

Peta Nedelja – SQL: DML –

Autori: Mr. Miloš Cvetanović

- **Naredba upita**
 - Redni upit
 - Skalarni upit
 - Svodni upit

- **Naredbe ažuriranja**
 - Ubacivanje
 - Izmena
 - Uklanjanje

- **RedniUpit ::=**
 SELECT R-Lista
 FROM Tabela | Pogled
 [WHERE R-Predikat] ;
- **R-Lista ::=** * | { [ALL | DISTINCT] R-Izraz , ... }
- **R-PredikatRelacioni ::=** R-Izraz { \leq | \geq | \equiv | $\leq\equiv$ | $\geq\equiv$ | $\leq>$ } R-Izraz
- **Klauzula:** [ORDER BY { R-Izraz [ASC | DESC] } , ...]

- **SELECT ***
FROM Oblast ;
- **SELECT SifN**
FROM Knjiga ;
- **SELECT DISTINCT SifN**
FROM Knjiga ;
- **SELECT SifN, Naziv**
FROM Naslov
WHERE SifO = 'PJ' ;

- **SkalarniUpit ::=**
SELECT G-Lista
FROM Tabela | Pogled
[WHERE R-Predikat] ;
- **G-Lista ::= G-Izraz , ...**
- **Agregatne funkcije :**
SUM (Kolona)
AVG (Kolona)
MIN (Kolona)
MAX (Kolona)
COUNT (*)
COUNT ([ALL | DISTINCT] Kolona , ...)

- **SELECT COUNT(*)
FROM Clan ;**
- **SELECT COUNT(DISTINCT SifC)
FROM Pozajmica ;**
- **SELECT SUM (Dana)
FROM Pozajmica ;**
- **SELECT MIN (Dana), MAX (Dana)
FROM Pozajmica ;**
- **SELECT SUM (Dana), AVG (Dana)
FROM Pozajmica
WHERE SifC = 'JJ1' ;**



- **SvodniUpit ::=**
SELECT S-ListaKolona [, G-Lista]
FROM Tabela | Pogled
[WHERE R-Predikat]
GROUP BY G-ListaKolona
[HAVING G-Predikat] ;
- **S-ListaKolona ::= Kolona , ...**
- **G-Lista ::= G-Izraz , ...**
- **G-ListaKolona ::= Kolona , ...**

- **SELECT SifA, COUNT(*)
FROM Je_Autor
GROUP BY SifA ;**
- **SELECT SifC,
FROM Pozajmica
GROUP BY SifC
HAVING SUM (Dana) > 10 ;**
- **SELECT SifC, COUNT(*), SUM (Dana)
FROM Pozajmica
WHERE Dana > 2
GROUP BY SifC ;**

- **RedniUpitViseTabela ::=**
SELECT R-Lista
FROM { Tabela | Pogled [Nadimak] }, ...
[WHERE R-Predikat] ;
- **SkalarniUpitViseTabela ::=**
SELECT G-Lista
FROM { Tabela | Pogled [Nadimak] }, ...
[WHERE R-Predikat] ;
- **SvodniUpitViseTabela ::=**
SELECT S-ListaKolona [, G-Lista]
FROM { Tabela | Pogled [Nadimak] }, ...
[WHERE R-Predikat]
GROUP BY G-ListaKolona
[HAVING G-Predikat] ;

- **SELECT N.Naziv, O.Naziv
FROM Naslov N, Oblast O
WHERE N.SifO = O.SifO
ORDER BY N.Naziv ;**
- **SELECT SUM (Dana)
FROM Pozajmica P, Naslov N
WHERE P.SifN = N.SifN
AND SifO = 'BP' ;**
- **SELECT P.SifC, Ime, COUNT(*)
FROM Pozajmica P, Clan C
GROUP BY P.SifC, Ime ;**

- **SlozenPredikat ::=**
 [NOT] ProstPredikat [AND | OR SlozenPredikat]
- **Izraz { <= | >= | = | <=> | >=> | <> } Izraz**
- **Izraz [NOT] BETWEEN Izraz AND Izraz**
- **Kolona IS [NOT] NULL**
- **ZnakovniIzraz [NOT] LIKE ZnakovnaMaska**
- **Izraz [NOT] IN (Konstanta , ...)**
- **Izraz { <= | >= | = | <=> | >=> | <> } ANY (Konstanta , ...)**
- **Izraz { <= | >= | = | <=> | >=> | <> } ALL (Konstanta , ...)**

- **SELECT DISTINCT SifC
 FROM Pozajmica
 WHERE Dana BETWEEN 5 and 10 ;**

- **Podupit – SELECT naredba unutar WHERE ili HAVING kaluzule**
 - Nekorelisani podupit – izvršava samo jednom na početku izvršavanja “spoljnog” upita
 - Korelisani podupit – izvršava se za svaki red tabele koju obrađuje “spoljni” upit

- **Forme predikata**
 - Relacioni predikad, BETWEEN – rade i sa Skalarnim upitima
 - IN, ANY, ALL – rade i sa Rednim upitima

- **Sedma forma predikata :**
[NOT] EXISTS (RedniUpit)

- **SELECT ***
FROM Pozajmica
WHERE Dana > (SELECT MAX (Dana)
FROM Pozajmica
WHERE SifC = 'PP0');

- **SELECT Ime**
FROM Autor A
WHERE (SELECT COUNT(*)
FROM Je_Autor J, Naslov N
WHERE J.SifN = N.SifN
AND J.SifA = A.SifA
AND N.SifO = 'PJ')

=
(SELECT COUNT(*)
FROM Naslov
WHERE SifO = 'PJ');

- **KombinovaniUpit ::=**
 { Upit Operator KombinovaniUpit } | Upit
- **Operator ::= { UNION [ALL] } | EXCEPT | INTERSECT**
- **SELECT SifK**
 FROM Drzi

UNION

SELECT DISTINCT SifK
 FROM Pozajmica ;

- **NaredbaUbacivanja ::=**
INSERT INTO Tabela [(Kolona , ...)]
{ VALUES (Konstanta , ...) } | RedniUpit ;
- **INSERT INTO Naslov**
VALUES ('PJP0', 'Programski jezik PASCAL', 'PJ') ;
- **INSERT INTO NaslovPJ (SifraNaslova, NazivNaslova)**
SELECT SifN, Naziv
FROM Naslov
WHERE SifO = 'PJ' ;

- **NaredbaIzmene ::=**
UPDATE Tabela [Nadimak]
SET { Kolona \equiv R-Izraz | SkalarniUpit } , ...
[WHERE R-Predikat] ;

- **UPDATE Naslov**
SET SifO = 'BP'
WHERE SifN = 'RBPO' ;

- **NaredbaUklanjanja ::=**
DELETE FROM Tabela [Nadimak]
[WHERE R-Predikat] ;
- **DELETE FROM Clan**
WHERE SifC = 'MM0' ;

- **Data je šema relacione baze podataka fudbalskog saveza za potrebe evidencije utakmica jedne sezone (pretpostavka je da fudbaleri ne mogu da menjaju tim u kome igraju, u toku sezone):**

FUDBALER (SifF, Ime, SifT)

TIM (SifT, Naziv, Mesto)

UTAKMICA (SifU, SifTDomaci, SifTGost, Kolo, Ishod, Godina)

IGRAO (SifF, SifU, PozicijaIgraca)

GOL (SifG, SifU, SifF, RedniBrGola, Minut)

KARTON (SifK, SifU, SifF, Tip, Minut)

Napomena: Ishod (1-pobeda domaćih, 2-pobeda gostiju, X-nerešeno)

- **Sastaviti SQL skript koji kao rezultat daje šifre i imena fudbalera kao i prosečan broj golova po utakmici koji su ti fudbaleri dali, ali samo za one fudbalere koji su dali bar jedan gol nakon što su dobili žuti karton, i to u utakmici gde njihov tim bio gostujuć.**
- **Sastaviti SQL skript koji kao rezultat daje šifre i imena fudbalera, ali samo za one fudbalere koji su dali bar dva gola i to na nekoj od utakmica na kojoj je njihov tim pobedio, i pri tom su tačno dva gola tog fudbalera bili ujedno i poslednji golovi na utakmici.**

```
CREATE VIEW BrUtakmica(SifF, BrUtakmica)
AS SELECT SifF, COUNT(*) FROM IGRAO GROUP BY SifF;
```

```
CREATE VIEW BrGolova(SifF, BrGolova)
AS SELECT SifF, COUNT(*) FROM GOL GROUP BY SifF;
```

```
CREATE VIEW DaliGolNakonKartona (SifF)
AS SELECT SifF
FROM FUDBALER F, UTAKMICA U, GOL G, KARTON K
WHERE F.SifF=G.SifF
AND F.SifF=K.SifF
AND G.Minut > K.Minut
AND G.SifU=K.SifU
AND G.SifU=U.SifU
AND F.SifT=U.SifTGost;
```

```
FUDBALER (SifF, Ime, SifT)
TIM (SifT, Naziv, Mesto)
UTAKMICA (SifU, SifTDomaci, SifTGost, Kolo, Ishod, Godina)
IGRAO (SifF, SifU, PozicijaIgraca)
GOL (SifG, SifU, SifF, RedniBrGola, Minut)
KARTON (SifK, SifU, SifF, Tip, Minut)
```

-- Nismo gledali tip kartona, jer se samo nakon žutog može nastaviti i dati gol

```
SELECT F.SifF, F.Ime, BG.BrGolova / BU.BrUtakmica
FROM FUDBALER F, BrGolova BG, BrUtakmica BU
WHERE F.SifF=BG.SifF AND F.SifF=BU.SifF
AND F.SifF IN (SELECT SifF FROM DaliGolNakonKartona);
```

-- Poslednji uslov je tu da bi dobili samo one igrače koji su od interesa

```
CREATE VIEW BrUtakmica(SifF, SifU)
AS SELECT F.SifF, U.SifU
FROM FUDBALER F, UTAKMICA U, GOL G
WHERE F.SifF=G.SifF
      AND G.SifU=U.SifU
      AND ( (F.SifT=U.SifTDomaci AND Ishod='1')
            OR
            (F.SifT=U.SifTGost AND Ishod='2')
          )
      AND NOT EXISTS (SELECT SifF
                      FROM GOL L
                      WHERE L.SifF <> F.SifF
                             AND L.SifU=U.SifU
                             AND L.Minut > G.Minut
                      )
GROUP BY F.SifF, U.SifU
HAVING COUNT(*)=2;

SELECT DISTINCT F.SifF, F.Ime
FROM FUDBALER F, USLOV U
WHERE F.SifF=U.SifF;
```

```
FUDBALER (SifF, Ime, SifT)
TIM (SifT, Naziv, Mesto)
UTAKMICA (SifU, SifTDomaci, SifTGost, Kolo, Ishod, Godina)
IGRAO (SifF, SifU, PozicijaIgraca)
GOL (SifG, SifU, SifF, RedniBrGola, Minut)
KARTON (SifK, SifU, SifF, Tip, Minut)
```

- **Data je šema relacione baze podataka za potrebe skladišta robe u toku jedne godine:**
ROBA(SifR, Naziv, Opis, SifD);
DOBAVLJAC(SifD, Naziv, Adresa);
NABAVKA(SifN, Datum, Kolicina, Cena, SifR);
- **Sastaviti SQL skript koji vraća cenu pri kojoj bi trebalo prodati 40 jedinica robe sa šifrom 30, ukoliko te robe ima u dovoljnoj količini. Cena se formira tako što se najpre rasprodaju količine koje su najskorije nabavljene (LIFO).**

```

CREATE VIEW LIFO (Proizvod, Datum, JedCena, TotalKol, TotalCena)
AS SELECT R1.SifR, R1.Datum, R1.Cena, SUM(R2.Kolicina), SUM(R2.Kolicina*R2.Cena)
FROM Nabavka R1, Nabavka R2
WHERE R2.Datum >= R1.Datum AND R2.SifR = R1.SifR
GROUP BY R1.SifR, R1.Datum, R1.Cena;

```

```

SELECT (TotalCena - ((TotalKol- 40)*JedCena))
FROM LIFO
WHERE Proizvod = 30
AND Datum = ( SELECT MAX (Datum)
              FROM LIFO
              WHERE TotalKol >= 40
                AND Proizvod =30);

```

Rezultat za dati sadržaj: 50

```

ROBA(SifR, Naziv, Opis, SifD);
DOBAVLJAC(SifD, Naziv, Adresa);
NABAVKA(SifN, Datum, Kolicina, Cena, SifR);

```

Nabavka	SifN	Datum	Kolicina	Cena	SifR
	10	1	20	1	30
	20	1	10	2	10
	30	2	10	1	20
	40	2	20	2	20
	50	2	30	1	10
	60	5	10	1	30
	70	6	10	2	20
	80	6	20	2	10
	90	6	30	2	20
	100	8	20	1	30
	110	8	10	1	10
	120	9	10	2	30

- **Data je šema relacione baze podataka, za potrebe izračunavanja pomoću matrica:**
MATRICA(SifM, Naziv, BrVrsta, BrKolona);
PODACI(SifP, I, J, Vrednost, SifM);
- **Sastaviti SQL skript koji vraća vrstu, kolonu i vrednost svakog od elemenata matrice nastale množenjem matrica sa nazivima A i B.**

```

SELECT T2.I, T4.J, SUM(T2.Vrednost * T4. Vrednost)
FROM Matrica T1, Podaci T2, Matrica T3, Podaci T4
WHERE T1.SifM = T2.SifM AND T1.Naziv = 'A'
      AND T3.SifM = T4.SifM AND T3.Naziv = 'B'
      AND T2.J = T4.I
GROUP BY T2.I, T4.J;
    
```

Podaci	SifP	I	J	Vrednost	SifM
	1	1	2	3	10
	2	2	1	1	10
	3	1	1	2	10
	4	2	2	2	10
	5	1	3	1	20
	6	1	1	3	20
	7	2	1	2	20
	8	2	2	5	20
	9	2	3	2	20
	10	1	2	4	20
	11	2	1	4	30
	12	1	1	5	30

Odgovor za dati sadržaj:

```

1  1  12
2  1   7
1  2  23
2  2  14
1  3   8
2  3   5
    
```

Matrica	SifM	Naziv	BrVrsta	BrKolona
	10	'A'	2	2
	20	'B'	2	3
	30	'C'	2	1

- **Data je šema relacione deo baze podataka fakulteta:**
STUDENT (SifS, Ime, BrIndeksa)
PROFESOR (SifP, Ime, SifO)
ODSEK (SifO, Naziv)
KURS (SifK, Naziv, BrKredita, SifO)
UČIONICA (SifU, BrMesta)
PREDUSLOV (SifK, SifKP)
POHAĐA(SifS, SifR)
RASPORED (SifR, SifP, SifK, SifU, Termin, Dan, Br.Prijavljenih)
- **Sastaviti SQL skript koji proverava kvalitet rasporeda studenata i pored broja indeksa ispisuje i odgovarajuću poruku. Raspored studenta je loš ukoliko u svom rasporedu bar jednog dana ima prekid u terminima u kojima prati predavanje, u suprotnom raspored se smatra dobrim. Za sve studente sa lošim rasporedom treba ispisati broj indeksa i poruku LOS, a pored broja indeksa studenata sa dobrim rasporedom poruku DOBAR.**

```
CREATE VIEW LosRaspored(SifS, Dan)  
AS SELECT P.SifS, R.Dan  
FROM POHADJA P, RASPORED R  
WHERE P.SifR=R.SifR  
GROUP BY P.SifS, R.Dan  
HAVING COUNT(R.Termin) < (MAX(R.Termin)-MIN(R.Termin)+1);
```

```
SELECT S.BrIndeksa, 'LOS'  
FROM STUDENT S  
WHERE S.SifS IN (SELECT SifS FROM LosRaspored)  
UNION  
SELECT S.BrIndeksa, 'DOBAR'  
FROM STUDENT S  
WHERE S.SifS NOT IN (SELECT SifS FROM LosRaspored)  
AND S.SifS IN (SELECT SifS FROM POHADJA);
```

```
STUDENT (SifS, Ime, BrIndeksa)  
PROFESOR (SifP, Ime, SifO)  
ODSEK (SifO, Naziv)  
KURS (SifK, Naziv, BrKredita, SifO)  
UČIONICA (SifU, BrMesta)  
PREDUSLOV (SifK, SifKP)  
POHAĐA(SifS, SifR)  
RASPORED (SifR, SifP, SifK, SifU, Termin, Dan, Br.Prijavljenih)
```

- **Data je šema relacione baze podataka veleprodajnog lanca prodavnica:**
PRODAVNICA(SifP, Adresa, SifM);
MESTO(SifM, Naziv);
KLIJENT(SifK, Naziv, SifM);
RACUN(SifR, SifK, SifP, SifRa, Datum);
PROIZVOD(SifPr, Naziv, Cena);
STAVKA_RACUNA(SifS, SifR, SifPr, RedniBr, Kolicina, Iznos);
RADNIK(SifRa, Ime, SifP);
- **Sastaviti SQL skript koji za svaki datum, za koji je izdat bar jedan račun, daje ukupan iznos računa izdatih do tog datuma (uključujući i posmatrani datum).**

```
CREATE VIEW IznosPoDatumu (Datum, Total)  
AS SELECT R.Datum, SUM(S.Iznos)  
FROM RACUN R, STAVKA_RACUNA S  
WHERE R.SifR=S.SifR  
GROUP BY R.Datum;
```

```
SELECT A.Datum, SUM(B.Total)  
FROM IznosPoDatumu A, IznosPoDatumu B  
WHERE A.Datum >= B.Datum  
GROUP BY A.Datum;
```

```
PRODAVNICA(SifP, Adresa, SifM);  
MESTO(SifM, Naziv);  
KLIJENT(SifK, Naziv, SifM);  
RACUN(SifR, SifK, SifP, SifRa, Datum);  
PROIZVOD(SifPr, Naziv, Cena);  
STAVKA_RACUNA(SifS, SifR, SifPr, RedniBr, Kolicina, Iznos);  
RADNIK(SifRa, Ime, SifP);
```