ბ. არ შეიქმნება Sqema1 სქემა
- გ. წაიშლება Sqema1 სქემა

19.4.25.Sqema1 სქემის შესაქმნელად უნდა შევასრულოთ ბრძანება:

- ა. DROP SCHEMA Sqema1;
- ბ. CREATE SCHEMA Sqema1;
- გ. CREATE TABLE Sqema1;

19.4.26.DROP SCHEMA Sqema1; ბრძანების შესრულების შედეგად:

- ა. შეიქმნება Sqema1 სქემა

- ბ. არ წაიშლება Sqema1 სქემა
- გ. წაიშლება Sqema1 სქემა

19.4.27. Sqema1 სქემის წასაშლელად უნდა შევასრულოთ ბრძანება:

- ა. DROP INDEX Sqema1;
- ბ. CREATE SCHEMA Sqema1;
- გ. DROP SCHEMA Sqema1;

მომხმარებლები

19.5.1. მას შემდეგ, რაც მომხმარებელმა წარმატებით გაიარა აუტენტიფიცირება და მიიღო საადრიცხვო ჩანაწერის იდენტიფიკატორი (login ID), ის:

- ა. ითვლება დარეგისტრირებულად და მას ეძლევა სერვერთან მიმართვის უფლება
- ბ. არ ითვლება დარეგისტრირებულად და მას ეძლევა სერვერთან მიმართვის უფლება
- გ. ითვლება დარეგისტრირებულად და მას არ ეძლევა სერვერთან მიმართვის უფლება

19.5.2. მომხმარებლის საადრიცხვო ჩანაწერი (login):

- ა. არ ასოცირდება კონკრეტული მონაცემთა ბაზის მომხმარებელთან (user)
- ბ. ასოცირდება კონკრეტული მონაცემთა ბაზის მომხმარებელთან (user)
- გ. ასოცირდება კონკრეტული მონაცემთა ბაზის პაროლთან (user)

19.5.3. მომხმარებლები წარმოადგენენ სერვერის სპეციალურ ობიექტებს, რომელთა საშუალებით:

- ა. განისაზღვრება მიმართვის უფლებები და ცხრილში ობიექტების მფლობელობა
- ბ. არ განისაზღვრება მიმართვის უფლებები და მონაცემთა ბაზაში ობიექტების მფლობელობა
- გ. განისაზღვრება მიმართვის უფლებები და მონაცემთა ბაზაში ობიექტების მფლობელობა

19.5.4. მონაცემთა ბაზის შექმნის დროს განისაზღვრება:

- ა. ორი სტანდარტული მომხმარებელი: dbo და guest
- ბ. ერთი სტანდარტული მომხმარებელი guest
- გ. ერთი სტანდარტული მომხმარებელი dbo

19.5.5. თუ საადრიცხვო ჩანაწერი აშკარად არ არის დაკავშირებული მომხმარებელთან მაშინ მომხმარებელს:

- ა. არ ეძლევა მონაცემთა ბაზებთან არააშკარა მიმართვის უფლება guest სტუმრის სახელის გამოყენებით
- ბ. ეძლევა მონაცემთა ბაზებთან არააშკარა მიმართვის უფლება guest სტუმრის სახელის გამოყენებით
- გ. ეძლევა მონაცემთა ბაზებთან არააშკარა მიმართვის უფლება guest სტუმრის სახელის გამოყენების გარეშე

19.5.6. თუ საადრიცხვო ჩანაწერი აშკარად არ არის დაკავშირებული მომხმარებელთან მაშინ მომხმარებელს ეძლევა მონაცემთა ბაზებთან არააშკარა მიმართვის უფლება guest სტუმრის სახელის გამოყენებით. ე.ი. ასეთი საადრიცხვო ჩანაწერი, რომელმაც:

- ა. მიიღო სერვერთან მიმართვის უფლება, ავტომატურად არ აისახება guest მომხმარებელში ყველა მონაცემთა ბაზისთვის
- ბ. არ მიიღო სერვერთან მიმართვის უფლება, ავტომატურად აისახება guest მომხმარებელში ყველა მონაცემთა ბაზისთვის
- გ. მიიღო სერვერთან მიმართვის უფლება, ავტომატურად აისახება guest მომხმარებელში ყველა მონაცემთა ბაზისთვის

19.5.7. მაქსიმალური უსაფრთხოების უზრუნველყოფისათვის შეგვიძლია:

- ა. წავშალოთ guest მომხმარებელი ყველა მონაცემთა ბაზიდან, გარდა master და tempdb სისტემური ბაზებიდან
- ბ. არ წავშალოთ guest მომხმარებელი ყველა მონაცემთა ბაზიდან, გარდა master და tempdb სისტემური ბაზებიდან
- გ. წავშალოთ guest მომხმარებელი ყველა მონაცემთა ბაზიდან, master და tempdb სისტემური ბაზების ჩათვლით

19.5.8. master მონაცემთა ბაზაში guest:

- ა. არ გამოიყენება სისტემური შენახული პროცედურების შესასრულებლად ჩვეულებრივი მომხმარებლების მიერ
- ბ. გამოიყენება სისტემური შენახული პროცედურების შესასრულებლად ჩვეულებრივი მომხმარებლების მიერ
- გ. გამოიყენება სისტემური შენახული პროცედურების წასაშლელად ჩვეულებრივი მომხმარებლების მიერ

19.5.9. tempdb მონაცემთა ბაზაში guest:

- ა. არ გამოიყენება დროებითი ობიექტების შესაქმნელად ნებისმიერი მომხმარებლის მიერ
- ბ. გამოიყენება დროებითი ობიექტების წასაშლელად ნებისმიერი მომხმარებლის მიერ
- გ. გამოიყენება დროებითი ობიექტების შესაქმნელად ნებისმიერი მომხმარებლის მიერ

19.5.10. მონაცემთა ბაზის მფლობელი (DataBase Owner, dbo) - სპეციალური მომხმარებელია, რომელსაც:

- ა. მაქსიმალური უფლებები აქვს მონაცემთა ბაზაში
- ბ. მაქსიმალური უფლებები არ აქვს მონაცემთა ბაზაში
- გ. მინიმალური უფლებები აქვს მონაცემთა ბაზაში

19.5.11. dbo მომხმარებლის წაშლა:

- ა. შეიძლება
- ბ. არ შეიძლება
- გ. ხანდახან შეიძლება

19.5.12. საადრიცხო ჩანაწერის (login) დასაკავშირებლად გარკვეულ მომხმარებელთან (user):

- ა. შეგვიძლია გამოვიყენოთ sp_dbuser შენახული პროცედურა
- ბ. არ შეგვიძლია გამოვიყენოთ sp_adduser შენახული პროცედურა
- გ. შეგვიძლია გამოვიყენოთ sp_adduser შენახული პროცედურა

19.5.13. sp_adduser შენახული პროცედურა მიმდინარე მონაცემთა :

- ა. ბაზას უმატებს ახალ მომხმარებელს

- ბ. ბაზას უმატებს ძველ მომხმარებელს
- გ. ბაზიდან შლის ახალ მომხმარებელს

19.5.14.sp_adduser შენახული პროცედურის სინტაქსია:

- ა. sp_adduser [, 'მომხმარებლის_სახელი'] [, 'როლის_სახელი']
- ბ. sp_adduser 'სააღრიცხვო_ჩანაწერის_სახელი' [, 'მომხმარებლის_სახელი'] [, 'როლის_სახელი']
- გ. sp_adduser [, 'როლის_სახელი']

19.5.15.EXEC sp_adduser 'Login_1', 'Login_1', 'Roli_1' ბრძანების შესრულების შედეგად:

- ა. შესრულდება Roli_1 მომხმარებლის ჩართვა მიმდინარე მონაცემთა ბაზის Login_1 როლში Login_1 სააღრიცხვო ჩანაწერის გამოყენებით
- ბ. არ შესრულდება Login_1 მომხმარებლის ჩართვა მიმდინარე მონაცემთა ბაზის Roli_1 როლში Login_1 სააღრიცხვო ჩანაწერის გამოყენებით
- გ. შესრულდება Login_1 მომხმარებლის ჩართვა მიმდინარე მონაცემთა ბაზის Roli_1 როლში Login_1 სააღრიცხვო ჩანაწერის გამოყენებით

19.5.16.მიმდინარე მონაცემთა ბაზაში მომხმარებლის დასამატებლად:

- ა. შეგვიძლია გამოვიყენოთ CREATE USER ბრძანება
- ბ. არ შეგვიძლია გამოვიყენოთ CREATE USER ბრძანება
- გ. შეგვიძლია გამოვიყენოთ DROP USER ბრძანება

19.5.17.CREATE USER ბრძანების სინტაქსია:

- ა. CREATE USER [WITH DEFAULT_SCHEMA = სქემის_სახელი]
- ბ. CREATE USER მომხმარებლის_სახელი
[{ FOR LOGIN სააღრიცხვო_ჩანაწერის_სახელი | WITHOUT LOGIN }]
[WITH DEFAULT_SCHEMA = სქემის_სახელი]
- გ. CREATE USER [{ FOR LOGIN სააღრიცხვო_ჩანაწერის_სახელი | WITHOUT LOGIN }]

19.5.18.USE Baza_1

CREATE USER User_1 FOR LOGIN [Romani-PC\Romani] ბრძანებების შესრულების შედეგად:

- ა. Romani-PC\Romani სააღრიცხვო ჩანაწერისთვის არ იქმნება User_1 მომხმარებელი Baza_1 ბაზაში
- ბ. User_1 სააღრიცხვო ჩანაწერისთვის იქმნება Romani-PC\Romani მომხმარებელი Baza_1 ბაზაში
- გ. Romani-PC\Romani სააღრიცხვო ჩანაწერისთვის იქმნება User_1 მომხმარებელი Baza_1 ბაზაში

სერვერის როლები. მონაცემთა ბაზების როლები

19.6.1. როლი საშუალებას:

- ა. გვაძლევს გავაერთიანოთ მომხმარებლები, რომლებიც ერთნაირ ფუნქციებს ასრულებენ
- ბ. არ გვაძლევს გავაერთიანოთ მომხმარებლები, რომლებიც ერთნაირ ფუნქციებს ასრულებენ
- გ. გვაძლევს გავაერთიანოთ მომხმარებლები, რომლებიც ერთნაირ ფუნქციებს არ ასრულებენ

19.6.2. როლი საშუალებას გვაძლევს გავაერთიანოთ მომხმარებლები, რომლებიც ერთნაირ ფუნქციებს ასრულებენ:

- ა. ეს საჭიროა სერვერის უსაფრთხოების სისტემის ადმინისტრირების გასართულებლად
- ბ. ეს საჭიროა სერვერის უსაფრთხოების სისტემის ადმინისტრირების გასამარტივებლად
- გ. ეს არ არის საჭირო სერვერის უსაფრთხოების სისტემის ადმინისტრირების გასამარტივებლად

19.6.3. სერვერზე რეალიზებულია:

- ა. ერთი სტანდარტული როლი სერვერის დონეზე
- ბ. ერთი სტანდარტული როლი მონაცემთა ბაზის დონეზე
- გ. ორი სტანდარტული როლი: სერვერის დონეზე და მონაცემთა ბაზის დონეზე

19.6.4. სერვერის დაყენების დროს იქმნება სერვერის:

- ა. ცხრა ფიქსირებული როლი
- ბ. რვა ფიქსირებული როლი
- გ. ათი ფიქსირებული როლი

19.6.5. სერვერის დაყენების დროს იქმნება მონაცემთა ბაზის:

- ა. რვა ფიქსირებული როლი
- ბ. ცხრა ფიქსირებული როლი
- გ. ათი ფიქსირებული როლი

19.6.6. სერვერის დაყენების დროს იქმნება სერვერის ცხრა ფიქსირებული როლი და მონაცემთა ბაზის ცხრა ფიქსირებული როლი. ამ როლებს ჩვენ:

- ა. ვერ წავშლით, მაგრამ შეიძლება მათი უფლებების მოდიფიცირება
- ბ. წავშლით და შეიძლება მათი უფლებების მოდიფიცირება
- გ. ვერ წავშლით და არ შეიძლება მათი უფლებების მოდიფიცირება

19.6.7. იმისათვის, რომ მომხმარებელს მივანიჭოთ ის უფლებები, რომელიც აქვს სერვერის რომელიმე ფიქსირებულ როლს, საჭიროა:

- ა. მომხმარებლის ამ როლში ჩართვა
- ბ. მომხმარებლის ამ როლიდან წაშლა
- გ. ამ როლის წაშლა

19.6.8. სერვერის Sysadmin როლის წევრს:

- ა. არ შეუძლია ნებისმიერი მოქმედების შესრულება სერვერზე
- ბ. შეუძლია ნებისმიერი მოქმედების შესრულება სერვერზე
- გ. შეუძლია მხოლოდ სერვერის ფაილების მართვა

19.6.9. სერვერის Dbcreator როლის წევრს:

- ა. შეუძლია მონაცემთა ბაზების შექმნა, მაგრამ არ შეუძლია მისი მოდიფიცირება
- ბ. არ შეუძლია მონაცემთა ბაზების შექმნა და მოდიფიცირება
- გ. შეუძლია მონაცემთა ბაზების შექმნა და მოდიფიცირება

19.6.10. სერვერის Diskadmin როლის წევრი:

- ა. მართავს სერვერის ფაილებს
- ბ. არ მართავს სერვერის ფაილებს
- გ. მონაცემთა ბაზების შექმნასა და მოდიფიცირებას

19.6.11. მონაცემთა ბაზების როლები (database role) საშუალებას:

- ა. არ გვაძლევს მომხმარებლები გავაერთიანოთ ერთ ადმინისტრაციულ ერთეულში და მასთან ვიმუშაოთ როგორც ჩვეულებრივ მომხმარებელთან
- ბ. გვაძლევს მომხმარებლები გავაერთიანოთ ერთ ადმინისტრაციულ ერთეულში და მასთან ვიმუშაოთ როგორც ჩვეულებრივ მომხმარებელთან
- გ. გვაძლევს მომხმარებლები გავაერთიანოთ ერთ ადმინისტრაციულ ერთეულში და მასთან არ ვიმუშაოთ როგორც ჩვეულებრივ მომხმარებელთან

19.6.12. განვსაზღვრავთ რა მონაცემთა ბაზის ობიექტებთან მიმართვის უფლებებს კონკრეტული როლისთვის, ჩვენ ამით:

- ა. ავტომატურად არ ვაძლევთ იგივე უფლებებს ამ როლის ყველა წევრს
- ბ. არაავტომატურად ვაძლევთ იგივე უფლებებს ამ როლის ყველა წევრს
- გ. ავტომატურად ვაძლევთ იგივე უფლებებს ამ როლის ყველა წევრს

19.6.13. თუ მომხმარებელს უნდა მიენიჭოს იგივე უფლებები, როგორც მინიჭებული აქვს როლს, მაშინ ეს მომხმარებელი ამ:

- ა. როლში უნდა ჩავრთოთ
- ბ. როლში არ უნდა ჩავრთოთ
- გ. როლიდან უნდა წავშალოთ

19.6.14. მონაცემთა ბაზის როლში შეგვიძლია ჩავრთოთ:

- ა. მხოლოდ სერვერის მომხმარებელი
- ბ. სერვერის მომხმარებელი; სერვერის როლი; Windows NT-ის მომხმარებლები; Windows NT-ის ჯგუფები, რომლებსაც წინასწარ მიცემული აქვთ საჭირო მონაცემთა ბაზასთან მიმართვის უფლება
- გ. მხოლოდ სერვერის როლი

19.6.15. მიმდინარე მონაცემთა ბაზაში როლის დასამატებლად გამოიყენება შენახული პროცედურა:

- ა. sp_add
- ბ. sp_role
- გ. sp_addrole

19.6.16. მიმდინარე მონაცემთა ბაზაში როლის დასამატებლად sp_addrole შენახული პროცედურა გამოიყენება. მისი სინტაქსია:

- ა. sp_addrole [@rolename =] 'როლის_სახელი' [, [@ownername =] 'მფლობელის_სახელი']
- ბ. sp_addrole [, [@ownername =] 'მფლობელის_სახელი']
- გ. sp_addrole [@rolename =] [, 'მფლობელის_სახელი']

19.6.17. USE Shekveta; EXEC sp_addrole 'Roli_2' კოდის შესრულების შედეგად:

- ა. Shekveta მონაცემთა ბაზაში არ იქმნება Roli_2 როლი, რომლის მფლობელი იქნება dbo
- ბ. Shekveta მონაცემთა ბაზაში იქმნება Roli_2 როლი, რომლის მფლობელი იქნება dbo

გ. Shekveta მონაცემთა ბაზაში იქმნება Roli_2 როლი, რომლის მფლობელი არ იქნება dbo

19.6.18. USE Shekveta; EXEC sp_addrole 'Roli_3', 'User_3' კოდის შესრულების შედეგად:

ა. Shekveta მონაცემთა ბაზაში არ იქმნება Roli_3 როლი, რომლის მფლობელი იქნება მიმდინარე მონაცემთა ბაზის User_3 მომხმარებელი

ბ. Shekveta მონაცემთა ბაზაში იქმნება Roli_3 როლი, რომლის მფლობელი არ იქნება მიმდინარე მონაცემთა ბაზის User_3 მომხმარებელი

გ. Shekveta მონაცემთა ბაზაში იქმნება Roli_3 როლი, რომლის მფლობელი იქნება მიმდინარე მონაცემთა ბაზის User_3 მომხმარებელი

19.6.19. მიმდინარე მონაცემთა ბაზის როლში მონაცემთა ბაზის მომხმარებლის, მონაცემთა ბაზის როლის, Windows-ის სააღრიცხვო ჩანაწერისა და Windows-ის ჯგუფის ჩასართავად გამოიყენება შენახული პროცედურა:

ა. sp_addrolemember

ბ. sp_addrole

გ. sp_grantdbaccess

19.6.20. მიმდინარე მონაცემთა ბაზის როლში მონაცემთა ბაზის მომხმარებლის, მონაცემთა ბაზის როლის, Windows-ის სააღრიცხვო ჩანაწერისა და Windows-ის ჯგუფის ჩასართავად გამოიყენება sp_addrolemember შენახული პროცედურა. მისი სინტაქსია:

ა. sp_addrolemember [@rolename =] 'როლის_სახელი'

ბ. sp_addrolemember [@rolename =] 'როლის_სახელი',
[@membername =] 'უსაფრთხოების_ობიექტის_სახელი'

გ. sp_addrolemember [@membername =] 'უსაფრთხოების_ობიექტის_სახელი'

19.6.21. EXEC sp_addrolemember 'Roli_1', 'User_1' პროცედურის შესრულების შედეგად:

ა. მოხდება Roli_1 მომხმარებლის ჩართვა მიმდინარე მონაცემთა ბაზის User_1 როლში

ბ. არ მოხდება User_1 მომხმარებლის ჩართვა მიმდინარე მონაცემთა ბაზის Roli_1 როლში

გ. მოხდება User_1 მომხმარებლის ჩართვა მიმდინარე მონაცემთა ბაზის Roli_1 როლში

19.6.22. მონაცემთა ბაზის db_owner ფიქსირებულ როლს:

ა. აქვს ყველა უფლება მონაცემთა ბაზაში

ბ. არ აქვს ყველა უფლება მონაცემთა ბაზაში

გ. აქვს ყველა უფლება სერვერზე

19.6.23. მონაცემთა ბაზის db_accessadmin ფიქსირებულ როლს:

ა. არ შეუძლია მომხმარებლების დამატება და წაშლა

ბ. შეუძლია მომხმარებლების დამატება და წაშლა

გ. შეუძლია მომხმარებლების მხოლოდ დამატება

უსაფრთხოების სისტემის ადმინისტრირება. სააღრიცხვო ჩანაწერების შექმნა და მართვა

19.7.1. სააღრიცხვო ჩანაწერის შესაქმნელად გამოიყენება:

ა. sp_add შენახული პროცედურა

ბ. sp_login შენახული პროცედურა

გ. sp_addlogin შენახული პროცედურა

19.7.2. საადრიცხო ჩანაწერის შესაქმნელად გამოიყენება sp_addlogin შენახული პროცედურა. მისი სინტაქსია:

- ა. sp_addlogin [@loginame =] 'საადრიცხო_ჩანაწერის_სახელი'
[, [@passwd =] 'პაროლი'] [, [@defdb =] 'მონაცემთა_ბაზის_სახელი']
- ბ. sp_addlogin [, [@passwd =] 'პაროლი']
- გ. sp_addlogin [, [@defdb =] 'მონაცემთა_ბაზის_სახელი']

19.7.3. sp_addlogin 'Login_2', 'paroli' შენახული პროცედურის შესრულების შედეგად:

- ა. არ შეიქმნება Login_2 საადრიცხო ჩანაწერი, რომლის პაროლია paroli
- ბ. შეიქმნება Login_2 საადრიცხო ჩანაწერი, რომლის პაროლია paroli
- გ. შეიქმნება paroli საადრიცხო ჩანაწერი, რომლის პაროლია Login_2

19.7.4. sp_addlogin 'Login_3', 'paroli', 'Shekveta' შენახული პროცედურის შესრულების შედეგად:

- ა. შეიქმნება Login_3 საადრიცხო ჩანაწერი. რომლის პაროლია Shekveta, ხოლო ნაგულისხმევი მონაცემთა ბაზა - paroli
- ბ. არ შეიქმნება Login_3 საადრიცხო ჩანაწერი. რომლის პაროლია paroli, ხოლო ნაგულისხმევი მონაცემთა ბაზა - Shekveta
- გ. შეიქმნება Login_3 საადრიცხო ჩანაწერი. რომლის პაროლია paroli, ხოლო ნაგულისხმევი მონაცემთა ბაზა - Shekveta

19.7.5. სერვერის საადრიცხო ჩანაწერის პაროლს ცვლის:

- ა. sp_password შენახული პროცედურა
- ბ. sp_addlogin შენახული პროცედურა
- გ. sp_addrolemember შენახული პროცედურა

19.7.6. EXEC sp_password 'Paroli1', 'Paroli2', 'Login_2' შენახული პროცედურის შესრულების შედეგად:

- ა. Paroli2 ძველი პაროლი შეიცვლება Paroli1 ახალი პაროლით Login_2 საადრიცხო ჩანაწერისთვის
- ბ. Paroli1 ძველი პაროლი შეიცვლება Paroli2 ახალი პაროლით Login_2 საადრიცხო ჩანაწერისთვის
- გ. Paroli1 ძველი პაროლი არ შეიცვლება Paroli2 ახალი პაროლით Login_2 საადრიცხო ჩანაწერისთვის

მიმართვის უფლებები. მონაცემთა ბაზის ობიექტებთან მიმართვის უფლებები

19.8.1. როცა მომხმარებლები უერთდებიან სერვერს, მოქმედებები, რომელთა შესრულებაც მათ შეუძლიათ, განისაზღვრება უფლებებით (ნებართვებით), რომლებიც:

- ა. მიენიჭა მათ საადრიცხო ჩანაწერებს, ჯგუფს ან როლს, რომელშიც ისინი არ შედიან
- ბ. არ მიენიჭა მათ საადრიცხო ჩანაწერებს, ჯგუფს ან როლს, რომელშიც ისინი შედიან
- გ. მიენიჭა მათ საადრიცხო ჩანაწერებს, ჯგუფს ან როლს, რომელშიც ისინი შედიან

19.8.2. სერვერზე უფლებები შეიძლება შემდეგ კატეგორიად დაიყოს:

- ა. მონაცემთა ბაზების ობიექტებთან მიმართვის უფლებები; Transact SQL-ის ბრძანებების შესრულების უფლებები; არაცხადი უფლებები (ნებართვები)
- ბ. Transact SQL-ის ბრძანებების შესრულების უფლებები; არაცხადი უფლებები (ნებართვები)
- გ. მონაცემთა ბაზების ობიექტებთან მიმართვის უფლებები; Transact SQL-ის ბრძანებების შესრულების უფლებები

19.8.3. შექმნის შემდეგ, მომხმარებელს:

- ა. არ აქვს მხოლოდ მონაცემთა ბაზის public როლისთვის მინიჭებული უფლებები
- ბ. აქვს მხოლოდ მონაცემთა ბაზის public როლისთვის მინიჭებული უფლებები
- გ. აქვს მხოლოდ მონაცემთა ბაზის public როლისთვის წართმეული უფლებები

19.8.4. public როლისთვის მინიჭებული უფლებები:

- ა. არც ერთი მომხმარებლისათვის არაა მისაწვდომი
- ბ. არა მისაწვდომია ყველა მომხმარებლისათვის
- გ. მისაწვდომია ყველა მომხმარებლისათვის

19.8.5. მომხმარებელს უფლებები ეძლევა:

- ა. ადმინისტრატორის, მონაცემთა ბაზის მფლობელის ან ობიექტის მფლობელის მიერ
- ბ. მხოლოდ ადმინისტრატორის მიერ
- გ. მხოლოდ მონაცემთა ბაზის მფლობელის მიერ

19.8.6. მომხმარებლისათვის უფლებების გარკვეული ნაკრების:

- ა. მისანიჭებლად როლები არ უნდა გამოვიყენოთ
- ბ. მისანიჭებლად როლები უნდა გამოვიყენოთ
- გ. მისანიჭებლად პაროლები უნდა გამოვიყენოთ

19.8.7. შესაბამის როლებში ჩართვის შემდეგ, მომხმარებელი:

- ა. ავტომატურად არ იღებს ყველა იმ უფლებას, რომლებიც როლისთვისაა განკუთვნილი
- ბ. არაავტომატურად იღებს ყველა იმ უფლებას, რომლებიც როლისთვისაა განკუთვნილი
- გ. ავტომატურად იღებს ყველა იმ უფლებას, რომლებიც როლისთვისაა განკუთვნილი

19.8.8. მონაცემთა ბაზის სტანდარტულ როლებს:

- ა. აქვთ მინიჭებული უფლებების გარკვეული ნაკრები
- ბ. არ აქვთ მინიჭებული უფლებების გარკვეული ნაკრები
- გ. აქვთ მინიჭებული უფლებების არანაირი ნაკრები

19.8.9. მონაცემთა ბაზების ობიექტებთან მიმართვის უფლებები:

- ა. არ მართავენ მომხმარებლების მიერ ბრძანებების (SELECT, INSERT, UPDATE და DELETE) შესრულების შესაძლებლობას ცხრილებისა და წარმოდგენებისათვის
- ბ. მართავენ მომხმარებლების მიერ ბრძანებების (SELECT, INSERT, UPDATE და DELETE) შესრულების შესაძლებლობას ცხრილებისა და წარმოდგენებისათვის
- გ. მართავენ მომხმარებლების მიერ ბრძანებების (SELECT, INSERT, UPDATE და DELETE) შესრულების შესაძლებლობას მხოლოდ ცხრილებისათვის

19.8.10. თუ მომხმარებელს სჭირდება ცხრილში ახალი მონაცემების დამატება, მაშინ მას:

- ა. არ უნდა მივცეთ INSERT უფლება
- ბ. უნდა მივცეთ EXECUTE უფლება
- გ. უნდა მივცეთ INSERT უფლება

19.8.11.თუ მომხმარებელს სჭირდება შენახული პროცედურების შესრულება, მაშინ მას:

- ა. უნდა მივცეთ EXECUTE უფლება
- ბ. არ უნდა მივცეთ EXECUTE უფლება
- გ. უნდა მივცეთ INSERT უფლება

19.8.12.სხვადასხვა ობიექტებისთვის გამოიყენება მათთან:

- ა. მიმართვის პაროლების სხვადასხვა ნაკრებები
- ბ. მიმართვის უფლებების სხვადასხვა ნაკრებები
- გ. მიმართვის უფლებების ერთნაირი ნაკრებები

19.8.13.SELECT და UPDATE უფლებები:

- ა. არ გამოიყენება ცხრილის ან წარმოდგენის სვეტის მიმართ
- ბ. გამოიყენება მხოლოდ ცხრილის
- გ. გამოიყენება ცხრილის ან წარმოდგენის სვეტის მიმართ

19.8.14.SELECT, INSERT, UPDATE, DELETE და REFERENCES უფლებები:

- ა. გამოიყენება ცხრილებისა და წარმოდგენებისათვის
- ბ. არ გამოიყენება ცხრილებისა და წარმოდგენებისათვის
- გ. გამოიყენება ცხრილებისა და ფუნქციებისთვის

19.8.15.EXECUTE უფლება:

- ა. არ გამოიყენება მხოლოდ შენახული პროცედურებისა და ფუნქციების მიმართ
- ბ. გამოიყენება მხოლოდ შენახული პროცედურებისა და ფუნქციების მიმართ
- გ. გამოიყენება მხოლოდ ცხრილების მიმართ

19.8.16.INSERT უფლება საშუალებას:

- ა. გვადლევს ცხრილში ან წარმოდგენაში შევცვალოთ ახალი სტრიქონები
- ბ. არ გვადლევს ცხრილში ან წარმოდგენაში ჩავსვათ ახალი სტრიქონები
- გ. გვადლევს ცხრილში ან წარმოდგენაში ჩავსვათ ახალი სტრიქონები

19.8.17.INSERT უფლება საშუალებას გვადლევს ცხრილში ან წარმოდგენაში ჩავსვათ ახალი სტრიქონები. შედეგად, INSERT უფლება:

- ა. შეიძლება გაიცეს მხოლოდ ცხრილის ან წარმოდგენის დონეზე და არა სვეტის დონეზე
- ბ. არ შეიძლება გაიცეს მხოლოდ ცხრილის ან წარმოდგენის დონეზე და არა სვეტის დონეზე
- გ. შეიძლება გაიცეს მხოლოდ სვეტის დონეზე

19.8.18.UPDATE უფლება:

- ა. არ გაიცემა ცხრილის დონეზე, რაც საშუალებას არ გვადლევს მთელ ცხრილში შევცვალოთ მონაცემები, ან სვეტის დონეზე, რაც საშუალებას გვადლევს მხოლოდ კონკრეტულ სვეტში შევცვალოთ მონაცემები

ბ. გაიცემა ცხრილის დონეზე, რაც საშუალებას გვაძლევს მთელ ცხრილში შევცვალოთ მონაცემები, ან სვეტის დონეზე, რაც საშუალებას გვაძლევს მხოლოდ კონკრეტულ სვეტში შევცვალოთ მონაცემები

გ. არ გაიცემა ცხრილის დონეზე, რაც საშუალებას გვაძლევს მთელ ცხრილში შევცვალოთ მონაცემები, ან სვეტის დონეზე, რაც საშუალებას არ გვაძლევს მხოლოდ კონკრეტულ სვეტში შევცვალოთ მონაცემები

19.8.19.DELETE უფლება:

ა. გვაძლევს ცხრილში ან წარმოდგენაში სტრიქონების ჩამატების საშუალებას

ბ. არ გვაძლევს ცხრილიდან ან წარმოდგენიდან სტრიქონების წაშლის საშუალებას

გ. გვაძლევს ცხრილიდან ან წარმოდგენიდან სტრიქონების წაშლის საშუალებას

19.8.20.DELETE უფლება:

ა. შეიძლება გაიცეს მხოლოდ ცხრილის ან წარმოდგენის დონეზე და არა სვეტის დონეზე

ბ. არ შეიძლება გაიცეს მხოლოდ ცხრილის ან წარმოდგენის დონეზე და არა სვეტის დონეზე

გ. შეიძლება გაიცეს მხოლოდ სვეტის დონეზე

19.8.21.SELECT უფლება იძლევა მონაცემების ამორჩევის შესაძლებლობას და:

ა. შეიძლება გაიცეს მხოლოდ ცხრილის დონეზე

ბ. შეიძლება გაიცეს როგორც ცხრილის, ისე სვეტის დონეზე

გ. არ შეიძლება გაიცეს როგორც ცხრილის, ისე სვეტის დონეზე

19.8.22.მონაცემთა ბაზის მომხმარებლისთვის მონაცემთა ბაზის ყველა ობიექტთან მიმართვის უფლების მიცემა ან წართმევა:

ა. შეიძლება მონაცემთა ბაზის არაჩადგმული როლების საშუალებით

ბ. არ შეიძლება მონაცემთა ბაზის ჩადგმული როლების საშუალებით

გ. შეიძლება მონაცემთა ბაზის ჩადგმული როლების საშუალებით

19.8.23.მონაცემთა ბაზის ყველა ცხრილიდან და წარმოდგენიდან მონაცემების წაკითხვის უფლების მისაცემად:

ა. საკმარისია მომხმარებელი ჩავრთოთ db_datareader ფიქსირებულ როლში

ბ. არასაკმარისია მომხმარებელი ჩავრთოთ db_datareader ფიქსირებულ როლში

გ. საკმარისია მომხმარებელი ჩავრთოთ db_datawriter ფიქსირებულ როლში

19.8.24.GRANT ბრძანება:

ა. არ გამოიყენება მონაცემთა ბაზის ობიექტებთან მომხმარებლის მიმართვის უფლებების მართვისთვის

ბ. გამოიყენება მონაცემთა ბაზის ობიექტებთან მომხმარებლის მიმართვის უფლებების მართვისთვის

გ. გამოიყენება მონაცემთა ბაზის ობიექტებთან მომხმარებლის მიმართვის უფლებების აკრძალვისათვის

19.8.25.დავუშვათ, გვინდა რომ SELECT და INSERT ბრძანებების გამოყენების უფლება მივცეთ Roli_1 როლს Pers_1 ცხრილის მიმართ. ამასთან, საჭიროა, რომ მომავალში ამ ჯგუფის წევრებს

ანალოგიური უფლებების სხვა მომხმარებლებისათვის გადაცემა შეეძლოთ. ამისათვის, უნდა შევასრულოთ შემდეგი ბრძანებები:

- ა. USE Shekveta; GRANT SELECT, INSERT ON Pers_1 TO Roli_1
- ბ. USE Shekveta; GRANT DELETE, UPDATE ON Pers_1 TO Roli_1 WITH GRANT OPTION
- გ. USE Shekveta; GRANT SELECT, INSERT ON Pers_1 TO Roli_1 WITH GRANT OPTION

19.8.26. USE Shekveta; GRANT SELECT ON Pers_1 TO public;

GRANT INSERT, UPDATE, DELETE ON Pers_1 TO User1, User2, Login_1 ბრძანებების შესრულების შედეგად:

- ა. ჯერ public როლს მიენიჭება SELECT უფლება, შემდეგ კი INSERT, UPDATE, DELETE უფლებები მიენიჭა User1, User2, Login_1 მომხმარებლებს. შედეგად, ამ მომხმარებლებს ექნებათ Pers_1 ცხრილთან მუშაობის ყველა უფლება
- ბ. ჯერ public როლს მიენიჭება SELECT უფლება, შემდეგ კი INSERT, UPDATE, DELETE უფლებები მიენიჭა User1, User2, Login_1 მომხმარებლებს. შედეგად, ამ მომხმარებლებს არ ექნებათ Pers_1 ცხრილთან მუშაობის ყველა უფლება
- გ. ჯერ public როლს მიენიჭება SELECT უფლება, შემდეგ კი INSERT, UPDATE, DELETE უფლებები მიენიჭა მხოლოდ User2, Login_1 მომხმარებლებს. შედეგად, ამ მომხმარებლებს ექნებათ Pers_1 ცხრილთან მუშაობის ყველა უფლება

Transact-SQL-ის ბრძანებების შესრულების უფლებები

19.9.1. Transact-SQL-ის ბრძანებების შესრულების უფლებები:

- ა. მართავს მხოლოდ მონაცემთა ბაზის შექმნის შესაძლებლობას
- ბ. მართავს მონაცემთა ბაზის შექმნის, მონაცემთა ბაზის ობიექტებისა და სარეზერვო ასლის შექმნის პროცედურის შესრულების შესაძლებლობას
- გ. არ მართავს მონაცემთა ბაზის შექმნის, მონაცემთა ბაზის ობიექტებისა და სარეზერვო ასლის შექმნის პროცედურის შესრულების შესაძლებლობას

19.9.2. თუ მომხმარებელს სურს წარმოდგენის შექმნა, მაშინ ადმინისტრატორმა მას:

- ა. უნდა მისცეს CREATE TABLE ბრძანების შესრულების უფლება
- ბ. არ უნდა მისცეს CREATE VIEW ბრძანების შესრულების უფლება
- გ. უნდა მისცეს CREATE VIEW ბრძანების შესრულების უფლება

19.9.3. თუ მომხმარებელს სურს მონაცემთა ბაზის შექმნა, მაშინ ადმინისტრატორმა მას:

- ა. უნდა მისცეს CREATE DATABASE ბრძანების შესრულების უფლება
- ბ. უნდა მისცეს CREATE VIEW ბრძანების შესრულების უფლება
- გ. უნდა მისცეს CREATE TABLE ბრძანების შესრულების უფლება

19.9.4. თუ მომხმარებელს სურს ცხრილის შექმნა, მაშინ ადმინისტრატორმა მას:

- ა. უნდა მისცეს CREATE VIEW ბრძანების შესრულების უფლება
- ბ. უნდა მისცეს CREATE TABLE ბრძანების შესრულების უფლება
- გ. უნდა მისცეს CREATE DATABASE ბრძანების შესრულების უფლება

19.9.5. თუ მომხმარებელს სურს შენახული პროცედურის შექმნა, მაშინ ადმინისტრატორმა მას:

- ა. უნდა მისცეს CREATE DATABASE ბრძანების შესრულების უფლება
- ბ. უნდა მისცეს CREATE TABLE ბრძანების შესრულების უფლება

გ. უნდა მისცეს CREATE PROCEDURE ბრძანების შესრულების უფლება

19.9.6. თუ მომხმარებელს სურს მონაცემთა ბაზის სარეზერვო ასლის შექმნა, მაშინ ადმინისტრატორმა მას:

- ა. უნდა მისცეს BACKUP DATABASE ბრძანების შესრულების უფლება
- ბ. უნდა მისცეს CREATE TABLE ბრძანების შესრულების უფლება
- გ. უნდა მისცეს CREATE DATABASE ბრძანების შესრულების უფლება

19.9.7. თუ მომხმარებელს სურს ჰქონდეს ყველა ბრძანების შესრულების უფლება, მაშინ ადმინისტრატორმა მას:

- ა. არ უნდა მისცეს ALL უფლება
- ბ. უნდა მისცეს ALL უფლება
- გ. უნდა მისცეს CREATE ALL უფლება

19.9.8. სერვერის sysadmin ფიქსირებული როლის და მონაცემთა ბაზის db_owner ფიქსირებული როლის წევრებს:

- ა. ავტომატურად აქვთ მხოლოდ CREATE DATABASE უფლება
- ბ. ავტომატურად არ აქვთ ALL უფლება
- გ. ავტომატურად აქვთ ALL უფლება

19.9.9. Transact-SQL-ის ბრძანებების შესრულების უფლების მისაცემად:

- ა. გამოიყენება GRANT ბრძანება
- ბ. არ გამოიყენება GRANT ბრძანება
- გ. გამოიყენება DENY ბრძანება

19.9.10. GRANT ბრძანების სინტაქსია:

- ა. GRANT { ALL | ბრძანება } TO
- ბ. GRANT { ALL | ბრძანება } TO უსაფრთხოების_სისტემის_ობიექტის_სახელი
- გ. GRANT TO უსაფრთხოების_სისტემის_ობიექტის_სახელი

19.9.11. GRANT CREATE TABLE TO Roli_1 ბრძანების შესრულების შედეგად:

- ა. CREATE TABLE უფლება არ მიეცემა Roli_1 როლის წევრებს
- ბ. DROP TABLE უფლება მიეცემა Roli_1 როლის წევრებს
- გ. CREATE TABLE უფლება მიეცემა Roli_1 როლის წევრებს

19.9.12. CREATE VIEW და BACKUP DATABASE უფლებების მისაცემად Roli_1 და Roli_2 როლის წევრებისთვის უნდა შევასრულოთ ბრძანებები:

- ა. GRANT CREATE VIEW, BACKUP DATABASE TO Roli_1, Roli_2
- ბ. GRANT BACKUP DATABASE TO Roli_1, Roli_2
- გ. GRANT CREATE VIEW, BACKUP DATABASE TO Roli_1

არაცხადი უფლებები

19.10.1. ზოგიერთი მოქმედების შესრულებას:

- ა. სჭირდება ცხადად უფლებების მინიჭება და მისაწვდომია ავტომატურად
- ბ. არ სჭირდება ცხადად უფლებების მინიჭება და მისაწვდომია ავტომატურად

გ. არ სჭირდება ცხადად უფლებების მინიჭება და არაა მისაწვდომი ავტომატურად

19.10.2.ზოგიერთი მოქმედების შესრულებას არ სჭირდება ცხადად უფლებების მინიჭება და მისაწვდომია ავტომატურად. ეს მოქმედებები:

- ა. შეიძლება შესრულდეს მხოლოდ სერვერის როლის წევრების მიერ
- ბ. არ შეიძლება შესრულდეს მხოლოდ სერვერის როლის წევრების მიერ ან ობიექტების მფლობელების მიერ მონაცემთა ბაზაში
- გ. შეიძლება შესრულდეს მხოლოდ სერვერის როლის წევრების მიერ ან ობიექტების მფლობელების მიერ მონაცემთა ბაზაში

19.10.3.არაცხადი უფლებები მომხმარებელს:

- ა. არ გადაეცემა უშუალოდ
- ბ. გადაეცემა უშუალოდ
- გ. სერთოდ არ გადაეცემა

19.10.4.არაცხადი უფლებები მომხმარებელს არ გადაეცემა უშუალოდ. ის მათ:

- ა. არ იღებს მხოლოდ გარკვეულ გარემოებებში
- ბ. იღებს მხოლოდ გარკვეულ გარემოებებში
- გ. იღებს მხოლოდ არაგარკვეულ გარემოებებში

19.10.5.მომხმარებელი შეიძლება გახდეს მონაცემთა ბაზის ობიექტის მფლობელი:

- ა. თუ ის თვითონ შექმნის ამ ობიექტს ან თუ სხვა მფლობელი არ გადასცემს მას თავისი ობიექტის ფლობის უფლებას
- ბ. თუ ის თვითონ არ შექმნის ამ ობიექტს ან თუ სხვა მფლობელი გადასცემს მას თავისი ობიექტის ფლობის უფლებას
- გ. თუ ის თვითონ შექმნის ამ ობიექტს ან თუ სხვა მფლობელი გადასცემს მას თავისი ობიექტის ფლობის უფლებას

19.10.6.ობიექტის მფლობელი:

- ა. ავტომატურად იღებს ამ ობიექტზე ნებისმიერი მოქმედების შესრულების უფლებას
- ბ. ავტომატურად არ იღებს ამ ობიექტზე ნებისმიერი მოქმედების შესრულების უფლებას
- გ. არაავტომატურად იღებს ამ ობიექტზე ნებისმიერი მოქმედების შესრულების უფლებას

19.10.7.ობიექტის მფლობელი:

- ა. არაავტომატურად იღებს ამ ობიექტზე ნებისმიერი მოქმედების შესრულების უფლებას, მათ შორის, სხვა მომხმარებლისთვის ამ ობიექტთან მიმართვის გადაცემის უფლებას
- ბ. ავტომატურად იღებს ამ ობიექტზე ნებისმიერი მოქმედების შესრულების უფლებას, მათ შორის, სხვა მომხმარებლისთვის ამ ობიექტთან მიმართვის გადაცემის უფლებას
- გ. ავტომატურად არ იღებს ამ ობიექტზე ნებისმიერი მოქმედების შესრულების უფლებას, მათ შორის, სხვა მომხმარებლისთვის ამ ობიექტთან მიმართვის გადაცემის უფლებას

19.10.8 მხოლოდ ობიექტის ფლობის ფაქტი მომხმარებელს საშუალებას:

- ა. აძლევს არ შეასრულოს ნებისმიერი მოქმედებები ამ ობიექტზე
- ბ. არ აძლევს შეასრულოს ნებისმიერი მოქმედებები ამ ობიექტზე
- გ. აძლევს შეასრულოს ნებისმიერი მოქმედებები ამ ობიექტზე

19.10.9.რომელი მსჯელობაა სწორი:

- ა. ჩვენ არ შეგვიძლია კონკრეტულ მომხმარებელს მივცეთ სერვერის პროცესების მართვის უფლება. ეს შესაძლებელია მომხმარებლის ჩართვით processadmin როლში
- ბ. ჩვენ შეგვიძლია კონკრეტულ მომხმარებელს მივცეთ სერვერის პროცესების მართვის უფლება. ეს შესაძლებელია მომხმარებლის ჩართვით processadmin როლში
- გ. ჩვენ არ შეგვიძლია კონკრეტულ მომხმარებელს მივცეთ სერვერის პროცესების მართვის უფლება. ეს შეუძლებელია მომხმარებლის ჩართვით processadmin როლში

19.10.10. კონკრეტული მოქმედებისთვის, რომელიც უფლებებით იმართება, არსებობს მიმართვის მდგომარეობის შემდეგი ვარიანტები:

- ა. მინიჭება და აკრძალვა
- ბ. მინიჭება, აკრძალვა და არაცხადი უარყოფა
- გ. აკრძალვა და არაცხადი უარყოფა

მიმართვის აკრძალვა

19.11.1.სერვერის უსაფრთხოების სისტემას:

- ა. აქვს არაიერარქიული სტრუქტურა
- ბ. არ აქვს იერარქიული სტრუქტურა
- გ. აქვს იერარქიული სტრუქტურა

19.11.2.რომელი მსჯელობაა სწორი:

- ა. სერვერის უსაფრთხოების სისტემას აქვს იერარქიული სტრუქტურა. ეს მონაცემთა ბაზის როლებს საშუალებას აძლევს შეიცავდეს Windows NT-ის საადრიცხვო ჩანაწერებსა და ჯგუფებს, სერვერის მომხმარებლებსა და როლებს. მომხმარებელი, თავის მხრივ, შეიძლება რამდენიმე როლში მონაწილეობდეს
- ბ. სერვერის უსაფრთხოების სისტემას არ აქვს იერარქიული სტრუქტურა. ეს მონაცემთა ბაზის როლებს საშუალებას არ აძლევს შეიცავდეს Windows NT-ის საადრიცხვო ჩანაწერებსა და ჯგუფებს, სერვერის მომხმარებლებსა და როლებს. მომხმარებელი, თავის მხრივ, შეიძლება რამდენიმე როლში მონაწილეობდეს
- გ. სერვერის უსაფრთხოების სისტემას აქვს იერარქიული სტრუქტურა. ეს მონაცემთა ბაზის როლებს საშუალებას არ აძლევს შეიცავდეს Windows NT-ის საადრიცხვო ჩანაწერებსა და ჯგუფებს, სერვერის მომხმარებლებსა და როლებს. მომხმარებელი, თავის მხრივ, არ შეიძლება რამდენიმე როლში მონაწილეობდეს

19.11.3.უსაფრთხოების სისტემის იერარქიული სტრუქტურის შედეგია ის, რომ ერთ მომხმარებელს ერთდროულად:

- ა. არ შეიძლება ჰქონდეს მიმართვის სხვადასხვა უფლებები სხვადასხვა როლებში
- ბ. შეიძლება ჰქონდეს მიმართვის სხვადასხვა უფლებები სხვადასხვა როლებში
- გ. უნდა ჰქონდეს მიმართვის ერთნაირი უფლებები სხვადასხვა როლებში

19.11.4.თუ ერთ-ერთ როლს, რომელშიც მომხმარებელი იმყოფება:

- ა. აქვს მონაცემებთან მიმართვის უფლება, მაშინ მომხმარებელსაც ავტომატურად არ ექნება იგივე უფლებები
- ბ. არ აქვს მონაცემებთან მიმართვის უფლება, მაშინ მომხმარებელსაც ავტომატურად ექნება იგივე უფლებები

გ. აქვს მონაცემებთან მიმართვის უფლება, მაშინ მომხმარებელსაც ავტომატურად ექნება იგივე უფლებები

19.11.5. როცა მომხმარებელს ან Transact_SQL-ის ბრძანებებს ვუკრძალავთ მონაცემებთან მიმართვას (deny access), ამით:

- ა. ანულირდება მიმართვის ყველა უფლება, რომელიც მომხმარებელმა მიიღო იერარქიის ნებისმიერ დონეზე
- ბ. არ ანულირდება მიმართვის ყველა უფლება, რომელიც მომხმარებელმა მიიღო იერარქიის ნებისმიერ დონეზე
- გ. ანულირდება მიმართვის ყველა უფლება, რომელიც მომხმარებელმა არ მიიღო იერარქიის ნებისმიერ დონეზე

19.11.6. როცა მომხმარებელს ან Transact_SQL-ის ბრძანებებს ვუკრძალავთ მონაცემებთან მიმართვას (deny access), ამით ანულირდება მიმართვის ყველა უფლება, რომელიც მომხმარებელმა მიიღო იერარქიის ნებისმიერ დონეზე. ამასთან, მიმართვა:

- ა. არ დარჩება აკრძალული, მიუხედავად უფრო მაღალ დონეზე გაცემული ნებართვებისა
- ბ. დარჩება აკრძალული, მიუხედავად უფრო მაღალ დონეზე გაცემული ნებართვებისა
- გ. დარჩება ნებადართული, მიუხედავად უფრო მაღალ დონეზე გაცემული ნებართვებისა

19.11.7. DENY ბრძანება:

- ა. გამოიყენება მომხმარებლისათვის მონაცემთა ბაზის ობიექტთან მიმართვის უფლების მისაცემად
- ბ. არ გამოიყენება მომხმარებლისათვის მონაცემთა ბაზის ობიექტთან მიმართვის აკრძალვისათვის
- გ. გამოიყენება მომხმარებლისათვის მონაცემთა ბაზის ობიექტთან მიმართვის აკრძალვისათვის

19.11.8. DENY CREATE TABLE TO User_3 CASCADE ბრძანების შესრულების შედეგად:

- ა. User_3 მომხმარებელს ჩამორთმევა ცხრილის შექმნის უფლება
- ბ. User_3 მომხმარებელს მიეცემა ცხრილის შექმნის უფლება
- გ. User_1 მომხმარებელს ჩამორთმევა ცხრილის შექმნის უფლება

19.11.9. იმისათვის, რომ User_1 და User_2 მომხმარებლებს ავუკრძალოთ CREATE DATABASE და CREATE TABLE ბრძანებების შესრულება, უნდა შევასრულოთ ბრძანება:

- ა. DENY CREATE DATABASE, CREATE TABLE TO User_1
- ბ. DENY CREATE DATABASE, CREATE TABLE TO User_1, User_2
- გ. DENY CREATE DATABASE, CREATE TABLE TO User_2

მიმართვის არაცხადი უარყოფა

19.12.1. არაცხადი უარყოფა მიმართვის აკრძალვის:

- ა. მსგავსი არაა, იმ განსხვავებით, რომ ის მოქმედებს მხოლოდ იმ დონეზე, რომელზეც განსაზღვრულია
- ბ. მსგავსია, იმ განსხვავებით, რომ ის არ მოქმედებს მხოლოდ იმ დონეზე, რომელზეც განსაზღვრულია

გ. მსგავსია, იმ განსხვავებით, რომ ის მოქმედებს მხოლოდ იმ დონეზე, რომელზეც განსაზღვრულია

19.12.2.თუ მომხმარებლისთვის განსაზღვრულ დონეზე უარყოფილია მიმართვა, მან ის:

ა. მაინც შეიძლება მიიღოს იერარქიის სხვა დონეზე როლში წევრობის გზით, რომელსაც აქვს მიმართვის უფლება

ბ. არ შეიძლება მიიღოს იერარქიის სხვა დონეზე როლში წევრობის გზით, რომელსაც აქვს მიმართვის უფლება

გ. მაინც შეიძლება მიიღოს იერარქიის სხვა დონეზე როლში წევრობის გზით, რომელსაც არ აქვს მიმართვის უფლება

19.12.3.რომელი მსჯელობაა სწორი:

ა. არაავტომატურად, მონაცემებთან მომხმარებლის მიმართვა არაცხადად უარყოფილია

ბ. ავტომატურად, მონაცემებთან მომხმარებლის მიმართვა არაცხადად უარყოფილია

გ. ავტომატურად, მონაცემებთან მომხმარებლის მიმართვა ცხადად უარყოფილია

19.12.4.REVOKE ბრძანება გამოიყენება მონაცემთა ბაზის ობიექტებთან მიმართვის:

ა. ნახევრად ცხადი უარყოფისათვის

ბ. ცხადი უარყოფისათვის

გ. არაცხადი უარყოფისათვის

19.12.5.Transact_SQL-ის ბრძანების შესრულების უფლების არაცხადი უარყოფისათვის გამოიყენება REVOKE ბრძანება. მისი სინტაქსია:

ა. REVOKE { ALL | ბრძანება [,...n] } FROM უსაფრთხოების_სისტემის_ობიექტის_სახელი

ბ. REVOKE { ALL | ბრძანება [,...n] } FROM

გ. REVOKE { ALL | ბრძანება [,...n] }

19.12.6.CREATE TABLE უფლების არაცხადი უარყოფისთვის, რომელიც მინიჭებული ჰქონდათ Roli_6 და Roli_5 როლის წევრებს, უნდა შევასრულოთ ბრძანება:

ა. REVOKE CREATE TABLE FROM Roli_6

ბ. REVOKE CREATE TABLE FROM Roli_6, Roli_5

გ. REVOKE DROP TABLE FROM Roli_6, Roli_5

19.12.7.CREATE TABLE და CREATE DEFAULT უფლების არაცხადი უარყოფისთვის, რომელიც მინიჭებული ჰქონდათ User_6 და User_5 მომხმარებლებს, უნდა შევასრულოთ ბრძანება:

ა. REVOKE DROP TABLE, CREATE DEFAULT FROM User_6, User_5

ბ. REVOKE CREATE TABLE, CREATE DEFAULT FROM User_6

გ. REVOKE CREATE TABLE, CREATE DEFAULT FROM User_6, User_5

ღანართი. ტესტების პასუხები

თაზი 1. მონაცემთა ბაზების თეორიული საფუძვლები

საინფორმაციო სისტემები

1.1.1.	ა	1.1.2.	ბ	1.1.3.	გ	1.1.4.	ბ	1.1.5.	ბ
1.1.6.	ა								

მონაცემთა ბაზები

1.2.1.	ა	1.2.2.	ბ	1.2.3.	გ	1.2.4.	ა	1.2.5.	გ
1.2.6.	ა	1.2.7.	ბ	1.2.8.	ბ	1.2.9.	ა	1.2.10.	გ
1.2.11	გ	1.2.12.	ბ	1.2.13	ა	1.2.14.	ბ	1.2.15.	გ

მონაცემთა ბაზების მოდელები

1.3.1.	ა	1.3.2.	გ	1.3.3.	ბ	1.3.4.	ბ	1.3.5.	ა
1.3.6.	ა	1.3.7.	გ	1.3.8.	ბ	1.3.9.	გ	1.3.10.	ა
1.3.11.	ბ	1.3.12.	გ	1.3.13.	ა	1.3.14.	გ	1.3.15.	ბ
1.3.16.	გ	1.3.17.	ა	1.3.18.	ა	1.3.19.	ბ	1.3.20.	გ
1.3.21.	გ	1.3.22.	ა						

მონაცემთა რელაციური ბაზები

1.4.1.	ა	1.4.2.	ბ	1.4.3.	ბ	1.4.4.	გ	1.4.5.	გ
--------	---	--------	---	--------	---	--------	---	--------	---

ცხრილების ნორმალიზება

1.5.1.	გ	1.5.2.	ა	1.5.3.	ბ	1.5.4.	ა	1.5.5.	ბ
1.5.6.	ა	1.5.7.	გ	1.5.8.	გ	1.5.9.	ბ	1.5.10.	ა
1.5.11	გ								

ცხრილებს შორის არსებული კავშირები

1.6.1.	ბ	1.6.2.	ა	1.6.3.	ბ	1.6.4.	ა	1.6.5.	გ
1.6.6.	ბ	1.6.7.	ა	1.6.8.	გ	1.6.9.	ა	1.6.10.	ბ
1.6.11.	გ	1.6.12	გ	1.6.13.	გ	1.6.14.	ა	1.6.15	ბ
1.6.16.	ა	1.6.17.	გ	1.6.18.	გ	1.6.19.	ბ	1.6.20.	გ
1.6.21.	ა	1.6.22.	გ	1.6.23.	ბ	1.6.24.	ა	1.6.25.	გ
1.6.26.	ა	1.6.27.	ბ						

თაზი 2. მონაცემთა ბაზები

მონაცემთა ბაზების არქიტექტურა და ობიექტები

2.1.1.	გ	2.1.2.	ბ	2.1.3.	ა	2.1.4.	ბ	2.1.5.	ა
2.1.6.	ა	2.1.7.	ბ	2.1.8.	გ	2.1.9.	გ	2.1.10.	ბ
2.1.11.	ა	2.1.12.	გ	2.1.13.	ბ	2.1.14.	ა	2.1.15	გ
2.1.16	ა	2.1.17.	ბ	2.1.18.	გ	2.1.19	ა	2.1.20.	ბ
2.1.21.	ბ	2.1.22.	ა	2.1.23.	გ	2.1.24.	ა	2.1.25.	გ
2.1.26.	გ	2.1.27	ა	2.1.28	გ	2.1.29	გ	2.1.30	გ

2.1.31	ა	2.1.32	ბ	2.1.33	ბ	2.1.34	ა	2.1.35	ა
2.1.36	ბ	2.1.37	ბ	2.1.38	ა	2.1.39	გ	2.1.40	ბ
2.1.41	ა	2.1.42	ბ	2.1.43	გ	2.1.44	ა	2.1.45	ბ
2.1.46	ა	2.1.47	ბ	2.1.48	გ	2.1.49	ა	2.1.50	ბ
2.1.51	ბ	2.1.52	ა	2.1.53	გ	2.1.54	ბ	2.1.55	ბ
2.1.56	ა	2.1.57	გ	2.1.58	ა	2.1.59	ა	2.1.60	გ
2.1.61	გ	2.1.62	ბ	2.1.63	ბ	2.1.64	ა	2.1.65	ა
2.1.66	გ	2.1.67	ბ	2.1.68	ა	2.1.69	გ	2.1.70	ა
2.1.71	ა	2.1.72	გ	2.1.73	ბ	2.1.74	გ	2.1.75	ბ
2.1.76	ა	2.1.77	ა	2.1.78	ბ	2.1.79	გ	2.1.80	ა
2.1.81	ა	2.1.82	ბ	2.1.83	გ	2.1.84	ა	2.1.85	ბ
2.1.86	გ	2.1.87	ა	2.1.88	ბ	2.1.89	გ	2.1.90	ა

სისტემური მონაცემთა ბაზები. ფაილები და ფაილების ჯგუფები

2.2.1	ა	2.2.2	ბ	2.2.3	გ	2.2.4	ბ	2.2.5	ა
2.2.6	გ	2.2.7	ბ	2.2.8	ა	2.2.9	გ	2.2.10	ა
2.2.11	ა	2.2.12	ბ	2.2.13	ბ	2.2.14	ბ	2.2.15	გ
2.2.16	ბ	2.2.17	ა	2.2.18	ა	2.2.19	გ	2.2.20	ა
2.2.21	ა	2.2.22	ბ	2.2.23	გ	2.2.24	ბ	2.2.25	ა
2.2.26	ბ	2.2.27	ა	2.2.28	გ	2.2.29	ა	2.2.30	ბ

მონაცემთა ბაზების შექმნა, წაშლა და სახელის შეცვლა

2.3.1	ბ	2.3.2	ა	2.3.3	გ	2.3.4	ა	2.3.5	ა
2.3.6	გ	2.3.7	გ	2.3.8	ბ	2.3.9	გ	2.3.10	ა
2.3.11	ბ	2.3.12	გ	2.3.13	ბ	2.3.14	გ	2.3.15	ა
2.3.16	ა	2.3.17	ბ	2.3.18	გ	2.3.19	ბ	2.3.20	ა
2.3.21	გ	2.3.22	ბ	2.3.23	ა	2.3.24	ა	2.3.25	გ
2.3.26	ა	2.3.27	ბ	2.3.28	გ	2.3.29	ბ	2.3.30	ა
2.3.31	ბ	2.3.32	გ	2.3.33	ა	2.3.34	ბ	2.3.35	გ
2.3.36	ა	2.3.37	ბ	2.3.38	გ	2.3.39	ბ	2.3.40	ა
2.3.41	ა	2.3.42	ა	2.3.43	გ	2.3.44	ბ	2.3.45	ა
2.3.46	ა	2.3.47	ბ	2.3.48	გ	2.3.49	ა	2.3.50	ბ
2.3.51	გ								

მონაცემთა ბაზის მიერთება და გამორთვა. ინფორმაციის მიღება მონაცემთა ბაზების შესახებ

2.4.1	ბ	2.4.2	გ	2.4.3	ბ	2.4.4	გ	2.4.5	ა
2.4.6	ბ	2.4.7	ბ	2.4.8	გ	2.4.9	ბ	2.4.10	ა
2.4.11	გ	2.4.12	გ	2.4.13	ა	2.4.14	გ	2.4.15	ა
2.4.16	ა	2.4.17	ბ	2.4.18	ბ	2.4.19	გ		

მონაცემთა მომხმარებლის ტიპის შექმნა

2.5.1	გ	2.5.2	გ	2.5.3	ბ	2.5.4	ა	2.5.5	ა
2.5.6	ბ	2.5.7	გ	2.5.8	ბ	2.5.9	გ	2.5.10	ა
2.5.11	ა	2.5.12	ბ	2.5.13	გ	2.5.14	გ	2.5.15	ბ

2.5.16	გ	2.5.17	გ	2.5.18	ა	2.5.19	ა		
--------	---	--------	---	--------	---	--------	---	--	--

თავი 3. ცხრილები

ცხრილების დაპროექტება

3.1.1	ა	3.1.2	ბ	3.1.3	ა	3.1.4	ბ	3.1.5	გ
3.1.6	ბ	3.1.7	გ	3.1.8	ა	3.1.9	ა	3.1.10	ბ
3.1.11	გ	3.1.12	ბ	3.1.13	ბ	3.1.14	ა	3.1.15	გ
3.1.16	ბ	3.1.17	გ	3.1.18	ა	3.1.19	ბ	3.1.20	გ
3.1.21	ა	3.1.22	გ	3.1.23	ბ	3.1.24	ა	3.1.25	გ
3.1.26	ბ	3.1.27	ბ	3.1.28	ა	3.1.29	გ	3.1.30	ა
3.1.31	გ	3.1.32	ა	3.1.33	ბ	3.1.34	ბ	3.1.35	გ
3.1.36	ა	3.1.37	ა	3.1.38	ბ	3.1.39	გ	3.1.40	ა
3.1.41	ბ	3.1.42	გ	3.1.43	ბ	3.1.44	ა	3.1.45	ა
3.1.46	გ								

ცხრილის შექმნა. ცხრილის წაშლა

3.2.1	ა	3.2.2	გ	3.2.3	ბ	3.2.4	ა	3.2.5	ბ
3.2.6	ა	3.2.7	გ	3.2.8	გ	3.2.9	ა	3.2.10	ბ
3.2.11	გ	3.2.12	ბ	3.2.13	ა	3.2.14	ა	3.2.15	ბ
3.2.16	გ	3.2.17	გ	3.2.18	ა	3.2.19	ბ	3.2.20	გ
3.2.21	ბ	3.2.22	ა	3.2.23	გ	3.2.24	ბ	3.2.25	ა
3.2.26	გ	3.2.27	ა	3.2.28	გ	3.2.29	ბ	3.2.30	ბ
3.2.31	ა	3.2.32	გ	3.2.33	ბ	3.2.34	გ	3.2.35	ბ
3.2.36	ბ	3.2.37	ა	3.2.38	ბ	3.2.39	ბ	3.2.40	გ
3.2.41	გ	3.2.42	ა	3.2.43	ბ	3.2.44	გ	3.2.45	ბ
3.2.46	ა	3.2.47	გ	3.2.48	ბ	3.2.49	გ	3.2.50	ა
3.2.51	ბ	3.2.52	გ	3.2.53	ა	3.2.54	გ	3.2.55	ბ
3.2.56	ა	3.2.57	გ	3.2.58	გ	3.2.59	ბ	3.2.60	ა
3.2.61	ბ	3.2.62	გ	3.2.63	ა	3.2.64	ა	3.2.65	ბ
3.2.66	გ	3.2.67	ა	3.2.68	ბ	3.2.69	გ	3.2.70	ა
3.2.71	ბ	3.2.72	გ	3.2.73	ა	3.2.74	ბ	3.2.75	გ
3.2.76	ა	3.2.77	ბ	3.2.78	გ	3.2.79	ა	3.2.80	ბ
3.2.81	გ	3.2.82	ა	3.2.83	ა	3.2.84	ბ	3.2.85	გ
3.2.86	ა								

ცხრილისათვის სახელის შეცვლა. ინფორმაციის მიღება ცხრილების შესახებ. დიაგრამები

3.3.1	ა	3.3.2	ბ	3.3.3	გ	3.3.4	ბ	3.3.5	ა
3.3.6	ბ	3.3.7	გ	3.3.8	ა	3.3.9	ბ	3.3.10	გ
3.3.11	ბ	3.3.12	ა	3.3.13	ბ	3.3.14	გ	3.3.15	ა
3.3.16	ბ	3.3.17	ა	3.3.18	ბ	3.3.19	გ	3.3.20	ა
3.3.21	ბ	3.3.22	გ	3.3.23	ა	3.3.24	გ	3.3.25	ა
3.3.26	ბ	3.3.27	გ						

თავი 4. მონაცემების მართვა

Transact SQL-ის საფუძვლები

4.1.1	ა	4.1.2	ბ	4.1.3	გ	4.1.4	ბ	4.1.5	ა
4.1.6	გ	4.1.7	ბ	4.1.8	გ	4.1.9	გ	4.1.10	ბ
4.1.11	ა	4.1.12	ბ	4.1.13	ა	4.1.14	ბ	4.1.15	გ
4.1.16	ა	4.1.17	გ	4.1.18	ა	4.1.19	ბ	4.1.20	გ
4.1.21	ა	4.1.22	გ	4.1.23	ბ	4.1.24	ბ	4.1.25	გ
4.1.26	ა	4.1.27	ა	4.1.28	გ	4.1.29	ბ	4.1.30	ა
4.1.31	ა	4.1.32	ა	4.1.33	გ	4.1.34	გ	4.1.35	ბ
4.1.36	გ	4.1.37	ა	4.1.38	ბ	4.1.39	გ	4.1.40	ბ
4.1.41	ა	4.1.42	გ	4.1.43	ბ	4.1.44	ა	4.1.45	გ
4.1.46	ბ	4.1.47	გ	4.1.48	ბ	4.1.49	გ	4.1.50	ა
4.1.51	გ	4.1.52	ბ	4.1.53	გ	4.1.54	ბ	4.1.55	ა
4.1.56	ა	4.1.57	ბ	4.1.58	გ	4.1.59	გ	4.1.60	ა
4.1.61	ა	4.1.62	ბ	4.1.63	გ	4.1.64	გ	4.1.65	ა
4.1.66	ბ	4.1.67	გ	4.1.68	ა	4.1.69	გ	4.1.70	ბ
4.1.71	ა	4.1.72	გ	4.1.73	ა	4.1.74	ბ	4.1.75	გ
4.1.76	ა	4.1.77	ბ	4.1.78	გ	4.1.79	ა	4.1.80	ბ
4.1.81	გ	4.1.82	ა	4.1.83	ბ	4.1.84	გ		

ცხრილში მონაცემების ჩამატება, შეცვლა და წაშლა

4.2.1	ა	4.2.2	ბ	4.2.3	გ	4.2.4	ბ	4.2.5	გ
4.2.6	ბ	4.2.7	ბ	4.2.8	ა	4.2.9	გ	4.2.10	გ
4.2.11	გ	4.2.12	გ	4.2.13	გ	4.2.14	გ	4.2.15	გ
4.2.16	გ	4.2.17	გ	4.2.18	ა	4.2.19	ა	4.2.20	გ
4.2.21	ბ	4.2.22	ა	4.2.23	ბ	4.2.24	გ	4.2.25	ა
4.2.26	ბ	4.2.27	ა	4.2.28	ბ	4.2.29	ბ	4.2.30	ა
4.2.31	გ	4.2.32	ბ	4.2.33	გ	4.2.34	ა	4.2.35	ა
4.2.36	ბ	4.2.37	გ	4.2.38	გ	4.2.39	ა	4.2.40	ბ
4.2.41	გ	4.2.42	ა	4.2.43	ბ	4.2.44	გ	4.2.45	ა
4.2.46	ბ	4.2.47	გ	4.2.48	ა	4.2.49	ბ	4.2.50	გ
4.2.51	ა	4.2.52	ბ	4.2.53	გ	4.2.54	ა	4.2.55	ბ
4.2.56	ბ	4.2.57	გ						

ცხრილიდან მონაცემების ამორჩევა. SELECT განყოფილება

4.3.1	ა	4.3.2	ბ	4.3.3	გ	4.3.4	ა	4.3.5	ბ
4.3.6	ბ	4.3.7	ა	4.3.8	გ	4.3.9	გ	4.3.10	გ
4.3.11	გ	4.3.12	ა	4.3.13	ბ	4.3.14	ბ	4.3.15	გ
4.3.16	ა	4.3.17	ბ	4.3.18	გ	4.3.19	ა	4.3.20	ბ
4.3.21	გ	4.3.22	ა	4.3.23	ბ	4.3.24	გ	4.3.25	ა
4.3.26	ა	4.3.27	ა	4.3.28	ბ	4.3.29	გ	4.3.30	ა
4.3.31	ა								

WHERE განყოფილება. ლოგიკის ოპერატორები

4.4.1	ა	4.4.2	ბ	4.4.3	გ	4.4.4	ა	4.4.5	ბ
4.4.6	ა	4.4.7	ბ	4.4.8	ა	4.4.9	გ	4.4.10	ა
4.4.11	გ	4.4.12	ბ	4.4.13	ა	4.4.14	ა	4.4.15	ა

4.4.16	ბ	4.4.17	გ	4.4.18	ს	4.4.19	ბ	4.4.20	გ
4.4.21	ს	4.4.22	ბ	4.4.23	გ	4.4.24	ს	4.4.25	ბ
4.4.26	გ	4.4.27	ს	4.4.28	ბ	4.4.29	გ	4.4.30	ს
4.4.31	ბ								

LIKE ოპერატორი

4.5.1	ბ	4.5.2	ბ	4.5.3	ს	4.5.4	გ	4.5.5	ბ
4.5.6	ს	4.5.7	ს	4.5.8	ბ	4.5.9	გ	4.5.10	ს
4.5.11	ბ	4.5.12	გ	4.5.13	ს	4.5.14	ბ	4.5.15	გ
4.5.16	ს	4.5.17	ბ	4.5.18	გ	4.5.19	გ		

აგრეგირების ფუნქციები

4.6.1	ს	4.6.2	გ	4.6.3	გ	4.6.4	გ	4.6.5	გ
4.6.6	ს	4.6.7	ბ	4.6.8	ს	4.6.9	გ	4.6.10	ბ
4.6.11	ბ	4.6.12	ს	4.6.13	ბ	4.6.14	ბ	4.6.15	გ
4.6.16	ს	4.6.17	ბ	4.6.18	გ	4.6.19	ს	4.6.20	გ
4.6.21	ბ	4.6.22	გ						

GROUP BY განყოფილება

4.7.1	ს	4.7.2	გ	4.7.3	ბ	4.7.4	ს	4.7.5	ბ
4.7.6	გ	4.7.7	ს	4.7.8	ბ				

HAVING განყოფილება

4.8.1	გ	4.8.2	ს	4.8.3	ბ	4.8.4	გ	4.8.5	ს
4.8.6	ბ	4.8.7	გ						

ORDER BY განყოფილება

4.9.1	ბ	4.9.2	ს	4.9.3	გ	4.9.4	გ	4.9.5	ს
4.9.6	ბ	4.9.7	გ	4.9.8	ს	4.9.9	ბ	4.9.10	გ
4.9.11	გ								

COMPUTE განყოფილება

4.10.1	ს	4.10.2	ბ	4.10.3	გ	4.10.4	ს	4.10.5	გ
4.10.6	ს	4.10.7	ბ	4.10.8	გ	4.10.9	ს	4.10.10	ბ

NULL მნიშვნელობასთან მუშაობა

4.11.1	ს	4.11.2	ბ	4.11.3	გ	4.11.4	ს	4.11.5	ბ
4.11.6	გ	4.11.7	გ	4.11.8	ს	4.11.9	ბ	4.11.10	გ
4.11.11	ბ								

მონაცემების მასობრივი გადაწერა

4.12.1	ს	4.12.2	ბ	4.12.3	ბ	4.12.4	გ	4.12.5	ს
4.12.6	ბ	4.12.7	გ	4.12.8	ბ	4.12.9	გ	4.12.10	ს

4.12.11	ბ	4.12.12	გ	4.12.13	გ	4.12.14	ბ	4.12.15	გ
---------	---	---------	---	---------	---	---------	---	---------	---

თაზი 5. ზმართობები

FROM განყოფილება

5.1.1	ა	5.1.2	ბ	5.1.3	გ	5.1.4	ა	5.1.5	ბ
5.1.6	ა	5.1.7	გ	5.1.8	ა	5.1.9	გ	5.1.10	ბ
5.1.11	გ	5.1.12	ა	5.1.13	ბ	5.1.14	გ	5.1.15	ა
5.1.16	გ	5.1.17	ა	5.1.18	გ	5.1.19	ა	5.1.20	ბ
5.1.21	ბ								

ჯვარედინი შეერთებები

5.2.1	ა	5.2.2	ბ	5.2.3	გ	5.2.4	ა	5.2.5	ბ
5.2.6	გ	5.2.7	ა	5.2.8	ბ	5.2.9	გ		

შიდა შეერთებები. შიდა შეერთების სახეები. შედგენილი შეერთებები. შეერთებები უტოლობის შემთხვევაში. მრავალცხრილიანი შეერთებები

5.3.1	ა	5.3.2	ბ	5.3.3	გ	5.3.4	ა	5.3.5	ბ
5.3.6	ა	5.3.7	გ	5.3.8	ა	5.3.9	ბ	5.3.10	გ
5.3.11	ა	5.3.12	ბ	5.3.13	გ				

გარე შეერთებები

5.4.1	ა	5.4.2	ბ	5.4.3	გ	5.4.4	გ	5.4.5	ბ
5.4.6	ა	5.4.7	ბ	5.4.8	ა	5.4.9	გ	5.4.10	ბ

თაზი 6. მმართველი კონსტრუქციები

BEGIN...END

6.1.1	ა	6.1.2	ბ	6.1.3	გ	6.1.4	გ	6.1.5	ა
6.1.6	ბ	6.1.7	გ	6.1.8	ბ	6.1.9	ბ		

IF...ELSE

6.2.1	გ	6.2.2	ა	6.2.3	ბ	6.2.4	გ	6.2.5	ა
-------	---	-------	---	-------	---	-------	---	-------	---

CASE

6.3.1	ბ	6.3.2	ა	6.3.3	გ	6.3.4	ა	6.3.5	ბ
-------	---	-------	---	-------	---	-------	---	-------	---

COALESCE

6.4.1	გ	6.4.2	ა						
-------	---	-------	---	--	--	--	--	--	--

WHILE...BREAK & CONTINUE

6.5.1	ბ	6.5.2	ა	6.5.3	გ	6.5.4	ა	6.5.5	ბ
6.5.6	ა	6.5.7	ბ	6.5.8	ბ				

თავი 7. ქვემოთხოვნები

დამოუკიდებელი სკალარული ქვემოთხოვნები. დამოუკიდებელი ქვემოთხოვნები მრავლობითი მნიშვნელობებით

7.1.1	გ	7.1.2	ა	7.1.3	ბ	7.1.4	გ	7.1.5	ა
7.1.6	ბ	7.1.7	გ	7.1.8	გ	7.1.9	ბ	7.1.10	ა
7.1.11	ა	7.1.12	ბ	7.1.13	გ	7.1.14	გ	7.1.15	ა
7.1.16	ბ	7.1.17	ა	7.1.18	ა	7.1.19	ბ	7.1.20	ბ

ბმული ქვემოთხოვნები

7.2.1	ბ	7.2.2	ა	7.2.3	ბ	7.2.4	გ	7.2.5	გ
-------	---	-------	---	-------	---	-------	---	-------	---

ლოგიკის ოპერატორები. ALL და ANY ოპერატორები

7.3.1	ა	7.3.2	ბ	7.3.3	ბ	7.3.4	გ	7.3.5	ა
7.3.6	ბ	7.3.7	ა	7.3.8	ბ	7.3.9	ა	7.3.10	ბ

EXISTS ოპერატორი

7.4.1	გ	7.4.2	ბ	7.4.3	გ	7.4.4	გ		
-------	---	-------	---	-------	---	-------	---	--	--

თავი 8. ოპერაციები სიმრავლეებზე

შესავალი. UNION ოპერაცია

8.1.1	ა	8.1.2	ბ	8.1.3	გ	8.1.4	ა	8.1.5	ა
8.1.6	გ	8.1.7	ბ	8.1.8	ბ	8.1.9	გ	8.1.10	ა
8.1.11	ბ	8.1.12	გ	8.1.13	გ	8.1.14	ბ	8.1.15	ა
8.1.16	ბ	8.1.17	ბ	8.1.18	ა	8.1.19	გ	8.1.20	ბ
8.1.21	ა	8.1.22	გ	8.1.23	ა	8.1.24	ბ	8.1.25	გ
8.1.26	გ								

INTERSECT ოპერაცია

8.2.1	ა	8.2.2	ა	8.2.3	ბ	8.2.4	გ	8.2.5	გ
8.2.6	ა	8.2.7	ბ						

EXCEPT ოპერაცია

8.3.1	ბ	8.3.2	ბ	8.3.3	ა	8.3.4	გ	8.3.5	ა
8.3.6	ბ	8.3.7	გ						

პრიორიტეტი

8.4.1	ბ	8.4.2	გ	8.4.3	გ	8.4.4	ა		
-------	---	-------	---	-------	---	-------	---	--	--

თავი 9. ფუნქციები

მომხმარებლის ფუნქციები

9.1.1	ა	9.1.2	ბ	9.1.3	გ	9.1.4	ბ	9.1.5	ა
9.1.6	გ	9.1.7	ბ	9.1.8	ბ	9.1.9	ა	9.1.10	ბ
9.1.11	ა	9.1.12	გ	9.1.13	ბ	9.1.14	ა		

Scalar ტიპის ფუნქციები

9.2.1	გ	9.2.2	ბ	9.2.3	ბ	9.2.4	ა	9.2.5	ბ
9.2.6	ბ	9.2.7	ბ	9.2.8	ა	9.2.9	გ	9.2.10	ბ
9.2.11	ა	9.2.12	ბ	9.2.13	გ	9.2.14	ა	9.2.15	ბ

Inline ტიპის ფუნქციები

9.3.1	ა	9.3.2	გ	9.3.3	გ	9.3.4	ბ	9.3.5	გ
9.3.6	ა								

Multi-statement ფუნქციები

9.4.1	ა	9.4.2	ბ	9.4.3	ა	9.4.4	ბ	9.4.5	გ
9.4.6	გ	9.4.7	ბ						

ფუნქციის წაშლა

9.5.1	ა	9.5.2	გ	9.5.3	ბ	9.5.4	ბ		
-------	---	-------	---	-------	---	-------	---	--	--

ჩადგმული ფუნქციები. მათემატიკის ფუნქციები

9.6.1	ა	9.6.2	გ	9.6.3	ბ	9.6.4	ბ	9.6.5	ა
9.6.6	ბ	9.6.7	გ	9.6.8	ა	9.6.9	ა	9.6.10	ბ
9.6.11	ა	9.6.12	გ	9.6.13	გ	9.6.14	ბ	9.6.15	ბ
9.6.16	ა	9.6.17	გ	9.6.18	გ	9.6.19	ბ	9.6.20	ა
9.6.21	ბ	9.6.22	გ	9.6.23	ა	9.6.24	ბ	9.6.25	ბ

სტრიქონებთან სამუშაო ფუნქციები

9.7.1	ა	9.7.2	ა	9.7.3	ბ	9.7.4	ბ	9.7.5	გ
9.7.6	გ	9.7.7	გ	9.7.8	ა	9.7.9	ბ	9.7.10	გ
9.7.11	ა	9.7.12	გ	9.7.13	ბ	9.7.14	ბ	9.7.15	ა
9.7.16	გ	9.7.17	გ	9.7.18	ბ	9.7.19	ბ	9.7.20	გ
9.7.21	ა	9.7.22	ბ	9.7.23	გ	9.7.24	ა	9.7.25	ბ
9.7.26	გ	9.7.27	ა	9.7.28	ბ				

თარიღებთან სამუშაო ფუნქციები

9.8.1	ა	9.8.2	ბ	9.8.3.	ა	9.8.4	ბ	9.8.5	გ
9.8.6	ა	9.8.7	ბ	9.8.8	ბ	9.8.9	გ	9.8.10	ა
9.8.11	ბ	9.8.12	გ						

თავი 10. შენახული პროცედურები

შესავალი. შენახული პროცედურების ტიპები

10.1.1	ა	10.1.2	ბ	10.1.3	გ	10.1.4	ბ	10.1.5	ა
10.1.6	ბ	10.1.7	ბ	10.1.8	გ	10.1.9	ბ	10.1.10	ა
10.1.11	ბ	10.1.12	გ	10.1.13	ა	10.1.14	ბ	10.1.15	გ
10.1.16	ბ	10.1.17	ა	10.1.18	გ	10.1.19	ბ	10.1.20	ა
10.1.21	გ	10.1.22	ა	10.1.23	ბ	10.1.24	გ		

შენახული პროცედურის შექმნა

10.2.1	ა	10.2.2	ბ	10.2.3	გ	10.2.4	ა	10.2.5	გ
10.2.6	ბ	10.2.7	გ	10.2.8	ბ	10.2.9	ა	10.2.10	გ
10.2.11	ბ	10.2.12	ა	10.2.13	გ	10.2.14	გ	10.2.15	ა
10.2.16	ბ	10.2.17	ბ	10.2.18	ა	10.2.19	ბ	10.2.20	გ
10.2.21	ა	10.2.22	გ	10.2.23	ბ	10.2.24	ა	10.2.25	გ
10.2.26	ბ	10.2.27	ბ	10.2.28	გ	10.2.29	გ	10.2.30	ბ
10.2.31	ა	10.2.32	გ	10.2.33	ბ	10.2.34	ა	10.2.35	ა
10.2.36	ა	10.2.37	ა	10.2.38	ა	10.2.39	ბ	10.2.40	გ
10.2.41	ა								

შენახული პროცედურების მართვა. შენახული პროცედურის შესახებ ინფორმაციის მიღება. შენახული პროცედურის სახელის შეცვლა. შენახული პროცედურის წაშლა. შენახული პროცედურის ავტომატურად შესრულების მართვა

10.3.1	ა	10.3.2	ა	10.3.3	ა	10.3.4	ა	10.3.5	ბ
10.3.6	გ	10.3.7	ა	10.3.8	ბ	10.3.9	გ	10.3.10	ა

თაზი 11. ინფორმაცია

შესავალი. ინდექსების გამოყენების დაგეგმვა

11.1.1	ა	11.1.	ბ	11.1.3	გ	11.1.4	ბ	11.1.5	ა
11.1.6	ბ	11.1.7	გ	11.1.8	გ	11.1.9	ა	11.1.10	ბ
11.1.11	ა	11.1.12	ბ	11.1.13	გ				

არაკლასტერული ინდექსი. კლასტერული ინდექსი. უნიკალური ინდექსი

11.2.1	ა	11.2.2	ბ	11.2.3	გ	11.2.4	ა	11.2.5	ბ
11.2.6	გ	11.2.7	ა	11.2.8	ბ	11.2.9	გ	11.2.10	ა
11.2.11	ა	11.2.12	ბ	11.2.13	ა	11.2.14	ბ	11.2.15	ა
11.2.16	ბ	11.2.17	გ	11.2.18	ბ	11.2.19	ბ	11.2.20	გ
11.2.21	ა	11.2.22	გ	11.2.23	ა	11.2.24	ბ	11.2.25	ა

შევსების ფაქტორი

11.3.1	ა	11.3.2	ბ	11.3.3	გ	11.3.4	ბ	11.3.5	ა
11.3.6	გ	11.3.7	ა	11.3.8	ბ	11.3.9	გ		

ინდექსის შექმნა

11.4.1	ბ	11.4.2	ა	11.4.3	გ	11.4.4	ბ	11.4.5	გ
11.4.6	ბ	11.4.7	ა	11.4.8	გ	11.4.9	ბ	11.4.10	გ
11.4.11	ა	11.4.12	ა	11.4.13	ბ	11.4.14	ა	11.4.15	ა
11.4.16	ბ	11.4.17	ბ	11.4.18	გ	11.4.19	გ	11.4.20	ბ
11.4.21	ბ	11.4.22	ა	11.4.23	გ	11.4.24	ბ	11.4.25	ა
11.4.26	ბ	11.4.27	გ	11.4.28	ა	11.4.29	ა	11.4.30	ა
11.4.31	ბ	11.4.32	გ	11.4.33	ა	11.4.34	ბ	11.4.35	გ

ინდექსების მართვა. ინდექსისთვის სახელის შეცვლა. ინდექსის წაშლა. ინდექსების გადაწყობა. ინდექსის შესახებ ინფორმაციის მიღება

11.5.1	ა	11.5.2	ბ	11.5.3	ა	11.5.4	ბ	11.5.5	ბ
11.5.6	ა	11.5.7	ბ	11.5.8	ბ	11.5.9	ა	11.5.10	ა
11.5.11	ბ	11.5.12	ბ	11.5.13	ბ	11.5.14	ა	11.5.15	ა
11.5.16	ბ	11.5.17	ბ	11.5.18	ა	11.5.19	ბ	11.5.20	ბ
11.5.21	ა	11.5.22	ბ	11.5.23	ა	11.5.24	ბ	11.5.25	ბ

თაზი 12. წარმოდგენები

შესავალი

12.1.1	ბ	12.1.2	ა	12.1.3	ბ	12.1.4	ა	12.1.5	ბ
12.1.6	ა	12.1.7	ბ	12.1.8	ბ	12.1.9	ბ	12.1.10	ბ
12.1.11	ბ	12.1.12	ბ	12.1.13	ა	12.1.14	ა	12.1.15	ბ
12.1.16	ბ	12.1.17	ა	12.1.18	ბ	12.1.19	ბ	12.1.20	ბ
12.1.21	ბ	12.1.22	ბ	12.1.23	ა	12.1.24	ა		

წარმოდგენის შექმნა

12.2.1	ბ	12.2.2	ა	12.2.3	ბ	12.2.4	ბ	12.2.5	ბ
12.2.6	ბ	12.2.7	ა	12.2.8	ბ	12.2.9	ბ	12.2.10	ა
12.2.11	ბ								

წარმოდგენების მართვა. წარმოდგენის სახელის შეცვლა. წარმოდგენის წაშლა. წარმოდგენის შესახებ ინფორმაციის მიღება. წარმოდგენის დამოკიდებულებების ნახვა

12.3.1	ა	12.3.2	ბ	12.3.3	ბ	12.3.4	ა	12.3.5	ბ
12.3.6	ბ	12.3.7	ბ	12.3.8	ბ	12.3.9	ა	12.3.10	ბ
12.3.11	ბ	12.3.12	ა	12.3.13	ბ	12.3.14	ბ	12.3.15	ა

თაზი 13. კურსორები

შესავალი. კურსორების რეალიზება

13.1.1	ბ	13.1.2	ა	13.1.3	ა	13.1.4	ბ	13.1.5	ბ
13.1.6	ბ	13.1.7	ბ	13.1.8	ა	13.1.9	ბ	13.1.10	ბ
13.1.11	ა	13.1.12	ბ	13.1.13	ბ	13.1.14	ა	13.1.15	ბ
13.1.16	ბ								

კურსორის ტიპები. სტატიკური კურსორები. დინამიკური კურსორები. მიმდევრობითი კურსორები. საგასაღებო კურსორები

13.2.1	ა	13.2.2	ბ	13.2.3	ა	13.2.4	ბ	13.2.5	ა
13.2.6	ბ	13.2.7	ბ	13.2.8	ბ	13.2.9	ა	13.2.10	ბ
13.2.11	ბ	13.2.12	ბ	13.2.13	ბ	13.2.14	ა	13.2.15	ბ
13.2.16	ა	13.2.17	ბ	13.2.18	ბ	13.2.19	ა	13.2.20	ბ
13.2.21	ბ	13.2.22	ბ	13.2.23	ა				

კურსორების მართვა

13.3.1	ბ	13.3.2	ბ	13.3.3	ბ	13.3.4	ა	13.3.5	გ
--------	---	--------	---	--------	---	--------	---	--------	---

კურსორის შექმნა. კურსორის გახსნა

13.4.1	გ	13.4.2	გ	13.4.3	ბ	13.4.4	ა	13.4.5	გ
13.4.6	ბ	13.4.7	ბ	13.4.8	ა	13.4.9	გ	13.4.10	ბ
13.4.11	ა	13.4.12	გ	13.4.13	ბ	13.4.14	ა	13.4.15	ბ
13.4.16	ა	13.4.17	ბ	13.4.18	გ	13.4.19	ბ	13.4.20	ა
13.4.21	ა	13.4.22	ბ	13.4.23	გ	13.4.24	ა	13.4.25	ა
13.4.26	ბ	13.4.27	ბ	13.4.28	გ	13.4.29	ა	13.4.30	ბ

მონაცემების წაკითხვა

13.5.1	ა	13.5.2	ბ	13.5.3	გ	13.5.4	ა	13.5.5	ბ
13.5.6	გ	13.5.7	ბ	13.5.8	ა	13.5.9	ბ	13.5.10	გ
13.5.11	ბ	13.5.12	ა	13.5.13	ბ	13.5.14	გ	13.5.15	ბ
13.5.16	გ	13.5.17	ა	13.5.18	ბ	13.5.19	ა	13.5.20	გ
13.5.21	ბ	13.5.22	ა	13.5.23	ა	13.5.24	ბ	13.5.25	ბ
13.5.26	გ								

მონაცემების შეცვლა. მონაცემების წაშლა

13.6.1	ბ	13.6.2	გ	13.6.3	ბ	13.6.4	ა	13.6.5	გ
--------	---	--------	---	--------	---	--------	---	--------	---

კურსორის დახურვა. კურსორის გათავისუფლება

13.7.1	ა	13.7.2	ბ	13.7.3	გ	13.7.4	ბ	13.7.5	ა
13.7.6	გ								

თავი 14. ტრანზაქციები და დაბლოკვა

შესავალი

14.1.1	ა	14.1.2	გ	14.1.3	ა	14.1.4	ბ	14.1.5	გ
14.1.6	ა	14.1.7	ბ	14.1.8	ა	14.1.9	გ	14.1.10	ბ
14.1.11	გ	14.1.12	ბ	14.1.13	ა				

ტრანზაქციების მართვა. აშკარა ტრანზაქციები. ავტომატური ტრანზაქციები. არააშკარა ტრანზაქციები. განაწილებული ტრანზაქციები

14.2.1	გ	14.2.2	ა	14.2.3	ბ	14.2.4	გ	14.2.5	ბ
14.2.6	გ	14.2.7	ა	14.2.8	ბ	14.2.9	ბ	14.2.10	ბ
14.2.11	ბ	14.2.12	ა	14.2.13	გ	14.2.14	ბ	14.2.15	ა
14.2.16	გ	14.2.17	ბ	14.2.18	ბ	14.2.19	ა	14.2.20	ბ
14.2.21	ბ	14.2.22	გ	14.2.23	ბ	14.2.24	გ	14.2.25	ბ
14.2.26	გ	14.2.27	ა						

ჩადგმული ტრანზაქციები. ტრანზაქციებში აკრძალული Transact SQL-ის ბრძანებები

14.3.1	ბ	14.3.2	ა	14.3.3	ბ	14.3.4	გ	14.3.5	ბ
14.3.6	ა	14.3.7	ბ	14.3.8	გ	14.3.9	ა	14.3.10	ა

14.3.11	ბ								
---------	---	--	--	--	--	--	--	--	--

დაბლოკების მართვა

14.4.1	ა	14.4.2	ბ	14.4.3	გ	14.4.4	ა	14.4.5	ბ
14.4.6	გ	14.4.7	ა	14.4.8	ბ	14.4.9	გ	14.4.10	ა
14.4.11	ბ	14.4.12	გ	14.4.13	ა	14.4.14	ბ	14.4.15	გ
14.4.16	ა	14.4.17	ბ	14.4.18	გ				

თავი 15. ტრიბუნა

შესავალი

15.1.1	ა	15.1.2	ბ	15.1.3	გ	15.1.4	ბ	15.1.5	გ
15.1.6	ა	15.1.7	ბ	15.1.8	გ	15.1.9	ბ	15.1.10	გ
15.1.11	ა	15.1.12	გ	15.1.13	გ	15.1.14	ბ	15.1.15	ბ
15.1.16	ა	15.1.17	ა	15.1.18	ბ	15.1.19	გ	15.1.20	ა
15.1.21	გ	15.1.22	ა	15.1.23	ბ				

ტრიგერის შექმნა. ტრიგერის წაშლა

15.2.1	ა	15.2.2	გ	15.2.3	ბ	15.2.4	გ	15.2.5	ბ
15.2.6	გ	15.2.7	ა	15.2.8	გ	15.2.9	ა	15.2.10	ბ
15.2.11	ა	15.2.12	ბ	15.2.13	გ	15.2.14	ბ	15.2.15	ბ
15.2.16	ა	15.2.17	ბ	15.2.18	გ	15.2.19	ბ	15.2.20	გ
15.2.21	ა	15.2.22	გ	15.2.23	ა	15.2.24	ა	15.2.25	გ
15.2.26	ბ	15.2.27	გ	15.2.28	ა	15.2.29	გ	15.2.30	ა
15.2.31	ბ	15.2.32	ბ	15.2.33	ა	15.2.34	ა	15.2.35	გ
15.2.36	გ	15.2.37	ბ	15.2.38	გ	15.2.39	ა	15.2.40	ა
15.2.41	ბ	15.2.42	გ	15.2.43	ბ	15.2.44	ა	15.2.45	ბ
15.2.46	გ	15.2.47	ა	15.2.48	ბ				

ტრიგერების შექმნისას გასათვალისწინებელი რეკომენდაციები. ტრიგერების მართვა. ტრიგერის სახელის შეცვლა. ტრიგერის შესახებ ინფორმაციის მიღება

15.3.1	ა	15.3.2	გ	15.3.3	ბ	15.3.4	გ	15.3.5	ა
15.3.6	გ	15.3.7	ბ	15.3.8	ა	15.3.9	გ	15.3.10	ბ
15.3.11	ა	15.3.12	ბ	15.3.13	ბ	15.3.14	გ	15.3.15	გ
15.3.16	ა	15.3.17	ა	15.3.18	ბ	15.3.19	ბ	15.3.20	ა
15.3.21	ა	15.3.22	გ	15.3.23	გ	15.3.24	ა	15.3.25	ბ

თავი 16. ცხრილური ბამოსახულებები

წარმოებული ცხრილები. სვეტებისთვის ფსევდონიმების მინიჭება. არგუმენტების გამოყენება. ჩადგმულობა. მრავლობითი მიმართვები

16.1.1	ა	16.1.2	ბ	16.1.3	გ	16.1.4	ა	16.1.5	ა
16.1.6	ბ	16.1.7	გ	16.1.8	ა	16.1.9	ბ	16.1.10	გ
16.1.11	ა	16.1.12	ბ						

საერთო ცხრილური გამოსახულებები. სვეტებისთვის ფსევდონიმების დანიშვნა. არგუმენტების გამოყენება. რამდენიმე საერთო ცხრილური გამოსახულების განსაზღვრა. მრავლობითი მიმართვები

16.2.1	ბ								
--------	---	--	--	--	--	--	--	--	--

APPLY ოპერაცია

16.3.1	ა	16.3.2	ბ	16.3.3	ბ	16.3.4	ა		
--------	---	--------	---	--------	---	--------	---	--	--

თავი 17. მონაცემების რეორბანიზაცია და დაჯგუფების ნაკრებები

მონაცემების გაშლა. მონაცემების გაშლა T-SQL ენის PIVOT ოპერაციის გამოყენებით

17.1.1	გ	17.1.2	ა	17.1.3	ბ	17.1.4	ა	17.1.5	ა
17.1.6	ბ	17.1.7	გ	17.1.8	ა				

მონაცემების შეკვეცა. მონაცემების შეკვეცა სტანდარტული SQL ენის საშუალებით. შეკვეცა T-SQL-ის UNPIVOT ოპერაციის საშუალებით

17.2.1	გ	17.2.2	ა	17.2.3	ბ	17.2.4	გ		
--------	---	--------	---	--------	---	--------	---	--	--

დაჯგუფების ნაკრებები. GROUPING SETS ჩადგმული ელემენტი. CUBE ჩადგმული ელემენტი. ROLLUP ჩადგმული ელემენტი. GROUPING() და GROUPING_ID() ფუნქციები

17.3.1	ა	17.3.2	ბ	17.3.3	გ	17.3.4	ბ	17.3.5	ა
17.3.6	გ	17.3.7	ბ	17.3.8	ბ	17.3.9	გ		

თავი 18. მონაცემთა ბაზისის სარეზერვო ასლები

სარეზერვო ასლის ტიპები

18.1.1	ა	18.1.2	ბ	18.1.3	გ	18.1.4	ა	18.1.5	ბ
18.1.6	გ	18.1.7	ა	18.1.8	ბ	18.1.9	გ	18.1.10	ა
18.1.11	ბ	18.1.12	გ	18.1.13	ა	18.1.14	ბ	18.1.15	გ

დამგროვებლის არჩევა. მოწყობილობის განსაზღვრა სარეზერვო ასლის მოსათავსებლად

18.2.1	ა	18.2.2	ბ	18.2.3	გ	18.2.4	ა	18.2.5	ბ
18.2.6	გ	18.2.7	ბ	18.2.8	გ	18.2.9	ა	18.2.10	ბ
18.2.11	გ	18.2.12	ა						

მონაცემთა ბაზის სრული და დიფერენცირებული ასლები

18.3.1	ა	18.3.2	ბ	18.3.3	გ	18.3.4	ა	18.3.5	ა
18.3.6	ბ	18.3.7	გ	18.3.8	ა	18.3.9	ბ	18.3.10	გ
18.3.11	ა	18.3.12	ბ						

სრული და დიფერენცირებული სარეზერვო ასლების შექმნა

18.4.1	გ	18.4.2	ა	18.4.3	ბ	18.4.4	გ	18.4.5	ა
18.4.6	ბ	18.4.7	ბ	18.4.8	გ				

სრული და დიფერენცირებული სარეზერვო ასლიდან აღდგენა

18.5.1	გ	18.5.2	ა	18.5.3	ბ	18.5.4	გ	18.5.5	ა
18.5.6	ბ								

თავი 19. უსაფრთხოების სისტემა

უფლებამოსილების გამიჯვნის წესები

19.1.1	ა	19.1.2	ბ	19.1.3	გ	19.1.4	ა	19.1.5	ბ
19.1.6	გ	19.1.7	ა	19.1.8	ბ	19.1.9	გ	19.1.10	ა
19.1.11	ბ	19.1.12	გ	19.1.13	ა	19.1.14	ბ	19.1.15	გ
19.1.16	ა	19.1.17	ბ	19.1.18	გ	19.1.19	ა		

სერვერის უსაფრთხოების სისტემის არქიტექტურა. აუტენტიფიცირების რეჟიმები.
Windows NT-ის აუტენტიფიცირების რეჟიმი. სერვერის აუტენტიფიცირების რეჟიმი

19.2.1	ა	19.2.2	ბ	19.2.3	გ	19.2.4	ა	19.2.5	ბ
19.2.6	გ	19.2.7	ა	19.2.8	ბ	19.2.9	გ	19.2.10	ა
19.2.11	ბ	19.2.12	გ	19.2.13	ა	19.2.14	ბ	19.2.15	გ
19.2.16	ა	19.2.17	ბ	19.2.18	გ	19.2.19	ა	19.2.20	ბ
19.2.21	გ	19.2.22	ა	19.2.23	ბ	19.2.24	გ	19.2.25	ა
19.2.26	ბ	19.2.27	გ	19.2.28	ა	19.2.29	ბ	19.2.30	გ
19.2.31	ა								

უსაფრთხოების სისტემის ელემენტები

19.3.1	ბ	19.3.2	ა	19.3.	ბ	19.3.4	გ	19.3.5	ა
19.3.6	ბ	19.3.7	გ	19.3.8	ა				

სქემები

19.4.1	ბ	19.4.2	გ	19.4.3	ა	19.4.4	ბ	19.4.5	გ
19.4.6	ა	19.4.7	ბ	19.4.8	გ	19.4.9	ა	19.4.10	ბ
19.4.11	გ	19.4.12	ა	19.4.13	ბ	19.4.14	გ	19.4.15	ა
19.4.16	ბ	19.4.17	გ	19.4.18	ა	19.4.19	ბ	19.4.20	გ
19.4.21	ა	19.4.22	ბ	19.4.23	გ	19.4.24	ა	19.4.25	ბ
19.4.26	გ	19.4.27	გ						

მომხმარებლები

19.5.1	ა	19.5.2	ბ	19.5.3	გ	19.5.4	ა	19.5.5	ბ
19.5.6	გ	19.5.7	ა	19.5.8	ბ	19.5.9	გ	19.5.10	ა
19.5.11	ბ	19.5.12	გ	19.5.13	ა	19.5.14	ბ	19.5.15	გ
19.5.16	ა	19.5.17	ბ	19.5.18	გ				

სერვერის როლები. მონაცემთა ბაზების როლები

19.6.1	ა	19.6.2	ბ	19.6.3	გ	19.6.4	ა	19.6.5	ბ
19.6.6	გ	19.6.7	ა	19.6.8	ბ	19.6.9	გ	19.6.10	ა
19.6.11	ბ	19.6.12	გ	19.6.13	ა	19.6.14	ბ	19.6.15	გ
19.6.16	ა	19.6.17	ბ	19.6.18	გ	19.6.19	ა	19.6.20	ბ
19.6.21	გ	19.6.22	ა	19.6.23	ბ				

უსაფრთხოების სისტემის ადმინისტრირება. სააღრიცხვო ჩანაწერების შექმნა და მართვა

19.7.1	გ	19.7.2	ა	19.7.3	ბ	19.7.4	გ	19.7.5	ა
19.7.6	ბ								

მიმართვის უფლებები. მონაცემთა ბაზის ობიექტებთან მიმართვის უფლებები

19.8.1	გ	19.8.2	ა	19.8.3	ბ	19.8.4	გ	19.8.5	ა
19.8.6	ბ	19.8.7	გ	19.8.8	ა	19.8.9	ბ	19.8.10	გ
19.8.11	ა	19.8.12	ბ	19.8.13	გ	19.8.14	ა	19.8.15	ბ
19.8.16	გ	19.8.17	ა	19.8.18	ბ	19.8.19	გ	19.8.20	ა
19.8.21	ბ	19.8.22	გ	19.8.23	ა	19.8.24	ბ	19.8.25	გ
19.8.26	ა								

Transact-SQL-ის ბრძანებების შესრულების უფლებები

19.9.1	ბ	19.9.2	გ	19.9.3	ა	19.9.4	ბ	19.9.5	გ
19.9.6	ა	19.9.	ბ	19.9.8	გ	19.9.9	ა	19.9.10	ბ
19.9.11	გ	19.9.12	ა						

არაცხადი უფლებები

19.10.1	ბ	19.10.2	გ	19.10.3	ა	19.10.4	ბ	19.10.5	გ
19.10.6	ა	19.10.7	ბ	19.10.8	გ	19.10.9	ა	19.10.10	ბ

მიმართვის აკრძალვა

19.11.1	გ	19.11.2	ა	19.11.3	ბ	19.11.4	გ	19.11.5	ა
19.11.6	ბ	19.11.7	გ	19.11.8	ა	19.11.9	ბ		

მიმართვის არაცხადი უარყოფა

19.12.1	გ	19.12.2	ა	19.12.3	ბ	19.12.4	გ	19.12.5	ა
19.12.6	ბ	19.12.7	გ						

ლიტერატურა

1. რ. სამხარაძე. SQL სერვერი.
2. А.В. Фролов, Г.В. Фролов. Базы Данных в Интернете.
3. Администрирование SQL Server 2000. Учебный курс MCSA/MCSE, MCDBA/Пер. с англ. — М.: Издательско_торговый дом «Русская Редакция», 2002. — 816 стр.: ил.
4. Основы реляционных баз данных. Пер. с англ. — М.: Издательско_торговый дом «Русская Редакция», 2001. - 384 стр.: ил.
5. Р. Саукап. Основы Microsoft SQL Server 6.5. Пер. с англ. — М.: Издательский отдел "Русская редакция" ТОО «Channel Trading Ltd.». - 1999. - 704 стр.: ил.
6. Реализация баз данных Microsoft SQL Server 7.0. Учебный курс. Пер. с англ. — М.: Издательско_торговый дом «Русская Редакция», 2000. - 528 стр.: ил.
7. Малкольм Г. Программирование для Microsoft SQL Server 2000 с использованием XML. М.: Издательско_торговый дом «Русская Редакция», 2002. - 320 стр.: ил.
8. Проектирование и реализация баз данных. М.: Издательско_торговый дом «Русская Редакция», 2001. - 704 стр.: ил.
9. Г. Дейтел. Введение в операционные системы. В 2-х томах. Пер. с англ. - М.: Мир, 1987.
10. Э. Таненбаум. Современные операционные системы. - СПб.: Питер, 2002. - 1040 с.: ил.
11. А. Джонс. Руководство системного администратора Windows.

კომპიუტერული უზრუნველყოფა ლ. გაჩეჩილაძის

იბეჭდება ავტორის მიერ წარმოდგენილი სახით

წარმოებას გადაეცა . ხელმოწერილია დასაბეჭდად . ქალაქის ზომა 60X84 1/8.
პირობითი ნაბეჭდი თაბახი 18,25. ტირაჟი 50 ეგზ.

სტუ-ს "IT-კონსალტინგის სამეცნირო ცენტრი", თბილისი, კოსტავას 77