

Predavanja

Svibanj, 2021.



Vrste tablica (relacija)

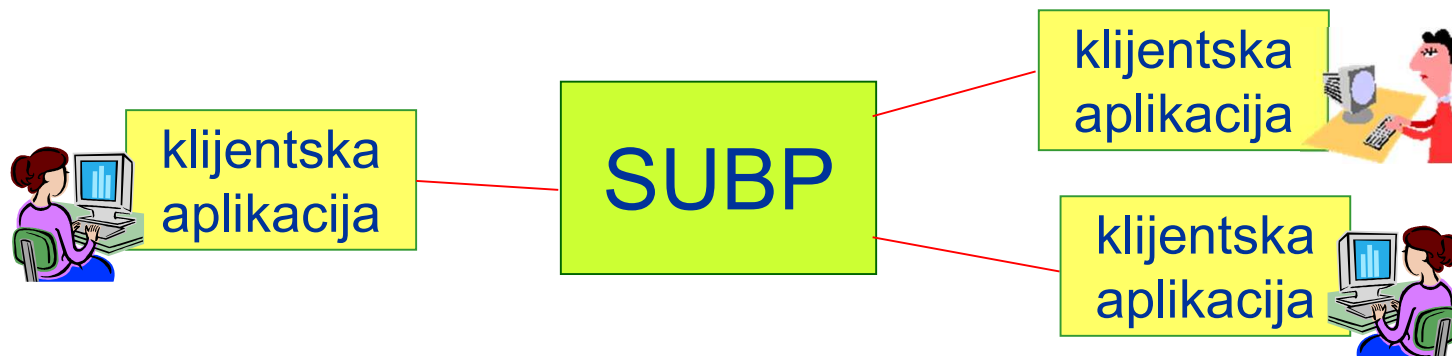
1. Temeljna tablica (*base table*)
2. Privremena tablica (*temporary table*)
3. Virtualna tablica (*virtual table, view*)

1. Temeljna tablica

- Tablica korespondentna relaciji, odnosno skupu entiteta u konceptualnoj shemi, čija su shema i sadržaj trajno pohranjeni u bazi podataka
- Obavljanjem naredbe CREATE TABLE u rječnik podataka se pohranjuju metapodaci
 - naziv tablice
 - nazivi i tipovi atributa
 - integritetska ograničenja
 - ostali metapodaci (vrijeme kreiranja, vlasnik, primijenjena fizička organizacija, itd.)
- Shema i sadržaj temeljne tablice su **postojani**: pohranjeni su u bazi podataka na neograničeno vrijeme
 - mijenjaju se tek u slučaju obavljanja eksplicitnih operacija za izmjenu sadržaja (UPDATE, DELETE, INSERT) ili sheme tablice (ALTER TABLE)

(SQL-sjednica)

- SQL-sjednica (*SQL-session*) je kontekst u kojem jedan korisnik obavlja niz SQL naredbi putem jedne veze (*SQL-Connection*) prema sustavu za upravljanje bazama podataka
 - SQL-sjednica započinje u trenutku kada korisnik ostvari vezu (*connect*) sa sustavom za upravljanje bazama podataka
 - npr. u trenutku kada korisnik uporabom klijentske aplikacije pgAdmin ostvari vezu s PostgreSQL sustavom za upravljanje bazama podataka
 - SQL-sjednica završava u trenutku kada korisnik prekine vezu (*disconnect*) prema sustavu za upravljanje bazama podataka



2. Privremena tablica

- Tablica čija su shema i sadržaj u bazu podataka pohranjeni privremeno
- Stvara se obavljanjem naredbe `CREATE TEMP TABLE`
 - sintaksa preostalog dijela naredbe je identična sintaksi naredbe `CREATE TABLE`, uz određena ograničenja
 - npr. nije moguće definirati ograničenje referencijskog integriteta
- Privremena tablica je u dosegu ("vidljiva je") isključivo u okviru SQL-sjednice tijekom koje je kreirana
 - **svaka SQL-sjednica koristi svoje privremene tablice**
- Privremene tablice se koriste kao pomoćni objekti, npr. za pohranu međurezultata pri obavljanju složenijih upita
 - **zašto temeljne tablice nisu prikladne za tu namjenu?**
- Privremena tablica se uklanja iz baze podataka:
 - obavljanjem naredbe `DROP TABLE nazivPrivremenetablice` ili
 - završetkom SQL-sjednice tijekom koje je ta privremena tablica kreirana, ili
 - (PostgreSQL: moguće je kod definicije privremene tablice navesti da se uklanja po završetku transakcije)

Vježba

- Svaka SQL-sjednica koristi svoje privremene tablice
- Otvorite dvije sjednice (npr. pgAdmin) i u svakoj obavite:

```
CREATE TEMP TABLE t1 (  
    id int  
);  
INSERT INTO t1 VALUES (1);  
SELECT * FROM t1;
```

Što vraća?

Što vraća?

```
SELECT * FROM t1;
```

```
SELECT * FROM t1;
```

Što vraća?

```
CREATE TEMP TABLE t1 (  
    id int  
);  
INSERT INTO t1 VALUES (2);  
SELECT * FROM t1;
```

Što vraća?

Primjer

- Ispisati najmanju i najveću prosječnu ocjenu predmeta u obliku:

minPros	maksPros
3.00	4.00

```
CREATE TEMP TABLE prosjek (  
    sifPred INTEGER  
    , prosOcj DECIMAL(3,2));
```

```
INSERT INTO prosjek  
    SELECT sifPred, AVG(ocj)  
    FROM polozeniIspit  
    GROUP BY sifPred;
```

```
SELECT MIN(prosOcj) AS minPros  
    , MAX(prosOcj) AS maksPros  
    FROM prosjek;
```

polozeniIspit

mbr	sifPred	ocj
100	1001	5
101	1001	4
102	1001	3
100	1002	2
101	1002	5
100	1003	3
101	1003	3

prosjek

sifPred	prosOcj
---------	---------

prosjek

sifPred	prosOcj
1001	4.00
1002	3.50
1003	3.00

minPros	maksPros
3.00	4.00

Za vježbu riješiti bez korištenja privremene tablice!

Skraćena sintaksa stvaranja TEMP tablice

- Većina SUBP-ova podržava i skraćenu sintaksu stvaranja TEMP tablice, na temelju upita.
- Npr. u PostgreSQL-u, sljedeća naredba zamjenjuje prve dvije naredbe s prethodne prikaznice:

```
CREATE TEMP TABLE prosjek (sifPred,prosOcj)
AS
SELECT sifPred, AVG(ocj) AS prosOcj
FROM polozeniIspit
GROUP BY sifPred;
```


Primjer (nastavak)

- Treba primijetiti: sadržaj privremene tablice *prosjek* **neće** se "automatski" promijeniti nakon upisa još jedne n-torke u temeljnu tablicu *polozeniIspit*

```
INSERT INTO polozeniIspit VALUES(102, 1003, 2);
```

polozeniIspit	mbr	sifPred	ocj
	100	1001	5
	101	1001	4
	102	1001	3
	100	1002	2
	101	1002	5
	100	1003	3
	101	1003	3
	102	1003	2



Sadržaj
privremene
tablice se
time nije
promijenio:



prosjek	sifPred	prosOcj
	1001	4.00
	1002	3.50
	1003	3.00

prosjek za predmet
1003 bi trebao biti 2.67

```
SELECT MIN(prosOcj) AS minPros  
       , MAX(prosOcj) AS maksPros  
FROM prosjek;
```

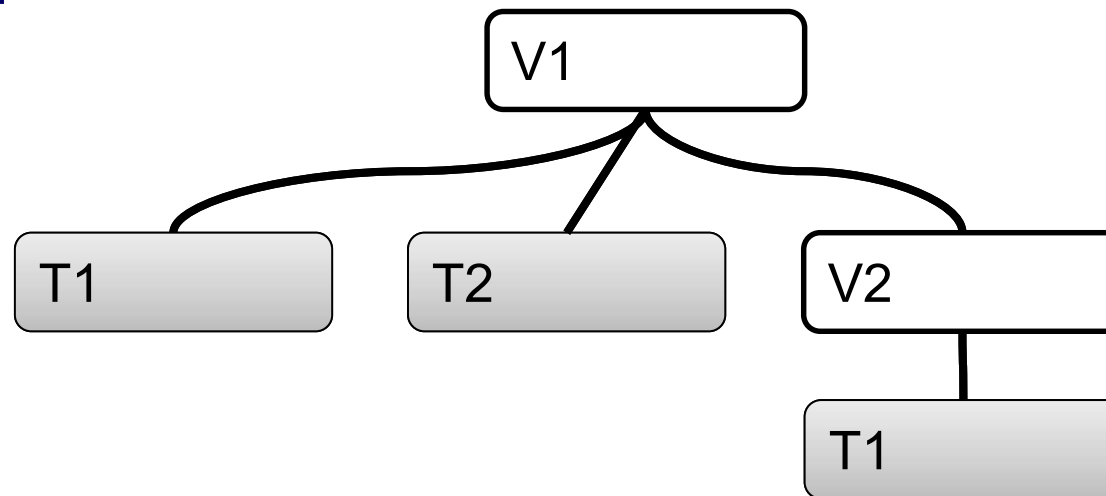


minPros	maksPros
3.00	4.00

**neispravan
rezultat**

3. Virtualne tablice

- Tablica kojoj su shema i sadržaj definirani izrazom relacijske algebre čiji su operandi temeljne ili virtualne tablice.
 - u praksi, shema i sadržaj virtualne tablice opisuju se u obliku SQL upita



- Sadržaj virtualne tablice dinamički se određuje u trenutku obavljanja operacije nad virtualnom tablicom: ovisi o trenutnom stanju temeljnih tablica

Virtualna tablica (primjer)

- Problem "zastarijevanja" podataka u privremenim tablicama može se izbjeći uporabom virtualnih tablica

polozeniIspit

mbr	sifPred	ocj
100	1001	5
101	1001	4
102	1001	3
100	1002	2
101	1002	5
100	1003	3
101	1003	3

```
CREATE VIEW prosjek (sifPred  
                    , prosOcj) AS  
SELECT sifPred, AVG(ocj)  
FROM polozeniIspit  
GROUP BY sifPred;
```

prosjek

sifPred	prosOcj
	?

- Tek u trenutku obavljanja upita, SUBP dinamički određuje sadržaj virtualne tablice *prosjek*

prosjek

sifPred	prosOcj
1001	4.00
1002	3.50
1003	3.00

```
SELECT MIN(prosOcj) AS minPros  
      , MAX(prosOcj) AS maksPros  
FROM prosjek;
```

minPros	maksPros
3.00	4.00

Primjer (nastavak)

- Sadržaj virtualne tablice se ponovno određuje pri izvršavanju svakog upita koji koristi tu virtualnu tablicu

```
INSERT INTO polozeniIspit VALUES(102, 1003, 2);
```

```
SELECT MIN(prosOcj) AS minPros  
      , MAX(prosOcj) AS maksPros  
FROM prosjek;
```

prosjek

sifPred	prosOcj
1001	4.00
1002	3.50
1003	2.67

minPros	maksPros
2.67	4.00

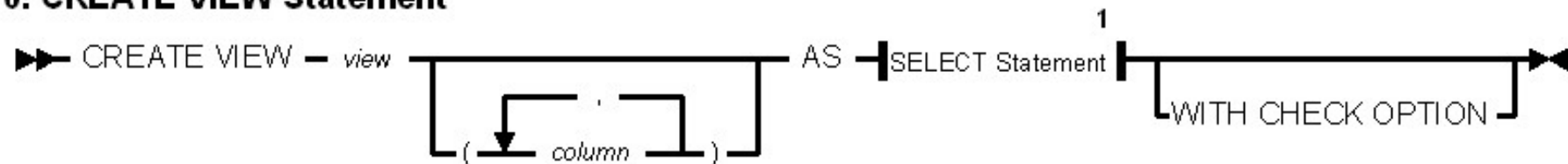
polozeniIspit

mbr	sifPred	ocj
100	1001	5
101	1001	4
102	1001	3
100	1002	2
101	1002	5
100	1003	3
101	1003	3
102	1003	2

Naredba za kreiranje virtualne tablice

- Virtualna tablica se kreira naredbom CREATE VIEW

6. CREATE VIEW Statement



- SELECT Statement* može sadržavati sve prethodno opisane dijelove SELECT naredbe
- Virtualna tablica se uklanja iz baze podataka naredbom:
 - DROP VIEW *nazivVirtualnetablice*;

Svojstva virtualne tablice

- Obavljanjem naredbe CREATE VIEW u rječnik podataka se pohranjuje samo definicija virtualne tablice
 - sadržaj virtualne tablice se određuje tek za vrijeme izvršavanja upita koji koristi virtualnu tablicu
 - odnosno, sadržaj virtualne tablice uvijek odražava sadržaj temeljnih tablica u trenutku izvršavanja upita u kojem se virtualna tablica koristi
- virtualne tablice se u upitima mogu koristiti na svim mjestima gdje se mogu koristiti temeljne tablice
 - između ostalog i za kreiranje novih virtualnih tablica
- za razliku od privremene tablice
 - definicija virtualne tablice je trajno pohranjena u bazi podataka
 - virtualna tablica je u dosegu ("vidljiva je") u svim SQL-sjednicama
 - (PostgreSQL podržava i koncept TEMP VIEW, virtualna tablica koja traje samo koliko i sjednica)

Atributi virtualne tablice

- Ako se nazivi atributa u definiciji virtualne tablice ne navedu, nazivi atributa virtualne tablice određeni su nazivima atributa u SELECT naredbi kojom se definira sadržaj virtualne tablice
- tipovi podataka za attribute virtualne tablice proizlaze iz tipova podataka atributa temeljnih tablica koje se koriste u definiciji virtualne tablice

osoba			
mbr	ime	prez	pbrStan
101	Ana	Kolar	23000
102	Tomo	Novak	21000
103	Tea	Ban	23000

```
CREATE VIEW zadrani1 AS
  SELECT mbr, ime, prez
    FROM osoba
   WHERE pbrStan = 23000;
SELECT * FROM zadrani1;
```

mbr	ime	prez
101	Ana	Kolar
103	Tea	Ban

```
CREATE VIEW zadrani2 (matBr
                      , imeSt
                      , prezSt) AS
  SELECT mbr, ime, prez
    FROM osoba
   WHERE pbrStan = 23000;
SELECT * FROM zadrani2;
```

matBr	imeSt	prezSt
101	Ana	Kolar
103	Tea	Ban

Atributi virtualne tablice

- Ako se u listi za selekciju pri definiciji virtualne tablice koriste izrazi, nazive atributa virtualne tablice treba eksplicitno navesti

polozeniIspit

mbr	sifPred	ocj
100	1001	2
101	1001	4
102	1001	3
100	1002	2
101	1002	5
100	1003	3
101	1003	3

```
CREATE VIEW prosjek (sifPred  
                    , prosOcj) AS  
SELECT sifPred, AVG(ocj)  
FROM polozeniIspit  
GROUP BY sifPred;
```

ispravno

```
CREATE VIEW prosjek AS  
SELECT sifPred  
      , AVG(ocj)  
FROM polozeniIspit  
GROUP BY sifPred;
```

neispravno

```
CREATE TEMP TABLE T1 AS  
SELECT 1 AS A, NULL AS B;
```

Dopušteno, ali **besmisleno** u PostgreSQLu, atribut će se zvati „?column?“. Ako se navedu dva izraza, onda dolazi do greške.

Kod ovog pristupa treba pripaziti na:
Nazive atributa koje vraća upit (jer će poslužiti za imenovanje atributa TEMP tablice)

Materijalizirane virtualne tablice

- SUBP fizički pohranjuje sadržaj virtualne tablice. Kada se promijeni sadržaj neke od temeljnih tablica pomoću kojih je virtualna tablica definirana, SUBP automatski mijenja i sadržaj materijalizirane virtualne tablice
 - prednost: virtualne tablice koje se vrlo često koriste, a čiji se sadržaj određuje složenim upitima, ne moraju se svaki puta kada neki korisnik koristi tu virtualnu tablicu ponovno izračunavati
 - nedostatak: ako se temeljne tablice pomoću kojih je virtualna tablica definirana često mijenjaju, pri svakoj izmjeni temeljnih tablica troši se dodatno vrijeme radi izmjene sadržaja virtualne tablice
- Podržavaju rijetki, Oracle prvi, SQL Server (*Indexed views*)
- PostgreSQL ne održava automatski podatke u materijaliziranoj virtualnoj tablici ažurnim, tj. potrebno je ručno osvježiti podatke

Implementacija virtualnih tablica

- Kako sustavi za upravljanje bazama podataka izvršavaju upite koji sadrže virtualne tablice (ako se ne radi o materijaliziranim virtualnim tablicama)?

Modifikacijom upita

- SUBP ugrađuje elemente definicije virtualne tablice u originalni SQL upit koji koristi virtualnu tablicu - umjesto originalnog SQL upita izvršava se modificirani SQL upit

Izvršavanje modifikacijom upita

- Primjer:

ispit		
mbr	predmet	ocj
100	Elektronika	3
100	Fizika	2
101	Elektronika	5
101	Fizika	2
102	Fizika	1
103	Fizika	5

stud			
mbr	ime	prez	pbrStan
100	Ivan	Kolar	52100
101	Ana	Horvat	42230
102	Jura	Novak	52100
103	Ana	Ban	52100

mjesto	
pbr	nazMjesto
42000	Varaždin
52100	Pula
42230	Ludbreg

studenti koji su položili predmet Fizika

```
CREATE VIEW polFiz AS
SELECT stud.*, ocj
FROM ispit, stud
WHERE ispit.mbr = stud.mbr
AND predmet = 'Fizika'
AND ocj > 1;
```

korisnik obavlja:

```
SELECT * FROM polFiz;
```



SUBP modificira upit

```
SELECT stud.*, ocj
FROM ispit, stud
WHERE ispit.mbr = stud.mbr
AND predmet = 'Fizika'
AND ocj > 1;
```



mbr	ime	prez	pbrStan	ocj
100	Ivan	Kolar	52100	2
101	Ana	Horvat	42230	2
103	Ana	Ban	52100	5

Primjer (nastavak)

- Ispisati prezime, ime i dobivenu ocjenu iz Fizike za studente koji su položili Fiziku, a stanuju u Puli

korisnik obavlja:

```
SELECT polFiz.prez, polFiz.ime, polFiz.ocj
FROM polFiz, mjesto
WHERE polFiz.pbrStan = mjesto.pbr
AND nazMjesto = 'Pula';
```



SUBP modificira upit

```
SELECT stud.prez, stud.ime, ispit.ocj
FROM ispit, stud, mjesto
WHERE ispit.mbr = stud.mbr
AND predmet = 'Fizika'
AND ocj > 1
AND stud.pbrStan = mjesto.pbr
AND nazMjesto = 'Pula';
```



prez	ime	ocj
Kolar	Ivan	2
Ban	Ana	5

```
CREATE VIEW polFiz AS
SELECT stud.*, ocj
FROM ispit, stud
WHERE ispit.mbr = stud.mbr
AND predmet = 'Fizika'
AND ocj > 1;
```

Virtualna tablica: INSERT, UPDATE, DELETE

- virtualne tablice se također mogu koristiti u naredbama INSERT, UPDATE i DELETE

```
CREATE VIEW splitStud AS  
  SELECT mbr, ime, prez, pbrStan  
  FROM stud  
  WHERE pbrStan = 21000;
```

```
INSERT INTO splitStud  
VALUES (102, 'Jure', 'Novak', 21000);
```

```
SELECT * FROM splitStud;
```

```
INSERT INTO splitStud  
VALUES (103, 'Tea', 'Ban', 10000);
```

```
SELECT * FROM splitStud;
```

n-torka jest unesena u temeljnu tablicu,
ali se "ne vidi" u virtualnoj tablici

```
SELECT * FROM stud;
```

stud			
mbr	ime	prez	pbrStan
100	Ivan	Kolar	31000
101	Ana	Horvat	21000

mbr	ime	prez	pbrStan
101	Ana	Horvat	21000
102	Jure	Novak	21000

mbr	ime	prez	pbrStan
101	Ana	Horvat	21000
102	Jure	Novak	21000

mbr	ime	prez	pbrStan
100	Ivan	Kolar	31000
101	Ana	Horvat	21000
102	Jure	Novak	21000
103	Tea	Ban	10000

Virtualna tablica: INSERT, UPDATE, DELETE

- SUBP ne može promijeniti "sadržaj virtualne tablice" - umjesto toga mora promijeniti sadržaj temeljnih tablica koje se koriste u definiciji te virtualne tablice

ispit

mbr	predmet	ocj
100	Elektronika	1
100	Fizika	5
101	Elektronika	1
101	Fizika	3

```
CREATE VIEW prosli AS  
SELECT * FROM ispit  
WHERE ocj > 1;
```

```
CREATE VIEW pali AS  
SELECT * FROM ispit  
WHERE ocj = 1;
```

korisnik
obavlja:

```
UPDATE prosli SET ocj = 4  
WHERE mbr = 100  
AND predmet = 'Fizika';
```



SUBP modificira upit

```
UPDATE ispit SET ocj = 4  
WHERE ocj > 1  
AND mbr = 100  
AND predmet = 'Fizika';
```

Virtualna tablica: problem migrirajućih n-torki

- n-torka se pojavljuje u virtualnoj tablici onda kada zadovoljava uvjet iz definicije virtualne tablice
 - n-torka unesena u virtualnu tablicu ili izmijenjena u virtualnoj tablici može "nestati" iz te virtualne tablice (i eventualno se "pojaviti" u nekoj drugoj virtualnoj tablici)

ispit

	mbr	predmet	ocj
t ₁	100	Elektronika	1
t ₂	100	Fizika	5
t ₃	101	Elektronika	1
t ₄	101	Fizika	3

```
CREATE VIEW prosli AS  
  SELECT * FROM ispit  
  WHERE ocj > 1;
```

```
CREATE VIEW pali AS  
  SELECT * FROM ispit  
  WHERE ocj = 1;
```

korisnik
obavlja:

```
UPDATE prosli SET ocj = 1  
  WHERE mbr = 100  
        AND predmet = 'Fizika';  
INSERT INTO prosli  
  VALUES (102, 'Elektronika', 1);
```

- n-torka t₂ je "nestala" iz *prosli* i "pojaviła" se u *pali*
- nova n-torka <102, Elektronika, 1> unesena preko *prosli* se "pojaviła" u *pali*

Virtualna tablica: problem migrirajućih n-torki

- **Rješenje:** virtualne tablice koje se koriste u naredbama koje mijenjaju podatke obavezno se kreiraju uz opciju **WITH CHECK OPTION**
 - SUBP tada ne dopušta izmjenu ili unos n-torke putem virtualne tablice ukoliko n-torka nakon obavljanja operacije više ne bi pripadala virtualnoj tablici putem koje je izmijenjena ili unesena

ispit

mbr	predmet	ocj
100	Elektronika	1
100	Fizika	5
101	Elektronika	1
101	Fizika	3

```
CREATE VIEW prosli AS  
SELECT * FROM ispit  
WHERE ocj > 1  
WITH CHECK OPTION;
```

```
CREATE VIEW pali AS  
SELECT * FROM ispit  
WHERE ocj = 1  
WITH CHECK OPTION;
```

```
UPDATE prosli SET ocj = 1  
WHERE mbr = 100  
AND predmet = 'Fizika';
```

pogreška

```
UPDATE prosli SET ocj = 4  
WHERE mbr = 100  
AND predmet = 'Fizika';
```

O.K.

```
INSERT INTO prosli  
VALUES (102, 'Fizika', 1);
```

pogreška

```
INSERT INTO prosli  
VALUES (102, 'Fizika', 3);
```

O.K.

```
INSERT INTO pali  
VALUES (102, 'Fizika', 3);
```

pogreška

Neizmjenjive virtualne tablice

- SUBP ne može promijeniti "sadržaj virtualne tablice" - umjesto toga mora promijeniti sadržaj temeljnih tablica koje se koriste u definiciji te virtualne tablice
 - ako je virtualna tablica definirana tako da SUBP nije u stanju **jednoznačno** odrediti koje operacije treba obaviti na temeljnim tablicama, tada je virtualna tablica **neizmjenjiva** (*non-updatable*)

polozeniIspit

mbr	sifPred	ocj
100	1001	5
101	1001	4
102	1001	3
100	1002	2
101	1002	5
100	1003	3
101	1003	3

```
CREATE VIEW prosjek (sifPred  
                    , prosOcj) AS  
SELECT sifPred, AVG(ocj)  
FROM polozeniIspit  
GROUP BY sifPred;
```

SELECT * FROM prosjek;

sifPred	prosOcj
1001	4.00
1002	3.50
1003	3.00

```
UPDATE prosjek SET prosOcj = 4.5  
WHERE sifPred = 1001;
```

?

```
INSERT INTO prosjek VALUES (1004, 2.5);
```

?

Izmjenjive virtualne tablice

- Virtualna tablica je izmjenjiva ako u glavnom SELECT dijelu definicije virtualne tablice koristi attribute iz samo jedne temeljne tablice $r(R)$ i pri tome:
 - ne sadrži eliminaciju duplikata pomoću DISTINCT
 - ne sadrži izraze u listi za selekciju (osim trivijalnih izraza koji sadrže samo ime atributa)
 - izostavljeni atributi ne smiju imati NOT NULL ograničenje ili moraju imati pretpostavljenu (*default*) vrijednost
 - ne sadrži spajanje ili uniju
 - ne sadrži grupiranje i postavljanje uvjeta nad grupom (GROUP BY i HAVING)
- Prethodno navedena ograničenja se ne odnose na eventualne podupite koji se koriste unutar WHERE dijela SELECT naredbe koja se koristi za definiciju virtualne tablice

Primjeri izmjenjivih virtualnih tablica

ispit	matBr	sifPred	datIsp	ocj	sifNas
	1111	1001	29.01.2011	1	101
	1111	1001	05.02.2011	3	101
	1111	1003	28.06.2011	2	303
	1111	1002	27.06.2011	4	202
	1234	1001	29.01.2011	3	202

stud	matBr	prez	ime	pbrSt
	1111	Novak	Ivan	10000
	4444	Ban	Marko	51000
	1234	Kolar	Petar	23000

```
CREATE VIEW poloziliNista AS
  SELECT * FROM stud
  WHERE NOT EXISTS
    (SELECT * FROM ispit
     WHERE ispit.matBr = stud.matBr
       AND ocj > 1)
  WITH CHECK OPTION;
```

```
CREATE VIEW ispitiZadranal AS
  SELECT matBr
        , sifPred
        , datIsp
        , ocj
  FROM ispit
  WHERE matBr IN (
    SELECT matBr FROM stud
    WHERE pbrSt = 23000)
  WITH CHECK OPTION;
```

Primjeri neizmjenjivih virtualnih tablica

ispit	matBr	sifPred	datIsp	ocj	sifNas
	1111	1001	29.01.2011	1	1111
	1111	1001	05.02.2011	3	1111
	1111	1003	28.06.2011	2	3333
	1111	1002	27.06.2011	4	2222
	1234	1001	29.01.2011	3	2222

```
CREATE VIEW ispitizadrana2 AS
  SELECT ispit.matBr
    , sifPred
    , datIsp
    , ocj
  FROM ispit, stud
 WHERE ispit.matBr = stud.matBr
    AND pbrSt = 23000;
```



usporediti s izmjenjivom virtualnom
tablicom `ispitiZadrana1` s
prethodne stranice!

stud	matBr	prez	ime	pbrSt
	1111	Novak	Ivan	10000
	1234	Kolar	Petar	21000

```
CREATE VIEW prosjek
  (matBr, prosOcj) AS
  SELECT matBr, AVG(ocj)
    FROM ispit
   GROUP BY matBr;
```

```
CREATE VIEW stud1 (ime_prez) AS
  SELECT ime || prez
    FROM stud;
```

```
CREATE VIEW poloziliNesto AS
  SELECT DISTINCT matBr
    FROM ispit
   WHERE ocj > 1;
```

Implementacija eksternih shema pomoću virtualnih tablica

- konceptualna shema: baza podataka u banci

klijent	jmbg	ime	prez	uplata	splata	brRac	vrijeme	valuta	iznos
	123456	Ana	Horvat			1001	7.8.2007 08:20	HRK	15.00
	654321	Ivan	Novak			1002	9.4.2006 12:31	EUR	-100.21
	123654	Tea	Kolar			1001	6.5.2007 14:15	HRK	452.15
racun	brRac	jmbgVI	tipRac	datRac		1004	5.5.2007 16:42	HRK	1200.00
	1001	123456	kunski	7.2.2007		1004	9.9.2005 10:15	HRK	-350.50
	1002	123456	devizni	1.3.2006		1002	7.2.2007 15:01	EUR	235.20
	1003	654321	devizni	4.8.2004		1003	1.4.2005 12:44	USD	2750.00
	1004	123654	kunski	8.9.2005		1001	1.9.2007 12:19	HRK	-250.35
						1004	8.2.2006 11:55	HRK	420.00

- za različite kategorije korisnika (aplikacija) definiraju se različite eksterne sheme
 - aplikacija za otvaranje računa
 - aplikacija za deviznu uplatu/isplatu
 - aplikacija za kunsku uplatu/isplatu
 - aplikacija za pregled trenutnog stanja sredstava u banci

Implementacija eksternih shema pomoću virtualnih tablica

```
CREATE VIEW devRacun AS
  SELECT * FROM racun
    WHERE tipRac = 'devizni'
  WITH CHECK OPTION;
```

```
CREATE VIEW devUplIspl AS
  SELECT * FROM uplataIsplata
    WHERE valuta <> 'HRK'
  WITH CHECK OPTION;
```

```
CREATE VIEW kunRacun AS
  SELECT * FROM racun
    WHERE tipRac = 'kunski'
  WITH CHECK OPTION;
```

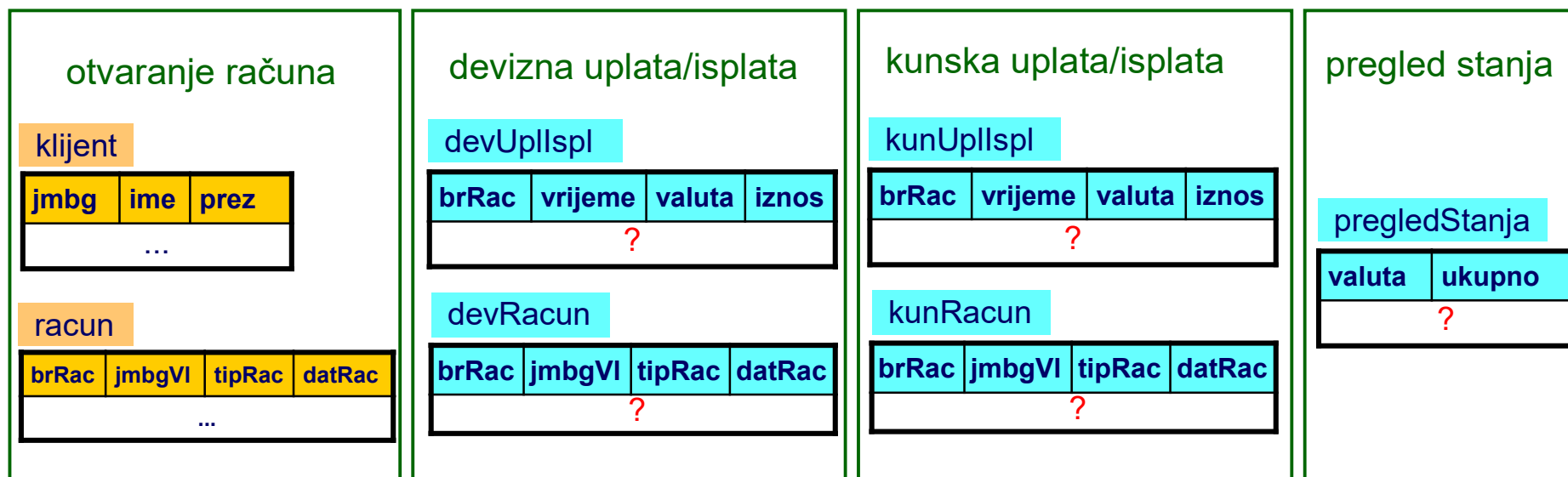
```
CREATE VIEW kunUplIspl AS
  SELECT * FROM uplataIsplata
    WHERE valuta = 'HRK'
  WITH CHECK OPTION;
```

```
CREATE VIEW pregledStanja
      (valuta, ukupno) AS
  SELECT valuta, SUM(iznos)
    FROM uplataIsplata
   GROUP BY valuta;
```

- eksterne sheme za aplikacije
 - za otvaranje računa: **klijent**, **racun**
 - za deviznu uplatu/isplatu: **devRacun**, **devUplIspl**
 - za kunsku uplatu/isplatu: **kunRacun**, **kunUplIspl**
 - za pregled trenutnog stanja sredstava: **pregledStanja**

Implementacija eksternih shema pomoću virtualnih tablica

eksterne sheme



preslikavanja: konceptualna ↔ eksterne sheme

konceptualna shema

