

JEDNOSTAVNI SELECT UPITI

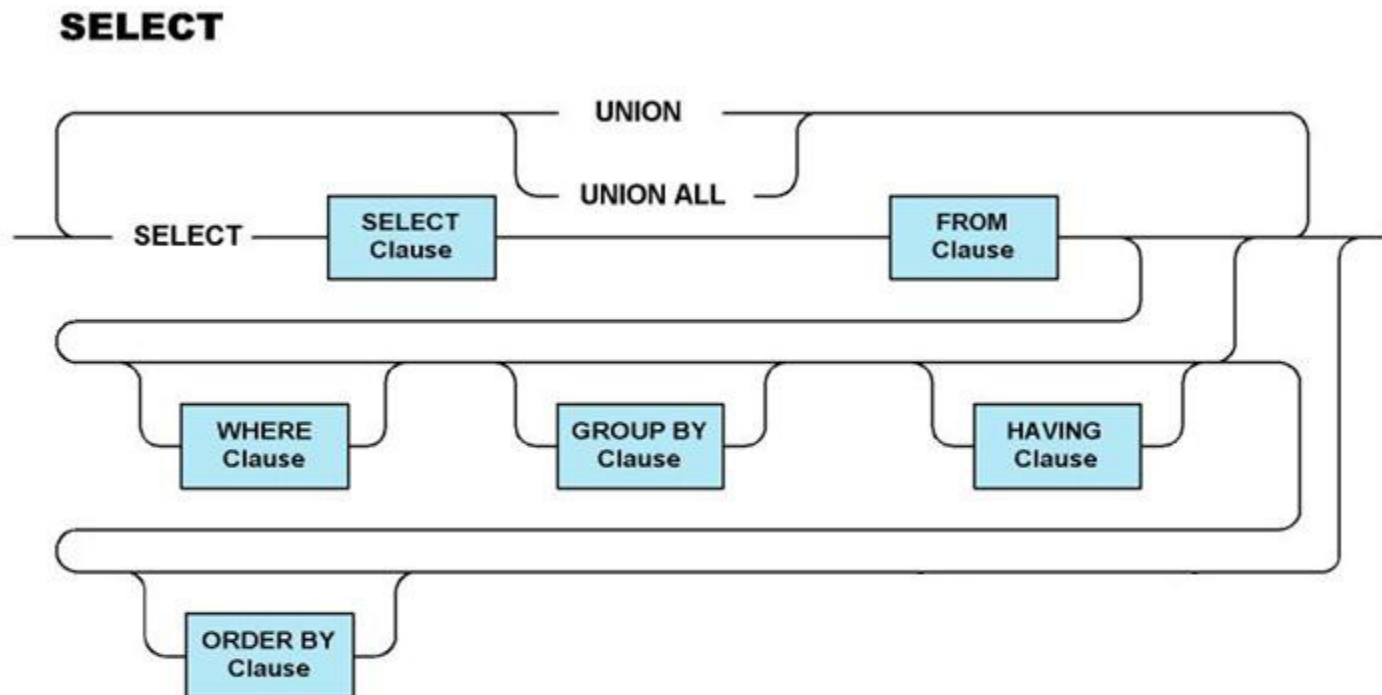
II auditorne vježbe

Teme

- ▶ Jednostavni upiti `SELECT` – `FROM` ...
- ▶ Izraz (Expression)
 - ▶ Atributi, konstante, aritmetički i znakovni operatori, funkcije
 - ▶ Datumske funkcije u SQL-u
- ▶ Definisanje uslova dohvata
 - ▶ Uslovi za dohvat i uslovi poređenja
 - ▶ Operatori `LIKE` i `RLIKE`



Sintaksa i način upotrebe naredbe SELECT

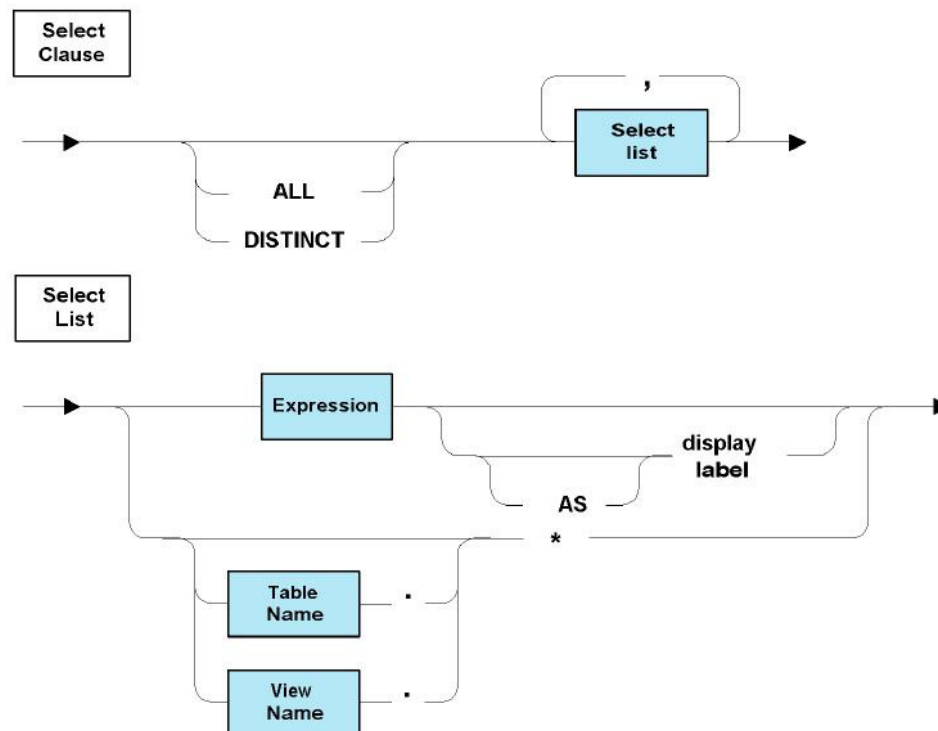


- ▶ Prema prikazanom sintaksnom dijagramu svi dijelovi osim *SELECT Clause* (*SELECT klauzule*) i *FROM Clause* (*FROM klauzule*) su opcionalni
 - ▶ Minimalni oblik **SELECT** naredbe sadrži *SELECT Clause* i *FROM Clause*
-



Sintaksa i način upotrebe naredbe SELECT

- ▶ *SELECT Clause* sadrži:
 - ▶ *Projection Clause* - opcionalni kvalifikator ALL ili DISTINCT
 - ▶ *SELECT List* (listu za selekciju) - listu atributa ili izraza čije se vrijednosti trebaju pojaviti u izlaznoj listi rezultata
- ▶ U *FROM Clause* navodi se ime relacije iz koje se dohvaćaju podaci



Lista za selekciju (*SELECT List*)

- ▶ **Primjer:** Ispis matičnih brojeva, imena i prezimena svih studenata (iz relacije *stud*)

```
SELECT mbrStud, imeStud, prezStud FROM stud
```

mbrStud	imeStud	prezStud
1120	Zdenko	Kolac
1121	Danijel	Marović
1123	Jozefina	Antončić
1124	Tihomir	Crnković
...

- ▶ Ispred imena atributa uvijek se **može** navesti ime relacije, ali se ime relacije **mora** navoditi ispred imena atributa jedino u slučajevima kad ime atributa nije jednoznačno određeno unutar liste za selekciju

```
SELECT stud.mbrStud, stud.imeStud, prezStud  
FROM stud
```



Lista za selekciju (*SELECT List*)

- ▶ *SELECT List* može sadržavati i oznaku '*' ili **imeRelacije.*** što znači da će se u izlaznoj listi pojaviti vrijednosti svih atributa iz relacije

`SELECT * FROM stud` ili `SELECT stud.* FROM stud` ili

`SELECT mbrStud, imeStud, prezStud, pbrRod,
pbrStan, datRodStud, jmbgStud FROM stud`

mbrStud	imeStud	prezStud	pbrRod	pbrStan	datRodStud	jmbgStud
1120	Zdenko	Kolac	31000	40000	1983-06-09	0906983330186
1121	Danijel	Marović	21000	10000	1982-12-21	2112982330118
1123	Jozefina	Antončić	10000	10000	1982-01-23	2301982330133
1124	Tihomir	Crnković	34000	10000	1983-05-04	0405983303228
...



Alias ime za atribut ili izraz

- ▶ *Alias* ime – u rezultatu se pojavljuje kao naziv atributa ili izraza
- ▶ Ključna riječ **AS** je opcionalna
- ▶ Ne smije se koristiti unutar upita osim u **ORDER BY Clause**

```
SELECT mbrStud MATICNI_BROJ,  
       imeStud AS IME,  
       prezStud AS PREZIME  
FROM stud
```

MATICNI_BROJ	IME	PREZIME
1120	Zdenko	Kolac
1121	Danijel	Marović
1123	Jozefina	Antončić
1124	Tihomir	Crnković
...



Operacija projekcije - kvalifikator DISTINCT

- ▶ **DISTINCT** – u rezultatu nema višestrukih pojava istih vrijednosti n-torki (eliminišu se duplikati)
 - ▶ Implementacija operacije projekcije u SQL-u
- ▶ **Primjer:** Lista različitih imena studenata (svako ime se u listi treba pojaviti samo jednom, bez obzira koliko studenata ima jednako ime)

```
SELECT DISTINCT imeStud FROM stud
```

imeStud
Zdenko
Danijel
Jozefina
Tihomir
...

- ▶ **ALL** – suprotno značenje od **DISTINCT**, ali se može izostaviti jer se primjenjuje po definiciji (*by default*)



Izraz (*Expression*)

- ▶ Lista za selekciju se gradi iz izraza (*Expression*)
- ▶ Izrazi se mogu formirati iz naziva atributa, konstanti, aritmetičkih operatora i različitih funkcija
- ▶ Redoslijed obavljanja operacija je uobičajen kao u ostalim jezicima, a izrazi se mogu grupisati pomoću zagrada
- ▶ Binarni operatori za sabiranje, oduzimanje, množenje i dijeljenje, te unarni operatori + i – odgovaraju istim takvim operatorima u drugim programskim jezicima
- ▶ *Pipe sign* (||) – operator za spajanje niza znakova
 - ▶ mora se uključiti opcija PIPES_AS_CONCAT u MySQL
- ▶ Navodnik se u niz znakova može ubaciti tako da se dva navodnika pišu jedan iza drugog, npr. 'O' 'Shaughnessy'



Primjeri upita sa izrazima u listi za selekciju

- ▶ **Primjer:** Ispis prezimena i imena nastavnika i iznosa plaće, ako se plaća nastavnika dobije množenjem broja 800 s koeficijentom koji je prije toga uvećan za 0.4

```
SELECT preznastavnik, imenastavnik, 800 * ( koef +  
0.4 ) as Plata  
FROM nastavnik;
```

- ▶ **Primjer:** Ispis matičnog broja, prezimena i imena studenata. Prezime i ime studenta ispisati u jednoj koloni pod nazivom STUDENT.

```
SELECT mbrstud, CONCAT (prezstud, ' ', imestud) AS  
STUDENT  
FROM stud;
```



Funkcijski izrazi

- ▶ Datumske
- ▶ Algebarske – ABS, MOD, POW, ROUND, SQRT, TRUNCATE
- ▶ Eksponencijalne i logaritamske – EXP, LOG, LOG2, LOG10
- ▶ Trigonometrijske – COS, SIN, TAN, ASIN, ACOS, ATAN, ATAN2
- ▶ Funkcija HEX za zadani cjelobrojni izraz vraća njegovu heksadecimalnu notaciju
- ▶ Funkcije za rad sa nizovima znakova (stringovima)
- ▶ Agregatne funkcije



Datumske funkcije u MySQL-u

- ▶ ADDDATE, DATE_ADD – dodaje vremenske vrijednosti (intervale) na datumsku vrijednost
 - ▶ CURDATE, CURRENT_DATE – vraća trenutni datum
 - ▶ DATE_SUB, SUBDATE – oduzima vrijednost (interval) od datuma
 - ▶ DATE – izdvaja dio datuma iz date ili datetime izraza
 - ▶ DATEDIFF – oduzima dva datuma
 - ▶ DAY, DAYOFMONTH – redni broj dana u mjesecu za zadani datum
 - ▶ DAYNAME – vraća naziv dana u sedmici
 - ▶ DAYOFYEAR – redni broj dana u godini za zadani datum (1 – 366)
 - ▶ EXTRACT – izdvaja dio datuma
 - ▶ FROM_DAYS – konvertuje broj dana u datum
 - ▶ MAKEDATE – kreira datum iz godine i rednog broja dana u godini
 - ▶ MONTH – za zadani datum vraća redni broj mjeseca
 - ▶ MONTHNAME – vraća naziv mjeseca
 - ▶ WEEK – vraća redni broj sedmice
 - ▶ WEEKDAY – indeks dana u sedmici za zadani datum (0 – ponedjeljak, 1 – utorak, itd.)
 - ▶ YEAR – redni broj godine za zadani datum
-



Funkcije za rad sa stringovima

- ▶ `CHAR_LENGTH(str)` – dužina stringa (broj karaktera)
 - ▶ `CONCAT(string1, string2 [, ...])` – spajanje stringova
 - ▶ `INSERT(str, pos, len, newstr)` – u stringu str mijenja len karaktera od pozicije pos sa novim stringom newstr
 - ▶ `LEFT(str, len)`, `RIGHT(str, len)` – vraća len karaktera s početka (s kraja) stringa str
 - ▶ `LENGTH(str)` – vraća dužinu stringa u bajtima
 - ▶ `LOCATE(substr, str, [pos])` – vraća poziciju prvog pojavljivanja substr u str počevši od pozicije pos (ukoliko se navede)
 - ▶ `LOWER(str)`, `UPPER(str)` – vraća string u kojem su sva slova mala, odnosno velika, respektivno
 - ▶ `LTRIM(str)`, `RTRIM(str)` – uklanja vodeća, odnosno prateća, prazna mjesta iz stringa, respektivno
 - ▶ `REPLACE(str, from_str, to_str)` – u stringu str zamjenjuje sva pojavljivanja from_str sa to_str
 - ▶ `REVERSE(str)` – obrće redoslijed karaktera u stringu
 - ▶ `SUBSTRING(str, pos, [len])` – iz stringa str vraća podstring od pozicije pos dužine len ili do kraja str ukoliko se len ne navede
 - ▶ `TRIM([{ BOTH | LEADING | TRAILING } [remstr] FROM] str)` – iz stringa str uklanja vodeća i/ili prateća pojavljivanja remstr
-



Primjeri sa datumskim funkcijama u listi za selekciju

- ▶ **Primjer:** Ispis imena i prezimena studenta, datuma rođenja, današnjeg datuma te broja dana koji su protekli nakon rođenja.

```
SELECT imestud, prezstud, datrodstud, CURDATE(),  
DATEDIFF(CURRENT_DATE, datrodstud) FROM stud;
```

- ▶ **Primjer:** Za svakog studenta ispisati ime, prezime, datum rođenja i datum na koji će navršiti 25 godina.

```
SELECT imestud, prezstud, datrodstud,  
DATE_ADD(datrodstud, INTERVAL 25 YEAR) FROM stud;
```

ili

```
SELECT imestud, prezstud, datrodstud,  
ADDDATE(datrodstud, INTERVAL 25 YEAR) FROM stud;
```



Primjeri sa datumskim funkcijama u listi za selekciju

- ▶ **Primjer:** Ispis imena i prezimena studenta, datuma rođenja, nazive dana u sedmici i mjeseca kada je rođen.

```
SELECT imestud, prezstud, datrodstud,  
DAYNAME(datRodStud), MONTHNAME(datrodstud) FROM  
stud;
```

- ▶ **Primjer:** Ispis imena i prezimena studenta, datuma rođenja i broja dana koji je protekao od rođenja studenta do datuma 01.01.2000.

```
SELECT imestud, prezstud, datrodstud,  
DATEDIFF(STR_TO_DATE('01/01/2000', '%d/%m/%Y'),  
datRodStud) FROM stud;
```



Primjeri sa funkcijama za rad sa stringovima u listi za selekciju

- ▶ **Primjer:** Ispisati nazive mjesta na način da se ispred naziva doda riječ 'GRAD-'

```
SELECT CONCAT('GRAD-', nazMjesto) FROM mjesto;
```

- ▶ **Primjer:** Ispisati nazive mjesta velikim slovima i dužine naziva mjesta.

```
SELECT UPPER(nazMjesto), CHAR_LENGTH(nazMjesto)  
FROM mjesto;
```

- ▶ **Primjer:** Ispisati inicijale studenta i prezime studenta na način da se svako pojavljivanje 'ić' zamijeni sa 'ich'.

```
SELECT CONCAT(SUBSTRING(imeStud, 1, 1), ' .',  
              SUBSTRING(prezStud, 1, 1), ' .'),  
       REPLACE(prezStud, 'ić', 'ich') FROM stud;
```



Definisanje uslova dohvata

- ▶ Uslov za dohvat (*Condition*) je izreka izgrađena od jednog ili više uslova usporedbe (*Comparison Condition*) koji se međusobno povezuju logičkim operatorima AND, OR i NOT
- ▶ Postoji šest različitih oblika uslova usporedbe:
 - I. Usporedba izraza sa drugim izrazom pomoću relacijskog operatora poređenja

Primjer: Ispisati sve podatke o nastavnicima čiji je koeficijent za plaću veći ili jednak od 5.50

```
SELECT * FROM nastavnik WHERE koef >= 5.50
```

Primjer: Ispisati poštanske brojeve i nazive mjesta koja se ne nalaze u županiji sa šifrom 17

```
SELECT pbr, nazMjesto FROM mjesto  
WHERE sifZupanija <> 17
```



Definisanje uslova dohvata

2. Ispitivanje je li izraz unutar (ili izvan) intervala omeđenog sa druga dva izraza

Primjer: Ispisati podatke o matičnom broju, prezimenu i datumu rođenja studenta čiji je datum rođenja između '1.1.1981' i '5.4.1981'

```
SELECT mbrStud, prezStud, datRodStud FROM stud
      WHERE datRodStud BETWEEN '1983-1-1' AND
      '1983-4-5'
```

Primjer: Ispisati kratice i nazive predmeta koji imaju upisano manje od 5 ili više od 50 studenata

```
SELECT kratPred, nazPred FROM pred
      WHERE upisanoStud NOT BETWEEN 5 AND 50
```



Definisanje uslova dohvata

3. Ispitivanje je li izraz element nekog skupa (ili nije element skupa) koji može sadržavati numeričke, znakovne konstante, te “variable” CURRENT_DATE i CURRENT_USER

Primjer: Popis matičnih brojeva studenata koji su polagali ispit iz predmeta koji ima šifru unutar zadanog skupa šifara predmeta

```
SELECT DISTINCT mbrStud FROM ispit  
WHERE sifPred IN (346, 347, 348)
```

Primjer: Ispisati sve podatke o studentima koji nisu rođeni ‘23.7.1981’, ‘5.11.1980’ ili ‘31.1.1982’

```
SELECT * FROM stud  
WHERE datRodStud NOT IN  
( '1981-7-23', '1980-11-5', '1982-1-31' )
```



Definisanje uslova dohvata

4. Ispitivanje je li vrijednost atributa NULL (ili nije NULL)

Primjer: Ispisati ime i prezime studenata čiji je datum rođenja nepoznat ili nije upisan

```
SELECT imeStud, prezStud FROM stud  
WHERE datRodStud IS NULL;
```

Primjer: Ispisati naziv predmeta te broj upisanih studenata za sve predmete za koje je taj podatak poznat

```
SELECT nazPred, upisanoStud FROM pred  
WHERE upisanoStud IS NOT NULL;
```



Definisanje uslova dohvata

5. Ispitivanje zadovoljava li (ili ne zadovoljava) vrijednost atributa zadani uzorak (pattern) uz korištenje LIKE operatora

Pri korištenju LIKE operatora mogu se koristiti sljedeći wildcard znakovi:

- ▶ % - zamjenjuje bilo koju kombinaciju znakova
- ▶ _ - zamjenjuje jedan znak
- ▶ \ - ukida specijalno značenje znaka ispred kojeg se pojavi

Primjer: Ispis svih podataka o studentima čije ime sadrži kombinaciju znakova 'ar'

```
SELECT * FROM stud WHERE imeStud LIKE '%ar%'
```

Primjer: Ispis svih podataka o predmetima u čijem se nazivu, na bilo kojoj poziciji pojavljuje znak %

```
SELECT * FROM pred WHERE nazPred LIKE '%\%%'
```



Definisanje uslova dohvata

Primjer: Ispis poštanskog broja i naziva mjesta za sva mjesta koja u nazivu imaju drugo slovo 'a'

```
SELECT pbr, nazMjesto FROM mjesto
WHERE nazMjesto LIKE '_a%'
```

Primjer: Ispis imena, prezimena i datuma rođenja studenata čije se prezime ne završava na 'ić'

```
SELECT imeStud, prezStud, datRodStud FROM stud
WHERE prezStud NOT LIKE '%ić'
```

Primjer: Ispisati imena studenata koja počinju slovom 'B' i imaju 5 slova

```
SELECT imeStud FROM stud
WHERE imeStud LIKE 'B_____' -- 4 puta navedeno _
```



Definisanje uslova dohvata

6. Ispitivanje zadovoljava li (ili ne zadovoljava) vrijednost atributa zadani uzorak (pattern) uz korištenje RLIKE operatora

RLIKE operator omogućava korištenje sljedećih wildcard znakova i kombinacija sa slovima

- ▶ \wedge - da li se poklapa sa početkom stringa
- ▶ $\$$ - da li se poklapa sa krajem stringa
- ▶ $.$ – zamjenjuje bilo koji karakter
- ▶ $de | abc$ – zamjenjuje sekvencu znakova de ili abc
- ▶ $(abc)^*$ - zamjenjuje nula ili više sekvenci abc
- ▶ $[a-dX], [\wedge a-dX]$ – zamjenjuje bilo koji karakter koji je (nije – ako se koristi \wedge) a, b, c, d ili X



Definisanje uslova dohvata

- ▶ **Primjer:** Podaci o studentima čije ime sadrži jednu od kombinacija znakova – AR, aR, Ar, ar

```
SELECT * FROM stud
      WHERE imeStud RLIKE '[Aa][Rr]'
```

- ▶ **Primjer:** Ispis svih studenata čije prezime počinje nekim od znakova koji se nalaze između slova A i D, te završava jednim od znakova a, b, c, d, e

```
SELECT * FROM stud
      WHERE prezStud RLIKE '^[A-D]'
      AND prezStud RLIKE '[abcde]$'
```

- ▶ **Primjer:** Ispis svih studenata čije prezime počinje nekim od znakova koji se nalaze između slova E i G, i nakon tog prvog znaka sadrži bar dva znaka koji nisu samoglasnici

```
SELECT * FROM stud
      WHERE prezStud RLIKE '^[E-G]' AND prezStud RLIKE
      '[^aeiou]{2,}'
```



Definisanje uslova dohvata

- ▶ Uslovi za usporedbu mogu se međusobno kombinovati pomoću logičkih operatora AND, OR i NOT

Primjer: Ispis svih studenata koji su rođeni u mjestu s poštanskim brojem 76000 ili im prezime počinje znakom D, te uz taj uslov takođe zadovoljavaju uslov da stanuju u mjestu sa poštanskim brojem 75000

```
SELECT * FROM stud
      WHERE (pbrRod=76000 OR prezStud LIKE 'D%')
      AND pbrStan=75000
```



Evaluacija rezultata upita

- ▶ Uslov dohvata se može posmatrati kao predikat kojeg n-torka iz relacije mora zadovoljiti da bi se pojavila u izlaznoj listi rezultata upita
- ▶ Uvrštavanjem konkretnih vrijednosti iz n-torke u predikat dobiva se sud, koji se evaluira kao istinit (*true*) ili lažan (*false*)
- ▶ Samo one n-torke za koje se taj sud evaluira kao istinit, pojavit će se u rezultatu SELECT naredbe

Primjer: Neka relacija mjesto sadrži n-torke 75000, Tuzla; 71000, Sarajevo; 72000, Zenica

```
SELECT * FROM mjesto
```

```
WHERE pbr > 71000 AND nazMjesto LIKE 'Z%'
```

1. n-torka 75000 > 71000 AND 'Tuzla' LIKE 'Z%' → false

2. n-torka 71000 > 71000 AND 'Sarajevo' LIKE 'Z%' → false

3. n-torka 72000 > 71000 AND 'Zenica' LIKE 'Z%' → true

Rezultat upita će biti n-torka 72000, Zenica



Primjeri projekcije i selekcije

- ▶ **Primjer:** Ispisati inicijal imena, prezime i mjesec rođenja studenata koji su rođeni radnim danom u nekom od mjesta sa poštanskim brojem 10000, 21000 ili 51000 i ime im počinje suglasnikom.

```
SELECT CONCAT(SUBSTRING(imeStud,1,1),'.'), prezStud,  
MONTH(datRodStud) FROM stud  
WHERE WEEKDAY(datRodStud) BETWEEN 0 AND 4  
AND pbrRod IN (10000, 21000, 51000)  
AND imeStud RLIKE `^[^AEIOU]`
```

- ▶ **Primjer:** Ispisati ime nastavnika, prezime nastavnika velikim slovima i koeficijent za platu nastavnika za koje je poznato mjesto stanovanja i zaposleni su u organizacionoj jedinici sa parnom šifrom.

```
SELECT imeNastavnik, UPPER(prezNastavnik), koef  
FROM nastavnik  
WHERE pbrStan IS NOT NULL  
AND MOD(sifOrgjed, 2) = 0
```

