

საქართველოს ტექნიკური უნივერსიტეტი

რომან სამხარაძე, ნინო ჯოჯუა, ლია გაჩეჩილაძე

მეთოდური მითითებები ლაბორატორიული
სამუშაოების შესასრულებლად საგანში:
პროექტი მონაცემთა ბაზებში

I ნაწილი

საქართველოს ტექნიკური უნივერსიტეტი

რომან სამხარაძე, ნინო ჯოჯუა, ლია გაჩეჩილაძე

მეთოდური მითითებები ლაბორატორიული
სამუშაოების შესასრულებლად საგანში:
პროექტი მონაცემთა ბაზებში

I ნაწილი

დამტკიცებულია
მეთოდურ მითითებად
სტუ-ის სარედაქციო
საგამომცემლო
საბჭოს მიერ

თბილისი - 2013

სასწავლო კურსის ათვისების შემდეგ სტუდენტი შეძლებს: საინფორმაციო და მიკროპროცესორული სისტემებისთვის პროგრამული უზრუნველყოფის შემუშავებასა და განხორციელებას .NET პლატფორმაზე; სერიული პორტების, ADO.NET კლასებისა და ობიექტების შესწავლას და მათ გამოყენებას აღნიშნული ტიპის სისტემების შესაქმნელად.

რეცენზენტები: სრული პროფესორი მედეა ანდლულაძე
ასოცირებული პროფესორი ზაურ ჯოჯუა

სარჩევი

თავი 1. მონაცემთა ბაზის სტრუქტურის შემუშავება. მისი ობიექტების განსაზღვრა.	6
თავი 2. ცხრილების დაპროექტება.....	7
თავი 3. ცხრილებისათვის ინდექსების განსაზღვრა.	11
თავი 4. INSERT ბრძანების ფორმირება ცხრილებში მონაცემების ჩასამატებლად.	12
თავი 5. UPDATE და DELETE ბრძანებების ფორმირება ცხრილებში მონაცემების შესაცვლელად.	13
თავი 6. SELECT ბრძანების ფორმირება ცხრილების დახარისხებისა და გაფილტვრისათვის.	14
თავი 7. ფუნქციების შემუშავება გამოთვლების შესასრულებლად.	15
თავი 8. შენახული პროცედურების შედგენა.	16
თავი 9. წარმოდგენების ფორმირება.....	17
თავი 10. ტრიგერების შემუშავება ცხრილებისათვის.	19
თავი 11. მონაცემთა ბაზის სარეზერვო ასლის შექმნა. სარეზერვო ასლიდან აღდგენა.	21

ამოცანები 23

ლიტერატურა 25

თავი 1. მონაცემთა ბაზის სტრუქტურის შემუშავება. მისი ობიექტების განსაზღვრა.

მონაცემთა ბაზას ვადგენთ გაზის კომპანიის გაყიდვების განყოფილებისთვის. განყოფილება ახდენს ფიზიკური პირების მიერ მოხმარებული გაზის აღრიცხვას. ბაზა შეცავს ორ ცხრილს.

- პირველი ცხრილი შეიცავს ინფორმაციას აბონენტის შესახებ (ნომერი, სახელი და გვარი, რაიონი, მისამართი, ტელეფონი, მრიცხველის ნომერი, დავალიანება, გადახდის თარიღი). ცხრილის სახელია abonenti.

- მეორე ცხრილი შეიცავს ინფორმაციას ქვითრის შესახებ (ნომერი, გაცემის თარიღი, მიმდინარე გადასახადი, დავალიანება, ავანსი, დაფარვის დრო). ცხრილის სახელია qviTari.

აღნიშნულ ცხრილებს ვათავსებთ company მონაცემთა ბაზაში.

თავი 2. ცხრილების დაპროექტება

ცხრილების სტრუქტურა შემდეგია:

ცხრილი 1. abonenti ცხრილის სტრუქტურა

abID int, primary key	
ab_nomeri	nvarchar(8)
gvari	nvarchar(30)
raioni	nvarchar(20)
misamarTi	nvarchar(50)
telefoni	nvarchar(8)
mricxvelis_nomeri	nvarchar(10)
davalianeba	float null

ცხრილი 1. qviTari ცხრილის სტრუქტურა

qvID int, primary key	
qv_nomeri	nvarchar(12)
ab_nomeri	nvarchar(8)
davalianeba	float null
zveli_chveneba	int
mricxvelis_chveneba	int

aRricxvis_TariRi	datetime
zveli_davalianeba	float null
gadasaxdelia as (axali_chveneba- zveli_chveneba)*0.5090+zveli_ davalianeba	

ცხრილი 1. abonenti ცხრილის შექმნა

ცხრილის პირველი abID სვეტი იქნება პირველადი გასაღები სვეტი-მთვლელი. მთვლელის საწყისი მნიშვნელობა და ნაზრდი 1-ის ტოლია;

davalianeba სვეტი შეიცავს ნაგულისხმევ მნიშვნელობას;

USE company

```
CREATE TABLE abonenti
```

```
(
```

```
abID int PRIMARY KEY IDENTITY (1,1),
```

```
ab_nomeri nvarchar(8),
```

```
gvari nvarchar(30),
```

```
raioni nvarchar(20),
```

```
misamarTi nvarchar(50),
```

```
telefoni nvarchar(8),
```

```
mricxvelis_nomeri nvarchar(10),
```

```
davalianeba float default null,
```

```
aRricxvis_TariRi datetime
```

```
)
```

ცხრილი 2. qviTari ცხრილის შექმნა

qv_nomeri სვეტი იქნება გარე გასაღები და დაუკავშირდება მთავარი abonenti ცხრილის abID პირველად გასაღებს.

```

USE company
CREATE TABLE qviTari
(
qvID int primary key,
qv_nomeri int NULL REFERENCES abonenti1(abID),
davalianeba float null,
zveli_chveneba int ,
mricxvelis_chveneba int ,
aRricxvis_TariRi datetime,
zveli_davalianeba float null,
gadasaxdelia as (mricxvelis_chveneba-zveli_chveneba)
*0.5090+zveli_davalianeba
)

```

თავი 3. ცხრილებისათვის ინდექსების განსაზღვრა.

დავალება: შექმნათ Ind_raioni_gvari უნიკალური, არაკლასტერული, შედგენილი ინდექსი raioni და gvari სვეტებისათვის. შევსების ფაქტორია 40%. raioni სვეტი დავალაგოთ კლებადობით, gvari სვეტი ზრდადობით

მოთხოვნას ექნება სახე:

USE company

GO

-- თუ ინდექსი არსებობს, მაშინ ის წაიშლება

```
IF EXISTS ( SELECT name FROM sys.indexes WHERE name =  
N'Ind_ raioni _gvari' )
```

```
DROP INDEX Ind_raioni_gvari ON abonenti
```

GO

-- ინდექსის შექმნა

```
CREATE UNIQUE NONCLUSTERED INDEX Ind_raioni_gvari  
ON abonenti ([raioni] DESC, [gvari] ASC) WITH FILLFACTOR =  
40
```

თავი 4. INSERT ბრძანების ფორმირება ცხრილებში მონაცემების ჩასამატებლად.

დავალება: company მონაცემთა ბაზის abonenti ცხრილს დავუმატოთ ერთი სტრიქონი. ჩასამატებელი მნიშვნელობების მიმდევრობა შეესაბამება სვეტების მიმდევრობას ცხრილში. შედეგი ეკრანზე გამოვიტანოთ.

მოთხოვნას ექნება სახე:

```
USE company
```

```
INSERT INTO abonenti VALUES (N'4512, N'სამხარაძე ბექა',  
N'ვაკე', N'კავსადის 4, ბ12', '899-012345', N'451268', 30.55)
```

```
SELECT * FROM abonenti
```

INSERT მოთხოვნაში არ არის მითითებული personaliID სვეტი, რადგან მის IDENTITY თვისებას Yes მნიშვნელობა აქვს მინიჭებული და სერვერი თვითონ შეასრულებს ამ სვეტის მნიშვნელობის გენერირებას.

**თავი 5. UPDATE და DELETE ბრძანებების ფორმირება
ცხრილებში მონაცემების შესაცვლელად.**

დავალება: qviTari ცხრილში ძველი დავალიანება სვეტის მნიშვნელობა 50%-ით შევამციროთ იმ აბონენტებისთვის, რომელთა დავალიანება აღემატება 100 ლარს

მოთხოვნას ექნება სახე:

```
USE company
```

```
UPDATE qviTari SET zveli_davalianeba = zveli_davalianeba/2  
WHERE zveli_davalianeba>100  
SELECT * FROM qviTari
```

დავალება: qviTari ცხრილი გადავწეროთ qvit_1 ცხრილში. qvit_1 ცხრილიდან წავშალოთ ის აბონენტები, რომლებშიც ძველი დავალიანება ნულის ტოლია

მოთხოვნას ექნება სახე:

```
USE company
```

```
SELECT * INTO qvit_1 FROM qviTari  
DELETE FROM qvit_1 WHERE zveli_davalianeba = 0  
SELECT * FROM qvit_1
```

თავი 6. SELECT ბრძანების ფორმირება ცხრილების დახარისხებისა და გაფილტვრისათვის.

დავალება: abonenti ცხრილში ინფორმაცია დაახარისხეთ გვარის ზრდის და დავალიანების კლების მიხედვით, შეასრულეთ მონაცემების ფილტრაცია იმ აბონენტების მიხედვით რომელთა ძველი დავალიანება აღემატება 50 ლარს.

მოთხოვნას ექნება სახე:

USE company

გო

```
select gvარი, davalianeბა from abonenti where davalianeბა>10  
order by gvარი, davalianeბა desc
```

თავი 7. ფუნქციების შემუშავება გამოთვლების შესასრულებლად.

დავალება: შექმენით ფუნქცია, რომელიც დავალიანების ჯამურ თანხას

მოთხოვნას ექნება სახე:

```
use company
go
CREATE FUNCTION Vali()
returns float
as
begin
declare @davalianeba int
set @davalianeba=
(
select sum(davalianeba) from abonenti
)
return @davalianeba
end

use company
SELECT dbo.Vali() as [სულ დავალიანება]
```

თავი 8. შენახული პროცედურების შედგენა.

დავალება: შექმენით შენახული პროცედურა, რომელიც გასცემს დავალიანების მქონე აბონენტების სიას მითითებული რაიონის მიხედვით

მოთხოვნას ექნება სახე:

```
use company
go
if OBJECT_ID('[dbo].[davalianebis_mqone]','P') is not null
drop procedure[dbo].[davalianebis_mqone];
go
create procedure davalianebis_mqone @raioni nvarchar (20)
as
select * from abonenti where raioni=@raioni and davalianeba>0
go
exec [dbo].[davalianebis_mqone ]N'ვაკე'
```

თავი 9. წარმოდგენების ფორმირება.

დავალეზა: შეექმნათ წარმოდგენა, რომელშიც გამოჩნდება abonenti ცხრილის gvari სვეტი და qviTari ცხრილის qalaqi სვეტი. abonenti ცხრილი არის მთავარი, qviTari ცხრილი - კი დამოკიდებული. გამოვიტანოთ ინფორმაცია იმ აბონენტების შესახებ, რომელთა დავალიანება 100 ლარზე მეტია. აბონენტების გვარები კლებადობით დავალაგოთ.

მოთხოვნას ექნება სახე:

```
USE company
```

```
GO
```

```
-- თუ წარმოდგენა არსებობს, მაშინ ის წაიშლება
```

```
IF OBJECT_ID('View_1', 'VIEW') IS NOT NULL
```

```
DROP VIEW View_1
```

```
GO
```

```
-- წარმოდგენის შექმნა
```

```
CREATE VIEW View_1
```

```
AS
```

```
SELECT TOP 100 PERCENT dbo.abonenti.gvari,
```

```
dbo.qviTari.zveli_davalianeba
```

```
FROM dbo.abonenti INNER JOIN
```

```
dbo.qviTari ON dbo.abonenti.ab_nomeri = dbo.qviTari.  
ab_nomeri  
WHERE dbo.qviTari.zveli_davalianeba >100  
ORDER BY dbo.abonenti.gvari DESC  
GO
```

--წარმოდგენის გახსნა:

```
SELECT * FROM View_1
```

თავი 10. ტრიგერების შემუშავება ცხრილებისათვის.

დავალბა: შევქმნათ abonenti_2 ცხრილი. მისთვის შევქმნათ UPDATE ტრიგერი, რომელიც გამოიტანს შეცვლილი სტრიქონების რაოდენობას და ინფორმაციას შეცვლის მცდელობის შესახებ.

მოთხოვნას ექნება სახე:

```
USE company  
GO
```

-- თუ ტრიგერი არსებობს, მაშინ ის წაიშლება

```
IF EXISTS ( SELECT name FROM sysobjects WHERE name = '  
abonenti_2_Update' AND type = 'TR' )  
DROP TRIGGER abonenti_2_Update  
GO
```

-- ტრიგერის შექმნა

```
CREATE TRIGGER abonenti_2_Update ON abonenti_2  
FOR UPDATE AS  
PRINT STR(@@ROWCOUNT) + N' სტრიქონის წაშლის  
მცდელობა abonenti_2 ცხრილიდან'  
PRINT N'მომხმარებელი - ' + CURRENT_USER  
IF CURRENT_USER <> 'dbo'  
BEGIN
```

```
PRINT N'შეცვლა აკრძალულია'  
ROLLBACK TRANSACTION  
END  
ELSE  
PRINT N'შეცვლა ნებადართულია'
```

ახლა abonenti_2 ცხრილიდან ერთი სტრიქონი წავშალოთ:

```
UPDATE abonenti_2 SET mricxvelis_nomeri = '1245'
```

თავი 11. მონაცემთა ბაზის სარეზერვო ასლის შექმნა. სარეზერვო ასლიდან აღდგენა.

სარეზერვო ასლი წარმოადგენს ერთ ან მეტ ფაილს, რომელშიც მონაცემთა ბაზაა მოთავსებული მთლიანად ან ნაწილობრივ. სარეზერვო ასლის შექმნა საჭიროა მონაცემების აღსადგენად კომპიუტერის, დისკის ან მონაცემების დაზიანების შემთხვევაში და ა.შ.

სარეზერვო ასლის შექმნისათვის სერვერზე შექმნილი ლოგიკური მოწყობილობების სია ინახება master მონაცემთა ბაზის sysdevices სისტემურ წარმოდგენაში. ახალი მოწყობილობის დასამატებლად გამოიყენება sp_addumpdevice სისტემური შენახული პროცედურა. მისი სინტაქსია:

```
sp_addumpdevice [ @devtype = ] 'მოწყობილობის_ტიპი' ,  
[ @logicalname = ] 'ლოგიკური_სახელი' , [ @physicalname =  
] 'ფიზიკური_სახელი'
```

სადაც,

- 'მოწყობილობის_ტიპი'. მოწყობილობის ტიპია და იღებს შემდეგ მნიშვნელობებს: TAPE (ლენტი), DISK (დისკი) და PIPE (სახელდებელი არხი);

- 'ლოგიკური_სახელი' მოწყობილობის ლოგიკური სახელია;

- 'ფიზიკური_სახელი' მოწყობილობის ფიზიკური სახელია.

დავალეზა: შექმნათ company მონაცემთა ბაზის სრული ასლი.

მოთხოვნას ექნება სახე:

ვიყენებთ შენახულ პროცედურას

USE company

-- თუ შენახული პროცედურა არსებობს, მაშინ ის

წაიშლება

IF OBJECT_ID ('sarezervo_asli', 'P') IS NOT NULL

DROP PROCEDURE sazezervo_asli;

GO

-- შენახული პროცედურის შექმნა

CREATE PROCEDURE sazezervo_asli

AS

EXEC sp_addumpdevice 'DISK', ' company_asli',

'C:\Program Files\Microsoft SQL

Server\MSSQL.1\MSSQL\Backup\ company_asli.bak'

BACKUP DATABASE company TO company _asli WITH

NAME = ' company _asli'

GO

ამოცანები

შეადგინეთ მონაცემთა ბაზა მეტეოროლოგიური სადგურისთვის, რომელიც შედგება ორი ცხრილისგან. პირველ ცხრილში მოათავსეთ ინფორმაცია მეტეოროლოგიური ქვესადგურების შესახებ, ხოლო მეორეში მათ მიერ მოწოდებული მონაცემების შესახებ. თითოეული ქვესადგურის მონაცემები შეინახეთ ცალკე ფაილში. შექმენით ბმა ამ ცხრილებს შორის. პირველ ცხრილში მონაცემები დაალაგეთ ანბანის მიხედვით, ხოლო მეორეში თარიღის მიხედვით. პირველ ცხრილში შეასრულეთ მონაცემების ფილტრაცია CO-ს მიხედვით. იპოვეთ მაქსიმალური ტენიანობა კვირის განმავლობაში.

შეადგინეთ მონაცემთა ბაზა პროდუქტის საწყობისთვის. ბაზა შედგება ორი ცხრილისგან, პირველი შეიცავს ინფორმაციას არსებული საქონლის შესახებ, ხოლო მეორე ინფორმაციას მომწოდებლების შესახებ. საქონლის აღრიცხვა ხდება ელექტრონულად და მონაცემები ინახება ცალკე ფაილში. შექმენით ბმა ამ ცხრილებს შორის. მონაცემები დაალაგეთ პირველ ცხრილში მოწოდების ინტენსიურობის მიხედვით,

ხოლო მეორეში მოწოდებული საქონლის რაოდენობის მიხედვით. პირველ ცხრილში მონაცემები გაფილტრეთ მიხედვით, ხოლო მეორეში ვარგისიანობის მიხედვით. გამოავლინეთ თვის ყვლაზე გაყიდვადი საქონელი.

შეადგინეთ მონაცემთა ბაზა სააღრიცხვო ფირმისთვის. ბაზას აქვს ორი ცხრილი, რომელთაგან პირველ ცხრილში ინახება ინფორმაცია პერსონალის შესახებ, ხოლო მეორეში მრიცხველების შესახებ. კონკრეტული პერსონალის მიერ აღრიცხული მონაცემები ინახება ფაილში. ანათვალის აღება ხდება სპეციალური აპარატის საშუალებით და ამავე აპარატის საშუალებით ხდება ქვითრის გამოწერა. შექმენით ბმა ამ ცხრილებს შორის. მონაცემები დაალაგეთ პირველ ცხრილში პერსონალის მიხედვით, ხოლო მეორეში ანათვლების მიხედვით. მეორე ცხრილში მონაცემები გაფილტრეთ კონკრეტული მისამართის (რაიონი) მიხედვით. გამოავლინეთ ანათვალის საშუალო მნიშვნელობა.

ლიტერატურა

1. SQL სერვერი. რომან სამხარაძე. საგამომცემლო სახლი „ტექნიკური უნივერსიტეტი“, თბილისი.2009.
ISBN 978-9941-14-190-4 .

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

2. V i s u a l C#.N E T. რომან სამხარაძე. საგამომცემლო სახლი „ტექნიკური უნივერსიტეტი“, თბილისი.2009. ISBN978-9941-14-593.

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

რედაქტორი

ტექნიკური რედაქტორი

კორექტორი

კომპიუტერული უზრუნველყოფა ნ. ჯოჯუასი

წარმოებას გადაეცა . ხელმოწერილია დასაბეჭდად .
. ქალაქის ზომა პირობითი ნაბეჭდი თაბახი .
სააღრიცხვო-საგამომცენლო თაბახი . ტირაჟი 20 ეგზ.

გამომცემლობა “ტექნიკური უნივერსიტეტი”, თბილისი
კოსტავას 77
