Fakultet organizacije i informatike, Varaždin

Vježba 6

Prof. dr. sc. Kornelije Rabuzin

Sadržaj

Priprema za vježbu 6

- U knjizi „Uvod u SQL” proučite poglavlja 8, 10.1 i 10.7

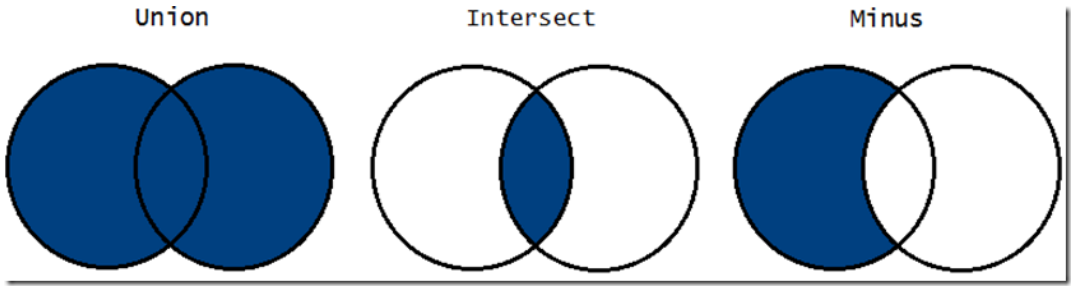
Skupovni upiti

- Povezivanje relacija u upitu skupovnim operacijama (unija, presjek, razlika)
- SQL klauzule: UNION, INTERSECT, EXCEPT (MINUS) (ovisno o SUBP-u)
  - Povezivanje više upita ili relacija **s istim brojem i tipom argumenata** u odgovoru

Sintaksa:

```
SELECT ...
UNION / INTERSECT / EXCEPT [ALL | DISTINCT]
SELECT ...
```

Vraća duplikate, ali brži



http://sqlblog.com/blogs/dejan_sarka/image_thumb_2B1B460F.png

Primjeri upita (UNION)

1) Napišite upit koji će vraćati prezimena i imena svih autora i članova iz baze:

```
SELECT prezime, ime FROM autor
UNION
SELECT prezime, ime FROM clan;
```

2) Napišite upit koji će vraćati prezimena i imena svih autora i članova iz baze sortiranih silazno po prezimenu, ali neka u odgovoru nema više od 4 n-torke:

```
SELECT prezime, ime FROM autor
UNION
SELECT prezime, ime FROM clan
ORDER BY prezime DESC
LIMIT 4;
```

3) Napišite upit koji vraća ukupan broj knjiga i broj 10:

```
SELECT COUNT(*) FROM knjiga
UNION
SELECT 10;
```

Primjeri upita (INTERSECT i EXCEPT)

Napišite upit koji će vraćati ista prezimena koja se javljaju u tablicama **autor** i **clan**:

```
SELECT prezime FROM autor
INTERSECT
SELECT prezime FROM clan;
```

Napišite upit koji će vraćati prezimena koja se javljaju u tablici **autor**, ali ne i u tablici **clan**:

```
SELECT prezime FROM autor
EXCEPT
SELECT prezime FROM clan;
```

Pogledi

- Virtualne nematerijalizirane relacije ili pohranjeni upiti
- Pogled se koristi za:
  - Ograničavanje pristupa specifičnim retcima ili stupcima
  - Spajanje stupaca iz više tablica tako da izgledaju kao jedna tablica
  - Agregiranje podataka (npr. za izvještaje)
- Smanjena redundancija u pisanju upita
- Materijalizirani pogledi – *snapshot* upita radi boljih performansi

```
CREATE [ OR REPLACE] [TEMP | TEMPORARY] VIEW name [(colum_name [,..])]
AS query
```

Brisanje pogleda:

```
DROP VIEW name;
```

PRIMJER:

```
CREATE VIEW v1 AS SELECT ime, prezime FROM clan;

SELECT * FROM v1;
```

Primjeri pogleda

Kreirajte pogled na temelju upita koji vraća broj posudbi prema datumu posudbe:

```
CREATE VIEW pos1 AS
SELECT datum_posudbe, COUNT(*) FROM posudba
GROUP BY 1;
```

```
SELECT * FROM pos1;
```

Kreirajte pogled na temelju upita koji vraća broj posudbi prema datumu posudbe, ali samo ako je taj broj minimalno 2:

```
CREATE OR REPLACE VIEW pos2 AS
SELECT datum_posudbe, COUNT(*) FROM posudba
GROUP BY 1 HAVING COUNT(*)>1;
```

```
SELECT * FROM pos2;
```

Primjeri pogleda

Kreirajte pogled koji vraća sve osobe koje postoje u bazi, bilo da se radi o autorima ili članovima:

```
CREATE VIEW osobe AS
SELECT prezime, ime FROM autor
UNION
SELECT prezime, ime FROM clan;
```

Kreirajte pogled koji vraća ime i prezime člana, ime i prezime autora, naziv knjige koja je posuđena, naziv izdavača i datum posudbe.

ALTER TABLE

- Naredba za dodavanje, brisanje i promjenu stupaca i ograničenja u postojećoj tablici → promjene u strukturi tablice

```
ALTER TABLE [ ONLY ] name [*] {
ADD [ COLUMN ] column type [ column_constraint [...]] |
DROP [ COLUMN ] column [RESTRICT | CASCADE] |
ALTER [ COLUMN ] column { SET DEFAULT expression | DROP DEFAULT } |
ALTER [ COLUMN ] column { SET | DROP } NOT NULL | SET WITHOUT OIDS |
RENAME [ COLUMN ] column TO new_column
ADD table_constraint |
DROP CONSTRAINT constraint_name [ RESTRICT | CASCADE ]
}
```

Primjeri (ALTER TABLE)

Kreirajte privremenu tablicu „privremena” s 2 atributa; *sifra VARCHAR(50) s imenovanim ograničenjem NOT NULL*, te *naziv VARCHAR(50) DEFAULT NULL*:

```
CREATE TEMP TABLE privremena (
                sifra VARCHAR(50) CONSTRAINT sif NOT NULL,
                naziv VARCHAR(50) DEFAULT NULL
);
```

Preimenujte tablicu „privremena” u „temp”:

```
ALTER TABLE privremena RENAME TO temp;
```

Dodajte tablicu „temp” stupac „opis” (VARCHAR(50)), s tim da postoji imenovano ograničenje da sve vrijednosti tog stupca budu jedinstvene:

```
ALTER TABLE temp
ADD COLUMN opis VARCHAR(50) CONSTRAINT c1 UNIQUE;
```

Primjeri (ALTER TABLE)

Promijenite naziv stupca „opis” u „Opis proizvoda”:

```
ALTER TABLE temp
RENAME COLUMN opis TO „Opis proizvoda”;
```

Maknite ograničenje UNIQUE nad stupcem „Opis proizvoda”:

```
ALTER TABLE temp
DROP CONSTRAINT c1;
```

Obrišite stupac „Opis proizvoda”:

```
ALTER TABLE temp
DROP COLUMN „Opis proizvoida”;
```

Dodajte imenovano ograničenje na razini tablice kojim stupac „sifra” postaje primarni ključ:

```
ALTER TABLE temp
ADD CONSTRAINT pk1 PRIMARY KEY(sifra);
```

Zadatak

- Kreirajte tablicu proizvoljnog naziva s minimalno 3 atributa i neka u tablici postoji barem jedno imenovano ograničenje
- Isprobajte sve mogućnosti naredbe ALTER TABLE
- Posebno kreirajte jednu novu tablicu te naredbom ALTER TABLE u istu dodajte vanjski ključ

Transakcije

- Niz operacija (naredbi) koje se izvršavaju kao jedna logička jedinica posla (princip „sve ili nijedna”)
- Npr. bankovne transakcije
- 4 svojstva (ACID):
  1. Atomnost (eng. Atomic) - izvrše se sve naredbe unutar transakcije ili se ne izvrši nijedna
  2. Konzistentnost (eng. Consistent) – nakon završene transakcije BP ostaje u konzistentnom stanju
  3. Izoliranost (eng. Isolated) – svaka transakcija se izvršava zasebno i vidi BP u konzistentnom stanju
  4. Trajnost (eng. Durable) – rezultati transakcije ostaju trajno pohranjeni u BP

Transakcije

- Transakcija započinje naredbom **BEGIN**; nakon toga dolaze preostale naredbe sve do naredbe **COMMIT** ili **ROLLBACK (ABORT):**

Transakcijski blok

```
BEGIN;
...
SQL naredbe
...
COMMIT | ROLLBACK (ABORT);
```

- Temeljni koncept SUBP-ova
- Svaka naredba je transakcija (implicitna), a pomoću BEGIN...COMMIT se kreiraju eksplicitne transakcije

Primjeri transakcija

1

```
SELECT * FROM autor;

BEGIN;
   INSERT INTO autor   VALUES
(DEFAULT, ‘Stephen’, ‘King’);
   SELECT * FROM autor,
ROLLBACK (ili ABORT);

SELECT * FROM autor;
```

2

```
SELECT * FROM autor;

BEGIN;
   INSERT INTO autor VALUES (DEFAULT,
‘Stephen’, ‘King);
   INSERT INTO autor VALUES (DEFAULT,
‘David’, ‘Maier’);
COMMIT;

SELECT * FROM autor;
```

Koliko će redaka vratiti SELECT naredba nakon završene transakcije u oba slučaja?

Popunite tablicu posudba s 3 sloga koristeći jednu transakciju.