

## БАЗА ФАКУЛТЕТ

Креирати базу података Факултет у Microsoft SQL Server Management Studio-у. Потребно је да чува податке о студентима, професорима (наставницима), предметима као и о одслушаним и положеним испитима, смеру, локацији (или граду). Атрибуте одредити на основу задатих упита. Дефинисати све примарне као и секундарне кључеве.

### /\* УПИТИ

1. Формирати списак студената.
2. Формирати списак свих студената сортиран по презимену студента у растућем поретку. Уколико више студената има исто презиме податке сортирати додатно по имену студента у истом поретку.
3. Формирати списак студената који студирају на смеру Информационе технологије.
4. Приказати све личне информације/податке студената.
5. Приказати презимена и имена свих студената који станују у Зрењанину или Новом Саду.
6. Које предмете са одређеним фондом часова предају наставници (издвојити презиме и име наставника, назив предмета и недељни фонд часова за тај предмет).
7. Приказати које предмете на катедри за Информационе технологије предаје наставник Петар Хотовски.
8. Издвојити податке о студентима који су одслушали предмет Базе података 1.
9. Колико студената је слушало предмет Базе података 1.
10. Издвојити податке за уверење о положеним испитима студента Пејовић Предрага, брИндекса 150/98. Списак уредити хронолошки.
11. Израчунати просечне оцене свих студената.

\*/

```
/* КРЕИРАЊЕ БАЗЕ*/
CREATE DATABASE FAKULTET
USE FAKULTET

CREATE TABLE STUDENT(
    ucenikID INT PRIMARY KEY,
    Ime VARCHAR(30),
    Prezime VARCHAR(30),
    brIndeksa VARCHAR(30),
    gradID INT,
    smerID INT
);
CREATE TABLE SLUSA(
    slusaID INT PRIMARY KEY,
    brIndeksa VARCHAR(30),
    predmetID INT
);
CREATE TABLE POLOZENIISPITI(
    piID INT PRIMARY KEY,
    brIndeksa VARCHAR(30),
    ocena INT,
    predmetID INT
);
CREATE TABLE SMER(
    smerID INT PRIMARY KEY,
    Naziv VARCHAR(30),
);
CREATE TABLE GRAD(
    gradID INT,
    Naziv VARCHAR(30)
);
CREATE TABLE NASTAVNIK(
    nastavnikID INT PRIMARY KEY,
    Ime VARCHAR(30),
```

```
    Prezime VARCHAR(30),  
);  
CREATE TABLE PREDMET(  
    predmetID INT,  
    Naziv VARCHAR(30),  
    Fond INT,  
    nastavnikID INT,  
    smerID INT  
);
```

```
/*УБАЦИВАЊЕ ПОДАТАКА*/
```

```
INSERT INTO STUDENT VALUES  
(1, 'Ljubomir', 'Micic', '213/23', 18400, 3),  
(2, 'Predrag', 'Pejovic', '150/98', 1, 2)
```

```
INSERT INTO SLUSA VALUES  
(1, '213/23', 1),  
(2, '150/98', 3),  
(3, '150/98', 4),  
(4, '150/98', 5)
```

```
INSERT INTO POLOZENIIISPITI VALUES  
(1, '150/98', 10, 4),  
(2, '213/23', 9, 1)
```

```
INSERT INTO GRAD VALUES  
(1, 'Beograd'),  
(2, 'Novi Sad'),  
(3, 'Zrenjanin'),  
(18400, 'Prokuplje')
```

```
INSERT INTO SMER VALUES  
(1, 'Matematika'),  
(2, 'Fizika'),
```

```
(3, 'Informacione tehnologije')
```

```
INSERT INTO NASTAVNIK VALUES
```

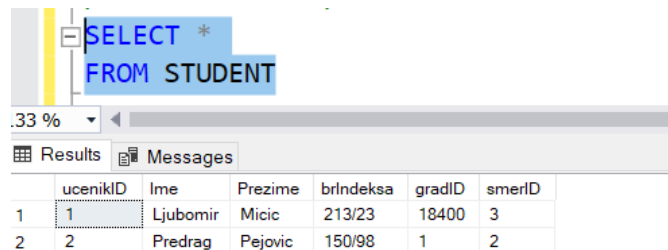
```
(1, 'Jovana', 'Savicevic'),  
(2, 'Petar', 'Hotovski')
```

```
INSERT INTO PREDMET VALUES
```

```
(1, 'Baze podataka 1', 111, 1, 3),  
(2, 'Programiranje 3', 123, 1, 3),  
(3, 'Inf. i net tehnologije', 134, 2, 3),  
(4, 'Računarska matematika', 124, 2, 3),  
(5, 'UI apk sa WEB dizajnom', 132, 2, 3)
```

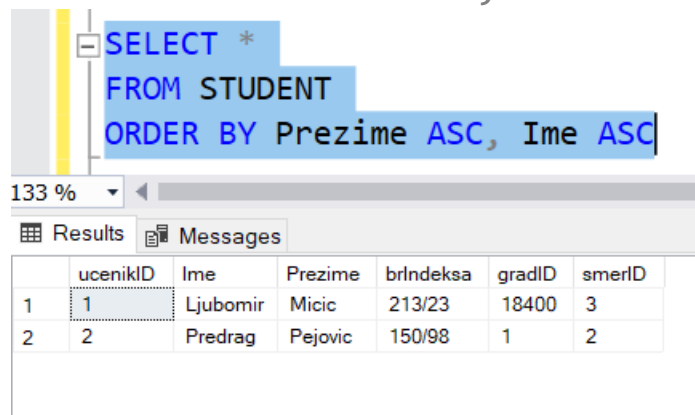
```
/*РЕШЕЊЕ УПИТА*/
```

```
SELECT * FROM STUDENT
```



	ucenikID	Ime	Prezime	brIndeksa	gradID	smerID
1	1	Ljubomir	Micic	213/23	18400	3
2	2	Predrag	Pejovic	150/98	1	2

```
SELECT *  
FROM STUDENT  
ORDER BY Prezime ASC, Ime ASC
```



	ucenikID	Ime	Prezime	brIndeksa	gradID	smerID
1	1	Ljubomir	Micic	213/23	18400	3
2	2	Predrag	Pejovic	150/98	1	2

```

SELECT *
FROM STUDENT
INNER JOIN SMER
ON STUDENT.smerID = SMER.smerID
WHERE SMER.Naziv = 'Informacione tehnologije'

```

133 %

Results Messages

	ucenikID	Ime	Prezime	brIndeksa	gradID	smerID	smerID	Naziv
1	1	Ljubomir	Micic	213/23	18400	3	3	Informacione tehnologije

```

SELECT *
FROM STUDENT
INNER JOIN GRAD
ON STUDENT.gradID = GRAD.gradID
INNER JOIN SMER
ON STUDENT.smerID = SMER.smerID

```

133 %

Results Messages

	ucenikID	Ime	Prezime	brIndeksa	gradID	smerID	gradID	Naziv	smerID	Naziv
1	2	Predrag	Pejovic	150/98	1	2	1	Beograd	2	Fizika
2	1	Ljubomir	Micic	213/23	18400	3	18400	Prokuplje	3	Informacione tehnologije

```
SELECT Ime, Prezime, brIndeksa, Naziv
FROM STUDENT, GRAD
WHERE (GRAD.Naziv = 'Novi Sad' OR GRAD.Naziv =
'Zrenjanin')
```

133 %

Results Messages

	Ime	Prezime	brIndeksa	Naziv
1	Ljubomir	Micic	213/23	Novi Sad
2	Predrag	Pejovic	150/98	Novi Sad
3	Ljubomir	Micic	213/23	Zrenjanin
4	Predrag	Pejovic	150/98	Zrenjanin

```
SELECT PREDMET.Naziv, NASTAVNIK.Ime,
NASTAVNIK.Prezime, PREDMET.Fond
FROM PREDMET
INNER JOIN NASTAVNIK
ON PREDMET.nastavnikID = NASTAVNIK.nastavnikID
```

133 %

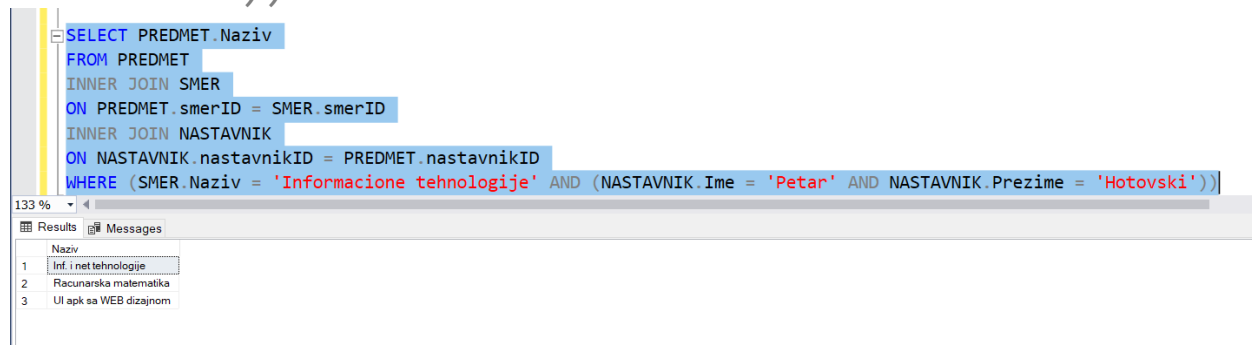
Results Messages

	Naziv	Ime	Prezime	Fond
1	Baze podataka 1	Jovana	Savicevic	111
2	Programiranje 3	Jovana	Savicevic	123
3	Inf. i net tehnologije	Petar	Hotovski	134
4	Racunarska matematika	Petar	Hotovski	124
5	UI apk sa WEB dizajnom	Petar	Hotovski	132

```

SELECT PREDMET.Naziv
FROM PREDMET
INNER JOIN SMER
ON PREDMET.smerID = SMER.smerID
INNER JOIN NASTAVNIK
ON NASTAVNIK.nastavnikID = PREDMET.nastavnikID
WHERE (SMER.Naziv = 'Informacione tehnologije' AND
(NASTAVNIK.Ime = 'Petar' AND NASTAVNIK.Prezime =
'Hotovski'))

```



```

SELECT STUDENT.Ime, STUDENT.Prezime
FROM STUDENT
INNER JOIN SLUSA
ON STUDENT.brIndeksa = SLUSA.brIndeksa
WHERE SLUSA.predmetID = 1

```

```

SELECT STUDENT.Ime,
COUNT (SLUSA.brIndeksa) AS [Broj Studenta]
FROM STUDENT
INNER JOIN SLUSA
ON STUDENT.brIndeksa = SLUSA.brIndeksa
WHERE SLUSA.predmetID = 1
GROUP BY STUDENT.Ime

```

```

SELECT STUDENT.Ime,
COUNT (SLUSA.brIndeksa) AS [Broj Studenta]
FROM STUDENT
INNER JOIN SLUSA
ON STUDENT.brIndeksa = SLUSA.brIndeksa
WHERE SLUSA.predmetID = 1
GROUP BY STUDENT.Ime

```

133 %

Results Messages

	Ime	Broj Studenta
1	Ljubomir	1

```

SELECT PREDMET.Naziv
FROM PREDMET
INNER JOIN POLOZENIIISPITI
ON PREDMET.predmetID = POLOZENIIISPITI.predmetID
INNER JOIN STUDENT
ON POLOZENIIISPITI.brIndeksa = STUDENT.brIndeksa
WHERE (STUDENT.Ime = 'Predrag' AND STUDENT.Prezime =
'Pejovic')
ORDER BY PREDMET.Naziv ASC

```

```

SELECT PREDMET.Naziv
FROM PREDMET
INNER JOIN POLOZENIIISPITI
ON PREDMET.predmetID = POLOZENIIISPITI.predmetID
INNER JOIN STUDENT
ON POLOZENIIISPITI.brIndeksa = STUDENT.brIndeksa
WHERE (STUDENT.Ime = 'Predrag' AND STUDENT.Prezime = 'Pejovic')
ORDER BY PREDMET.Naziv ASC

```

133 %

Results Messages

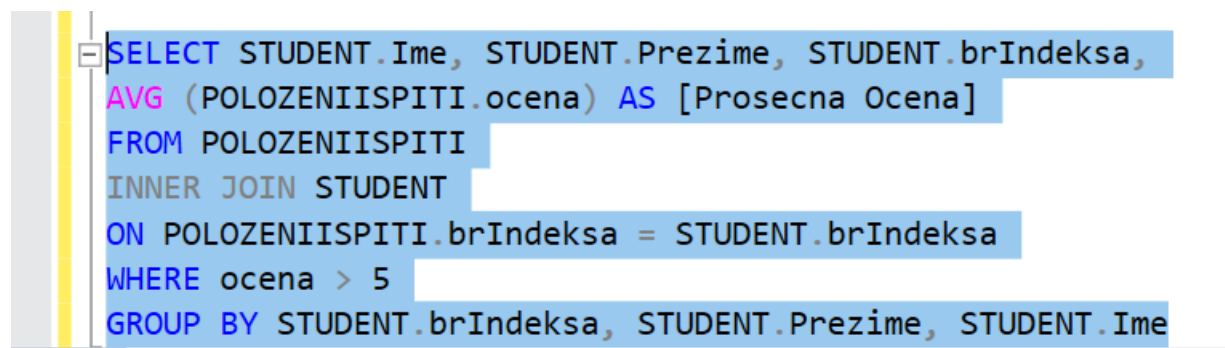
	Naziv
1	Racunarska matematika



```

SELECT STUDENT.Ime, STUDENT.Prezime,
STUDENT.brIndeksa,
AVG (POLOZENIIISPITI.ocena) AS [Prosecna Ocena]
FROM POLOZENIIISPITI
INNER JOIN STUDENT
ON POLOZENIIISPITI.brIndeksa = STUDENT.brIndeksa
WHERE ocena > 5
GROUP BY STUDENT.brIndeksa, STUDENT.Prezime,
STUDENT.Ime

```



133 %

Results Messages

	Ime	Prezime	brIndeksa	Prosecna Ocena
1	Predrag	Pejovic	150/98	10
2	Ljubomir	Micic	213/23	9

**/\*ПОДУПИТИ\*/** -само пример уколико би постојала плата унутар табеле НАСТАВНИК.

Који наставник/ца има највећу плату?

**И начин**

```

SELECT Prezime, Ime, Zvanje, plataMesec
FROM NASTAVNIK
WHERE plataMesec >= (SELECT max(plataMesec)

```

**FROM NASTAVNIK)**

## **II начин**

**SELECT Prezime, Ime, Zvanje, plataMesec**

**FROM NASTAVNIK**

**WHERE plataMesec >= ALL (SELECT (plataMesec)**

**FROM NASTAVNIK)**

**/\*ПОГЛЕД - VIEW\*/** -само пример уколико би постојала плата унутар табеле НАСТАВНИК.

Написати SQL View за приказ података о наставницима који имају највећу плату, сортиран по звањима наставника. Такође, написати упите, за промену, позив и брисање погледа.

**/\*Креирање погледа\*/**

**GO**

**CREATE VIEW [Spisak nastavnika sa najvecom platom]**

**AS SELECT Prezime, Ime, plataMesec, Zvanje**

**FROM NASTAVNIK**

**WHERE plataMesec >= (SELECT max (plata Mesec)**

**FROM NASTAVNIK)**

**GO**

**/\*Измена погледа\*/**

**ALTER VIEW [Spisak nastavnika sa najvecom platom]**

**AS SELECT Prezime, Ime, plataMesec**

**FROM NASTAVNIK**

**WHERE plataMesec >= (SELECT max (plata Mesec)**

**FROM NASTAVNIK)**

**/\*Позив/приказ\*/**

```
SELECT * FROM [Spisak nastavnika sa najvecom platom]  
ORDER BY plataMesec
```

**/\*Брисање погледа\*/**

```
DROP VIEW [Spisak nastavnika sa najvecom platom]
```

**/\*УСКЛАДИШТЕНЕ ПРОЦЕДУРЕ\*/** само пример уколико би постојала плата унутар табеле НАСТАВНИК.

Креирати ускладиштену процедуру за приказ наставника који имају већу месечну плату од неког наставника чији се подаци прослеђују процедури као параметар, написати команду за извршење процедуре.

**/\*Креирање процедуре\*/**

**GO**

```
CREATE PROCEDURE [Spisak nastavnika]
```

```
@Prezime NVARCHAR(30),
```

```
@Ime NVARCHAR(30)
```

**AS**

```
SELECT Prezime, Ime, plataMesec, Zvanje
```

```
FROM NASTAVNIK
```

```
WHERE plataMesec >= (SELECT plata Mesec,
```

```
FROM NASTAVNIK
```

```
WHERE Prezime=@Prezime
```

```
AND Ime=@Ime);
```

**GO**

**/\*Код за извршење\*/**

```
EXECUTE [Spisak nastavnika] 'Vukovic', 'Dusa'
```

# БАЗА ФАКУЛТЕТ - код

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The left pane shows the 'Object Explorer' with the 'FAKULTET' database selected under the 'DESKTOP-CSC7503' server. The right pane shows the 'Query Editor' with the following SQL script:

```
-- CREATE DATABASE FAKULTET
USE FAKULTET

--CREATE TABLE STUDENT(
ucenikID INT PRIMARY KEY,
Ime VARCHAR(30),
Prezime VARCHAR(30),
brIndeksa VARCHAR(30),
gradID INT,
smerID INT
);

--CREATE TABLE SLUSA(
slusaID INT PRIMARY KEY,
brIndeksa VARCHAR(30),
predmetID INT
);

--CREATE TABLE POLOZENIISPITI(
piID INT PRIMARY KEY,
brIndeksa VARCHAR(30),
ocena INT,
predmetID INT
);

--CREATE TABLE SMER(
smerID INT PRIMARY KEY,
Naziv VARCHAR(30),
);

--CREATE TABLE GRAD(
gradID INT,
Naziv VARCHAR(30)
);

--CREATE TABLE NASTAVNIK(
nastavnikID INT PRIMARY KEY,
Ime VARCHAR(30),
Prezime VARCHAR(30),
);

--CREATE TABLE PREDMET(
predmetID INT,
Naziv VARCHAR(30),
Fond INT,
nastavnikID INT
);
```

The bottom status bar indicates 'Query executed successfully.' and '0 rows'.

# DESIGN – табела СТУДЕНТ

DESKTOP-CSC7503.FAKULTET - dbo.STUDENT - Microsoft SQL Server Enterprise Manager

Object Explorer

- DESKTOP-CSC7503 (SQL Server 15.0.2080)
- Databases
  - System Databases
  - Database Snapshots
  - FAKULTET
    - Database Diagrams
    - Tables
      - System Tables
      - FileTables
      - External Tables
      - Graph Tables
      - dbo.GRAD
      - dbo.NASTAVNIK
      - dbo.POLOZENISPITI
      - dbo.PREDMET
      - dbo.SLUSA
      - dbo.SMER
      - dbo.STUDENT
    - Views
    - External Resources
    - Synonyms
    - Programmability
    - Service Broker
    - Storage
    - Security
  - Security
  - Server Objects
  - Replication
  - PolyBase
  - Management
  - XEvent Profiler

Table Designer: FAKULTET.sql - DES...SC7503Uovana (53)

Column Name	Data Type	Allow Nulls
ucenikID	int	<input type="checkbox"/>
Ime	varchar(30)	<input checked="" type="checkbox"/>
Prezime	varchar(30)	<input checked="" type="checkbox"/>
brIndeksa	varchar(30)	<input checked="" type="checkbox"/>
gradID	int	<input checked="" type="checkbox"/>
smerID	int	<input checked="" type="checkbox"/>

Column Properties

(General)

Property	Value
(Name)	ucenikID
Allow Nulls	No
Data Type	int
Default Value or Binding	

Table Designer

(General)

Ready

Откуцајте овде да бисте претражили

65°F Sunny 19:23 14.04.2022.