



Računarstvo i informatika

Katedra za računarstvo

Elektronski fakultet u Nišu

Baze podataka (Računske vežbe) SQL naredbe za rad sa pogledima i indeksima

Letnji semestar 2017/2018



Sadržaj

- Pogledi
- Indeksi



Pogledi

- Pogled predstavlja uskladišteni SQL upit.
- Pogled se tretira kao virtuelna tabela koja sadrži podatke koji predstavljaju rezultat izvršavanja uskladištenog upita.
- Pogled ne postoji fizički odnosno podaci se ne dupliraju.
- Prilikom svakom pristupanja pogledu njegovi podaci se formiraju dinamički izvršavanjem uskladištenog upita.

- Naredba za kreiranje pogleda:

```
CREATE VIEW <ime_pogled>
```

```
AS <upit>;
```

- Naredba za brisanje pogleda:

```
DROP VIEW <ime_pogled>;
```



Pogledi

- U nastavku je data SQL naredba koja kreira pogled koji sadrži imena i prezimena radnika, imena projekata na kojima rade kao i broj radnih časova koje tokom nedelje provode na tom projektu.

```
CREATE VIEW RADNIK_PROJEKAT  
AS SELECT LIME, PREZIME, NAZIV, SATI  
FROM RADNIK, PROJEKAT, RADI_NA  
WHERE MATBR = MBR AND BRPR = BROJPR;
```

- Prikazati podatke iz kreiranog pogleda.

```
SELECT * FROM RADNIK_PROJEKAT;
```

LIME	PREZIME	NAZIV	SATI
Jovan	Obradović	Informacioni sistem	10
Marko	Petrović	ProizvodX	32
Marko	Petrović	ProizvodY	7
Velibor	Jovanović	ProizvodZ	40
Jelena	Janković	ProizvodX	20
Jelena	Janković	ProizvodY	20
Sima	Todorović	ProizvodY	10
Sima	Todorović	ProizvodZ	10
Sima	Todorović	Reorganizacija	10
Sima	Todorović	Informacioni sistem	10
Valentina	Kovačević	Godišnji izveštaj	30
Valentina	Kovačević	Reorganizacija	10
Stanko	Manojlović	Reorganizacija	25
Stanko	Manojlović	Informacioni sistem	5
Aleksandra	Petrović	Godišnji izveštaj	20
Aleksandra	Petrović	Informacioni sistem	15



Pogledi

- Kreirati pogled SEK_INFO koji sadrži informacije o nazivu sektora, o broju radnika koji rade u sektoru i prosečnoj zaradi radnika u sektoru.

```
CREATE VIEW SEK_INFO
AS SELECT NAZIV NAZIV_SEKTORA,
        COUNT(*) BROJ_RADNIKA,
        AVG(PLATA) PROSECNA_PLATA
FROM SEKTOR, RADNIK
WHERE BRSEK = SBROJ
GROUP BY NAZIV;
```

NAZIV_SEKTORA	BROJ_RADNIKA	PROSECNA_PLATA
Administracija	3	31000
Uprava	1	55000
Razvoj	4	32750



Pogledi

- Ekvivalentna SQL naredba za prethodni primer je:

```
CREATE VIEW SEK_INFO (NAZIV_SEKTORA, BROJ_RADNIKA,  
    PROSECNA_PLATA)
```

```
AS SELECT NAZIV, COUNT(*), AVG(PLATA)
```

```
FROM SEKTOR, RADNIK
```

```
WHERE BRSEK = SBROJ
```

```
GROUP BY NAZIV;
```

- **Ukoliko SQL upit sadrži kolone čija vrednost se računa, u pogledu im je neophodno dodeliti alijase.**



Pogledi

- Pogledi se mogu koristiti za ažuriranje podataka u tabelama koje se nalaze u pozadini (u FROM klauzuli uskladištenog upita).
- Da bi ažuriranje podataka korišćenjem tabela bilo moguće, DBMS mora da bude u stanju da mapira strukturu pogleda na strukturu tabela u pozadini (kolone i vrste).
- Za ažuriranje podataka korišćenjem pogleda koriste se standardne SQL naredbe za ažuriranje podataka: INSERT...INTO, UPDATE...SET, DELETE FROM.
- Ne može se ažurirati vrednost kolona pogleda koje su dobijene kao rezultat SQL funkcija (aritmetičkih funkcija, funkcija za rad sa stringovima ili funkcija agregacije).



Pogledi

- Kreirati pogled RAD_SEK_I koji sadrži matične brojeve, puna imena i plate svih radnika koji rade u sektoru I.

```
CREATE VIEW RAD_SEK_I
```

```
AS SELECT MATBR, LIME || ' ' || PREZIME PUNO_IME, PLATA
```

```
FROM RADNIK
```

```
WHERE BRSEK = I;
```

- Kreirani pogled se može iskoristiti za ažuriranje plate radnika.

```
UPDATE RAD_SEK_I
```

```
SET PLATA = I.I * PLATA;
```

- **Kreirani pogled se ne može iskoristiti za ažuriranje podataka o punom imenu radnika.**



Pogledi

- Dobre strane korišćenja pogleda:
 - Pogled može da sadrži samo podskup podataka iz tabele
 - Pogled može da spoji podatke iz većeg broja tabela u jednu virtuelnu tabelu
 - Pogled može da sadrži podatke koji predstavljaju rezultat izvršavanja funkcija (aritmetičke funkcije, funkcije za rad sa stringovima, funkcije agregacije).
 - Pogledi mogu da sakriju kompleksnost podataka
 - Povećava se bezbednost baze podataka
 - Sakrivaju detalje o šemi baze podataka od korisnika.



Indeksi

- Indeks je struktura podataka koja poboljšava performanse pretraživanje u relacionoj bazi podataka.
- Indeksi se najčešće implementiraju kao stabla traženja.
- Indeksi se mogu kreirati nad jednom ili većim brojem kolona tabele relacione baze podataka.
- Indeks sadrži kopiju vrednosti kolona nad kojima se kreira (ključevi indeksa).
- Indeks fizički postoji odnosno indeks zauzima prostor na disku.
- Pojedini DBMS-ovi omogućavaju da se indeksi kreiraju i nad rezultatima operacija nad kolonama tabela.
- Indeksi se (u opštem slučaju) ne mogu kreirati nad pogledima.



Indeksi

- Primarni ključ tabele je automatski indeksiran.
- Obično se indeksiraju strani ključevi u tabeli (da bi se ubrzale operacije spajanja tabela) i kolone (ili grupe kolona) koje se koriste za postavljanje uslova u upitima.
- Indeksi ubrzavaju operacije pretraživanja ali mogu da uspore operacije ažuriranja podataka. Zbog toga je potrebno pravilno izbalansirati indekse u bazi podataka.
- U SELECT naredni nije potrebno posebno specificirati indekse. DBMS sam određuje koje indekse treba koristiti za optimalno izvršavanje upita.



Indeksi

- Sintaksa SQL naredbe za kreiranje indeksa:

```
CREATE [UNIQUE] INDEX <ime_indeksa>  
ON {ime_tabele (ime_kolone[ASC | DESC]  
                [, ime_kolone[ASC | DESC]]...)};
```

- Opcija UNIQUE se koristi kada se kreira indeks nad primarnim ključem;
- Sintaksa SQL naredbe za brisanje indeksa:

```
DROP INDEX <ime_indeksa>;
```



Indeksi

- SQL naredba koja kreira indeks nad kolonom lokacija projekta.

```
CREATE INDEX PROJEKAT_LOK_IND ON PROJEKAT(LOKPR);
```

- SQL naredba koja kreira indeks nad kolonama ime i prezime radnika projekta.

```
CREATE INDEX RADNIK_IME_IND ON RADNIK(LIME DESC, PREZIME ASC);
```

- SQL naredba koja kreira indeks za potrebe primarnog ključa (**prilikom kreiranja ograničenja primarnog ključa DBMS automatski kreira i UNIQUE indeks**).

```
CREATE UNIQUE INDEX RADNIK_MATBR_IND ON RADNIK(MATBR);
```