საქართველოს ტექნიკური უნივერსიტეტი

ნომერ სამხარეთ, ბორო ჯაყუ, ლოს გამენოფაძე, ა.ქურდაძე

მეთოდური მითითებები ლაბორატორიული სამუშაოების შესასრულებლად

Transact-SQL ვერსია 1

I ნაწილი
საქართველოს ტექნიკური უნივერსიტეტი

რომა სამხარაძე, ნინო ჯოჯუა, ლია გაჩჩელაძე, მ.ქურდაძე

მეთოდური მითითებები ლაბორატორიული სამუშაოების შესასრულებლად საგანში:

Transact-SQL გვე 1

I ნაწილი
სტუდენტური მოთავსების პროგრამის ნაწილი განკუთვნილია იმ ფაკულტეტის ბაკალავრიატის სტუდენტების საგანგებო სამუშაოების ჩატარებისთვის. მასში განხილულია მონაცემთა ბაზების არქიტექტურა, ცხრილები, მონაცემთა და მომხმარებლების ტიეფა, ცხრილში მონაცემების ჩამატება, მონაცემების ამორჩევა, ჩადგმული და ბმული მოთხოვნები და შ.შ. თანხმები, უსაფრთხო საქმიანობა სახ. პროფესორი ზაურ ჯოჯუა, პროფესორი მედეა ანდღულაძე, ასოც. პროფ. რეცენზები:
<table>
<thead>
<tr>
<th>სახაზული</th>
<th>გვ.</th>
</tr>
</thead>
<tbody>
<tr>
<td>სარჩევი ლაბორატორიული სამუშაო N1</td>
<td>6</td>
</tr>
<tr>
<td>SQL Server  გარემოს აღწერა</td>
<td>6</td>
</tr>
<tr>
<td>ლაბორატორიული სამუშაო N2</td>
<td>9</td>
</tr>
<tr>
<td>მონაცემთა ბაზის შექმნა</td>
<td>9</td>
</tr>
<tr>
<td>ლაბორატორიული სამუშაო N3</td>
<td>13</td>
</tr>
<tr>
<td>ცხრილების შექმნა</td>
<td>13</td>
</tr>
<tr>
<td>ლაბორატორიული სამუშაო N4</td>
<td>17</td>
</tr>
<tr>
<td>ცხრილში მონაცემების ჩამატება</td>
<td>17</td>
</tr>
<tr>
<td>ლაბორატორიული სამუშაო N5</td>
<td>20</td>
</tr>
<tr>
<td>ცხრილში მონაცემების ამორჩევა. SELECT ბრძანება</td>
<td>20</td>
</tr>
<tr>
<td>ლაბორატორიული სამუშაო N6</td>
<td>22</td>
</tr>
<tr>
<td>SELECT ბრძანება, FROM განყოფილება</td>
<td>22</td>
</tr>
<tr>
<td>ლაბორატორიული სამუშაო N7</td>
<td>25</td>
</tr>
<tr>
<td>ჩადგმული და ბმული მოთხოვნები</td>
<td>25</td>
</tr>
<tr>
<td>ლაბორატორიული სამუშაო N8</td>
<td>28</td>
</tr>
<tr>
<td>GROUP BY  განყოფილება და აგრეგირების ფუნქციები</td>
<td>28</td>
</tr>
<tr>
<td>ლაბორატორიული სამუშაო N9</td>
<td>30</td>
</tr>
<tr>
<td>HAVING, ORDER BY  და UNION  განყოფილებები</td>
<td>30</td>
</tr>
<tr>
<td>ლაბორატორიული სამუშაო N10</td>
<td>33</td>
</tr>
</tbody>
</table>
ცხრილის მონაცემების ცვლილება და ტექსტიდან მონაცემების წაშლა. UPDATE და DELETE ბრძანებები. ........33
ლაბორატორიული სამუშაო N11 ........................................35
Transact SQL-ის ძირთასაზღვრებები, ქველთავები........35
ლაბორატორიული სამუშაო N12 ........................................39
მაგარითული პროგრამავენრი: END...BEGIN, IF...ELSE,
CASE...END ........................................................................39
ლაბორატორიული სამუშაო N13 ........................................42
მაგარითული პროგრამავენრი: WHILE...BREAK&CONTINUE42
ლაბორატორიული სამუშაო N14 ........................................44
ლოგიკის თვალსაზრისი: ALL, IN, LIKE .................................44
ლაბორატორიული სამუშაო N15 ........................................47
მონაცემების შესაბამის გადაწერა .........................47
ლიტერატურა .......................................................................50
ლაბორატორიული სამუშაო N1
SQL Server გარემოს აღწერა

SQL Server Management Studio-ს (სურ.1) გამოყენების შემდეგ გამოჩნდება დიალოგური ფანჯარა (სურ.2), სადაც ვითარებს SQL Server-ის გარემოს აღწერა. Windows ან SQL Server-ის შემთხვევაში უნდა ვითარებოთ SQL Server Management Studio-ის login ველში უჯრავთ sa, ხოლო password ველში პაროლს.

Database Engine შეიძლება მიიღოს უამრავ უზრუნველყოფით, რომელიც წარმოადგენს Database Engine-ის ლოგისტიკის და დაცვის მეთოდებს.

მონაცემების ამოცანები:

- უზრუნველყოფს მონაცემების სიმკვრივი დაცვა
- უზრუნველყოფს მონაცემების საშუალო ინფორმაციას
- უზრუნველყოფს მონაცემების პოლიტიკების დამტკიცება
- უზრუნველყოფს მონაცემების უსაფრთხოების
- მონაცემების საიმედოობის და ადეკვატურობის ადგილზე შესაძლო ინფორმაციის პირობებს
- უზრუნველყოფს მონაცემების ვერსიების დამტკიცება

სურ.1
სურ. 3-4-ს შეფარდებით შეიძლება მიღებული იქნებოდა მონაცემთა ახალი ბაზის, მოთხოვნების და სხვადასხვა ამოცანების, მონაცემთა ბაზის სარეზერვო სხვა, განახლება, წაშლა, მონაცემთა ბაზის სარეზერვო ასლიდან შექმნა, აღდგენა სარეზერვო ასლიდან, მონაცემთა იმპორტი და ექსპორტი და ს.შ. შესრულება.
ლაბორატორიული სამუშაო N2
მონაცემთა ბაზის შექმნა

მონაცემთა ბაზის შექმნაქმებთ უფლებით მოხუცით მოსახმავად, ბლოგით სამუშაო, გაგრძელდება მოსახმავად (ეს იქნება მონაცემთა ბაზის შექმნის მოთხოვნების), უოპ, ფაილები და ფაილების ჯგუფები, რომლებითაც უნდა გადადეს შესაქმნილი ინფორმაციის უფლება.

იმისთვის, რომ შექმნათ მონაცემთა ბაზა, უნდა გამოიყენოთ CREATE DATABASE ბრძანება. მისი სინტაქსია:

```
CREATE DATABASE მონაცემთა_ბაზის_სახელი 
[ ON [ PRIMARY ] [ <ფაილის_განსაზღვრა> [ ,...n] ] [ ,
<ფაილების_ჯგუფ> [ ,...n ] ]
[ LOG ON { <ფაილის_განსაზღვრა> [ ,...n ] } ]
[ FOR ATTACH ]
```

სადაც,

- მონაცემთა_ბაზის_სახელი შექმნილი იქნება შექმნილი ბაზის სახელის.
- ON მოთხოვნა, რომ ფაილებს შექმნათ საჭირო ფაილების განსაზღვრა.
- PRIMARY მოთხოვნა, რომ ფაილებს შექმნათ პირველად მომხმარებლის ფაილი.
- LOG მოთხოვნა, რომ ფაილები შექმნათ ერთ ფაილში საჭირო ფაილების განსაზღვრა.

თუ პირველადი განსაზღვრებები სხვასაგან არ არიან განსაზღვრული, შექმნილი ბაზა ვერ შეიქმნა. შექმნილი ბაზა შეიქმნა შექმნილი
იქნება <ფაილის_ენისაუზრედ> კონსტრუქციის მითითებული პირველი ფაილი.

• LOG ON გამოსახულება ბუნებრისხის კონსტრუქციის ფაილის <ფაილის_განსაზღვრა> განსაზღვრა ფაილის ლოგიკურ სახელით. თუ ეს არ გამოქვეყნდება ამ ფაილის მითითებულ, მისი სტატუსი განიცვალებს ჩატარების გამო ფაილში, ნოველის სეიტი იქნება მინიმალურად შაბლონის სახელისათვის _Log სიმბოლოების ფაილის გზით.

• FOR ATTACH. გამოვიყენოთ შემდეგ ვებიასში მოიცავათ შაბლონის ინფორმაცია. მათთვის უნდა აღიარდგეს შესაბამისი ფაილები. შესაბამისობით შაბლონის სახელი აღიარებული მივლინებით შეიცავს ფაილის პირველად ფაილის ადგილმდებარეობის მოთხოვნა.

<ფაილის_ენისაუზრედ> კონსტრუქციის სინტაქსია:
<ფაილის_ენისაუზრედ> ::=
 ( [ NAME = ფაილის_ლოგიკური_სახელი, ] FILENAME = 'ფაილის_ფიზიკური_სახელი'
 [ , SIZE = ზომა ] [ , MAXSIZE = { მაქსიმალური_ზომა | UNLIMITED } ]

• NAME = ფაილის_ლოგიკური_სახელი ფაილის ლოგიკური სახელია. ამ სახელს ფაილის ფარგლებში უნიკალური პირობებით შესაბამისად უნდა იყოს.

• FILENAME = 'ფაილის_ფიზიკური_სახელი' შეიცავს ფაილის ფიზიკური სახელს და გზას, რომელიც შეიცავს უმცრო ინფორმაციის დოკუმენტ.
• SIZE = ზომა. მითუთოვან ფაილთა სერვერის ზომის გამოყენება (Mb).
• FILEGROWTH = ირგოვით გრაფიკად მითუთოვან ფაილთა ზომის გაზრდის განსაზღვრა. მითუთოვან ფაილთა ზომის გაზრდის მითუთოვან ზომის გაზრდა შეიძლება მცირეადობით მცირეადობას ან პროცენტულ მცირეადობას (ფაილთა ზომის გაზრდა)
• MAXSIZE = { მაქსიმალური_ზომა | UNLIMITED }. ფაილთა მაქსიმალური ზომა ესაზღვრა მცირეადობას. თუ ფაილთა ზომის მცირეადობა არ უნდა შეიზღუდოს, მაშინ უნდა მოთხოვნა UNLIMITED შეიძლება ან სამხრეთიდან გამოგზავნა MAXSIZE არგუმენტი.

დამატება 1. შექმენით Baza_2 მონაცემთა ბაზა, რომელიც შედგება მონაცემების ერთი ფაილით და ტრანზაქციების ჟურნალის ერთი ფაილით. მონაცემების ფაილთა ზომა შეზღუდული არ არის.

USE Master
GO
IF DB_ID (N'Baza_2') IS NOT NULL
DROP DATABASE Baza_2;
GO
CREATE DATABASE Baza_2 ON
( NAME = Baza2,
  FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\Baza_2.mdf',
  MAXSIZE = UNLIMITED )
LOG ON
( NAME = Baza2_log,
FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\Baza_2_Log.ldf',
MAXSIZE = UNLIMITED )
GO

გამოსტურება 2. შექმნისთვის Baza2 მონაცემთა ბაზა. შექმნის
დროს შექმნის ფაილები მოთხოვნას ექნება.

USE Master
GO
IF DB_ID (N'Baza2') IS NOT NULL
DROP DATABASE Baza2;
GO
CREATE DATABASE Baza2
ON
( NAME = Baza2,
FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\Baza2.mdf',
SIZE = 15, MAXSIZE = 36, FILEGROWTH = 3 )
LOG ON
( NAME = Baza2_log,
FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\Baza2_Log.ldf',
SIZE = 3MB, MAXSIZE = 25MB, FILEGROWTH = 2MB )
ლაბორატორიული სამუშაო N3
გზითქმის შეტყობინება.

ცხრილის შექმნაზე CREATE TABLE მომხმარებისთვის. მისი სინტაქსით:

CREATE TABLE
[მონაცემთა_ბაზის_სახელი][სვეტის_სახელი]
({<სვეტის_განსაზღვრა>|<სვეტის_შეზღუდვა> } [,...n] )
განხორციელდება ორი ან მეტი ნაკადის ლოგიკით. მათგან:

• მონაცემთა_ბაზის_სახელი არის იმ ბაზის სახელი, რომელშიც ცხრილი იქმნება.
• საქემის_სახელი არის იმ სქემის სახელი, რომელსაც ცხრილი ეკუთვნის.
• ცხრილი_სახელი არის შექმნილი ცხრილის სახელი. ცხრილის სახელისა და სვეტის სახელის კომბინაციის უძველესი უდრში ოთხ მონაცემთა ბაზის დანგრძობილში.
• სვეტის_განსაზღვრა ან სვეტის_შეზღუდვა უძველესი ნაკადის შედეგი აღმოჩენილი მემკვიდრეობით თავისუფალი ლოგიკით.

სვეტის_განსაზღვრა> ართობების სიმწონები:
<სვეტის_სახელი> := სვეტი_სახელი ტიპი
[[DEFAULT განსაზღვრა]] [[IDENTITY[<სვეტის_სახელი>]]]  ]
[<სვეტის_შეზღუდვა> [,...n] [ ROWGUIDCOL ]]

სადელ,
• ტიპი განსაზღვრავს სვეტში მოთავსებული მონაცემების ტიპს.
• DEFAULT განსაზღვრავს ავტომატურ მნიშვნელობას მოთავსებულ სვეტისთვის.
• IDENTIFY მოთავთავებს, როგორ ქონებს სვეტში იქნება სვეტ-მოთავა.
• ROWGUIDCOL განსაზღვრავს ავტომატურ მნიშვნელობას, რომლებსაც სვეტში მოცემული იქნება.

დავალება 1. შექმენით ცხრილი, რომელიც ერთ-ერთი სვეტი არ უშვებს NULL მნიშვნელობას.

USE Baza_1
GO

-- თუ ცხრილი არსებობს, მაშინ ის წაიშლება
IF OBJECT_ID(N'Cxrili_1',N'U') IS NOT NULL
DROP TABLE Cxrili_1;
GO
-- ცხრილის შექმნა
CREATE TABLE Cxrili_1
(
    cxriliID int PRIMARY KEY CLUSTERED,
    sveti1 int,
დავალება 2. შექმენით ცხრილი. მისი ერთ-ერთი სვეტის ბაზაზე შექმენით არაკლასტერული ინდექსი.

USE Baza_1
GO
-- თუ ცხრილი არსებობს, მაშინ ის წაიშლება
IF OBJECT_ID(N'Cxrili_2',N'U') IS NOT NULL
DROP TABLE Cxrili_2;
GO
-- შექმნა
CREATE TABLE Cxrili_2
(
  cxriliID int PRIMARY KEY CLUSTERED,
  sveti1 int,
  sveti2 nvarchar(20) NOT NULL UNIQUE NONCLUSTERED,
  sveti3 float,
  sveti4 datetime
)

დავალება 3. შექმენით ცხრილი. განსაზღვრეთ რეჰიმი, როცა მთავარი ცხრილიდან შტრიხის წაშლას დამოკიდებული ცხრილში შტრიხები წაიშლება.
USE Baza_1
GO
-- თუ ცხრილი არსებობს, მაშინ ის წაიშლება
IF OBJECT_ID(N'Cxrili_4',N'U') IS NOT NULL
DROP TABLE Cxrili_4;
GO
-- ცხრილის შექმნა
CREATE TABLE Cxrili_4
(
  cxriliID int PRIMARY KEY,
  sveti1 int NULL FOREIGN KEY (sveti1) REFERENCES Cxrili_1(cxriliID) ON DELETE CASCADE,
  sveti2 nvarchar(20),
  sveti3 float,
  sveti4 datetime
)
INSERT მიმართვება იძლევა ცხრილში ერთი ან მეტი სტრიქონის ჩამატების საშუალებას. მისი სინტაქსიაი:

```
INSERT [INTO] { ცხრილის_სახელი | წარმოდგენის_სახელი } [ ( სვეტების_სია ) ] VALUES ( { DEFAULT | NULL | გამოსახულება } | მიღებულ_ცხრილი } ) | DEFAULT VALUES სადაც,
• [INTO] არაუცხოვალიყო არგუმენტი, რომელსაც მოიცავს იმ ცხრილის სახელს, რომელშიც მონაცემების ჩამატება ხდება;
• ცხრილის_სახელი იმ ცხრილის სახელს, რომელშიც სტრიქონების ჩამატება ხდება;
• წარმოდგენის_სახელი იმ წარმოდგენის სახელს, რომელშიც ხდება სტრიქონების ჩამატება.
• (სვეტების_სია) იმ სვეტების სახელების სია, რომლებისთვის უნდა ხდოთ მონაცემების ჩამატება. სვეტების მნიშვნელობები ვითარდება VALUES არგუმენტები. სვეტის მნიშვნელობა შეგვიძლია ამ პარამეტრით მაქსიმუმ მეტი მონაცემით, თუმცა სვეტის სია უფრო სახელის წარმოდგენაში არ გამოიყენება, NULL სვეტებით შეეხება ფუნქცია VALUES არგუმენტი.
• VALUES ( { DEFAULT | NULL | გამოსახულება } ). არგუმენტები ცხრილში ხსოვად მონაცემებას. მთავარი არგუმენტი ცხრილის სახელით უნდა ვითარდოს ცხოვრობა ან სვეტების_სია არგუმენტით მოთავსებული სვეტის მნიშვნელობა. DEFAULT არგუმენტი მოთავსდება, თუმცა
მოხდება ავტომატურად განსაზღვრული მნიშვნელობების საშიშ. თუ სენსორთან ავტომატურად მიმდინარეობს განსაზღვრული ამ აქტი და სვეტში NULL მნიშვნელობაა შემავალ, მისი სამშენო NULL მნიშვნელობაზე მთავაზარებდება.

- მოდულში შეიძლება SELECT ბრძანება შეიძლება შესაძლოა SELECT ბრძანები, რომლის საშუალებით ჩაწვეული ცხოვრობის შესაფერისი აქტიური შეიძლება შეიტანოთ სხვა ცხრილიდან მოძრაობა. თუმცა ცხოვრობის ნახვაში შეიძლება უდიდეს შეგროვდეს.
- DEFAULT VALUES არგუმენტის შემთხვევაში თითოეულ სვეტში მთავაზარებდება ავტომატურად განსაზღვრული მნიშვნელობა.

დასაშვები გზით 1. Personalı ცხრილს დავუმატოთ სტრიქონი, რომლის ყველა სვეტში ავტომატურად განსაზღვრად გამოყოფილი მნიშვნელობა შემავალ.

USE Shekveta
INSERT INTO Personalı DEFAULT VALUES
SELECT * FROM Personalı

დასაშვები გზით 2. ცხრილში ჩამოაქვთ მორ სტრიქონი. შესამჩღველი მნიშვნელობებში შევარდნენ მადილებათი შექმნით.

USE Shekveta
SET DATEFORMAT dmy
INSERT INTO Personali (gvari, xelfasi, asaki, qalaqi, raioni, misamarti, staji, tarigi_dabadebis, email, mobiluri, teleponi_samsaxuris, teleponi_saxlis, ganyofileba)
VALUES (N'სამხარაძე რომანი', 378, 37, N'თბილისი', N'', N'სტალინის ქ.5', 5, '20.02.1989', N'romani@geo.net.ge', '899-345012', '23-83-39', '23-21-71', N'სამედიცინო')

INSERT INTO Personali (email, mobiluri, teleponi_samsaxuris, teleponi_saxlis, gvari, xelfasi, asaki, qalaqi, raioni, misamarti, staji, tarigi_dabadebis, ganyofileba)
VALUES (N'archili@geo.net.ge', '899-012345', '30-80-09', '13-76-78', N'სამხარაძე არჩილი', 378, 37, N'ქუთაისი', N'', N'წერეთლის ქ.5', 5, '10.02.1959', N'სასპორტო')

SELECT * FROM Personali
ლაბორატორიული სამუშაო N5
გჟორირელი მონაცემების ამოღება. SELECT ბრძანება.

ქმნილობაში მონაცემების ამოღება თურქულებით SELECT ბრძანების გამოყენება. მის სინტაქსია:

```
SELECT [ INTO ახალი_ცხრილის_სახელი ] FROM საწყისი_ცხრილის_სახელი [ WHERE მექნის_პირობა ]
[ GROUP BY დაჯგუფების_გამოსახულება ]
[ HAVING ძებნის_პირობა ]
[ ORDER BY სისხლის_გამოსახულება [ ASC | DESC ] ]
```

SELECT გამოირჩევს საყვარელ საერთო სვეტების სიას. მის სინტაქსია:

```
SELECT [ ALL | DISTINCT ] [ TOP n [ PERCENT ] ] <სია>
```

სადაც,

- ALL არგუმენტი მოხიდებს, თუმცა მოთამაშობის ზემოხანგარეველი შესაძლოდაც გამოთვლილი (ერთწლოვანი) სტრიქონების გამოსახულება.
- DISTINCT არგუმენტი ეძლება გამოიყენა სტრიქონების გამოსახულება.
- TOP n [ PERCENT ] კონსტრუქცია გამოიყენება სიმძლავრის პირველ დასახელება და პირველი n პროცენტი. თუ TOP n არ მოდის, მაშინ მას მიართვა პროცენტი.
- <სია> კონსტრუქცია შეესაბამება სვეტების სიას.

ცხოვრობს 1. Personali და Shemkveti ცხრილებიდან თანამშრომლების სახელები და მათი ქალაქები. ტექსტის სიმძლავრის შესახელებათა ჩამორჩევისთვის, როგორც რომ მოძღვანი.
USE Shekveta
SELECT Personali.gvari AS თანამშრომლის გვარი, Personali.qalaqi AS თანამშრომლის ადგილმდებარეობა, Shemkveti.gvari AS შემკვეთის გვარი, Shemkveti.qalaqi AS შემკვეთის ადგილმდებარეობა FROM Personali, Shemkveti WHERE Personali.personaliID = Shemkveti.personaliID

2. გამოიყენება მონაცემთა ბიბლიოთეკაში, რომელიც მოეწონა ასაკი 50-ზე ან ნაკლები 30-ზე.

USE Shekveta
SELECT * FROM Personali WHERE asaki > 50 OR asaki < 30
ლაბორატორიული სამუშაო

SELECT ბრძანება, FROM განყოფილება.

FROM სიტყვის შემდევრ იხსნება იმ ეთითების ან წარმოდგენის სახელებთან, რაც გამოიყენება სვეტების ამორჩევა. მისი სინტაქსია:

FROM { ცხრილის_სახელი [ [ AS ] ცხრილის_ფსევდონიმი ]
  | წარმოდგენის_სახელი [ [ AS ] წარმოდგენის_ფსევდონიმი ] | <ბმული_ცხრილი> }

| <ბმული_ცხრილი> პოლიმფირლად გამოიყენება სახ- ბრძანების უფრო მსგავსად ნამდინი ცხრილი მორალ. მისი სინტაქსია:

<ბმული_ცხრილი> ::=  
| <მარცხნის_წსევდო> <მარჯვნის_წსევდო> <მარჯვენა_ტიპი> <მარცხენა_ტიპი> ON <ბმულ_პირობა>  | <მარცხენა_ცხრილი> CROSS JOIN <მარჯვენა_ცხრილი> | <ბმული_ცხრილი>

| <მარცხენა_წსევდო> პოლიმფირლად შეიძლება მსგავ- სახლი ცხრილის სახელი, რომლის მეტა-სხვაზომოდი სხვა ცხრილებზე. <მარჯვენ-ტიპი> პოლიმფირლად გამოიყენება არ ცხრილი მეტი ეთითების გამო. შეასრულოს ცხრილის სხვა გამოკვეტა ქონსული ეთითების მარჯვენ-ტიპში, ჰასურგამ მათ გამოკვეტა მთავართა ცხრილის სხვა.

<მარჯვე-ტიპი> პოლიმფირლად ნამდინ იქნება:

<მარჯვე-ტიპი> ::= [ INNER | { { LEFT | RIGHT | FULL } [ OUTER ] } ] JOIN

სადაც,
• INNER კავშირის შიდა ავტომატურად გამოყენება.
 შედეგში სახიერი თუ სურათის უფალმად ის პირობები, თითო ობიექტსაც არ ეხება მარცხენა ცხრილში მოცემული მარჯვენა ცხრილში.

• LEFT [ OUTER ] არგუმენტის გამოყენებით შედეგში ჩართული იქნება მარცხენა ცხრილის მხოლოდ იქით სტრიქონები, რომელთაგან უთან ამავდროულად გვირჩება, არჩევნის თუ არ არ მარჯვნიში აღმართვის სტრიქონები მარჯვენა ცხრილში.

• RIGHT [ OUTER ] არგუმენტის გამოყენებით შედეგში ჩართული იქნება მარჯვენა ცხრილის ყველა სტრიქონი მიუხედავად იმისა, თუ უთან ამავე მარცხენა ცხრილში არ არსებობს სტრიქონები.

• FULL [OUTER] არგუმენტის გამოყენებით შედეგში ჩართული იქნება მარჯვენა ცხრილის ყველა სტრიქონი მარცხენა ცხრილში.

• JOIN არგუმენტის შემდეგ ცხრილში მარჯვენა ცხრილი წარმოდგება.

• ON <შედადება> სიტყვის მეორე ცხრილის გათვალისწინებით შესაძლოა პირობები.

• CROSS JOIN სიტყვის მეორე ცხრილის გათვალისწინებით ხელოვნურად მგებლო ხაზმახაზო თანამორჩილ სტრიქონები მარჯვენა ცხრილის თანამორჩილ სტრიქონები.

მაგალითი 1. Shemkveti და Xelshekruleba ცხრილებში შორის LEFT OUTER კავშირი ფასდება.
USE Shekveta
SELECT Shemkveti.gvari, Shemkveti.qalaqi,
Xelshekruleba.gadasaxdeli_1, Xelshekruleba.vali_1
FROM Shemkveti LEFT OUTER JOIN Xelshekruleba
ON Shemkveti.ShemkvetiID = Xelshekruleba.ShemkvetiID

დავალება 2. 4.15. Shemkveti და Xelshekruleba ცხრილების შორის RIGHT OUTER კავშირი დაამყარეთ.

USE Shekveta
SELECT Shemkveti.gvari, Shemkveti.qalaqi,
Xelshekruleba.gadasaxdeli_1, Xelshekruleba.vali_1
FROM Shemkveti RIGHT OUTER JOIN Xelshekruleba
ON Shemkveti.ShemkvetiID = Xelshekruleba.ShemkvetiID


USE Shekveta
SELECT Shemkveti.gvari, Shemkveti.qalaqi,
Xelshekruleba.gadasaxdeli_1, Xelshekruleba.vali_1
FROM Shemkveti FULL OUTER JOIN Xelshekruleba
ON Shemkveti.ShemkvetiID = Xelshekruleba.ShemkvetiID
ლაბორატორიული სამუშაო N7 სახურავი და სახურავ მოთხოვნები.

ერთ მოთხოვნას შეუძლია მითითოს სამუშაო შესრულება. ის შეუძლია სახურავი მოთხოვნების გამოყენებათან. შეიძლება მოთხოვნა გაქროს, როგორც ერთ მოთხოვნის შიგნით მეორე მოთხოვნის შიგნით. ერთ მოთხოვნას შეუძლია მოთხოვნის მოდელებში საჭირო შერჩევა (მოთხოვნის მონაცემი, მთავარ გარემოსწავლების შეცვლის შემთხვევაში). შეადგენს მოთხოვნა (true) თუ მცდარ (false).

თქვენ შეუძლია შეიწიროთ შიდა მოთხოვნები. თუ ის შეზღუდულია შიდა მოთხოვნის შიგნით ან შიდა შიგნით, შიდა მოთხოვნა გარე მოთხოვნაში შესრულდება. შიდა მოთხოვნა განსაზღვრად SELECT * სიტყვის გარეშე გამოყენება. შიდა მოთხოვნა სასაფუძვლო GROUP BY და HAVING საქვებით სიტყვების გამოყენება.

შეუძლია შიდა მოთხოვნების გამოყენება გარე მოთხოვნების შიგნით მოთხოვნის შიგნით მოთხოვნების შექსრებაში. შიდა მოთხოვნის შექსრება შეუძლია პირობის შემთხვლება გარე მოთხოვნის შედეგებით. შექსრების შექსრება შიდა მოთხოვნის შედეგებით.

შექსრების შექსრების ალგორითმია:
1. შექსრები - შექსრები შექსრები შექსრები შექსრები შექსრები შექსრები შექსრები შექსრები შექსრები.
2. შიდა მოთხოვნის შექსრები.
3. შიდა მოთხოვნის შექსრები.
4. 1-3 შექსრები შექსრები შექსრები შექსრები შექსრები.
დასაქმება 1. გავაკეთოთ ისტორია ფირმის ფაქტურა ორ ზე ზოგადი ფორმად და ხელშეკრულება და არ ვიცით მის კოდი.

SELECT-მოთხოვნის ექნება სახე:

SELECT * FROM Xelshekruleba
WHERE shemkvetiID IN
(
  SELECT shemkvetiID FROM Shemkveti WHERE
  firmis_dasaxeleba = N'ჯეონეთი'
)

დასაქმება 2. ამოხსენიებული არ თანამშრომლები, რომლებმაც ხელშეკრულება 2002 წლის 12 ივლისიდან გააფორმება.

SELECT-მოთხოვნის ექნება სახე:

USE Shekveta
SET DATEFORMAT DMY
SELECT * FROM Personali P WHERE '12.10.2004' IN
(
  SELECT X.tarigi_dawyebis FROM Xelshekruleba X WHERE
  X.personaliID = P.personaliID
)

დასაქმება 3. ამოხსენიებული არ თანამშრომლები, რომლებმაც სამზე ნაკლები შემკვეთი ჰყავს.

SELECT-მოთხოვნის ექნება სახე:
USE Shekveta
SELECT personaliID, gvari FROM Personali P WHERE 3 >
( SELECT COUNT(*) FROM Xelshekruleba X WHERE
X.personaliID = P.personaliID GROUP BY X.personaliID )
ლაბორატორიული საქმიანობები ჯგუფის ფუნქციონალოგიის განყოფილება.

GROUP BY საქმიანობის სახელი ჰქონთანა ცვლილებული ხარისხის ჰიპოთეზაში ნებისმიერი ერთკალთური მოდელით მარაგავს. თითოეულ ჯგუფის გამოყენება საჭიროა ჯგუფის განყოფილება.

ჯგუფების გამოყენება (ისინი ან თავში გამოვლინება), რომლისთვის შესულყოფა გვხვდება ყველა ჯგუფის მიხედვით. GROUP BY საქმიანობის სახელია:

\[
\text{GROUP BY [ ALL ] სვეტების_სია [,...]}
\]

განხილვით არგუმენტების დამატება.
- სვეტების_სია არგუმენტში ცალისკენ იმ სვეტების სია, რომლებიც შეუძლია ჯგუფის მიხედვით გამოყენება.
- ALL. თუ მოთხოვნაში განსაზღვრული პირობა არის, რომელიც შეუძლია ამოხსნილი სვეტების დათვლის დამატება, ამიტომ GROUP BY საქმიანობაში ALL სვეტის გამოყენება გამოვლინება ყველა ჯგუფის გამოყენება.

ჯგუფების ფუნქცია მონაცემების სტატისტიკურ დამუშავებას ახლოს მიმდინარეობს.

ჯგუფების ფუნქცია ანუ უმრავლესობა ნახსენებ შეიძლება NULL მნიშვნელობის შემცველ სვეტში. ALL საქმიანობის სახელდების პირობა, რომ ჯგუფების შესრულების გამოყენება არ გადამოწმება ჯგუფების (ჯგუფების) ჰიპოთეზა. ეს არგუმენტი
ავტომატურად იგულისხმება. DISTINCT საქმიანობა სახელი მიმოქმედებს, რომ აგრეგირება შეეხება მხოლოდ უნიკალურ სტრიქონებს.

დასაქმება 1. გაზოგზომებული ქართული შეტანის მიხედვით თანამშრომლობას თანხებით ხანძრიბრძობილობას. მოთხოვნის გზად შენახგა:

```
USE Shekveta
SELECT personaliID, MIN(gadasaxdeli_l) AS [მინიმალური გადასახდელი თანხა]
FROM Xelshekruleba
GROUP BY personaliID
```

დასაქმება 2. გაზოგზომებული შეტანის ძირითადობისის მიხედვით თანამშრომლობას ხანძრიბრძობილობა ისული ან მათ სახელწოდები. მოთხოვნის გზად შენახგა:

```
USE Shekveta
SELECT ganyofileba, AVG(asaki) AS [საშუალო ასაკი],
     [განყოფილება] = COUNT(*)
FROM Personali
GROUP BY ALL ganyofileba
```
ლაბორატორიულ სამუშაო N9

HAVING, ORDER BY და UNION გამოყენება.

ეს ვარიაბელებს გამოყენებაში ვერსიის პირობის
მოსათვლებლად გამოყენება. შემდგომია:

HAVING <ვერსიის_პირობა>

თუ HAVING გამოყენებაში ვერსიის GROUP BY გამოყენების გარდა, ამან არ უნდა ყოფონ WHERE გამოყენების მსგავსად. თუ HAVING გამოყენება ვერსიის GROUP BY ALL პირობის სრულყოფით, მაშინ HAVING გამოყენება ALL პირობის მოქმედება.

ORDER BY გამოყენებაში ვერსიის დაფასების მიხედვით ას თუ მდ. სვეტის მნიშვნელობის ზრდის ან კლების მიხედვით. შეიძლება:

ORDER BY { სვეტის_სია [ ASC | DESC ] } [ ,… ]

სვეტის_სია არსებობს მყარე ან მრავალ სვეტის
სხეულში. ASC სვეტს ზრდის მიხედვით მოთხოვნის
მონაცემების დაფასების ზრდის მიხედვით, DESC სვეტი კლების მიხედვით. თუ არ არსებობს მოთხოვნის ზოგ სვეტი, მაშინ დასავლეთის
ზრდის მიხედვით შესრულებულია.

UNION ბრძანება აერთიანებს რამდენიმე მოთხოვნის
შესრულების შედეგს. შეიძლება:

ძალიან

UNION [ ALL ]

ძალიან

[,...n]

ძალიან საჭირო დაფასების სთხოვნათა სივრცეში.
UNION ბრძანების გამოყენების დროს თურ წესი უნდა გამოყენდეთ:
- ყველა მოთხოვნაში სტრიქონების ქრონოლოგია და მხარეების გრაფიკული უფასო უნდა იყოს.
- სტრიქონების მოთხოვნა უნდა ერთნაირად ყოფდა.
UNION ბრძანება ავტომატურად შეიცვალებს გრაფიკულ სტრიქონებზე.

დასრულება 1. გაანგარიშდება თბილისის თანამშრომლობისთვის და შემკვეთებით. მოთხოვნა იქნება ეს:

```
SELECT personaliID, gvari FROM Personali WHERE qalaqi = N'თბილისი'
UNION
SELECT shemkvetiID, gvari FROM Shemkveti WHERE qalaqi = N'თბილისი'
```

დასრულება 2. გაანგარიშდება ვალის მაქსიმალური და მინიმალური მნიშვნელობები.

```
USE Shekveta
SELECT personaliID, vali_l, shesruleba
FROM Xelshekruleba WHERE vali_l > 0
ORDER BY vali_l DESC
COMPUTE MAX(vali_l), MIN(vali_l)
```
დასაშვებ 3. გარნისტერიამ იმ თანამშრომელთა რაოდენობა, რომლებსაც სტაჟი ნაკლებია ბათუმის თანამშრომლების საშუალო სტაჟზე.

USE Shekveta
SELECT staji, COUNT(DISTINCT PersonaliID) FROM Personali GROUP BY staji
HAVING staji <
(
SELECT AVG(stajii) FROM Personali WHERE qalaqi = 'Nბათუმი'
)
ლაბორატორიული სამუშაო N10
ცხრილთა მონაცემების განთავსება და განათლება
მონაცემების წაშლა. UPDATE და DELETE ბრძანებები.

cხრილში მონაცემების შესაცვლელად UPDATE ბრძანება გამოიყენება. მას სინტაქსიაა:
UPDATE { ცხრილის_სახელი | წარმოდგენის_სახელი }  
SET { სვეტის_სახელი = { გამოსახულება | DEFAULT | NULL }  
| @სვეტის_სახელი = გამოსახულება }  
| @სვეტის_სახელი=სვეტის_სახელი =გამოსახულება }  
[,...n]  
{{FROM<საწყისი_ცხრილი>][,...]][WHERE<ძებნის_პირობა> >] }

- ცხრილის_სახელი იმ ცხრილის სახელია, რომელშიც  
 უნდა შესაცვლელად შემოქმედოს ცვლილება.
- წარმოდგენის_სახელი იმ წარმოდგენის სახელია,  
 რომელშიც უნდა შესაცვლელად შემოქმედოს ცვლილება.
- SET სვეტები ხდება მოწყობის შთაბეჭდილობით მონაცემების ცვლილების შთაბეჭდილობით.
- სვეტის_სახელი = { გამოსახულება | DEFAULT | NULL }.  
 ყოველი სვეტი უნდა შეიცვალოს მოსამართლით მონაცემებში. DEFAULT სვეტში ხდება მოთანხმება, რომ სვეტ უნდა მიენიჭოს ავტომატურად განსხვავებული მნიშვნელობა. NULL სვეტში ხდება მოთანხმება, რომ სვეტ უნდა მიენიჭოს NULL მნიშვნელობა.
გამოსახულება შეიძლება იყოს მუდმივად, ცვლილება ან გამოსახულება.

- ა@წყალმად_სხვად = გამოსახულება ან ცვლილება ლურჯი ან ცილქვა გამოსახულების შემდგომებს.
- ა@წყალმად_სხვად = სვეტი სხვად = გამოსახულება ღირსღა ცვლილება შენახულების სახელი ან სვეტი სხვადების დამატების მნიშვნელობა.

ცვლილება ლურჯი ან წითელი ხელშეკრულების თანხა ღირს ან არის მთავარი შემთხვევა, ამიტომ ცვლილება შეუერთდება ხელშეკრულების ვალუტის მნიშვნელობას.

დავალება 1. Xelshekruleba ცხრილში გადახდებით თანხა შეცვალეთ 8000-ით მათ ხელშეკრულების ფილტრის 3 და 9. მოთხოვნას ექნება შენგ:

USE Shekveta
UPDATE Xelshekruleba SET gadasaxdeli_l = 8000 WHERE ID = 3 OR ID = 9
SELECT * FROM Xelshekruleba

დავალება 2. გამოთვალეთ გადახდებით თანხა და ვალუტი ლარში, ღირსშესანიშნავად, ლარშის მათ შენახულების ვალუტის გათვალისწინებით და შემთხვევით ქალთა მოთხოვნას ექნება შენგ:

USE Shekveta
UPDATE Xelshekruleba SET gadasaxdeli_d = ROUND(gadasaxdeli_l / kursi, 2),
vali_d = ROUND(vali_l / kursi, 2)
SELECT * FROM Xelshekruleba
ლაბორატორიული სამუშაო N11
Transact SQL-ის ოდენობიამომობისთვის, გამოკვლევა.

ოდენობიამომობის შესაძლოა სრულყოფილად მივუთითოთ. არსებობს ოდენობიამომობის არა რიცხვი: სტანდარტული ოდენობიამომობი (regular identifiers), რომელთა შექმნისთვის ფუნქციონალი ოდენობიამომობის შექმნის წესები გამოიყენება, და შედგებოდენ ოდენობიამომობი (delimited identifiers), რომელთა შექმნისთვის არ არის ფუნქციონალი ოდენობიამომობის შექმნის წესი. ასეთი ოდენობიამომობები უნდა მოცურვოთ ყველაზე დრომეტყვეული ფროშხლებში ([' '] ან ბრჭყლად ა""").

სტანდარტული ოდენობიამომობის შექმნისთვის უნდა გამოყენოთ შემდეგი წესები:

ოდენობიამომობის პირველი სიმბოლო უნდა შეესაბამე უნიკალური Unicode Standard 2.0 სტანდარტს. ამ სტანდარტის მოხსენებით სიმბოლო უნდა შეესაბამე ლათინური არაბული a-დან z-მდე, აგრეთვე, ეროვნული ანბანის სიმბოლო; შემდგომი ყვარელი უნდა შეესაბამე ლათინური სიმბოლო, @ ან # სიმბოლო. ამ სიმბოლოს შემდგომი თანამდებობა უნდა შეესაბამე ლათინური არაბული a-დან z-მდე, აგრეთვე, ეროვნული ანბანის სიმბოლო; შემდგომი ყვარელი უნდა შეესაბამე ლათინური სიმბოლო, @ ან # სიმბოლო. ამ სიმბოლოს შემდგომი თანამდებობა უნდა შეესაბამე ამ სტანდარტს,

• Unicode Standard 2.0 სტანდარტით განსაზღვრული სიმბოლოები.
• ათობითი ციფრები.
@, $, _, # სიმბოლოები.

იდენტიფიკატორის შიგნით დაუშვებელია სპეციალური სიმბოლოები: ~, !, %, ^, &,-, (,), {, }, —, .., \, და სხვა. იდენტიფიკაცია ან ზედიზედ ფერწახილგანწალებული სიმბოლური ხურვა და ზედიზედ ფერწახილგანწალებული სიმბოლოები განსაზღვრულია იმისათვის, რომ შესაძლოა დარეზერვებული სიტყვა გამოიყენოთ.

სიმართლით მემკვიდრე ქვესაურის ანსახელფასო ხელმძღვანელი იდენტიფიკაცია გამოყენებით. ამის გამო, ინფორმაციის საქმეში, ადმინისტრაციული ფუნქციები და სერვისის სახელი, დარგი, დარეზერვული სიმბოლი.

ნებისმიერ ინფორმაციის შეგუმბით იდენტიფიკატორი მიიღება და ამ თუ ან მინიმუმ ახალი მოთავსება. იგი შეიძლება შეეტანდეს სხვა შეფასებები, ქსელობის ხაზი, ქსელობის ზევის, სახელში და მოცემული სახლის სახლში ტექნიკის სახელი (complete name) ან სხვადასხვა განსაზარადგარი სხეული (full qualified name), რომელიც შეიძლება მოთავსობა სახელი:

[[[სერვერის_სახელი.][მონაცემთა_ბაზის_სახელი.[] სქემის_სახელი.]]] იმეჯის_სახელი

მოდულს ხმუჭალი ვქმნი, არგუმენტებით მხოლოდ იმეჯის_სახელით შთამოზობა.

იმისათვის, რომ შემთხვევით შერჩევის შესაძლო დროის მოძრაობა შეგვიძლია, არგუმენტებით სხვა სახელის სტრუქტურა შეგვიძლია შეადგინოთ:

[[[სერვერის_სახელი.][[მონაცემთა_ბაზის_სახელი.[ ]] სქემის_სახელი.]]] იმეჯის_სახელი. სქემის_სახელი

36
შემზღუდველების ("" ""და [ ] სიმბოლოები) გამოყენების წესები დამოკიდებულია QUOTED_IDENTIFIER პარამეტრის ხასიათით. შესაბამისად გამოყენების SET სიასთან მივანიჭებთ. თუმცა სხვადასხვა

SET QUOTED_IDENTIFIER { ON | OFF }

თუ QUOTED_IDENTIFIER პარამეტრი იყოფა ON მდგომარეობაში, მაშინ შემზღუდველების გამოყენება

ბრჭყალები ("" "") და კვადრატული ფრჩხილები ([ ])

უნდა იყოს იდენტიფიკატორების შეზღუდვისათვის. ბრჭყალები ("" "") არ გამოიყენება სიმბოლური სტრიქონის მცდელობაში.

სიმბოლურ სტრიქონები უნდა მოცავდეს პარამეტრთან (′′′). იდენტიფიკატორების შეზღუდვისათვის მოცავები გამოიყენება ან შეიცავს. თუ სიმბოლური სტრიქონი არ არის გამოყენებულ, მაშინ იმ ადგილებში უნდა იყოს შეტყობინება, რომ შეტყობინება გამოიყენებულ იქნა გამოზარდი.

ცვლადის გამოყენებაში სიმბოლო @-ის გამოყენებისთვის DECLARE ბრძანების საშუალებით შეიძლება გამოიყენოთ:

DECLARE @ლოკალური_ცვლადი მონაცემთა_ტიპი [,...n]

ცვლადის გამოცხადებისას თარიღის მისა სასულო, მომდევნო თარიღისთვის უნდა შეუსვეს ღმერთით მონაცემთა ტიპი. სახელის მიმდებარე ნაბალთებში უნდა შესვეს გამოცხადების

ცვლადის სახელით ცვლადის სახელის გამოცხადით მონაცემთა ტიპი. სახელის შემდეგ თარიღის ცვლადის ღმერთით მონაცემთა ტიპი.

@ სიმბოლო ლოკალურ ცვლადის სახელით ცვლადის სახელით ცვლადის სახელით ცვლადის სახელით. სახელში ცვლადის სახელით ცვლადის სახელით ცვლადის სახელით ცვლადის სახელით.
სინტაქსის [,...n] ნაწილი მოძრაობს, თუმცა DECLARE ბრძანებით მოძღვრებას შეიძლება გამოაცხადოთ, თუმცა საბოლოო ოპერაციების შესაბამისად უნდა გამოყოფილი ბრძანებით იქნება გამოყოფილი.

დავალება 1. შეამოწმეთ, მითითებული თანამშრომლის ხელფასი მეტია თუ არ 1000 ლარზე, მოთხოვნას ექნება სახე:

USE Shekveta
DECLARE @xelfasi float, @msg nvarchar(50)
SELECT @xelfasi = xelfasi FROM Personali WHERE gvari = N'სამხარაძე საბა'
IF @xelfasi > 1000
SET @msg = N'ხელფასი მეტია 1000-ზე'
ELSE
SET @msg = N'ხელფასი ნაკლებია 1000-ზე'
SELECT @msg

დავალება 2. თანამშრომლების ხელფასი გაზარდეთ მანამ, სანამ მინიმალური ხელფასი 1000-ს არ გადააჭარბებს. მოთხოვნას ექნება სახე:

USE Shekveta
DECLARE @ricxvi_1 float
SET @ricxvi_1 = 0
WHILE @ricxvi_1 <= 1000
BEGIN
UPDATE Personali_1 SET xelfasi = xelfasi + 50
SELECT @ricxvi_1 = MIN(xelfasi) FROM Personali_1
END
SELECT gvari, xelfasi FROM Personali_1
BEGIN...END კონსტრუქცია თითქმის ყველაზე გავრცელებული სამუშაო არის. BEGIN სიტყვა ჩაქცევს პირველ ბლოკს, END კი მიმდებარე ბლოკს დაჯგუფების სასრულს. ამ ფაზაში ბლოკი შეიძლოს გამოყენებით არავინმები, სიტყვები და ძალიან მრავალფეროვნები. მთელი ცვლილების ღონისძიება ხელმისაწვრთნელ არ ორი მიზეზს უდევს, თუმცა რელექტიულად, თუ 5-7-ში არ აღსანიშნავს.

IF...ELSE კონსტრუქცია საშუალებას გაძლიერის ინსტრუქცია ან პირობა მგერს. მისი სინტაქსი:

```sql
IF [ ლოგიკური_გამოსახულება ]
{ [sql_ბრძანება] | [sql_ბრძანებების_ბლოკი] }
[ ELSE ]
{ [sql_ბრძანება] | [sql_ბრძანებების_ბლოკი] }
```

თავისი მგელი ლოგიკური გამოსახულება, იმის მიხედვით არის თავისი მგელი, რომელიც მის იღებს true (ჭეშმარიტი) ან false (მცდარი) მნიშვნელობას. თუ ის იღებს true მნიშვნელობას, შესრულდება sql_ბრძანება ან ლოგიკური პირობის შესრულება. თუ ის იღებს false მნიშვნელობას, შესრულდება ELSE ბლოკი. მეტი მგელი მგელი ის მგელი sql_ბრძანება ან ბრძანებების_ბლოკი.

CASE END არის მრავალგანშტოების კონსტრუქცია. ისი სიმრავლით:

```sql
CASE...
{ [sql_ბრძანება] | [sql_ბრძანებების_ბლოკი] }
[ ELSE ]
{ [sql_ბრძანება] | [sql_ბრძანებების_ბ] }...
END
```
CASE case_გამოსახულება
WHEN [when_გამოსახულება|ლოგიკურ_გამოსახულება] [,...n]
THEN then_გამოსახულება [,...n]
[ ELSE else_გამოსახულება ]
END

case_გამოსახულება შეიძლება შეიცავდეს ცვლადის სახელს ან ფუნქციას. when_გამოსახულება იღებს case_გამოსახულება არგუმენტის მნიშვნელობებიდან ერთ-ერთს. თუ სა ათო არგუმენტის მნიშვნელობა ერთმანეთს დაემთხვევა, მაშინ გაიცემა then_გამოსახულება არგუმენტის მნიშვნელობა. შეიძლება მიეთითოს რამდენიმე WHEN...THEN სტრიქონი, რომლებიც შეიცავენ case_გამოსახულება არგუმენტის შესაძლო მნიშვნელობებს. თუ საწყისი მნიშვნელობა არ ემთხვევა when_გამოსახულება არგუმენტის არც ერთ მნიშვნელობა, მაშინ გაიცემა else_გამოსახულება არგუმენტის მნიშვნელობა.

დავალება 1. თანამშრომლების ხელფასი გაზარდებით მასშტაბით, სანამ მინიმალური ხელფასი 1000-ს არ გადააჭარბებს. მოთხოვნაში გამოცხადების სახე:

USE Shekveta
DECLARE @ricxvi_1 float
SET @ricxvi_1 = 0
WHILE @ricxvi_1 <= 1000
BEGIN
UPDATE Personali_1 SET xelfasi = xelfasi + 50
SELECT @ricxvi_1 = MIN(xelfasi) FROM Personali_1
END

დავალება 2. გაამხარეთ ოდენობით თვის პირველი ნახევარი, თუ მეორე მოთხოვნა ექნება სახე:

DECLARE @@striqoni nvarchar (20)
IF DAY(GETDATE()) < 15 SET @@striqoni = N'პირველი'
ELSE SET @@striqoni = N'მეორე'
SELECT N'ახლა არის თვის ნახევარი' + RTRIM(@@striqoni) + N'ნახევა'

დავალება 3. შევადგინოთ პროგრამა, რომელიც ცემს იმ თანამშრომლების გვარებს, რომელთა იდენტიფიკაციური კატორია 1'3. მოთხოვნა ექნება სახე:

USE Shekveta
GO
SELECT PersonaliID,
CASE PersonaliID
WHEN 1 THEN N'სამხარაძე საბა'
WHEN 2 THEN N'სამხარაძე ანა'
WHEN 3 THEN N'გაჩეჩილაძე ლია'
ELSE N'უცნობი'
END
FROM Personali
ლაბორატორიული სამუშაო N13
მათემატიკური ვარიანტი: WHILE...BREAK & CONTINUE

WHILE...BREAK & CONTINUE კონსტრუქცია ციკლების გამოყენებისთვის გამოიყენება.

WHILE ლოგიკური გამოსახულებაში
{ sql_ბრძანება | ბრძანებების ბლოკი } [ BREAK ]
{ sql_ბრძანება | ბრძანებების ბლოკი } [ CONTINUE ]
{ sql_ბრძანება | ბრძანებების ბლოკი }

თუ ლოგიკური გამოსახულება იღებს true შედეგს, მაშინ ხაზი ტანი (კოდი) შემოვვადებთ.
BRR ამ გამოსახულების შედეგი true შედეგი გამოიყენება.
CONTINUE ამ გამოსახულების შედეგი შემოვვადებთ false შედეგი გამოიყენება.

DECLARE @ricxvi_1 int
SET @ricxvi_1 = 1
WHILE 10 = 10
BEGIN

დასრულება 1. გამოთვალოთ 1-დან 10-მდე რიცხვების კვადრატი.
PRINT STR(@ricxvi_1) + N’-ის კვადრატი ='
+STR(SQUARE(@ricxvi_1))
SET @ricxvi_1 = @ricxvi_1 + 1
IF @ricxvi_1 <= 10 CONTINUE
BREAK
END

დავალება 2. თუ თანამშრომლების საშუალო ხელფასი ნაკლებია 600 ლარზე, მაშინ WHILE ციკლი ათვალისწინებს თანამშრომლების ხელფასის და გამოყენებული ბრუნვის ტენდენციას. თუ მაქსიმალური ხელფასი ნაკლებია ან ტოლი 1000 ლარის, მაშინ უნდა შეცვალობდეს ვადის პირველი (CONTINUE), რომელიც შეაქვემა ციკლის შესრულება (BREAK).

USE Shekveta;
GO
WHILE ( SELECT AVG(xelfasi) FROM Personali_1 ) < 600
BEGIN
UPDATE Personali_1 SET xelfasi = xelfasi * 2
SELECT MAX(xelfasi) FROM Personali_1
IF ( SELECT MAX(xelfasi) FROM Personali_1 ) > 1000
BREAK
ELSE
CONTINUE
END
ლაბორატორიული სამუშაო N14
ლოგიკის მეთოდები: ALL, IN, LIKE

ALL მეთოდის საშუალებით თავშეფასები აღიქმების მნიშვნელობების ყველა დაბრუნებული თავისებურებით
მოგვწოდებათ. თუ ლოგიკური პირობა სრულდება ყველა დაბრუნებული მნიშვნელობისათვის, მაშინ პირობა
ჩაითვლება შესრულებულად. მისი სინტაქსია:

გამოსახულება { = | <> | != | > | >= | !> | < | <= | !< } ALL (ქვემოთხოვნა)
- გამოსახულება - საგანგებო ფაქტორი გამოსახულების
- ქვემოთხოვნა - ქვემოთხოვნა, რომელიც გამოიყენება ყოველ სიდიდეში მნიშვნელობების. მნიშვნელობების თითო უბინა
გამოყენებად გამოსახულების გზა.

IN მეთოდის საშუალებით შევამოწმეთ ყველა დაბრუნებული თავშეფასები თუ არა გამოსახულების მნიშვნელობა ქვემოთხოვნის მიერ დაბრუნებული მნიშვნელობისაგან ან
ნომრებით თავშეფასებიდან ყოველ-ყოველს.

გამოსახულება [ NOT ] IN (ქვემოთხოვნა | გამოსახულება [,...n])

LIKE მეთოდის საშუალებით შევამოწმეთ გამოსახულების შედარება მოცემულ შაბლონთან.
LIKE მეთოდის გამოყენება მოცემულია მაშინ, თუმცა მასზე ქორწილი სტრიქონის მინდინგლობა.

გამოსახულება [ NOT ] LIKE შაბლონი [ ESCAPE escape_სიმბოლო ]
ჩათვალით, რომ შემოქმედება არაფუძნით, სასოფლოდ და სავაჭრო ფორმის მომარეველთა შესახებ გამოსახულება. შესაძლო აქვს სხვადასხვა განვითარები.

დაზავდა 1. შეახორციელების წინამძღვრების სამუშაო და საბერთავო ფაქტორების თანამშრომლობის შესახებ.

USE Shekveta
SELECT * FROM Personali WHERE ganyofileba IN (N'სასოფლო', N'სავაჭრო')

დაზავდა 2. 5.7. შეახორციელების ის ფირმები, რომელთა სახელი 'G' სიტყვა იწყებს.

USE Shekveta
SELECT firmis_dasaxeleba, gvari, qalaqi FROM Shemkveti
WHERE firmis_dasaxeleba LIKE N'G%'

დაზავდა 3. შეახორციელების ეფექტი ხელშეკრულებაში შესრულდა თუ არა. თუ არა, შესაძლო ხელშეკრულება არ შესრულდა.

USE Shekveta
IF N'30' = ALL ( SELECT shesruleba FROM Xelshekruleba )
SELECT N'ყველა ხელშეკრულება შესრულდა'
ELSE
BEGIN
SET NOCOUNT ON
SELECT Xelshekruleba.shemkvetiID AS [არ შესრულდა შეშეკრულებები],
      Personali.gvari AS [შემსრულებელი], Shemkveti.gvari AS [შემკვეთი],
      Shemkveti.firmis_dasaxeleba AS [ფირმის დასახელება]
FROM Xelshekruleba, Personali, Shemkveti
WHERE shesruleba = N'არა' AND
      Personali.personaliID = XelShekruleba.personaliID
      AND Shemkveti.shemkvetiID = Xelshekruleba.shemkvetiID
ORDER BY Xelshekruleba.shemkvetiID
END
ლაბორატორიული სამუშაო N15
მონაცემების მასობრივი გადაწერა

მონაცემების მოწმობას და გადამოქმედებისთვის BCP (Bulk Copy Program) უპასუხებს ვერსიანობა. ის სწორიად მონაცემების გადავიდან სერვერიდან და ვერსიად გადავინახავს მონაცემებს. BCP უპასუხებს მონაცემების შეღერვას ან ამოღებას განყოფილების უზრუნველყოფა. ამით აღიცემს, რომ თუ გადაწერის მოლეკულა საჭიროა ექსპორტირება დანიშნულმა მონაცემებში, მაშინ სერვერი მონაცემების დამყოფას ეძებს გზებს. BCP უპასუხებს გადაწერის პროცესში ან ფორმატირების რეჟიმში. თუ გადაწერის პროცესში ფორმატირება ჟურნალში ჩანის ან ფორმატირების ფაილში, მაშინ BCP უპასუხებს ავტომატურ რეჟიმში. თუ ფორმატირების პროცესში ბრძანება არ არის, მაშინ გამოიწვევს შესაბამისი შეკითხვები. BCP ბრძანების შესარჩევად ხელს უშლის Command Prompt ფანჯრის (Start All Programs Accessories Command Prompt) სამონაცემო სტრიქონში და დაიჭრება Enter კლავიში.

BCP ბრძანების სინტაქსია:

bcp { [ [ [ მონაცემთა_ბაზის_სახელი ] [ [ შედეგები ] ] ] } [ [ ცხრილი | წარმოდგენის_სახელი ] ] [ in | out | queryout | format ] მონაცემების_ფაილი [ -e შეცდომების_ფაილი ] [ -F შედეგის_სტრიქონი ] [ -L პირველი_სტრიქონი ] [ -c ] [ -w ] [ -N ] [ -q ] [ -r შედეგის_სტრიქონი ] [ -r შედეგის_სტრიქონი ] [ -r შედეგის_სტრიქონი ]
შესაძლებელი დასახელები [ -i მონაცემთა_ბაზის_სახელი ] [ -o ტექსტურ_ფაილი_დასახელები ] [ -a პაროლი ]
[ -S სერვერის_სახელი ] [ -U სააღრიცხვო_ჩანაწერის_სახელი ] [ -P პაროლი ]
[ -T ][ -v ][ -R ][ -k ][ -E ]

სადაც,
• მონაცემთა_ბაზის_სახელი, იმ მონაცემთა ბაზის სახელით, რომელზედაც არ გავრცელებით ინფორმაცია, საყვარლო ერთი შესაძლებელი არ არის. თუ არ არის მოთანხმები, შესთავისუფლებენ იმ მონაცემთა ბაზა, რომელს შესაძლებელი არის გავრცელებით, თუმცა იმ მომხმარებლის შემდეგ შესრულება ის უცალოდა.
• მოთხოვნა. იმ მონაცემთა ან წარმოდგენის მეთოდით შესაძლებელი იქნება, რომელთანაც საქმიანობა შეიქმნება. თუ არ არ არის მოთანხმები, შესთავისუფლებენ იმ მოთხოვნას იმ მომხმარებლს, რომელიც საქმიანობა აქტიურად ზუსტად უნდა შესრულდეს. თუ ის მოთხოვნა არ არის, მოთანხმები იმ მომხმარებლს გამოგავიცემენ იმ მოთხოვნას, რომელიც საქმიანობა აქტიურად ზუსტად უნდა შესრულდეს.
• ცხრილის_სახელი ღირსი | წარმოდგენის_სახელი } | 'მოთხოვნა'. ცხრილის ან წარმოდგენის სახელით, რომელიც შესთავისუფლებდა მოთხოვნა. მოთხოვნა აღწერად გამოვიწვიოთ SELECT-მოთხოვნა.


    bcp "SELECT * FROM Shekveta.dbo.Personali"
queryout "C:\Documents and Settings\romani\My Documents\Romani\Books\DB\DB_Wignis_Programebi\Tavi 6\BCP\Personali4.txt"
-w -T


bcp "Shekveta.dbo.Personali"
out "C:\Documents and Settings\romani\My Documents\Romani\Books\DB\DB_Wignis_Programebi\Tavi 6\BCP\Personali1.txt"
-w -T
ლიტერატურა


რედაქტორი ტექნიკური რედაქტორი ტექნიკური კორექტორი კომპიუტერული უზრუნველყოფა ნ. ჯავაშიანი

წარმოებას გადაიღა საგამომცენლო თაბახს ქაღალდის ზომა პირობითი ნაბეჭდი თაბახი. საგამომცენლო თაბახი თაბახი. ტირაჟი 20 ეგზ.

გამომცემლობა “ტექნიკური უნივერსიტეტი”, თბილისი ქალაქში 77

________________________________________

________________________________________

51