

# BAZE PODATAKA - šalabahter

---

## privremene i temeljne tablice

---

### TEMELJNE TABLICE

doslovno one najnormalnije tablice od prije

### PRIVREMENE TABLICE

postoji samo tijekom jedne SQL sjednice

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

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

ovo gore se može i pomoću jedne naredbe

```
CREATE TEMP TABLE prosjek (sifPred,prosOcj)  
  
AS --u zagradu stavimo one attribute koje ćemo koristiti u upitu  
  
SELECT sifPred, AVG(ocj) AS prosOcj  
FROM polozeniIspit  
GROUP BY sifPred;
```

ako mijenjamo temeljne tablice, privremene tablice koje se temelje nad tim temeljnim s

### VIRTUALNE TABLICE

stvaranje tablice

```
CREATE VIEW prosjek (sifra, prosjek) AS
```

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

upit nad virtualnom tablicom

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

ako mijenjamo temeljne tablice, virtualne koje se oslanjaju na te temeljne će se autor

uklanjanje iz baze podataka

```
DROP VIEW nazivVirtualneTablice;
```

virtualna tablica je \_trajno\_ pohranjena u bazi podataka, vidljiva u svim SQL sjednicar

migrirajuće n-torke

recimo da postoje virtualne tablice

```
CREATE VIEW jedinice AS  
SELECT mbr, ocjena  
WHERE ocjena = 1  
FROM ocjeneIzFizike;
```

```
CREATE VIEW dvojke AS  
SELECT mbr, ocjena  
WHERE ocjena = 2  
FROM ocjeneIzFizike;
```

i ako sad napišemo naredbu

```
UPDATE jedinice  
SET ocjena = 2  
WHERE mbr = 1000;
```

i sad će zapis 1000 2 nestati iz tablice jedinice i pobjeći u tablicu dvojke

kako bismo to izbjegli dodajemo liniju WITH CHECK OPTION u svaku create naredbu virtualne tablice koju planiramo mijenjati (UPDATE, INSERT, DELETE)

```
CREATE VIEW jedinice AS  
SELECT mbr, ocjena  
WHERE ocjena = 1  
FROM ocjeneIzFizike  
WITH CHECK OPTION;
```

i sad će gornja naredba UPDATE jedinice... baciti grešku

FACT: ako updateamo virtualnu updeateat ćemo i pripadnu temeljnu!!!

uvjeti IZMJENJIVOSTI virtualne tablice

- u glavnom SELECT dijelu definicije virtualne tablice koristi atribut iz samo jedne temeljne tablice r(R)
- 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

## Transakcije i obnova baze podataka u slučaju razrušenja

### TRANSAKCIJA

Jedinica rada nad bazom podataka. BEGIN WORK = početak transakcije

COMMIT WORK = uspješan završetak

**ROLLBACK WORK** = neuspješan završetak, poništavanje svih izmjena koje je transakcija obavila

svojstva:

1. **ATOMARNOST**: Korisnik mora biti siguran da je zadatak obavljen potpuno i samo jednom (ili ništa nije obavljeno)
2. **IZDRŽLJIVOST**: Bez obzira u kojem se trenutku nakon potvrđivanja transakcije dogodio kvar, sustav mora osigurati da su njezini efekti trajno pohranjeni

o njima (1. i 2.) brine **Podsustav za upravljanje transakcijama i obnovu baze podataka u slučaju razrušenja brine**

---

3. **KONZISTENTNOST**

4. **ATOMARNOST**

o njima (3. i 4.) brine **Podsustav za upravljanje istodobnim pristupom**

---

## **OBNOVA BAZE PODATAKA**

Velike baze podataka – dijeljene, višekorisničke – nužno moraju posjedovati mehanizme obnove

**REDUNDANCIJA**- svaki se podatak mora moći rekonstruirati iz nekih drugih informacija

...ovo dalje nema smisla