

რომან სამხარაძე

მონაცემთა ბაზები

(ტესტების კრებული)

„ტექნიკური უნივერსიტეტი“

საქართველოს ტექნიკური უნივერსიტეტი

რომან სამხარაძე

მონაცემთა ბაზები

(ტესტების კრებული)

რეგისტრირებულია სტუ-ს
სარედაქციო-საგამომცემლო
საბჭოს მიერ.17.03.2010, ოქმი №2

თბილისი
2010

უაკ 681.3.06(075)

კრებულში წარმოდგენილია SQL სერვერისთვის შემუშავებული ტესტები. ტესტებში ასახულია SQL სერვერთან მუშაობის საფუძვლები, მონაცემთა ბაზებისა და ცხრილების მართვის საკითხები, ფუნქციებთან, შენახულ პროცედურებთან, ინდექსებთან, წარმოდგენებთან, კურსორებსა და ტრიგერებთან მუშაობის პრინციპები.

განკუთვნილია კომპიუტერული სისტემებისა და ქსელების სპეციალობის სტუდენტებისათვის (შიფრი - 2201).

რეცენზენტი პედაგოგიკის მეცნიერებათა დოქტორი,
სრული პროფესორი გ. ჩაჩანიძე

© საგამომცემლო სახლი "ტექნიკური უნივერსიტეტი", 2010

ISBN 978-9941-14-809-5

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

ყველა უფლება დაცულია. ამ წიგნის არც ერთი ნაწილი (იქნება ეს ტექსტი, ფოტო, ილუსტრაცია თუ სხვა) არანაირი ფორმით და საშუალებით (იქნება ეს ელექტრონული თუ მექანიკური), არ შეიძლება გამოყენებულ იქნას გამომცემლის წერილობითი ნებართვის გარეშე.

საავტორო უფლების დარღვევა ისჯება კანონით.

სარჩევი

თავი 1. მონაცემთა ბაზების თეორიული საფუძვლები 9

- საინფორმაციო სისტემები..... 9
- მონაცემთა ბაზები 9
- მონაცემთა ბაზების მოდელები11
- მონაცემთა რელაციური ბაზები14
- ცხრილების ნორმალიზება.....14
- ცხრილებს შორის არსებული კავშირები16
- "ერთი-ერთთან" კავშირი19
- "ერთი-ბევრთან" კავშირი19
- "ბევრი-ბევრთან" კავშირი.....20

თავი 2. მონაცემთა ბაზები21

- მონაცემთა ბაზების არქიტექტურა.....21
- მონაცემთა ბაზების ობიექტები21
- ცხრილები.....21
- მომხმარებლების ტიპები.....24
- წარმოდგენები.....24
- შენახული პროცედურები25
- ტრიგერები25
- ინდექსები.....26
- მთლიანობის შეზღუდვა26
- ნაგულისხმევი მნიშვნელობები.....30
- ფუნქციები.....30
- სისტემური მონაცემთა ბაზები31
- Master სისტემური მონაცემთა ბაზა31
- Tempdb სისტემური მონაცემთა ბაზა32
- Model სისტემური მონაცემთა ბაზა32
- Msdb სისტემური მონაცემთა ბაზა.....32
- ფაილები და ფაილების ჯგუფები33
- მონაცემთა ბაზის ფიზიკური სტრუქტურა35
- მონაცემთა ბაზების შექმნა, წაშლა და სახელის შეცვლა.....37
- მონაცემთა ბაზის შექმნა37
- მონაცემთა ბაზის წაშლა42
- მონაცემთა ბაზის სახელი შეცვლა.....42
- მონაცემთა ბაზის მიერთება და გამორთვა.....42
- მონაცემთა ბაზების მართვა44
- ფლობის უფლების გადაცემა44
- მონაცემთა ბაზის შეკუმშვა45
- მონაცემთა ბაზის თვისებების მართვა.....50
- მონაცემთა ბაზის საკონფიგურაციო პარამეტრებთან მუშაობის სხვა საშუალებები56
- მონაცემთა მომხმარებლის ტიპის შექმნა59

თავი 3. ცხრილები.....	62
ცხრილების დაპროექტება.....	62
ცხრილის პირველადი გასაღები.....	62
ცხრილის გარე გასაღები.....	63
სვეტის უნიკალურობის განსაზღვრა.....	64
შემმოწმებელი შეზღუდვების განსაზღვრა.....	64
ნაგულისხმევი მნიშვნელობის განსაზღვრა.....	65
სვეტ-მთველის განსაზღვრა.....	66
ცხრილის შექმნა.....	69
ცხრილის წაშლა.....	77
ცხრილის შეცვლა.....	77
ცხრილების მართვა.....	78
ცხრილისათვის სახელის შეცვლა.....	78
ინფორმაციის მიღება ცხრილების შესახებ.....	79
ინფორმაციის მიღება ცხრილზე დამოკიდებული ობიექტების შესახებ.....	79
ცხრილებს შორის კავშირის შესახებ ინფორმაციის მიღება.....	80
დიაგრამები.....	81
თავი 4. მონაცემების მართვა.....	82
ცხრილში მონაცემების ჩამატება.....	82
INSERT ბრძანება.....	82
SELECT...INTO ბრძანება.....	84
ცხრილიდან მონაცემების ამორჩევა.....	85
SELECT ბრძანება.....	85
SELECT განყოფილება.....	85
FROM განყოფილება.....	88
WHERE განყოფილება.....	90
ჩადგმული მოთხოვნები.....	91
ბმული მოთხოვნები.....	92
GROUP BY განყოფილება.....	92
HAVING განყოფილება.....	93
UNION განყოფილება.....	93
ORDER BY განყოფილება.....	94
COMPUTE განყოფილება.....	95
აგრეგირების ფუნქციები.....	95
ცხრილის მონაცემების ცვლილება.....	97
UPDATE ბრძანება.....	97
ცხრილიდან მონაცემების წაშლა.....	98
DELETE ბრძანება.....	98
მონაცემების მასობრივი გადაწერა.....	99
თავი 5. Transact SQL-ის საფუძვლები.....	100
შესავალი.....	100

გამოსახულებები	101
ოპერატორები	102
იდენტიფიკატორები.....	104
ცვლადები	108
მმართველი კონსტრუქციები.....	109
BEGIN...END	109
IF...ELSE.....	110
CASEEND	110
COALESCE	111
WHILE...BREAK & CONTINUE	111
ლოგიკის ოპერატორები	112
ALL ოპერატორი	112
SOME და ANY ოპერატორები	112
BETWEEN ოპერატორი	113
EXISTS ოპერატორი	113
IN ოპერატორი	113
LIKE ოპერატორი	114
თავი 6. ფუნქციები.....	115
მომხმარებლის ფუნქციები.....	115
Scalar ტიპის ფუნქციები.....	116
Inline ტიპის ფუნქციები	118
Multi-statement ფუნქციები.....	119
ფუნქციის შეცვლა	119
ფუნქციის წაშლა.....	120
ჩადგმული ფუნქციები	120
მათემატიკის ფუნქციები	120
სტრიქონებთან სამუშაო ფუნქციები	123
თარიღებთან სამუშაო ფუნქციები.....	126
კონფიგურირების ფუნქციები.....	127
სისტემური ფუნქციები.....	129
უსაფრთხოების სისტემის ფუნქციები	132
მეტამონაცემების მართვის ფუნქციები.....	133
სტატისტიკის ფუნქციები	136
თავი 7. შენახული პროცედურები	138
შესავალი	138
შენახული პროცედურების ტიპები.....	139
შენახული პროცედურის შექმნა	141
შენახული პროცედურების მართვა.....	146
შენახული პროცედურის შეცვლა	146
შენახული პროცედურის შესახებ ინფორმაციის მიღება	146
შენახული პროცედურის სახელის შეცვლა	146
შენახული პროცედურის წაშლა.....	146
შენახული პროცედურის ავტომატურად შესრულების მართვა	146

თავი 8. ინდექსები.....	148
შესავალი	148
ინდექსების გამოყენების დაგეგმვა	148
არაკლასტერული ინდექსი.....	149
კლასტერული ინდექსი.....	149
უნიკალური ინდექსი.....	151
შევსების ფაქტორი.....	152
ინდექსის შექმნა	153
ინდექსების მართვა.....	158
ინდექსის შეცვლა	158
ინდექსისთვის სახელის შეცვლა	158
ინდექსის წაშლა.....	158
ინდექსების გადაწყობა	159
ინდექსის შესახებ ინფორმაციის მიღება.....	160
თავი 9. წარმოდგენები.....	162
შესავალი	162
წარმოდგენები და უსაფრთხოება.....	165
წარმოდგენის შექმნა	165
წარმოდგენების მართვა.....	166
წარმოდგენის შეცვლა	166
წარმოდგენის სახელის შეცვლა	166
წარმოდგენის წაშლა.....	166
წარმოდგენის შესახებ ინფორმაციის მიღება.....	167
წარმოდგენის დამოკიდებულებების ნახვა	167
თავი 10. კურსორები.....	168
შესავალი	168
კურსორების რეალიზება.....	168
კურსორის ტიპები.....	169
სტატიკური კურსორები	169
დინამიკური კურსორები.....	170
მიმდევრობითი კურსორები	170
საგასაღებო კურსორები	171
კურსორების მართვა.....	172
კურსორის შექმნა.....	173
კურსორის გახსნა.....	176
მონაცემების წაკითხვა	177
მონაცემების შეცვლა	179
მონაცემების წაშლა.....	180
კურსორის დახურვა.....	180
კურსორის გათავისუფლება.....	180
თავი 11. ტრანზაქცია და დაბლოკვა.....	181
შესავალი	181
ტრანზაქციების მართვა.....	186

აშკარა ტრანზაქციები.....	187
ავტომატური ტრანზაქციები.....	189
არააშკარა ტრანზაქციები.....	189
განაწილებული ტრანზაქციები.....	190
ჩადგმული ტრანზაქციები.....	190
თავი 12. ტრიგერები	193
შესავალი	193
ტრიგერის შექმნა.....	195
ტრიგერის შეცვლა.....	198
ტრიგერის წაშლა	198
ტრიგერების დაპროგრამება.....	199
ტრიგერების მართვა	202
ტრიგერის სახელის შეცვლა	202
ტრიგერების შექმნის დროს არსებული შეზღუდვები	202
ტრიგერის შესახებ ინფორმაციის მიღება	202
დანართი. ტესტების პასუხები	204
ლიტერატურა	223

თავი 1. მონაცემთა ბაზების თეორიული საფუძვლები

საინფორმაციო სისტემები

1.1.1.1.1. საინფორმაციო სისტემა არის:

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

1.1.1.1.2. არსებობს საინფორმაციო სისტემების:

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

1.1.1.1.3. საინფორმაციო-სამეზნი სისტემები ორიენტირებულია:

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

1.1.1.1.4. მონაცემების დამუშავების სისტემები ორიენტირებულია:

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

მონაცემთა ბაზები

1.2.1.1.1. მონაცემთა ბაზა არის გარკვეული საგნობრივი სფეროს ობიექტების შესახებ:

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

1.2.1.1.2. მონაცემთა ბაზასთან მუშაობა შეიძლება:

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

1.2.1.1.3. 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

1.2.1.1.4. მონაცემთა ბაზა შეიძლება იყოს:

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

1.2.1.1.5. მონაცემთა განაწილებული ბაზის ნაწილები განთავსებულია:

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

1.2.1.1.6. მონაცემთა ცენტრალიზებული ბაზა მოთავსებულია:

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

1.2.1.1.7. არქიტექტურის მიხედვით, მონაცემთა ცენტრალიზებული ბაზა შეიძლება იყოს:

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

1.2.1.1.8. ფაილ-სერვერული არქიტექტურის შემთხვევაში:

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

1.2.1.1.9. 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

1.2.1.1.10. კლიენტ-სერვერული არქიტექტურის შემთხვევაში მონაცემთა ბაზა მოთავსებულია:

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

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

- ა. მხოლოდ კლიენტის ნაწილისგან
- ბ. მხოლოდ სერვერის ნაწილისგან
- გ. კლიენტის და სერვერის ნაწილებისგან

1.2.1.1.12. 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

1.2.1.1.13. 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

1.2.1.1.14. 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

1.2.1.1.15. კლიენტ-სერვერული ტექნოლოგიისათვის:

- ა. არ არის დამახასიათებელი SQL ენის გამოყენება
- ბ. დამახასიათებელია C# ენის გამოყენება
- გ. დამახასიათებელია SQL ენის გამოყენება

მონაცემთა ბაზების მოდელები

1.3.1.1.1. 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

1.3.1.1.2. 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

1.3.1.1.3. არსებობს მონაცემების:

- ა. მხოლოდ რელაციური მოდელი
- ბ. იერარქიული, ქსელური და რელაციური მოდელები
- გ. მხოლოდ ქსელური მოდელი

1.3.1.1.4. მონაცემთა ბაზა შეიძლება იყოს:

- ა. მხოლოდ რელაციური
- ბ. იერარქიული, ქსელური და რელაციური
- გ. მხოლოდ ქსელური

1.3.1.1.5. მონაცემების იერარქიული მოდელის თითოეული ელემენტი:

- ა. სხვა ელემენტებს წარმოქმნის
- ბ. სხვა ელემენტებს არ წარმოქმნის
- გ. ინდექსებს წარმოქმნის

1.3.1.1.6. მონაცემების იერარქიული მოდელის ნებისმიერ წარმოქმნილ ელემენტს აქვს:

- ა. მხოლოდ ერთი წარმომქმნელი ელემენტი
- ბ. რამდენიმე წარმომქმნელი ელემენტი
- გ. მხოლოდ ორი წარმომქმნელი ელემენტი

1.3.1.1.7. მონაცემების იერარქიული მოდელის ნაკლია ის, რომ:

- ა. ხანდახან შესაძლებელია იერარქიის შეცვლა
- ბ. შესაძლებელია შემუშავებული იერარქიის შეცვლა
- გ. შეუძლებელია შემუშავებული იერარქიის შეცვლა

1.3.1.1.8. 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

1.3.1.1.9. ქსელურ მოდელის ნაკლია:

- ა. იმისათვის, რომ მონაცემები დავამუშაოთ, არ უნდა ვიცოდეთ მონაცემთა ბაზის სტრუქტურა
- ბ. იმისათვის, რომ მონაცემები დავამუშაოთ, უნდა ვიცოდეთ ტრიგერის სტრუქტურა
- გ. იმისათვის, რომ მონაცემები დავამუშაოთ, უნდა ვიცოდეთ მონაცემთა ბაზის სტრუქტურა

1.3.1.1.10. 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

1.3.1.1.11. მარტივ შემთხვევაში, რელაციური მოდელი აღწერს:

- ა. ერთ ერთგანზომილებიან ცხრილს
- ბ. ერთ ორგანზომილებიან ცხრილს
- გ. ერთ სამგანზომილებიან ცხრილს

1.3.1.1.12. ხშირ შემთხვევაში, რელაციური მოდელი:

- ა. არ აღწერს არაფერს
- ბ. აღწერს ერთი ცხრილის სტრუქტურასა და მის შორის კავშირს
- გ. აღწერს რამდენიმე ცხრილის სტრუქტურასა და მათ შორის კავშირს

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

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

1.3.1.1.14. მონაცემთა ბაზის შიგნით ცხრილის სახელი:

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

1.3.1.1.15. ცხრილი შედგება:

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

1.3.1.1.16. ცხრილი შეიცავს ინფორმაციას:

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

1.3.1.1.17. ცხრილის სტრიქონი შეიცავს ინფორმაციას:

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

1.3.1.1.18. ცხრილის თითოეული სვეტი:

- ა. არის ობიექტების კონკრეტული ატრიბუტის მნიშვნელობების ერთობლიობა
- ბ. არ არის ობიექტების კონკრეტული ატრიბუტის მნიშვნელობების ერთობლიობა
- გ. არის ობიექტების არაკონკრეტული ატრიბუტის მნიშვნელობების ერთობლიობა

1.3.1.1.19. თითოეულ სვეტს აქვს:

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

1.3.1.1.20. ერთი ცხრილის სვეტებს:

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

1.3.1.1.21. სხვადასხვა ცხრილს:

- ა. არ შეიძლება ერთნაირი სახელის მქონე სვეტები ჰქონდეს
- ბ. აუცილებლად უნდა ჰქონდეს სხვადასხვა სახელი
- *გ. შეიძლება ერთნაირი სახელის მქონე სვეტები ჰქონდეს

1.3.1.1.22. ცხრილის სტრიქონებს სახელები:

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

მონაცემთა რელაციური ბაზები

1.4.1.1.1. 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

1.4.1.1.2. მონაცემთა ბაზაში თითოეულ ცხრილს:

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

1.4.1.1.3. სტრიქონის ერთ სვეტში მოთავსებული უნდა იყოს:

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

1.4.1.1.4. ცხრილში:

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

1.4.1.1.5. სტრიქონები უნდა განსხვავდებოდეს:

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

ცხრილების ნორმალიზება

1.5.1.1.1. მონაცემთა ბაზის დაპროექტების მიზანია:

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

1.5.1.1.2. ცხრილების ნორმალიზება არის:

- ა. მონაცემების წარმოდგენის პროცესი მარტივი ორგანოზომილებიანი ცხრილის სახით
- ბ. მონაცემების წარმოდგენის პროცესი მარტივი სამგანოზომილებიანი ცხრილის სახით
- გ. მონაცემების წარმოდგენის პროცესი მარტივი ერთგანოზომილებიანი ცხრილის სახით

1.5.1.1.3. ნორმალიზების მიზანია:

- ა. ცხრილების რაოდენობის მაქსიმუმამდე დაყვანა

- ბ. მონაცემების სიჭარბის გამორიცხვა
- გ. ცხრილების გადაწერა

1.5.1.1.4. ნორმალიზების მიზანია:

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

1.5.1.1.5. არსებობს ცხრილის ნორმალიზების:

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

1.5.1.1.6. 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

1.5.1.1.7. პირველ ნორმალურ ფორმაში მყოფ ცხრილს:

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

1.5.1.1.8. 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, მასთან დაკავშირებულია სრული ფუნქციური 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, მასთან დაკავშირებულია სრული ფუნქციური დამოკიდებულებით

1.5.1.1.9. 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

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

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

1.5.1.1.11. 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

პირველადი და გარე გასაღებები

1.6.1.1.1. ორ ცხრილს შორის კავშირის დამყარების დროს:

- ა. ორივე მთავარია
- ბ. ერთი მთავარია (მშობელი), მეორე - დამოკიდებული (შვილობილი)
- გ. ორივე დამოკიდებულია

1.6.1.1.2. 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

1.6.1.1.3. პირველადი გასაღები:

- ა. ცალსახად არ განსაზღვრავს თითოეულ სტრიქონს ცხრილში
- ბ. ცალსახად განსაზღვრავს თითოეულ სტრიქონს ცხრილში
- გ. ცალსახად განსაზღვრავს თითოეულ სვეტს ცხრილში

1.6.1.1.4. პირველადი გასაღები უნდა აკმაყოფილებდეს შემდეგ მოთხოვნებს:

- ა. უნიკალურობა და მინიმალურობა
- ბ. მხოლოდ უნიკალურობა
- გ. მხოლოდ მინიმალურობა

1.6.1.1.5. უნიკალურობა ნიშნავს, რომ:

- ა. ცხრილში უნდა იყოს ორი ისეთი სტრიქონი, რომლებსაც პირველადი გასაღების ერთნაირი მნიშვნელობები ექნება
- ბ. ცხრილში არ უნდა იყოს სამი ისეთი სტრიქონი, რომლებსაც პირველადი გასაღების ერთნაირი მნიშვნელობები ექნება
- გ. ცხრილში არ უნდა იყოს ორი ისეთი სტრიქონი, რომლებსაც პირველადი გასაღების ერთნაირი მნიშვნელობები ექნება

1.6.1.1.6. მინიმალურობა 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

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

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

1.6.1.1.8. 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

1.6.1.1.9. 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

1.6.1.1.10. გარე გასაღები (FOREIGN KEY) იქმნება:

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

1.6.1.1.11. გარე გასაღები შეიძლება შედგებოდეს:

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

1.6.1.1.12. 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

1.6.1.1.13. 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

1.6.1.1.14. მთავარ ცხრილში პირველადი გასაღების მნიშვნელობის შეცვლისას არსებობს:

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

1.6.1.1.15. მთავარ ცხრილში პირველადი გასაღების მნიშვნელობის შეცვლისას არსებობს დამოკიდებულებული ცხრილის ქცევის შემდეგი ვარიანტები:

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

1.6.1.1.16. 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

1.6.1.1.17. 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

1.6.1.1.18. 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

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

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

1.6.1.1.20. 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

1.6.1.1.21. 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

1.6.1.1.22. 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)

"ერთი-ერთთან" კავშირი

1.6.2.1.1. 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

1.6.2.1.2. „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

1.6.2.1.3. „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.

"ერთი-ბევრთან" კავშირი

1.6.3.1.1. „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

"ბევრი-ბევრთან" კავშირი

1.6.4.1.1. „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

2.1.0.1.1. მონაცემთა ბაზის სტრუქტურა შეიძლება იყოს:

- ა. მხოლოდ ლოგიკური
- ბ. მხოლოდ ფიზიკური
- გ. ლოგიკური და ფიზიკური

2.1.0.1.2. 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

2.1.0.1.3. 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

მონაცემთა ბაზების ობიექტები

2.1.1.1.1. მონაცემთა ბაზების ობიექტებია:

- ა. მხოლოდ ცხრილები
- ბ. ცხრილები, შენახული პროცედურები, ტრიგერები, წარმოდგენები, ინდექსები, მომხმარებლების ტიპები, მომხმარებლების ფუნქციები
- გ. მხოლოდ ცხრილები და ინდექსები

ცხრილები

2.1.2.1.1. მონაცემთა ბაზის ობიექტებიდან საკუთრივ მონაცემებს შეიცავს მხოლოდ:

- ა. ცხრილი
- ბ. ტრიგერი
- გ. წარმოდგენა

2.1.2.1.2. ცხრილის ზოგიერთი სვეტი შეიძლება იყოს:

- ა. გამოთვლადი
- ბ. წარმოდგენა
- გ. პირობითი

2.1.2.1.3. გამოთვლად სვეტებში მიეთითება:

- ა. წარმოდგენა

- ბ. ფორმულა
- გ. პირობა

2.1.2.1.4. სისტემური ცხრილები შეიცავს:

- ა. ინფორმაციას მხოლოდ ტრიგერებისა და ინდექსების შესახებ
- ბ. სერვერის მუშაობისათვის არასაჭირო ინფორმაციას
- გ. სერვერის მუშაობისათვის საჭირო ინფორმაციას

2.1.2.1.5. დროებითი ცხრილები განკუთვნილია:

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

2.1.2.1.6. დროებითი ცხრილები მოთავსებულია:

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

2.1.2.1.7. 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

2.1.2.1.8. დროებითი ცხრილი:

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

2.1.2.1.9. დროებითი ცხრილები ავტომატურად:

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

2.1.2.1.10. ლოკალური დროებითი ცხრილის სახელი:

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

2.1.2.1.11. 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

2.1.2.1.12. შეერთების დახურვის დროს:

- ა. ლოკალური დროებითი ცხრილი ავტომატურად იშლება

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

2.1.2.1.13. 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

2.1.2.1.14. გლობალური დროებითი ცხრილის სახელი:

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

2.1.2.1.15. 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

2.1.2.1.16. გლობალურ ცხრილზე მიმართვა:

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

2.1.2.1.17. დროებითი გლობალური ცხრილები:

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

2.1.2.1.18. 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

2.1.2.1.19. გლობალური დროებითი ცხრილის წაშლა და შეცვლა:

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

2.1.2.1.20. დროებითი ცხრილი ყოველთვის იქმნება:

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

2.1.2.1.21. სერვერის გაჩერების შემთხვევაში:

- ა. ავტომატურად იშლება მხოლოდ ლოკალური ცხრილი

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

მომხმარებლების ტიპები

2.1.3.1.1. მონაცემთა მომხმარებლის ტიპები:

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

2.1.3.1.2. მომხმარებლის ტიპის სახელი უნიკალური:

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

2.1.3.1.3. 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

წარმოდგენები

2.1.4.1.1. წარმოდგენა არის ვირტუალური ცხრილი, რომლის შემცველობა განისაზღვრება:

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

2.1.4.1.2. წარმოდგენა შედგება:

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

2.1.4.1.3. წარმოდგენა მონაცემებს:

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

2.1.4.1.4. 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

2.1.4.1.5. რთული წარმოდგენა შეიცავს:

- ა. ერთი ცხრილის გაუფილტრავ ინფორმაციას

- ბ. მხოლოდ ერთ ცხრილს
- გ. რამდენიმე ცხრილის გაფილტრულ ინფორმაციას

შენახული პროცედურები

2.1.5.1.1. 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

2.1.5.1.2. სისტემური შენახული პროცედურების სახელები:

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

2.1.5.1.3. საკუთარი შენახული პროცედურების შექმნა:

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

2.1.5.1.4. როცა შენახული პროცედურა პირველად სრულდება, მაშინ სერვერი:

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

2.1.5.1.5. შენახული პროცედურის განმეორებით შესრულებისას, სერვერი:

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

ტრიგერები

2.1.6.1.1. 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

2.1.6.1.2. ტრიგერები სამ კატეგორიად იყოფა:

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

2.1.6.1.3. 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

2.1.6.1.4. ერთი ტრიგერიდან:

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

ინდექსები

2.1.7.1.1. ინდექსი განკუთვნილია შესაბამის ცხრილში ან წარმოდგენაში ინფორმაციის:

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

2.1.7.1.2. ინდექსი განისაზღვრება:

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

2.1.7.1.3. ინდექსი:

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

მთლიანობის შეზღუდვა

2.1.8.1.1. მთლიანობის შეზღუდვა:

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

2.1.8.1.2. 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

2.1.8.1.3. მთლიანობის შეზღუდვა:

- ა. შეიძლება გამოიყენებულ იქნეს სვეტის ან ცხრილის დონეზე

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

2.1.8.1.4. 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

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

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

2.1.8.1.6. არსებობს მთლიანობის შეზღუდვის:

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

2.1.8.1.7. NULL მთლიანობის შეზღუდვა მოქმედებს:

- ა. მხოლოდ სვეტის დონეზე
- ბ. სვეტისა და მომხმარებლის ტიპის დონეზე
- გ. მონაცემთა ბაზის დონეზე

2.1.8.1.8. 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

2.1.8.1.9. 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

2.1.8.1.10. 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

2.1.8.1.11. NULL არის სპეციალური მნიშვნელობა, რომელიც აღნიშნავს:

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

2.1.8.1.12. NULL არ არის იგივე, რაც:

- ა. მნიშვნელობის არყოფნა

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

2.1.8.1.13. NULL მნიშვნელობა შეიძლება შეფასდეს, როგორც:

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

2.1.8.1.14. 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

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

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

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

- ა. აკრძალება და შეტყობინება გაიცემა შეცდომის შესახებ
- ბ. არ აკრძალება და შეტყობინება გაიცემა შეცდომის შესახებ
- გ. არ აკრძალება და შეტყობინება არ გაიცემა შეცდომის შესახებ

2.1.8.1.17. ერთი სვეტისთვის შეგვიძლია შევქმნათ:

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

2.1.8.1.18. 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

2.1.8.1.19. UNIQUE მთლიანობის შეზღუდვა:

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

2.1.8.1.20. PRIMARY KEY შეზღუდვა მოქმედებს:

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

2.1.8.1.21. 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

2.1.8.1.22. 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

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

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

2.1.8.1.24. 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

2.1.8.1.25. 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

2.1.8.1.26. ცხრილში შეგვიძლია შევქმნათ მხოლოდ:

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

2.1.8.1.27. FOREIGN KEY შეზღუდვა იქმნება:

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

2.1.8.1.28. დამოკიდებული ცხრილის გარე გასაღები უკავშირდება:

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

- 2.1.8.1.29. 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

- 2.1.8.1.30. 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

ნაგულისხმები მნიშვნელობები

- 2.1.9.1.1. ნაგულისხმებია 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

- 2.1.9.1.2. ნაგულისხმები 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

ფუნქციები

- 2.1.10.1.1. 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

- 2.1.10.1.2. არსებობს ფუნქციების ორი ტიპი:
- ა. მხოლოდ მომხმარებლის ფუნქციები
 - ბ. მხოლოდ ჩადგმული ფუნქციები
 - გ. ჩადგმული ფუნქციები და მომხმარებლის ფუნქციები

- 2.1.10.1.3. 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

2.2.0.1.1. სერვერის სისტემური მონაცემთა ბაზებია:

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

2.2.0.1.2. სისტემურ ცხრილებთან მუშაობა დასაშვებია:

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

2.2.0.1.3. სისტემურ ცხრილებში მონაცემების შეცვლა:

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

Master სისტემური მონაცემთა ბაზა

2.2.1.1.1. 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

2.2.1.1.2. master მონაცემთა ბაზა შემდეგი ფაილებისაგან შედგება:

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

2.2.1.1.3. master სისტემურ მონაცემთა ბაზაში მომხმარებლის ობიექტების შექმნა:

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

Tempdb სისტემური მონაცემთა ბაზა

2.2.2.1.1. tempdb მონაცემთა ბაზა:

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

2.2.2.1.2. tempdb მონაცემთა ბაზა ორი ფაილისგან შედგება:

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

2.2.2.1.3. 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

Model სისტემური მონაცემთა ბაზა

2.2.3.1.1. model სისტემური მონაცემთა ბაზა ორი ფაილისგან შედგება:

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

MsdB სისტემური მონაცემთა ბაზა

2.2.4.1.1. 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

2.2.4.1.2. msdb სისტემური მონაცემთა ბაზა გამოიყენება:

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

2.2.4.1.3. msdb ბაზა ორი ფაილისგან შედგება:

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

ფაილები და ფაილების ჯგუფები

2.3.0.1.1. თითოეული მონაცემთა ბაზა მინიმუმ:

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

2.3.0.1.2. თითოეული მონაცემთა ბაზა შემდეგი ფაილებისაგან შედგება:

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

2.3.0.1.3. მონაცემთა ბაზის შესაძლებელი შემდეგი ტიპის ფაილები გამოიყენება:

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

2.3.0.1.4. მონაცემთა ბაზაში გამოყენებულ თითოეულ ფაილს აქვს:

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

2.3.0.1.5. მონაცემთა ბაზის:

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

2.3.0.1.6. 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

2.3.0.1.7. 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

2.3.0.1.8. 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

2.3.0.1.9. 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

2.3.0.1.10. მონაცემთა ბაზაში ფაილების:

- ა. სამი ჯგუფია ნაგულისხმევი
- ბ. ორი ჯგუფია ნაგულისხმევი
- გ. ერთ-ერთი ჯგუფია ნაგულისხმევი

2.3.0.1.11. 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

2.3.0.1.12. 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

2.3.0.1.13. 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

2.3.0.1.14. ფაილების მხოლოდ:

- ა. ერთი ჯგუფი შეიძლება იყოს ნაგულისხმევი
- ბ. ორი ჯგუფი შეიძლება იყოს ნაგულისხმევი
- გ. სამი ჯგუფი შეიძლება იყოს ნაგულისხმევი

2.3.0.1.15. ნებისმიერი ფაილი შეიძლება ჩართული იყოს ფაილების მხოლოდ:

- ა. სამ ჯგუფში
- ბ. ორ ჯგუფში
- გ. ერთ ჯგუფში

2.3.0.1.16. ფაილების ჯგუფის განსაზღვრა ტრანზაქციების ჟურნალის ფაილებისთვის:

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

2.3.0.1.17. 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 ბაზის ფიზიკური სტრუქტურა

2.3.1.1.1. სერვერზე მონაცემების შენახვის ძირითადი ერთეულია:

- ა. კატალოგი
- ბ. ფაილი
- გ. გვერდი

2.3.1.1.2. მონაცემთა ბაზაში მონაცემების ფაილისთვის გამოყოფილი დისკური სივრცე ლოგიკურად იყოფა:

- ა. ფაილებად
- ბ. გვერდებად
- გ. კატალოგებად

2.3.1.1.3. შეტანა-გამოტანის ოპერაციები სრულდება:

- ა. გვერდის დონეზე
- ბ. ფაილის დონეზე
- გ. კატალოგის დონეზე

2.3.1.1.4. გვერდები:

- ა. უწყვეტად არ ინომრება 0-დან n -მდე
- ბ. უწყვეტად ინომრება 0-დან n -მდე
- გ. უწყვეტად ინომრება 10-დან $n-10$ -მდე

2.3.1.1.5. სერვერი წერს და კითხულობს მონაცემების:

- ა. ორ-ორ გვერდს
- ბ. ნახევარ გვერდებს
- გ. მთელ გვერდებს

2.3.1.1.6. გვერდი არის:

- ა. მინიმალური ბლოკი, რომელსაც სერვერი გამოყოფს მონაცემების შესანახად მონაცემთა ბაზაში
- ბ. მაქსიმალური ბლოკი, რომელსაც სერვერი გამოყოფს მონაცემების შესანახად მონაცემთა ბაზაში
- გ. მინიმალური ბლოკი, რომელსაც სერვერი არ გამოყოფს მონაცემების შესანახად მონაცემთა ბაზაში

2.3.1.1.7. თითოეული ფაილი:

- ა. კატალოგებისაგან შედგება
- ბ. გვერდებისაგან არ შედგება
- გ. გვერდებისაგან შედგება

2.3.1.1.8. გვერდების უფრო ეფექტური მართვისთვის სერვერი:

- ა. არ იყენებს გვერდების ჯგუფებს ანუ ექსტენტებს
- ბ. იყენებს გვერდების ჯგუფებს ანუ ექსტენტებს
- გ. იყენებს ექსტენტების ჯგუფებს

2.3.1.1.9. ექსტენტი:

- ა. გამოიყენება მონაცემებისა და ინდექსების შესანახად
- ბ. არ გამოიყენება მონაცემებისა და ინდექსების შესანახად
- გ. გამოიყენება მონაცემების, კატალოგებისა და ინდექსების შესანახად

2.3.1.1.10. თითოეულ ექსტენტში:

- ა. 8 მეზობლად მდებარე გვერდია
- ბ. 4 მეზობლად მდებარე გვერდია
- გ. 2 მეზობლად მდებარე გვერდია

2.3.1.1.11. ექსტენტებში ინახება:

- ა. ზოგიერთი გვერდი
- ბ. ყველა გვერდი
- გ. არც ერთი გვერდი

2.3.1.1.12. 1 გვერდის ზომაა:

- ა. 1 კილობაიტი
- ბ. 8 კილობაიტი
- გ. 16 კილობაიტი

2.3.1.1.13. 1 მეგაბაიტი შედგება:

- ა. 1 გვერდისაგან
- ბ. 16 გვერდისაგან
- გ. 128 გვერდისაგან

2.3.1.1.14. თითოეული გვერდი იწყება:

- ა. 916-ბაიტისანი სათაურისაგან
- ბ. 196-ბაიტისანი სათაურისაგან
- გ. 96-ბაიტისანი სათაურისაგან

2.3.1.1.15. გვერდის სათაური:

- ა. გამოიყენება გვერდის შესახებ სისტემური ინფორმაციის შესანახად
- ბ. არ გამოიყენება გვერდის შესახებ სისტემური ინფორმაციის შესანახად
- გ. გამოიყენება გვერდის შესახებ არასისტემური ინფორმაციის შესანახად

2.3.1.1.16. გვერდის სათაური:

- ა. არ შეიცავს გვერდის ნომერს, გვერდის ტიპს, გვერდზე ცარიელი უბნის ზომასა და გვერდის მფლობელი ობიექტის საიდენტიფიკაციო ნომერს
- ბ. შეიცავს გვერდის ნომერს, გვერდის ტიპს, გვერდზე ცარიელი უბნის ზომასა და გვერდის მფლობელი ობიექტის საიდენტიფიკაციო ნომერს
- გ. შეიცავს გვერდის ნომერს, გვერდის ტიპს, გვერდზე ცარიელი უბნის ზომასა და გვერდის მფლობელი ობიექტის სახელს

2.3.1.1.17. არსებობს გვერდების შემდეგი ტიპები:

- ა. Page Free Space, Differential Changed Map, SELECT
- ბ. Data, Index, Text/Image, Global Allocation Map
- გ. Index Allocation Map, Bulk Changed Map, UPDATE

2.3.1.1.18. ექსტენტი იკავებს:

- ა. 64 კილობაიტს
- ბ. 16 კილობაიტს
- გ. 4 კილობაიტს

2.3.1.1.19. არსებობს ორი ტიპის ექსტენტი:

- ა. model და Uniform
- ბ. Mixed და model
- გ. Mixed და Uniform

2.3.1.1.20. Mixed ტიპის ექსტენტები:

- ა. შეიძლება შეიცავდეს კატალოგებს
- ბ. არ შეიძლება შეიცავდეს გვერდებს, რომლებიც სხვადასხვა მფლობელს ეკუთვნის
- გ. შეიძლება შეიცავდეს გვერდებს, რომლებიც სხვადასხვა მფლობელს ეკუთვნის

2.3.1.1.21. როცა ერთი მფლობელის მონაცემების შესანახად გამოყოფილია 8 გვერდი Mixed ტიპის სხვადასხვა ექსტენტში, მაშინ ისინი:

- *ა. ავტომატურად ერთიანდება Uniform ტიპის ექსტენტში
- ბ. ავტომატურად არ ერთიანდება Uniform ტიპის ექსტენტში
- გ. ავტომატურად ერთიანდება Mixed ტიპის ექსტენტში

მონაცემთა ბაზების შექმნა, წაშლა და სახელის შეცვლა

2.4.0.1.1. სერვერთან შეერთებისას კავშირი:

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

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

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

2.4.0.1.3. სხვა მონაცემთა ბაზაზე გადართვისათვის გამოიყენება:

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

მონაცემთა ბაზის შექმნა

2.4.1.1.1. 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

2.4.1.1.2. ერთ სერვერზე შესაძლებელია მაქსიმუმ:

- ა. 256 მონაცემთა ბაზის შექმნა
- ბ. 32 767 მონაცემთა ბაზის შექმნა
- გ. 128 მონაცემთა ბაზის შექმნა

2.4.1.1.3. 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

2.4.1.1.4. მონაცემთა ბაზის შემქმნელი მომხმარებელი ავტომატურად:

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

2.4.1.1.5. 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

2.4.1.1.6. 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

2.4.1.1.7. 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

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

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

2.4.1.1.9. მონაცემთა ბაზის სახელი:

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

2.4.1.1.10. 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

2.4.1.1.11. 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

2.4.1.1.12. 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

2.4.1.1.13. 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

2.4.1.1.14. 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

2.4.1.1.15. 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

2.4.1.1.16. CREATE DATABASE ბრძანების LOG ON არგუმენტი:

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

2.4.1.1.17. 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

2.4.1.1.18. 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

2.4.1.1.19. 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

2.4.1.1.20. 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

2.4.1.1.21. CREATE DATABASE ბრძანების NAME არგუმენტი განსაზღვრავს:

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

2.4.1.1.22. CREATE DATABASE ბრძანების FILENAME არგუმენტი განსაზღვრავს:

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

2.4.1.1.23. CREATE DATABASE ბრძანების SIZE არგუმენტი განსაზღვრავს:

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

2.4.1.1.24. CREATE DATABASE ბრძანების FILEGROWTH არგუმენტი განსაზღვრავს:

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

2.4.1.1.25. ნაზარდის ზომა:

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

2.4.1.1.26. ავტომატურად ნაზარდის ზომა აიღება:

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

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

- ა. 10%-ით
- ბ. 20%-ით
- გ. 30%-ით

2.4.1.1.28. CREATE DATABASE ბრძანების MAXSIZE არგუმენტი განსაზღვრავს:

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

2.4.1.1.29. ფაილის მაქსიმალური ზომა ეთითება:

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

2.4.1.1.30. ერთი ფაილის ზომის შეზღუდვა:

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

2.4.1.1.31. 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

2.4.1.1.32. 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

2.4.1.1.33. თუ ფაილის ზომა არ უნდა შეიზღუდოს, მაშინ:

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

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

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

2.4.1.1.35. CREATE DATABASE ბრძანების FILEGROUP არგუმენტი განსაზღვრავს:

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

მონაცემთა ბაზის წაშლა

2.4.2.1.1. მონაცემთა ბაზის წასაშლელად გამოიყენება:

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

2.4.2.1.2. ერთი DROP DATABASE ბრძანებით შესაძლებელია:

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

მონაცემთა ბაზის სახელი შეცვლა

2.4.3.1.1. მონაცემთა ბაზის სახელის შესაცვლელად გამოიყენება:

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

2.4.3.1.2. 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

მონაცემთა ბაზის მიერთება და გამორთვა

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

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

2.5.1.1.2. მონაცემთა ბაზების მიერთებისა და გამორთვის ოპერაციების შესრულება შეუძლია:

- ა. არც ერთ მომხმარებელს
- ბ. ნებისმიერ მომხმარებელს
- გ. მხოლოდ სერვერის sysadmin როლის წევრებს

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

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

2.5.1.1.4. 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

2.5.1.1.5. სერვერიდან გამორთვის შემდეგ:

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

2.5.1.1.6. მონაცემთა ბაზის გამოსართავად გამოიყენება:

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

2.5.1.1.7. sp_detach_db შენახული პროცედურის 'skipchecks' არგუმენტი:

- ა. მართავს სტატისტიკის გაახლებას მონაცემთა ბაზასთან მიერთების დროს
- ბ. არ მართავს სტატისტიკის გაახლებას მონაცემთა ბაზის გამორთვის დროს
- გ. მართავს სტატისტიკის გაახლებას მონაცემთა ბაზის გამორთვის დროს

2.5.1.1.8. თუ p_detach_db შენახული პროცედურის 'skipchecks' არგუმენტი იღებს TRUE მნიშვნელობას, მაშინ:

- ა. სტატისტიკის გაახლება არ შესრულდება
- ბ. სტატისტიკის გაახლება შესრულდება
- გ. სტატისტიკის გაახლება დამთავრდება

2.5.1.1.9. თუ p_detach_db შენახული პროცედურის 'skipchecks' არგუმენტი იღებს FALSE მნიშვნელობას, მაშინ:

- ა. სტატისტიკის გაახლება არ შესრულდება
- ბ. სტატისტიკის გაახლება შესრულდება
- გ. სტატისტიკის გაახლება გადაიდება

2.5.1.1.10. თუ p_detach_db შენახული პროცედურის 'skipchecks' არგუმენტი მითითებული არ არის, მაშინ იგულისხმება:

- ა. NULL
- ბ. FALSE
- გ. TRUE

2.5.1.1.11. 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

2.5.1.1.12. სერვერთან მონაცემთა ბაზის მისაერთებლად გამოიყენება:

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

2.5.1.1.13. 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

2.5.1.1.14. სერვერთან მონაცემთა ბაზის მიერთებისას ტრანზაქციების ჟურნალის მიერთება:

- ა. აუცილებელი არ არის
- ბ. აუცილებელია
- გ. ზოგჯერ არის აუცილებელი

2.5.1.1.15. 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

მონაცემთა ბაზების მართვა

ფლობის უფლების გადაცემა

2.6.1.1.1. მონაცემთა ბაზის ფლობის უფლების გადასაცემად ერთი მომხმარებლიდან მეორისთვის გამოიყენება:

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

2.6.1.1.2. sp_changedbowner შენახულ პროცედურაში მითითებული სააღრიცხვო ჩანაწერი:

- ა. არ უნდა მიმართავდეს მონაცემთა ბაზას არც ფსევდონიმით, არც მონაცემთა ბაზის მომხმარებელში ასახვით
- ბ. უნდა მიმართავდეს მონაცემთა ბაზას ფსევდონიმით ან მონაცემთა ბაზის მომხმარებელში ასახვით
- გ. უნდა მიმართავდეს მონაცემთა ბაზას ფსევდონიმით

2.6.1.1.3. sp_changedbowner შენახულ პროცედურაში ალამი არგუმენტი იღებს:

- ა. მხოლოდ TRUE მნიშვნელობას

- ბ. TRUE ან FALSE მნიშვნელობას
- გ. მხოლოდ FALSE მნიშვნელობას

2.6.1.1.4. თუ sp_changedbowner შენახულ პროცედურაში ალამი არგუმენტი იღებს TRUE მნიშვნელობას, მაშინ:

- ა. მონაცემთა ბაზის ძველი მფლობელის სააღრიცხვო ჩანაწერი აისახება ახალი მფლობელის სააღრიცხვო ჩანაწერში
- ბ. ძველი მფლობელის სააღრიცხვო ჩანაწერი უნდა წაიშალოს
- გ. ყველა არსებული dbi აისახება მონაცემთა ბაზის ახალი მფლობელის სააღრიცხვო ჩანაწერში

2.6.1.1.5. თუ sp_changedbowner შენახულ პროცედურაში ალამი არგუმენტი იღებს FALSE მნიშვნელობას, მაშინ:

- ა. მონაცემთა ბაზის ძველი მფლობელის სააღრიცხვო ჩანაწერი აისახება ახალი მფლობელის სააღრიცხვო ჩანაწერში
- ბ. ძველი მფლობელის სააღრიცხვო ჩანაწერი უნდა წაიშალოს
- გ. ყველა არსებული dbi აისახება მონაცემთა ბაზის ახალი მფლობელის სააღრიცხვო ჩანაწერში

2.6.1.1.6. თუ sp_changedbowner შენახულ პროცედურაში ალამი არგუმენტი იღებს NULL მნიშვნელობას, მაშინ:

- ა. ძველი მფლობელის სააღრიცხვო ჩანაწერი უნდა წაიშალოს
- ბ. მონაცემთა ბაზის ძველი მფლობელის სააღრიცხვო ჩანაწერი აისახება ახალი მფლობელის სააღრიცხვო ჩანაწერში
- გ. ყველა არსებული dbi აისახება მონაცემთა ბაზის ახალი მფლობელის სააღრიცხვო ჩანაწერში

2.6.1.1.7. sp_changedbowner შენახული პროცედურა:

- ა. უნდა შესრულდეს იმ მონაცემთა ბაზის კონტექსტში, რომლის მფლობელიც უნდა შეიცვალოს
- ბ. არ უნდა შესრულდეს იმ მონაცემთა ბაზის კონტექსტში, რომლის მფლობელიც უნდა შეიცვალოს
- გ. უნდა შესრულდეს იმ მონაცემთა ბაზის კონტექსტში, რომლის მფლობელიც არ უნდა შეიცვალოს

მონაცემთა ბაზის შეკუმშვა

2.6.2.1.1. სერვერს შეუძლია მონაცემთა ბაზის ზომა:

- ა. გაზარდოს ან შეამციროს
- ბ. მხოლოდ გაზარდოს
- გ. მხოლოდ შეამციროს

2.6.2.1.2. შემცირება საჭიროა მაშინ, როცა:

- ა. მონაცემთა ბაზიდან წავშალებთ მცირე მოცულობის მონაცემები
- ბ. მონაცემთა ბაზიდან წავშალებთ დიდი მოცულობის მონაცემები და ოპერაციულ სისტემას

უნდა დაუბრუნდეს გამოუყენებელი დისკური სივრცე

გ. მონაცემთა ბაზიდან არ წავშაღეთ მონაცემები

2.6.2.1.3. მონაცემთა ბაზის ზომის შემცირების პროცესს ეწოდება მონაცემთა ბაზის:

ა. შეკუმშვა

ბ. ჩამოჭრა

გ. გაზრდა

2.6.2.1.4. მონაცემთა ბაზის შეკუმშვის დროს სრულდება მონაცემთა ბაზის ფაილების ზომების:

ა. ჩამოჭრა

ბ. გაზრდა

გ. შემცირება

2.6.2.1.5. მონაცემთა ბაზის შეკუმშვა უმჯობესია შევასრულოთ მაშინ, როცა ყველაზე:

ა. მეტია მომხმარებლების აქტიურობა

ბ. ნაკლებია მომხმარებლების აქტიურობა

გ. მაქსიმალურია მომხმარებლების აქტიურობა

2.6.2.1.6. მონაცემთა ბაზის ზომის ავტომატურად შემცირება სრულდება მაშინ, როცა:

ა. სერვერი მონაცემთა ბაზაში აღმოაჩენს დიდი ზომის თავისუფალ სივრცეს

ბ. სერვერი მონაცემთა ბაზაში აღმოაჩენს მცირე ზომის თავისუფალ სივრცეს

გ. სერვერი მონაცემთა ბაზაში ვერ აღმოაჩენს დიდი ზომის თავისუფალ სივრცეს

2.6.2.1.7. მონაცემთა ბაზის ზომის ავტომატურად გაზრდის ან შემცირების ნების დასართავად გამოიყენება:

ა. CREATE DATABASE ბრძანება

ბ. ALTER DATABASE ბრძანება

გ. DROP DATABASE ბრძანება

2.6.2.1.8. მონაცემთა ბაზის ზომის შემცირება შეიძლება:

ა. DBCC ბრძანების გამოყენებით

ბ. CREATE DATABASE ბრძანების გამოყენებით

გ. DROP DATABASE ბრძანების გამოყენებით

2.6.2.1.9. DBCC ბრძანების სინტაქსია:

ა. DBCC SHRINKDATABASE(მონაცემთა_ბაზის_სახელი[,პროცენტი]
[,,{NOTRUNCATE|TRUNCATEONLY}])

ბ. DBCC SHRINKDATABASE(მონაცემთა_ბაზის_სახელი[,,{NOTRUNCATE|TRUNCATEONLY}])

გ. DBCC SHRINKDATABASE(მონაცემთა_ბაზის_სახელი[,პროცენტი][,|TRUNCATEONLY])

2.6.2.1.10. DBCC ბრძანების „პროცენტი“ არგუმენტი:

ა. არის თავისუფალი სივრცის პროცენტული ზომა, რომელიც არ უნდა დარჩეს მონაცემთა ბაზაში მისი შეკუმშვის შემდეგ

ბ. არ არის თავისუფალი სივრცის პროცენტული ზომა, რომელიც უნდა დარჩეს მონაცემთა ბაზაში მისი შეკუმშვის შემდეგ

გ. არის თავისუფალი სივრცის პროცენტული ზომა, რომელიც უნდა დარჩეს მონაცემთა ბაზაში მისი შეკუმშვის შემდეგ

2.6.2.1.11. თუ DBCC ბრძანების „პროცენტი“ არგუმენტში მითითებული ზომა აღემატება თავისუფალი სივრცის მიმდინარე პროცენტის ზომას, მაშინ ფაილის ზომის შემცირება:

- ა. ნაწილობრივ შესრულდება
- ბ. შესრულდება
- გ. არ შესრულდება

2.6.2.1.12. თუ მითითებულია DBCC ბრძანების NOTRUNCATE არგუმენტი, მაშინ:

- ა. ფაილის თავისუფალი სივრცე უბრუნდება ოპერაციულ სისტემას
- ბ. ფაილის თავისუფალი სივრცე არ უბრუნდება ოპერაციულ სისტემას, არამედ რეზერვირდება შემდგომი გამოყენებისათვის
- გ. ფაილის თავისუფალი სივრცე არ უბრუნდება ოპერაციულ სისტემას და არც რეზერვირდება შემდგომი გამოყენებისათვის

2.6.2.1.13. თუ მითითებულია DBCC ბრძანების NOTRUNCATE არგუმენტი, მაშინ:

- ა. ფაილის ზომის ფიზიკურად შემცირება არ სრულდება
- ბ. ფაილის ზომის ფიზიკურად შემცირება სრულდება
- გ. ფაილის ზომის ფიზიკურად შემცირება ნაწილობრივ სრულდება

2.6.2.1.14. თუ მითითებულია DBCC ბრძანების TRUNCATEONLY არგუმენტი, მაშინ:

- ა. სერვერი ფაილიდან არ შლის მთელ თავისუფალ სივრცეს, დაწყებული უკანასკნელად გამოყენებული ექსტენტიდან
- ბ. სერვერი ფაილიდან შლის მთელ თავისუფალ სივრცეს, დაწყებული პირველი ექსტენტიდან
- გ. სერვერი ფაილიდან შლის მთელ თავისუფალ სივრცეს, დაწყებული უკანასკნელად გამოყენებული ექსტენტიდან

2.6.2.1.15. თუ მითითებულია DBCC ბრძანების TRUNCATEONLY არგუმენტი, მაშინ:

- ა. დროს „პროცენტი“ არგუმენტის მნიშვნელობა არ იგნორირდება
- ბ. დროს „პროცენტი“ არგუმენტის მნიშვნელობა იგნორირდება
- გ. დროს „პროცენტი“ არგუმენტის მნიშვნელობა ნაწილობრივ იგნორირდება

2.6.2.1.16. DBCC SHRINKDATABASE ბრძანების გამოყენებით მონაცემთა ბაზის შეკუმშვის უფლება აქვთ:

- ა. მხოლოდ მონაცემთა ბაზის dbowner ფიქსირებული როლის წევრებს
- ბ. მხოლოდ სერვერის sysadmin ფიქსირებული როლის წევრებს
- გ. მხოლოდ სერვერის sysadmin და მონაცემთა ბაზის dbowner ფიქსირებული როლის წევრებს

2.6.2.1.17. მონაცემთა ბაზის რომელიმე ფაილის ზომის შესამცირებლად გამოიყენება ბრძანება:

- ა. DBCC SHRINKDATABASE
- ბ. DBCC SHRINKFILE
- გ. CREATE DATABASE

2.6.2.1.18. DBCC SHRINKFILE ბრძანების სინტაქსია:

- ა. DBCC SHRINKFILE({ფაილის_სახელი|ფაილის_იდენტიფიკატორი} {[,EMPTYFILE] | [,ზომა][,{NOTTRUNCATE|TRUNCATEONLY}]})
- ბ. DBCC SHRINKFILE({ფაილის_სახელი| {[,EMPTYFILE] | [,ზომა][,{NOTTRUNCATE|TRUNCATEONLY}]})
- გ. DBCC SHRINKFILE({ფაილის_სახელი|ფაილის_იდენტიფიკატორი} {[,EMPTYFILE] | [,ზომა])

2.6.2.1.19. DBCC SHRINKFILE ბრძანების 'ფაილის_სახელი' არგუმენტი:

- ა. არ არის შესაკუმში ფაილის სახელი
- ბ. არის შესაკუმში ინდექსის სახელი
- გ. არის შესაკუმში ფაილის სახელი

2.6.2.1.20. DBCC SHRINKFILE ბრძანების 'ფაილის_იდენტიფიკატორი' არგუმენტი:

- ა. არ არის შესაკუმში ფაილის იდენტიფიკატორი
- ბ. არის შესაკუმში ფაილის იდენტიფიკატორი
- გ. არის შესაკუმში ინდექსის იდენტიფიკატორი

2.6.2.1.21. ფაილის საიდენტიფიკაციო ნომრის მისაღებად გამოიყენება:

- ა. FILE_ID('ფაილის_სახელი') ბრძანება
- ბ. DBCC SHRINKFILE ბრძანება
- გ. DROP DATABASE

2.6.2.1.22. მონაცემთა ბაზის ფაილების სახელები, მათი საიდენტიფიკაციო ნომრები, ლოგიკური და ფიზიკური სახელები ინახება მიმდინარე მონაცემთა ბაზის:

- ა. sysdatabases წარმოდგენაში
- ბ. sysfiles წარმოდგენაში
- გ. systables წარმოდგენაში

2.6.2.1.23. DBCC SHRINKFILE ბრძანების „ზომა“ არგუმენტი არის:

- ა. ფაილის ზომა, რომელიც არ ექნება შეკუმშვის შემდეგ
- ბ. ფაილის ზომა, რომელიც ექნება შეკუმშვამდე
- გ. ფაილის ზომა, რომელიც ექნება შეკუმშვის შემდეგ

2.6.2.1.24. თუ DBCC SHRINKFILE ბრძანების „ზომა“ არგუმენტი არ არის მითითებული, მაშინ:

- ა. ფაილი შეიკუმშება მინიმალურად შესაძლო ზომამდე
- ბ. ფაილი შეიკუმშება მაქსიმალურად შესაძლო ზომამდე
- გ. ფაილი შეიკუმშება საშუალო ზომამდე

2.6.2.1.25. თუ DBCC SHRINKFILE ბრძანების „ზომა“ არგუმენტის მნიშვნელობა ნაკლებია ფაილის მინიმალურ ზომაზე, მაშინ:

- ა. ფაილის შეკუმშვა შესრულდება მის საშუალო ზომამდე
- ბ. ფაილის შეკუმშვა შესრულდება მის მაქსიმალურად შესაძლო ზომამდე
- გ. ფაილის შეკუმშვა შესრულდება მის მინიმალურად შესაძლო ზომამდე

2.6.2.1.26. თუ ფაილის ზომა შეკუმშვის შემდეგ ხდება მის საწყის ზომაზე ნაკლები, მაშინ:

- ა. ფაილის ახალი ზომა გახდება მისი მაქსიმალური ზომა
- ბ. ფაილის ახალი ზომა გახდება მისი მინიმალური ზომა
- გ. ფაილის ახალი ზომა გახდება მისი საშუალო ზომა

2.6.2.1.27. 25 მეგაბაიტი ზომის ფაილი შეიცავს 19 მეგაბაიტ მონაცემებს და ხდება მისი შეკუმშვის მცდელობა 15 მეგაბაიტამდე. ამ შემთხვევაში ფაილის ზომა გახდება:

- ა. 15 მეგაბაიტი
- ბ. 19 მეგაბაიტი
- გ. 25 მეგაბაიტი

2.6.2.1.28. თუ მითითებულია DBCC SHRINKFILE ბრძანების EMPTYFILE არგუმენტი, მაშინ:

- ა. სერვერს მონაცემები ამ ფაილიდან გადააქვს ამავე ჯგუფის სხვა ფაილში
- ბ. სერვერს მონაცემები ამ ფაილიდან არ გადააქვს ამავე ჯგუფის სხვა ფაილში
- გ. სერვერს მონაცემები ამ ფაილიდან ხანდახან გადააქვს ამავე ჯგუფის სხვა ფაილში

2.6.2.1.29. თუ მითითებულია DBCC SHRINKFILE ბრძანების EMPTYFILE არგუმენტი, მაშინ:

- ა. სერვერი ამ ფაილს ხანდახან დაუმატებს ახალ მონაცემს
- ბ. სერვერი ამ ფაილს დაუმატებს ახალ მონაცემს
- გ. სერვერი ამ ფაილს არ დაუმატებს ახალ მონაცემს

2.6.2.1.30. თუ მითითებულია DBCC SHRINKFILE ბრძანების EMPTYFILE არგუმენტი, მაშინ ასეთი ფაილის წაშლა:

- ა. შესაძლებელია ბრძანებით - ALTER DATABASE REMOVE FILE
- ბ. შეუძლებელია ბრძანებით - ALTER DATABASE REMOVE FILE
- გ. შესაძლებელია ბრძანებით - CREATE DATABASE REMOVE FILE

2.6.2.1.31. თუ მითითებულია DBCC SHRINKFILE ბრძანების NOTRUNCATE არგუმენტი, მაშინ:

- ა. სერვერი ოპერაციულ სისტემას უბრუნებს ფაილის გამოთავისუფლებულ სივრცეს
- ბ. სერვერი ოპერაციულ სისტემას არ უბრუნებს ფაილის გამოთავისუფლებულ სივრცეს
- გ. სერვერი ოპერაციულ სისტემას არ უბრუნებს კატალოგის გამოთავისუფლებულ სივრცეს

2.6.2.1.32. თუ DBCC SHRINKFILE ბრძანების NOTRUNCATE არგუმენტი არ არის მითითებული, მაშინ:

- ა. გამოთავისუფლებული სივრცე არ უბრუნდება ოპერაციულ სისტემას
- ბ. გამოთავისუფლებული სივრცე უბრუნდება ოპერაციულ სისტემას
- გ. გამოთავისუფლებული სივრცე უბრუნდება სერვერს

2.6.2.1.33. თუ მითითებულია DBCC SHRINKFILE ბრძანების TRUNCATEONLY არგუმენტი, მაშინ:

- ა. სერვერი ჩამოჭრის ფაილის ნაწილს, დაწყებული პირველი გვერდიდან ფაილის ბოლომდე
- ბ. სერვერი ჩამოჭრის ფაილის ნაწილს, დაწყებული უკანასკნელად გამოყენებული გვერდიდან ფაილის ბოლომდე
- გ. სერვერი არ ჩამოჭრის ფაილის ნაწილს, დაწყებული უკანასკნელად გამოყენებული გვერდიდან ფაილის ბოლომდე

2.6.2.1.34. თუ მითითებულია DBCC SHRINKFILE ბრძანების TRUNCATEONLY არგუმენტი, მაშინ:

- ა. „ზომა“ არგუმენტი იგნორირდება
- ბ. „ზომა“ არგუმენტი არ იგნორირდება
- გ. „ზომა“ არგუმენტი ხანდახან იგნორირდება

2.6.2.1.35. DBCC SHRINKFILE ბრძანება:

- ა. უნდა შესრულდეს იმ მონაცემთა ბაზის კონტექსტში, რომლის ფაილის ზომის შემცირებაც არ გვინდა
- ბ. არ უნდა შესრულდეს იმ მონაცემთა ბაზის კონტექსტში, რომლის ფაილის ზომის შემცირებაც გვინდა
- გ. უნდა შესრულდეს იმ მონაცემთა ბაზის კონტექსტში, რომლის ფაილის ზომის შემცირებაც გვინდა

მონაცემთა ბაზის თვისებების მართვა

2.6.3.1.1. მონაცემთა ბაზას აქვს:

- ა. მხოლოდ ლოგიკური პარამეტრები
- ბ. მხოლოდ ფიზიკური პარამეტრები
- გ. ლოგიკური და ფიზიკური პარამეტრები

2.6.3.1.2. მონაცემთა ბაზის ლოგიკური პარამეტრებია:

- ა. ტრანზაქციების ჟურნალის ავტომატურად ჩამოჭრის, სტატისტიკის ავტომატურად შექმნისა და გაახლების შესაძლებლობა
- ბ. ფაილის ზომა
- გ. ჩადგმული ტრიგერების შესრულების შესაძლებლობა და ფაილის ზომა

2.6.3.1.3. მონაცემთა ბაზის კონფიგურირების პარამეტრების შესაცვლელად გამოიყენება:

- ა. DBCC SHRINKFILE ბრძანება
- ბ. ALTER DATABASE ბრძანება
- გ. DROP DATABASE ბრძანება

2.6.3.1.4. ALTER DATABASE ბრძანების საშუალებით შესაძლებელია:

- ა. მონაცემთა ბაზის ან მასთან დაკავშირებული ფაილებისა და ფაილების ჯგუფის შეცვლა, ფაილებისა და ჯგუფების დამატება და წაშლა მონაცემთა ბაზიდან, მონაცემთა ბაზის ან მისი ფაილებისა და ჯგუფების ატრიბუტების შეცვლა, მონაცემთა ბაზის რეჟიმების დაყენება
- ბ. მონაცემთა ბაზის შექმნა, მონაცემთა ბაზის ან მისი ფაილებისა და ჯგუფების ატრიბუტების შეცვლა, მონაცემთა ბაზის რეჟიმების დაყენება
- გ. მონაცემთა ბაზის წაშლა, მონაცემთა ბაზის ან მასთან დაკავშირებული ფაილებისა და ფაილების ჯგუფის შეცვლა

2.6.3.1.5. ALTER DATABASE ბრძანების MODIFY NAME არგუმენტი განსაზღვრავს:

- ა. მონაცემთა ბაზის ძველ სახელს
- ბ. მონაცემთა ბაზის შუალედურ სახელს
- გ. მონაცემთა ბაზის ახალ სახელს

2.6.3.1.6. ALTER DATABASE ბრძანების ADD FILE არგუმენტი:

- ა. მონაცემთა ბაზიდან ფაილს შლის
- ბ. მონაცემთა ბაზას ფაილს უმატებს
- გ. არაფერს არ აკეთებს

2.6.3.1.7. ALTER DATABASE ბრძანების TO FILEGROUP არგუმენტი:

- ა. განსაზღვრავს ფაილების ჯგუფს, რომელსაც უნდა დაემატოს მითითებული ფაილი
- ბ. განსაზღვრავს ფაილების ჯგუფს, რომელსაც არ უნდა დაემატოს მითითებული ფაილი
- გ. არ განსაზღვრავს ფაილების ჯგუფს, რომელსაც უნდა დაემატოს მითითებული ფაილი

2.6.3.1.8. თუ ALTER DATABASE ბრძანების TO FILEGROUP არგუმენტი იღებს DEFAULT მნიშვნელობას, მაშინ:

- ა. ფაილი დაემატება რომელიმე ჯგუფს
- ბ. ფაილი არ დაემატება ავტომატურად განსაზღვრულ ჯგუფს
- გ. ფაილი დაემატება ავტომატურად განსაზღვრულ ჯგუფს

2.6.3.1.9. ფაილების ჯგუფების შესახებ ინფორმაცია მოთავსებულია:

- ა. მიმდინარე მონაცემთა ბაზის sys.files წარმოდგენაში
- ბ. მიმდინარე მონაცემთა ბაზის sys.filegroups წარმოდგენაში
- გ. მიმდინარე მონაცემთა ბაზის sys.databases წარმოდგენაში

2.6.3.1.10. ALTER DATABASE ბრძანების ADD LOG FILE არგუმენტი:

- ა. ლოგიკურ ფაილს შლის მონაცემთა ბაზიდან
- ბ. ფიზიკურ ფაილს ამატებს მონაცემთა ბაზაში
- გ. ლოგიკურ ფაილს ამატებს მონაცემთა ბაზაში

2.6.3.1.11. ALTER DATABASE ბრძანების REMOVE FILE არგუმენტი:

- ა. შლის ლოგიკური ფაილის აღწერას და ფიზიკურ ფაილს. ფაილი ცარიელი უნდა იყოს
- ბ. შლის ლოგიკური ფაილის აღწერას და ფიზიკურ ფაილს. ფაილი ცარიელი არ უნდა იყოს
- გ. არ შლის ლოგიკური ფაილის აღწერას და ფიზიკურ ფაილს. ფაილი ცარიელი უნდა იყოს

2.6.3.1.12. ALTER DATABASE ბრძანების MODIFY FILE არგუმენტი:

- ა. განსაზღვრავს ფაილს, რომელიც უნდა შეიცვალოს
- ბ. განსაზღვრავს ფაილს, რომელიც არ უნდა შეიცვალოს
- გ. არ განსაზღვრავს ფაილს, რომელიც უნდა შეიცვალოს

2.6.3.1.13. თუ ALTER DATABASE ბრძანების MODIFY FILE არგუმენტში მითითებულია SIZE განყოფილება, მაშინ:

- ა. ახალი ზომა ფაილის მიმდინარე ზომაზე ნაკლები უნდა იყოს
- ბ. ახალი ზომა ფაილის მიმდინარე ზომაზე დიდი უნდა იყოს
- გ. ახალი ზომა ფაილის მიმდინარე ზომაზე ნაკლები ან ტოლი უნდა იყოს

2.6.3.1.14. თუ მითითებულია ALTER DATABASE ბრძანების OFFLINE არგუმენტი, მაშინ:

- ა. ფაილი გადადის ავტონომიურ (ოფლაინ) რეჟიმში და ფაილების ჯგუფის ყველა ობიექტი მისაწვდომი ხდება
- ბ. ფაილი არ გადადის ავტონომიურ (ოფლაინ) რეჟიმში და ფაილების ჯგუფის ყველა

ობიექტი მიუწვდომელი ხდება

გ. ფაილი გადადის ავტონომიურ (ოფლაინ) რეჟიმში და ფაილების ჯგუფის ყველა ობიექტი მიუწვდომელი ხდება

2.6.3.1.15. ALTER DATABASE ბრძანების ADD FILEGROUP არგუმენტი:

- ა. მონაცემთა ბაზას უმატებს ფაილების ჯგუფს
- ბ. მონაცემთა ბაზას არ უმატებს ფაილების ჯგუფს
- გ. მონაცემთა ბაზიდან შლის ფაილების ჯგუფს

2.6.3.1.16. ALTER DATABASE ბრძანების REMOVE FILEGROUP არგუმენტი:

- ა. მონაცემთა ბაზიდან შლის ფაილების ჯგუფს. ფაილების ჯგუფი ცარიელი უნდა იყოს
- ბ. მონაცემთა ბაზიდან შლის ფაილების ჯგუფს. ფაილების ჯგუფი ცარიელი არ უნდა იყოს
- გ. მონაცემთა ბაზას უმატებს ფაილების ჯგუფს

2.6.3.1.17. ALTER DATABASE ბრძანების MODIFY FILEGROUP არგუმენტი:

- ა. ცვლის ფაილის პარამეტრებს
- ბ. არ ცვლის ფაილების ჯგუფის პარამეტრებს
- გ. ცვლის ფაილების ჯგუფის პარამეტრებს

2.6.3.1.18. ALTER DATABASE ბრძანების MODIFY FILEGROUP არგუმენტის <ფაილების_ჯგუფის_გახლების_რეჟიმი> კონსტრუქციის READ_ONLY | READONLY მნიშვნელობა მიუთითებს, რომ:

- ა. ფაილების ჯგუფი არ არის მხოლოდ წაკითხვადი
- ბ. ფაილების ჯგუფი არის მხოლოდ წაკითხვადი
- გ. ფაილების ჯგუფი არის მხოლოდ ჩაწერადი

2.6.3.1.19. PRIMARY ფაილების ჯგუფი:

- ა. შეიძლება იყოს მხოლოდ წაკითხვადი
- ბ. არ შეიძლება იყოს მხოლოდ წაკითხვადი
- გ. უნდა იყოს მხოლოდ წაკითხვადი

2.6.3.1.20. ALTER DATABASE ბრძანების MODIFY FILEGROUP არგუმენტის <ფაილების_ჯგუფის_გახლების_რეჟიმი> კონსტრუქციის READ_WRITE | READWRITE მნიშვნელობა მიუთითებს, რომ:

- ა. ფაილების ჯგუფი არ მუშაობს ჩაწერა-წაკითხვის რეჟიმში
- ბ. ფაილების ჯგუფი მუშაობს ჩაწერა-წაკითხვის რეჟიმში
- გ. ფაილების ჯგუფი მუშაობს ჩაწერის რეჟიმში

2.6.3.1.21. ALTER DATABASE ბრძანების <მონაცემთა_ბაზის_რეჟიმები> კონსტრუქციის, <რეჟიმები> კონსტრუქციის, <მდგომარეობის_რეჟიმი> კონსტრუქციის OFFLINE მნიშვნელობა მიუთითებს, რომ:

- ა. მონაცემთა ბაზა არ იხურება და არ მოინიშნება, როგორც ავტონომიური. ამის შემდეგ, მონაცემთა ბაზის ობიექტებს ვეღარ შევცვლით
- ბ. მონაცემთა ბაზა იხურება და მოინიშნება, როგორც ავტონომიური. ამის შემდეგ, მონაცემთა ბაზის ობიექტებს ვეღარ შევცვლით
- გ. მონაცემთა ბაზა იხურება და მოინიშნება, როგორც ავტონომიური. ამის შემდეგ, მონაცემთა ბაზის ობიექტების შეცვლას შევძლებთ

2.6.3.1.22. ALTER DATABASE ბრძანების <მონაცემთა_ბაზის_რეჟიმები> კონსტრუქციის, <რეჟიმები> კონსტრუქციის, <მდგომარეობის_რეჟიმი> კონსტრუქციის ONLINE მნიშვნელობა:

- ა. მონაცემთა ბაზას ხსნის და მისი ობიექტები მიუწვდომელი ხდება
- ბ. მონაცემთა ბაზას ხსნის და მისი ობიექტები მისაწვდომი ხდება
- გ. მონაცემთა ბაზას არ ხსნის და მისი ობიექტები მისაწვდომი ხდება

2.6.3.1.23. ALTER DATABASE ბრძანების <მონაცემთა_ბაზის_რეჟიმები> კონსტრუქციის, <რეჟიმები> კონსტრუქციის, <მომხმარებლის_მიმართვის_რეჟიმი> კონსტრუქციის SINGLE_USER მნიშვნელობა მიუთითებს, რომ:

- ა. დროის ნებისმიერ მომენტში მხოლოდ ორ მომხმარებელს შეუძლია მონაცემთა ბაზასთან მუშაობა
- ბ. დროის ნებისმიერ მომენტში მხოლოდ ერთ მომხმარებელს შეუძლია მონაცემთა ბაზასთან მუშაობა
- გ. დროის ნებისმიერ მომენტში არც ერთ მომხმარებელს არ შეუძლია მონაცემთა ბაზასთან მუშაობა

2.6.3.1.24. თუ მიუთითებულა ALTER DATABASE ბრძანების <მონაცემთა_ბაზის_რეჟიმები> კონსტრუქციის, <რეჟიმები> კონსტრუქციის, <მომხმარებლის_მიმართვის_რეჟიმი> კონსტრუქციის SINGLE_USER მნიშვნელობა და მომხმარებლები მიერთებული არიან მონაცემთა ბაზასთან, მაშინ:

- ა. ALTER DATABASE ბრძანება არ იქნება დაბლოკილი მანამ, სანამ ყველა მომხმარებელი არ იქნება გამორთული მონაცემთა ბაზიდან
- ბ. ALTER DATABASE ბრძანება დაბლოკილი იქნება მანამ, სანამ ყველა მომხმარებელი არ იქნება გამორთული მონაცემთა ბაზიდან
- გ. ALTER DATABASE ბრძანება დაბლოკილი იქნება მანამ, სანამ ყველა მომხმარებელი არ იქნება მიერთებული მონაცემთა ბაზასთან

2.6.3.1.25. ALTER DATABASE ბრძანების <მონაცემთა_ბაზის_რეჟიმები> კონსტრუქციის, <რეჟიმები> კონსტრუქციის, <მომხმარებლის_მიმართვის_რეჟიმი> კონსტრუქციის RESTRICTED_USER მნიშვნელობა მიუთითებს, რომ:

- ა. მონაცემთა ბაზასთან მიერთება შეუძლია ნებისმიერ მომხმარებელს
- ბ. მონაცემთა ბაზასთან მიერთება შეუძლიათ მხოლოდ მონაცემთა ბაზის db_owner ფიქსირებული როლის წევრებს და სერვერის dbcreator და sysadmin ფიქსირებული როლის წევრებს
- გ. მონაცემთა ბაზასთან მიერთება შეუძლიათ მხოლოდ მონაცემთა ბაზის db_owner ფიქსირებული როლის წევრებს

2.6.3.1.26. ALTER DATABASE ბრძანების <მონაცემთა_ბაზის_რეჟიმები> კონსტრუქციის, <რეჟიმები> კონსტრუქციის, <მომხმარებლის_მიმართვის_რეჟიმი> კონსტრუქციის MULTI_USER მნიშვნელობა მიუთითებს, რომ:

- ა. შესაბამისი ნებართვის მქონე ყველა მომხმარებელს არ შეეძლება მონაცემთა ბაზასთან დაკავშირება
- ბ. შესაბამისი ნებართვის მქონე ყველა მომხმარებელს შეეძლება მონაცემთა ბაზასთან დაკავშირება
- გ. შესაბამისი ნებართვის არმქონე ყველა მომხმარებელს შეეძლება მონაცემთა ბაზასთან დაკავშირება

2.6.3.1.27. ALTER DATABASE ბრძანების <მონაცემთა_ბაზის_რეჟიმები> კონსტრუქციის, <რეჟიმები> კონსტრუქციის, <გაახლების_რეჟიმი> კონსტრუქციის READ_ONLY მნიშვნელობა მიუთითებს, რომ:

- ა. მიუთითებს, რომ მონაცემთა ბაზიდან შეუძლებელია მონაცემების მხოლოდ წაკითხვა
- ბ. მიუთითებს, რომ მონაცემთა ბაზიდან შესაძლებელია მონაცემების მხოლოდ წაკითხვა
- გ. მიუთითებს, რომ მონაცემთა ბაზიდან შესაძლებელია მონაცემების მხოლოდ ჩაწერა

2.6.3.1.28. ALTER DATABASE ბრძანების <მონაცემთა_ბაზის_რეჟიმები> კონსტრუქციის, <რეჟიმები> კონსტრუქციის, <გაახლების_რეჟიმი> კონსტრუქციის READ_WRITE მნიშვნელობა მიუთითებს, რომ:

- ა. შეუძლებელია მონაცემთა ბაზის მონაცემების წაკითხვა და შეცვლა
- ბ. შესაძლებელია მონაცემთა ბაზის მონაცემების წაკითხვა და შეცვლა
- გ. შესაძლებელია მონაცემთა ბაზის მონაცემების მხოლოდ წაკითხვა

2.6.3.1.29. ALTER DATABASE ბრძანების <მონაცემთა_ბაზის_რეჟიმები> კონსტრუქციის, <რეჟიმები> კონსტრუქციის, <კურსორის_რეჟიმი> კონსტრუქციის CURSOR_CLOSE_ON_COMMIT ON მნიშვნელობა მიუთითებს, რომ:

- ა. ნებისმიერი გახსნილი კურსორი იხსნება ტრანზაქციის დადასტურების ან უკუქცევის შემთხვევაში
- ბ. ნებისმიერი გახსნილი კურსორი არ იხურება ტრანზაქციის დადასტურების ან უკუქცევის შემთხვევაში
- *გ. ნებისმიერი გახსნილი კურსორი იხურება ტრანზაქციის დადასტურების ან უკუქცევის შემთხვევაში

2.6.3.1.30. ALTER DATABASE ბრძანების <მონაცემთა_ბაზის_რეჟიმები> კონსტრუქციის, <რეჟიმები> კონსტრუქციის, <კურსორის_რეჟიმი> კონსტრუქციის CURSOR_CLOSE_ON_COMMIT OFF მნიშვნელობა მიუთითებს, რომ:

- ა. კურსორი რჩება ღია ტრანზაქციის დადასტურების შემთხვევაში
- ბ. კურსორი არ რჩება ღია ტრანზაქციის დადასტურების შემთხვევაში
- გ. კურსორი რჩება დახურული ტრანზაქციის დადასტურების შემთხვევაში

2.6.3.1.31. ALTER DATABASE ბრძანების <მონაცემთა_ბაზის_რეჟიმები> კონსტრუქციის, <რეჟიმები> კონსტრუქციის, <კურსორის_რეჟიმი> კონსტრუქციის CURSOR_DEFAULT LOCAL მნიშვნელობა მიუთითებს, რომ:

- ა. კურსორი ლოკალური იქნება
- ბ. კურსორი ლოკალური არ იქნება
- გ. კურსორი გლობალური იქნება

2.6.3.1.32. ALTER DATABASE ბრძანების <მონაცემთა_ბაზის_რეჟიმები> კონსტრუქციის, <რეჟიმები> კონსტრუქციის, <კურსორის_რეჟიმი> კონსტრუქციის CURSOR_DEFAULT { LOCAL | GLOBAL მნიშვნელობა მიუთითებს, რომ:

- ა. კურსორი გლობალური არ იქნება
- ბ. კურსორი გლობალური იქნება
- გ. კურსორი ლოკალური იქნება

2.6.3.1.33. ALTER DATABASE ბრძანების <მონაცემთა_ბაზის_რეჟიმები> კონსტრუქციის,

<რეჟიმები> კონსტრუქციის, <ავტომატური_რეჟიმი> კონსტრუქციის AUTO_CLOSE ON მნიშვნელობა მიუთითებს, რომ:

- ა. მონაცემთა ბაზა ნორმალურად არ დაიხურა და მისი რესურსები თავისუფალია მას შემდეგ, რაც უკანასკნელმა მომხმარებელმა მუშაობა დაამთავრა
- ბ. მონაცემთა ბაზა ნორმალურად დაიხურა და მისი რესურსები თავისუფალია მას შემდეგ, რაც უკანასკნელმა მომხმარებელმა მუშაობა დაამთავრა
- გ. მონაცემთა ბაზა ნორმალურად დაიხურა და მისი რესურსები თავისუფალი არ არის მას შემდეგ, რაც უკანასკნელმა მომხმარებელმა მუშაობა დაამთავრა

2.6.3.1.34. ALTER DATABASE ბრძანების <მონაცემთა_ბაზის_რეჟიმები> კონსტრუქციის, <რეჟიმები> კონსტრუქციის, <ავტომატური_რეჟიმი> კონსტრუქციის AUTO_CLOSE OFF მნიშვნელობა მიუთითებს, რომ:

- ა. მონაცემთა ბაზა რჩება ღია მას შემდეგ, რაც უკანასკნელმა მომხმარებელმა მუშაობა არ დაამთავრა
- ბ. მონაცემთა ბაზა არ რჩება ღია მას შემდეგ, რაც უკანასკნელმა მომხმარებელმა მუშაობა დაამთავრა
- გ. მონაცემთა ბაზა რჩება ღია მას შემდეგ, რაც უკანასკნელმა მომხმარებელმა მუშაობა დაამთავრა

2.6.3.1.35. ALTER DATABASE ბრძანების <მონაცემთა_ბაზის_რეჟიმები> კონსტრუქციის, <რეჟიმები> კონსტრუქციის, <ავტომატური_რეჟიმი> კონსტრუქციის AUTO_CREATE_STATISTICS ON მნიშვნელობა მიუთითებს, რომ:

- ა. ავტომატურად უნდა შესრულდეს სტატისტიკის გაახლება. თუ სტატისტიკა არ არსებობს, მაშინ ის არ შეიქმნება
- ბ. ავტომატურად არ უნდა შესრულდეს სტატისტიკის გაახლება. თუ სტატისტიკა არ არსებობს, მაშინ ის შეიქმნება
- გ. ავტომატურად უნდა შესრულდეს სტატისტიკის გაახლება. თუ სტატისტიკა არ არსებობს, მაშინ ის შეიქმნება

2.6.3.1.36. ALTER DATABASE ბრძანების <მონაცემთა_ბაზის_რეჟიმები> კონსტრუქციის, <რეჟიმები> კონსტრუქციის, <ავტომატური_რეჟიმი> კონსტრუქციის AUTO_CREATE_STATISTICS OFF მნიშვნელობა მიუთითებს, რომ:

- ა. სტატისტიკა ხელით უნდა შეიქმნას
- ბ. სტატისტიკა ხელით არ უნდა შეიქმნას
- გ. სტატისტიკა ავტომატურად უნდა შეიქმნას

2.6.3.1.37. ALTER DATABASE ბრძანების <მონაცემთა_ბაზის_რეჟიმები> კონსტრუქციის, <რეჟიმები> კონსტრუქციის, <ავტომატური_რეჟიმი> კონსტრუქციის AUTO_SHRINK ON მნიშვნელობა მიუთითებს, რომ:

- ა. არ უნდა შესრულდეს მონაცემთა ბაზის ფაილების ზომების ავტომატურად შემცირება
- ბ. უნდა შესრულდეს მონაცემთა ბაზის ფაილების ზომების ავტომატურად შემცირება
- გ. უნდა შესრულდეს მონაცემთა ბაზის ფაილების ზომების ავტომატურად გაზრდა

2.6.3.1.38. ALTER DATABASE ბრძანების <მონაცემთა_ბაზის_რეჟიმები> კონსტრუქციის, <რეჟიმები> კონსტრუქციის, <ავტომატური_რეჟიმი> კონსტრუქციის AUTO_SHRINK OFF მნიშვნელობა მიუთითებს, რომ:

- ა. არ შესრულდება მონაცემთა ბაზის ფაილების ზომების ავტომატურად გაზრდა

- ბ. არ შესრულდება მონაცემთა ბაზის ფაილების ზომების ავტომატურად შემცირება
- გ. შესრულდება მონაცემთა ბაზის ფაილების ზომების ავტომატურად შემცირება

2.6.3.1.39. ALTER DATABASE ბრძანების <მონაცემთა_ბაზის_რეჟიმები> კონსტრუქციის, <რეჟიმები> კონსტრუქციის, <ავტომატური_რეჟიმი> კონსტრუქციის AUTO_UPDATE_STATISTICS ON მნიშვნელობა მიუთითებს, რომ:

- ა. არ უნდა მოხდეს ნებისმიერი ვადაგასული სტატისტიკის გაახლება
- ბ. უნდა მოხდეს ნებისმიერი ვადაგასული სტატისტიკის გაახლება
- გ. უნდა მოხდეს ნებისმიერი ვადაგასული სტატისტიკის წაშლა

2.6.3.1.40. ALTER DATABASE ბრძანების <მონაცემთა_ბაზის_რეჟიმები> კონსტრუქციის, <რეჟიმები> კონსტრუქციის, <ავტომატური_რეჟიმი> კონსტრუქციის AUTO_UPDATE_STATISTICS OFF მნიშვნელობა მიუთითებს, რომ:

- ა. სტატისტიკის გაახლება ხელით არ უნდა შესრულდეს
- ბ. სტატისტიკის გაახლება ხელით უნდა შესრულდეს
- გ. სტატისტიკის გაახლება ავტომატურად უნდა შესრულდეს

2.6.3.1.41. ALTER DATABASE ბრძანების <მონაცემთა_ბაზის_რეჟიმები> კონსტრუქციის, <რეჟიმები> კონსტრუქციის, <ავტომატური_რეჟიმი> კონსტრუქციის AUTO_UPDATE_STATISTICS_ASYNC ON მნიშვნელობა მიუთითებს, რომ:

- ა. მოთხოვნები, რომლებიც ახდენენ ვადაგასული სტატისტიკების ავტომატური გაახლების ინიცირებას, დაელოდებიან სტატისტიკის გაახლებას. მომდევნო მოთხოვნები გამოიყენებენ გაახლებულ სტატისტიკებს, როცა ისინი მისაწვდომი გახდება
- ბ. მოთხოვნები, რომლებიც ახდენენ ვადაგასული სტატისტიკების ავტომატური გაახლების ინიცირებას, არ დაელოდებიან სტატისტიკის გაახლებას. მომდევნო მოთხოვნები გამოიყენებენ გაახლებულ სტატისტიკებს, როცა ისინი მისაწვდომი გახდება
- გ. მოთხოვნები, რომლებიც ახდენენ ვადაგასული სტატისტიკების ავტომატური გაახლების ინიცირებას, არ დაელოდებიან სტატისტიკის გაახლებას. მომდევნო მოთხოვნები არ გამოიყენებენ გაახლებულ სტატისტიკებს, როცა ისინი მისაწვდომი გახდება

2.6.3.1.42. ALTER DATABASE ბრძანების <მონაცემთა_ბაზის_რეჟიმები> კონსტრუქციის, <რეჟიმები> კონსტრუქციის, <ავტომატური_რეჟიმი> კონსტრუქციის AUTO_UPDATE_STATISTICS_ASYNC OFF მნიშვნელობა მიუთითებს, რომ:

- ა. მოთხოვნები, რომლებიც არ ახდენენ ვადაგასული სტატისტიკების ავტომატური გაახლების ინიცირებას, იცდიან მანამ, სანამ შესაძლებელი იქნება გაახლებული სტატისტიკების გამოყენება
- ბ. მოთხოვნები, რომლებიც ახდენენ ვადაგასული სტატისტიკების ავტომატური გაახლების ინიცირებას, იცდიან მანამ, სანამ შესაძლებელი იქნება გაახლებული სტატისტიკების გამოყენება
- გ. მოთხოვნები, რომლებიც ახდენენ ვადაგასული სტატისტიკების ავტომატური გაახლების ინიცირებას, არ იცდიან მანამ, სანამ შესაძლებელი იქნება გაახლებული სტატისტიკების გამოყენება

მონაცემთა ბაზის საკონფიგურაციო პარამეტრებთან მუშაობის სხვა საშუალებები

2.6.4.1.1. მონაცემთა ბაზის საკონფიგურაციო პარამეტრებთან სამუშაოდ გამოიყენება:

- ა. sp_dboption შენახული პროცედურა

- ბ. sp_change შენახული პროცედურა
- გ. sp_replace შენახული პროცედურა

2.6.4.1.2. sp_dboption შენახული პროცედურის სინტაქსია:

- ა. sp_dboption ['მონაცემთა_ბაზის_სახელი' [, 'საკონფიგურაციო_პარამეტრი',]
- ბ. sp_dboption ['მონაცემთა_ბაზის_სახელი' [, 'საკონფიგურაციო_პარამეტრი', 'რეჟიმი']]
- გ. sp_dboption ['მონაცემთა_ბაზის_სახელი' 'რეჟიმი']]

2.6.4.1.3. თუ sp_dboption ბრძანებაში არ მივუთითებთ არც ერთ პარამეტრს, მაშინ:

- ა. გაიცემა კონფიგურირების ყველა მიუწვდომელი პარამეტრის სია
- ბ. არ გაიცემა კონფიგურირების ყველა მისაწვდომი პარამეტრის სია
- გ. გაიცემა კონფიგურირების ყველა მისაწვდომი პარამეტრის სია

2.6.4.1.4. კონფიგურირების პარამეტრების მისაღებად გამოიყენება:

- ა. sp_change ბრძანება
- ბ. DATABASEPROPERTYEX ბრძანება
- გ. sp_replace ბრძანება

2.6.4.1.5. DATABASEPROPERTYEX ბრძანების სინტაქსია:

- ა. DATABASEPROPERTYEX (, 'თვისება')
- ბ. DATABASEPROPERTYEX ('მონაცემთა_ბაზის_სახელი',)
- გ. DATABASEPROPERTYEX ('მონაცემთა_ბაზის_სახელი', 'თვისება')

2.6.4.1.6. მონაცემთა ბაზების შესახებ საერთო ინფორმაციის მისაღებად შეგვიძლია გამოვიყენოთ:

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

2.6.4.1.7. sp_helpdb ბრძანების სინტაქსია:

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

2.6.4.1.8. sp_helpdb ბრძანება გასცემს:

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

2.6.4.1.9. ინფორმაციის მისაღებად კონკრეტული ფაილის შესახებ გამოიყენება:

- ა. sp_helpdb შენახული პროცედურა
- ბ. sp_helpfile შენახული პროცედურა
- გ. sp_help შენახული პროცედურა

2.6.4.1.10. sp_helpfile შენახული პროცედურის სინტაქსია:

- ა. sp_helpfile [[@filename =] 'ფაილის_სახელი']
- ბ. sp_helpfile [[@filename =
- გ. sp_helpfile [[@filename = 'ფაილის_სახელი']

2.6.4.1.11. ფაილების ჯგუფის შესახებ ინფორმაციის მისაღებად გამოიყენება:

- ა. sp_help შენახული პროცედურა
- ბ. sp_helpfilegroup შენახული პროცედურა
- გ. sp_helpdb შენახული პროცედურა

2.6.4.1.12. sp_helpfilegroup შენახული პროცედურის სინტაქსია:

- ა. sp_helpfilegroup [[@filegroupname = 'ფაილების_ჯგუფის_სახელი']
- ბ. sp_helpfilegroup [[@filegroupname =] 'ფაილების_ჯგუფის_სახელი'
- გ. sp_helpfilegroup [[@filegroupname =] 'ფაილების_ჯგუფის_სახელი']

2.6.4.1.13. ინფორმაციის მისაღებად მონაცემთა ბაზის შიგნით არსებული სივრცის გამოყენების შესახებ გამოიყენება:

- ა. sp_helpfile შენახული პროცედურა
- ბ. sp_spaceused შენახული პროცედურა
- გ. sp_helpfilegroup შენახული პროცედურა

2.6.4.1.14. sp_spaceused შენახული პროცედურის სინტაქსია:

- ა. sp_spaceused [[@objname =] 'ობიექტის_სახელი' @updateusage =] 'ალამი'
- ბ. sp_spaceused [[@objname =] 'ობიექტის_სახელი' , [@updateusage 'ალამი'
- გ. sp_spaceused [[@objname =] 'ობიექტის_სახელი'] [, [@updateusage =] 'ალამი'

2.6.4.1.15. sp_spaceused შენახული პროცედურის 'ობიექტის_სახელი' არგუმენტი:

- ა. შეიცავს ცხრილის სახელს, რომლის შესახებაც გვინდა ინფორმაციის მიღება
- ბ. შეიცავს ბაზის სახელს, რომლის შესახებაც გვინდა ინფორმაციის მიღება
- გ. შეიცავს შენახული პროცედურის სახელს, რომლის შესახებაც გვინდა ინფორმაციის მიღება

2.6.4.1.16. sp_spaceused შენახული პროცედურის 'ალამი' არგუმენტი:

- ა. მართავს DBCC UPDATEUSAGE ბრძანების შესრულებას და იღებს TRUE ან FALSE მნიშვნელობას
- ბ. არ მართავს DBCC UPDATEUSAGE ბრძანების შესრულებას და იღებს TRUE ან FALSE მნიშვნელობას
- გ. მართავს DBCC UPDATEUSAGE ბრძანების შესრულებას და არ იღებს TRUE ან FALSE მნიშვნელობას

2.6.4.1.17. თუ sp_spaceused შენახული პროცედურის არგუმენტები მითითებული არ არის, მაშინ:

- ა. გაიცემა ინფორმაცია მიმდინარე ცხრილში სივრცის გამოყენების შესახებ
- ბ. გაიცემა ინფორმაცია მიმდინარე მონაცემთა ბაზაში სივრცის გამოყენების შესახებ
- გ. არ გაიცემა ინფორმაცია მიმდინარე მონაცემთა ბაზაში სივრცის გამოყენების შესახებ

2.6.4.1.18. ტრანზაქციების ჟურნალის შესახებ ინფორმაციის მისაღებად გამოიყენება:

- ა. DBCC UPDATEUSAGE ბრძანება
- ბ. sp_spaceused ბრძანება
- გ. DBCC SQLPERF ბრძანება

2.6.4.1.19. DBCC SQLPERF ბრძანება გაცემს ინფორმაციას:

- ა. ტრანზაქციების ჟურნალის გამოყენების შესახებ სისტემურ მონაცემთა ბაზებში
- ბ. ტრანზაქციების ჟურნალის გამოყენების შესახებ თითოეულ მონაცემთა ბაზაში
- გ. ტრანზაქციების ჟურნალის გამოყენების შესახებ მომხმარებლების მონაცემთა ბაზებში

2.6.4.1.20. ფორმაცია მონაცემთა ბაზის ფაილების შესახებ ინახება:

- ა. sysfiles სისტემურ წარმოდგენაში
- ბ. sysgroups სისტემურ წარმოდგენაში
- გ. sysfilegroups წარმოდგენაში

2.6.4.1.21. ინფორმაცია ფაილების ჯგუფის შესახებ ინახება:

- ა. sysfilegroups წარმოდგენაში
- ბ. sysgroups სისტემურ წარმოდგენაში
- გ. sysfiles სისტემურ წარმოდგენაში

მონაცემთა მომხმარებლის ტიპის შექმნა

2.7.1.1.1. მონაცემთა მომხმარებლის ტიპის შესაქმნელად გამოიყენება სისტემური შენახული პროცედურა:

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

2.7.1.1.2. 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']

2.7.1.1.3. sp_addtype შენახული პროცედურის 'ტიპის_სახელი' არგუმენტი განსაზღვრავს:

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

2.7.1.1.4. 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

2.7.1.1.5. 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

2.7.1.1.6. 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

2.7.1.1.7. 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

2.7.1.1.8. sp_addtype შენახული პროცედურის 'null_ტიპი' არგუმენტი იღებს NONNULL მნიშვნელობას, მაშინ:

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

2.7.1.1.9. 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

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

- ა. sp_addtype შენახული პროცედურის 'null_ტიპი' არგუმენტის მნიშვნელობა იგნორირდება
- ბ. sp_addtype შენახული პროცედურის 'null_ტიპი' არგუმენტს ენიჭება NULL
- გ. sp_addtype შენახული პროცედურის 'null_ტიპი' არგუმენტის მნიშვნელობა არ იგნორირდება

2.7.1.1.11. მომხმარებლის ტიპის სახელის შესაცვლელად გამოიყენება:

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

2.7.1.1.12. 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=]

2.7.1.1.13. 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

2.7.1.1.14. 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

2.7.1.1.15. sp_rename შენახული პროცედურის 'ობიექტის_ტიპი' არგუმენტს მომხმარებლის
ტიპისათვის სახელის შესაცვლელად უნდა მივანიჭოთ:

- ა. 'USERDATATYPE' მნიშვნელობა
- ბ. 'USERDBTYPE' მნიშვნელობა
- გ. 'USERDATETYPE' მნიშვნელობა

თავი 3. ცხრილები

ცხრილების დაპროექტება

ცხრილის პირველადი გასაღები

3.1.1.1.1. პირველადი გასაღების დანიშნულებაა:

- ა. ცხრილის მთლიანობის უზრუნველყოფა
- ბ. მონაცემთა ბაზის მთლიანობის უზრუნველყოფა
- გ. სვეტის მთლიანობის უზრუნველყოფა

3.1.1.1.2. პირველადი გასაღები შეიძლება შედგებოდეს:

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

3.1.1.1.3. თუ პირველადი გასაღები ერთი სვეტისგან შედგება, მაშინ ამ სვეტის მნიშვნელობები:

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

3.1.1.1.4. თუ პირველადი გასაღები რამდენიმე სვეტისგან შედგება, მაშინ თითოეული სვეტის შიგნით მნიშვნელობები შეიძლება:

- ა. არ გამეორდეს
- ბ. გამეორდეს
- გ. არ იყოს დუბლირებული

3.1.1.1.5. პირველად გასაღებში შემავალი ყველა სვეტის მნიშვნელობების ნებისმიერი კომბინაცია:

- ა. არ უნდა იყოს უნიკალური
- ბ. უნდა იყოს დუბლირებული
- გ. უნდა იყოს უნიკალური

3.1.1.1.6. პირველადი გასაღების შექმნის დროს მასში შემავალი სვეტებისათვის სერვერი ავტომატურად:

- ა. არ ქმნის უნიკალურ ინდექსს
- ბ. ქმნის უნიკალურ ინდექსს
- გ. ქმნის უნიკალურ ფაილს

3.1.1.1.7. მოთხოვნებში პირველადი გასაღების გამოყენების დროს უნიკალური ინდექსი პირველად გასაღებში შემავალი სვეტების მონაცემებთან მიმართებას:

- ა. ამცირებს
- ბ. არ აჩქარებს
- გ. აჩქარებს

3.1.1.1.8. ცხრილს შეიძლება ჰქონდეს:

- ა. მხოლოდ ერთი PRIMARY KEY შეზღუდვა

- ბ. რამდენიმე PRIMARY KEY შეზღუდვა
- გ. ორი PRIMARY KEY შეზღუდვა

3.1.1.1.9. 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

3.1.1.1.10. პირველადი გასაღების ერთ-ერთი დანიშნულებაა:

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

ცხრილის გარე გასაღები

3.1.2.1. 1. გარე გასაღები შეიძლება შეიცავდეს:

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

3.1.2.1. 2. გარე გასაღები გამოიყენება:

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

3.1.2.1. 3. დამოკიდებული ცხრილის გარე გასაღებში შემავალ სვეტებში მონაცემებმა მხოლოდ ის მნიშვნელობები უნდა მიიღოს, რომელიც აქვს:

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

3.1.2.1.4. 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

3.1.2.1.5. 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

3.1.2.1.6. 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

- 3.1.2.1.7. გარე გასაღების ინდექსირება:
- ა. ანელებს საჭირო მონაცემების მოძებნას
 - ბ. არ აჩქარებს საჭირო მონაცემების მოძებნას
 - გ. აჩქარებს საჭირო მონაცემების მოძებნას

სვეტის უნიკალურობის განსაზღვრა

- 3.1.3.1.1. UNIQUE შეზღუდვა განკუთვნილია:
- ა. სვეტში მნიშვნელობების უნიკალურობის უზრუნველსაყოფად
 - ბ. სტრიქონში მნიშვნელობების უნიკალურობის უზრუნველსაყოფად
 - გ. მონაცემთა ბაზაში მნიშვნელობების უნიკალურობის უზრუნველსაყოფად
- 3.1.3.1.2. PRIMARY KEY შეზღუდვისათვის:
- ა. ავტომატურად არ ხდება მნიშვნელობების უნიკალურობის უზრუნველყოფა
 - ბ. ავტომატურად ხდება მნიშვნელობების უნიკალურობის უზრუნველყოფა
 - გ. ავტომატურად ხდება მნიშვნელობების უნიკალურობის შეცვლა
- 3.1.3.1.3. 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
- 3.1.3.1.4. 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
- 3.1.3.1.5. UNIQUE შეზღუდვის გამოყენებისას შესაბამის სვეტში შეიძლება იყოს:
- ა. სამი NULL მნიშვნელობა
 - ბ. ორი NULL მნიშვნელობა
 - გ. მხოლოდ ერთი NULL მნიშვნელობა

შემმოწმებელი შეზღუდვების განსაზღვრა

- 3.1.4.1.1. CHECK შეზღუდვა განსაზღვრავს შესაძლო მნიშვნელობების დიაპაზონს:
- ა. მხოლოდ ერთი სვეტისთვის
 - ბ. ერთი ან მეტი სვეტისთვის
 - გ. ერთი სტრიქონისთვის
- 3.1.4.1.2. ერთი სვეტისთვის შეიძლება განისაზღვროს:
- ა. რამდენიმე CHECK შეზღუდვა
 - ბ. მხოლოდ ერთი CHECK შეზღუდვა
 - გ. არც ერთი CHECK შეზღუდვა
- 3.1.4.1.3. 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

3.1.4.1.4. CHECK შეზღუდვას საფუძვლად უდევს:

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

3.1.4.1.5. 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

3.1.4.1.6. 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

3.1.4.1.7. CHECK შეზღუდვებში:

- ა. დასაშვებია სხვა სტრიქონების სახელების მითითებაც
- ბ. დაუშვებელია სხვა სვეტების სახელების მითითებაც
- გ. დასაშვებია სხვა სვეტების სახელების მითითებაც

ნაგულისხმევი მნიშვნელობის განსაზღვრა

3.1.5.1. 1. თითოეული სტრიქონის სვეტი ყოველთვის შეიცავს:

- ა. კონკრეტულ ან NULL მნიშვნელობას
- ბ. მხოლოდ NULL მნიშვნელობას
- გ. მხოლოდ კონკრეტულ მნიშვნელობას

3.1.5.1.2. 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

3.1.5.1.3. 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

3.1.5.1.4. ნაგულისხმევი მნიშვნელობის განსაზღვრისას:

- ა. არ შეგვიძლია ჩადგმული ფუნქციების გამოყენებაც
- ბ. შეგვიძლია ჩადგმული ფუნქციების გამოყენებაც
- გ. შეუძლებელია ჩადგმული ფუნქციების გამოყენებაც

სვეტ-მთვლელის განსაზღვრა

3.1.6.1.1. სვეტ-მთვლელისთვის სრულდება უნიკალური მნიშვნელობების:

- ა. ხელით გენერირება
- ბ. ავტომატურად გენერირება
- გ. არანაირი გენერირება

3.1.6.1.2. 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

3.1.6.1.3. სვეტ-მთვლელის განსაზღვრისას უნდა მივუთითოთ:

- ა. მისი საწყისი მნიშვნელობა და ბიჯი
- ბ. მისი ბიჯი
- გ. მისი საწყისი მნიშვნელობა

3.1.6.1.4. თუ სვეტ-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

3.1.6.1.5. ცხრილს შეიძლება ჰქონდეს:

- ა. ორი სვეტ-მთვლელი
- ბ. მხოლოდ ერთი სვეტ-მთვლელი
- გ. სამი სვეტ-მთვლელი

3.1.6.1.6. თუ სვეტისთვის განსაზღვრულია IDENTITY თვისება, მაშინ ამ სვეტის ტიპი უნდა იყოს:

- ა. datetime, decimal
- ბ. float, numeric
- გ. bigint, decimal, int, numeric, smallint, tinyint

3.1.6.1.7. 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

3.1.6.1.8. თუ სვეტისთვის განსაზღვრულია IDENTITY თვისება, მაშინ ამ სვეტისთვის:

- ა. განსაზღვრული უნდა იყოს ავტომატური (ნაგულისხმევი) მნიშვნელობა

- ბ. განსაზღვრული არ უნდა იყოს ავტომატური (ნაგულისხმევი) მნიშვნელობა
- გ. შეიძლება განსაზღვრული იყოს ავტომატური (ნაგულისხმევი) მნიშვნელობა

3.1.6.1.9. სვეტ-მთვლელი:

- ა. აუცილებლად უნდა შევიდეს ცხრილის პირველადი გასაღების შედგენილობაში
- ბ. არ უნდა შევიდეს ცხრილის პირველადი გასაღების შედგენილობაში
- გ. შეიძლება შევიდეს ცხრილის პირველადი გასაღების შედგენილობაში

3.1.6.1.10. 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

3.1.6.1.11. თუ ცხრილიდან ყველა სტრიქონს წავშლით:

- ა. ეს არ გამოიწვევს სვეტ-მთვლელში ნუმერაციის თავიდან დაწყებას
- ბ. ეს გამოიწვევს სვეტ-მთვლელში ნუმერაციის თავიდან დაწყებას
- გ. ეს გამოიწვევს სვეტ-მთვლელში ნუმერაციის ნულიდან დაწყებას

3.1.6.1.12. თუ სტრიქონის ჩასმა არ მოხერხდა, მაშინ სერვერი:

- ა. მაინც გაზრდის სვეტ-მთვლელის მნიშვნელობას
- ბ. არ გაზრდის სვეტ-მთვლელის მნიშვნელობას
- გ. შეამცირებს სვეტ-მთვლელის მნიშვნელობას

3.1.6.1.13. თუ ცხრილის უკანასკნელ სტრიქონს წავშლით, რომელშიც სვეტ-მთვლელის მნიშვნელობა იყო 20 და შემდეგ ერთ სტრიქონს დავემატებთ, მაშინ დამატებულ სტრიქონში სვეტი-მთვლელის მნიშვნელობა იქნება:

- ა. 19
- ბ. 20
- გ. 21

3.1.6.1.14. სვეტ-მთვლელთან სამუშაოდ გამოიყენება ფუნქცია:

- ა. sys.identity_columns სისტემური წარმოდგენა
- ბ. IDENT_CURRENT('ცხრილის_სახელი'), IDENT_INCR('ცხრილის_სახელი') და IDENT_SEED('ცხრილის_სახელი')
- გ. master მონაცემთა ბაზა

3.1.6.1.15. IDENT_CURRENT('ცხრილის_სახელი') ფუნქცია მითითებული ცხრილისათვის გასცემს:

- ა. სვეტ-მთვლელის საწყის მნიშვნელობას
- ბ. სვეტ-მთვლელის უკანასკნელ მნიშვნელობას
- გ. სვეტ-მთვლელის შუალედურ მნიშვნელობას

3.1.6.1.16. IDENT_INCR('ცხრილის_სახელი') ფუნქცია მითითებული ცხრილისთვის გასცემს:

- ა. სვეტ-მთვლელის საწყის მნიშვნელობას
- ბ. სვეტ-მთვლელის შუალედურ მნიშვნელობას
- გ. სვეტ-მთვლელის უკანასკნელ მნიშვნელობას

3.1.6.1.17. IDENT_SEED('ცხრილის_სახელი') ფუნქცია მითითებული ცხრილისთვის გაცემს:

- ა. სვეტ-მთვლელის უკანასკნელ მნიშვნელობას
- ბ. სვეტ-მთვლელის საწყის მნიშვნელობას
- გ. ნაზარდის მნიშვნელობას

3.1.6.1.18. შეგვიძლია ნება დავრთოთ სვეტ-მთვლელისათვის ნებისმიერი სიდიდის მინიჭებას. ამისათვის, გამოიყენება:

- ა. IDENT_INCR('ცხრილის_სახელი') ფუნქცია
- ბ. SET IDENTITY_INSERT ბრძანება
- გ. INSERT ბრძანება

3.1.6.1.19. SET ბრძანება მოქმედებს:

- ა. მხოლოდ მიმდინარე შეერთების ფარგლებში
- ბ. ყველა შეერთების ფარგლებში
- გ. სხვა შეერთების ფარგლებში

3.1.6.1.20. თუ ერთი მომხმარებელი დართავს სვეტ-მთვლელის მნიშვნელობის შეცვლის ნებას, მაშინ მეორე მომხმარებელს:

- ა. ხანდახან ექნება ამის უფლება
- ბ. არ ექნება ამის უფლება
- გ. ექნება ამის უფლება

3.1.6.1.21. ერთი შეერთების შიგნით სვეტ-მთვლელის მნიშვნელობის შეცვლის ნებართვის გაცემა შესაძლებელია:

- ა. რამდენიმე ცხრილისთვის
- ბ. მხოლოდ ორი ცხრილისთვის
- გ. მხოლოდ ერთი ცხრილისათვის

3.1.6.1.22. თუ საჭიროა მეორე ცხრილისათვის სვეტ-მთვლელის მნიშვნელობის შეცვლის ნებართვის გაცემა, მაშინ:

- ა. ჯერ პირველი ცხრილისათვის უნდა გავაუქმოთ ეს ნებართვა და შემდეგ მეორე ცხრილისათვის განვსაზღვროთ
- ბ. ჯერ მეორე ცხრილისათვის უნდა გავაუქმოთ ეს ნებართვა და შემდეგ პირველი ცხრილისათვის განვსაზღვროთ
- გ. ჯერ პირველი ცხრილისათვის უნდა განვსაზღვროთ ეს ნებართვა და შემდეგ მეორე ცხრილისათვის გავაუქმოთ

3.1.6.1.23. DBCC CHECKIDENT ბრძანება საშუალებას გვაძლევს:

- ა. არ შევცვალოთ სვეტ-მთვლელის მნიშვნელობა
- ბ. შევცვალოთ სვეტ-მთვლელის მნიშვნელობა
- გ. წავშალოთ სვეტ-მთვლელის მნიშვნელობა

3.1.6.1.24. DBCC CHECKIDENT ბრძანების NORESEED საკვანძო სიტყვა მიუთითებს, რომ:

- ა. მუშაობა არ შესრულდება ნახვის რეჟიმში
- ბ. გაცივმა უკანასკნელი მნიშვნელობა. აგრეთვე მნიშვნელობა, რომელიც მიიღებოდა გარედან ჩარევა რომ მომხდარიყო

გ. მუშაობა შესრულდება ნახვის რეჟიმში. გაიცემა უკანასკნელი მნიშვნელობა. აგრეთვე მნიშვნელობა, რომელიც მიიღებოდა გარედან ჩარევა რომ არ მომხდარიყო

3.1.6.1.25. DBCC CHECKIDENT ბრძანების RESEED საკვანძო სიტყვის მითითების შედეგად IDENTITY მნიშვნელობა ისე იცვლება, რომ ხდება:

- ა. მინიმალური სვეტ-მთვლელობა ნაპოვნ მნიშვნელობებს შორის
- ბ. მაქსიმალური სვეტ-მთვლელობა ნაპოვნ მნიშვნელობებს შორის
- გ. საშუალო სვეტ-მთვლელობა ნაპოვნ მნიშვნელობებს შორის

3.1.6.1.26. DBCC CHECKIDENT ბრძანების RESEED საკვანძო სიტყვა გამოიყენება მაშინ, როცა:

- ა. არ შესრულდა სვეტ-მთვლელობა მნიშვნელობის უშუალოდ ჩასმა ან მიმდინარე მნიშვნელობის ცვლილება
- ბ. შესრულდა სვეტ-მთვლელობა მნიშვნელობის უშუალოდ ჩასმა ან მიმდინარე მნიშვნელობის ცვლილება
- გ. შესრულდა სვეტ-მთვლელობა მნიშვნელობის უშუალოდ წაკითხვა ან მიმდინარე მნიშვნელობის წაკითხვა

3.1.6.1.27. DBCC CHECKIDENT ბრძანების 'მიმდინარე_მნიშვნელობა' პარამეტრის საშუალებით:

- ა. აშკარად მიეთითება სვეტ-მთვლელობის მიმდინარე მნიშვნელობა. შესაბამისად, მომდევნო მნიშვნელობის გენერირება შესრულდება ამ მნიშვნელობის ბაზაზე
- ბ. აშკარად არ მიეთითება სვეტ-მთვლელობის მიმდინარე მნიშვნელობა. შესაბამისად, მომდევნო მნიშვნელობის გენერირება შესრულდება ამ მნიშვნელობის ბაზაზე
- გ. აშკარად მიეთითება სვეტ-მთვლელობის მიმდინარე მნიშვნელობა. შესაბამისად, მომდევნო მნიშვნელობის გენერირება არ შესრულდება ამ მნიშვნელობის ბაზაზე

ცხრილის შექმნა

3.2.1.1. 1. ცხრილის შესაქმნელად გამოიყენება ბრძანება:

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

3.2.1.1.2. თუ CREATE TABLE ბრძანებაში მითითებული 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

3.2.1.1.3. 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

3.2.1.1.4. 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

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

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

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

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

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

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

3.2.1.1.8. გამოთვლადი სვეტები:

- ა. ვირტუალური სტრიქონებია
- ბ. არავირტუალური სვეტებია
- გ. ვირტუალური სვეტებია

3.2.1.1.9. გამოთვლადი სვეტები:

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

3.2.1.1.10. 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

3.2.1.1.11. 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

3.2.1.1.12. CREATE TABLE ბრძანების 'გამოსახულება' არგუმენტში ქვემოთხოვნების გამოყენება:

- ა. დასაშვებია

- ბ. დაუშვებელია
- გ. ხანდახან დასაშვებია

3.2.1.1.13. 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

3.2.1.1.14. გამოთვლადი სვეტები:

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

3.2.1.1.15. გამოთვლადი სვეტები:

- ა. არ შეიძლება გამოვიყენოთ, როგორც პირველადი გასაღების ნაწილი
- ბ. შეიძლება გამოვიყენოთ, როგორც პირველადი გასაღების ნაწილი
- გ. შეიძლება გამოვიყენოთ, როგორც გარე გასაღების ნაწილი

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

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

3.2.1.1.17. დაუშვებია, ცხრილში გვაქვს ორი მთელრიცხვა A და B სვეტი, გამოთვლადი სვეტის მნიშვნელობა გამოითვლება $A + B$ გამოსახულებით. ასეთ შემთხვევაში, ამ სვეტის ინდექსირება:

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

3.2.1.1.18. თუ გამოთვლადი სვეტი განისაზღვრება $A + \text{DATEPART}(dd, \text{GETDATE}())$ გამოსახულებით, მაშინ მისი ინდექსირება:

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

3.2.1.1.19. გამოთვლადი სვეტისათვის:

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

3.2.1.1.20. გამოთვლადი სვეტები:

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

3.2.1.1.21. გამოთვლადი სვეტებისთვის:

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

3.2.1.1.22. გამოთვლადი სვეტების გამოყენება:

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

3.2.1.1.23. სვეტის განსაზღვრისას DEFAULT არგუმენტი:

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

3.2.1.1.24. 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

3.2.1.1.25. ავტომატური (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

3.2.1.1.26. ავტომატური (DEFAULT) მნიშვნელობა:

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

3.2.1.1.27. 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

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

- ა. იქნება სტრიქონ-მთვლელი
- ბ. არ იქნება სვეტ-მთვლელი
- *გ. იქნება სვეტ-მთვლელი

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

- ა. მთვლელის ნაზარდი
- *ბ. მთვლელის საწყისი მნიშვნელობა
- გ. მთვლელის საბოლოო მნიშვნელობა

3.2.1.1.30. თუ სვეტის განსაზღვრისას გამოყენებულია IDENTITY არგუმენტი, მაშინ 'ნაზარდი' არის:

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

3.2.1.1.31. ROWGUIDCOL არგუმენტი მიუთითებს, რომ სვეტში მოთავსებული იქნება:

- ა. გლობალური იდენტიფიკატორი
- ბ. ლოკალური იდენტიფიკატორი
- გ. არაგლობალური იდენტიფიკატორი

3.2.1.1.32. გლობალური იდენტიფიკატორის შემცველი სვეტის ტიპი:

- ა. უნდა იყოს int
- ბ. არ უნდა იყოს uniqueidentifier
- გ. უნდა იყოს uniqueidentifier

3.2.1.1.33. ცხრილს შეიძლება გლობალური იდენტიფიკატორის შემცველი:

- ა. მხოლოდ ორი სვეტი ჰქონდეს
- ბ. მხოლოდ ერთი სვეტი ჰქონდეს
- გ. რამდენიმე სვეტი ჰქონდეს

3.2.1.1.34. გლობალური იდენტიფიკატორის შემცველი სვეტისთვის მნიშვნელობა:

- ა. ავტომატურად განისაზღვრება NEWID() ფუნქციის საფუძველზე
- ბ. ავტომატურად არ განისაზღვრება NEWID() ფუნქციის საფუძველზე
- გ. ავტომატურად განისაზღვრება CONSTRAINT პარამეტრის საფუძველზე

3.2.1.1.35. სვეტის შეზღუდვის მითითებისას CONSTRAINT პარამეტრის შემდეგ მიეთითება:

- 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

3.2.1.1.36. სვეტის შეზღუდვის მითითებისას NULL პარამეტრი:

- ა. გვადლევს სტრიქონში NULL მნიშვნელობის შენახვის უფლებას
- ბ. არ გვადლევს სვეტში NULL მნიშვნელობის შენახვის უფლებას
- გ. გვადლევს სვეტში NULL მნიშვნელობის შენახვის უფლებას

3.2.1.1.37. სვეტის შეზღუდვის მითითებისას NOT NULL პარამეტრი:

- ა. არ კრძალავს სვეტში NULL მნიშვნელობის შენახვას
- ბ. კრძალავს სვეტში NULL მნიშვნელობის შენახვას
- გ. კრძალავს სვეტში ნულის შენახვას

3.2.1.1.38. 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

3.2.1.1.39. თითოეული ცხრილისათვის შეიძლება შეიქმნას მხოლოდ:

- ა. ორი PRIMARY KEY შეზღუდვა
- ბ. ერთი PRIMARY KEY შეზღუდვა
- გ. რამდენიმე PRIMARY KEY შეზღუდვა

3.2.1.1.40. თუ სვეტის შეზღუდვაში მითითებულია PRIMARY KEY პარამეტრი, მაშინ პირველადი გასაღები განისაზღვრება:

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

3.2.1.1.41. თუ სვეტის შეზღუდვაში მითითებულია PRIMARY KEY პარამეტრი, მაშინ პირველადი გასაღები შედგება მხოლოდ:

- ა. ერთი სვეტისაგან
- ბ. ორი სვეტისაგან
- გ. რამდენიმე სვეტისაგან

3.2.1.1.42. თუ საჭიროა პირველადი გასაღების ფორმირება ორი და მეტი სვეტის ბაზაზე, მაშინ საჭიროა მისი კონფიგურირება:

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

3.2.1.1.43. 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

3.2.1.1.44. UNIQUE მთლიანობის შეზღუდვისათვის ავტომატურად:

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

3.2.1.1.45. ერთ ცხრილში სხვადასხვა სვეტისთვის:

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

3.2.1.1.46. CLUSTERED პარამეტრი გამოიყენება სვეტისათვის:

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

3.2.1.1.47. NOT CLUSTERED პარამეტრი გამოიყენება სვეტისათვის:

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

3.2.1.1.48. 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

3.2.1.1.49. PRIMARY KEY შეზღუდვისათვის ავტომატურად:

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

3.2.1.1.50. UNIQUE შეზღუდვისათვის ავტომატურად:

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

3.2.1.1.51. [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

3.2.1.1.52. 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

3.2.1.1.53. 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

3.2.1.1.54. 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

3.2.1.1.55. ON DELETE პარამეტრში NO ACTION რეჟიმი:

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

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

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

3.2.1.1.57. 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

3.2.1.1.58. 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

3.2.1.1.59. 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

3.2.1.1.60. 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

3.2.1.1.61. <ცხრილის_შეზღუდვა> კონსტრუქცია განსაზღვრავს მთლიანობის შეზღუდვებს:

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

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

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

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

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

ცხრილის წაშლა

3.3.1.1. 1. ცხრილის წასაშლელად გამოიყენება ბრძანება:

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

3.3.1.1. 2. იმისათვის, რომ შევძლოთ ცხრილის წაშლა:

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

3.3.1.1. 3. ვერ წავშლით ცხრილს, თუ:

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

ცხრილის შეცვლა

3.4.1.1. 1. ცხრილის სტრუქტურის შესაცვლელად გამოიყენება ბრძანება:

- ა. CREATE TABLE

- ბ. ALTER TABLE
- გ. DROP DATABASE

3.4.1.1. 2. ALTER TABLE ბრძანებაში თუ მითითებულია WITH CHECK პარამეტრი, მაშინ:

- ა. შესრულდება ჩასამატებელი მონაცემების შემოწმება PRIMARY KEY შეზღუდვასთან შესაბამისობაზე
- ბ. არ შესრულდება ჩასამატებელი მონაცემების შემოწმება FOREIGN KEY და CHECK შეზღუდვებთან შესაბამისობაზე
- გ. შესრულდება ჩასამატებელი მონაცემების შემოწმება FOREIGN KEY და CHECK შეზღუდვებთან შესაბამისობაზე

3.4.1.1. 3. ALTER TABLE ბრძანებაში თუ მითითებულია WITH NOCHECK პარამეტრი, მაშინ:

- ა. ჩასამატებელი მონაცემების შემოწმება არ შესრულდება
- ბ. ჩასამატებელი მონაცემების შემოწმება შესრულდება
- გ. ჩასამატებელი მონაცემების შემოწმება ხანდახან შესრულდება

3.4.1.1. 4. ALTER TABLE ბრძანებაში თუ მითითებულია ADD ROWGUIDCOL პარამეტრი, მაშინ:

- ა. სტრიქონს დაემატება ROWGUIDCOL თვისება
- ბ. სვეტიდან წაიშლება ROWGUIDCOL თვისება
- გ. სვეტს დაემატება ROWGUIDCOL თვისება

3.4.1.1. 5. ALTER TABLE ბრძანებაში თუ მითითებულია DROP ROWGUIDCOL პარამეტრი, მაშინ:

- ა. სტრიქონიდან წაიშლება ROWGUIDCOL თვისება
- ბ. სვეტიდან წაიშლება ROWGUIDCOL თვისება
- გ. სვეტს დაემატება ROWGUIDCOL თვისება

3.4.1.1. 6. ALTER TABLE ბრძანებაში თუ მითითებულია ALL პარამეტრი, მაშინ:

- ა. ამუშავდება ან გამოირთვება ცხრილთან დაკავშირებული ყველა ტრიგერი
- ბ. ამუშავდება ან გამოირთვება ცხრილთან დაკავშირებული ყველა ინდექსი
- გ. ამუშავდება ან გამოირთვება ცხრილთან დაკავშირებული ყველა წარმოდგენა

ცხრილების მართვა

ცხრილისათვის სახელის შეცვლა

3.5.1.1. 1. ცხრილისათვის სახელის შესაცვლელად გამოიყენება:

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

3.5.1.1.2. ცხრილის სვეტისთვის სახელის შესაცვლელად გამოიყენება:

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

3.5.1.1.3. `sp_rename 'Cxri10', 'Cxri120'` პროცედურის შესრულების შედეგად:

- ა. Cxri10 ცხრილს დაერქმევა Cxri10 სახელი
- ბ. Cxri120 ცხრილს დაერქმევა Cxri10 სახელი
- გ. Cxri10 ცხრილს დაერქმევა Cxri120 სახელი

3.5.1.1.4. `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

ინფორმაციის მიღება ცხრილების შესახებ

3.5.2.1. 1. 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

3.5.2.1. 2. კონკრეტული ცხრილის შესახებ ინფორმაციის მისაღებად გამოიყენება:

- ა. `sp_depends` შენახული პროცედურა
- ბ. `OBJECTPROPERTY` ფუნქცია
- გ. `sp_rename` შენახული პროცედურა

3.5.2.1. 3. `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

3.5.2.1. 4. ცხრილის თვისებების მისაღებად შეგვიძლია გამოვიყენოთ სისტემური შენახული პროცედურა:

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

3.5.2.1. 5. `sp_help Personali` პროცედურის შესრულების შედეგად გაიცემა ინფორმაცია:

- ა. `sp_help` პროცედურის შესახებ
- ბ. Personali ცხრილის თვისებების შესახებ
- გ. ორივე ობიექტის შესახებ

ინფორმაციის მიღება ცხრილზე დამოკიდებული ობიექტების შესახებ

3.5.3.1. 1. ინფორმაცია ცხრილზე დამოკიდებული ობიექტების შესახებ სასარგებლოა მაშინ, როცა:

- ა. ვაპირებთ ცხრილის შექმნას

- ბ. არ ვაპირებთ ცხრილის წაშლას
- გ. ვაპირებთ ცხრილის წაშლას

3.5.3.1. 2. ინფორმაციის მისაღებად ცხრილზე დამოკიდებული ობიექტების შესახებ უნდა გამოვიყენოთ სისტემური პროცედურა:

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

3.5.3.1. 3. sp_depends 'Personali' პროცედურის შესრულების შედეგად გაიცემა ინფორმაცია:

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

ცხრილებს შორის კავშირის შესახებ ინფორმაციის მიღება

3.5.4.1. 1. 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

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

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

3.5.4.1. 3. 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 დაუკავშირებელი ცხრილის შესახებ

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

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

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

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

- გ. Xelshekruleba ცხრილთან დაკავშირებული ცხრილების შესახებ. ეს მთავარი ცხრილია

დიაგრამები

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

- ა. გრაფიკები
- ბ. დიაგრამები
- გ. სვეტები

3.6.1.1. 2. დიაგრამების დანიშნულებაა მონაცემთა ბაზის:

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

თავი 4. მონაცემების მართვა

ცხრილში მონაცემების ჩამატება

INSERT ბრძანება

4.1.1.1. 1. ცხრილში მონაცემების ჩასამატებლად გამოიყენება:

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

4.1.1.1.2. 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

4.1.1.1.3. 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

4.1.1.1.4. 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. უკუმიმდევრობით

4.1.1.1.5. სვეტის მნიშვნელობა შეგვიძლია არ მივუთითოთ მხოლოდ იმ შემთხვევაში, თუ ამ სვეტისთვის განსაზღვრულია:

- ა. IDENTITY თვისება, NULL მნიშვნელობის შენახვის შესაძლებლობა ან timestap ტიპი
- ბ. IDENTITY თვისება, NULL მნიშვნელობა ან CHECK შეზღუდვა
- გ. NULL მნიშვნელობა, timestap ტიპი ან FOREIGN KEY შეზღუდვა

4.1.1.1.6. INSERT ბრძანებაში VALUES არგუმენტი განსაზღვრავს:

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

4.1.1.1.7. 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

4.1.1.1.8. 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

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

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

4.1.1.1.10. 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

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

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

4.1.1.1.12. '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

4.1.1.1.13. 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)

4.1.1.1.14. 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

4.1.1.1.15. 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

4.1.1.1.16. INSERT ბრძანების DEFAULT VALUES არგუმენტის მითითების შემთხვევაში:

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

4.1.1.1.17. 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

4.1.1.1.18. 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

SELECT...INTO ბრძანება

4.1.2.1. 1. 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

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

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

4.1.2.1. 3. 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

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

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

4.1.2.1. 5. SELECT...INTO ბრძანებაში INTO 'ახალი_ცხრილს_სახელი' არგუმენტი:

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

4.1.2.1. 6. SELECT...INTO ბრძანებაში INTO 'ახალი_ცხრილს_სახელი' არგუმენტის გამოყენებით თუ დროებით ცხრილს შევქმნით, მაშინ ცხრილის სახელის:

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

4.1.2.1. 7. SELECT...INTO ბრძანებაში FROM არგუმენტი:

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

4.1.2.1. 8. 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

ცხრილიდან მონაცემების ამორჩევა

SELECT ბრძანება

SELECT განყოფილება

4.2.1.1.1. SELECT ბრძანების SELECT განყოფილებაში:

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

4.2.1.1.2. 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

4.2.1.1.3. SELECT ბრძანების SELECT განყოფილებაში ALL არგუმენტი:

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

4.2.1.1.4. SELECT ბრძანების SELECT განყოფილებაში DISTINCT არგუმენტი:

- ა. კრძალავს ერთნაირი სტრიქონების გამოტანას
- ბ. არ კრძალავს ერთნაირი სტრიქონების გამოტანას
- გ. კრძალავს ერთნაირი სვეტების გამოტანას

4.2.1.1.5. 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

4.2.1.1.6. SELECT ბრძანების SELECT განყოფილების TOP n PERCENT არგუმენტში:

- ა. n არ არის პროცენტი
- ბ. n არის პროცენტი
- გ. n არის მთელი რიცხვი

4.2.1.1.7. 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

4.2.1.1.8. SELECT განყოფილების <სია> კონსტრუქციაში 'ცხრილის_სახელი' იმ შემთხვევაში მიეთითება, როცა მოთხოვნაში:

- ა. რამდენიმე ინდექსია მითითებული
- ბ. არც ერთი ცხრილია მითითებული
- გ. რამდენიმე ცხრილია მითითებული

4.2.1.1.9. SELECT განყოფილების <სია> კონსტრუქციაში 'წარმოდგენის_სახელი' იმ შემთხვევაში მიეთითება, როცა მოთხოვნაში:

- ა. რამდენიმე ინდექსია მითითებული
- ბ. არც ერთი წარმოდგენაა მითითებული
- გ. რამდენიმე წარმოდგენაა მითითებული

4.2.1.1.10. SELECT განყოფილების <სია> კონსტრუქციაში 'ცხრილის_ფსევდონიმი' შეგვიძლია გამოვიყენოთ:

- ა. ტრიგერის სახელის ნაცვლად
- ბ. მონაცემთა ბაზის სახელის ნაცვლად
- გ. ცხრილის სახელის ნაცვლად

4.2.1.1.11. 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

4.2.1.1.12. 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

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

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

4.2.1.1.14. 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

4.2.1.1.15. 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

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

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

4.2.1.1.17. 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

4.2.1.1.18. თუ SELECT მოთხოვნაში რამდენიმე ცხრილია მითითებული, რომლებსაც IDENTITYCOL სვეტები აქვს, 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

FROM განყოფილება

4.2.1.2.1. SELECT მოთხოვნაში FROM სიტყვის შემდეგ მიეთითება იმ:

- ა. ცხრილების ან წარმოდგენების სახელები, საიდანაც ხდება სვეტების ამორჩევა
- ბ. ინდექსების სახელები, საიდანაც ხდება სვეტების ამორჩევა
- გ. პროცედურების სახელები, საიდანაც ხდება სვეტების ამორჩევა

4.2.1.2.2. FROM განყოფილებაში <ბმული_ცხრილი> კონსტრუქცია გამოიყენება კავშირის უზრუნველსაყოფად:

- ა. რამდენიმე პროცედურას შორის
- ბ. რამდენიმე ფუნქციას შორის
- გ. რამდენიმე ცხრილს შორის

4.2.1.2.3. 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

4.2.1.2.4. 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

4.2.1.2.5. FROM განყოფილების <ბმული_ცხრილი> კონსტრუქციაში <ბმის_ტიპი> კონსტრუქციის:

- ა. მარცხნივ მიეთითება მთავარი ცხრილის სახელი
- ბ. მარჯვნივ მიეთითება მთავარი ცხრილის სახელი
- გ. მარცხნივ მიეთითება დამოკიდებული ცხრილის სახელი

4.2.1.2.6. FROM განყოფილების <ბმული_ცხრილი> კონსტრუქციაში <ბმის_ტიპი> კონსტრუქციის:

- ა. მარცხნივ მიეთითება დამოკიდებული ცხრილის სახელი
- ბ. მარჯვნივ მიეთითება დამოკიდებული ცხრილის სახელი
- გ. მარჯვნივ მიეთითება მთავარი ცხრილის სახელი

4.2.1.2.11. FROM განყოფილების <ბმული_ცხრილი> კონსტრუქციაში <ბმის_ტიპი> კონსტრუქციის JOIN არგუმენტის შემდეგ მიეთითება:

- ა. შუა ცხრილი
- ბ. მარცხენა ცხრილი
- გ. მარჯვენა ცხრილი

4.2.1.2.12. 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

4.2.1.2.13. 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

4.2.1.2.14. 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

WHERE განყოფილება

4.2.1.3.1. WHERE განყოფილება გამოიყენება ცხრილის:

- ა. სტრიქონების გაფილტვრისათვის
- ბ. სვეტების გაფილტვრისათვის
- გ. სტრიქონების გაერთიანებისათვის

4.2.1.3.2. 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

4.2.1.3.3. 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

4.2.1.3.4. 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

4.2.1.3.5. თუ დასამუშავებელია ათასობით სტრიქონი, მაშინ უმჯობესია:

- ა. WHERE პირობის გამოყენება
- ბ. { INNER | { { LEFT | RIGHT | FULL } [OUTER] } JOIN } საკვანძო სიტყვების გამოყენება
- გ. FROM განყოფილების გამოყენება

4.2.1.3.6. 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

ჩადგმული მოთხოვნები

4.2.1.4.1. ერთ მოთხოვნას მეორე მოთხოვნის მართვა:

- ა. შეუძლია
- ბ. არ შეუძლია
- გ. ხანდახან შეუძლია

4.2.1.4.2. ერთ მოთხოვნას შეუძლია მეორე მოთხოვნის მართვა:

- ა. ჩადგმული მოთხოვნების გამოყენების გარეშე
- ბ. ჩადგმული მოთხოვნების გამოყენებით
- გ. არაჩადგმული მოთხოვნების გამოყენებით

4.2.1.4.3. ჩადგმული მოთხოვნა გვაქვს მაშინ, როცა ერთი მოთხოვნის შიგნით:

- ა. ოთხი მოთხოვნაა მოთავსებული
- ბ. სამი მოთხოვნაა მოთავსებული
- გ. მეორეა მოთავსებული

4.2.1.4.4. შიგა მოთხოვნის მიერ გაცემული შედეგები:

- ა. არ მოწმდება გარე (ძირითადი) მოთხოვნის პირობაში
- ბ. მოწმდება გარე (ძირითადი) მოთხოვნის პირობაში
- გ. მოწმდება შიგა (ძირითადი) მოთხოვნის პირობაში

4.2.1.4.5. ჩადგმული მოთხოვნის შიგა მოთხოვნაში აკრძალულია:

- ა. SELECT * ტიპის ბრძანების გამოყენება
- ბ. UPDATE * ტიპის ბრძანების გამოყენება
- გ. DELETE * ტიპის ბრძანების გამოყენება

4.2.1.4.6. შიგა მოთხოვნებში:

- ა. დასაშვებია მხოლოდ HAVING საკვანძო სიტყვის გამოყენება

- ბ. დაუშვებელია GROUP BY და HAVING საკვანძო სიტყვების გამოყენება
- გ. დასაშვებია GROUP BY და HAVING საკვანძო სიტყვების გამოყენება

4.2.1.4.7. ჩადგმულ მოთხოვნაში, იმისათვის, რომ გარე მოთხოვნა შესრულდეს საჭიროა, რომ შიგა მოთხოვნამ:

- ა. გასცეს ერთი ან მეტი ჩანაწერი
- ბ. არ გასცეს ერთი ან მეტი ჩანაწერი
- გ. არ გასცეს არც ერთი ჩანაწერი

ბმული მოთხოვნები

4.2.1.5.1. ბმული მოთხოვნების გამოყენების დროს:

- ა. შიგა მოთხოვნა სრულდება გარე მოთხოვნაში მითითებული ცხრილის თითოეული სტრიქონისათვის
- ბ. გარე მოთხოვნა სრულდება შიგა მოთხოვნაში მითითებული ცხრილის თითოეული სტრიქონისათვის
- გ. შიგა მოთხოვნა სრულდება შიგა მოთხოვნაში მითითებული ცხრილის თითოეული სტრიქონისათვის

4.2.1.5.2. ბმული მოთხოვნების გამოყენების დროს:

- ა. ჯერ ამოირჩევა შიგა მოთხოვნის სტრიქონი, შემდეგ კი მისთვის შესრულდება გარე მოთხოვნა
- ბ. ჯერ ამოირჩევა გარე მოთხოვნის სტრიქონი, შემდეგ კი მისთვის შესრულდება შიგა მოთხოვნა
- გ. ჯერ ამოირჩევა შიგა მოთხოვნის სტრიქონი, შემდეგ კი მისთვის შესრულდება შიგა მოთხოვნა

GROUP BY განყოფილება

4.2.1.6.1. 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

4.2.1.6.2. 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

4.2.1.6.3. GROUP BY განყოფილების 'სვეტების_სია' არგუმენტში:

- ა. არ მიეთითება იმ სვეტების სახელები, რომელთა მიხედვით უნდა შესრულდეს დაჯგუფება
- ბ. მიეთითება იმ სვეტების სახელები, რომელთა მიხედვით უნდა შესრულდეს დაჯგუფება

გ. მითითება იმ სვეტების სახელები, რომელთა მიხედვით არ უნდა შესრულდეს დაჯგუფება

4.2.1.6.4. 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

4.2.1.6.5. 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

HAVING განყოფილება

4.2.1.7.1. HAVING განყოფილება გამოიყენება ძეზნის პირობის მისათითებლად:

- ა. ინდექსისთვის
- ბ. ტრიგერისთვის
- გ. ჯგუფისთვის

4.2.1.7.2. 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

4.2.1.7.3. 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

UNION განყოფილება

4.2.1.8.1. UNION ბრძანება:

- ა. არ აერთიანებს რამდენიმე მოთხოვნის შესრულების შედეგს
- ბ. აერთიანებს რამდენიმე მოთხოვნის შესრულების შედეგს
- გ. აერთიანებს ერთი მოთხოვნის შესრულების შედეგს

4.2.1.8.2. UNION ბრძანების გამოყენების 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

4.2.1.8.3. UNION ბრძანების გამოყენების დროს:

- ა. სტრიქონების ტიპები უნდა ემთხვეოდეს
- ბ. სვეტების ტიპები არ უნდა ემთხვეოდეს
- გ. სვეტების ტიპები უნდა ემთხვეოდეს

4.2.1.8.4. UNION ბრძანება ავტომატურად:

- ა. არ გამორიცხავს ერთნაირ სტრიქონებს
- ბ. გამორიცხავს ერთნაირ სტრიქონებს
- გ. გამორიცხავს ერთნაირ სვეტებს

4.2.1.8.5. UNION ბრძანების გამოყენების 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

ORDER BY განყოფილება

4.2.1.9.1. ORDER BY განყოფილება გამოიყენება ცხრილის დასახარისხებლად ამა თუ იმ სვეტის მნიშვნელობების:

- ა. მხოლოდ ზრდის მიხედვით
- ბ. ზრდის ან კლების მიხედვით
- გ. მხოლოდ კლების მიხედვით

4.2.1.9.2. ცხრილი შეგვიძლია დავახარისხოთ:

- ა. ერთი ან მეტი სვეტის მიხედვით
- ბ. მხოლოდ ერთი სვეტის მიხედვით
- გ. არც ერთი სვეტის მიხედვით

4.2.1.9.3. ASC საკვანძო სიტყვა გამოიყენება მითითებული სვეტის მონაცემების დასალაგებლად:

- ა. არც ზრდადობის და არც კლებადობის მიხედვით
- ბ. კლებადობის მიხედვით
- გ. ზრდადობის მიხედვით

4.2.1.9.4. DESC საკვანძო სიტყვა გამოიყენება მითითებული სვეტის მონაცემების დასალაგებლად:

- ა. ზრდადობის მიხედვით
- ბ. არც ზრდადობის და არც კლებადობის მიხედვით
- გ. კლებადობის მიხედვით

4.2.1.9.5. 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 და არც კლებადობის მიხედვით

COMPUTE განყოფილება

4.2.1.10.1. COMPUTE განყოფილება საშუალებას გვაძლევს არჩეული:

- ა. სვეტების მიმართ გამოვიყენოთ აგრეგირების ფუნქციები
- ბ. სვეტების მიმართ არ გამოვიყენოთ აგრეგირების ფუნქციები
- გ. სტრიქონების მიმართ გამოვიყენოთ აგრეგირების ფუნქციები

4.2.1.10.2. 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

4.2.1.10.3. 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

4.2.1.10.4. COMPUTE განყოფილებაში BY სიტყვა მიუთითებს, რომ გამოთვლის შედეგი:

- ა. უნდა დაჯგუფდეს
- ბ. არ უნდა დაჯგუფდეს
- გ. უნდა დაჯგუფდეს როგორც სვეტების, ისე სტრიქონების მიხედვით

4.2.1.10.5. 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

4.2.1.10.6. COMPUTE განყოფილებაში:

- ა. დაუშვებელია DISTINCT საკვანძო სიტყვის გამოყენება
- ბ. დასაშვებია DISTINCT საკვანძო სიტყვის გამოყენება
- გ. დასაშვებია CREATE TABLE ბრძანების გამოყენება

აგრეგირების ფუნქციები

4.2.1.11.1. აგრეგირების ფუნქციები მონაცემების:

- ა. სტატისტიკურ დამუშავებას ასრულებს
- ბ. ფიზიკურ დამუშავებას ასრულებს
- გ. სტატისტიკურ დამუშავებას არ ასრულებს

4.2.1.11.2. AVG ფუნქცია მითითებული სვეტისთვის გამოთვლის:

- ა. მნიშვნელობების ჯამს

- ბ. მინიმალურ მნიშვნელობას
- გ. საშუალო არითმეტიკულს

4.2.1.11.3. თუ მონაცემები დაჯგუფებულია, მაშინ AVG ფუნქცია:

- ა. თითოეული ჯგუფისთვის გამოითვლის მაქსიმუმს
- ბ. თითოეული ჯგუფისთვის არ გამოითვლის საშუალო არითმეტიკულს
- გ. თითოეული ჯგუფისთვის გამოითვლის საშუალო არითმეტიკულს

4.2.1.11.4. აგრეგირების ფუნქციების უმრავლესობა:

- ა. ითვალისწინებს NULL მნიშვნელობის შემცველ სვეტებს
- ბ. თითოეული ჯგუფისთვის გამოითვლის მაქსიმუმს
- გ. არ ითვალისწინებს NULL მნიშვნელობის შემცველ სვეტებს

4.2.1.11.5. 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

4.2.1.11.6. AVG ფუნქციაში ALL არგუმენტი:

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

4.2.1.11.7. AVG ფუნქციაში DISTINCT საკვანძო სიტყვა მიუთითებს, რომ აგრეგირება შეეხება:

- ა. მხოლოდ ერთნაირ სტრიქონებს
- ბ. მხოლოდ უნიკალურ სტრიქონებს
- გ. მხოლოდ არაუნიკალურ სტრიქონებს

4.2.1.11.8. COUNT ფუნქცია:

- ა. თვლის მნიშვნელობების რაოდენობას მოცემულ სვეტში ან სტრიქონების საერთო რაოდენობას ცხრილში
- ბ. გამოთვლის სვეტის მნიშვნელობების საშუალო არითმეტიკულს
- გ. გამოთვლის სვეტის მნიშვნელობების ჯამს

4.2.1.11.9. COUNT ფუნქციაში * მიუთითებს, რომ გაიცემა ცხრილში მოთავსებული სტრიქონების:

- ა. საშუალო მნიშვნელობა
- ბ. ჯამი
- გ. რაოდენობა

4.2.1.11.10. SUM ფუნქცია გამოთვლის მითითებული სვეტის მნიშვნელობების:

- ა. საშუალო მნიშვნელობას
- ბ. ჯამს
- გ. რაოდენობას

4.2.1.11.11. MAX ფუნქცია გასცემს მითითებული სვეტის მნიშვნელობებიდან:

- ა. საშუალო მნიშვნელობას
- ბ. მაქსიმალურს
- გ. მინიმალურს

4.2.1.11.12. MIN ფუნქცია გასცემს მითითებული სვეტის მნიშვნელობებიდან:

- ა. მინიმალურს
- ბ. ჯამს
- გ. მაქსიმალურს

ცხრილის მონაცემების ცვლილება

UPDATE ბრძანება

4.3.1.1.1. ცხრილებში მონაცემების შესაცვლელად გამოიყენება:

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

4.3.1.1.2. 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

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

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

4.3.1.1.4. UPDATE ბრძანება მონაცემებს წარმოდგენის:

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

4.3.1.1.5. 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

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

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

4.3.1.1.7. 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

4.3.1.1.8. 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

4.3.1.1.9. 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

4.3.1.1.10. 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

4.3.1.1.11. ფილტრის მითითების შემთხვევაში UPDATE ბრძანებით განხორციელებული ცვლილებები შეეხება:

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

4.3.1.1.12. 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

ცხრილიდან მონაცემების წაშლა

DELETE ბრძანება

4.4.1.1.1. ცხრილიდან სტრიქონების წასაშლელად გამოიყენება:

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

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

- ა. ერთი სტრიქონი წაიშლება

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

მონაცემების მასობრივი გადაწერა

4.5.1.1.1. მონაცემების იმპორტისა და ექსპორტისათვის გამოიყენება:

- ა. BCP უტილიტი
- ბ. DELETE ბრძანება
- გ. UPDATE ბრძანება

4.5.1.1.2. 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

4.5.1.1.3. BCP უტილიტი ცხრილის სტრიქონებს წერს:

- ა. Excel-ის ფაილში და პირიქით
- ბ. ტექსტურ ფაილში და პირიქით
- გ. Access-ის ფაილში და პირიქით

4.5.1.1.4. BCP უტილიტის მუშაობის შედეგები:

- ა. ხანდახან აისახება ტრანზაქციების ჟურნალში
- ბ. აისახება ტრანზაქციების ჟურნალში
- გ. არ აისახება ტრანზაქციების ჟურნალში

4.5.1.1.5. თუ BCP უტილიტის მიერ გადაწერის ოპერაცია წარუმატებლად დამთავრდა, მაშინ:

- ა. საწყისი მონაცემების აღდგენას ვერ შევძლებთ
- ბ. საწყისი მონაცემების აღდგენას შევძლებთ
- გ. საბოლოო მონაცემების აღდგენას შევძლებთ

თავი 5. Transact SQL-ის საფუძვლები

შესავალი

5.1.1.1.1. იდენტიფიკატორები გამოიყენება:

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

5.1.1.1.2. იდენტიფიკატორი:

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

5.1.1.1.3. 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

5.1.1.1.4. Transact-SQL-ის მიერ კომენტარის დამუშავება:

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

5.1.1.1.5. არის:

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

5.1.1.1.6. სტრიქონული კომენტარი:

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

5.1.1.1.7. 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

5.1.1.1.8. დარეზერვებული საკვანძო სიტყვები გამოიყენება სერვერის მუშაობის მართვისთვის და:

- ა. არის დამოკიდებული რეგისტრზე და არ შეიძლება წარმოადგენდეს იდენტიფიკატორს

- ბ. არის დამოკიდებული რეგისტრზე და შეიძლება წარმოადგენდეს იდენტიფიკატორს
- გ. არ არის დამოკიდებული რეგისტრზე და არ შეიძლება წარმოადგენდეს იდენტიფიკატორს

გამოსახულებები

5.2.1.1.1. 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

5.2.1.1.2. გამოსახულება:

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

5.2.1.1.3. ეს მუდმივი სიდიდეებია, რომელთა მნიშვნელობები:

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

5.2.1.1.4. 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

5.2.1.1.5. 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

5.2.1.1.6. სვეტის სახელი შეიძლება გამოვიყენოთ, როგორც:

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

5.2.1.1.7. ცხრილის ერთი სტრიქონიდან მეორეზე გადასვლის დროს სვეტის სახელს:

- ა. შესაბამისი მნიშვნელობა არ ენიჭება

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

5.2.1.1.8. 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

5.2.1.1.9. ფუნქციას შეიძლება:

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

ოპერატორები

5.3.1.1.1. უმარტივესი ოპერატორებია:

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

5.3.1.1.2. უნარული ოპერატორები, რომლებიც მხოლოდ:

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

5.3.1.1.3. უმარტივესი ოპერატორები გამოიყენება:

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

5.3.1.1.4. უმარტივესი ოპერატორებია:

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

5.3.1.1.5. „+“ ოპერატორი გამოიყენება:

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

5.3.1.1.6. „-“ ოპერატორი გამოიყენება:

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

5.3.1.1.7. „~“ ოპერატორი გამოიყენება:

- ა. რიცხვის დამატების მისაღებად

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

5.3.1.1.8. 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

5.3.1.1.9. 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

5.3.1.1.10. ცვლადების გამოცხადების ოპერატორია:

- ა. "="
- ბ. SET
- გ. DECLARE

5.3.1.1.11. არითმეტიკის ოპერატორებია:

- ა. +, -, *, /, %
- ბ. +, -, <, >=
- გ. >, <=, *, /

5.3.1.1.12. ბინარული ოპერატორებია:

- ა. ~, +, %
- ბ. +, -, *, /, %
- გ. -, *, /, ~

5.3.1.1.13. კონკატენაციის ოპერატორია:

- ა. ~
- ბ. %
- გ. +

5.3.1.1.14. შედარების ოპერატორებია:

- ა. <, !=, !<, =
- ბ. >=, !>, +
- გ. -, <=, *

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

- ა. NULL
- ბ. false
- გ. true

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

- ა. NULL
- ბ. false
- გ. true

5.3.1.1.17. თუ შედარება შეუძლებელია, მაგალითად, ტიპებია განსხვავებული, მაშინ:

- ა. გაიცემა NULL
- ბ. გაიცემა true
- გ. გაიცემა false

5.3.1.1.18. შედარების შედეგების დასამუშავებლად გამოიყენება:

- ა. NULL ოპერატორი
- ბ. SET ოპერატორი
- გ. IF ოპერატორი

5.3.1.1.19. ლოგიკის ოპერატორებია:

- ა. ALL, AND, ANY, BETWEEN, EXISTS, IN, LIKE, NOT, OR და SOME
- ბ. ALL, AND, ANY, DECLARE
- გ. BETWEEN, EXISTS, IN, SET

5.3.1.1.20. ბიტობრივი ოპერატორებია:

- ა. ბიტობრივი AND, ბიტობრივი OR და BETWEEN
- ბ. ბიტობრივი AND, ბიტობრივი OR და ბიტობრივი XOR
- გ. ბიტობრივი AND, ბიტობრივი OR და DECLARE

იდენტიფიკატორები

5.4.1.1.1. სტანდარტული იდენტიფიკატორების განსაზღვრისას:

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

5.4.1.1.2. შეზღუდული იდენტიფიკატორების განსაზღვრისას:

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

5.4.1.1.3. შეზღუდული იდენტიფიკატორები უნდა მოვითავსოთ:

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

5.4.1.1.4. შეზღუდული იდენტიფიკატორებია:

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

5.4.1.1.5. 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

5.4.1.1.6. იდენტიფიკატორი, რომელიც # სიმბოლოთი იწყება, აღნიშნავს:

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

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

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

5.4.1.1.8. იდენტიფიკატორი, რომელიც @ სიმბოლოთი იწყება, აღნიშნავს:

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

5.4.1.1.9. 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

5.4.1.1.10. იდენტიფიკატორის შიგნით:

- ა. დაუშვებელია სპეციალური სიმბოლოების გამოყენება: ~, !, %, ^, &, - , (,), {, }, “, ., \, ‘ და ინტერვალი
- ბ. დასაშვებია სპეციალური სიმბოლოების გამოყენება: ~, !, %, ^, a, d, (, \, ‘ და ინტერვალი
- გ. დასაშვებია სპეციალური სიმბოლოების გამოყენება: ~, r, h, ^, a, d, (, \, ‘ და ინტერვალი

5.4.1.1.11. 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

5.4.1.1.12. სახელების განსხვავება რეგისტრის მიხედვით:

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

5.4.1.1.13. 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

5.4.1.1.14. ზოგიერთი შეზღუდვის გვერდის ასავლელად:

ა. შეგვიძლია შეზღუდული იდენტიფიკატორები გამოვიყენოთ

ბ. არ შეგვიძლია შეზღუდული იდენტიფიკატორები გამოვიყენოთ

გ. ხანდახან შეგვიძლია შეუზღუდავი იდენტიფიკატორები გამოვიყენოთ

5.4.1.1.15. შეზღუდული იდენტიფიკატორების 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

5.4.1.1.16. 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

5.4.1.1.17. 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

5.4.1.1.18. 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

5.4.1.1.19. 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

5.4.1.1.20. 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

5.4.1.1.21. 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

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

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

5.4.1.1.23. თუ მითითებული არ არის სქემის სახელი, მაშინ:

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

5.4.1.1.24. შემზღვევლების (“ “ და [] სიმბოლოები) გამოყენების წესები:

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

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

- ა. 1. კუთხური (< >) და კვადრატული ფრჩხილები ([]) გამოიყენება იდენტიფიკატორების შეზღუდვისათვის. ბრჭყალები (“ “) გამოიყენება სიმბოლური სტრიქონებისათვის.
2. სიმბოლური სტრიქონები არ უნდა მოთავსდეს აპოსტროფებში (‘ ‘). იდენტიფიკატორის შეზღუდვისათვის 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 შეზღუდვისათვის. ბრჭყალები (“ “) არ გამოიყენება სიმბოლური სტრიქონებისათვის.
2. სიმბოლური სტრიქონები არ უნდა მოთავსდეს აპოსტროფებში (‘ ‘). იდენტიფიკატორის შეზღუდვისათვის 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 შეზღუდვისათვის. ბრჭყალები (“ “) არ გამოიყენება სიმბოლური სტრიქონებისათვის.
2. სიმბოლური სტრიქონები უნდა მოთავსდეს აპოსტროფებში (‘ ‘) ან ბრჭყალებში (“ “). თუ სტრიქონი აპოსტროფს შეიცავს, მაშინ ეს სტრიქონი შეგვიძლია ბრჭყალებში მოვათავსოთ

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

- ა. 1. იდენტიფიკატორების შეზღუდვისათვის კუთხური ფრჩხილები (< >) გამოიყენება. 2. სიმბოლური სტრიქონები უნდა მოთავსდეს ამპერსანდებში (& &) ან ბრჭყალებში (“ “). თუ სტრიქონი აპოსტროფს შეიცავს, მაშინ ეს სტრიქონი შეგვიძლია ბრჭყალებში მოვათავსოთ
- ბ. 1. იდენტიფიკატორების შეზღუდვისათვის კვადრატული ფრჩხილები ([]) გამოიყენება.
2. სიმბოლური სტრიქონები უნდა მოთავსდეს აპოსტროფებში (‘ ‘) ან ბრჭყალებში (“ “). თუ სტრიქონი აპოსტროფს შეიცავს, მაშინ ეს სტრიქონი შეგვიძლია ბრჭყალებში მოვათავსოთ
- გ. 1. იდენტიფიკატორების შეზღუდვისათვის კვადრატული ფრჩხილები ([]) არ გამოიყენება. 2. სიმბოლური სტრიქონები არ უნდა მოთავსდეს აპოსტროფებში (‘ ‘) ან ბრჭყალებში (“ “). თუ სტრიქონი აპოსტროფს შეიცავს, მაშინ ეს სტრიქონი არ შეგვიძლია ბრჭყალებში მოვათავსოთ

5.4.1.1.27. 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'

5.4.1.1.28. კვადრატული ფრჩხილები, როგორც შემზღვევლები:

- ა. ხანდახან გამოიყენება
- ბ. არ გამოიყენება ყოველთვის, მიუხედავად QUOTED_IDENTIFIER პარამეტრის მნიშვნელობისა
- გ. გამოიყენება ყოველთვის, მიუხედავად QUOTED_IDENTIFIER პარამეტრის მნიშვნელობისა

ცვლადები

5.5.1.1.1. ცვლადის გამოყენებამდე საჭიროა მისი გამოცხადება:

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

5.5.1.1.2. ცვლადის სახელი:

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

5.5.1.1.3. @ სიმბოლო ლოკალური ცვლადების სახელების გარდა გამოიყენება:

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

5.5.1.1.4. ერთი DECLARE ბრძანებით:

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

5.5.1.1.5. ცვლადებს მნიშვნელობები შეგვიძლია მივანიჭოთ:

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

5.5.1.1.6. მინიჭების რომელი ოპერატორი შეიცავს შეცდომას:

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

5.5.1.1.7. ცვლადს:

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

5.5.1.1.8. 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]

5.5.1.1.9. ცვლადების მნიშვნელობების გამოსატანად შეგვიძლია:

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

მმართველი კონსტრუქციები

BEGIN...END

5.6.1.1.1. 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

5.6.1.1.2. BEGIN საკვანძო სიტყვა:

- ა. არც იწყებს და არც ამთავრებს ბლოკს
- ბ. იწყებს ბლოკს
- გ. ამთავრებს ბლოკს

5.6.1.1.3. END საკვანძო სიტყვა:

- ა. იწყებს ბლოკს
- ბ. არც იწყებს და არც ამთავრებს ბლოკს
- გ. ამთავრებს ბლოკს

5.6.1.1.4. 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

5.6.1.1.5. BEGIN...END ბლოკი:

- ა. შეიძლება იყოს ჩადგმული
- ბ. შეიძლება იყოს ჩაუდგმელი
- გ. არ შეიძლება იყოს ჩადგმული

IF...ELSE

5.6.2.1.1. IF...ELSE კონსტრუქციაში 'ლოგიკური_გამოსახულება':

- ა. არ არის ლოგიკური პირობა
- ბ. არის ფიზიკური პირობა
- გ. არის ლოგიკური პირობა

5.6.2.1.2. 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

5.6.2.1.3. თუ IF...ELSE კონსტრუქციაში 'ლოგიკური_გამოსახულება' იღებს true მნიშვნელობას, მაშინ:

- ა. არ 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'

5.6.2.1.4. 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'

CASEEND

5.6.3.1.1. CASEEND:

- ა. არის ერთგანშტოებიანი კონსტრუქცია
- ბ. არის მრავალგანშტოებიანი კონსტრუქცია
- გ. არ არის მრავალგანშტოებიანი კონსტრუქცია

5.6.3.1.2. '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

5.6.3.1.3. CASEEND 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

5.6.3.1.4. თუ CASEEND 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

5.6.3.1.5. თუ CASEEND 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

5.6.4.1.1. COALESCE კონსტრუქცია:

- ა. გასცემს მეორე არანულოვან მნიშვნელობას
- ბ. არ გასცემს პირველ არანულოვან მნიშვნელობას
- გ. გასცემს პირველ არანულოვან მნიშვნელობას

WHILE...BREAK & CONTINUE

5.6.5.1.1. WHILE...BREAK & CONTINUE კონსტრუქცია:

- ა. არ გამოიყენება ციკლების ორგანიზებისათვის
- ბ. გამოიყენება ციკლების ორგანიზებისათვის
- გ. გამოიყენება ციკლების წაშლისთვის

5.6.5.1.2. 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)

5.6.5.1.3. თუ WHILE...BREAK & CONTINUE კონსტრუქციაში 'ლოგიკური_გამოსახულება' false მნიშვნელობას მიიღებს, მაშინ:

- ა. ციკლის შესრულება გაგრძელდება
- ბ. ციკლის შესრულება არ შეწყდება
- გ. ციკლის შესრულება შეწყდება

5.6.5.1.4. BREAK ბრძანების შესრულება:

- ა. იწვევს ციკლის შესრულების იძულებით შეწყვეტას და ციკლიდან გამოსვლას
- ბ. იწვევს ლოგიკური პირობის შემოწმებაზე გადასვლას
- გ. არ იწვევს ციკლის შესრულების იძულებით შეწყვეტას და ციკლიდან გამოსვლას

5.6.5.1.5. CONTINUE ბრძანების შესრულება:

- ა. არ იწვევს ლოგიკური პირობის შემოწმებაზე გადასვლას
- ბ. იწვევს ლოგიკური პირობის შემოწმებაზე გადასვლას
- გ. იწვევს ციკლის შესრულების იძულებით შეწყვეტას და ციკლიდან გამოსვლას

5.6.5.1.6. CONTINUE ბრძანების შემდეგ მოთავსებული ბრძანებები:

- ა. არ შესრულდება
- ბ. შესრულდება
- გ. ხანდახან შესრულდება

ლოგიკის ოპერატორები

ALL ოპერატორი

5.7.1.1.1. ALL ოპერატორი სკალარულ სიდიდეს:

- ა. ადარებს ქვემოთხოვნის მიერ დაბრუნებულ თითოეულ მნიშვნელობას
- ბ. გასცემს ქვემოთხოვნის შედეგს
- გ. არ ადარებს ქვემოთხოვნის მიერ დაბრუნებულ თითოეულ მნიშვნელობას

5.7.1.1.2. თუ ALL ოპერატორში ლოგიკური პირობა სრულდება ყველა დაბრუნებული მნიშვნელობისათვის, მაშინ პირობა:

- ა. ჩაითვლება ნახევრად შესრულებულად
- ბ. ჩაითვლება შესრულებულად
- გ. არ ჩაითვლება შესრულებულად

5.7.1.1.3. ALL ოპერატორში ქვემოთხოვნა:

- ა. არ გასცემს ერთი სვეტის მნიშვნელობებს
- ბ. გასცემს ერთი სვეტის მნიშვნელობებს
- გ. გასცემს რამდენიმე სვეტის მნიშვნელობებს

5.7.1.1.4. ALL ოპერატორში ქვემოთხოვნა გასცემს ერთი სვეტის მნიშვნელობებს, რომლის მნიშვნელობების ტიპი:

- ა. არ უნდა ემთხვეოდეს 'გამოსახულების' ტიპს
- ბ. ხანდახან უნდა ემთხვეოდეს 'გამოსახულების' ტიპს
- გ. უნდა ემთხვეოდეს 'გამოსახულების' ტიპს

SOME და ANY ოპერატორები

5.7.2.1.1. SOME და ANY ოპერატორები სკალარულ სიდიდეს:

- ა. ადარებს ქვემოთხოვნის მიერ გაცემულ თითოეულ მნიშვნელობას
- ბ. ადარებს პროცედურის მიერ გაცემულ თითოეულ მნიშვნელობას
- გ. არ ადარებს ქვემოთხოვნის მიერ გაცემულ თითოეულ მნიშვნელობას

5.7.2.1.2. თუ SOME და ANY ოპერატორებში ლოგიკური პირობა სრულდება ერთ-ერთი მათგანისათვის, მაშინ პირობა:

- ა. არ ჩაითვლება შესრულებულად

- ბ. ჩაითვლება შესრულებულად
- გ. ჩაითვლება ნახევრად შესრულებულად

BETWEEN ოპერატორი

5.7.3.1.1. BETWEEN ოპერატორი საშუალებას:

- ა. გვაძლევს მივუთითოთ მნიშვნელობების დასაწყისი
- ბ. არ გვაძლევს მივუთითოთ მნიშვნელობების დიაპაზონი
- გ. გვაძლევს მივუთითოთ მნიშვნელობების დიაპაზონი

5.7.3.1.2. 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

5.7.3.1.3. BETWEEN ოპერატორში 'საწყისი_გამოსახულება' განსაზღვრავს:

- ა. დიაპაზონის დასაწყისს
- ბ. დიაპაზონის დასასრულს
- გ. არაფერს

5.7.3.1.4. BETWEEN ოპერატორში 'საბოლოო_გამოსახულება' განსაზღვრავს:

- *ა. დიაპაზონის დასასრულს
- ბ. არაფერს
- გ. დიაპაზონის დასაწყისს

EXISTS ოპერატორი

5.7.4.1.1. EXISTS ოპერატორი გასცემს true მნიშვნელობას, თუ ქვემოთხოვნა:

- ა. არ გასცემს ორ სტრიქონს
- ბ. არ გასცემს თუნდაც ერთ სტრიქონს
- გ. გასცემს თუნდაც ერთ სტრიქონს

5.7.4.1.2. EXISTS ოპერატორი გასცემს false მნიშვნელობას, თუ ქვემოთხოვნა:

- ა. გასცემს თუნდაც ერთ სტრიქონს
- ბ. არ გასცემს თუნდაც ერთ სტრიქონს
- გ. გასცემს ორ სტრიქონს

IN ოპერატორი

5.7.5.1.1. 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

LIKE ოპერატორი

5.7.6.1.1. LIKE ოპერატორის საშუალებით:

- ა. შეუძლებელია გამოსახულების შედარება მოცემულ შაბლონთან
- ბ. შესაძლებელია გამოსახულების შედარება მოცემულ შაბლონთან
- გ. გამოსახულება გადაიწერება შაბლონში

5.7.6.1.2. LIKE ოპერატორში 'გამოსახულება' არგუმენტი:

- ა. არ განსაზღვრავს შესამოწმებელ გამოსახულებას
- ბ. განსაზღვრავს შესამოწმებელ გამოსახულებას
- გ. განსაზღვრავს წილად რიცხვებს

5.7.6.1.3. LIKE ოპერატორში 'შაბლონი':

- ა. შეიცავს სიმბოლო_შემცვლელებს
- ბ. არ შეიცავს სიმბოლო_შემცვლელებს
- გ. შეიცავს ფუნქციის სახელებს

5.7.6.1.4. LIKE ოპერატორის შაბლონში გამოიყენება დარეზერვებული სიმბოლოები:

- ა. %, _, @ და ^
- ბ. \$, [,] და ^
- გ. %, _, [,], # და ^

5.7.6.1.5. 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

5.7.6.1.6. LIKE ოპერატორში % სიმბოლო:

- ა. ცვლის სიმბოლოების ნებისმიერ მიმდევრობას
- ბ. არ ცვლის სიმბოლოების ნებისმიერ მიმდევრობას
- გ. ცვლის '%' სიმბოლოს

5.7.6.1.7. LIKE ოპერატორში _ (ხაზგასმის სიმბოლო) სიმბოლო:

- ა. ცვლის ნებისმიერ ერთ სიმბოლოს
- ბ. ცვლის სიმბოლოების ნებისმიერ მიმდევრობას
- გ. არ ცვლის ნებისმიერ ერთ სიმბოლოს

თავი 6. ფუნქციები

მომხმარებლის ფუნქციები

6.1.1.1.1. მომხმარებლის მიერ შემუშავებული ფუნქციები:

- ა. არის მონაცემთა ბაზის დამოუკიდებელი ობიექტები და მოთავსებულია შესაბამის მონაცემთა ბაზაში
- ბ. არ არის მონაცემთა ბაზის დამოუკიდებელი ობიექტები და მოთავსებულია შესაბამის მონაცემთა ბაზაში
- გ. არის მონაცემთა ბაზის დამოუკიდებელი ობიექტები და მოთავსებული არ არის შესაბამის მონაცემთა ბაზაში

6.1.1.1.2. არსებობს მომხმარებლების მიერ შემუშავებული ფუნქციების:

- ა. ორი კლასი
- ბ. სამი კლასი
- გ. ერთი კლასი

6.1.1.1.3. არსებობს მომხმარებლების მიერ შემუშავებული ფუნქციების შემდეგი ტიპები:

- ა. Scalar, Multi-statement და table
- ბ. Scalar, Inline და cursor
- გ. Scalar, Inline და Multi-statement

6.1.1.1.4. Scalar ტიპის ფუნქციები გასცემს 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

6.1.1.1.5. 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

6.1.1.1.6. Inline ტიპის ფუნქციები შეიცავს მხოლოდ:

- ა. ერთ UPDATE ბრძანებას
- ბ. ორ SELECT ბრძანებას
- გ. ერთ SELECT ბრძანებას

6.1.1.1.7. მისი საშუალებით ფორმირდება მონაცემთა ნაკრები, რომელიც გაიცემა:

- ა. SELECT ტიპის მნიშვნელობის სახით
- ბ. table ტიპის მნიშვნელობის სახით
- გ. UPDATE ტიპის მნიშვნელობის სახით

6.1.1.1.8. Multi-statement ტიპის ფუნქციები გასცემს:

- ა. UPDATE ტიპის მნიშვნელობას

- ბ. table ტიპის მნიშვნელობას
- გ. SELECT ტიპის მნიშვნელობას

6.1.1.1.9. Multi-statement ფუნქციის ტანი:

- ა. შეიძლება რამდენიმე ბრძანებას შეიცავდეს
- ბ. შეიძლება არ შეიცავდეს არც ერთ ბრძანებას
- გ. უნდა შეიცავდეს მხოლოდ ორ ბრძანებას

6.1.1.1.10. ფუნქციებს შეიძლება ჰქონდეს:

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

6.1.1.1.11. 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

6.1.1.1.12. 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

6.1.1.1.13. ფუნქციის კოდში INSERT, UPDATE და DELETE ბრძანებების გამოყენებისას:

- ა. დასაშვებია მუშაობა მხოლოდ იმ მონაცემებთან, რომელიც იწახება ფუნქციის ტანში შექმნილ int ტიპის ცვლადში
- ბ. დასაშვებია მუშაობა მხოლოდ იმ მონაცემებთან, რომელიც იწახება ფუნქციის ტანში შექმნილ table ტიპის ცვლადში
- გ. დაუშვებელია მუშაობა მხოლოდ იმ მონაცემებთან, რომელიც იწახება ფუნქციის ტანში შექმნილ table ტიპის ცვლადში

6.1.1.1.14. 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

6.1.2.1.1. CREATE FUNCTION ბრძანებაში "@პარამეტრის_სახელი მონაცემთა_ტიპი [=DEFAULT]" აგრუმენტი:

- ა. მხოლოდ მთელ რიცხვებს განსაზღვრავს

- ბ. ფუნქციის პარამეტრებს არ განსაზღვრავს
- გ. ფუნქციის პარამეტრებს განსაზღვრავს

6.1.2.1.2. 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

6.1.2.1.3. CREATE FUNCTION ბრძანებაში:

- ა. ჯერ მიეთითება პარამეტრის ტიპი, შემდეგ პარამეტრის სახელი
- ბ. ჯერ მიეთითება პარამეტრის სახელი, შემდეგ პარამეტრის ტიპი
- გ. არ მიეთითება პარამეტრის სახელი და არც პარამეტრის ტიპი

6.1.2.1.4. 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

6.1.2.1.5. DEFAULT არგუმენტი 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

6.1.2.1.6. პარამეტრები ერთმანეთისაგან:

- ა. წერტილებით გამოიყოფა
- ბ. მძიმეებით გამოიყოფა
- გ. წერტილ-მძიმეებით გამოიყოფა

6.1.2.1.7. 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

6.1.2.1.8. 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

6.1.2.1.9. თუ ფუნქციის კოდი დაშიფრული არ არის, მაშინ მისი ნახვა:

- ა. UPDATE ბრძანებით შეიძლება
- ბ. INSERT ბრძანებით შეიძლება
- გ. SELECT ბრძანებით შეიძლება

6.1.2.1.10. ფუნქცია შეიძლება მიმართავდეს მონაცემთა ბაზის სხვადასხვა ობიექტს. 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

6.1.2.1.11. ფუნქცია შეიძლება მიმართავდეს მონაცემთა ბაზის სხვადასხვა ობიექტს. ამ ობიექტების შეცვლა ან წაშლა:

- ა. გამოიწვევს ფუნქციის შესრულების შეფერხებას
- ბ. გამოიწვევს ფუნქციის შესრულების დაწყებას
- გ. არ გამოიწვევს ფუნქციის შესრულების შეფერხებას

6.1.2.1.12. BEGIN...END საკვანძო სიტყვებს შორის:

- ა. არ არის მოთავსებული ფუნქციის ტანი (კოდი)
- ბ. მოთავსებულია ფუნქციის ტანი (კოდი)
- გ. მოთავსებულია UPDATE ბრძანების კოდი

6.1.2.1.13. 'RETURN სკალარული_გამოსახულება' ფუნქციის მუშაობას:

- ა. არ ამთავრებს და შედეგს გასცემს
- ბ. ამთავრებს და შედეგს არ გასცემს
- გ. ამთავრებს და შედეგს გასცემს

Inline ტიპის ფუნქციები

6.1.3.1.1. Inline ტიპის ფუნქცია:

- ა. აბრუნებს table ტიპის მნიშვნელობას
- ბ. არ აბრუნებს table ტიპის მნიშვნელობას
- გ. აბრუნებს int ტიპის მნიშვნელობას

6.1.3.1.2. 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

6.1.3.1.3. Inline ფუნქციის თავისებურებაა ის, რომ:

- ა. view მნიშვნელობის სტრუქტურა სტატიკურად იქმნება მოთხოვნის შესრულების დროს
- ბ. table მნიშვნელობის სტრუქტურა დინამიკურად არ იქმნება მოთხოვნის შესრულების დროს
- გ. table მნიშვნელობის სტრუქტურა დინამიკურად იქმნება მოთხოვნის შესრულების დროს

6.1.3.1.4. გაცემული table მნიშვნელობა შეგვიძლია უშუალოდ გამოვიყენოთ SELECT მოთხოვნის:

- ა. WHERE განყოფილებაში ფუნქციის სახელის მითითებით
- ბ. FROM განყოფილებაში ფუნქციის სახელის მითითებით
- გ. GROUP BY განყოფილებაში ფუნქციის სახელის მითითებით

Multi-statement ფუნქციები

6.1.4.1.1. 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

6.1.4.1.2. Multi-statement ფუნქციის ტანში:

- ა. დაუშვებელია Transact_SQL-ის ნებისმიერი კონსტრუქციის გამოყენება
- ბ. დასაშვებია Transact_SQL-ის ნებისმიერი კონსტრუქციის გამოყენება
- გ. დასაშვებია მხოლოდ BEGIN...END კონსტრუქციის გამოყენება

6.1.4.1.3. Multi-statement ფუნქციის ტანში მონაცემთა დასაბრუნებელი ნაკრები:

- ა. უნდა შეივსოს INSERT ბრძანებით
- ბ. არ უნდა შეივსოს INSERT ბრძანებით
- გ. უნდა შეივსოს UPDATE ბრძანებით

6.1.4.1.4. Multi-statement ფუნქციის განსაზღვრისას table ტიპის მნიშვნელობას:

- ა. არ უნდა მიენიჭოს სახელი
- ბ. აუცილებლად უნდა მიენიჭოს სახელი
- გ. ხანდახან შეიძლება მიენიჭოს სახელი

6.1.4.1.5. 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

6.1.4.1.6. 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

6.1.4.1.7. 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 შეცვლა

6.2.1.1.1. ფუნქციის შესაცვლელად გამოიყენება:

- ა. ALTER FUNCTION ბრძანება
- ბ. CREATE TABLE ბრძანება
- გ. UPDATE TABLE ბრძანება

6.2.1.1.2. არსებული ფუნქციის შეცვლის დროს:

- ა. იცვლება მასზე დამოკიდებული ფუნქციები, შენახული პროცედურები და ტრიგერები

ბ. არ იცვლება მასზე დამოკიდებული ფუნქციები, შენახული პროცედურები და ტრიგერები

გ. არ იცვლება მასზე დამოკიდებული ფუნქციები, მაგრამ იცვლება შენახული პროცედურები და ტრიგერები

6.2.1.1.3. ALTER FUNCTION ბრძანების საშუალებით:

ა. შესაძლებელია scalar ფუნქციის გარდაქმნა inline ფუნქციად

ბ. შეუძლებელია scalar ფუნქციის გარდაქმნა multi-statement ფუნქციად

გ. შეუძლებელია scalar ფუნქციის გარდაქმნა inline ფუნქციად

6.2.1.1.4. ALTER FUNCTION ბრძანების საშუალებით:

ა. შეუძლებელია inline ფუნქციის გარდაქმნა multi-statement ფუნქციად

ბ. inline ფუნქციის გარდაქმნა scalar ფუნქციად

გ. შესაძლებელია inline ფუნქციის გარდაქმნა multi-statement ფუნქციად

ფუნქციის წაშლა

6.3.1.1.1. ფუნქციის წასაშლელად გამოიყენება ბრძანება:

ა. DROP FUNCTION

ბ. ALTER FUNCTION

გ. CREATE FUNCTION

6.3.1.1.2. DROP FUNCTION ბრძანებით:

ა. შეუძლებელია სამივე ტიპის ფუნქციის წაშლა

ბ. შესაძლებელია მხოლოდ ერთი ტიპის ფუნქციის წაშლა

გ. შესაძლებელია სამივე ტიპის ფუნქციის წაშლა

6.3.1.1.3. 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

6.3.1.1.4. 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

6.4.1.1.1. 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 ქვემოთ

6.4.1.1.2. 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

6.4.1.1.3. SIGN(რიცხვითი_გამოსახულება) ფუნქცია:

ა. მითითებული არგუმენტისთვის გასცემს უახლოეს მინიმალურ მთელ რიცხვს, ანუ ასრულებს დამრგვალებას ქვემოთ

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

6.4.1.1.4. RAND([რიცხვი]) ფუნქცია:

ა. მითითებული არგუმენტისთვის გასცემს უახლოეს მინიმალურ მთელ რიცხვს, ანუ ასრულებს დამრგვალებას ქვემოთ

ბ. გასცემს შემთხვევით წილად რიცხვს $0 \leq x < 1$ 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 ზემოთ

6.4.1.1.5. FLOOR(რიცხვითი_გამოსახულება) ფუნქცია:

ა. მითითებული არგუმენტისთვის გასცემს უახლოეს მინიმალურ მთელ რიცხვს ანუ ასრულებს დამრგვალებას ქვემოთ

ბ. გასცემს უახლოეს მთელ რიცხვს, რომელიც არგუმენტზე მეტი ან ტოლია ანუ ასრულებს დამრგვალებას ზემოთ

გ. ასრულებს 'რიცხვითი_გამოსახულება' დამრგვალებას მითითებული სიზუსტით. დამრგვალება შეიძლება შესრულდეს როგორც ათობითი წერტილის შემდეგ (სიგრძე > 0), ისე ათობით წერტილამდე (სიგრძე < 0)

6.4.1.1.6. 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 ზემოთ

გ. მითითებული არგუმენტისთვის გასცემს უახლოეს მინიმალურ მთელ რიცხვს, ანუ ასრულებს დამრგვალებას ქვემოთ

6.4.1.1.7. ROUND(რიცხვითი_გამოსახულება, სიგრძე [, ფუნქცია]) ფუნქცია:

ა. გასცემს უახლოეს მთელ რიცხვს, რომელიც არგუმენტზე მეტი ან ტოლია ანუ ასრულებს დამრგვალებას ზემოთ

ბ. მითითებული არგუმენტისთვის გასცემს უახლოეს მინიმალურ მთელ რიცხვს ანუ

ასრულებს დამრგვალებას ქვემოთ

გ. ასრულებს 'რიცხვითი_გამოსახულება' დამრგვალებას მითითებული სიზუსტით. დამრგვალება შეიძლება შესრულდეს როგორც ათობითი წერტილის შემდეგ (სიგრძე > 0), ისე ათობით წერტილამდე (სიგრძე < 0)

6.4.1.1.8. POWER(რიცხვითი_გამოსახულება, ხარისხი) ფუნქცია:

- ა. გასცემს რიცხვის ხარისხს
- ბ. გასცემს არგუმენტის კვადრატს
- გ. არგუმენტიდან იღებს კვადრატულ ფესვს

6.4.1.1.9. SQUARE(წილადი_გამოსახულება) ფუნქცია:

- ა. გასცემს არგუმენტის კვადრატს
- ბ. გასცემს π მნიშვნელობას (3.14)
- გ. გასცემს რიცხვის ხარისხს

6.4.1.1.10. SQRT(წილადი_გამოსახულება) ფუნქცია:

- ა. გასცემს რიცხვის ხარისხს
- ბ. არგუმენტიდან იღებს კვადრატულ ფესვს
- გ. გასცემს π მნიშვნელობას (3.14)

6.4.1.1.11. PI():

- ა. გასცემს π მნიშვნელობას (3.14)
- ბ. გასცემს არგუმენტის კვადრატს
- გ. ასრულებს კუთხის გარდაქმნას რადიანიდან გრადუსში

6.4.1.1.12. DEGREES(რიცხვითი_გამოსახულება) ფუნქცია:

- ა. ასრულებს კუთხის გარდაქმნას გრადუსიდან რადიანში
- ბ. რადიანებში გასცემს მითითებული კუთხის სინუსს
- გ. ასრულებს კუთხის გარდაქმნას რადიანიდან გრადუსში

6.4.1.1.13. RADIANS(რიცხვითი_გამოსახულება) ფუნქცია:

- ა. რადიანებში გასცემს მითითებული კუთხის სინუსს
- ბ. ასრულებს კუთხის გარდაქმნას რადიანიდან გრადუსში
- გ. ასრულებს კუთხის გარდაქმნას გრადუსიდან რადიანში

6.4.1.1.14. SIN(წილადი_გამოსახულება) ფუნქცია:

- ა. გასცემს არგუმენტის ექსპონენტას
- ბ. რადიანებში გასცემს მითითებული კუთხის სინუსს
- გ. რადიანებში გასცემს მითითებული კუთხის ტანგენსს

6.4.1.1.15. COS(წილადი_გამოსახულება) ფუნქცია:

- ა. რადიანებში გასცემს მითითებული კუთხის ტანგენსს
- ბ. რადიანებში გასცემს მითითებული კუთხის კოსინუსს
- გ. რადიანებში გასცემს მითითებული კუთხის კოტანგენსს

6.4.1.1.16. TAN(წილადი_გამოსახულება) ფუნქცია:

- ა. რადიანებში გასცემს მითითებული კუთხის ტანგენსს

- ბ. გასცემს არგუმენტის ექსპონენტას
- გ. რადიანებში გასცემს მითითებული კუთხის კოტანგენსს

6.4.1.1.17. COT(წილადი_გამოსახულება) ფუნქცია:

- ა. გასცემს არგუმენტის ექსპონენტას
- ბ. გასცემს არგუმენტის ათობით ლოგარითმს
- გ. რადიანებში გასცემს მითითებული კუთხის კოტანგენსს

6.4.1.1.18. EXP(წილადი_გამოსახულება) ფუნქცია:

- ა. გასცემს არგუმენტის ნატურალურ ლოგარითმს
- ბ. გასცემს არგუმენტის ათობით ლოგარითმს
- გ. გასცემს არგუმენტის ექსპონენტას

6.4.1.1.19. LOG(წილადი_გამოსახულება) ფუნქცია:

- ა. რადიანებში გასცემს მითითებული კუთხის ტანგენსს
- ბ. გასცემს არგუმენტის ნატურალურ ლოგარითმს
- გ. გასცემს არგუმენტის ათობით ლოგარითმს

6.4.1.1.20. LOG10(წილადი_გამოსახულება) ფუნქცია:

- ა. გასცემს არგუმენტის ათობით ლოგარითმს
- ბ. რადიანებში გასცემს მითითებული კუთხის ტანგენსს
- გ. გასცემს არგუმენტის ნატურალურ ლოგარითმს

სტრიქონებთან სამუშაო ფუნქციები

6.4.2.1.1. ASCII(სტრიქონი) ფუნქცია:

- ა. გასცემს სტრიქონის მარცხენა სიმბოლოს ASCII კოდს
- ბ. გასცემს სტრიქონის მარცხენა სიმბოლოს UNICODE კოდს
- გ. გასცემს არგუმენტის შესაბამის სიმბოლოს

6.4.2.1.2. UNICODE('უნიკოდ_სტრიქონი') ფუნქცია:

- ა. გასცემს სტრიქონის მარცხენა სიმბოლოს UNICODE კოდს
- ბ. გასცემს სტრიქონის მარცხენა სიმბოლოს ASCII კოდს
- გ. გასცემს არგუმენტის შესაბამის Unicode სიმბოლოს

6.4.2.1.3. CHAR(მთელრიცხვა_გამოსახულება) ფუნქცია:

- ა. გასცემს არგუმენტის ზომას ბაიტებში
- ბ. გასცემს არგუმენტის შესაბამის სიმბოლოს
- გ. გასცემს სტრიქონის მარცხენა სიმბოლოს ASCII კოდს

6.4.2.1.4. NCHAR(მთელრიცხვა_გამოსახულება) ფუნქცია:

- ა. გასცემს არგუმენტის შესაბამის სიმბოლოს
- ბ. გასცემს არგუმენტის შესაბამის Unicode სიმბოლოს
- გ. გასცემს არგუმენტის ზომას ბაიტებში

6.4.2.1.5. LEN(სტრიქონი) ფუნქცია:

- ა. გასცემს არგუმენტის შესაბამის Unicode სიმბოლოს

- ბ. შლის ინტერვალს სტრიქონის დასაწყისიდან
- გ. გასცემს არგუმენტის ზომას ბაიტებში

6.4.2.1.6. LTRIM(სტრიქონი) ფუნქცია:

- ა. გასცემს მითითებული რაოდენობის სიმბოლოს სტრიქონის დასაწყისიდან
- ბ. შლის ინტერვალს სტრიქონის ბოლოდან
- გ. შლის ინტერვალს სტრიქონის დასაწყისიდან

6.4.2.1.7. RTRIM(სტრიქონი) ფუნქცია:

- ა. გასცემს მითითებული რაოდენობის სიმბოლოს სტრიქონის დასაწყისიდან
- ბ. შლის ინტერვალს სტრიქონის დასაწყისიდან
- გ. შლის ინტერვალს სტრიქონის ბოლოდან

6.4.2.1.8. LEFT(სტრიქონი, მთელი რიცხვა_გამოსახულება) ფუნქცია:

- ა. გასცემს მითითებული რაოდენობის სიმბოლოს სტრიქონის დასაწყისიდან
- ბ. სტრიქონის სიმბოლოები ქვედა რეგისტრში გადაჰყავს
- გ. გასცემს მითითებული რაოდენობის სიმბოლოს სტრიქონის ბოლოდან

6.4.2.1.9. RIGHT(სტრიქონი, მთელი რიცხვა_გამოსახულება) ფუნქცია:

- ა. სტრიქონის სიმბოლოები ქვედა რეგისტრში გადაჰყავს
- ბ. გასცემს მითითებული რაოდენობის სიმბოლოს სტრიქონის ბოლოდან
- გ. სტრიქონის სიმბოლოები ზედა რეგისტრში გადაჰყავს

6.4.2.1.10. LOWER(სტრიქონი) ფუნქცია:

- ა. გასცემს მითითებული რაოდენობის სიმბოლოს სტრიქონის ბოლოდან
- ბ. სტრიქონის სიმბოლოები ზედა რეგისტრში გადაჰყავს
- გ. სტრიქონის სიმბოლოები ქვედა რეგისტრში გადაჰყავს

6.4.2.1.11. 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

6.4.2.1.12. 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 სიმბოლოურ მნიშვნელობად გარდაქმნის. პირველი არგუმენტია გარდასაქმნელი რიცხვი, მეორეა - თანრიგების საერთო რაოდენობა წერტილის ჩათვლით, მესამე კი - ათობითი წერტილის შემდეგ თანრიგების რაოდენობა

6.4.2.1.13. 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

6.4.2.1.14. 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 რომელიმეს მოსაძებნად უნდა მივუთითოთ ძებნის დაწყების პოზიცია

გ. ახდენს სტრიქონის ტირაჟირებას იმდენჯერ, რა რიცხვიც მითითებულია მეორე პარამეტრით

6.4.2.1.15. 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

6.4.2.1.16. 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

6.4.2.1.17. 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

6.4.2.1.18. 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

6.4.2.1.19. 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((,;,""))

6.4.2.1.20. 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

6.4.2.1.21. 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

თარიღებთან სამუშაო ფუნქციები

6.4.3.1.1. 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

6.4.3.1.2. 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

6.4.3.1.3. DAY(თარიღი) ფუნქცია:

ა. გასცემს დღეს მითითებული თარიღიდან

ბ. გასცემს თვეს მითითებული თარიღიდან

გ. გასცემს წელს მითითებული თარიღიდან

6.4.3.1.4. 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.

6.4.3.1.5. 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

6.4.3.1.6. 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

6.4.3.1.7. 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

6.4.3.1.8. 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

6.4.3.1.9. 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

კონფიგურირების ფუნქციები

6.4.4.1.1. @@CONNECTIONS ფუნქცია:

- ა. გასცემს ლოკალურ სერვერთან დაკავშირების მცდელობების რაოდენობას მისი უკანასკნელი გაშვების მომენტიდან
- ბ. გასცემს კვირის პირველი დღის მიმდინარე მნიშვნელობას
- გ. მიმდინარე მონაცემთა ბაზისათვის გასცემს timestamp მთვლელის მიმდინარე მნიშვნელობას

6.4.4.1.2. @@DATEFIRST ფუნქცია:

- ა. მიმდინარე მონაცემთა ბაზისათვის გასცემს timestamp მთვლელის მიმდინარე მნიშვნელობას
- ბ. გასცემს კვირის პირველი დღის მიმდინარე მნიშვნელობას
- გ. გასცემს მიმდინარე ენას

6.4.4.1.3. @@DBTS ფუნქცია:

- ა. მიმდინარე მონაცემთა ბაზისათვის გასცემს timestamp მთვლელის მიმდინარე მნიშვნელობას
- ბ. გასცემს მიმდინარე ენას
- გ. გასცემს კვირის პირველი დღის მიმდინარე მნიშვნელობას

6.4.4.1.4. @@LANGUAGE ფუნქცია:

- ა. გასცემს მიმდინარე ენას
- ბ. გასცემს მილიწამების რაოდენობას, რომლის განმავლობაში სერვერი დაელოდება მოთხოვნის შესრულებას. ამ დროის გავლის შემდეგ მოთხოვნის შესრულება შეწყდება.

ავტომატურად დაყენებულია -1 მნიშვნელობა, რომელიც უსასრულო ლოდინს ნიშნავს
გ. გასცემს მიმდინარე ენის იდენტიფიკატორს

6.4.4.1.5. @@LANGID ფუნქცია:

- ა. გასცემს მილიწამების რაოდენობას, რომლის განმავლობაში სერვერი დაელოდება მოთხოვნის შესრულებას. ამ დროის გავლის შემდეგ მოთხოვნის შესრულება შეწყდება. ავტომატურად დაყენებულია -1 მნიშვნელობა, რომელიც უსასრულო ლოდინს ნიშნავს
- ბ. გასცემს მიმდინარე ენის იდენტიფიკატორს
- გ. გასცემს მიმდინარე ენას

6.4.4.1.6. @@LOCK_TIMEOUT ფუნქცია:

- ა. გასცემს მიმდინარე ენას
- ბ. გასცემს მიმდინარე ენის იდენტიფიკატორს
- გ. გასცემს ერთობლივი შეერთებების მაქსიმალურად დასაშვებ რაოდენობას. ერთდროული შეერთებების მაქსიმალური რაოდენობაა 32767 და წარმოადგენს ავტომატურ მნიშვნელობას

6.4.4.1.7. @@MAX_CONNECTIONS ფუნქცია:

- ა. განსაზღვრავს ნიშნების რაოდენობას numeric და decimal ტიპებისათვის. ავტომატური მნიშვნელობაა 28 ნიშანი.
- ბ. გასცემს მიმდინარე ენას
- გ. გასცემს ერთობლივი შეერთებების მაქსიმალურად დასაშვებ რაოდენობას. ერთდროული შეერთებების მაქსიმალური რაოდენობაა 32767 და წარმოადგენს ავტომატურ მნიშვნელობას

6.4.4.1.8. @@MAX_PRECISION ფუნქცია:

- ა. გასცემს მიმდინარე ენას
- ბ. განსაზღვრავს ნიშნების რაოდენობას numeric და decimal ტიპებისათვის. ავტომატური მნიშვნელობაა 28 ნიშანი
- გ. გასცემს შენახული პროცედურების ჩადგმულობის მიმდინარე დონეს

6.4.4.1.9. @@NESTLEVEL ფუნქცია:

- ა. განსაზღვრავს ნიშნების რაოდენობას numeric და decimal ტიპებისათვის. ავტომატური მნიშვნელობაა 28 ნიშანი
- ბ. გასცემს შენახული პროცედურების ჩადგმულობის მიმდინარე დონეს
- გ. გასცემს დაშორებული სერვერის სახელს

6.4.4.1.10. @@REMSERVER ფუნქცია:

- ა. გასცემს მიმდინარე ენას
- ბ. გასცემს შენახული პროცედურების ჩადგმულობის მიმდინარე დონეს
- გ. გასცემს დაშორებული სერვერის სახელს

6.4.4.1.11. @@SPID ფუნქცია:

- ა. გასცემს მიმდინარე პროცესის საიდენტიფიკაციო ნომერს
- ბ. გასცემს დაშორებული სერვერის სახელს
- გ. გასცემს შენახული პროცედურების ჩადგმულობის მიმდინარე დონეს

6.4.4.1.12. @@SERVERNAME ფუნქცია:

- ა. გასცემს ლოკალური სერვერის სახელს

- ბ. გასცემს შენახული პროცედურების ჩადგმულობის მიმდინარე დონეს
- გ. გასცემს მიმდინარე პროცესის საიდენტიფიკაციო ნომერს

6.4.4.1.13. @@TEXTSIZE ფუნქცია:

- ა. გასცემს დაშორებული სერვერის სახელს
- ბ. გასცემს ლოკალური სერვერის სახელს
- გ. გასცემს SELECT ბრძანების მიერ image და text ტიპის მონაცემებისათვის გამოსატანი ტექსტური ბლოკის მიმდინარე მაქსიმალურ ზომას ბაიტებში

6.4.4.1.14. @@VERSION ფუნქცია:

- ა. გასცემს დაშორებული სერვერის სახელს
- ბ. გასცემს ინფორმაციას თარიღის, სერვერისა და პროცესორის ვერსიისა და ტიპის შესახებ
- გ. გასცემს SELECT ბრძანების მიერ image და text ტიპის მონაცემებისათვის გამოსატანი ტექსტური ბლოკის მიმდინარე მაქსიმალურ ზომას ბაიტებში

სისტემური ფუნქციები

6.4.5.1.1. APP_NAME() ფუნქცია:

- ა. გასცემს პროგრამა-დანართის სახელს, რომელშიც მოცემული ბრძანება სრულდება
- ბ. გასცემს მიმდინარე მომხმარებლის სახელს
- გ. გასცემს მიმდინარე თარიღსა და დროს

6.4.5.1.2. CURRENT_TIMESTAMP ფუნქცია:

- ა. გასცემს მიმდინარე მომხმარებლის სახელს
- ბ. გასცემს მიმდინარე თარიღსა და დროს
- გ. გასცემს პროგრამა-დანართის სახელს, რომელშიც მოცემული ბრძანება სრულდება

6.4.5.1.3. CURRENT_USER ფუნქცია:

- ა. გასცემს მიმდინარე მომხმარებლის სახელს
- ბ. გასცემს მითითებული გამოსახულების შესანახად საჭირო ბაიტების რაოდენობას
- გ. გასცემს მიმდინარე შეერთებაში აღძრული უკანასკნელი შეცდომის კოდს

6.4.5.1.4. DATALENGTH ფუნქცია:

- ა. გასცემს მითითებული გამოსახულების შესანახად საჭირო ბაიტების რაოდენობას
- ბ. გასცემს მითითებული მონაცემთა ბაზისათვის მიმდინარე შეერთებაში დაყენებულ ANSI NULL მნიშვნელობას
- გ. გასცემს მიმდინარე თარიღსა და დროს

6.4.5.1.5. @@ERROR ფუნქცია:

- ა. გასცემს მითითებული მონაცემთა ბაზისათვის მიმდინარე შეერთებაში დაყენებულ ANSI NULL მნიშვნელობას
- ბ. გასცემს მიმდინარე შეერთებაში აღძრული უკანასკნელი შეცდომის კოდს
- გ. გასცემს მითითებული გამოსახულების შესანახად საჭირო ბაიტების რაოდენობას

6.4.5.1.6. GETANSINULL('მონაცემთა_ბაზის_სახელი') ფუნქცია:

- ა. გასცემს ქსელში კომპიუტერის საიდენტიფიკაციო ნომერს
- ბ. გასცემს მიმდინარე შეერთებაში აღძრული უკანასკნელი შეცდომის კოდს

გ. გასცემს მითითებული მონაცემთა ბაზისათვის მიმდინარე შეერთებაში დაყენებულ ANSI NULL მნიშვნელობას

6.4.5.1.7. IDENT_INCR('ცხრილის_ან_წარმოდგენის_სახელი') ფუნქცია:

- ა. გასცემს მიმდინარე შეერთებაში აღძრული უკანასკნელი შეცდომის კოდს
- ბ. გასცემს ქსელში კომპიუტერის საიდენტიფიკაციო ნომერს
- გ. ცხრილის ან წარმოდგენის IDENTITY სვეტისათვის გასცემს ინკრემენტის მნიშვნელობას

6.4.5.1.8. HOST_ID() ფუნქცია:

- ა. გასცემს ქსელში კომპიუტერის საიდენტიფიკაციო ნომერს
- ბ. გასცემს კომპიუტერის სახელს, რომელზეც ეს ბრძანება სრულდება
- გ. ცხრილის ან წარმოდგენის IDENTITY სვეტისათვის გასცემს ინკრემენტის მნიშვნელობას

6.4.5.1.9. HOST_NAME() ფუნქცია:

- ა. ცხრილის ან წარმოდგენის IDENTITY სვეტისათვის გასცემს ინკრემენტის მნიშვნელობას
- ბ. გასცემს კომპიუტერის სახელს, რომელზეც ეს ბრძანება სრულდება
- გ. გასცემს ქსელში კომპიუტერის საიდენტიფიკაციო ნომერს

6.4.5.1.10. IDENT_SEED('ცხრილის_ან_წარმოდგენის_სახელი') ფუნქცია:

- ა. გასცემს IDENTITY სვეტში ჩასმულ უკანასკნელ მნიშვნელობას
- ბ. გასცემს კომპიუტერის სახელს, რომელზეც ეს ბრძანება სრულდება
- გ. ცხრილის ან წარმოდგენის IDENTITY სვეტისათვის გასცემს საწყის მნიშვნელობას

6.4.5.1.11. @@IDENTITY ფუნქცია:

- ა. ცხრილის ან წარმოდგენის IDENTITY სვეტისათვის გასცემს საწყის მნიშვნელობას
- ბ. გასცემს IDENTITY სვეტში ჩასმულ უკანასკნელ მნიშვნელობას
- გ. მითითებულ გამოსახულებას ამოწმებს NULL მნიშვნელობის არსებობაზე. თუ 'გამოსახულების' მნიშვნელობაა NULL, მაშინ გაიცემა 'სიდიდე' არგუმენტი, წინააღმდეგ შემთხვევაში – 'გამოსახულების' მნიშვნელობა

6.4.5.1.12. IDENTITY (მონაცემის_ტიპი [,საწყისი_მნიშვნელობა, ნაზარდი]) AS სვეტის_სახელი) ფუნქცია:

- ა. გამოიყენება INTO განყოფილების შემცველ SELECT ბრძანების ახალ ცხრილში სვეტი-მთვლელის შექმნისას. პირველი არგუმენტი მიუთითებს მონაცემის ტიპს, მეორე – საწყის მნიშვნელობას, მესამე კი – ბიჯს. ავტომატურად, მეორე და მესამე პარამეტრი 1-ის ტოლი აიღება
- ბ. გასცემს uniqueidentifier ტიპის გლობალურ საიდენტიფიკაციო ნომერს. ფუნქცია ხშირად გამოიყენება, როგორც ავტომატური მნიშვნელობა ცხრილების სვეტებისათვის. გაცემული მნიშვნელობა დიდი ალბათობით უნიკალურია პლანეტის მასშტაბით
- გ. ცხრილის ან წარმოდგენის IDENTITY სვეტისათვის გასცემს საწყის მნიშვნელობას

6.4.5.1.13. ISNULL(გამოსახულება, სიდიდე) ფუნქცია:

- ა. გასცემს uniqueidentifier ტიპის გლობალურ საიდენტიფიკაციო ნომერს. ფუნქცია ხშირად გამოიყენება, როგორც ავტომატური მნიშვნელობა ცხრილების სვეტებისათვის. გაცემული მნიშვნელობა დიდი ალბათობით უნიკალურია პლანეტის მასშტაბით
- ბ. გასცემს IDENTITY სვეტში ჩასმულ უკანასკნელ მნიშვნელობას
- გ. მითითებულ გამოსახულებას ამოწმებს NULL მნიშვნელობის არსებობაზე. თუ

'გამოსახულების' მნიშვნელობაა NULL, მაშინ გაიცემა 'სიდიდე' არგუმენტი, წინააღმდეგ შემთხვევაში – 'გამოსახულების' მნიშვნელობა

6.4.5.1.14. NEWID() ფუნქცია:

- ა. გასცემს uniqueidentifier ტიპის გლობალურ საიდენტიფიკაციო ნომერს. ფუნქცია ხშირად გამოიყენება, როგორც ავტომატური მნიშვნელობა ცხრილების სვეტებისათვის. გაცემული მნიშვნელობა დიდი ალბათობით უნიკალურია პლანეტის მასშტაბით
- ბ. გასცემს NULL მნიშვნელობას თუ ორივე არგუმენტი ერთნაირია, წინააღმდეგ შემთხვევაში გაიცემა პირველი არგუმენტი
- გ. ობიექტის სრულ სახელს შემადგენელ ნაწილებად ყოფს

6.4.5.1.15. NULLIF(გამოსახულება_1, გამოსახულება_2) ფუნქცია:

- ა. ობიექტის სრულ სახელს შემადგენელ ნაწილებად ყოფს
- ბ. გასცემს NULL მნიშვნელობას თუ ორივე არგუმენტი ერთნაირია, წინააღმდეგ შემთხვევაში გაიცემა პირველი არგუმენტი
- გ. გასცემს სტრიქონების რაოდენობას, რომლებიც დამუშავდა უკანასკნელი ბრძანებით

6.4.5.1.16. PARSENAME('ობიექტის_სახელი', სახელი_ნაწილი) ფუნქცია:

- ა. ობიექტის სრულ სახელს შემადგენელ ნაწილებად ყოფს
- ბ. გასცემს სტრიქონების რაოდენობას, რომლებიც დამუშავდა უკანასკნელი ბრძანებით
- გ. გასცემს NULL მნიშვნელობას, თუ ორივე არგუმენტი ერთნაირია, წინააღმდეგ შემთხვევაში გაიცემა პირველი არგუმენტი

6.4.5.1.17. @@ROWCOUNT ფუნქცია:

- ა. გასცემს მომხმარებლის სახელს, რომელმაც მიმდინარე შეერთება დაამყარა
- ბ. გასცემს ინდექსის სტატისტიკის უკანასკნელი გაახლების თარიღს
- გ. გასცემს სტრიქონების რაოდენობას, რომლებიც დამუშავდა უკანასკნელი ბრძანებით

6.4.5.1.18. SESSION_USER ფუნქცია:

- ა. გასცემს ინდექსის სტატისტიკის უკანასკნელი გაახლების თარიღს
- ბ. გასცემს აქტიური ტრანზაქციების რაოდენობას მიმდინარე შეერთებაში. ტრანზაქციის დაწყებისას ამ ცვლადის მნიშვნელობა 1-ით იზრდება, ტრანზაქციის დამთავრებისას კი 1-ით მცირდება
- გ. გასცემს მომხმარებლის სახელს, რომელმაც მიმდინარე შეერთება დაამყარა

6.4.5.1.19. STATS_DATE(ცხრილის_იდენტიფიკატორი, ინდექსის_იდენტიფიკატორი) ფუნქცია:

- ა. გასცემს მომხმარებლის სახელს, რომელმაც მიმდინარე შეერთება დაამყარა
- ბ. გასცემს ინდექსის სტატისტიკის უკანასკნელი გაახლების თარიღს
- გ. გასცემს მომხმარებლის სახელს, რომელმაც მიმდინარე შეერთება არ დაამყარა

6.4.5.1.20. SYSTEM_USER ფუნქცია:

- ა. გასცემს მომხმარებლის სახელს, რომელმაც მიმდინარე შეერთება დაამყარა
- ბ. გასცემს საადრიცხვო ჩანაწერის სახელს, რომელიც ასახულია მონაცემთა ბაზის მიმდინარე მომხმარებელში
- გ. გასცემს ინდექსის სტატისტიკის უკანასკნელი გაახლების თარიღს

6.4.5.1.21. @@TRANCOUNT ფუნქცია:

- ა. გაცემს აქტიური ტრანზაქციების რაოდენობას მიმდინარე შეერთებაში. ტრანზაქციის დაწყებისას ამ ცვლადის მნიშვნელობა 1-ით იზრდება, ტრანზაქციის დამთავრებისას კი 1-ით მცირდება
- ბ. გაცემს ინდექსის სტატისტიკის უკანასკნელი გაახლების თარიღს
- გ. გაცემს მომხმარებლის სახელს, რომელმაც მიმდინარე შეერთება დაამყარა

უსაფრთხოების სისტემის ფუნქციები

6.4.6.1.1. IS_MEMBER({ 'ჯგუფი' | 'როლი' }) ფუნქცია:

- ა. ამოწმებს, არის თუ არა მიმდინარე მომხმარებელი სერვერის მონაცემთა ბაზის მითითებული როლის ან Windows NT-ის ჯგუფის წევრი
- ბ. ამოწმებს მითითებული სააღრიცხვო ჩანაწერის წევრობას სერვერის ამა თუ იმ როლში
- გ. გაცემს სააღრიცხვო ჩანაწერის საიდენტიფიკაციო ნომერს მისი სახელის მიხედვით

6.4.6.1.2. IS_SRVROLEMEMBER('როლი' [, 'სააღრიცხვო_ჩანაწერი']) ფუნქცია:

- ა. გაცემს სააღრიცხვო ჩანაწერის საიდენტიფიკაციო ნომერს მისი სახელის მიხედვით
- ბ. ამოწმებს, არის თუ არა მიმდინარე მომხმარებელი სერვერის მონაცემთა ბაზის მითითებული როლის ან Windows NT-ის ჯგუფის წევრი
- გ. ამოწმებს მითითებული სააღრიცხვო ჩანაწერის წევრობას სერვერის ამა თუ იმ როლში

6.4.6.1.3. SUSER_ID(['სააღრიცხვო_ჩანაწერი']) ფუნქცია:

- ა. გაცემს სააღრიცხვო ჩანაწერის სახელს მისი საიდენტიფიკაციო ნომრის მიხედვით.
- ბ. ამოწმებს, არის თუ არა მიმდინარე მომხმარებელი სერვერის მონაცემთა ბაზის მითითებული როლის ან Windows NT-ის ჯგუფის წევრი
- გ. გაცემს სააღრიცხვო ჩანაწერის საიდენტიფიკაციო ნომერს მისი სახელის მიხედვით

6.4.6.1.4. SUSER_NAME([მომხმარებლის_იდენტიფიკატორი]) ფუნქცია:

- ა. გაცემს მითითებული სააღრიცხვო ჩანაწერის უსაფრთხოების სისტემის საიდენტიფიკაციო ნომერს
- ბ. გაცემს სააღრიცხვო ჩანაწერის სახელს მისი საიდენტიფიკაციო ნომრის მიხედვით.
- გ. ამოწმებს მითითებული სააღრიცხვო ჩანაწერის წევრობას სერვერის ამა თუ იმ როლში

6.4.6.1.5. SUSER_SID(['სააღრიცხვო_ჩანაწერი']) ფუნქცია:

- ა. გაცემს სააღრიცხვო ჩანაწერის საიდენტიფიკაციო ნომერს მისი სახელის მიხედვით
- ბ. გაცემს მითითებული სააღრიცხვო ჩანაწერის უსაფრთხოების სისტემის საიდენტიფიკაციო ნომერს
- გ. გაცემს სააღრიცხვო ჩანაწერის სახელს მისი საიდენტიფიკაციო ნომრის მიხედვით

6.4.6.1.6. SUSER_SNAME([მომხმარებლის_იდენტიფიკატორი]) ფუნქცია:

- ა. გაცემს სააღრიცხვო ჩანაწერის სახელს მისი უსაფრთხოების სისტემის საიდენტიფიკაციო ნომრის მიხედვით
- ბ. გაცემს სააღრიცხვო ჩანაწერის სახელს მისი საიდენტიფიკაციო ნომრის მიხედვით
- გ. გაცემს მომხმარებლის სახელს მისი საიდენტიფიკაციო ნომრის მიხედვით. თუ 'იდენტიფიკატორი' არგუმენტი მითითებული არ არის, მაშინ გაციემა მიმდინარე მომხმარებლის სახელი

6.4.6.1.7. USER_NAME([იდენტიფიკატორი]) ფუნქცია:

- ა. გასცემს სააღრიცხვო ჩანაწერის სახელს მისი უსაფრთხოების სისტემის საიდენტიფიკაციო ნომრის მიხედვით
- ბ. გასცემს მითითებული სააღრიცხვო ჩანაწერის უსაფრთხოების სისტემის საიდენტიფიკაციო ნომერს
- გ. გასცემს მომხმარებლის სახელს მისი საიდენტიფიკაციო ნომრის მიხედვით. თუ 'იდენტიფიკატორი' არგუმენტი მითითებული არ არის, მაშინ გაიცემა მიმდინარე მომხმარებლის სახელი

6.4.6.1.8. USER_ID(['მომხმარებელი']) ფუნქცია:

- ა. გასცემს მონაცემთა ბაზის მიმდინარე მომხმარებლის სახელს
- ბ. გასცემს მომხმარებლის ან მონაცემთა ბაზის როლის საიდენტიფიკაციო ნომერს. არგუმენტი არის მონაცემთა ბაზის მომხმარებლის სახელი
- გ. გასცემს სააღრიცხვო ჩანაწერის სახელს მისი უსაფრთხოების სისტემის საიდენტიფიკაციო ნომრის მიხედვით

6.4.6.1.9. USER ფუნქცია:

- ა. გასცემს მონაცემთა ბაზის მიმდინარე მომხმარებლის სახელს
- ბ. გასცემს სააღრიცხვო ჩანაწერის სახელს მისი უსაფრთხოების სისტემის საიდენტიფიკაციო ნომრის მიხედვით
- გ. გასცემს მომხმარებლის სახელს მისი საიდენტიფიკაციო ნომრის მიხედვით. თუ 'იდენტიფიკატორი' არგუმენტი მითითებული არ არის, მაშინ გაიცემა მიმდინარე მომხმარებლის სახელი

მეტამონაცემების მართვის ფუნქციები

6.4.7.1.1. COL_LENGTH('ცხრილის_სახელი', 'სვეტის_სახელი') ფუნქცია:

- ა. გასცემს მითითებული ცხრილის მითითებული სვეტისათვის გამოყოფილი ბაიტების რაოდენობას
- ბ. გასცემს ინფორმაციას ცხრილის სვეტის ან შენახული პროცედურის პარამეტრების შესახებ
- გ. გასცემს სვეტის სახელს ცხრილისა და სვეტის საიდენტიფიკაციო ნომრის მიხედვით

6.4.7.1.2. COL_NAME (ცხრილის_იდენტიფიკატორი, სვეტის_იდენტიფიკატორი) ფუნქცია:

- ა. გასცემს ინფორმაციას ცხრილის სვეტის ან შენახული პროცედურის პარამეტრების შესახებ
- ბ. გასცემს სვეტის სახელს ცხრილისა და სვეტის საიდენტიფიკაციო ნომრის მიხედვით
- გ. გასცემს მითითებული ცხრილის მითითებული სვეტისათვის გამოყოფილი ბაიტების რაოდენობას

6.4.7.1.3. COLUMNPROPERTY (იდენტიფიკატორი, სვეტის_სახელი, თვისება) ფუნქცია:

- ა. გასცემს ინფორმაციას ცხრილის სვეტის ან შენახული პროცედურის პარამეტრების შესახებ
- ბ. გასცემს მითითებული ცხრილის მითითებული სვეტისათვის გამოყოფილი ბაიტების რაოდენობას
- გ. გასცემს სვეტის სახელს ცხრილისა და სვეტის საიდენტიფიკაციო ნომრის მიხედვით

6.4.7.1.4. FILEPROPERTY(ფაილის_სახელი, თვისება) ფუნქცია:

- ა. გასცემს მითითებული ცხრილის მითითებული სვეტისათვის გამოყოფილი ბაიტების რაოდენობას
- ბ. გასცემს მონაცემთა ბაზის ფაილის საიდენტიფიკაციო ნომერს მისი სახელის მიხედვით
- გ. გასცემს ინფორმაციას მონაცემთა ბაზის ფაილის თვისებების შესახებ

6.4.7.1.5. FILE_ID('ფაილის_სახელი') ფუნქცია:

- ა. გასცემს ინფორმაციას ცხრილის სვეტის ან შენახული პროცედურის პარამეტრების შესახებ
- ბ. გასცემს ინფორმაციას მონაცემთა ბაზის ფაილის თვისებების შესახებ
- გ. გასცემს მონაცემთა ბაზის ფაილის საიდენტიფიკაციო ნომერს მისი სახელის მიხედვით

6.4.7.1.6. FILE_NAME (ფაილის_იდენტიფიკატორი) ფუნქცია:

- ა. გასცემს მონაცემთა ბაზის ფაილის საიდენტიფიკაციო ნომერს მისი სახელის მიხედვით
- ბ. გასცემს მონაცემთა ბაზის ფაილის სახელს მისი საიდენტიფიკაციო ნომრის მიხედვით
- გ. გასცემს სვეტის სახელს ცხრილისა და სვეტის საიდენტიფიკაციო ნომრის მიხედვით

6.4.7.1.7. FILEGROUP_ID ('ფაილების_ჯგუფის_სახელი') ფუნქცია:

- ა. გასცემს სვეტის სახელს ცხრილისა და სვეტის საიდენტიფიკაციო ნომრის მიხედვით
- ბ. გასცემს მონაცემთა ბაზის ფაილების ჯგუფის საიდენტიფიკაციო ნომერს მისი სახელის მიხედვით
- გ. გასცემს ინფორმაციას მონაცემთა ბაზის ფაილის თვისებების შესახებ

6.4.7.1.8. FILEGROUP_NAME(ფაილების_ჯგუფის_იდენტიფიკატორი) ფუნქცია:

- ა. გასცემს მონაცემთა ბაზის ფაილების ჯგუფის სახელს მისი საიდენტიფიკაციო ნომრის მიხედვით
- ბ. გასცემს მონაცემთა ბაზის ფაილის საიდენტიფიკაციო ნომერს მისი სახელის მიხედვით
- გ. გასცემს სვეტის სახელს ცხრილისა და სვეტის საიდენტიფიკაციო ნომრის მიხედვით

6.4.7.1.9. FILEGROUPPROPERTY (ფაილების_ჯგუფის_სახელი, თვისება) ფუნქცია:

- ა. გასცემს მონაცემთა ბაზის ფაილის სახელს მისი საიდენტიფიკაციო ნომრის მიხედვით
- ბ. გასცემს ინფორმაციას ცხრილის სვეტის ან შენახული პროცედურის პარამეტრების შესახებ
- გ. გასცემს ინფორმაციას ფაილების ჯგუფის თვისებების შესახებ

6.4.7.1.10. DB_ID (['მონაცემთა_ბაზის_სახელი']) ფუნქცია:

- ა. გასცემს მონაცემთა ბაზის ფაილების ჯგუფის საიდენტიფიკაციო ნომერს მისი სახელის მიხედვით
- ბ. გასცემს მონაცემთა ბაზის საიდენტიფიკაციო ნომერს. მონაცემთა ბაზების სახელების სია ინახება master მონაცემთა ბაზის sysdatabases ცხრილში
- გ. გასცემს მონაცემთა ბაზის ფაილის სახელს მისი საიდენტიფიკაციო ნომრის მიხედვით

6.4.7.1.11. DB_NAME (მონაცემთა_ბაზის_იდენტიფიკატორი) ფუნქცია:

- ა. გასცემს მონაცემთა ბაზის სახელს მისი იდენტიფიკატორის მიხედვით
- ბ. გასცემს ინფორმაციას ცხრილის სვეტის ან შენახული პროცედურის პარამეტრების შესახებ
- გ. გასცემს მონაცემთა ბაზის ფაილის სახელს მისი საიდენტიფიკაციო ნომრის მიხედვით

6.4.7.1.12. DATABASEPROPERTY (მონაცემთა_ბაზის_სახელი, თვისება) ფუნქცია:

- ა. გასცემს ინფორმაციას ცხრილის სვეტის ან შენახული პროცედურის პარამეტრების შესახებ
- ბ. გასცემს მონაცემთა ბაზის ფაილების ჯგუფის საიდენტიფიკაციო ნომერს მისი სახელის მიხედვით
- გ. გასცემს ინფორმაციას მონაცემთა ბაზის თვისებების შესახებ. database არის მონაცემთა ბაზის სახელი. property არის მონაცემთა ბაზის თვისება

6.4.7.1.13. OBJECT_ID('ობიექტი_სახელი') ფუნქცია:

- ა. გასცემს მონაცემთა ბაზის ფაილის სახელს მისი საიდენტიფიკაციო ნომრის მიხედვით
- ბ. გასცემს ინფორმაციას მონაცემთა ბაზის ფაილის თვისებების შესახებ
- გ. გასცემს მონაცემთა ბაზის მითითებული ობიექტის საიდენტიფიკაციო ნომერს

6.4.7.1.14. OBJECT_NAME (ობიექტი_იდენტიფიკატორი) ფუნქცია:

- ა. გასცემს მონაცემთა ბაზის სახელს მისი საიდენტიფიკაციო ნომრის მიხედვით
- ბ. გასცემს ინფორმაციას ცხრილის სვეტის ან შენახული პროცედურის პარამეტრების შესახებ
- გ. გასცემს მონაცემთა ბაზის ფაილების ჯგუფის სახელს მისი საიდენტიფიკაციო ნომრის მიხედვით

6.4.7.1.15. OBJECTPROPERTY (იდენტიფიკატორი, თვისება) ფუნქცია:

- ა. გასცემს ინფორმაციას მონაცემთა ბაზის ობიექტის შესახებ მისი საიდენტიფიკაციო ნომრის მიხედვით. id არის ობიექტის იდენტიფიკატორი. 'თვისება' არის ობიექტის თვისება
- ბ. გასცემს მონაცემთა ბაზის ფაილების ჯგუფის საიდენტიფიკაციო ნომერს მისი სახელის მიხედვით
- გ. გასცემს ინფორმაციას მონაცემთა ბაზის ფაილის თვისებების შესახებ

6.4.7.1.16. INDEX_COL('ცხრილის_სახელი', ინდექსის_იდენტიფიკატორი, სვეტის_იდენტიფიკატორი) ფუნქცია:

- ა. გასცემს მონაცემთა ბაზის ფაილების ჯგუფის სახელს მისი საიდენტიფიკაციო ნომრის მიხედვით
- ბ. გასცემს ცხრილის ინდექსირებული სვეტის სახელს. 'ინდექსის_იდენტიფიკატორი' არის ინდექსის საიდენტიფიკაციო ნომერი, 'სვეტის_იდენტიფიკატორი' კი – სვეტის ნომერი ინდექსში
- გ. გასცემს ინფორმაციას ცხრილის სვეტის ან შენახული პროცედურის პარამეტრების შესახებ

6.4.7.1.17. INDEXPROPERTY (ცხრილის_იდენტიფიკატორი, ინდექსის_სახელი, თვისება) ფუნქცია:

- ა. გასცემს მონაცემთა ბაზის ფაილების ჯგუფის სახელს მისი საიდენტიფიკაციო ნომრის მიხედვით
- ბ. გასცემს ინფორმაციას მონაცემთა ბაზის ფაილის თვისებების შესახებ
- გ. გასცემს ინფორმაციას ინდექსის თვისებების შესახებ. 'ცხრილის_იდენტიფიკატორი' ცხრილის იდენტიფიკატორია, 'თვისება' კი - ინდექსის თვისება

6.4.7.1.18. @@PROCID ფუნქცია:

- ა. გასცემს მონაცემთა ბაზის ფაილების ჯგუფის საიდენტიფიკაციო ნომერს მისი სახელის მიხედვით
- ბ. გასცემს მოცემულ მომენტში გაშვებული შენახული პროცედურის საიდენტიფიკაციო ნომერს
- გ. გასცემს ინფორმაციას მონაცემთა ბაზის ფაილის თვისებების შესახებ

6.4.7.1.19. TYPEPROPERTY(ტიპი, თვისება) ფუნქცია:

- ა. გასცემს ინფორმაციას მონაცემთა ტიპების თვისებების შესახებ
- ბ. გასცემს ინფორმაციას მონაცემთა ბაზის ფაილის თვისებების შესახებ
- გ. გასცემს მონაცემთა ბაზის ფაილების ჯგუფის სახელს მისი საიდენტიფიკაციო ნომრის მიხედვით

სტატისტიკის ფუნქციები

6.4.8.1.1. @@CPU_BUSY ფუნქცია:

- ა. გასცემს მილიწამების რაოდენობას, რომელიც ცენტრალურმა პროცესორმა დახარჯა სერვერის უკანასკნელი გაშვების დროს
- ბ. გასცემს მიკროწამების რაოდენობას პროცესორის 1 ტაქტში
- გ. გასცემს მილიწამების რაოდენობას, რომლის განმავლობაში სერვერი ელოდებოდა შეტანა-გამოტანის ოპერაციის შესრულებას უკანასკნელი გაშვების დროს

6.4.8.1.2. @@IO_BUSY ფუნქცია:

- ა. გასცემს მიკროწამების რაოდენობას პროცესორის 1 ტაქტში
- ბ. გასცემს მილიწამების რაოდენობას, რომლის განმავლობაში სერვერი ელოდებოდა შეტანა-გამოტანის ოპერაციის შესრულებას უკანასკნელი გაშვების დროს
- გ. გასცემს მილიწამების რაოდენობას, რომელიც ცენტრალურმა პროცესორმა დახარჯა სერვერის უკანასკნელი გაშვების დროს

6.4.8.1.3. @@TIMETICKS ფუნქცია:

- ა. გასცემს პაკეტების რაოდენობას, რომელიც სერვერით გაეგზავნა კლიენტებს უკანასკნელი გაშვების მომენტიდან
- ბ. გასცემს მილიწამების რაოდენობას, რომელიც ცენტრალურმა პროცესორმა დახარჯა სერვერის უკანასკნელი გაშვების დროს
- გ. გასცემს მიკროწამების რაოდენობას პროცესორის 1 ტაქტში

6.4.8.1.4. @@PACK_SENT ფუნქცია:

- ა. გასცემს პაკეტების რაოდენობას, რომელიც სერვერით გაეგზავნა კლიენტებს უკანასკნელი გაშვების მომენტიდან
- ბ. გასცემს პაკეტების რაოდენობას, რომელიც სერვერმა მიიღო კლიენტებისაგან უკანასკნელი გაშვების მომენტიდან
- გ. გასცემს არასწორი პაკეტების რაოდენობას, რომელიც მიიღო სერვერმა უკანასკნელი გაშვების მომენტიდან

6.4.8.1.5. @@PACK_RECEIVED ფუნქცია:

- ა. გასცემს მილიწამების რაოდენობას, რომლის განმავლობაში სერვერი ელოდებოდა შეტანა-გამოტანის ოპერაციის შესრულებას უკანასკნელი გაშვების დროს

ბ. გასცემს პაკეტების რაოდენობას, რომელიც სერვერმა მიიღო კლიენტებისაგან უკანასკნელი გაშვების მომენტიდან

გ. გასცემს პაკეტების რაოდენობას, რომელიც სერვერით გაეგზავნა კლიენტებს უკანასკნელი გაშვების მომენტიდან

6.4.8.1.6. @@PACKET_ERRORS ფუნქცია:

ა. გასცემს არასწორი პაკეტების რაოდენობას, რომელიც სერვერმა მიიღო უკანასკნელი გაშვების მომენტიდან

ბ. გასცემს არასწორი პაკეტების რაოდენობას, რომელიც სერვერმა არ მიიღო უკანასკნელი გაშვების მომენტიდან

გ. გასცემს მილიწამების რაოდენობას, რომლის განმავლობაში სერვერი ელოდებოდა შეტანა-გამოტანის ოპერაციის შესრულებას უკანასკნელი გაშვების დროს

6.4.8.1.7. @@IDLE ფუნქცია:

ა. გასცემს პაკეტების რაოდენობას, რომელიც სერვერმა მიიღო კლიენტებისაგან უკანასკნელი გაშვების მომენტიდან

ბ. გასცემს მიკროწამების რაოდენობას პროცესორის 1 ტაქტში

გ. გასცემს მილიწამების რაოდენობას, რომელიც გავიდა სერვერის უკანასკნელი გაშვების მომენტიდან

6.4.8.1.8. @@TOTAL_READ ფუნქცია:

ა. გასცემს მიკროწამების რაოდენობას პროცესორის 1 ტაქტში

ბ. გასცემს მილიწამების რაოდენობას, რომელიც გავიდა სერვერის უკანასკნელი გაშვების მომენტიდან

გ. გასცემს დისკიდან ფიზიკური წაკითხვის ოპერაციების რაოდენობას, რომელიც შეასრულა სერვერმა უკანასკნელი გაშვების მომენტიდან

6.4.8.1.9. @@TOTAL_WRITE ფუნქცია:

ა. გასცემს პაკეტების რაოდენობას, რომელიც სერვერმა მიიღო კლიენტებისაგან უკანასკნელი გაშვების მომენტიდან

ბ. გასცემს დისკზე ფიზიკური ჩაწერის ოპერაციების რაოდენობას, რომელიც სერვერმა შეასრულა უკანასკნელი გაშვების მომენტიდან

გ. გასცემს პაკეტების რაოდენობას, რომელიც სერვერმა გაუგზავნა კლიენტებს უკანასკნელი გაშვების მომენტიდან

6.4.8.1.10. @@TOTAL_ERRORS ფუნქცია:

ა. გასცემს დისკური შეტანა-გამოტანის შეცდომების რაოდენობას, რომელსაც ადგილი ჰქონდა სერვერის უკანასკნელი გაშვების მომენტიდან

ბ. გასცემს პაკეტების რაოდენობას, რომელიც სერვერით გაეგზავნა კლიენტებს უკანასკნელი გაშვების მომენტიდან

გ. გასცემს მიკროწამების რაოდენობას პროცესორის 1 ტაქტში

თავი 7. შენახული პროცედურები

შესავალი

7.1.1.1.1. 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

7.1.1.1.2. შენახული პროცედურა:

- ა. უშუალოდ სერვერზე არ ინახება
- ბ. უშუალოდ სერვერზე ინახება
- გ. უშუალოდ წარმოდგენაში ინახება

7.1.1.1.3. შენახული პროცედურა:

- ა. არ არის მონაცემთა ბაზის დამოუკიდებელი ობიექტი
- ბ. არის მონაცემთა ბაზის დამოუკიდებელი ობიექტი
- გ. არის მონაცემთა ბაზის დამოუკიდებელი ობიექტი

7.1.1.1.4. შენახული პროცედურის 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

7.1.1.1.5. 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

7.1.1.1.6. შენახული პროცედურის ფლობის უფლება:

- ა. საერთოდ არ გადაიცემა
- ბ. შეგვიძლია ერთი მომხმარებლიდან მეორეს გადავცეთ
- გ. არ შეგვიძლია ერთი მომხმარებლიდან მეორეს გადავცეთ

7.1.1.1.7. 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

7.1.1.1.8. მწარმოებლობის 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

7.1.1.1.9. შენახული პროცედურის განმეორებით გამოძახების შემთხვევაში სერვერი მაშინვე:

- ა. არ დაიწყებს ბრძანებების შესრულებას
- ბ. დაიწყებს ბრძანებების შესრულებას
- გ. დაიწყებს მის მომზადებას შესასრულებლად

7.1.1.1.10. შენახული პროცედურის შესასრულებლად:

- ა. საკმარისია მისი სახელის მითითება
- ბ. საკმარისი არ არის მისი სახელის მითითება
- გ. საკმარისია მისი პარამეტრების მითითება

7.1.1.1.11. 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

7.1.1.1.12. შენახული პროცედურების გამოყენება:

- ა. ზრდის ქსელზე დატვირთვებს
- ბ. არ ამცირებს ქსელზე დატვირთვებს
- გ. ამცირებს ქსელზე დატვირთვებს

შენახული პროცედურების ტიპები

7.2.1.1.1. სისტემური შენახული პროცედურები:

- ა. შედის სერვერის შედგენილობაში და განკუთვნილია ადმინისტრაციული მოქმედებების შესასრულებლად
- ბ. არ შედის სერვერის შედგენილობაში და განკუთვნილია ადმინისტრაციული მოქმედებების შესასრულებლად
- გ. შედის სერვერის შედგენილობაში და განკუთვნილი არ არის ადმინისტრაციული მოქმედებების შესასრულებლად

7.2.1.1.2. სისტემურ შენახულ პროცედურებს:

- ა. არ აქვს sp_ პრეფიქსი
- ბ. აქვს sp_ პრეფიქსი
- გ. აქვს _sp პრეფიქსი

7.2.1.1.3. სისტემური შენახული პროცედურები:

- ა. არ იწახება master სისტემურ მონაცემთა ბაზაში და შეგვიძლია ნებისმიერი მონაცემთა ბაზიდან გამოვიძახოთ
- ბ. იწახება master სისტემურ მონაცემთა ბაზაში და არ შეგვიძლია ნებისმიერი მონაცემთა ბაზიდან გამოვიძახოთ
- გ. იწახება master სისტემურ მონაცემთა ბაზაში და შეგვიძლია ნებისმიერი მონაცემთა ბაზიდან გამოვიძახოთ

7.2.1.1.4. მომხმარებლის მიერ შექმნილი შენახული პროცედურები:

- ა. არ იწახება 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

7.2.1.1.5. 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

7.2.1.1.6. დროებითი პროცედურები არსებობს:

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

7.2.1.1.7. 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

7.2.1.1.8. პროცედურის სახელები:

- ა. # სიმბოლოთი უნდა იწყებოდეს
- ბ. * სიმბოლოთი უნდა იწყებოდეს
- გ. # სიმბოლოთი არ უნდა იწყებოდეს

7.2.1.1.9. 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

7.2.1.1.10. გლობალური დროებითი შენახული პროცედურები:

- ა. შეიძლება გამოძახებული იყოს ნებისმიერი შეერთებიდან
- ბ. საერთოდ არ შეიძლება გამოძახებული იყოს
- გ. არ შეიძლება გამოძახებული იყოს ნებისმიერი შეერთებიდან

7.2.1.1.11. გლობალური დროებითი პროცედურის განსაზღვრისას მისი სახელის წინ:

- ა. უნდა მოვათავსოთ ** სიმბოლოები
- ბ. უნდა მოვათავსოთ ## სიმბოლოები
- გ. არ უნდა მოვათავსოთ ## სიმბოლოები

7.2.1.1.12. 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

7.3.1.1.1. შენახულ პროცედურას ექნება:

- ა. მისი შემქმნელი მომხმარებლის უფლებები
- ბ. მისი შემსრულებელი მომხმარებლის უფლებები
- გ. მისი წამშლელი მომხმარებლის უფლებები

7.3.1.1.2. 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

7.3.1.1.3. 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

7.3.1.1.4. პარამეტრების მაქსიმალური რაოდენობაა:

- ა. 1024
- ბ. 2048
- გ. 256

7.3.1.1.5. პროცედურის პარამეტრები:

- ა. შეგვიძლია გამოვიყენოთ, როგორც ჩვეულებრივი ცვლადები
- ბ. შეგვიძლია გამოვიყენოთ, როგორც არაჩვეულებრივი ცვლადები
- გ. არ შეგვიძლია გამოვიყენოთ, როგორც ჩვეულებრივი ცვლადები

7.3.1.1.6. 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

7.3.1.1.7. შენახული პროცედურის შესაქმნელად გამოიყენება:

- ა. CREATE DATABASE ბრძანება
- ბ. CREATE PROCEDURE ბრძანება
- გ. CREATE INDEX ბრძანება

7.3.1.1.8. შენახული პროცედურის შექმნისას:

- ა. დაუშვებელია მფლობელის სახელის, მაგრამ დასაშვებია მონაცემთა ბაზის სახელის მითითება

- ბ. დასაშვებია მფლობელის სახელისა და მონაცემთა ბაზის სახელის მითითება
- გ. დაუშვებელია მფლობელის სახელისა და მონაცემთა ბაზის სახელის მითითება

7.3.1.1.9. შენახული პროცედურა ყოველთვის:

- ა. არ იქმნება მიმდინარე მონაცემთა ბაზაში
- ბ. იქმნება მიმდინარე მონაცემთა ბაზაში
- გ. იქმნება master მონაცემთა ბაზაში

7.3.1.1.10. შენახული პროცედურის პარამეტრების სახელები:

- ა. @ სიმბოლოთი უნდა იწყებოდეს
- ბ. # სიმბოლოთი უნდა იწყებოდეს
- გ. * სიმბოლოთი უნდა იწყებოდეს

7.3.1.1.11. შენახული პროცედურის პარამეტრები ერთმანეთისაგან გამოიყოფა:

- ა. წერტილებით
- ბ. წერტილ-მძიმეებით
- გ. მძიმეებით

7.3.1.1.12. შენახული პროცედურის პარამეტრები წარმოადგენს:

- 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

7.3.1.1.13. შენახულ პროცედურაში პარამეტრის სახელი ცვლადის სახელს:

- ა. არ უნდა ემთხვეოდეს
- ბ. უნდა ემთხვეოდეს
- გ. ხანდახან უნდა ემთხვეოდეს

7.3.1.1.14. პარამეტრს:

- ა. არ შეიძლება ნებისმიერი ტიპი ჰქონდეს, მათ შორის მომხმარებლის მიერ განსაზღვრული ტიპები
- ბ. შეიძლება ნებისმიერი ტიპი ჰქონდეს, გარდა მომხმარებლის მიერ განსაზღვრული ტიპისა
- გ. შეიძლება ნებისმიერი ტიპი ჰქონდეს, მათ შორის მომხმარებლის მიერ განსაზღვრული ტიპები

7.3.1.1.15. რაც შეეხება cursor ტიპს, ის შეგვიძლია გამოვიყენოთ:

- ა. მხოლოდ შესასვლელი პარამეტრებისათვის
- ბ. როგორც გამოსასვლელი, ისე შესასვლელი პარამეტრებისთვის
- გ. მხოლოდ გამოსასვლელი პარამეტრებისათვის

7.3.1.1.16. 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

7.3.1.1.17. 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

7.3.1.1.18. 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

7.3.1.1.19. CREATE PROCEDURE ბრძანების VARYING არგუმენტი მიუთითებს, რომ გამოსასვლელი პარამეტრი:

ა. იქნება სიმრავლე

ბ. არ იქნება სიმრავლე

გ. იქნება მთელი რიცხვი

7.3.1.1.20. 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

7.3.1.1.21. 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

7.3.1.1.22. 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

7.3.1.1.23. CREATE PROCEDURE ბრძანების AS არგუმენტი:

ა. ხანდახან იწყებს შენახული პროცედურის ტანს

- ბ. არ იწყებს შენახული პროცედურის ტანს
- გ. იწყებს შენახული პროცედურის ტანს

7.3.1.1.24. შენახული პროცედურიდან გამოსასვლელად შეგვიძლია გამოვიყენოთ:

- ა. CREATE PROCEDURE ბრძანება
- ბ. RETURN ბრძანება
- გ. UPDATE ბრძანება

7.3.1.1.25. ერთი შენახული პროცედურის ტანიდან მეორე შენახული პროცედურის გამოძახებისას:

- ა. იქმნება ჩადგმული პროცედურები
- ბ. იქმნება ჩაუდგმელი პროცედურები
- გ. არ იქმნება ჩადგმული პროცედურები

7.3.1.1.26. შენახული პროცედურების ჩადგმულობის მიმდინარე დონის მისაღებად შეგვიძლია გამოვიყენოთ:

- ა. @@CPU_BUSY ფუნქცია
- ბ. @@TOTAL_READ ფუნქცია
- გ. @@NESTLEVEL ცვლადი

7.3.1.1.27. შექმნილი შენახული პროცედურის სახელი შეიტანება ამავე მონაცემთა ბაზის:

- ა. sys.all_views სისტემურ წარმოდგენაში
- ბ. sys.objects სისტემურ წარმოდგენაში
- გ. sys.indexes სისტემურ წარმოდგენაში

7.3.1.1.28. 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

7.3.1.1.29. მიმართვების სისწორის შემოწმება ხდება შენახული პროცედურის:

- ა. სინტაქსური შემოწმების დროს
- ბ. შექმნის მომენტში
- გ. კომპილირების დროს

7.3.1.1.30. თუ შენახული პროცედურის გამოძახება არის ერთადერთი ბრძანება შესასრულებლად გადასაცემ პაკეტში, მაშინ:

- ა. აუცილებელია EXECUTE ბრძანების გამოყენება
- ბ. შენახული პროცედურა არ შესრულდება მხოლოდ მისი სახელის მითითებით
- გ. შენახული პროცედურა შესრულდება მხოლოდ მისი სახელის მითითებით

7.3.1.1.31. 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 მითითებით

7.3.1.1.32. EXECUTE ბრძანება უნდა გამოვიყენოთ აგრეთვე შენახული პროცედურის გამოძახებისას:

- ა. სხვა შენახული პროცედურიდან ან ტრიგერიდან
- ბ. სხვა ინდექსიდან
- გ. სხვა ცხრილიდან

7.3.1.1.33. 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

7.3.1.1.34. 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

7.3.1.1.35. შენახული პროცედურის გამოძახებისას პარამეტრების სახელები შეგვიძლია:

- ა. არ მივუთითოთ
- ბ. მივუთითოთ
- გ. წავშალოთ

7.3.1.1.36. შენახული პროცედურის გამოძახებისას პარამეტრების სახელები შეგვიძლია არ მივუთითოთ. ასეთ შემთხვევაში, პარამეტრების მნიშვნელობები ისეთივე მიმდევრობით:

- ა. უნდა მივუთითოთ, როგორც იყვნენ ჩამოთვლილი შენახული პროცედურის შექმნის დროს
- ბ. არ უნდა მივუთითოთ, როგორც იყვნენ ჩამოთვლილი შენახული პროცედურის შექმნის დროს
- გ. უნდა მივუთითოთ, როგორც არ იყვნენ ჩამოთვლილი შენახული პროცედურის შექმნის დროს

7.3.1.1.37. შენახული პროცედურის პარამეტრისათვის ავტომატური მნიშვნელობის მინიჭება, ჩამონათვალში მისი სახელის გამოტოვებით:

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

7.3.1.1.38. შენახული პროცედურის ბოლო პარამეტრი შეგვიძლია გამოვტოვოთ, თუ მათთვის:

- ა. განსაზღვრულია ავტომატური მნიშვნელობა
- ბ. განსაზღვრული არ არის ავტომატური მნიშვნელობა
- გ. განსაზღვრულია არავტომატური მნიშვნელობა

7.3.1.1.39. შენახული პროცედურის გამოძახებისას:

- ა. შეგვიძლია მივუთითოთ პარამეტრის სახელი მნიშვნელობასთან ერთად ან მხოლოდ მნიშვნელობა სახელის გარეშე
- ბ. არ შეგვიძლია მივუთითოთ პარამეტრის სახელი მნიშვნელობასთან ერთად ან მხოლოდ მნიშვნელობა სახელის გარეშე
- გ. შეგვიძლია მივუთითოთ პარამეტრის სახელი მნიშვნელობის გარეშე ან მხოლოდ მნიშვნელობა სახელთან ერთად

შენახული პროცედურების მართვა

შენახული პროცედურის შეცვლა

7.4.1.1.1. შენახული პროცედურის შესაცვლელად გამოიყენება:

- ა. ALTER PROCEDURE ბრძანება
- ბ. CREATE PROCEDURE ბრძანება
- გ. DELETE PROCEDURE ბრძანება

შენახული პროცედურის შესახებ ინფორმაციის მიღება

7.4.2.1.1. შენახული პროცედურის შესახებ ინფორმაციის მისაღებად:

- ა. უნდა გამოვიყენოთ sp_help პროცედურა
- ბ. არ უნდა გამოვიყენოთ sp_help პროცედურა
- გ. უნდა გამოვიყენოთ sp_rename პროცედურა

შენახული პროცედურის სახელის შეცვლა

7.4.3.1.1. შენახული პროცედურის სახელის შესაცვლელად:

- ა. გამოიყენება sp_rename ბრძანება
- ბ. გამოიყენება sp_help პროცედურა
- გ. გამოიყენება sp_procoption პროცედურა

შენახული პროცედურის წაშლა

7.4.4.1.1. შენახული პროცედურის წასაშლელად:

- ა. გამოიყენება DROP PROCEDURE ბრძანება
- ბ. გამოიყენება sp_help პროცედურა
- გ. გამოიყენება sp_rename ბრძანება

შენახული პროცედურის ავტომატურად შესრულების მართვა

7.4.5.1.1. შენახული პროცედურის ავტომატურად შესრულებისათვის:

- ა. საჭიროა მისი კონფიგურირება
- ბ. საჭირო არაა მისი კონფიგურირება
- გ. საჭიროა ტრიგერის კონფიგურირება

7.4.5.1.2. შენახული პროცედურის კონფიგურირება:

- ა. არ სრულდება მისი თვისებების ცვლილების გზით
- ბ. სრულდება მისი თვისებების ცვლილების გზით
- გ. სრულდება მისი თვისებების უცვლელობის გზით

7.4.5.1.3. შენახული პროცედურის კონფიგურირებისათვის:

- ა. გამოიყენება `sp_rename` ბრძანება
- ბ. გამოიყენება `DROP PROCEDURE` ბრძანება
- გ. გამოიყენება `sp_procoption` პროცედურა

თავი 8. ინდექსები

შესავალი

8.1.1.1.1. არაინდექსირებულ ცხრილში, რომელშიც ათასობით სტრიქონია, მონაცემების ძებნა:

- ა. დიდ დროს იკავებს
- ბ. მცირე დროს იკავებს
- გ. დროს არ იკავებს

8.1.1.1.2. მონაცემების ძებნის დაჩქარების მიზნით გამოიყენება:

- ა. ფუნქციები
- ბ. ინდექსები
- გ. ტრიგერები

8.1.1.1.3. ინდექსი არის მიმართვების ნაკრები, რომლებიც:

- ა. მოწესრიგებულია გარკვეული სტრიქონის მნიშვნელობების მიხედვით
- ბ. მოწესრიგებული არ არის გარკვეული სვეტის მნიშვნელობების მიხედვით
- გ. მოწესრიგებულია გარკვეული სვეტის მნიშვნელობების მიხედვით

8.1.1.1.4. ინდექსი მონაცემთა ბაზის:

- ა. დამოკიდებული ობიექტია
- ბ. დამოუკიდებელი ობიექტია
- გ. არანაირი ობიექტი არ არის

8.1.1.1.5. ინდექსი შეიძლება მოიცავდეს ცხრილის:

- ა. ერთ ან მეტ სვეტს
- ბ. არც ერთ სვეტს
- გ. მხოლოდ ერთ სვეტს

ინდექსების გამოყენების დაგეგმვა

8.2.1.1.1. როცა სრულდება სტრიქონების ცვლილება, მაშინ:

- ა. სრულდება მხოლოდ ინდექსების გაახლება
- ბ. სრულდება მხოლოდ მონაცემების გაახლება
- გ. საკუთრივ მონაცემების გაახლების გარდა, სრულდება ყველა ინდექსის გაახლება

8.2.1.1.2. ინდექსების გამოყენების უპირატესობაა:

- ა. მონაცემების ამორჩევის მნიშვნელოვანი დაჩქარება
- ბ. მონაცემების გაახლების (დამატების, შეცვლის და წაშლის) პროცესის შენელება
- გ. მონაცემების ამორჩევის მნიშვნელოვანი შენელება

8.2.1.1.3. ინდექსების გამოყენების ნაკლია:

- ა. მონაცემების ამორჩევის არცთუ მნიშვნელოვანი შენელება
- ბ. მონაცემების გაახლების (დამატების, შეცვლის და წაშლის) პროცესის შენელება
- გ. მონაცემების ამორჩევის მნიშვნელოვანი დაჩქარება

8.2.1.1.4. არ უნდა მოვახდინოთ იმ სვეტების ინდექსირება, რომლებიც:

- ა. არ ასრულებს არანაირ როლს მოთხოვნის შესრულების დროს
- ბ. ასრულებს არანაირ როლს მოთხოვნის შესრულების დროს
- გ. არ ასრულებს უმნიშვნელო როლს მოთხოვნის შესრულების დროს

8.2.1.1.5. ძალიან გრძელი სვეტების ინდექსირება, რომელთა სიგრძეა რამდენიმე ათეული სიმბოლო:

- ა. სასურველია
- ბ. არასასურველია
- გ. ხანდახან სასურველია

8.2.1.1.6. არსებობს ინდექსების:

- ა. ორი ტიპი: არაკლასტერული და უნიკალური
- ბ. ერთი ტიპი: უნიკალური
- გ. სამი ტიპი: კლასტერული, არაკლასტერული და უნიკალური

არაკლასტერული ინდექსი

8.2.2.1.1. არაკლასტერული ინდექსი:

- ა. ახდენს მიმართვების ორგანიზებას შესაბამის სტრიქონებზე
- ბ. არ ახდენს მიმართვების ორგანიზებას შესაბამის სტრიქონებზე
- გ. ახდენს მიმართვების ორგანიზებას შესაბამის სვეტებზე

8.2.2.1.2. ცხრილში საჭირო სტრიქონის იდენტიფიცირებისათვის არაკლასტერული ინდექსი:

- ა. არ ახდენს სპეციალური მიმთითებლების ორგანიზებას
- ბ. ახდენს სპეციალური მიმთითებლების ორგანიზებას
- გ. არაფერს არ აკეთებს

8.2.2.1.3. მიმთითებელი შეიცავს:

- ა. სამეზნი სტრიქონის ნომერს შესაბამის გვერდზე; სვეტის შემცველობას და ტრიგერების სახელებს
- ბ. ინფორმაციას იმ ფაილის საიდენტიფიკაციო ნომრის (IDfile) შესახებ, რომელშიც სტრიქონი ინახება; ინფორმაციას შენახული პროცედურების შესახებ
- გ. ინფორმაციას იმ ფაილის საიდენტიფიკაციო ნომრის (IDfile) შესახებ, რომელშიც სტრიქონი ინახება; სამეზნი სტრიქონის ნომერს შესაბამის გვერდზე; სვეტის შემცველობას

კლასტერული ინდექსი

8.2.3.1.1. 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

8.2.3.1.2. კლასტერული ინდექსები მნიშვნელოვნად:

- ა. ზრდის მონაცემების ძეგლის მწარმოებლურობას

- ბ. ამცირებს მონაცემების ძეგლის მწარმოებლობას
- გ. ვერ ზრდის მონაცემების ძეგლის მწარმოებლობას

8.2.3.1.3. არაკლასტერული ინდექსის გამოყენების შემთხვევაში:

- ა. სერვერი აღარ მიმართავს ინდექსს და აღარ პოულობს საჭირო სტრიქონს ცხრილში
- ბ. აღარ შესრულდება ინდექსთან მიმართვის და სტრიქონის ძეგლის ზედმეტი ოპერაციები
- გ. სერვერი თავდაპირველად მიმართავს ინდექსს, შემდეგ პოულობს საჭირო სტრიქონს ცხრილში

8.2.3.1.4. კლასტერული ინდექსის გამოყენების შემთხვევაში, მონაცემების მომდევნო პორცია განლაგდება უშუალოდ მოძებნილი მონაცემების შემდეგ. შედეგად:

- ა. შესრულდება ინდექსთან მიმართვის და სტრიქონის ძეგლის ზედმეტი ოპერაციები
- ბ. აღარ შესრულდება ინდექსთან მიმართვის და სტრიქონის ძეგლის ზედმეტი ოპერაციები
- გ. სერვერი თავდაპირველად მიმართავს ინდექსს, შემდეგ პოულობს საჭირო სტრიქონს ცხრილში

8.2.3.1.5. კლასტერულ ინდექსად უნდა ავირჩიოთ ყველაზე:

- ა. ხშირად გამოყენებადი სვეტები
- ბ. იშვიათად გამოყენებადი სვეტები
- გ. ხშირად გამოყენებადი სტრიქონები

8.2.3.1.6. კლასტერული ინდექსი:

- ა. არ შეიძლება შეიცავდეს რამდენიმე სვეტს
- ბ. შეიძლება შეიცავდეს რამდენიმე სვეტს
- გ. შეიძლება შეიცავდეს მხოლოდ ერთ სვეტს

8.2.3.1.7. კლასტერული ინდექსში შემავალი სვეტების რაოდენობა შემლებისდაგვარად:

- ა. ძალიან ბევრი უნდა იყოს
- ბ. ბევრი უნდა იყოს
- გ. მცირე უნდა იყოს

8.2.3.1.8. კლასტერული ინდექსი ხშირად ცვალებადი სვეტებისათვის:

- ა. უნდა გამოვიყენოთ
- ბ. არ უნდა გამოვიყენოთ
- გ. ხანდახან უნდა გამოვიყენოთ

8.2.3.1.9. ხშირად ცვალებადი სვეტებისათვის უმჯობესია:

- ა. არაკლასტერული ინდექსის შექმნა
- ბ. კლასტერული ინდექსის შექმნა
- გ. არ შევქმნათ ინდექსი

8.2.3.1.10. 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

8.2.3.1.11. 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

8.2.3.1.12. 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

8.2.3.1.13. ცხრილში კლასტერული შეიძლება იყოს მხოლოდ:

ა. ორი ინდექსი

ბ. ერთი ინდექსი

გ. არც ერთი ინდექსი

უნიკალური ინდექსი

8.2.4.1.1. უნიკალური ინდექსები ინდექსირებულ ცხრილში მნიშვნელობების უნიკალურობის გარანტიას:

ა. ხანდახან იძლევა

ბ. არ იძლევა

გ. იძლევა

8.2.4.1.2. 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

8.2.4.1.3. 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

8.2.4.1.4. მონაცემების მთლიანობის უზრუნველსაყოფად უმჯობესია:

- ა. UNIQUE ან PRIMARY KEY შეზღუდვების გამოყენება
- ბ. უნიკალური ინდექსის გამოყენება
- გ. არ გამოვიყენოთ ინდექსი

შევსების ფაქტორი

8.2.5.1.1. შევსების ფაქტორი განსაზღვრავს გვერდზე მონაცემების ჩაწერის:

- ა. სიმჭიდროვეს
- ბ. სიჩქარეს
- გ. სიხშირეს

8.2.5.1.2. 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

8.2.5.1.3. რაც მეტია შევსების ფაქტორი მით 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

8.2.5.1.4. თუ ცხრილი ძირითადად გამოიყენება მხოლოდ წაკითხვისათვის, მაშინ უმჯობესია შევსების ფაქტორი დავაყენოთ:

- ა. 100%-თან ახლოს
- ბ. 50%-თან ახლოს
- გ. 10%-თან ახლოს

8.2.5.1.5. 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

8.2.5.1.6. თუ ინტენსიურად გამოიყენება ცხრილში დავაყენებთ მაღალი 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

8.2.5.1.7. ცხრილში მონაცემების ცვლილებასა და დამატებასთან ერთად:

- ა. არ იცვლება გვერდების შევსების ხარისხი
- ბ. იცვლება გვერდების შევსების ხარისხი
- გ. იცვლება გვერდების შევსების სისწრაფე

8.2.5.1.8. ინდექსის ზომის შემცირების მიზნით, პერიოდულად საჭირო:

- ა. ხდება ინდექსის წაშლა
- ბ. არ ხდება ინდექსის გადაწყობა
- გ. ხდება ინდექსის გადაწყობა

ინდექსის შექმნა

8.3.1.1.1. ინდექსის შექმნის უფლება აქვს:

- ა. მხოლოდ ფუნქციის მფლობელს
- ბ. მხოლოდ ცხრილის მფლობელს
- გ. მხოლოდ შენახული პროცედურის მფლობელს

8.3.1.1.2. ინდექსის შექმნის უფლების გადაცემა სხვა მომხმარებლისათვის:

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

8.3.1.1.3. პირველადი გასაღების შექმნისას:

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

8.3.1.1.4. ინდექსი ავტომატურად იქმნება:

- ა. CREATE DATABASE ბრძანების შესრულებისას
- ბ. UNIQUE შეზღუდვის განსაზღვრისას
- გ. DROP TABLE ბრძანების შესრულებისას

8.3.1.1.5. ინდექსის შესაქმნელად გამოიყენება:

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

8.3.1.1.6. CREATE INDEX ბრძანებაში UNIQUE არგუმენტის მითითების შემთხვევაში:

- ა. არ შეიქმნება უნიკალური ინდექსი
- ბ. შეიქმნება უნიკალური ინდექსი
- გ. შეიქმნება არაუნიკალური ინდექსი

8.3.1.1.7. უნიკალური ინდექსის შექმნისას სერვერი:

- ა. ასრულებს სვეტის წინასწარ შემოწმებას მნიშვნელობების უნიკალურობაზე
- ბ. არ ასრულებს სვეტის წინასწარ შემოწმებას მნიშვნელობების უნიკალურობაზე
- გ. არაფერს არ აკეთებს

8.3.1.1.8. 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

8.3.1.1.9. 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

8.3.1.1.10. საინდექსირებელ სვეტში სასურველია აგრეთვე NULL მნიშვნელობის შენახვის აკრძალვა, რათა:

- ა. ავიცილოთ სტრიქონებში მონაცემების გამეორება
- ბ. ავიცილოთ სვეტებში მონაცემების გამეორება
- გ. ავიცილოთ მნიშვნელობების უნიკალურობასთან დაკავშირებული პრობლემები

8.3.1.1.11. თუ სვეტში გვხვდება ორი NULL მნიშვნელობა, მაშინ სერვერი მას:

- ა. ორ ერთნაირ მნიშვნელობად ჩათვლის
- ბ. ორ სხვადასხვა მნიშვნელობად ჩათვლის
- გ. სამ ერთნაირ მნიშვნელობად ჩათვლის

8.3.1.1.12. 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

8.3.1.1.13. CREATE INDEX ბრძანებაში CLUSTERED არგუმენტი მიუთითებს, რომ შესაქმნელი ინდექსი იქნება:

- ა. არაკლასტერული
- ბ. კლასტერული
- გ. როგორც კლასტერული, ისე არაკლასტერული

8.3.1.1.14. CREATE INDEX ბრძანებაში NONCLUSTERED მიუთითებს, რომ შესაქმნელი ინდექსი იქნება:

- ა. არაკლასტერული
- ბ. კლასტერული
- გ. როგორც კლასტერული, ისე არაკლასტერული

8.3.1.1.15. თუ CREATE INDEX ბრძანებაში მითითებულია ერთზე მეტი სვეტი, მაშინ ინდექსი:

- ა. იქნება შედგენილი

- ბ. არ იქნება შედგენილი
- გ. იქნება შეუდგენელი

8.3.1.1.16. ერთი შედგენილი ინდექსი შეიძლება შეიცავდეს:

- ა. 64-მდე სვეტს
- ბ. 16-მდე სვეტს
- გ. 8-მდე სვეტს

8.3.1.1.17. 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

8.3.1.1.18. ინდექსში სვეტების სახელების მითითების მიმდევრობა გავლენას:

- ა. ახდენს მოთხოვნების შესრულების სიხშირეზე
- ბ. არ ახდენს მოთხოვნების შესრულების მწარმოებლურობაზე
- გ. ახდენს მოთხოვნების შესრულების მწარმოებლურობაზე

8.3.1.1.19. მაქსიმალური მწარმოებლურობის უზრუნველსაყოფად რეკომენდებულია:

- ა. მხოლოდ იმ სვეტების მითითება, რომლებსაც მაქსიმალური სიგრძე აქვს
- ბ. თავდაპირველად იმ სვეტების სახელების მითითება, რომლებსაც მაქსიმალური სიგრძე აქვს, შემდეგ იმ სვეტების სახელების მითითება, რომლებსაც მინიმალური სიგრძე აქვს
- გ. თავდაპირველად იმ სვეტების სახელების მითითება, რომლებსაც მინიმალური სიგრძე აქვს, შემდეგ იმ სვეტების სახელების მითითება, რომლებსაც მაქსიმალური სიგრძე აქვს

8.3.1.1.20. CREATE INDEX ბრძანებაში [ASC | DESC] არგუმენტი ინდექსში:

- ა. არ განსაზღვრავს საკვანძო ელემენტების დახარისხების მეთოდს
- ბ. განსაზღვრავს საკვანძო ელემენტების დახარისხების მეთოდს
- გ. განსაზღვრავს ინდექსის შექმნის მეთოდს

8.3.1.1.21. CREATE INDEX ბრძანებაში თუ მითითებულია ASC არგუმენტი, მაშინ მონაცემები დალაგდება:

- ა. კლებადობის მიხედვით
- ბ. ზრდადობის მიხედვით
- გ. არც ზრდადობით და არც კლებადობით

8.3.1.1.22. CREATE INDEX ბრძანებაში თუ მითითებულია DESC არგუმენტი, მაშინ მონაცემები დალაგდება:

- ა. კლებადობის მიხედვით
- ბ. ზრდადობის მიხედვით
- გ. არც ზრდადობით და არც კლებადობით

8.3.1.1.23. CREATE INDEX ბრძანებაში ავტომატურად:

- ა. იგულისხმება როგორც ASC, ისე DESC
- ბ. იგულისხმება DESC
- გ. იგულისხმება ASC

8.3.1.1.24. CREATE INDEX ბრძანებაში PAD_INDEX არგუმენტის მითითების შედეგად სერვერი ინდექსის თითოეული გვერდზე:

- ა. არ მოახდენს თავისუფალი სივრცის რეზერვირებას ახალი სტრიქონების ჩასმისათვის
- ბ. მოახდენს თავისუფალი სივრცის რეზერვირებას ახალი სტრიქონების ჩასმისათვის
- გ. მოახდენს თავისუფალი სივრცის რეზერვირებას ახალი სვეტების ჩასმისათვის

8.3.1.1.25. CREATE INDEX ბრძანებაში PAD_INDEX არგუმენტი უნდა გამოვიყენოთ:

- ა. FILLFACTOR არგუმენტთან ერთად
- ბ. FILLFACTOR არგუმენტის გარეშე
- გ. ყველა არგუმენტის გარეშე

8.3.1.1.26. 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

8.3.1.1.27. 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 გამოიწვია
- ბ. არ გასცემს შეტყობინებას შეცდომის შესახებ და გააუქმებს მხოლოდ იმ სტრიქონების შეცვლას, რომლებმაც დუბლირება გამოიწვია
- გ. გასცემს შეტყობინებას შეცდომის შესახებ და არ გააუქმებს მხოლოდ იმ სტრიქონების შეცვლას, რომლებმაც დუბლირება გამოიწვია

8.3.1.1.28. 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

8.3.1.1.29. 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

8.3.1.1.30. 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

8.3.1.1.31. 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

8.3.1.1.32. 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

8.3.1.1.33. 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

8.3.1.1.34. თუ CREATE INDEX ბრძანებაში STATISTICS_NORECOMPUTE არგუმენტი იღებს ON მნიშვნელობას, მაშინ:

- ა. ვადაგასული სტატისტიკა ავტომატურად გამოითვლება
- ბ. ვადაგასული სტატისტიკა ავტომატურად არ გამოითვლება
- გ. შესაძლებელია სტატისტიკის ავტომატური გაახლება

8.3.1.1.35. თუ CREATE INDEX ბრძანებაში STATISTICS_NORECOMPUTE არგუმენტი იღებს OFF მნიშვნელობას, მაშინ:

- ა. ვადაგასული სტატისტიკა ავტომატურად არ გამოითვლება
- ბ. შეუძლებელია სტატისტიკის ავტომატური გაახლება
- გ. შესაძლებელია სტატისტიკის ავტომატური გაახლება

8.3.1.1.36. 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

ინდექსების მართვა

ინდექსის შეცვლა

8.4.1.1.1. ცხრილის ან წარმოდგენის ინდექსის შესაცვლელად გამოიყენება:

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

8.4.1.1.2. ALTER INDEX ბრძანების ALL არგუმენტი მიუთითებს, რომ:

- ა. არ უნდა შეიცვალოს ცხრილთან ან წარმოდგენასთან დაკავშირებული ყველა ინდექსი
- ბ. უნდა შეიცვალოს ცხრილთან ან წარმოდგენასთან დაკავშირებული ყველა ინდექსი
- გ. უნდა შეიცვალოს ცხრილთან ან წარმოდგენასთან დაკავშირებული ინდექსების ნაწილი

8.4.1.1.3. ALTER INDEX ბრძანების REBUILD არგუმენტი მიუთითებს, რომ:

- ა. არ უნდა შესრულდეს ინდექსის ხელახალი აგება
- ბ. უნდა შესრულდეს ინდექსის ხელახალი აგება
- გ. უნდა შესრულდეს ინდექსის წაშლა

8.4.1.1.4. თუ ALTER INDEX ბრძანების SORT_IN_TEMPDB არგუმენტი იღებს ON მნიშვნელობას, მაშინ:

- ა. master მონაცემთა ბაზაში შეინახება დახარისხების შუალედური შედეგები, რომლებიც გამოყენებული იქნება ინდექსის ასაგებად
- ბ. დახარისხების შუალედური შედეგები მოთავსდება იმავე მონაცემთა ბაზაში, რომელშიც ინდექსი
- გ. tempdb მონაცემთა ბაზაში შეინახება დახარისხების შუალედური შედეგები, რომლებიც გამოყენებული იქნება ინდექსის ასაგებად

8.4.1.1.5. თუ ALTER INDEX ბრძანების SORT_IN_TEMPDB არგუმენტი იღებს OFF მნიშვნელობას, მაშინ:

- ა. tempdb მონაცემთა ბაზაში შეინახება დახარისხების შუალედური შედეგები, რომლებიც გამოყენებული იქნება ინდექსის ასაგებად
- ბ. დახარისხების შუალედური შედეგები მოთავსდება იმავე მონაცემთა ბაზაში, რომელშიც ინდექსი
- გ. დახარისხების შუალედური შედეგები არ მოთავსდება იმავე მონაცემთა ბაზაში, რომელშიც ინდექსი

ინდექსისთვის სახელის შეცვლა

8.4.2.1.1. ინდექსისთვის სახელის შესაცვლელად გამოიყენება:

- ა. sp_rename შენახული პროცედურა
- ბ. sp_help შენახული პროცედურა
- გ. DROP INDEX ბრძანება

ინდექსის წაშლა

8.4.3.1.1. ინდექსის წასაშლელად გამოიყენება:

- ა. sp_rename შენახული პროცედურა

- ბ. sp_help შენახული პროცედურა
- გ. DROP INDEX ბრძანება

8.4.3.1.2. 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

ინდექსების გადაწყობა

8.4.4.1.1. 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

8.4.4.1.2. 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

8.4.4.1.3. ინდექსის გადაწყობის გზით:

- ა. შესაძლებელია ინდექსში შემავალი სვეტების შედგენილობის შეცვლა
- ბ. შესაძლებელია ინდექსში შემავალი სვეტების შედგენილობის გაუქმება
- გ. შეუძლებელია ინდექსში შემავალი სვეტების შედგენილობის შეცვლა

8.4.4.1.4. ინდექსის გადაწყობის ერთ-ერთი გზაა:

- ა. ინდექსისთვის სახელის შეცვლა
- ბ. ინდექსის წაშლა
- გ. ინდექსის წაშლა და მისი ხელახლა შექმნა

8.4.4.1.5. ინდექსის გადაწყობის ეფექტური გზაა:

- ა. ინდექსის წაშლა და მისი ხელახლა შექმნა
- ბ. CREATE INDEX ბრძანებაში DROP_EXISTING არგუმენტის მითითება
- გ. ინდექსისთვის სახელის შეცვლა

8.4.4.1.6. ინდექსის გადაწყობის მესამე გზაა:

- ა. DBCC DBREINDEX ბრძანების გამოყენება
- ბ. sp_rename შენახული პროცედურის გამოყენება

გ. sp_help შენახული პროცედურის გამოყენება

ინდექსის შესახებ ინფორმაციის მიღება

8.4.5.1.1. ინდექსების შესახებ ინფორმაციის მისაღებად გამოიყენება:

- ა. sp_helpindex შენახული პროცედურა
- ბ. sp_help შენახული პროცედურის გამოყენება
- გ. sp_rename შენახული პროცედურის გამოყენება

8.4.5.1.2. ინდექსის თვისებების სანახავად გამოიყენება:

- ა. sp_rename შენახული პროცედურის გამოყენება
- ბ. INDEXPROPERTY ბრძანება
- გ. DBCC DBREINDEX ბრძანების გამოყენება

8.4.5.1.3. თუ ინდექსის თვისება დაყენებულია, მაშინ გაიცემა:

- ა. NULL
- ბ. 0
- გ. 1

8.4.5.1.4. თუ ინდექსის თვისება დაყენებული არაა, მაშინ გაიცემა:

- ა. NULL
- ბ. 0
- გ. 1

8.4.5.1.5. თუ ინდექსის თვისების მდგომარეობის განსაზღვრა შეუძლებელია, მაშინ გაიცემა:

- ა. NULL
- ბ. 0
- გ. 1

8.4.5.1.6. თუ ინდექსის Is Clustered თვისება 1 მდგომარეობაშია, მაშინ ინდექსი:

- ა. კლასტერულია
- ბ. არაკლასტერულია
- გ. განუსაზღვრელია

8.4.5.1.7. თუ ინდექსის Is Clustered თვისება 0 მდგომარეობაშია, მაშინ ინდექსი:

- ა. კლასტერულია
- ბ. არაკლასტერულია
- გ. განუსაზღვრელია

8.4.5.1.8. თუ ინდექსის Is Unique თვისება 1 მდგომარეობაშია, მაშინ ინდექსი:

- ა. არის უნიკალური და არაუნიკალური
- ბ. არ არის უნიკალური
- გ. უნიკალურია

8.4.5.1.9. თუ ინდექსის Is Unique თვისება 0 მდგომარეობაშია, მაშინ ინდექსი:

- ა. არ არის უნიკალური

- ბ. უნიკალურია
- გ. არის უნიკალური და არაუნიკალური

თავი 9. წარმოდგენები

შესავალი

9.1.1.1.1. წარმოდგენა არის:

- ა. ლოგიკური ცხრილი
- ბ. ფიზიკური ცხრილი
- გ. ვირტუალური ცხრილი

9.1.1.1.2. წარმოდგენა შედგება სვეტებისა და სტრიქონებისაგან, რომლებიც:

- ა. დინამიკურად ამოირჩევა ერთი ან მეტი ცხრილიდან და/ან წარმოდგენიდან
- ბ. სტატიკურად ამოირჩევა ერთი ან მეტი ცხრილიდან ან წარმოდგენიდან
- გ. დინამიკურად ამოირჩევა მხოლოდ ორი ცხრილიდან

9.1.1.1.3. ფიზიკურად წარმოდგენას აქვს:

- ა. INSERT მოთხოვნის სახე, რომლის საფუძველზე სრულდება მონაცემების ამორჩევა
- ბ. SELECT მოთხოვნის სახე, რომლის საფუძველზე სრულდება მონაცემების ამორჩევა
- გ. UPDATE მოთხოვნის სახე, რომლის საფუძველზე სრულდება მონაცემების ამორჩევა

9.1.1.1.4. ერთი წარმოდგენა შეიძლება შეიცავდეს:

- ა. 64-მდე სვეტს
- ბ. 128-მდე სვეტს
- გ. 1024-მდე სვეტს

9.1.1.1.5. თუ წარმოდგენაში ჩართული არ არის ცხრილის რომელიმე სვეტი, მაშინ ცხრილზე დადებულია:

- ა. ვერტიკალური ფილტრი
- ბ. ჰორიზონტალური ფილტრი
- გ. დიაგონალური ფილტრი

9.1.1.1.6. თუ მოთხოვნა შეიცავს სტრიქონების ამორჩევის პირობებს, მაშინ ცხრილზე დადებულია:

- ა. დიაგონალური ფილტრი
- ბ. ჰორიზონტალური ფილტრი
- გ. ვერტიკალური ფილტრი

9.1.1.1.7. წარმოდგენა ზმული ცხრილების სვეტებს:

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

9.1.1.1.8. 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

9.1.1.1.9. 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

9.1.1.1.10. წარმოდგენა ყოველთვის იქმნება მიმდინარე მონაცემთა ბაზაში:

- ა. model მონაცემთა ბაზაში
- ბ. master მონაცემთა ბაზაში
- გ. მიმდინარე მონაცემთა ბაზაში

9.1.1.1.11. 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

9.1.1.1.12. ერთი მომხმარებლის წარმოდგენებს:

- ა. ხანდახან შეიძლება ერთნაირი სახელები ჰქონდეს
- ბ. ერთნაირი სახელები უნდა ჰქონდეს
- გ. სხვადასხვა სახელები უნდა ჰქონდეს

9.1.1.1.13. წარმოდგენისა და ცხრილის სახელი ერთმანეთს:

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

9.1.1.1.14. 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

9.1.1.1.15. 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

9.1.1.1.16. წარმოდგენა დროებით ცხრილებს:

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

9.1.1.1.17. დროებითი წარმოდგენის შექმნა:

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

9.1.1.1.18. რადგან წარმოდგენა მონაცემებს სხვადასხვა ცხრილიდან და წარმოდგენიდან 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

9.1.1.1.19. წარმოდგენისთვის ჩვეულებრივი ან სრულტექსტოვანი ინდექსის შექმნა:

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

9.1.1.1.20. წარმოდგენის სვეტებისათვის ავტომატური მნიშვნელობები:

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

9.1.1.1.21. 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

9.1.1.1.22. წარმოდგენის საშუალებით:

- ა. არ შეიძლება მონაცემების ცვლილება იმ ცხრილებში, რომლებიც წარმოდგენაში მონაწილეობს
- ბ. შეიძლება მონაცემების ცვლილება იმ ცხრილებში, რომლებიც წარმოდგენაში მონაწილეობს
- გ. შეიძლება მონაცემების ცვლილება იმ ცხრილებში, რომლებიც წარმოდგენაში არ მონაწილეობს

9.1.1.1.23. წარმოდგენის საშუალებით რამდენიმე ცხრილში მონაცემების შეცვლა:

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

9.1.1.1.24. 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

წარმოდგენები და უსაფრთხოება

9.2.1.1.1. იმისათვის, რომ შევძლოთ წარმოდგენის შექმნა, ჩვენ:

- ა. უნდა გვქონდეს მონაცემთა ბაზის მფლობელის უფლებები და შესაბამისი უფლებები იმ ცხრილსა და წარმოდგენასთან სამუშაოდ, რომლებიც წარმოდგენაში მონაწილეობენ
- ბ. არ უნდა გვქონდეს მონაცემთა ბაზის მფლობელის უფლებები და შესაბამისი უფლებები იმ ცხრილსა და წარმოდგენასთან სამუშაოდ, რომლებიც წარმოდგენაში მონაწილეობენ
- გ. უნდა გვქონდეს მონაცემთა ბაზის მფლობელის უფლებები და არ უნდა გვქონდეს უფლებები იმ ცხრილსა და წარმოდგენასთან სამუშაოდ, რომლებიც წარმოდგენაში მონაწილეობენ

9.2.1.1.2. თუ საწყისი ობიექტები და წარმოდგენა ერთ მომხმარებელს ეკუთვნის, მაშინ მომხმარებლისათვის:

- ა. საკმარისი არ არის წარმოდგენასთან მიმართვის უფლებების მინიჭება
- ბ. საკმარისია წარმოდგენასთან მიმართვის უფლებების მინიჭება
- გ. საკმარისია წარმოდგენასთან მიმართვის უფლებების ჩამორთმევა

წარმოდგენის შექმნა

9.3.1.1.1. წარმოდგენის შესაქმნელად გამოიყენება:

- ა. CREATE TABLE ბრძანება
- ბ. CREATE INDEX ბრძანება
- გ. CREATE VIEW ბრძანება

9.3.1.1.2. ავტომატურად, წარმოდგენაში სვეტების სახელები:

- ა. ემთხვევა საწყისი ცხრილების სვეტების სახელებს
- ბ. არ ემთხვევა საწყისი ცხრილების სვეტების სახელებს
- გ. ემთხვევა საწყისი ცხრილების სტრიქონების სახელებს

9.3.1.1.3. 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

9.3.1.1.4. CREATE VIEW ბრძანების ENCRYPTION არგუმენტი მიუთითებს, რომ:

- ა. არ უნდა მოხდეს წარმოდგენის კოდის დაშიფვრა
- ბ. უნდა მოხდეს წარმოდგენის კოდის დაშიფვრა
- გ. უნდა მოხდეს წარმოდგენის კოდის დაშიფვრა

9.3.1.1.5. თუ CREATE VIEW ბრძანების SCHEMABINDING არგუმენტი 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

9.3.1.1.6. 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

9.3.1.1.7. 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

9.3.1.1.8. 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

წარმოდგენების მართვა

წარმოდგენის შეცვლა

9.4.1.1.1. შექმნილი წარმოდგენის შესაცვლელად გამოიყენება:

- ა. CREATE TABLE ბრძანება
- ბ. ALTER VIEW ბრძანება
- გ. CREATE VIEW ბრძანება

წარმოდგენის სახელის შეცვლა

9.4.2.1.1. წარმოდგენის სახელის შესაცვლელად გამოიყენება:

- ა. sp_rename ბრძანება
- ბ. DBCC DBREINDEX ბრძანება
- გ. CREATE TABLE ბრძანება

წარმოდგენის წაშლა

9.4.3.1.1. წარმოდგენის წასაშლელად გამოიყენება:

- ა. CREATE VIEW ბრძანება
- ბ. DROP TABLE ბრძანება
- გ. DROP VIEW ბრძანება

9.4.3.1.2. 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

9.4.3.1.3. თუ საჭიროა, რომ სხვა მომხმარებელმა წაშლოს წარმოდგენა, მაშინ მას:

- ა. უნდა გადავცეთ წარმოდგენის ფლობის უფლება
- ბ. არ უნდა გადავცეთ წარმოდგენის ფლობის უფლება
- გ. უნდა ჩამოვართვათ წარმოდგენის ფლობის უფლება

წარმოდგენის შესახებ ინფორმაციის მიღება

9.4.4.1.1. წარმოდგენის შესახებ ინფორმაციის მისაღებად გამოიყენება:

- ა. CREATE VIEW ბრძანება
- ბ. sp_help შენახული პროცედურა
- გ. sp_rename ბრძანება

9.4.4.1.2. კოდის მისაღებად, რომლის საშუალებითაც შეიქმნა წარმოდგენა, შეგვიძლია გამოვიყენოთ:

- ა. sp_rename ბრძანება
- ბ. CREATE VIEW ბრძანება
- გ. sp_helptext შენახული პროცედურა

წარმოდგენის დამოკიდებულებების ნახვა

9.4.5.1.1. იმ ობიექტების სანახავად, რომლებზეც წარმოდგენაა დამოკიდებული, შეგვიძლია:

- ა. sp_helptext შენახული პროცედურის გამოყენება
- ბ. sp_depends შენახული პროცედურის გამოყენება
- გ. sp_rename ბრძანების გამოყენება

თავი 10. კურსორები

შესავალი

10.1.1.1.1. 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

10.1.1.1.2. 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

კურსორების რეალიზება

10.1.1.1.1. არსებობს სამი სახის კურსორი:

- ა. ფუნქციების და Transact-SQL-ის კურსორები
- ბ. Transact-SQL-ის, სერვერის და კლიენტის კურსორები
- გ. ტრიგერის, სერვერის და კლიენტის კურსორები

10.1.1.1.2. 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

10.1.1.1.3. სერვერის კურსორები ახდენს პროგრამა-დანართების პროგრამული ინტერფეისის რეალიზებას:

- ა. ODBC, OLE DB და DB-Library სისტემებისთვის
- ბ. ტრიგერებისთვის
- გ. ინდექსებისთვის

10.1.1.1.4. კლიენტის კურსორების რეალიზებას:

- ა. არ ახდენს კლიენტი
- ბ. ახდენს კლიენტი
- გ. ახდენს სერვერი

10.1.1.1.5. 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

10.1.1.1.6. 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

10.1.1.1.7. 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

10.1.1.1.8. კურსორები შეიძლება ორ კატეგორიად დაიყოს:

- ა. მიმდევრობითი და გადახვევადი
- ბ. მხოლოდ მიმდევრობითი
- გ. მხოლოდ გადახვევადი

10.1.1.1.9. მიმდევრობითი კურსორები საშუალებას გვაძლევს მონაცემები:

- ა. ორივე მიმართულებით ამოვარჩიოთ და მივმართოთ ნებისმიერ სტრიქონს
- ბ. მიმდევრობით ამოვარჩიოთ მხოლოდ ერთი მიმართულებით - ბოლოდან დასაწყისისკენ
- გ. მიმდევრობით ამოვარჩიოთ მხოლოდ ერთი მიმართულებით - დასაწყისიდან ბოლოსკენ

10.1.1.1.10. გადახვევადი კურსორები საშუალებას გვაძლევს 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

10.2.1.1.1. სტატიკური კურსორის გახსნისას სერვერი სტრიქონების მთელი შედეგობრივი ნაკრების ასლს იხაზავს:

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

10.2.1.1.2. 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

10.2.1.1.3. სტატიკური კურსორი ყოველთვის:

- ა. იხსნება „მხოლოდ წაკითხვის“ რეჟიმში
- ბ. არ იხსნება „მხოლოდ წაკითხვის“ რეჟიმში
- გ. იხსნება „მხოლოდ ჩაწერის“ რეჟიმში

10.2.1.1.4. თუ სხვა მომხმარებლები საწყის ცხრილში ამატებენ, ცვლიან ან შლიან კურსორში შესულ სტრიქონებს, მაშინ ეს ცვლილებები სტატკურ კურსორში:

- ა. ხანდახან აისახება
- ბ. არ აისახება
- გ. აისახება

დინამიკური კურსორები

10.2.2.1.1. დინამიკური კურსორის გამოყენების დროს სრულდება სტრიქონების დინამიკურად 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

10.2.2.1.2. დინამიკურ კურსორში ამორჩევის მომენტში სერვერი სტრიქონებს:

- ა. ხანდახან ბლოკავს
- ბ. არ ბლოკავს
- გ. ბლოკავს

10.2.2.1.3. კურსორის შედეგობრივ ნაკრებში მომხმარებლების მიერ შეტანილი ცვლილებები დინამიკურ კურსორში:

- ა. არ გამოჩნდება ამორჩევის დროს
- ბ. გამოჩნდება ჩაწერის დროს
- გ. გამოჩნდება ამორჩევის დროს

10.2.2.1.4. თუ სხვა მომხმარებელმა სტრიქონი შეცვალა დინამიკურ კურსორში მისი ამორჩევის შემდეგ, მაშინ ეს ცვლილებები კურსორში:

- ა. აისახება
- ბ. არ აისახება
- გ. ხანდახან აისახება

10.2.2.1.5. კურსორში შეტანილი ცვლილებები უხილავია სხვა მომხმარებლებისთვის მანამ, სანამ კურსორი:

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

10.2.2.1.6. 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

მიმდევრობითი კურსორები

10.2.3.1.1. მიმდევრობითი კურსორები საშუალებას გვაძლევს სტრიქონები ამოვირჩიოთ:

- ა. მხოლოდ ბოლოდან დასაწყისისკენ
- ბ. მხოლოდ დასაწყისიდან ბოლოსკენ
- გ. მხოლოდ დასაწყისიდან დასაწყისისკენ

10.2.3.1.2. მიმდევრობითი კურსორი სტრიქონების მთელ ნაკრებს:

- ა. ინახავს
- ბ. არ ინახავს
- გ. ხანდახან ინახავს

10.2.3.1.3. ცხრილიდან მაშინ სრულდება სტრიქონების წაკითხვა, როცა ისინი კურსორში:

- ა. არ ამოირჩევა
- ბ. ხანდახან ამოირჩევა
- გ. ამოირჩევა

10.2.3.1.4. მიმდევრობითი კურსორი იძლევა მომხმარებლის მიერ ცხრილში:

- ა. INSERT, UPDATE და DELETE ბრძანებებით შეტანილი ყველა ცვლილების დინამიკური ასახვის შესაძლებლობას
- ბ. INSERT, UPDATE და DELETE ბრძანებებით შეტანილი ყველა ცვლილების სტატიკური ასახვის შესაძლებლობას
- გ. SELECT ბრძანებებით შეტანილი ყველა ცვლილების სტატიკური ასახვის შესაძლებლობას

10.2.3.1.5. მიმდევრობით კურსორში გამოჩნდება მონაცემების:

- ა. შუალედური მდგომარეობა
- ბ. საწყისი მდგომარეობა
- გ. უკანასკნელი მდგომარეობა

საგასაღებო კურსორები

10.2.4.1.1. საგასაღებო კურსორები ან გასაღებების ნაკრებზე დამოკიდებული კურსორები:

- ა. აგებულია უნიკალური იდენტიფიკატორების ბაზაზე
- ბ. არ არის აგებული უნიკალური იდენტიფიკატორების ბაზაზე
- გ. აგებულია არაუნიკალური იდენტიფიკატორების ბაზაზე

10.2.4.1.2. ცხრილის სტრიქონების უნიკალური იდენტიფიკატორების (გასაღებების) სიმრავლეს:

- ა. ტრიგერების სიმრავლე ეწოდება
- ბ. გასაღებების სიმრავლე ეწოდება
- გ. პროცედურების სიმრავლე ეწოდება

10.2.4.1.3. საგასაღებო კურსორი წარმოადგენს გასაღებების ნაკრებს, რომლებიც:

- ა. არ ახდენს კურსორის მთელი შედეგობრივი ნაკრების სტრიქონების იდენტიფიცირებას
- ბ. ახდენს კურსორის მთელი შედეგობრივი ნაკრების სტრიქონების იდენტიფიცირებას
- გ. ახდენს კურსორის საწყისი ნაკრების სვეტების იდენტიფიცირებას

10.2.4.1.4. გასაღებების ნაკრები ინახება:

- ა. tempdb სისტემურ მონაცემთა ბაზაში

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

10.2.4.1.5. რადგან ინფორმაცია ინახება მხოლოდ კურსორის მთელ შედეგობრივ ნაკრებში ჩართული სტრიქონების საგასაღებო სვეტების შესახებ, ამიტომ საგასაღებო კურსორები:

- ა. არ ასახავს სხვა მომხმარებლების მიერ შეტანილ ყველა ცვლილებას
- ბ. არ ასახავს სხვა მომხმარებლების მიერ შეტანილ არც ერთ ცვლილებას
- გ. ასახავს სხვა მომხმარებლების მიერ შეტანილ ყველა ცვლილებას

10.2.4.1.6. კურსორის გახსნის შემდეგ დამატებული სტრიქონები კურსორში:

- ა. ხანდახან გამოჩნდება
- ბ. გამოჩნდება
- გ. არ გამოჩნდება

10.2.4.1.7. კურსორში ჩართული სტრიქონები, მაგრამ წაშლილი სხვა მომხმარებლების მიერ, გამოჩნდება როგორც:

- ა. ხანდახან დაუზიანებელი
- ბ. დაუზიანებელი
- გ. დაზიანებული

10.2.4.1.8. გასაღებების ცხრილის შედგენის მომენტში სერვერი საწყისი ცხრილის სტრიქონებს:

- ა. ბლოკავს
- ბ. არ ბლოკავს
- გ. ხანდახან ბლოკავს

კურსორების მართვა

10.3.1.1.1. 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

10.3.1.1.2. შექმნილი, მაგრამ გაუხსნელი კურსორი მონაცემებს:

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

10.3.1.1.3. გახსნის ოპერაცია კურსორს მონაცემებით:

- ა. ხანდახან ავსებს
- ბ. ავსებს
- გ. არ ავსებს

10.3.1.1.4. კურსორთან მუშაობის დამთავრების შემდეგ ის უნდა დაეხუროთ. 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

10.3.1.1.5. კურსორის გათავისუფლების დროს კურსორი:

- ა. იცვლის სახელს
- ბ. არ იშლება
- გ. იშლება

კურსორის შექმნა

10.3.2.1.1. კურსორის შესაქმნელად გამოიყენება ბრძანება:

- ა. CREATE DATABASE
- ბ. CREATE TABLE
- გ. DECLARE CURSOR

10.3.2.1.2. 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

10.3.2.1.3. 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 დინამიკური კურსორი

10.3.2.1.4. თუ SQL-92 სტანდარტით განსაზღვრული DECLARE CURSOR ბრძანების INSENSITIVE არგუმენტი მითითებული არ არის, მაშინ:

- ა. შეიქმნება დინამიკური კურსორი
- ბ. შეიქმნება სტატიკური კურსორი
- გ. არ შეიქმნება დინამიკური კურსორი

10.3.2.1.5. SQL-92 სტანდარტით განსაზღვრული DECLARE CURSOR ბრძანების SCROLL არგუმენტი მიუთითებს, რომ:

- ა. არ შეიქმნება გადახვევადი კურსორი
- ბ. შეიქმნება მიმდევრობითი კურსორი
- გ. შეიქმნება გადახვევადი კურსორი

10.3.2.1.6. 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

10.3.2.1.7. 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

10.3.2.1.8. 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

10.3.2.1.9. 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

10.3.2.1.10. 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

10.3.2.1.11. „მხოლოდ წაკითხვადი“ კურსორი:

- ა. შეიძლება იყოს დინამიკური
- ბ. შეიძლება იყოს სტატიკური
- გ. არ შეიძლება იყოს დინამიკურიც

10.3.2.1.12. თუ SQL-92 სტანდარტით განსაზღვრულ DECLARE CURSOR ბრძანებაში 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

10.3.2.1.13. თუ SQL-92 სტანდარტით განსაზღვრულ DECLARE CURSOR ბრძანებაში მითითებულია FOR UPDATE [OF სვეტის_სახელი [,...n]] არგუმენტი, მაშინ კურსორში:

- ა. დაუშვებელი იქნება მხოლოდ 'სვეტის_სახელი' სიაში მითითებული სვეტების შეცვლა
- ბ. დასაშვები იქნება მხოლოდ 'სვეტის_სახელი' სიაში მითითებული სვეტების შეცვლა
- გ. დასაშვები იქნება მხოლოდ 'სვეტის_სახელი' სიაში მითითებული სტრიქონების შეცვლა

10.3.2.1.14. 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

10.3.2.1.15. Transact-SQL სტანდარტით განსაზღვრულ 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

10.3.2.1.16. პაკეტის, ტრიგერის ან შენახული პროცედურის დამთავრებისთანავე ლოკალური კურსორი:

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

10.3.2.1.17. Transact-SQL სტანდარტით განსაზღვრულ 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

10.3.2.1.18. Transact-SQL სტანდარტით განსაზღვრულ DECLARE CURSOR ბრძანებაში FORWARD_ONLY არგუმენტი მიუთითებს, რომ:

- ა. არ შეიქმნება მიმდევრობითი კურსორი
- ბ. შეიქმნება გადახვევადი კურსორი
- გ. შეიქმნება მიმდევრობითი კურსორი

10.3.2.1.19. Transact-SQL სტანდარტით განსაზღვრულ DECLARE CURSOR ბრძანებაში SCROLL მიუთითებს, რომ:

- ა. არ შეიქმნება გადახვევადი კურსორი
- ბ. შეიქმნება გადახვევადი კურსორი
- გ. შეიქმნება მიმდევრობითი კურსორი

10.3.2.1.20. Transact-SQL სტანდარტით განსაზღვრულ DECLARE CURSOR ბრძანებაში STATIC მიუთითებს, რომ:

- ა. შეიქმნება სტატიკური კურსორი
- ბ. არ შეიქმნება სტატიკური კურსორი
- გ. შეიქმნება დინამიკური კურსორი

10.3.2.1.21. Transact-SQL სტანდარტით განსაზღვრულ DECLARE CURSOR ბრძანებაში KEYSSET არგუმენტი მიუთითებს, რომ:

- ა. შეიქმნება საგასაღებო კურსორი
- ბ. არ შეიქმნება საგასაღებო კურსორი
- გ. ხანდახან შეიქმნება საგასაღებო კურსორი

10.3.2.1.22. Transact-SQL სტანდარტით განსაზღვრულ DECLARE CURSOR ბრძანებაში DYNAMIC მიუთითებს, რომ:

- ა. არ შეიქმნება დინამიკური კურსორი
- ბ. შეიქმნება დინამიკური კურსორი
- გ. შეიქმნება სტატიკური კურსორი

10.3.2.1.23. თუ Transact-SQL სტანდარტით განსაზღვრულ DECLARE CURSOR ბრძანებაში FAST_FORWARD არგუმენტი მიუთითებულა READ_ONLY არგუმენტთან ერთად, მაშინ:

- ა. მოხდება შექმნილი კურსორის ოპტიმიზება მონაცემებთან ნელი მიმართვის მიზნით
- ბ. არ მოხდება შექმნილი კურსორის ოპტიმიზება მონაცემებთან სწრაფი მიმართვის მიზნით
- გ. მოხდება შექმნილი კურსორის ოპტიმიზება მონაცემებთან სწრაფი მიმართვის მიზნით

10.3.2.1.24. Transact-SQL სტანდარტით განსაზღვრულ DECLARE CURSOR ბრძანებაში FAST_FORWARD არგუმენტი:

- ა. არ შეიძლება გამოყენებულ იქნეს FORWARD_ONLY და OPTIMISTIC არგუმენტებთან ერთად
- ბ. შეიძლება გამოყენებულ იქნეს FORWARD_ONLY და OPTIMISTIC არგუმენტებთან ერთად
- გ. არ შეიძლება გამოყენებულ იქნეს READ_ONLY, FORWARD_ONLY და OPTIMISTIC არგუმენტებთან ერთად

10.3.2.1.25. Transact-SQL სტანდარტით განსაზღვრულ 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

10.3.2.1.26. Transact-SQL სტანდარტით განსაზღვრულ DECLARE CURSOR ბრძანებაში TYPE_WARNING არგუმენტი თუ მიუთითებულა, 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

კურსორის გახსნა

10.3.3.1.1. კურსორის გახსნის და მონაცემებით შესავსებად გამოიყენება:

- ა. CLOSE ბრძანება
- ბ. OPEN ბრძანება
- გ. INSERT ბრძანება

10.3.3.1.2. თუ საჭიროა გლობალური კურსორის გახსნა, მაშინ OPEN ბრძანებაში:

- ა. არ უნდა მივუთითოთ GLOBAL არგუმენტი
- ბ. უნდა მივუთითოთ LOCAL არგუმენტი
- გ. უნდა მივუთითოთ GLOBAL არგუმენტი

მონაცემების წაკითხვა

10.3.4.1.1. კურსორიდან მონაცემების წასაკითხად გამოიყენება:

- ა. FETCH ბრძანება
- ბ. INSERT ბრძანება
- გ. OPEN ბრძანება

10.3.4.1.2. 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

10.3.4.1.3. 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

10.3.4.1.4. 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

10.3.4.1.5. 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

10.3.4.1.6. 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

10.3.4.1.7. FETCH ბრძანების ABSOLUTE { n | @nvar } არგუმენტში n ან @nvar შეიძლება იყოს:

- ა. მხოლოდ უარყოფითი
- ბ. როგორც დადებითი, ისე უარყოფითი
- გ. მხოლოდ დადებითი

10.3.4.1.8. თუ FETCH ბრძანების ABSOLUTE { n | @nvar } არგუმენტში მითითებულია დადებითი მნიშვნელობა, მაშინ სტრიქონი:

- ა. გადაითვლება ნაკრების დასაწყისიდან
- ბ. გადაითვლება ნაკრების ბოლოდან
- გ. გადაითვლება ნაკრების შუიდან

10.3.4.1.9. თუ FETCH ბრძანების ABSOLUTE { n | @nvar } არგუმენტში მითითებულია უარყოფითი მნიშვნელობა, მაშინ სტრიქონი:

- ა. გადაითვლება ნაკრების შუიდან
- ბ. გადაითვლება ნაკრების ბოლოდან
- გ. გადაითვლება ნაკრების დასაწყისიდან

10.3.4.1.10. თუ FETCH ბრძანების ABSOLUTE { n | @nvar } არგუმენტში მითითებულია ნულოვანი მნიშვნელობა, მაშინ სტრიქონი:

- ა. ხანდახან გაიცემა
- ბ. გაიცემა
- გ. არ გაიცემა

10.3.4.1.11. FETCH ბრძანების RELATIVE { n | @nvar } არგუმენტი გასცემს სტრიქონს, რომელიც მდებარეობს მიმდინარე სტრიქონიდან n სტრიქონის შემდეგ, თუ n:

- ა. უარყოფითია
- ბ. დადებითია
- გ. ნულოვანია

10.3.4.1.12. FETCH ბრძანების RELATIVE { n | @nvar } არგუმენტი გასცემს სტრიქონს, რომელიც მდებარეობს მიმდინარე სტრიქონიდან n სტრიქონით წინ, თუ n:

- ა. უარყოფითია
- ბ. დადებითია
- გ. ნულოვანია

10.3.4.1.13. თუ FETCH ბრძანების RELATIVE { n | @nvar } არგუმენტში მითითებულია ნულოვანი მნიშვნელობა, მაშინ გაიცემა:

- ა. წინა სტრიქონი
- ბ. მიმდინარე სტრიქონი
- გ. მომდევნო სტრიქონი

10.3.4.1.14. FETCH ბრძანების INTO @variable_name [...n] არგუმენტი მიუთითებს იმ ცვლადების სიას, რომელიც:

- ა. შეიცავს დასაბრუნებელი სტრიქონის სვეტების არაშესაბამის მნიშვნელობებს
- ბ. არ შეიცავს დასაბრუნებელი სტრიქონის სვეტების შესაბამის მნიშვნელობებს
- გ. შეიცავს დასაბრუნებელი სტრიქონის სვეტების შესაბამის მნიშვნელობებს

10.3.4.1.15. 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

10.3.4.1.16. 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

10.3.4.1.17. თუ FETCH ბრძანების INTO @variable_name [...n] არგუმენტი მითითებული არ არის, მაშინ მონაცემები:

- ა. ეკრანზე გამოჩნდება
- ბ. ფაილში ჩაიწერება
- გ. ბაზაში ჩაიწერება

მონაცემების შეცვლა

10.3.5.1.1. კურსორის საშუალებით მონაცემების შესაცვლელად გამოიყენება:

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

10.3.5.1.2. 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

10.3.5.1.3. 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

10.3.5.1.4. 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

მონაცემების წაშლა

10.3.6.1.1. კურსორიდან მონაცემების წასაშლელად გამოიყენება:

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

კურსორის დახურვა

10.3.7.1.1. 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

10.3.7.1.2. კურსორის დახურვის დროს:

- ა. არ მოიხსნება კურსორის მუშაობის დროს დაყენებული ბლოკირება
- ბ. მოიხსნება კურსორის მუშაობის დროს დაყენებული ბლოკირება
- გ. მოიხსნება კურსორის მუშაობის დროს მოხსნილი ბლოკირება

10.3.7.1.3. კურსორი, რომელიც დაიხურა, მაგრამ არ გაათავისუფლდა:

- ა. არ შეიძლება განმეორებით გაიხსნას
- ბ. შეიძლება მხოლოდ ერთხელ გაიხსნას
- გ. შეიძლება განმეორებით გაიხსნას

10.3.7.1.4. კურსორის დასახურად გამოიყენება:

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

კურსორის გაათავისუფლება

10.3.8.1.1. კურსორის გაათავისუფლება:

- ა. იწვევს მის წაშლას მონაცემთა ბაზიდან
- ბ. არ იწვევს მის წაშლას მონაცემთა ბაზიდან
- გ. იწვევს მის შექმნას მონაცემთა ბაზიდან

10.3.8.1.2. კურსორის გასათავისუფლებლად გამოიყენება:

- ა. CLOSE ბრძანება
- ბ. DELETE ბრძანება
- გ. DEALLOCATE ბრძანება

თავი 11. ტრანზაქცია და დაბლოკვა

შესავალი

11.1.1.1.1. 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

11.1.1.1.2. ტრანზაქცია შეიძლება შედგებოდეს:

- ა. როგორც ერთი, ისე ასობით ბრძანებისაგან
- ბ. მხოლოდ ერთი ბრძანებისაგან
- გ. მხოლოდ ათი ბრძანებისაგან

11.1.1.1.3. თუ ყველა ბრძანება წარმატებით შესრულდა, მაშინ:

- ა. მოხდება ტრანზაქციის უკუქცევა
- ბ. მოხდება ტრანზაქციის ფიქსირება
- გ. არ მოხდება ტრანზაქციის ფიქსირება

11.1.1.1.4. 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

11.1.1.1.5. ტრანზაქციის უკუქცევის შემთხვევაში სისტემა:

- ა. აღდგება საწყის მდგომარეობაში, რომელიც ჰქონდა ტრანზაქციის დაწყებამდე
- ბ. არ აღდგება საწყის მდგომარეობაში, რომელიც ჰქონდა ტრანზაქციის დაწყებამდე
- გ. აღდგება საწყის მდგომარეობაში, რომელიც ჰქონდა ტრანზაქციის დამთავრების მომენტში

11.1.1.1.6. ინფორმაცია სისტემის საწყისი მდგომარეობის შესახებ:

- ა. ინახება ტრიგერების ჟურნალში
- ბ. ინახება ტრანზაქციების ჟურნალში
- გ. არ ინახება ტრანზაქციების ჟურნალში

11.1.1.1.7. ტრანზაქციის შესრულების დროს მონაცემთა ბაზის მართვის სისტემამ უნდა დაიცვას შემდეგი მოთხოვნები:

- ა. ატომარულობა, შეთანხმებულობა, იზოლირებულობა და მდგრადობა
- ბ. შეთანხმებულობა, იზოლირებულობა და მხოლოდ წაკითხვადობა
- გ. შეთანხმებულობა, იზოლირებულობა და მხოლოდ ჩაწერადობა

11.1.1.1.8. ატომარულობის თანახმად:

- ა. არ უნდა შესრულდეს ტრანზაქციაში მოთავსებული ყველა მოქმედება ან არც ერთი
- ბ. უნდა შესრულდეს ტრანზაქციაში მოთავსებული ყველა მოქმედება ან არც ერთი
- გ. უნდა შესრულდეს ტრანზაქციაში მოთავსებული მხოლოდ ერთი მოქმედება

11.1.1.1.9. ტრანზაქციის შესრულების შემდეგ მონაცემები:

- ა. უნდა იმყოფებოდეს შეთანხმებულ მდგომარეობაში, ე.ი. დაცული არ უნდა იყოს

მთლიანობის შეზღუდვა

ბ. არ უნდა იმყოფებოდეს შეთანხმებულ მდგომარეობაში, ე.ი. დაცული უნდა იყოს მთლიანობის შეზღუდვა

გ. უნდა იმყოფებოდეს შეთანხმებულ მდგომარეობაში, ე.ი. დაცული უნდა იყოს მთლიანობის შეზღუდვა

11.1.1.1.10. შეთანხმებულობის მოთხოვნა შეიძლება დაცული იყოს:

ა. როგორც ტრანზაქციის თითოეული ბრძანებისთვის, ისე მთლიანად ტრანზაქციისთვის

ბ. მხოლოდ ტრანზაქციის თითოეული ბრძანებისთვის

გ. მთლიანად ტრანზაქციისთვის

11.1.1.1.11. სხვადასხვა ტრანზაქციის მიერ მონაცემების ცვლილება:

ა. არ უნდა იყოს ერთმანეთისაგან იზოლირებული

ბ. უნდა იყოს ერთმანეთისაგან იზოლირებული

გ. უნდა იყოს ერთმანეთთან მჭიდრო კავშირში

11.1.1.1.12. იზოლირებულობის მოთხოვნის მიხედვით ერთი ტრანზაქციის მიერ მონაცემების ცვლილება:

ა. უნდა იყოს დამოკიდებული მეორე ტრანზაქციის მიერ მონაცემების ცვლილებაზე

ბ. არ უნდა იყოს დამოკიდებული მეორე ტრანზაქციის მიერ მონაცემების ცვლილებაზე

გ. არ უნდა იყოს დამოკიდებული ამავე ტრანზაქციის მიერ მონაცემების ცვლილებაზე

11.1.1.1.13. იზოლირებულობის მოთხოვნის მიხედვით ტრანზაქციას შეუძლია მონაცემებს მიმართოს მხოლოდ მაშინ, როცა ამ მონაცემებთან სხვა ტრანზაქცია:

ა. მუშაობს

ბ. ხანდახან მუშაობს

გ. არ მუშაობს

11.1.1.1.14. იზოლირებულობის მოთხოვნის მიხედვით ტრანზაქციას:

ა. არ შეუძლია ნახოს სხვა ტრანზაქციის მუშაობის შუალედური შედეგები

ბ. არ შეუძლია ნახოს სხვა ტრანზაქციის მუშაობის საბოლოო შედეგები

გ. შეუძლია ნახოს სხვა ტრანზაქციის მუშაობის შუალედური შედეგები

11.1.1.1.15. იზოლირებულობის მოთხოვნის მიხედვით თუ ტრანზაქცია ერთსა და იმავე მონაცემს რამდენიმეჯერ კითხულობს, მაშინ ეს მონაცემები:

ა. ხანდახან უნდა შეიცვალოს

ბ. უნდა შეიცვალოს

გ. არ უნდა შეიცვალოს

11.1.1.1.16. მდგრადობის მოთხოვნის მიხედვით ტრანზაქციის წარმატებით დამთავრების შემდეგ:

ა. ხდება მისი ფიქსირება და სისტემა წინა მდგომარეობაში ვერ დაბრუნდება

ბ. არ ხდება მისი ფიქსირება და სისტემა წინა მდგომარეობაში ვერ დაბრუნდება

გ. ხდება მისი ფიქსირება და სისტემა წინა მდგომარეობაში დაბრუნდება

11.1.1.1.17. მდგრადობის მოთხოვნის მიხედვით ცვლილებები დაფიქსირებული:

ა. არ იქნება იმ შემთხვევაშიც, თუ ტრანზაქციის დამთავრებისთანავე ადგილი ექნა

კომპიუტერის, ოპერაციული სისტემის ან სერვერის შეცდომას

ბ. იქნება იმ შემთხვევაშიც, თუ ტრანზაქციის დამთავრებისთანავე ადგილი ექნა კომპიუტერის, ოპერაციული სისტემის ან სერვერის შეცდომას

გ. იქნება იმ შემთხვევაშიც, თუ ტრანზაქციის დამთავრებისთანავე ადგილი არ ექნა კომპიუტერის, ოპერაციული სისტემის ან სერვერის შეცდომას

11.1.1.1.18. სერვერი, მომდევნო გაშვებისას:

ა. არ შეასრულებს მონაცემების აღდგენას ტრანზაქციების ჟურნალის გამოყენებით

ბ. შეასრულებს მონაცემების აღდგენას ტრანზაქციების ჟურნალის გამოყენებით

გ. შეასრულებს მონაცემების აღდგენას ტრიგერების ჟურნალის გამოყენებით

11.1.1.1.19. თუ საჭიროა მონაცემების აღდგენა ტრანზაქციის შემდეგ, მაშინ აღდგენა:

ა. უნდა შევასრულოთ მონაცემთა ბაზის ან ტრიგერების ჟურნალის სარეზერვო ასლიდან

ბ. არ უნდა შევასრულოთ მონაცემთა ბაზის ან ტრანზაქციების ჟურნალის სარეზერვო ასლიდან

გ. უნდა შევასრულოთ მონაცემთა ბაზის ან ტრანზაქციების ჟურნალის სარეზერვო ასლიდან

11.1.1.1.20. 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

11.1.1.1.21. 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

11.1.1.1.22. 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

11.1.1.1.23. დაბლოკვა არის მონაცემების დამუშავების ზოგიერთ ოპერაციაზე:

ა. იშვიათად დადებული შეზღუდვა

ბ. ხანგრძლივად დადებული შეზღუდვა

გ. დროებით დადებული შეზღუდვა

11.1.1.1.24. 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

11.1.1.1.25. რომელი მსჯელობაა სწორი:

- ა. არსებობს სხვადასხვა სახის ბლოკირებები, რომლებსაც ბლოკირებების მენეჯერი მართავს
- ბ. არ არსებობს სხვადასხვა სახის ბლოკირებები, რომლებსაც ბლოკირებების მენეჯერი მართავს
- გ. არსებობს სხვადასხვა სახის ბლოკირებები, რომლებსაც ბლოკირებების მენეჯერი არ მართავს

11.1.1.1.26. რომელი მსჯელობაა სწორი:

- ა. ტრანზაქციები მონაცემებს არ ბლოკავს ატომარულობის, შეთანხმებულობის, იზოლირებულობის და მდგრადობის მოთხოვნების შესრულების მიზნით
- ბ. ტრანზაქციები მონაცემებს ბლოკავს მხოლოდ იზოლირებულობის და მდგრადობის მოთხოვნების შესრულების მიზნით
- გ. ტრანზაქციები მონაცემებს ბლოკავს ატომარულობის, შეთანხმებულობის, იზოლირებულობის და მდგრადობის მოთხოვნების შესრულების მიზნით

11.1.1.1.27. არსებობს მონაცემებთან მიმართვის შემდეგი პრობლემები:

- ა. მხოლოდ ფანტომების კითხვის პრობლემა
- ბ. უკანასკნელი შეცვლის პრობლემა, „ჭუჭყიანი“ კითხვის პრობლემა, არაგამეორებადი კითხვის პრობლემა, ფანტომების კითხვის პრობლემა
- გ. მხოლოდ არაგამეორებადი კითხვის პრობლემა

11.1.1.1.28. როცა რამდენიმე მომხმარებელი ერთდროულად ცვლის ერთი სტრიქონის მონაცემებს, მაშინ მონაცემების ნაწილი იკარგება, რადგან ყოველი მომდევნო ტრანზაქცია:

- ა. ცვლის მომდევნო ტრანზაქციის მიერ შესრულებულ ცვლილებებს
- ბ. არ ცვლის წინა ტრანზაქციის მიერ შესრულებულ ცვლილებებს
- გ. ცვლის წინა ტრანზაქციის მიერ შესრულებულ ცვლილებებს

11.1.1.1.29. „ჭუჭყიანი“ კითხვის პრობლემას ადგილი აქვს მაშინ, როცა

- ა. მონაცემების შეცვლის მომენტში სხვა მომხმარებელი არ წაიკითხავს ამ მონაცემებს
- ბ. მონაცემების შეცვლის მომენტში სხვა მომხმარებელი წაიკითხავს ამ მონაცემებს
- გ. მონაცემების წაკითხვის მომენტში სხვა მომხმარებელი წაიკითხავს ამ მონაცემებს

11.1.1.1.30. „ჭუჭყიანი“ კითხვის პრობლემის თავიდან ასაცილებლად, მონაცემების წაკითხვა:

- ა. უნდა შესრულდეს მხოლოდ მათი დამუშავების შემდეგ
- ბ. არ უნდა შესრულდეს მხოლოდ მათი დამუშავების შემდეგ
- გ. უნდა შესრულდეს მხოლოდ მათი დამუშავების წინ

11.1.1.1.31. არაგამეორებადი კითხვის პრობლემა წამოიჭრება მაშინ, როცა ტრანზაქცია:

- ა. არც ერთხელ არ კითხულობს ერთსა და იმავე მონაცემს
- ბ. ორჯერ კითხულობს ერთსა და იმავე მონაცემს
- გ. რამდენიმეჯერ კითხულობს ერთსა და იმავე მონაცემს

11.1.1.1.32. პირველი ტრანზაქციის შესრულების დროს თუ მეორე ტრანზაქციამ მონაცემები შეცვალა, მაშინ:

- ა. პირველი ტრანზაქცია მონაცემების განმეორებით წაკითხვისას სხვა მნიშვნელობებს არ

მიიღებს

ბ. პირველი ტრანზაქცია მონაცემების განმეორებით წაკითხვისას სხვა მნიშვნელობებს მიიღებს

გ. მეორე ტრანზაქცია მონაცემების განმეორებით წაკითხვისას სხვა მნიშვნელობებს მიიღებს

11.1.1.1.33. ფანტომების კითხვის პრობლემას ადგილი აქვს მაშინ, როცა პირველი ტრანზაქცია ცხრილიდან ირჩევს სტრიქონებს, მეორე ტრანზაქცია კი ახდენს ამავე:

ა. ცხრილში სტრიქონების ჩასმას მეორე ტრანზაქციის მუშაობის დამთავრებამდე

ბ. ცხრილში სტრიქონების წაკითხვას პირველი ტრანზაქციის მუშაობის დამთავრებამდე

გ. ცხრილში სტრიქონების ჩასმას პირველი ტრანზაქციის მუშაობის დამთავრებამდე

11.1.1.1.34. არსებობს ბლოკირების:

ა. 4 დონე

ბ. 2 დონე

გ. 3 დონე

11.1.1.1.35. ნულოვანი დონე არის დაბლოკვის ყველაზე:

ა. ნაკლებად მკაცრი დონე და მოითხოვს სისტემური რესურსების მაქსიმუმს

ბ. მეტად მკაცრი დონე და მოითხოვს სისტემური რესურსების მინიმუმს

გ. ნაკლებად მკაცრი დონე და მოითხოვს სისტემური რესურსების მინიმუმს

11.1.1.1.36. დაბლოკვის ნულოვან დონეზე ხდება:

ა. მონაცემების „დაბინძურების“ აკრძალვა

ბ. ფანტომების აკრძალვა

გ. „ჭუჭყიანი“ კითხვის აკრძალვა

11.1.1.1.37. დაბლოკვის პირველ დონეზე ხდება:

ა. არაგამეორებადი კითხვის აკრძალვა

ბ. „ჭუჭყიანი“ კითხვის აკრძალვა

გ. მონაცემების „დაბინძურების“ აკრძალვა

11.1.1.1.38. დაბლოკვის მეორე დონეზე ხდება:

ა. „ჭუჭყიანი“ კითხვის აკრძალვა

ბ. მონაცემების „დაბინძურების“ აკრძალვა

გ. არაგამეორებადი კითხვის აკრძალვა

11.1.1.1.39. დაბლოკვის მესამე დონეზე ხდება:

ა. ფანტომების აკრძალვა

ბ. „ჭუჭყიანი“ კითხვის აკრძალვა

გ. მონაცემების „დაბინძურების“ აკრძალვა

11.1.1.1.40. დაბლოკვის ნულოვანი დონე ითხოვს, რომ მხოლოდ:

ა. ორ ტრანზაქციას შეეძლოს მონაცემების შეცვლა

ბ. ერთ ტრანზაქციას შეეძლოს მონაცემების შეცვლა

გ. ერთ ტრანზაქციას არ შეეძლოს მონაცემების შეცვლა

11.1.1.1.41. დაბლოკვის ნულოვანი დონის შემთხვევაში თუ მეორე ტრანზაქციას უნდა მონაცემების შეცვლა, მაშინ ის:

- ა. უნდა დაელოდოს პირველი ტრანზაქციის დამთავრებას
- ბ. არ უნდა დაელოდოს პირველი ტრანზაქციის დამთავრებას
- გ. უნდა დაელოდოს მეორე ტრანზაქციის დამთავრებას

11.1.1.1.42. დაბლოკვის პირველი დონის შემთხვევაში თუ ტრანზაქციამ დაიწყო მონაცემების შეცვლა, მაშინ სხვა ტრანზაქციებს:

- ა. არ შეეძლება ამ მონაცემების წაკითხვა მანამ, სანამ პირველი ტრანზაქცია დაიწყება
- ბ. შეეძლება ამ მონაცემების წაკითხვა მანამ, სანამ პირველი ტრანზაქცია არ დამთავრდება
- გ. არ შეეძლება ამ მონაცემების წაკითხვა მანამ, სანამ პირველი ტრანზაქცია არ დამთავრდება

11.1.1.1.43. დაბლოკვის მეორე დონის შემთხვევაში თუ ტრანზაქცია კითხულობს მონაცემებს, მაშინ სხვა ტრანზაქციას:

- ა. არ შეუძლია ამ მონაცემების შეცვლა
- ბ. შეუძლია ამ მონაცემების შეცვლა
- გ. არ შეუძლია ამ მონაცემების წაკითხვა

11.1.1.1.44. დაბლოკვის მესამე დონის შემთხვევაში თუ ტრანზაქცია მონაცემებს მიმართავს, მაშინ სხვა ტრანზაქცია:

- ა. ვერ წაკითხავს ამ მონაცემებს
- ბ. შეცვლის ამ მონაცემებს
- გ. ვერ შეცვლის ამ მონაცემებს

11.1.1.1.45. დაბლოკვის მეოთხე დონის რეალიზება:

- ა. არ სრულდება გასაღებების დიაპაზონის ბლოკირებების გამოყენებით
- ბ. სრულდება გასაღებების დიაპაზონის ბლოკირებების გამოყენებით
- გ. სრულდება გასაღებების დიაპაზონის ბლოკირებების წაკითხვით

11.1.1.1.46. დაბლოკვის მეოთხე დონის შემთხვევაში დაიბლოკება ცხრილის ის სტრიქონები, რომლებიც:

- ა. ლოგიკურ პირობას არ აკმაყოფილებს
- ბ. ლოგიკურ პირობას აკმაყოფილებს
- გ. ფიზიკურ პირობას აკმაყოფილებს

ტრანზაქციების მართვა

11.2.1.1.1. შეერთების დახურვის დროს:

- ა. ავტომატურად იხურება ტრანზაქცია
- ბ. ავტომატურად არ იხურება ტრანზაქცია
- გ. ხელით იხურება ტრანზაქცია

11.2.1.1.2. 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

11.2.1.1.3. ტრანზაქცია არის ლოკალური თუ ის სრულდება:

- ა. რამდენიმე მონაცემთა ბაზის შიგნით
- ბ. ორი მონაცემთა ბაზის შიგნით
- გ. ერთი მონაცემთა ბაზის შიგნით

11.2.1.1.4. განწილებულია ტრანზაქცია, რომელიც მიმართავს:

- ა. ერთ მონაცემთა ბაზას
- ბ. რამდენიმე მონაცემთა ბაზას, რომლებიც შეიძლება სხვადასხვა სერვერზე იყოს განთავსებული
- გ. რამდენიმე მონაცემთა ბაზას, რომლებიც არ შეიძლება სხვადასხვა სერვერზე იყოს განთავსებული

აშკარა ტრანზაქციები

11.2.2.1.1. 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

11.2.2.1.2. BEGIN TRANSACTION ბრძანება განსაზღვრავს ტრანზაქციის:

- ა. დასაწყისს
- ბ. დასასრულს
- გ. შუა ნაწილს

11.2.2.1.3. 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

11.2.2.1.4. 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

11.2.2.1.5. BEGIN TRANSACTION ბრძანების WITH MARK ['აღწერა'] არგუმენტი საშუალებას:

- ა. გვაძლევს სპეციალური გზით მოვახდინოთ ტრანზაქციების მარკირება ტრანზაქციების ჟურნალში
- ბ. არ გვაძლევს სპეციალური გზით მოვახდინოთ ტრანზაქციების მარკირება ტრანზაქციების ჟურნალში
- გ. გვაძლევს სპეციალური გზით მოვახდინოთ ტრანზაქციების მარკირება ტრიგერების

ჟურნალში

11.2.2.1.6. რომელი მსჯელობაა სწორი:

- ა. ტრანზაქციის მარკირება საჭირო არ არის სარეზერვო ასლიდან ტრანზაქციების ჟურნალის აღდგენისათვის, მონაცემთა ბაზის დასაბრუნებლად იმ მდგომარეობაში, რომელშიც ის იყო ტრანზაქციის დაწყებამდე ან მისი დამთავრების შემდეგ
- ბ. ტრანზაქციის მარკირება საჭიროა სარეზერვო ასლიდან ტრანზაქციების ჟურნალის აღდგენისათვის, მონაცემთა ბაზის დასაბრუნებლად იმ მდგომარეობაში, რომელშიც ის იყო ტრანზაქციის დაწყების შემდეგ ან მის დამთავრებამდე
- გ. ტრანზაქციის მარკირება საჭიროა სარეზერვო ასლიდან ტრანზაქციების ჟურნალის აღდგენისათვის, მონაცემთა ბაზის დასაბრუნებლად იმ მდგომარეობაში, რომელშიც ის იყო ტრანზაქციის დაწყებამდე ან მისი დამთავრების შემდეგ

11.2.2.1.7. COMMIT TRANSACTION ბრძანებება განსაზღვრავს ტრანზაქციის:

- ა. დასაწყისს
- ბ. დასასრულს
- გ. შუა ნაწილს

11.2.2.1.8. 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

11.2.2.1.9. ROLLBACK TRANSACTION ბრძანებების შესრულება იწვევს ტრანზაქციის:

- ა. შეწყვეტას და უკუქცევას
- ბ. დაფიქსირებას
- გ. მხოლოდ შეწყვეტას

11.2.2.1.10. 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

11.2.2.1.11. ROLLBACK TRANSACTION ბრძანებების 'შენახვის_წერტილის_სახელი' არგუმენტი:

- ა. არის ტრანზაქციის საკონტროლო წერტილის დასასრული
- ბ. არ არის ტრანზაქციის საკონტროლო წერტილის სახელი
- გ. არის ტრანზაქციის საკონტროლო წერტილის სახელი

11.2.2.1.12. გარკვეულ წერტილამდე ტრანზაქციის უკუქცევა საშუალებას:

- ა. გვაძლევს გავაუქმოთ მხოლოდ ტრანზაქციის ნაწილი ანუ საკონტროლო წერტილის შექმნის შემდეგ გაკეთებული ცვლილებები
- ბ. არ გვაძლევს გავაუქმოთ მხოლოდ ტრანზაქციის ნაწილი ანუ საკონტროლო წერტილის შექმნის შემდეგ გაკეთებული ცვლილებები
- გ. გვაძლევს გავაუქმოთ მხოლოდ ტრანზაქციის ნაწილი ანუ საკონტროლო წერტილის შექმნამდე გაკეთებული ცვლილებები

ავტომატური ტრანზაქციები

11.2.3.1.1. ჩვეულებრივ, სერვერი:

- ა. არ მუშაობს ტრანზაქციის ავტომატურად დაწყების რეჟიმში
- ბ. მუშაობს ტრანზაქციის ავტომატურად დაწყების რეჟიმში
- გ. მუშაობს ტრანზაქციის ხელით დაწყების რეჟიმში

11.2.3.1.2. ავტომატურად დაწყების რეჟიმში თითოეული ბრძანება:

- ა. განიხილება, როგორც ცალკეული ტრანზაქცია
- ბ. არ განიხილება, როგორც ცალკეული ტრანზაქცია
- გ. განიხილება, როგორც მრავლობითი ტრანზაქცია

11.2.3.1.3. თუ საჭიროა ტრანზაქციის შექმნა, რომელიც რამდენიმე ბრძანებას შეიცავს, მაშინ ტრანზაქცია:

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

11.2.3.1.4. ტრანზაქციის ავტომატური განსაზღვრის რეჟიმის დასაყენებლად გამოიყენება ბრძანება:

- ა. SET IMPLICIT_TRANSACTION ON
- ბ. SET IMPLICIT_TRANSACTION OFF
- გ. SELECT

არააშკარა ტრანზაქციები

11.2.4.1.1. 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

11.2.4.1.2. მომხმარებელი არააშკარა ტრანზაქციის დასაწყებად:

- ა. არაფერს არ მიუთითებს
- ბ. სპეციალურ ბრძანებას მიუთითებს
- გ. SELECT ბრძანებას მიუთითებს

11.2.4.1.3. 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

11.2.4.1.4. ტრანზაქციის არააშკარა განსაზღვრის რეჟიმის დასაყენებლად გამოიყენება ბრძანება:

- ა. SELECT
- ბ. SET IMPLICIT_TRANSACTION OFF
- გ. SET IMPLICIT_TRANSACTION ON

განაწილებული ტრანზაქციები

11.2.5.1.1. როცა საჭიროა სხვადასხვა მონაცემთა ბაზაში მოთავსებულ რესურსებთან მიმართვა:

- ა. აუცილებელი არაა განაწილებული ტრანზაქციის გამოყენება
- ბ. აუცილებელია განაწილებული ტრანზაქციის გამოყენება
- გ. აუცილებელია არაგანაწილებული ტრანზაქციის გამოყენება

11.2.5.1.2. განაწილებული ტრანზაქცია შედგება რამდენიმე ცალკეული ტრანზაქციისაგან, რომლებიც:

- ა. ლოკალურად სრულდება მხოლოდ ერთ მონაცემთა ბაზაში
- ბ. გლობალურად სრულდება ცალკეულ მონაცემთა ბაზაში
- გ. ლოკალურად სრულდება ცალკეულ მონაცემთა ბაზაში

11.2.5.1.3. რესურსების მენეჯერი ეწოდება:

- ა. მონაცემების წყაროს, რომელსაც მომხმარებელი არ მიმართავს
- ბ. მონაცემების წყაროს, რომელსაც მომხმარებელი მიმართავს
- გ. მონაცემების წყაროს, რომელიც მომხმარებელს მიმართავს

11.2.5.1.4. რესურსების მენეჯერი ასრულებს:

- ა. ლოკალური ტრანზაქციების ფიქსირებას, უკუქცევასა და რესურსების სხვა მენეჯერებთან თავისი მოქმედებების კოორდინირებას
- ბ. გლობალური ტრანზაქციების ფიქსირებას, უკუქცევასა და რესურსების სხვა მენეჯერებთან თავისი მოქმედებების კოორდინირებას
- გ. ლოკალური ტრანზაქციების ფიქსირებას, უკუქცევას და არ ასრულებს რესურსების სხვა მენეჯერებთან თავისი მოქმედებების კოორდინირებას

11.2.5.1.5. ტრანზაქციების მენეჯერი ახდენს ლოკალური რესურსების მენეჯერების მუშაობის კოორდინირებას და:

- ა. არ იძლევა იმის გარანტიას, რომ ყველა ლოკალური ტრანზაქცია დაფიქსირებული ან უკუქცეული იქნება
- ბ. იძლევა იმის გარანტიას, რომ ყველა ლოკალური ტრანზაქცია დაფიქსირებული ან უკუქცეული იქნება
- გ. იძლევა იმის გარანტიას, რომ ზოგიერთი ლოკალური ტრანზაქცია დაფიქსირებული ან უკუქცეული იქნება

11.2.5.1.6. რომელი მსჯელობაა სწორი:

- ა. ზოგჯერ შეიძლება ტრანზაქციის ერთი ნაწილის ფიქსირება და მეორე ნაწილის უკუქცევა
- ბ. შეიძლება ტრანზაქციის ერთი ნაწილის ფიქსირება და მეორე ნაწილის უკუქცევა
- გ. არ შეიძლება ტრანზაქციის ერთი ნაწილის ფიქსირება და მეორე ნაწილის უკუქცევა

ჩადგმული ტრანზაქციები

11.2.6.1.1. 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

11.2.6.1.2. ჩადგმული ტრანზაქციების გამოყენება:

- ა. მისაწვდომია მხოლოდ აშკარა ტრანზაქციებში
- ბ. მისაწვდომი არაა მხოლოდ აშკარა ტრანზაქციებში
- გ. მისაწვდომია მხოლოდ არააშკარა ტრანზაქციებში

11.2.6.1.3. ავტომატური ან არააშკარა ტრანზაქციების შესრულებისას, ახალი ტრანზაქციის დაწყებამდე, წინა ტრანზაქცია:

- ა. დამთავრებული არ უნდა იყოს
- ბ. დამთავრებული უნდა იყოს
- გ. სანახევროდ უნდა იყოს დამთავრებული

11.2.6.1.4. ჩადგმული ტრანზაქციების ძირითადი დანიშნულებაა იმ ტრანზაქციების უზრუნველყოფა, რომლებიც:

- ა. სრულდება ჩადგმული ინდექსების მიერ
- ბ. არ სრულდება ჩადგმული პროცედურების მიერ
- გ. სრულდება ჩადგმული პროცედურების მიერ

11.2.6.1.5. 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

11.2.6.1.6. ზედა დონის ტრანზაქციის დამთავრება:

- ა. გადაიდება ჩადგმული ტრანზაქციის დამთავრებამდე
- ბ. არ გადაიდება ჩადგმული ტრანზაქციის დამთავრებამდე
- გ. გადაიდება ჩადგმული ტრანზაქციის დაწყებამდე

11.2.6.1.7. 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

11.2.6.1.8. თუ ქვედა დონეზე რამდენიმე ტრანზაქცია წარმატებით დამთავრდა (მაგრამ არ დაფიქსირდა), ხოლო საშუალო დონეზე წარუმატებლად, 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

11.2.6.1.9. 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

თავი 12. ტრიგერები

შესავალი

12.1.1.1.1. 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

12.1.1.1.2. თითოეული ტრიგერი დაკავშირებული:

- ა. არ არის კონკრეტულ ცხრილთან
- ბ. არის კონკრეტულ ცხრილთან
- გ. არის კონკრეტულ ინდექსთან

12.1.1.1.3. ცხრილში მონაცემების შეცვლის წინ სერვერი ავტომატურად გაუშვებს ტრიგერს, და თუ მისი შესრულება:

- ა. წარმატებით დამთავრდა, მაშინ არ შესრულდება ცხრილის მონაცემების ცვლილება
- ბ. წარუმატებლად დამთავრდა, მაშინ შესრულდება ცხრილის მონაცემების ცვლილება
- გ. წარმატებით დამთავრდა, მაშინ შესრულდება ცხრილის მონაცემების ცვლილება

12.1.1.1.4. ტრიგერის მიერ შესრულებული მოქმედებები:

- ა. არ განიხილება, როგორც ერთი ტრანზაქცია
- ბ. განიხილება, როგორც ერთი ტრანზაქცია
- გ. განიხილება, როგორც ორი ტრანზაქცია

12.1.1.1.5. 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

12.1.1.1.6. 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

12.1.1.1.7. 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

12.1.1.1.8. 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

12.1.1.1.9. 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

12.1.1.1.10. 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

12.1.1.1.11. ტექსტური ბლოკების დამუშავების ბრძანებების შესრულებისას, როგორცაა WRITETXT, READTXT და UPDATETXT ტრიგერი:

- ა. არ გაიშვება
- ბ. გაიშვება
- გ. აიკრძალება

12.1.1.1.12. რომელი მსჯელობაა სწორი:

- ა. ტრიგერის ქცევას არც ერთი პარამეტრი არ განსაზღვრავს:
- ბ. ტრიგერის ქცევას ერთი AFTER პარამეტრი განსაზღვრავს
- გ. ტრიგერის ქცევას ორი AFTER და INSTEAD OF პარამეტრები განსაზღვრავს

12.1.1.1.13. AFTER ტრიგერი სრულდება მისი გამომწვევი ბრძანებების წარმატებით შესრულების:

- ა. შუაში
- ბ. წინ
- გ. შემდეგ

12.1.1.1.14. თუ რაიმე მიზეზის გამო ბრძანებების შესრულება წარუმატებლად დამთავრდა, მაშინ ტრიგერი:

- ა. შესრულდება
- ბ. არ შესრულდება
- გ. ხანდახან შესრულდება

12.1.1.1.15. მონაცემების ცვლილება მოთხოვნის შესრულების შედეგად და ტრიგერის შესრულება ხდება:

- ა. ორი ტრანზაქციის ტანში
- ბ. ერთი ტრანზაქციის ტანში
- გ. სამი ტრანზაქციის ტანში

12.1.1.1.16. 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 გაუკეთებელი ცვლილებებიც

12.1.1.1.17. AFTER-ტრიგერები განისაზღვრება მხოლოდ:

- ა. ცხრილებისთვის და არ განისაზღვრება წარმოდგენებისთვის
- ბ. წარმოდგენებისთვის და არ განისაზღვრება ცხრილებისთვის
- გ. ცხრილებისთვის და განისაზღვრება წარმოდგენებისთვის

12.1.1.1.18. თითოეული INSERT, UPDATE და DELETE ბრძანებისათვის შეგვიძლია განვსაზღვროთ:

- ა. მხოლოდ ერთი ტრიგერი
- ბ. რამდენიმე AFTER-ტრიგერი
- გ. მხოლოდ ორი ტრიგერი

12.1.1.1.19. INSTEAD OF ტრიგერი გამოიძახება ბრძანების შესრულების:

- ა. უკან
- ბ. წინ
- გ. ნაცვლად

12.1.1.1.20. INSTEAD OF-ტრიგერები შეგვიძლია განვსაზღვროთ:

- ა. როგორც ცხრილებისთვის, ისე წარმოდგენებისთვის
- ბ. მხოლოდ ცხრილებისთვის
- გ. მხოლოდ წარმოდგენებისთვის

12.1.1.1.21. 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

12.1.1.1.22. ავტომატურად ყველა ტრიგერი:

- ა. AFTER-ტრიგერია
- ბ. DELETE-ტრიგერია
- გ. INSTEAD OF-ტრიგერია

12.1.1.1.23. ერთი ცხრილისთვის შეგვიძლია შევქმნათ:

- ა. მხოლოდ ერთი ერთტიპური ტრიგერი
- ბ. რამდენიმე ერთტიპური ტრიგერი
- გ. მხოლოდ ორი ერთტიპური ტრიგერი

ტრიგერის შექმნა

12.2.1.1.1. ტრიგერების შექმნა დროებითი ან სისტემური ცხრილებისათვის:

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

12.2.1.1.2. ტრიგერს:

- ა. არ შეუძლია მიმართოს დროებით ცხრილს
- ბ. ხანდახან შეუძლია მიმართოს დროებით ცხრილს
- გ. შეუძლია მიმართოს დროებით ცხრილს

12.2.1.1.3. თუ ტრიგერს სჭირდება სისტემურ ცხრილთან მიმართვა, მაშინ:

- ა. არ უნდა გამოვიყენოთ წარმოდგენები
- ბ. უნდა გამოვიყენოთ წარმოდგენები
- გ. უნდა გამოვიყენოთ ცხრილები

12.2.1.1.4. პაკეტში CREATE TRIGGER ბრძანება:

- ა. უნდა იყოს პირველი და არ უნდა გამოიყენებოდეს მხოლოდ ერთი ცხრილის მიმართ
- ბ. არ უნდა იყოს პირველი და უნდა გამოიყენებოდეს მხოლოდ ერთი ცხრილის მიმართ
- გ. უნდა იყოს პირველი და უნდა გამოიყენებოდეს მხოლოდ ერთი ცხრილის მიმართ

12.2.1.1.5. 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

12.2.1.1.6. 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

12.2.1.1.7. ტრიგერის შესაქმნელად გამოიყენება:

- ა. CREATE TRIGGER ბრძანება
- ბ. CREATE INDEX ბრძანება
- გ. CREATE DATABASE ბრძანება

12.2.1.1.8. ტრიგერის სახელი მონაცემთა ბაზის ფარგლებში:

- ა. ხანდახან უნდა იყოს უნიკალური
- ბ. არ უნდა იყოს უნიკალური
- გ. უნდა იყოს უნიკალური

12.2.1.1.9. თუ CREATE TRIGGER ბრძანებაში მითითებულია AFTER არგუმენტი, მაშინ:

- ა. ტრიგერი მხოლოდ იმ შემთხვევაში გაიშვება, როცა მისი გამომძახებელი ბრძანება წარმატებით შესრულდება
- ბ. ტრიგერი მხოლოდ იმ შემთხვევაში არ გაიშვება, როცა მისი გამომძახებელი ბრძანება წარმატებით შესრულდება
- გ. ტრიგერი მხოლოდ იმ შემთხვევაში გაიშვება, როცა მისი გამომძახებელი ბრძანება წარუმატებლად შესრულდება

12.2.1.1.10. თუ CREATE TRIGGER ბრძანებაში 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

12.2.1.1.11. 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

12.2.1.1.12. რომელი მსჯელობაა სწორი:

- ა. შეუძლებელია ტრიგერის დაკავშირება რამდენიმე ბრძანებასთან
- ბ. შესაძლებელია ტრიგერის დაკავშირება რამდენიმე ბრძანებასთან
- გ. შესაძლებელია ტრიგერის დაკავშირება მხოლოდ ერთ ბრძანებასთან

12.2.1.1.13. CREATE TRIGGER ბრძანების 'IF UPDATE (სვეტის_სახელი)' არგუმენტის გამოყენება საშუალებას:

- ა. გვაძლევს ტრიგერი შევასრულოთ ცხრილის კონკრეტული სტრიქონის მოდიფიცირების დროს
- ბ. არ გვაძლევს ტრიგერი შევასრულოთ ცხრილის კონკრეტული სვეტის მოდიფიცირების დროს
- გ. გვაძლევს ტრიგერი შევასრულოთ ცხრილის კონკრეტული სვეტის მოდიფიცირების დროს

12.2.1.1.14. 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

12.2.1.1.15. 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

12.2.1.1.16. 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

12.2.1.1.17. CREATE TRIGGER ბრძანების IF (COLUMNS_UPDATED()) არგუმენტი გამოიყენება მხოლოდ:

- ა. DELETE და UPDATE ბრძანებებისთვის
- ბ. UPDATE და INSERT ბრძანებებისათვის
- გ. INSERT და DELETE ბრძანებებისთვის

12.2.1.1.18. COLUMNS_UPDATED() ფუნქცია გასცემს ორობით რიცხვს, რომლის თითოეული:

- ა. ბაიტი შეესაბამება კონკრეტულ სვეტს
- ბ. ბიტი შეესაბამება კონკრეტულ სტრიქონს
- გ. ბიტი შეესაბამება კონკრეტულ სვეტს

12.2.1.1.19. 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

12.2.1.1.20. 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.

12.2.1.1.21. 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

12.2.1.1.22. 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

ტრიგერის შეცვლა

12.3.1.1.1. ტრიგერის შესაცვლელად გამოიყენება:

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

12.3.1.1.2. ALTER TRIGGER ბრძანების შესრულების დროს ჯერ:

- ა. იქმნება არსებული ტრიგერი, შემდეგ - იშლება ახალი
- ბ. იშლება არსებული ტრიგერი, შემდეგ - იქმნება ახალი
- გ. იშლება ახალი ტრიგერი, შემდეგ - იქმნება არსებული

ტრიგერის წაშლა

12.4.1.1.1. ტრიგერის წასაშლელად გამოიყენება:

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

12.4.1.1.2. რომელი მსჯელობაა სწორი:

- ა. შეუძლებელია რამდენიმე ტრიგერის ერთდროულად წაშლა

- ბ. შესაძლებელია რამდენიმე ტრიგერის ერთდროულად წაშლა
- გ. შესაძლებელია მხოლოდ ორი ტრიგერის ერთდროულად წაშლა

ტრიგერების დაპროგრამება

12.5.1.1.1. 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 დაუშვებელია სისტემური და მომხმარებლის მიერ განსაზღვრული შენახული პროცედურების გამოძახება

12.5.1.1.2. ტრიგერმა უმჯობესია:

- ა. ყოველთვის დააბრუნოს შედეგი
- ბ. ხშირად დააბრუნოს შედეგი
- გ. არ დააბრუნოს შედეგი

12.5.1.1.3. რომელი მსჯელობაა სწორი:

- ა. ტრიგერს ხანდახან შეუძლია შედეგის დაბრუნება
- ბ. ტრიგერს შეუძლია შედეგის დაბრუნება
- გ. ტრიგერს არ შეუძლია შედეგის დაბრუნება

12.5.1.1.4. 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

12.5.1.1.5. რომელი მსჯელობაა სწორი:

- ა. არ არის რეკომენდებული ტრიგერიდან დროებით ცხრილებზე მიმართვა, თუ არ არის დროებითი ცხრილების სტრუქტურის მუდმივობის გარანტია
- ბ. რეკომენდებულია ტრიგერიდან დროებით ცხრილებზე მიმართვა, თუ არ არის დროებითი ცხრილების სტრუქტურის მუდმივობის გარანტია
- გ. არ არის რეკომენდებული ტრიგერიდან დროებით ცხრილებზე მიმართვა, თუ არის დროებითი ცხრილების სტრუქტურის მუდმივობის გარანტია

12.5.1.1.6. ტრიგერის ტანში SET ბრძანებების გამოყენება:

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

12.5.1.1.7. SET ბრძანებებით დაყენებული მნიშვნელობები ძალაშია:

- ა. ტრიგერის მუშაობის დამთავრების შემდეგ
- ბ. მხოლოდ ტრიგერის მუშაობის პერიოდში
- გ. ტრიგერის მუშაობის დაწყებამდე

12.5.1.1.8. ტრიგერების ჩადგმულობის დონის განსაზღვრისათვის გამოიყენება ბრძანება:

- ა. SET NOCOUNT
- ბ. DROP TRIGGER
- გ. TRIGGER_NESTLEVEL ([ტრიგერის_იდენტიფიკატორი])

12.5.1.1.9. INSERT და UPDATE ბრძანებებით შეცვლილი სვეტების სიის მისაღებად გამოიყენება ფუნქცია:

- ა. COLUMN_UPDATE()
- ბ. TRIGGER_NESTLEVEL
- გ. SET NOCOUNT

12.5.1.1.10. სვეტის ცვლილების ფაქტის დასადასტურებლად გამოიყენება ფუნქცია:

- ა. UPDATE(სვეტის_სახელი)
- ბ. TRIGGER_NESTLEVEL
- გ. COLUMN_UPDATE()

12.5.1.1.11. სტრიქონების რაოდენობის შესახებ ინფორმაციის მისაღებად, რომლებიც შეიცვლება ტრიგერის წარმატებით დამთავრების შემთხვევაში, შეგვიძლია გამოვიყენოთ ფუნქცია:

- ა. TRIGGER_NESTLEVEL
- ბ. COLUMN_UPDATE()
- გ. @@ROWCOUNT

12.5.1.1.12. 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

12.5.1.1.13. შეცდომის შემთხვევაში ტრიგერის მუშაობის შესაწყვეტად და შესრულებული ცვლილებების გასაუქმებლად უნდა გამოვიყენოთ:

- ა. TRIGGER_NESTLEVEL ბრძანება
- ბ. COMMIT TRANSACTION ბრძანება
- გ. ROLLBACK TRANSACTION ბრძანება

12.5.1.1.14. ტრიგერის წარმატებით დამთავრების შემთხვევაში უნდა გამოვიყენოთ:

- ა. COMMIT TRANSACTION ბრძანება
- ბ. TRIGGER_NESTLEVEL ბრძანება
- გ. ROLLBACK TRANSACTION ბრძანება

12.5.1.1.15. როცა სერვერი იწყებს ტრიგერის შესრულებას, ის ქმნის ორ სპეციალურ ცხრილს:

- ა. inserted და updated
- ბ. updated და deleted
- გ. inserted და deleted

12.5.1.1.16. 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

12.5.1.1.17. 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

12.5.1.1.18. 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

12.5.1.1.19. თუ ტრიგერი გამოიწვია INSERT ბრძანებამ, მაშინ deleted ცხრილი:

- ა. არც ერთ სვეტს არ შეიცავს
- ბ. ყველა სვეტს არ შეიცავს
- გ. ყველა სტრიქონს შეიცავს

12.5.1.1.20. 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

12.5.1.1.21. 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

12.5.1.1.22. 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

12.5.1.1.23. 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

12.5.1.1.24. 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

12.5.1.1.25. 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

ტრიგერების მართვა

ტრიგერის სახელის შეცვლა

12.6.1.1.1. ტრიგერის სახელის შესაცვლელად გამოიყენება:

- ა. sp_rename სისტემური შენახული პროცედურა
- ბ. sp_help სისტემური შენახული პროცედურა
- გ. sp_delete სისტემური შენახული პროცედურა

ტრიგერების შექმნის დროს არსებული შეზღუდვები

12.7.1.1.1. ტრიგერის შიგნით დაუშვებელია შემდეგი ოპერაციების შესრულება:

- ა. ცხრილის შექმნა, შეცვლა და წაშლა; RECONFIGURE, DISK RESIZE და DISK INIT ბრძანებების შესრულება
- ბ. მონაცემთა ბაზის ან ტრანზაქციების ჟურნალის სარეზერვო ასლის აღდგენა; სტრიქონის წაშლა
- გ. მონაცემთა ბაზის შექმნა, შეცვლა და წაშლა; მონაცემთა ბაზის ან ტრანზაქციების ჟურნალის სარეზერვო ასლის აღდგენა; RECONFIGURE, DISK RESIZE და DISK INIT ბრძანებების შესრულება

12.7.1.1.2. ტრიგერის შიგნით დაუშვებელია შემდეგი ბრძანებების შესრულება: მონაცემთა ბაზის შექმნა, შეცვლა და წაშლა; მონაცემთა ბაზის ან ტრანზაქციების ჟურნალის სარეზერვო ასლის აღდგენა; RECONFIGURE, DISK RESIZE და DISK INIT, რადგან:

- ა. შეუძლებელია მათი გაუქმება იმ ტრანზაქციის უკუქცევის დროს, რომელშიც ტრიგერი არ სრულდება
- ბ. შეუძლებელია მათი გაუქმება იმ ტრანზაქციის უკუქცევის დროს, რომელშიც ტრიგერი სრულდება
- გ. შესაძლებელია მათი გაუქმება იმ ტრანზაქციის უკუქცევის დროს, რომელშიც ტრიგერი სრულდება

ტრიგერის შესახებ ინფორმაციის მიღება

12.7.2.1.1. 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

12.7.2.1.2. 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

12.7.2.1.3. იმ ობიექტების სანახავად, რომლებზეც ტრიგერია დამოკიდებული, უნდა შევასრულოთ:

- ა. sp_depends სისტემური შენახული პროცედურა
- ბ. sp_helptrigger სისტემური შენახული პროცედურა
- გ. sp_helptext სისტემური შენახული პროცედურა

12.7.2.1.4. სერვერის დონეზე ჩადგმული ტრიგერების შექმნის უფლების მისაცემად ან ასაკრძალვად უნდა გამოვიყენოთ:

- ა. sp_helptrigger სისტემური შენახული პროცედურა
- ბ. sp_configure სისტემური შენახული პროცედურა
- გ. sp_depends სისტემური შენახული პროცედურა

12.7.2.1.5. მონაცემთა ბაზის დონეზე ჩადგმული ტრიგერების შექმნის უფლების მისაცემად ან ასაკრძალვად უნდა გამოვიყენოთ:

- ა. sp_configure სისტემური შენახული პროცედურა
- ბ. sp_depends სისტემური შენახული პროცედურა
- გ. sp_dboption სისტემური შენახული პროცედურა

დანართი. ტესტების პასუხები

თავი 1. მონაცემთა ბაზების თეორიული საფუძვლები	1.3.1.1.21.	ბ	
საინფორმაციო სისტემები	1.3.1.1.22.	ა	
1.1.1.1.1.	ა	მონაცემთა რელაციური ბაზები	
1.1.1.1.2.	ბ	1.4.1.1.1.	ა
1.1.1.1.3.	გ	1.4.1.1.2.	ბ
1.1.1.1.4.	ბ	1.4.1.1.3.	ბ
მონაცემთა ბაზები		1.4.1.1.4.	ბ
1.2.1.1.1.	ა	1.4.1.1.5.	ბ
1.2.1.1.2.	ბ		
1.2.1.1.3.	გ	ცხრილების ნორმალიზება	
1.2.1.1.4.	ა	1.5.1.1.1.	ბ
1.2.1.1.5.	გ	1.5.1.1.2.	ა
1.2.1.1.6.	ა	1.5.1.1.3.	ბ
1.2.1.1.7.	ბ	1.5.1.1.4.	ა
1.2.1.1.8.	ბ	1.5.1.1.5.	ბ
1.2.1.1.9.	ა	1.5.1.1.6.	ა
1.2.1.1.10.	ბ	1.5.1.1.7.	ბ
1.2.1.1.11.	გ	1.5.1.1.8.	გ
1.2.1.1.12.	ბ	1.5.1.1.9.	ბ
1.2.1.1.13.	ა	1.5.1.1.10.	ა
1.2.1.1.14.	ბ	1.5.1.1.11.	ბ
1.2.1.1.15.	გ		
მონაცემთა ბაზების მოდელები		ცხრილებს შორის არსებული კავშირები პირველადი და გარე გასაღებები	
1.3.1.1.1.	ა	1.6.1.1.1.	ბ
1.3.1.1.2.	გ	1.6.1.1.2.	ა
1.3.1.1.3.	ბ	1.6.1.1.3.	ბ
1.3.1.1.4.	ბ	1.6.1.1.4.	ა
1.3.1.1.5.	ა	1.6.1.1.5.	გ
1.3.1.1.6.	ა	1.6.1.1.6.	ბ
1.3.1.1.7.	გ	1.6.1.1.7.	ა
1.3.1.1.8.	ბ	1.6.1.1.8.	გ
1.3.1.1.9.	გ	1.6.1.1.9.	ა
1.3.1.1.10.	ა	1.6.1.1.10.	ბ
1.3.1.1.11.	ბ	1.6.1.1.11.	გ
1.3.1.1.12.	გ	1.6.1.1.12.	ბ
1.3.1.1.13.	ა	1.6.1.1.13.	გ
1.3.1.1.14.	გ	1.6.1.1.14.	ა
1.3.1.1.15.	ბ	1.6.1.1.15.	ბ
1.3.1.1.16.	გ	1.6.1.1.16.	ა
1.3.1.1.17.	ა	1.6.1.1.17.	გ
1.3.1.1.18.	ა	1.6.1.1.18.	გ
1.3.1.1.19.	ბ	1.6.1.1.19.	ბ
1.3.1.1.20.	გ	1.6.1.1.20.	ბ

1.6.1.1.21.	ა	მომხმარებლების ტიპები
1.6.1.1.22.	ბ	2.1.3.1.1.
		2.1.3.1.2.
"ერთი-ერთთან" კავშირი		2.1.3.1.3.
1.6.2.1.1.	ბ	
1.6.2.1.2.	ა	წარმოდგენები
1.6.2.1.3.	ბ	2.1.4.1.1.
		2.1.4.1.2.
"ერთი-ბევრთან" კავშირი		2.1.4.1.3.
1.6.3.1.1.	ა	2.1.4.1.4.
		2.1.4.1.5.
"ბევრი-ბევრთან" კავშირი		
1.6.4.1.1.	ბ	შენახული პროცედურები
		2.1.5.1.1.
თავი 2. მონაცემთა ბაზები		2.1.5.1.2.
მონაცემთა ბაზების არქიტექტურა		2.1.5.1.3.
2.1.0.1.1.	ბ	2.1.5.1.4.
2.1.0.1.2.	ბ	2.1.5.1.5.
2.1.0.1.3.	ა	
		ტრიგერები
მონაცემთა ბაზების ობიექტები		2.1.6.1.1.
2.1.1.1.1.	ბ	2.1.6.1.2.
		2.1.6.1.3.
ცხრილები		2.1.6.1.4.
2.1.2.1.1.	ა	
2.1.2.1.2.	ა	ინდექსები
2.1.2.1.3.	ბ	2.1.7.1.1.
2.1.2.1.4.	ბ	2.1.7.1.2.
2.1.2.1.5.	ბ	2.1.7.1.3.
2.1.2.1.6.	ბ	
2.1.2.1.7.	ა	მთლიანობის შეზღუდვები
2.1.2.1.8.	ბ	2.1.8.1.1.
2.1.2.1.9.	ბ	2.1.8.1.2.
2.1.2.1.10.	ა	2.1.8.1.3.
2.1.2.1.11.	ბ	2.1.8.1.4.
2.1.2.1.12.	ა	2.1.8.1.5.
2.1.2.1.13.	ბ	2.1.8.1.6.
2.1.2.1.14.	ბ	2.1.8.1.7.
2.1.2.1.15.	ა	2.1.8.1.8.
2.1.2.1.16.	ბ	2.1.8.1.9.
2.1.2.1.17.	ბ	2.1.8.1.10.
2.1.2.1.18.	ა	2.1.8.1.11.
2.1.2.1.19.	ბ	2.1.8.1.12.
2.1.2.1.20.	ა	2.1.8.1.13.
2.1.2.1.21.	ბ	2.1.8.1.14.
		2.1.8.1.15.
		2.1.8.1.16.

2.1.8.1.17.	ბ	ფაილები და ფაილების ჯგუფები
2.1.8.1.18.	ბ	2.3.0.1.1.
2.1.8.1.19.	ბ	2.3.0.1.2.
2.1.8.1.20.	ბ	2.3.0.1.3.
2.1.8.1.21.	ა	2.3.0.1.4.
2.1.8.1.22.	ა	2.3.0.1.5.
2.1.8.1.23.	ბ	2.3.0.1.6.
2.1.8.1.24.	ბ	2.3.0.1.7.
2.1.8.1.25.	ა	2.3.0.1.8.
2.1.8.1.26.	ა	2.3.0.1.9.
2.1.8.1.27.	ბ	2.3.0.1.10.
2.1.8.1.28.	ბ	2.3.0.1.11.
2.1.8.1.29.	ა	2.3.0.1.12.
2.1.8.1.30.	ბ	2.3.0.1.13.

ნაგულისხმევი მნიშვნელობები

2.1.9.1.1.	ბ	2.3.0.1.14.	ა
2.1.9.1.2.	ა	2.3.0.1.15.	ბ
		2.3.0.1.16.	ა
		2.3.0.1.17.	ბ

ფუნქციები

2.1.10.1.1.1	ბ
2.1.10.1.2.	ბ
2.1.10.1.3.	ა

სისტემური მონაცემთა ბაზები

2.2.0.1.1.	ა
2.2.0.1.2.	ბ
2.2.0.1.3.	ბ

Master სისტემური მონაცემთა ბაზა

2.2.1.1.1.	ბ
2.2.1.1.2.	ა
2.2.1.1.3.	ბ

Tempdb სისტემური მონაცემთა ბაზა

2.2.2.1.1.	ბ
2.2.2.1.2.	ა
2.2.2.1.3.	ბ

Model სისტემური მონაცემთა ბაზა

2.2.3.1.1.	ა
------------	---

MsdB სისტემური მონაცემთა ბაზა

2.2.4.1.1.	ა
2.2.4.1.2.	ბ
2.2.4.1.3.	ბ

მონაცემთა ბაზის ფიზიკური სტრუქტურა

2.3.1.1.1.	ბ
2.3.1.1.2.	ბ
2.3.1.1.3.	ა
2.3.1.1.4.	ბ
2.3.1.1.5.	ბ
2.3.1.1.6.	ა
2.3.1.1.7.	ბ
2.3.1.1.8.	ბ
2.3.1.1.9.	ა
2.3.1.1.10.	ა
2.3.1.1.11.	ბ
2.3.1.1.12.	ბ
2.3.1.1.13.	ბ
2.3.1.1.14.	ბ
2.3.1.1.15.	ა
2.3.1.1.16.	ბ
2.3.1.1.17.	ბ
2.3.1.1.18.	ა
2.3.1.1.19.	ბ
2.3.1.1.20.	ბ
2.3.1.1.21.	ა

მონაცემთა ბაზის შექმნა, წაშლა და სახელის შეცვლა

2.4.0.1.1.	ბ
2.4.0.1.2.	ა
2.4.0.1.3.	ბ

მონაცემთა ბაზის შექმნა

2.4.1.1.1.	ა
2.4.1.1.2.	ბ
2.4.1.1.3.	გ
2.4.1.1.4.	გ
2.4.1.1.5.	ბ
2.4.1.1.6.	გ
2.4.1.1.7.	ა
2.4.1.1.8.	ბ
2.4.1.1.9.	გ
2.4.1.1.10.	ბ
2.4.1.1.11.	გ
2.4.1.1.12.	ა
2.4.1.1.13.	ა
2.4.1.1.14.	ბ
2.4.1.1.15.	გ
2.4.1.1.16.	ბ
2.4.1.1.17.	ა
2.4.1.1.18.	გ
2.4.1.1.19.	ბ
2.4.1.1.20.	ა
2.4.1.1.21.	ა
2.4.1.1.22.	გ
2.4.1.1.23.	ა
2.4.1.1.24.	ბ
2.4.1.1.25.	გ
2.4.1.1.26.	ბ
2.4.1.1.27.	ა
2.4.1.1.28.	ბ
2.4.1.1.29.	გ
2.4.1.1.30.	ა
2.4.1.1.31.	ბ
2.4.1.1.32.	გ
2.4.1.1.33.	ა
2.4.1.1.34.	ბ
2.4.1.1.35.	გ

მონაცემთა ბაზის წაშლა

2.4.2.1.1.	ბ
2.4.2.1.2.	ა

მონაცემთა ბაზის სახელის შეცვლა

2.4.3.1.1.	ა
2.4.2.1.2.	ა

მონაცემთა ბაზის მიერთება და გამორთვა

2.5.1.1.1.	ბ
2.5.1.1.2.	გ
2.5.1.1.3.	ბ
2.5.1.1.4.	გ
2.5.1.1.5.	ა
2.5.1.1.6.	ბ
2.5.1.1.7.	გ
2.5.1.1.8.	ა
2.5.1.1.9.	ბ
2.5.1.1.10.	ა
2.5.1.1.11.	ბ
2.5.1.1.12.	გ
2.5.1.1.13.	ბ
2.5.1.1.14.	ა
2.5.1.1.15.	გ

**მონაცემთა ბაზების მართვა
ფლობის უფლების გადაცემა**

2.6.1.1.1.	გ
2.6.1.1.2.	ა
2.6.1.1.3.	ბ
2.6.1.1.4.	ა
2.6.1.1.5.	ბ
2.6.1.1.6.	გ
2.6.1.1.7.	ა

მონაცემთა ბაზის შეკუმშვა

2.6.2.1.1.	ა
2.6.2.1.2.	ბ
2.6.2.1.3.	ა
2.6.2.1.4.	გ
2.6.2.1.5.	ბ
2.6.2.1.6.	ა
2.6.2.1.7.	ბ
2.6.2.1.8.	ა
2.6.2.1.9.	ა
2.6.2.1.10.	გ
2.6.2.1.11.	გ
2.6.2.1.12.	ბ
2.6.2.1.13.	ა
2.6.2.1.14.	გ
2.6.2.1.15.	ბ
2.6.2.1.16.	გ
2.6.2.1.17.	ბ
2.6.2.1.18.	ა
2.6.2.1.19.	გ

2.6.2.1.20. ბ
 2.6.2.1.21. ა
 2.6.2.1.22. ბ
 2.6.2.1.23. გ
 2.6.2.1.24. ა
 2.6.2.1.25. გ
 2.6.2.1.26. ბ
 2.6.2.1.27. ბ
 2.6.2.1.28. ა
 2.6.2.1.29. გ
 2.6.2.1.30. ა
 2.6.2.1.31. ბ
 2.6.2.1.32. ბ
 2.6.2.1.33. ბ
 2.6.2.1.34. ა
 2.6.2.1.35. გ

მონაცემთა ბაზის თვისებების მართვა

2.6.3.1.1. გ
 2.6.3.1.2. ა
 2.6.3.1.3. ბ
 2.6.3.1.4. ა
 2.6.3.1.5. გ
 2.6.3.1.6. ბ
 2.6.3.1.7. ა
 2.6.3.1.8. გ
 2.6.3.1.9. ბ
 2.6.3.1.10. გ
 2.6.3.1.11. ა
 2.6.3.1.12. ა
 2.6.3.1.13. ბ
 2.6.3.1.14. გ
 2.6.3.1.15. ა
 2.6.3.1.16. ა
 2.6.3.1.17. გ
 2.6.3.1.18. ბ
 2.6.3.1.19. ბ
 2.6.3.1.20. ბ
 2.6.3.1.21. ბ
 2.6.3.1.22. ბ
 2.6.3.1.23. ბ
 2.6.3.1.24. ბ
 2.6.3.1.25. ბ
 2.6.3.1.26. ბ
 2.6.3.1.27. ბ
 2.6.3.1.28. ბ
 2.6.3.1.29. გ

2.6.3.1.30. ა
 2.6.3.1.31. ა
 2.6.3.1.32. ბ
 2.6.3.1.33. ბ
 2.6.3.1.34. გ
 2.6.3.1.35. გ
 2.6.3.1.36. ა
 2.6.3.1.37. ბ
 2.6.3.1.38. ბ
 2.6.3.1.39. ბ
 2.6.3.1.40. ბ
 2.6.3.1.41. ბ
 2.6.3.1.42. ბ

მონაცემთა ბაზის საკონფიგურაციო პარამეტრებთან მუშაობის საშუალებები

2.6.4.1.1. ა
 2.6.4.1.2. ბ
 2.6.4.1.3. გ
 2.6.4.1.4. ბ
 2.6.4.1.5. გ
 2.6.4.1.6. გ
 2.6.4.1.7. ა
 2.6.4.1.8. გ
 2.6.4.1.9. ბ
 2.6.4.1.10. ა
 2.6.4.1.11. ბ
 2.6.4.1.12. გ
 2.6.4.1.13. ბ
 2.6.4.1.14. გ
 2.6.4.1.15. ა
 2.6.4.1.16. ა
 2.6.4.1.17. ბ
 2.6.4.1.18. გ
 2.6.4.1.19. ბ
 2.6.4.1.20. ა
 2.6.4.1.21. ა

მონაცემთა მომხმარებლის ტიპის შექმნა

2.7.1.1.1. გ
 2.7.1.1.2. გ
 2.7.1.1.3. ბ
 2.7.1.1.4. ა
 2.7.1.1.5. ა
 2.7.1.1.6. ბ
 2.7.1.1.7. გ

- 2.7.1.1.8. ბ
- 2.7.1.1.9. გ
- 2.7.1.1.10. ა
- 2.7.1.1.11. ა
- 2.7.1.1.12. ბ
- 2.7.1.1.13. გ
- 2.7.1.1.14. გ
- 2.7.1.1.15. ბ

თავი 3. ცხრილები
ცხრილების დაპროექტება
ცხრილის პირველადი გასაღები

- 3.1.1.1.1. ა
- 3.1.1.1.2. ბ
- 3.1.1.1.3. ა
- 3.1.1.1.4. ბ
- 3.1.1.1.5. გ
- 3.1.1.1.6. ბ
- 3.1.1.1.7. გ
- 3.1.1.1.8. ა
- 3.1.1.1.9. ა
- 3.1.1.1.10. ბ

ცხრილის გარე გასაღები

- 3.1.2.1.1. გ
- 3.1.2.1.2. ბ
- 3.1.2.1.3. ბ
- 3.1.2.1.4. ა
- 3.1.2.1.5. გ
- 3.1.2.1.6. ბ
- 3.1.2.1.7. გ

სვეტის უნიკალურობის განსაზღვრა

- 3.1.3.1.1. ა
- 3.1.3.1.2. ბ
- 3.1.3.1.3. გ
- 3.1.3.1.4. ა
- 3.1.3.1.5. გ

შემმოწმებელი შეზღუდვების განსაზღვრა

- 3.1.4.1.1. ბ
- 3.1.4.1.2. ა
- 3.1.4.1.3. გ
- 3.1.4.1.4. ბ
- 3.1.4.1.5. ბ
- 3.1.4.1.6. ა
- 3.1.4.1.7. გ

ნაგულისხმევი მნიშვნელობების განსაზღვრა

- 3.1.5.1.1. ა
- 3.1.5.1.2. გ
- 3.1.5.1.3. ა
- 3.1.5.1.4. ბ

სვეტ-მოვლელის განსაზღვრა

- 3.1.6.1.1. ბ
- 3.1.6.1.2. გ
- 3.1.6.1.3. ა
- 3.1.6.1.4. ა
- 3.1.6.1.5. ბ
- 3.1.6.1.6. გ
- 3.1.6.1.7. ა
- 3.1.6.1.8. ბ
- 3.1.6.1.9. გ
- 3.1.6.1.10. ბ
- 3.1.6.1.11. ა
- 3.1.6.1.12. ა
- 3.1.6.1.13. გ
- 3.1.6.1.14. ბ
- 3.1.6.1.15. ბ
- 3.1.6.1.16. ა
- 3.1.6.1.17. გ
- 3.1.6.1.18. ბ
- 3.1.6.1.19. ა
- 3.1.6.1.20. ბ
- 3.1.6.1.21. გ
- 3.1.6.1.22. ა
- 3.1.6.1.23. ბ
- 3.1.6.1.24. გ
- 3.1.6.1.25. ბ
- 3.1.6.1.26. ბ
- 3.1.6.1.27. ა

ცხრილის შექმნა

- 3.2.1.1.1. ა
- 3.2.1.1.2. გ
- 3.2.1.1.3. ბ
- 3.2.1.1.4. ა
- 3.2.1.1.5. ბ
- 3.2.1.1.6. ა
- 3.2.1.1.7. გ
- 3.2.1.1.8. გ
- 3.2.1.1.9. ა
- 3.2.1.1.10. ბ

3.2.1.1. 11. ბ
 3.2.1.1. 12. ბ
 3.2.1.1. 13. ა
 3.2.1.1.14. ა
 3.2.1.1.15. ბ
 3.2.1.1.16. გ
 3.2.1.1.17. ბ
 3.2.1.1.18. ა
 3.2.1.1.19. ბ
 3.2.1.1.20. გ
 3.2.1.1.21. ბ
 3.2.1.1.22. ა
 3.2.1.1.23. გ
 3.2.1.1.24. ბ
 3.2.1.1.25. ა
 3.2.1.1.26. გ
 3.2.1.1.27. ა
 3.2.1.1.28. გ
 3.2.1.1.29. ბ
 3.2.1.1.30. ბ
 3.2.1.1.31. ა
 3.2.1.1.32. გ
 3.2.1.1.33. ბ
 3.2.1.1.34. ა
 3.2.1.1.35. ა
 3.2.1.1.36. გ
 3.2.1.1.37. ბ
 3.2.1.1.38. გ
 3.2.1.1.39. ბ
 3.2.1.1.40. ბ
 3.2.1.1.41. ა
 3.2.1.1.42. ბ
 3.2.1.1.43. ბ
 3.2.1.1.44. გ
 3.2.1.1.45. გ
 3.2.1.1.46. ა
 3.2.1.1.47. ბ
 3.2.1.1.48. გ
 3.2.1.1.49. ბ
 3.2.1.1.50. ა
 3.2.1.1.51. გ
 3.2.1.1.52. ბ
 3.2.1.1.53. გ
 3.2.1.1.54. ა
 3.2.1.1.55. ბ
 3.2.1.1.56. გ
 3.2.1.1.57. ა

3.2.1.1.58. გ
 3.2.1.1.59. ბ
 3.2.1.1.60. ა
 3.2.1.1.61. გ
 3.2.1.1.62. გ
 3.2.1.1.63. ბ

ცხრილის წაშლა

3.3.1.1. 1. ა
 3.3.1.1.2. ბ
 3.3.1.1.3. გ

ცხრილის შეცვლა

3.4.1.1. 1. ბ
 3.4.1.1.2. გ
 3.4.1.1. 3. ა
 3.4.1.1.4. გ
 3.4.1.1. 5. ბ
 3.4.1.1.6. ა

ცხრილების მართვა

3.5.1.1. 1. ა
 3.5.1.1.2. ბ
 3.5.1.1. 3. გ
 3.5.1.1.4. ბ

ინფორმაციის მიღება ცხრილების შესახებ

3.5.2.1. 1. ა
 3.5.2.1.2. ბ
 3.5.2.1. 3. გ
 3.5.2.1.4. ა
 3.5.2.1.5. ბ

ინფორმაციის მიღება ცხრილზე დამოკიდებული ობიექტების შესახებ

3.5.3.1. 1. გ
 3.5.3.1. 2. ბ
 3.5.3.1.3. ა

ცხრილებს შორის კავშირების შესახებ ინფორმაციის მიღება

3.5.4.1. 1. ბ
 3.5.4.1. 2. გ
 3.5.4.1. 3. ა
 3.5.4.1. 4. ბ
 3.5.4.1.5. ა

დიაგრამები

- 3.6.1.1. 1. ბ
- 3.6.1.1. 1. გ

თავი 4. მონაცემების მართვა
ცხრილში მონაცემების ჩამატება
INSERT ბრძანება

- 4.1.1.1. 1. ა
- 4.1.1.1. 2. ბ
- 4.1.1.1. 3. გ
- 4.1.1.1. 4. ბ
- 4.1.1.1. 5. ა
- 4.1.1.1. 6. გ
- 4.1.1.1. 7. ბ
- 4.1.1.1. 8. ბ
- 4.1.1.1. 9. ა
- 4.1.1.1. 10. გ
- 4.1.1.1. 11. გ
- 4.1.1.1. 12. გ
- 4.1.1.1. 13. გ
- 4.1.1.1. 14. გ
- 4.1.1.1. 15. გ
- 4.1.1.1. 16. გ
- 4.1.1.1. 17. გ
- 4.1.1.1. 18. გ

SELECT...INTO ბრძანება

- 4.1.2.1. 1. ა
- 4.1.2.1. 2. გ
- 4.1.2.1. 3. ბ
- 4.1.2.1. 4. ა
- 4.1.2.1. 5. ბ
- 4.1.2.1. 6. გ
- 4.1.2.1. 7. ა
- 4.1.2.1. 8. ბ

ცხრილიდან მონაცემების ამორჩევა
SELECRT ბრძანება

SELECT განყოფილება

- 4.2.1.1.1. ა
- 4.2.1.1.2. ბ
- 4.2.1.1.3. გ
- 4.2.1.1.4. ა
- 4.2.1.1.5. ბ
- 4.2.1.1.6. ბ
- 4.2.1.1.7. ა
- 4.2.1.1.8. გ

- 4.2.1.1.9. გ
- 4.2.1.1.10. გ
- 4.2.1.1.11. გ
- 4.2.1.1.12. ა
- 4.2.1.1.13. ბ
- 4.2.1.1.14. ბ
- 4.2.1.1.15. გ
- 4.2.1.1.16. ა
- 4.2.1.1.17. ბ
- 4.2.1.1.18. გ

FROM განყოფილება

- 4.2.1.2.1. ა
- 4.2.1.2.2. გ
- 4.2.1.2.3. ბ
- 4.2.1.2.4. გ
- 4.2.1.2.5. ა
- 4.2.1.2.6. ბ
- 4.2.1.2.7. გ
- 4.2.1.2.8. ა
- 4.2.1.2.9. ბ
- 4.2.1.2.10. ა
- 4.2.1.2.11. გ
- 4.2.1.2.12. ა
- 4.2.1.2.13. ბ
- 4.2.1.2.14. გ

WHERE განყოფილება

- 4.2.1.3.1. ა
- 4.2.1.3.2. ბ
- 4.2.1.3.3. გ
- 4.2.1.3.4. ა
- 4.2.1.3.5. ბ
- 4.2.1.3.6. ა

ჩადგმული მოთხოვნები

- 4.2.1.4.1. ა
- 4.2.1.4.2. ბ
- 4.2.1.4.3. გ
- 4.2.1.4.4. ბ
- 4.2.1.4.5. ა
- 4.2.1.4.6. გ
- 4.2.1.4.7. ა

ბმული მოთხოვნები

- 4.2.1.5.1. ა
- 4.2.1.5.2. ბ

GROUP BY განყოფილება

- 4.2.1.6.1. ა
- 4.2.1.6.2. ბ
- 4.2.1.6.3. ბ
- 4.2.1.6.4. ა
- 4.2.1.6.5. ბ

HAVING განყოფილება

- 4.2.1.7.1. ბ
- 4.2.1.7.2. ა
- 4.2.1.7.3. ბ

UNION განყოფილება

- 4.2.1.8.1. ბ
- 4.2.1.8.2. ა
- 4.2.1.8.3. ბ
- 4.2.1.8.4. ბ
- 4.2.1.8.5. ა

ORDER BY განყოფილება

- 4.2.1.9.1. ბ
- 4.2.1.9.2. ა
- 4.2.1.9.3. ბ
- 4.2.1.9.4. ბ
- 4.2.1.9.5. ა

COMPUTE განყოფილება

- 4.2.1.10.1. ა
- 4.2.1.10.2. ბ
- 4.2.1.10.3. ბ
- 4.2.1.10.4. ა
- 4.2.1.10.5. ბ
- 4.2.1.10.6. ა

აგრეგირების ფუნქციები

- 4.2.1.11.1. ა
- 4.2.1.11.2. ბ
- 4.2.1.11.3. ბ
- 4.2.1.11.4. ბ
- 4.2.1.11.5. ბ
- 4.2.1.11.6. ა
- 4.2.1.11.7. ბ
- 4.2.1.11.8. ა
- 4.2.1.11.9. ბ
- 4.2.1.11.10. ბ
- 4.2.1.11.11. ბ

4.2.1.11.12. ა

ცხრილის მონაცემების ცვლილება

UPDATE ბრძანება

- 4.3.1.1.1. ა
- 4.3.1.1.2. ბ
- 4.3.1.1.3. ბ
- 4.3.1.1.4. ა
- 4.3.1.1.5. ბ
- 4.3.1.1.6. ბ
- 4.3.1.1.7. ბ
- 4.3.1.1.8. ა
- 4.3.1.1.9. ა
- 4.3.1.1.10. ბ
- 4.3.1.1.11. ბ
- 4.3.1.1.12. ბ

ცხრილიდან მონაცემების წაშლა

DELETE ბრძანება

- 4.4.1.1.1. ა
- 4.4.1.1.2. ბ

მონაცემების მასობრივი გადაწერა

- 4.5.1.1.1. ა
- 4.5.1.1.2. ბ
- 4.5.1.1.3. ბ
- 4.5.1.1.4. ბ
- 4.5.1.1.5. ა

თავი 5. Transact SQL-ის საფუძვლები

შესავალი

- 5.1.1.1.1. ა
- 5.1.1.1.2. ბ
- 5.1.1.1.3. ბ
- 5.1.1.1.4. ბ
- 5.1.1.1.5. ა
- 5.1.1.1.6. ბ
- 5.1.1.1.7. ბ
- 5.1.1.1.8. ბ

გამოსახულებები

- 5.2.1.1.1. ბ
- 5.2.1.1.2. ბ
- 5.2.1.1.3. ა
- 5.2.1.1.4. ბ
- 5.2.1.1.5. ა
- 5.2.1.1.6. ბ

5.2.1.1.7. ბ
 5.2.1.1.8. ა
 5.2.1.1.9. ბ

ოპერატორები

5.3.1.1.1. ა
 5.3.1.1.2. ა
 5.3.1.1.3. ბ
 5.3.1.1.4. ბ
 5.3.1.1.5. ბ
 5.3.1.1.6. ა
 5.3.1.1.7. ა
 5.3.1.1.8. ბ
 5.3.1.1.9. ბ
 5.3.1.1.10. ბ
 5.3.1.1.11. ა
 5.3.1.1.12. ბ
 5.3.1.1.13. ბ
 5.3.1.1.14. ა
 5.3.1.1.15. ბ
 5.3.1.1.16. ბ
 5.3.1.1.17. ა
 5.3.1.1.18. ბ
 5.3.1.1.19. ა
 5.3.1.1.20. ბ

იდენტიფიკატორები

5.4.1.1.1. ა
 5.4.1.1.2. ბ
 5.4.1.1.3. ბ
 5.4.1.1.4. ა
 5.4.1.1.5. ბ
 5.4.1.1.6. ბ
 5.4.1.1.7. ბ
 5.4.1.1.8. ბ
 5.4.1.1.9. ა
 5.4.1.1.10. ა
 5.4.1.1.11. ბ
 5.4.1.1.12. ბ
 5.4.1.1.13. ა
 5.4.1.1.14. ა
 5.4.1.1.15. ა
 5.4.1.1.16. ბ
 5.4.1.1.17. ბ
 5.4.1.1.18. ბ
 5.4.1.1.19. ბ
 5.4.1.1.20. ა

5.4.1.1.21. ბ
 5.4.1.1.22. ბ
 5.4.1.1.23. ბ
 5.4.1.1.24. ა
 5.4.1.1.25. ბ
 5.4.1.1.26. ბ
 5.4.1.1.27. ა
 5.4.1.1.28. ბ

ცვლადები

5.5.1.1.1. ბ
 5.5.1.1.2. ბ
 5.5.1.1.3. ბ
 5.5.1.1.4. ბ
 5.5.1.1.5. ა
 5.5.1.1.6. ბ
 5.5.1.1.7. ბ
 5.5.1.1.8. ბ
 5.5.1.1.9. ბ

მმართველი კონსტრუქციები

BEGIN...END

5.6.1.1.1. ა
 5.6.1.1.2. ბ
 5.6.1.1.3. ბ
 5.6.1.1.4. ბ
 5.6.1.1.5. ა

IF...ELSE

5.6.2.1.1. ბ
 5.6.2.1.2. ა
 5.6.2.1.3. ბ
 5.6.2.1.4. ბ

CASEEND

5.6.3.1.1. ბ
 5.6.3.1.2. ა
 5.6.3.1.3. ბ
 5.6.3.1.4. ა
 5.6.3.1.5. ბ

COALESCE

5.6.4.1.1. ბ

WHILE...BREAK & CONTINUE

5.6.5.1.1. ბ
 5.6.5.1.2. ა

5.6.5.1.3. ბ
 5.6.5.1.4. ა
 5.6.5.1.5. ბ
 5.6.5.1.6. ა

ლოგიკის ოპერატორები

ALL ოპერატორი

5.7.1.1.1. ა
 5.7.1.1.2. ბ
 5.7.1.1.3. ბ
 5.7.1.1.4. ბ

SOME და ANY ოპერატორები

5.7.2.1.1. ა
 5.7.2.1.2. ბ

BETWEEN ოპერატორი

5.7.3.1.1. ბ
 5.7.3.1.2. ბ
 5.7.3.1.3. ა
 5.7.3.1.4. ა

EXISTS ოპერატორი

5.7.4.1.1. ბ
 5.7.4.1.2. ბ

IN ოპერატორი

5.7.5.1.1. ა

LIKE ოპერატორი

5.7.6.1.1. ბ
 5.7.6.1.2. ბ
 5.7.6.1.3. ა
 5.7.6.1.4. ბ
 5.7.6.1.5. ბ
 5.7.6.1.6. ა
 5.7.6.1.7. ა

თავი 6. ფუნქციები

მომხმარებლის ფუნქციები

6.1.1.1.1. ა
 6.1.1.1.2. ბ
 6.1.1.1.3. გ
 6.1.1.1.4. ბ
 6.1.1.1.5. ა
 6.1.1.1.6. გ
 6.1.1.1.7. ბ

6.1.1.1.8. ბ
 6.1.1.1.9. ა
 6.1.1.1.10. ბ
 6.1.1.1.11. ა
 6.1.1.1.12. გ
 6.1.1.1.13. ბ
 6.1.1.1.14. ა

Scalar ტიპის ფუნქციები

6.1.2.1.1. გ
 6.1.2.1.2. ბ
 6.1.2.1.3. ბ
 6.1.2.1.4. ა
 6.1.2.1.5. ბ
 6.1.2.1.6. ბ
 6.1.2.1.7. ბ
 6.1.2.1.8. ა
 6.1.2.1.9. გ
 6.1.2.1.10. ბ
 6.1.2.1.11. ა
 6.1.2.1.12. ბ
 6.1.2.1.13. გ

Inline ტიპის ფუნქციები

6.1.3.1.1. ა
 6.1.3.1.2. გ
 6.1.3.1.3. გ
 6.1.3.1.4. ბ

Multi-statement ფუნქციები

6.1.4.1.1. ა
 6.1.4.1.2. ბ
 6.1.4.1.3. ა
 6.1.4.1.4. ბ
 6.1.4.1.5. გ
 6.1.4.1.6. გ
 6.1.4.1.7. ბ

ფუნქციის შეცვლა

6.2.1.1.1. ა
 6.2.1.1.2. ბ
 6.2.1.1.3. გ
 6.2.1.1.4. ა

ფუნქციის წაშლა

6.3.1.1.1. ა
 6.3.1.1.2. გ

6.3.1.1.3. ბ
6.3.1.1.4. ბ

**ჩადგმული ფუნქციები
მათემატიკის ფუნქციები**

6.4.1.1.1. ა
6.4.1.1.2. გ
6.4.1.1.3. ბ
6.4.1.1.4. ბ
6.4.1.1.5. ა
6.4.1.1.6. ბ
6.4.1.1.7. გ
6.4.1.1.8. ა
6.4.1.1.9. ა
6.4.1.1.10. ბ
6.4.1.1.11. ა
6.4.1.1.12. გ
6.4.1.1.13. გ
6.4.1.1.14. ბ
6.4.1.1.15. ბ
6.4.1.1.16. ა
6.4.1.1.17. გ
6.4.1.1.18. გ
6.4.1.1.19. ბ
6.4.1.1.20. ა

სტრიქონებთან სამუშაო ფუნქციები

6.4.2.1.1. ა
6.4.2.1.2. ა
6.4.2.1.3. ბ
6.4.2.1.4. ბ
6.4.2.1.5. გ
6.4.2.1.6. გ
6.4.2.1.7. გ
6.4.2.1.8. ა
6.4.2.1.9. ბ
6.4.2.1.10. გ
6.4.2.1.11. ა
6.4.2.1.12. გ
6.4.2.1.13. ბ
6.4.2.1.14. ბ
6.4.2.1.15. ა
6.4.2.1.16. გ
6.4.2.1.17. გ
6.4.2.1.18. ბ
6.4.2.1.19. ბ
6.4.2.1.20. გ

6.4.2.1.21. ა

თარიღებთან სამუშაო ფუნქციები

6.4.3.1.1. ა
6.4.3.1.2. ბ
6.4.3.1.3. ა
6.4.3.1.4. ბ
6.4.3.1.5. გ
6.4.3.1.6. ა
6.4.3.1.7. ბ
6.4.3.1.8. ბ
6.4.3.1.9. გ

კონფიგურირების ფუნქციები

6.4.4.1.1. ა
6.4.4.1.2. ბ
6.4.4.1.3. ა
6.4.4.1.4. ა
6.4.4.1.5. ბ
6.4.4.1.6. გ
6.4.4.1.7. გ
6.4.4.1.8. ბ
6.4.4.1.9. ბ
6.4.4.1.10. გ
6.4.4.1.11. ა
6.4.4.1.12. ა
6.4.4.1.13. გ
6.4.4.1.14. ბ

სისტემური ფუნქციები

6.4.5.1.1. ა
6.4.5.1.2. ბ
6.4.5.1.3. ა
6.4.5.1.4. ა
6.4.5.1.5. ბ
6.4.5.1.6. გ
6.4.5.1.7. გ
6.4.5.1.8. ა
6.4.5.1.9. ბ
6.4.5.1.10. გ
6.4.5.1.11. ბ
6.4.5.1.12. ა
6.4.5.1.13. გ
6.4.5.1.14. ა
6.4.5.1.15. ბ
6.4.5.1.16. ა
6.4.5.1.17. გ

- 6.4.5.1.18. ბ
- 6.4.5.1.19. ბ
- 6.4.5.1.20. ბ
- 6.4.5.1.21. ა

უსაფრთხოების სისტემის ფუნქციები

- 6.4.6.1.1. ა
- 6.4.6.1.2. ბ
- 6.4.6.1.3. გ
- 6.4.6.1.4. ბ
- 6.4.6.1.5. ბ
- 6.4.6.1.6. ა
- 6.4.6.1.7. გ
- 6.4.6.1.8. ბ
- 6.4.6.1.9. ა

მეტამონაცემების მართვის ფუნქციები

- 6.4.7.1.1. ა
- 6.4.7.1.2. ბ
- 6.4.7.1.3. ა
- 6.4.7.1.4. გ
- 6.4.7.1.5. გ
- 6.4.7.1.6. ბ
- 6.4.7.1.7. ბ
- 6.4.7.1.8. ა
- 6.4.7.1.9. გ
- 6.4.7.1.10. ბ
- 6.4.7.1.11. ა
- 6.4.7.1.12. გ
- 6.4.7.1.13. გ
- 6.4.7.1.14. ა
- 6.4.7.1.15. ა
- 6.4.7.1.16. ბ
- 6.4.7.1.17. გ
- 6.4.7.1.18. ბ
- 6.4.7.1.19. ა

სტატისტიკის ფუნქციები

- 6.4.8.1.1. ა
- 6.4.8.1.2. ბ
- 6.4.8.1.3. გ
- 6.4.8.1.4. ა
- 6.4.8.1.5. ბ
- 6.4.8.1.6. ა
- 6.4.8.1.7. გ
- 6.4.8.1.8. გ
- 6.4.8.1.9. ბ

- 6.4.8.1.10. ა

თავი 7. შენახული პროცედურები შესავალი

- 7.1.1.1.1. ა
- 7.1.1.1.2. ბ
- 7.1.1.1.3. გ
- 7.1.1.1.4. ბ
- 7.1.1.1.5. ა
- 7.1.1.1.6. ბ
- 7.1.1.1.7. ბ
- 7.1.1.1.8. გ
- 7.1.1.1.9. ბ
- 7.1.1.1.10. ა
- 7.1.1.1.11. ბ
- 7.1.1.1.12. გ

შენახული პროცედურების ტიპები

- 7.2.1.1.1. ა
- 7.2.1.1.2. ბ
- 7.2.1.1.3. გ
- 7.2.1.1.4. ბ
- 7.2.1.1.5. ა
- 7.2.1.1.6. გ
- 7.2.1.1.7. ბ
- 7.2.1.1.8. ა
- 7.2.1.1.9. გ
- 7.2.1.1.10. ა
- 7.2.1.1.11. ბ
- 7.2.1.1.12. გ

შენახული პროცედურის შექმნა

- 7.3.1.1.1. ა
- 7.3.1.1.2. ბ
- 7.3.1.1.3. გ
- 7.3.1.1.4. ა
- 7.3.1.1.5. ა
- 7.3.1.1.6. გ
- 7.3.1.1.7. ბ
- 7.3.1.1.8. გ
- 7.3.1.1.9. ბ
- 7.3.1.1.10. ა
- 7.3.1.1.11. გ
- 7.3.1.1.12. ბ
- 7.3.1.1.13. ა
- 7.3.1.1.14. გ
- 7.3.1.1.15. გ

7.3.1.1.16.	ა	თავი 8. ინდექსები
7.3.1.1.17.	ბ	შესავალი
7.3.1.1.18.	ბ	8.1.1.1.1.
7.3.1.1.19.	ა	8.1.1.1.2.
7.3.1.1.20.	ბ	8.1.1.1.3.
7.3.1.1.21.	გ	8.1.1.1.4.
7.3.1.1.22.	ა	8.1.1.1.5.
7.3.1.1.23.	გ	
7.3.1.1.24.	ბ	ინდექსების გამოყენების დაგეგმვა
7.3.1.1.25.	ა	8.2.1.1.1.
7.3.1.1.26.	გ	8.2.1.1.2.
7.3.1.1.27.	ბ	8.2.1.1.3.
7.3.1.1.28.	ბ	8.2.1.1.4.
7.3.1.1.29.	გ	8.2.1.1.5.
7.3.1.1.30.	გ	8.2.1.1.6.
7.3.1.1.31.	ბ	
7.3.1.1.32.	ა	არაკლასტერული ინდექსი
7.3.1.1.33.	გ	8.2.2.1.1.
7.3.1.1.34.	ბ	8.2.2.1.2.
7.3.1.1.35.	ა	8.2.2.1.3.
7.3.1.1.36.	ა	
7.3.1.1.37.	ა	კლასტერული ინდექსი
7.3.1.1.38.	ა	8.2.3.1.1.
7.3.1.1.39.	ა	8.2.3.1.2.
		8.2.3.1.3.
		8.2.3.1.4.
		8.2.3.1.5.
		8.2.3.1.6.
		8.2.3.1.7.
		8.2.3.1.8.
		8.2.3.1.9.
		8.2.3.1.10.
		8.2.3.1.11.
		8.2.3.1.12.
		8.2.3.1.13.
შენახული პროცედურების მართვა		
შენახული პროცედურის შეცვლა		
7.4.1.1.1.	ა	
შენახული პროცედურის შესახებ ინფორმაციის მიღება		
7.4.2.1.1.	ა	
შენახული პროცედურის სახელის შეცვლა		
7.4.3.1.1.	ა	
შენახული პროცედურის წაშლა		
7.4.4.1.1.	ა	უნიკალური ინდექსი
		8.2.4.1.1.
		8.2.4.1.2.
		8.2.4.1.3.
		8.2.4.1.4.
შენახული პროცედურის ავტომატურად შესრულების მართვა		
7.4.5.1.1.	ა	შევსების ფაქტორი
7.4.5.1.2.	ბ	8.2.5.1.1.
7.4.5.1.3.	გ	8.2.5.1.2.
		8.2.5.1.3.
		8.2.5.1.4.

8.2.5.1.5. გ
8.2.5.1.6. ა
8.2.5.1.7. ბ
8.2.5.1.8. გ

ინდექსის შექმნა

8.3.1.1.1. ბ
8.3.1.1.2. ა
8.3.1.1.3. გ
8.3.1.1.4. ბ
8.3.1.1.5. გ
8.3.1.1.6. ბ
8.3.1.1.7. ა
8.3.1.1.8. გ
8.3.1.1.9. ბ
8.3.1.1.10. გ
8.3.1.1.11. ა
8.3.1.1.12. ა
8.3.1.1.13. ბ
8.3.1.1.14. ა
8.3.1.1.15. ა
8.3.1.1.16. ბ
8.3.1.1.17. ბ
8.3.1.1.18. გ
8.3.1.1.19. გ
8.3.1.1.20. ბ
8.3.1.1.21. ბ
8.3.1.1.22. ა
8.3.1.1.23. გ
8.3.1.1.24. ბ
8.3.1.1.25. ა
8.3.1.1.26. ბ
8.3.1.1.27. ა
8.3.1.1.28. ბ
8.3.1.1.29. გ
8.3.1.1.30. ა
8.3.1.1.31. ა
8.3.1.1.32. ა
8.3.1.1.33. ბ
8.3.1.1.34. ბ
8.3.1.1.35. გ
8.3.1.1.36. გ

ინდექსების მართვა

ინდექსის შეცვლა

8.4.1.1.1. ა
8.4.1.1.2. ბ
8.4.1.1.3. ბ

8.4.1.1.4. გ
8.4.1.1.5. ბ

ინდექსისთვის სახელის შეცვლა

8.4.2.1.1. ა

ინდექსის წაშლა

8.4.3.1.1. გ
8.4.3.1.2. ა

ინდექსების გადაწყობა

8.4.4.1.1. გ
8.4.4.1.2. ბ
8.4.4.1.3. ა
8.4.4.1.4. გ
8.4.4.1.5. ბ
8.4.4.1.6. ა

ინდექსის შესახებ ინფორმაციის მიღება

8.4.5.1.1. ა
8.4.5.1.2. ბ
8.4.5.1.3. გ
8.4.5.1.4. ბ
8.4.5.1.5. ა
8.4.5.1.6. ა
8.4.5.1.7. ბ
8.4.5.1.8. გ
8.4.5.1.9. ა

თავი 9. წარმოდგენები

შესავალი

9.1.1.1.1. გ
9.1.1.1.2. ა
9.1.1.1.3. ბ
9.1.1.1.4. გ
9.1.1.1.5. ა
9.1.1.1.6. ბ
9.1.1.1.7. ა
9.1.1.1.8. გ
9.1.1.1.9. გ
9.1.1.1.10. გ
9.1.1.1.11. გ
9.1.1.1.12. გ
9.1.1.1.13. ბ
9.1.1.1.14. ა
9.1.1.1.15. ა

9.1.1.1.16.	ბ	
9.1.1.1.17.	ბ	კურსორების რეალიზება
9.1.1.1.18.	ა	10.1.1.1.1.
9.1.1.1.19.	ბ	10.1.1.1.2.
9.1.1.1.20.	ბ	10.1.1.1.3.
9.1.1.1.21.	ბ	10.1.1.1.4.
9.1.1.1.22.	ბ	10.1.1.1.5.
9.1.1.1.23.	ა	10.1.1.1.6.
9.1.1.1.24.	ა	10.1.1.1.7.
		10.1.1.1.8.
წარმოდგენები და უსაფრთხოება		10.1.1.1.9.
9.2.1.1.1.	ა	10.1.1.1.10.
9.2.1.1.2.	ბ	
		კურსორის ტიპები
წარმოდგენის შექმნა		სტატიკური კურსორები
9.3.1.1.1.	ბ	10.2.1.1.1.
9.3.1.1.2.	ა	10.2.1.1.2.
9.3.1.1.3.	ბ	10.2.1.1.3.
9.3.1.1.4.	ბ	10.2.1.1.4.
9.3.1.1.5.	ბ	
9.3.1.1.6.	ა	დინამიკური კურსორები
9.3.1.1.7.	ბ	10.2.2.1.1.
9.3.1.1.8.	ბ	10.2.2.1.2.
		10.2.2.1.3.
წარმოდგენების მართვა		10.2.2.1.4.
წარმოდგენის შეცვლა		10.2.2.1.5.
9.4.1.1.1.	ბ	10.2.2.1.6.
წარმოდგენის სახელის შეცვლა		მიმდევრობითი კურსორები
9.4.2.1.1.	ა	10.2.3.1.1.
		10.2.3.1.2.
წარმოდგენის წაშლა		10.2.3.1.3.
9.4.3.1.1.	ბ	10.2.3.1.4.
9.4.3.1.2.	ბ	10.2.3.1.5.
9.4.3.1.3.	ა	
		საგასაღებო კურსორები
წარმოდგენის შესახებ ინფორმაციის მიღება		10.2.4.1.1.
9.4.4.1.1.	ბ	10.2.4.1.2.
9.4.4.1.2.	ბ	10.2.4.1.3.
		10.2.4.1.4.
წარმოდგენის დამოკიდებულებების ნახვა		10.2.4.1.5.
9.4.5.1.1.	ბ	10.2.4.1.6.
		10.2.4.1.7.
		10.2.4.1.8.
თავი 10. კურსორები		
შესავალი		კურსორების მართვა
10.1.1.1.1.	ა	10.3.1.1.1.
10.1.1.1.2.	ბ	

10.3.1.1.2. ბ
10.3.1.1.3. ბ
10.3.1.1.4. ა
10.3.1.1.5. გ

კურსორის შექმნა

10.3.2.1.1. ბ
10.3.2.1.2. გ
10.3.2.1.3. ბ
10.3.2.1.4. ა
10.3.2.1.5. გ
10.3.2.1.6. ბ
10.3.2.1.7. ბ
10.3.2.1.8. ა
10.3.2.1.9. გ
10.3.2.1.10. ბ
10.3.2.1.11. ა
10.3.2.1.12. გ
10.3.2.1.13. ბ
10.3.2.1.14. ა
10.3.2.1.15. ბ
10.3.2.1.16. ა
10.3.2.1.17. ბ
10.3.2.1.18. გ
10.3.2.1.19. ბ
10.3.2.1.20. ა
10.3.2.1.21. ა
10.3.2.1.22. ბ
10.3.2.1.23. გ
10.3.2.1.24. ა
10.3.2.1.25. ა
10.3.2.1.26. ბ

კურსორის გახსნა

10.3.3.1.1. ბ
10.3.3.1.2. გ

მონაცემების წაკითხვა

10.3.4.1.1. ა
10.3.4.1.2. ბ
10.3.4.1.3. გ
10.3.4.1.4. ა
10.3.4.1.5. ბ
10.3.4.1.6. გ
10.3.4.1.7. ბ
10.3.4.1.8. ა
10.3.4.1.9. ბ

10.3.4.1.10. გ
10.3.4.1.11. ბ
10.3.4.1.12. ა
10.3.4.1.13. ბ
10.3.4.1.14. გ
10.3.4.1.15. ბ
10.3.4.1.16. გ
10.3.4.1.17. ა

მონაცემების შეცვლა

10.3.5.1.1. ბ
10.3.5.1.2. გ
10.3.5.1.3. ბ
10.3.5.1.4. ა

მონაცემების წაშლა

10.3.6.1.1. გ

კურსორის დახურვა

10.3.7.1.1. ა
10.3.7.1.2. ბ
10.3.7.1.3. გ
10.3.7.1.4. ბ

კურსორის გათავისუფლება

10.3.8.1.1. ა
10.3.8.1.2. გ

თავი 11. ტრანზაქციები და დაბლოკვები შესავალი

11.1.1.1.1. გ
11.1.1.1.2. ა
11.1.1.1.3. ბ
11.1.1.1.4. გ
11.1.1.1.5. ა
11.1.1.1.6. ბ
11.1.1.1.7. ა
11.1.1.1.8. ბ
11.1.1.1.9. გ
11.1.1.1.10. ა
11.1.1.1.11. ბ
11.1.1.1.12. ბ
11.1.1.1.13. გ
11.1.1.1.14. ა
11.1.1.1.15. გ
11.1.1.1.16. ა
11.1.1.1.17. ბ

- 11.1.1.1.18. ბ
- 11.1.1.1.19. გ
- 11.1.1.1.20. ა
- 11.1.1.1.21. გ
- 11.1.1.1.22. ბ
- 11.1.1.1.23. გ
- 11.1.1.1.24. ბ
- 11.1.1.1.25. ა
- 11.1.1.1.26. გ
- 11.1.1.1.27. ბ
- 11.1.1.1.28. გ
- 11.1.1.1.29. ბ
- 11.1.1.1.30. ა
- 11.1.1.1.31. გ
- 11.1.1.1.32. ბ
- 11.1.1.1.33. გ
- 11.1.1.1.34. ა
- 11.1.1.1.35. გ
- 11.1.1.1.36. ა
- 11.1.1.1.37. ბ
- 11.1.1.1.38. გ
- 11.1.1.1.39. ა
- 11.1.1.1.40. ბ
- 11.1.1.1.41. ა
- 11.1.1.1.42. გ
- 11.1.1.1.43. ა
- 11.1.1.1.44. გ
- 11.1.1.1.45. ბ
- 11.1.1.1.46. ბ

ტრანზაქციების მართვა

- 11.2.1.1.1. ა
- 11.2.1.1.2. ბ
- 11.2.1.1.3. გ
- 11.2.1.1.4. ბ

აშკარა ტრანზაქციები

- 11.2.2.1.1. გ
- 11.2.2.1.2. ა
- 11.2.2.1.3. ბ
- 11.2.2.1.4. ბ
- 11.2.2.1.5. ა
- 11.2.2.1.6. გ
- 11.2.2.1.7. ბ
- 11.2.2.1.8. ბ
- 11.2.2.1.9. ა
- 11.2.2.1.10. გ

- 11.2.2.1.11. გ
- 11.2.2.1.12. ა

ავტომატური ტრანზაქციები

- 11.2.3.1.1. ბ
- 11.2.3.1.2. ა
- 11.2.3.1.3. გ
- 11.2.3.1.4. ბ

არააშკარა ტრანზაქციები

- 11.2.4.1.1. ბ
- 11.2.4.1.2. ა
- 11.2.4.1.3. ბ
- 11.2.4.1.4. გ

განაწილებული ტრანზაქციები

- 11.2.5.1.1. ბ
- 11.2.5.1.2. გ
- 11.2.5.1.3. ბ
- 11.2.5.1.4. ა
- 11.2.5.1.5. ბ
- 11.2.5.1.6. გ

ჩადგმული ტრანზაქციები

- 11.2.6.1.1. ბ
- 11.2.6.1.2. ა
- 11.2.6.1.3. ბ
- 11.2.6.1.4. გ
- 11.2.6.1.5. ბ
- 11.2.6.1.6. ა
- 11.2.6.1.7. ბ
- 11.2.6.1.8. გ
- 11.2.6.1.9. ა

თავი 12. ტრიგერები

შესავალი

- 12.1.1.1.1. ა
- 12.1.1.1.2. ბ
- 12.1.1.1.3. გ
- 12.1.1.1.4. ბ
- 12.1.1.1.5. გ
- 12.1.1.1.6. ა
- 12.1.1.1.7. ბ
- 12.1.1.1.8. გ
- 12.1.1.1.9. ბ
- 12.1.1.1.10. გ
- 12.1.1.1.11. ა

- 12.1.1.1.12. ბ
- 12.1.1.1.13. ბ
- 12.1.1.1.14. ბ
- 12.1.1.1.15. ბ
- 12.1.1.1.16. ა
- 12.1.1.1.17. ა
- 12.1.1.1.18. ბ
- 12.1.1.1.19. ბ
- 12.1.1.1.20. ა
- 12.1.1.1.21. ბ
- 12.1.1.1.22. ა
- 12.1.1.1.23. ბ

ტრიგერის შექმნა

- 12.2.1.1.1. ა
- 12.2.1.1.2. ბ
- 12.2.1.1.3. ბ
- 12.2.1.1.4. ბ
- 12.2.1.1.5. ბ
- 12.2.1.1.6. ბ
- 12.2.1.1.7. ა
- 12.2.1.1.8. ბ
- 12.2.1.1.9. ა
- 12.2.1.1.10. ბ
- 12.2.1.1.11. ა
- 12.2.1.1.12. ბ
- 12.2.1.1.13. ბ
- 12.2.1.1.14. ბ
- 12.2.1.1.15. ბ
- 12.2.1.1.16. ა
- 12.2.1.1.17. ბ
- 12.2.1.1.18. ბ
- 12.2.1.1.19. ბ
- 12.2.1.1.20. ბ
- 12.2.1.1.21. ა
- 12.2.1.1.22. ბ

ტრიგერის შეცვლა

- 12.3.1.1.1. ბ
- 12.3.1.1.2. ბ

ტრიგერის წაშლა

- 12.4.1.1.1. ა
- 12.4.1.1.2. ბ

ტრიგერის დაპროგრამება

- 12.5.1.1.1. ა
- 12.5.1.1.2. ბ
- 12.5.1.1.3. ბ
- 12.5.1.1.4. ბ
- 12.5.1.1.5. ა
- 12.5.1.1.6. ბ
- 12.5.1.1.7. ბ
- 12.5.1.1.8. ბ
- 12.5.1.1.9. ა
- 12.5.1.1.10. ა
- 12.5.1.1.11. ბ
- 12.5.1.1.12. ბ
- 12.5.1.1.13. ბ
- 12.5.1.1.14. ა
- 12.5.1.1.15. ბ
- 12.5.1.1.16. ა
- 12.5.1.1.17. ბ
- 12.5.1.1.18. ბ
- 12.5.1.1.19. ა
- 12.5.1.1.20. ა
- 12.5.1.1.21. ბ
- 12.5.1.1.22. ბ
- 12.5.1.1.23. ბ
- 12.5.1.1.24. ბ
- 12.5.1.1.25. ა

ტრიგერების მართვა

ტრიგერის სახელის შეცვლა

- 12.6.1.1.1. ა

ტრიგერის შექმნის დროს არსებული შეზღუდვები

- 12.7.1.1.1. ბ
- 12.7.1.1.2. ბ

ტრიგერის შესახებ ინფორმაციის მიღება

- 12.7.2.1.1. ბ
- 12.7.2.1.2. ბ
- 12.7.2.1.3. ა
- 12.7.2.1.4. ბ
- 12.7.2.1.5. ბ

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.

რედაქტორი ლ. მამალაძე

კომპიუტერული უზრუნველყოფა რ. სამხარაძის

იბეჭდება ავტორის მიერ წარმოდგენილი სახით

გადაეცა წარმოებას 19.03.2010. ხელმოწერილია დასაბეჭდად 30.04.2010. ქალაქის ზომა 60X84 1/8. პირობითი ნაბეჭდი თაბახი 14. ტირაჟი 100 ეგზ.

საგამომცემლო სახლი “ტექნიკური უნივერსიტეტი”, თბილისი, კოსტავას 77