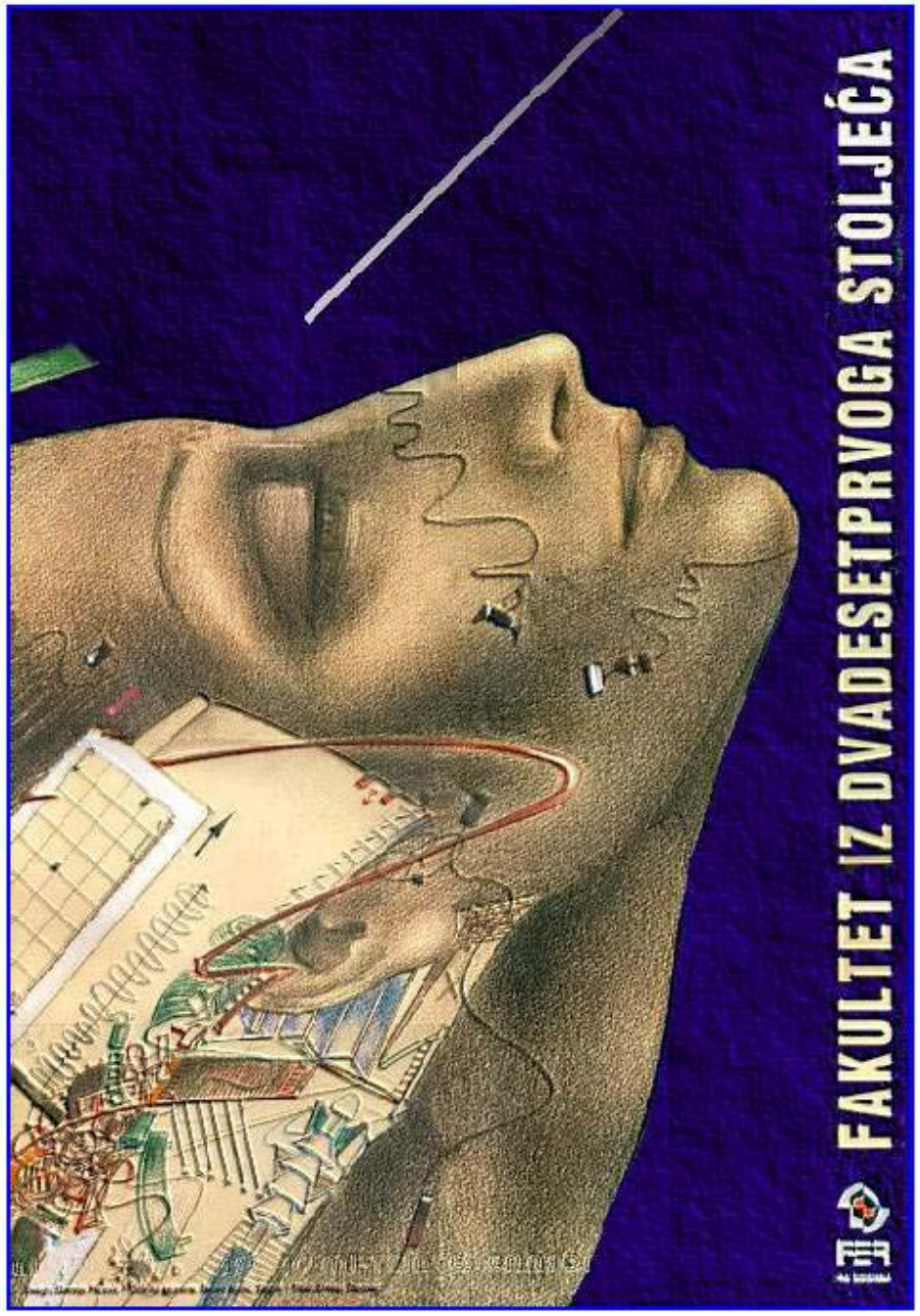
Baze podataka

Predavanja
svibanj 2008.

11. Privremene i virtualne relacije



Vrste relacija (tablica)

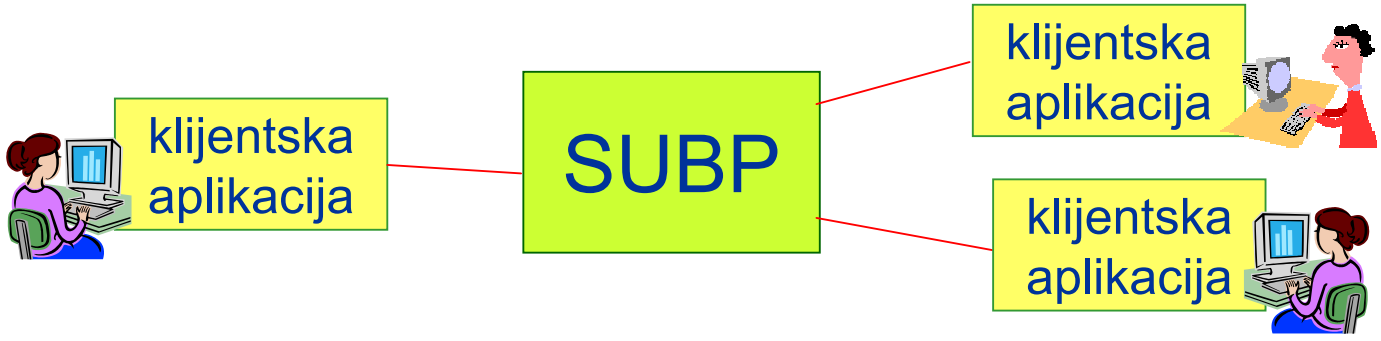
- **Temeljna relacija (*base relation*)**
  - relacija korespondentna skupu entiteta u konceptualnoj shemi, čija su shema i sadržaj trajno pohranjeni u bazi podataka
- **Privremena relacija (*temporary relation*)**
  - relacija čija su shema i sadržaj u bazu podataka pohranjeni privremeno
- **Virtualna relacija (*virtual relation, view*)**
  - relacija kojoj su shema i sadržaj definirani izrazom relacijske algebre čiji su operandi temeljne ili virtualne relacije
    - u praksi, shema i sadržaj virtualne relacije opisuju se u obliku SQL upita
  - sadržaj virtualne relacije dinamički se određuje u trenutku obavljanja operacije nad virtualnom relacijom: ovisi o trenutačnom stanju temeljnih relacija

Temeljna relacija

- Obavljanjem naredbe CREATE TABLE u rječnik podataka se pohranjuju metapodaci
  - naziv relacije
  - nazivi i tipovi atributa
  - integritetska ograničenja
  - ostali metapodaci (vrijeme kreiranja, vlasnik, primijenjena fizička organizacija, itd.)
- shema i sadržaj temeljne relacije 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 relacije (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 Aqua Data Studio ostvari vezu s IBM Informix sustavom za upravljanje bazama podataka
  - SQL-sjednica završava u trenutku kada korisnik prekine vezu (*disconnect*) prema sustavu za upravljanje bazama podataka



Privremena relacija

- Privremena relacija se kreira 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 relacija je u dosegu ("vidljiva je") isključivo u okviru SQL-sjednice tijekom koje je kreirana
  - "svaka SQL-sjednica koristi *svoje* privremene relacije"
- Privremene relacije se koriste kao pomoćni objekti, npr. za pohranu međurezultata pri obavljanju složenijih upita
  - zašto temeljne relacije nisu prikladne za tu namjenu?
- Privremena relacija se uklanja iz baze podataka:
  - obavljanjem naredbe DROP TABLE *nazivPrivremeneRelacije* ili
  - završetkom SQL-sjednice tijekom koje je ta privremena relacija kreirana

Primjer

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

- 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));
```

prosjek

sifPred	prosOcj

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

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

Za vježbu riješiti bez korištenja privremene relacije!

Primjer (nastavak)

- Treba primijetiti: sadržaj privremene relacije *prosjek* neće se "automatski" promijeniti nakon upisa još jedne n-torke u temeljnu relaciju *polozenIspit*

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

polozenIspit

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 relacije 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

Virtualna relacija (primjer)

- Problem "zastarijevanja" podataka u privremenim relacijama može se izbjeći uporabom virtualnih relacija

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 relacije *prosjek*

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

prosjek

sifPred	prosOcj
1001	4.00
1002	3.50
1003	3.00

minPros	maksPros
3.00	4.00

Primjer (nastavak)

- Sadržaj virtualne relacije se ponovno određuje pri izvršavanju svakog upita koji koristi tu virtualnu relaciju

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

```
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
102	1003	2

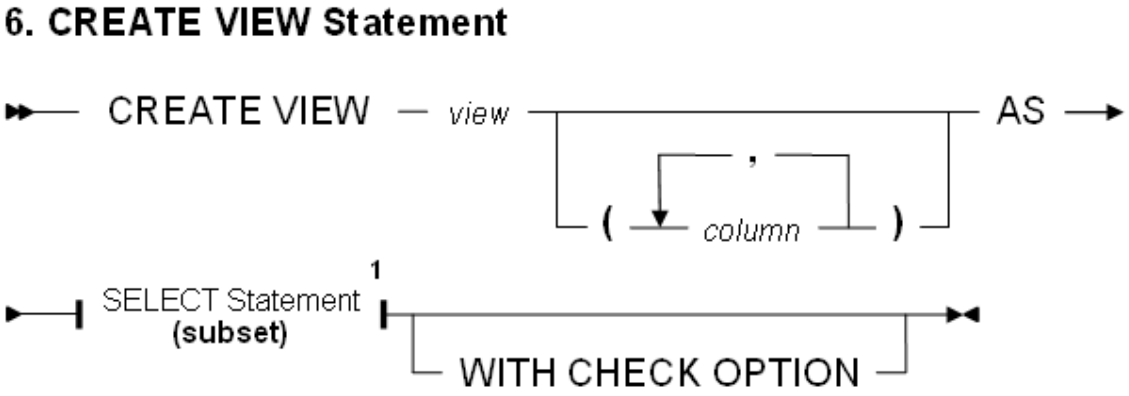
prosjek

sifPred	prosOcj
1001	4.00
1002	3.50
1003	2.67

minPros	maksPros
2.67	4.00

Naredba za kreiranje virtualne relacije

- Virtualna relacija se kreira naredbom CREATE VIEW



- *SELECT Statement* ***(subset)*** može sadržavati sve prethodno opisane dijelove SELECT naredbe osim
  - ORDER BY
  - FIRST n
- Virtualna relacija se uklanja iz baze podataka naredbom:
  - DROP VIEW *nazivVirtualneRelacije*;

Svojstva virtualne relacije

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

Atributi virtualne relacije

- Ukoliko se nazivi atributa u definiciji virtualne relacije ne navedu, nazivi atributa virtualne relacije određeni su nazivima atributa u SELECT naredbi kojom se definira sadržaj virtualne relacije
- tipovi podataka za atribute virtualne relacije proizlaze iz tipova podataka atributa temeljnih relacija koje se koriste u definiciji virtualne relacije

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 relacije

- Ukoliko se u listi za selekciju pri definiciji virtualne relacije koriste izrazi, nazivi atributa virtualne relacije se moraju 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 VIEW prosjek AS
  SELECT sifPred
       , AVG(ocj) AS prosOcj
    FROM polozeniIspit
    GROUP BY sifPred;
```

neispravno

Implementacija virtualnih relacija

- Kako sustavi za upravljanje bazama podataka izvršavaju upite koji sadrže virtualne relacije?
  - **modifikacijom upita**
    - SUBP ugrađuje elemente definicije virtualne relacije u originalni SQL upit koji koristi virtualnu relaciju - umjesto originalnog SQL upita izvršava se modificirani SQL upit
  - **korištenjem materijalizirane virtualne relacije**
    - SUBP fizički pohranjuje sadržaj virtualne relacije. Kada se promijeni sadržaj neke od temeljnih relacija pomoću kojih je virtualna relacija definirana, SUBP automatski mijenja i sadržaj materijalizirane virtualne relacije
    - prednost: virtualne relacije koje se vrlo često koriste, a čiji se sadržaj određuje složenim upitima, ne moraju se svaki puta kada neki korisnik koristi tu virtualnu relaciju ponovno izračunavati
    - nedostatak: ukoliko se temeljne relacije pomoću kojih je virtualna relacija definirana često mijenjaju, pri svakoj izmjeni temeljnih relacija troši se dodatno vrijeme radi izmjene sadržaja virtualne relacije

Implementacija virtualnih relacija 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)

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

- 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

Virtualna relacija: INSERT, UPDATE, DELETE

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

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

stud

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

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

```
SELECT * FROM splitStud;
```

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

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

```
SELECT * FROM splitStud;
```

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

n-torka jest unesena u temeljnu relaciju, ali se "ne vidi" u virtualnoj relaciji

```
SELECT * FROM stud;
```

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

Virtualna relacija: INSERT, UPDATE, DELETE

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

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 relacija: problem migrirajućih n-torki

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

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 t_2 je "nestala" iz *prosli* i "pojavila" se u *pali*
- nova n-torka <102, Elektronika, 1> unesena preko *prosli* se "pojavila" u *pali*

Virtualna relacija: problem migrirajućih n-torki

- **Rješenje:** virtualne relacije 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 relacije ukoliko n-torka nakon obavljanja operacije više ne bi pripadala virtualnoj relaciji 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 relacije

- SUBP ne može promijeniti "sadržaj virtualne relacije" - umjesto toga mora promijeniti sadržaj temeljnih relacija koje se koriste u definiciji te virtualne relacije
  - ako je virtualna relacija definirana tako da SUBP nije u stanju jednoznačno odrediti koje operacije treba obaviti na temeljnim relacijama, tada je virtualna relacija 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 relacije

- Virtualna relacija je izmjenjiva ukoliko u glavnom SELECT dijelu definicije virtualne relacije koristi atribute iz samo jedne temeljne relacije r(R) i pri tome ne sadrži:
  - eliminaciju duplikata pomoću DISTINCT
  - izraze u listi za selekciju (osim izraza koji sadrže samo ime atributa)
  - spajanje ili uniju
  - grupiranje i postavljanje uvjeta nad grupom (GROUP BY i HAVING)
- Prethodno navedena ograničenja se ne odnose na eventualne podupite koji se koriste unutar glavnog SELECT dijela definicije virtualne relacije, ali
  - podupiti ne smiju u svojem FROM dijelu koristiti relaciju r(R)

Primjeri izmjenjivih virtualnih relacija

ispit

matBr	sifPred	datIsp	ocj	sifNas
1111	1001	29.01.2006	1	101
1111	1001	05.02.2006	3	101
1111	1003	28.06.2006	2	303
1111	1002	27.06.2006	4	202
1234	1001	29.01.2006	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 poloziliBaremDva AS
  SELECT * FROM stud
    WHERE
     (SELECT COUNT(*) FROM ispit
       WHERE ispit.matBr = stud.matBr
        AND ocj > 1) >= 2
  WITH CHECK OPTION;
```

```
CREATE VIEW ispitiZadrana1 AS
  SELECT matBr
       , sifPred
       , datIsp
       , ocj
    FROM ispit
    WHERE matBr IN (
      SELECT matBr FROM stud
        WHERE pbrSt = 23000)
  WITH CHECK OPTION;
```

Primjeri neizmjenjivih virtualnih relacija

ispit

matBr	sifPred	datIsp	ocj	sifNas
1111	1001	29.01.06	1	1111
1111	1001	05.02.06	3	1111
1111	1003	28.06.06	2	3333
1111	1002	27.06.06	4	2222
1234	1001	29.01.06	3	2222

stud

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

```
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 relacijom `ispitiZadrana1` s prethodne stranice!

```
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;
```

```
CREATE VIEW boljiOdProsjeka AS
  SELECT * FROM ispit
    WHERE ocj >
      (SELECT AVG(ocj)
         FROM ispit);
```

Implementacija eksternih shema pomoću virtualnih relacija

- konceptualna shema: baza podataka u banci

klijent

jmbg	ime	prez
123456	Ana	Horvat
654321	Ivan	Novak
123654	Tea	Kolar

racun

brRac	jmbgVl	tipRac	datRac
1001	123456	kunski	7.2.2007
1002	123456	devizni	1.3.2006
1003	654321	devizni	4.8.2004
1004	123654	kunski	8.9.2005

uplataIsplata

brRac	vrijeme	valuta	iznos
1001	7.8.2007 08:20	HRK	15.00
1002	9.4.2006 12:31	EUR	-100.21
1001	6.5.2007 14:15	HRK	452.15
1004	5.5.2007 16:42	HRK	1200.00
1004	9.9.2005 10:15	HRK	-350.50
1002	7.2.2007 15:01	EUR	235.20
1003	1.4.2005 12:44	USD	2750.00
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 trenutačnog stanja sredstava u banci

Implementacija eksternih shema pomoću virtualnih relacija

```
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 trenutačnog stanja sredstava: **pregledStanja**

Implementacija eksternih shema pomoću virtualnih relacija

eksterne sheme

otvaranje računa

klijent

jmbg	ime	prez
...		

racun

brRac	jmbgVl	tipRac	datRac
...			

devizna uplata/isplata

devUplIspl

brRac	vrijeme	valuta	iznos
?			

devRacun

brRac	jmbgVl	tipRac	datRac
?			

kunska uplata/isplata

kunUplIspl

brRac	vrijeme	valuta	iznos
?			

kunRacun

brRac	jmbgVl	tipRac	datRac
?			

pregled stanja

pregledStanja

valuta	ukupno
?	

preslikavanja: konceptualna ↔ eksterne sheme

konceptualna shema

racun

brRac	jmbgVl	tipRac	datRac
...			

klijent

jmbg	ime	prez
...		

uplataIsplata

brRac	vrijeme	valuta	iznos
...			