



Seminarski rad iz kolegija
Baze podataka:

SQL

Studenti:

Vojislav Đuračković

Ivan Jelić

Boris Dokić

UVOD

- SQL – “Structured Query Language”
- SQL -je jezik baze podataka računala dizajniran za upravljanje podacima u relacijskim sustavima za upravljanje bazom podataka (RDBMS), a izvorno se temelji na Relacijskoj algebri
- Svrstavamo ga u neproceduralne programske jezike jer za razliku od npr. programskog jezika C opisuje što želimo dobiti kao rezultat, a ne kako doći do njega.

SQL kroz povijest

Povijest SQL-a seže u 1970-e, kada je grupa iz IBM-ovog istraživačkog centra u San Joseu-u razvila sustav baze podataka "System R" temeljen na Coddovu modelu

Prvotni naziv za SQL je bio Structured English Query Language ("SEQUEL") i bio dizajniran da manipulira i vadi podatke spremljene u System R

Naziv SEQUEL je kasnije preimenovan u SQL zbog toga što je riječ 'SEQUEL' bila zaštitni znak jedne zrakoplovne tvrtke u Ujedinjenom Kraljevstvu.

- Iako je SQL bio inspiriran Coddovim radom, IBMovci Donald D. Chamberlin i Raymond su postali autori SEQUEL jezičnog dizajna.
- Jezik je slabo razvijen do 80-tih godina. (nekomercijalna primjena)
- 1979. prvi komercijalni sustav temeljen na SQL jeziku
- 1986. zbog pojave raznih “dijalekata” donesen je ISO/ANSI standard za SQL (zadnja verzija 1998. godine).
- do 1989. godine 70-ak različitih verzija SQL sustava

- 1989. godine proširenje standarda (uvođenje referencijskog integriteta) pa nastaje SQL-89
- 1992. godine SQL-2 ili SQL-92 – proširenje standarda u pisanoj formi (broj stranica)
- 1999. godine objavljen SQL-3 ili SQL-99 s novim mogućnostima
- Jezik se postepeno usavršavao, a njegova dotjerana varijanta pojavljuje se u današnjem IBM-ovom relacijskom DBMS-u zvanom DB2
- Neke softverske kuće su ugradile SQL u svoje DBMS-e, a neke su pokušale razviti svoje jezike.

- SQL je uglavnom zasnovan na relacijskom računu, s time da je matematička notacija zamijenjena ključnim riječima nalik na govorni engleski jezik.
- pristup podacima se odvija prema modelu
klijent/poslužitelj
- SQL podržava ugnježdavanje –upite unutar upita
- Podupit je izraz tipa select - from- where koji se nalazi unutar drugog upita
- Obično se koriste za provjeru članstva u skupu, usporedbu skupova i izračun kardinalnosti skupova skupova

Podjela naredbi:

1) naredbe za četiri osnovne operacije nad podacima u bazi podataka:

- pronalaženje (SELECT) n-torki koje zadovoljavaju zadane uvjete

- upis (INSERT) novih n-torki

- brisanje ili izbacivanje (DELETE) podataka n-torki uz zadane uvjete

- promjena (UPDATE) podataka n-torki uz zadane uvjete

2) naredbe za definiranje relacijske sheme baze podataka, opis ograničenja u bazi podataka i definiranje pogleda na bazu podataka

3) naredbe za upravljanje bazom podataka (uspostavljanje indeksa, kontroliranje pristupa bazi i sl)

Prema standardu (donosi sintaksu i semantiku) jezik je podijeljen na dva podjezika:

1) podjezik za definiranje baze podataka (DDL - Data Definition Language)

Naredbe: SELECT, INSERT, DELETE, UPDATE
COMMIT i ROLLBACK

2) podjezik za upravljanje podacima u bazi (DML - Data Manipulation Table)

Naredbe: CREATE TABLE, ALTER TABLE, DROP TABLE
CREATE VIEW, DROP VIEW
CREATE INDEKS, DROP INDEKS
GRANT i REVOKE

TRGOVAC

MB	IME_PREZIME	DATUM_RODJENJA	MB_VG	PLACA	DODATAK
1	Ivan Perić	1948-11-26	1	120	0
3	Ana Horvat	1952-03-19	11	110	10
5	Josip Antić	1959-09-13	1	80	30
9	Ante Ivić	1962-05-06	1	100	45
11	Maja Markić	1939-12-01	11	120	0

PARTNER

PARTNER_SIFRA	PARTNER_NAZIV	PARTNER_ADRESA	MB
10	Zagreb d.o.o.	Zagreb	3
20	Horizont d.d.	Sesvete	5
30	Josip Ferić	D. Selo	11
40	A-banka d.d.	Zagreb	?
50	Brzopromet	Zagreb	1
60	Vicko Stić	Split	9
70	Kompakt d.o.o.	Osijek	5
80	Hotel F	Rijeka	9
90	Amalija d.d.	Pula	9

CREATE TABLE

- SQL-DDL naredba za kreiranje tablica
- Opći oblik:


```
CREATE TABLE "IME TABLICE"  
( „IME STUPCA1“ TIP PODATKA,  
  „IME STUPCA2“ TIP PODATKA,  
  
  „IME STUPCAN“ TIP PODATKA );
```

- Tipovi podataka mogu biti
 - ZNAKOVNI (char, varchar,..),
 - NUMERIČKI (int, float, double, numeric(p,q)...),
 - DATUMSKI (date, time, datetime, smalldatetime)

- Osim tipa podatka, uz deklaraciju svakog stupca mogu se naći i **ograničenja**:
 - UNIQUE-jedinstvenost vrijednosti (PRIMARY KEY)
 - NOT NULL-atribut mora sadržavati neku vrijednost
 - FOREIGN KEY-osigurava referencijalni integritet; preuzima vrijednosti drugog atributa
 - CHECK-provjerava se da li je podatak iz zadanog intervala, odnosno da li zadovoljava određene uvjete

- Sve naredbe ćemo razmatrati na primjeru dvije tablice TRGOVAC i PARTNER
- Tablica PARTNERI sadrži poslovne partnere neke tvrtke, a tablica TRGOVCI sve trgovce koji su dodijeljeni nekom poslovnom partneru (može se dogoditi da jedan trgovac bude dodijeljen više od jednom poslovnom partneru)
- Također svaki trgovac ima svog voditelja grupe, a to je također neko od svih navedenih trgovaca

❑ CREATE TABLE TRGOVAC (
MB NUMERIC(4) UNIQUE,
IME_PREZIME VARCHAR(30) NOT NULL,
DATUM_RODZENJA DATE,
MB_VG NUMERIC(4),
PLACA NUMERIC(5,2) CHECK(PLACA
BETWEEN 80 AND 250),
DODATAK NUMERIC(5,2),
PRIMARY KEY (MB),
FOREIGN KEY (MB_VG) REFERENCES TRGOVAC
(MB));



```
❑ CREATE TABLE PARTNER (  
    PARTNER_SIFRA INT NOT NULL,  
    PARTNER_NAZIV VARCHAR(20) NOT  
        NULL UNIQUE,  
    PARTNER_ADRESA VARCHAR(50),  
    MB NUMERIC(4),  
    PRIMARY KEY(PARTNER_SIFRA),  
    FOREIGN KEY (MB) REFERENCES  
        TRGOVAC (MB));
```

ALTER TABLE

- SQL-DDL naredba za dodavanje novog atributa u tablicu
- Za uvođenje novih ograničenja na postojeći atribut
- Za promjenu tipa podataka nekog atributa

Primjer-*dodavanje novog atributa u tablicu*

❑ ALTER TABLE TRGOVAC

ADD (ODJEL_BROJ NUMERIC(3));

- na ovaj smo način u tablicu trgovac dodali atribut ODJEL_BROJ

- **Primjer**-povećanje formata atributa *IME_PREZIME* s 30 na 35 znakova

- ❑ ALTER TABLE TRGOVAC

```
MODIFY (IME_PREZIME VARCHAR(35)  
NOT NULL));
```

- **Primjer**-dodavanje ograničenja na postojeći atribut

- ❑ ALTER TABLE TRGOVAC

```
ADD CONSTRAINT (DATUM_RODJENJA DATE  
NOT NULL);
```

DROP TABLE

- SQL-DDL naredba
- Služi za brisanje tablice
- Osim tablice i njenog sadržaja, ova naredba briše i sve relacije koje je ona imala unutar sebe i s drugim tablicama
- **Primjer**
- ❑ DROP TABLE PARTNER;
- Na ovaj način brišemo tablicu PARTNER

SELECT

Naredba Select najvažnija je, ali i najkompleksnija, naredba.

Služi za selekciju i ispis podataka jednog upita. U SQL-u se mogu jednostavno izraziti sve operacije relacijske algebre.

Osnovni oblik SELECT naredbe jedt (dio naredbe oiviče [] nije obvezatan):

```
SELECT [ALL|DISTINCT] atribut [,atribut...]  
      FROM relacija [,relacija...]  
      [WHERE uvjet]  
      [GROUP BY atribut[,atribut...]]  
      [HAVING uvjet]  
      [ORDER BY specifikacija uređenja];
```

Tipičan SELECT upit ima oblik:

```
SELECT A 1,A2, ... ,An  
      FROM R1, R2, ... , Rn  
      WHERE uvjet;
```

Ri predstavlja relaciju, a Ai atribut. Ako se navodi više atributa istog naziva, ispred naziva se stavlja naziv relacije koji se odvaja točkom (R.A)

PRIMJERI:

a) ispis čitavog sadržaja jedne relacije: ispisati sadržaj relacije TRGOVAC

SELECT * FROM TRGOVAC;

MB	IME_PREZIME	DATUM_RODJENJA	MB_VG	PLACA	DODATAK
1	Ivan Perić	1948-11-26	1	120	0
3	Ana Horvat	1952-03-19	11	110	10
5	Josip Antić	1959-09-13	1	80	30
9	Ante Ivić	1962-05-06	1	100	45
11	Maja Markić	1939-12-01	11	120	0

b) ispis pojedinih atributa jedne relacije: ispisati attribute
IME_PREZIME, DATUM_RODJENJA

SELECT IME_PREZIME, DATUM_RODJENJA **FROM** TRGOVAC;

IME_PREZIME	DATUM_RODJENJA
Ivan Perić	1948-11-26
Ana Horvat	1952-03-19
Josip Antić	1959-09-13
Ante Ivić	1962-05-06
Maja Markić	1939-12-01

c) selekcija pojedinih n-torki jedne relacije – operacija selekcije

c1) ispisati poslovne partnere iz Zagreba:

```
SELECT * FROM PARTNER
```

```
WHERE PARTNER_ADRESA = 'Zagreb';
```

PARTNER_SIFRA	PARTNER_NAZIV	PARTNER_ADRESA	MB
10	Zagreb d.o.o.	Zagreb	3
40	A-banka d.d.	Zagreb	?
60	Brzopromet	Zagreb	1

Uz operator = mogu se koristiti i drugi theta-operatori, koji se pišu ovako:

< > (različito)

< (manje od)

> (veće od)

<= (manje ili jednako)

>= (veće ili jednako).

Složeniji upiti dobivaju se korištenjem logičkih operatora **AND**, **OR** ili **NOT**

c2) ispisati matični broj, ime i prezime trgovca kojim je voditelj trgovac s matičnim brojem I koji ima plaću veću od 99:

```
SELECT MB, IME_PREZIME  
FROM TRGOVAC  
WHERE MB_VG=I AND PLACA>99;
```

MB	IME_PREZIME
1	Ivan Perić
9	Ante Ivić

MB	IME_PREZIME	DATUM_RODJENJA	MB_VG	PLACA	DODATAK
1	Ivan Perić	1948-11-26	1	120	0
3	Ana Horvat	1952-03-19	11	110	10
5	Josip Antić	1959-09-13	1	80	30
9	Ante Ivić	1962-05-06	1	100	45
11	Maja Markić	1939-12-01	11	120	0

c3) ispisati naziv partnera i adresu partnera čiji naziv počinje slovom A:

```
SELECT PARTNER_NAZIV, PARTNR_ADRESA  
FROM PARTNER  
WHERE PARTNER_NAZIV LIKE 'A%';
```

PARTNER_NAZIV	PARTNER_ADRESA
A-banka d.d.	Zagreb
Amalija d.d.	Pula

Znak % ovdje zamjenjuje niz (0 ili više) bilo kojih znakova.

c4) ispisati šifru i naziv poslovnog partnera te adresu partnera kojemu nije raspoređen trgovac (to su n-torke koje u atributu MB relacije PARTNER imaju nul-vrijednost):

```
SELECT PARTNER_SIFRA, PARTNE_NAZIV, PARTNER_ADRESA  
FROM PARTNER  
WHERE MB IS NULL;
```

PARTNER_SIFRA	PARTNER_NAZIV	PARTNER_ADRESA
40	A-banka d.d.	Zagreb

- d) selekcija n-torki jedne relacije i uređeni ispis: ispisati matični broj, ime i prezime, matični broj vođitelja grupe i plaću teko da budu uređeni po rastućem redoslijedu matičnog broja vođitelja i padajućoj plaći:

```
SELECT MB, IME_PREZIME, MB_VG, PLACA  
FROM TRGOVAC  
ORDER BY MB_VG ASC, PLACA DESC;
```

MB	IME_PREZIME	MB_VG	PLACA
1	Ivan Perić	1	120
9	Ante Ivić	1	100
5	Josip Antić	1	80
11	Maja Markić	11	120
3	Ana Horvat	11	110

SQL ima više korisnih funkcija kao što su:

- **AVG** (atribut) – srednja vrijednost atributa
- **MAX** (atribut) – maksimalna vrijednost atributa
- **MIN** (atribut) – minimalna vrijednost atributa
- **SUM** (atribut) – zbroj vrijednosti atributa
- **COUNT (*)** – broj selektiranih n-torki
- **COUNT** (atribut) – broj nenul-vrijednosti u atributu.

e) izračunati minimalnu, srednju i maksimalnu plaću te broj n-torki odnosno trgovaca koji su sudjelovali u izračunu:

```
SELECT MIN (PLACA), AVG (PLACA), MAX (PLACA), COUNT (*)  
FROM TRGOVAC;
```

MIN (PLACA)	AVG (PLACA)	MAX (PLACA)	COUNT (*)
80	106	120	5

eI) izračunati minimalnu, srednju i maksimalnu plaću te broj n-torki odnosno broj trgovaca koji su sudjelovali u izračunu; izračun načiniti po grupama:

```
SELECT MIN (PLACA), AVG (PLACA), MAX (PLACA), COUNT (*), MB_VG  
FROM TRGOVAC  
GROUP BY MB_VG;
```

MIN (PLACA)	AVG (PLACA)	MAX(PLACA)	COUNT(*)	MB_VG
80	100	120	3	1
110	115	120	2	11

GROUP BY određuje kriterij grupiranja.

Ovdje je to atribut MB_VG, točnije njegove vrijednosti.

e2) izračunati minimalnu, srednju i maksimalnu plaću te broj trgovaca koji su sudjelovali u izračunu; izračun načiniti po grupama u kojima radi više od dva trgovca:

```
SELECT MIN (PLACA) "Minimalna",  
        AVG (PLACA) "Srednja",  
        MAX (PLACA) "Maksimalna",  
        COUNT (*) "Broj u grupi",  
        MB_VG  
FROM TRGOVAC  
GROUP BY MB_VG  
HAVING COUNT (*) >2;
```

Naziv stupca u rezultirajućoj relaciji možemo posebno nasloviti tekstom iza imena atributa.

Minimalna	Srednja	Maksimalna	Broj u grupi	MB_VG
80	100	120	3	1

MB	IME_PREZIME	DATUM_RODJENJA	MB_VG	PLACA	DODATAK
1	Ivan Perić	1948-11-26	1	120	0
3	Ana Horvat	1952-03-19	11	110	10
5	Josip Antić	1959-09-13	1	80	30
9	Ante Ivić	1962-05-06	1	100	45
11	Maja Markić	1939-12-01	11	120	0

- f) selekcija n-torke jedne relacije uz korištenje podupita (selekcija u selekciji)
- f1) ispisati matični broj, ime i prezime trgovaca koji rade u grupi s Josipom Antićem:

```
SELECT MB, IME_PREZIME
FROM TRGOVAC
WHERE MB_VG=(SELECT MB_VG
FROM TRGOVAC
WHERE IME_PREZIME='Josip Antić');
```

MB	IME_PREZIME
1	Ivan Perić
5	Josip Antić
9	Ante Ivić

Korištenje podupita, može se izbjeći uz pomoć operacije spajanja relacija, tako bismo onda ovaj prethodni primjer riješili na sljedeći način:

```
SELECT DRUGA.MB, DRUGA.IME_PREZIME  
FROM TRGOVAC DRUGA, TRGOVAC PRVA  
WHERE PRVA.IME_PREZIME='Josip Antić' AND  
PRVA.MB_VG=DRUGA.MB_VG;
```

f2) ispisati naziv i adrese poslovnih partnera koji rade s trgovcima što su rođeni nakon 1957. godine:

```
SELECT PARTNER_NAZIV, PARTNER_ADRESA  
FROM PARTNER  
WHERE MB IN (SELECT MB FROM TRGOVAC  
WHERE DATUM_RODJENJA > '1957-12-31');
```

PARTNER_NAZIV	PARTNER_ADRESA
Horizont d.d.	Sesvete
Vicko Stić	Split
Kompakt d.o.o.	Osijek
Hotel F	Rijeka
Amalija d.d.	Pula

g) selekcija podataka n-torki iz više relacija – operacija spajanja

gl) ispisati naziv i adresu poslovnih partnera te ime i prezime trgovaca koji s njima rade:

```
SELECT PARTNER_NAZIV, PARTNER_ADRESA, IME_PREZIME  
FROM PARTNER, TRGOVAC  
WHERE PARTNER.MB=TRGOVAC.MB;
```

PARTNER_NAZIV	PARTNER_ADRESA	IME_PREZIME
Zagreb d.o.o.	Zagreb	Ana Horvat
Horizont d.d.	Sesvete	Josip Antić
Josip Ferić	Dugo Selo	Maja Markić
Brzopromet	Zagreb	Ivan Perić
Vicko Stić	Split	Ante Ivić
Kompakt d.o.o.	Osijek	Josip Antić
Hotel F	Rijeka	Ante Ivić
Amalija d.d.	Pula	Ante Ivić

U rezultatu nema poslovnih partnera kojima nije pridjeljen trgovac. Kako se one nalaze u n-torkama koje se ne mogu spojiti niti s jednom n-torkom trgovac, potrebno je upotrijebiti odgovarajuću opreciju vanjskog spajanja. U tom slučaju rješenje bi glasilo:

```
SELECT PARTNER_NAZIV, PARTNER_ADRESA, IME_PREZIME  
FROM PARTNER, TRGOVAC  
WHERE PARTNER.MB != TRGOVAC.MB;
```

Upotrijebljeno je lijevo vanjsko spajanje na jednakost, koje je označeno po uzoru na relacijsku algebru.

INSERT

Naredba INSERT služi za unos novih n-torki u relaciju, na jedan od dva načina:

- Upisom vrijednosti atributa i unosom jedne n-torke u relaciju
- Selektiranjem n-torki i njihovih atributa iz postojeće relacije (ili relacija) ili njihovim prijenosom u relaciju.

Osnovni oblik naredbe jest:

```
INSERT INTO relacija [(atribut[,atribut...])]  
  VALUES(vrijednost atributa[,vrijednost atributa...])|SELECT};
```

Unutar { } su znakom | odijeljene dvije varijante naredbe, od kojih treba izabrati jednu.

Naredbom INSERT najprije se formira n-torka s nul-vrijednostima svih atributa, a zatim unose upisane vrijednosti atributa prema upisanoj listi atributa.

a) unos vrijednosti atributa

a1) unijeti podatke novog poslovnog partnera s brojem 95, nazivom Marketing i adresom Zagreb te zasada nepoznatim trgovcem:

INSERT INTO PARTNER

VALUES (95, 'Marketing','Zagreb',NULL);

a2) unijeti podatke novog trgovca: matični broj 19, Marko Ferić, rođen 27.11.1956.:

INSERT INTO

TRGOVAC(MB,IME_PREZIME,DATUM_RODZENJA)

VALUES (19, 'Marko Ferić','1956-11-27');

b) unos n-torke (ili n-torki) selektiranjem: upisati podatke trgovca iz relacije TRGOVAC, rođenih prije '1950-01-01', u relaciju STARIJI_TRGOVAC (MB, IME_PREZIME):

```
INSERT INTO STARIJI_TRGOVAC(MB, IME_TRGOVCA)
SELECT MB, IME_PREZIME
FROM TRGOVAC
WHERE DATUM_RODJENJA < '1950-01-01';
```

STARIJI_TRGOVAC

MB	IME_PREZIME
I	Ivan Perić
II	Maja Markić

DELETE i UPDATE

- DELETE-služi brisanju n-torki
- **Primjer** (*brisanje svih poslovnih partnera iz Zagreba*)

DELETE FROM PARTNER

WHERE PARTNER_ADRESA="Zagreb";

- UPDATE-promjena podataka u n-torci
- **Primjer** (*povećat ćemo plaću za 20 svim trgovcima kojima je voditelj grupe pod šifrom 1*)

UPDATE TRGOVAC

SET PLACA=PLACA+20

WHERE MB_VG=1;

- **Primjer** (*promjenit ćemo ime partnera pod šifrom 80 u Granit i sjedište, tj. adresu u Vukovar*)

UPDATE PARTNER

SET PARTNER_NAZIV='Granit'

PARTNER_ADRESA='Vukovar'

WHERE PARTNER_SIFRA=80;

COMMIT i ROLLBACK

- Kada napravimo određenu transakciju (dio instrukcija koji čini nedjeljivu cjelinu) koristimo naredbu COMMIT WORK; kako bismo sve efekte koje smo postigli na bazi podataka sačuvali u sekundarnu memoriju
- Izvodimo ju nakon svake transakcije ili nakon nekoliko transakcija
- ROLLBACK naredba služi za poništavanje svih instrukcija od posljednje COMMIT naredbe ili sve od početka ukoliko nismo koristili COMMIT

CREATE VIEW i DROP VIEW

- Views (pogledi) su novo definirani prikazi postojećih podataka

Pomoću njih

- * Jednostavnije postavljamo upite
- * Skrivamo određene podatke u ispisu
- * Redefiniramo različite relacije
- Opći oblik

CREATE VIEW RELACIJA (ATRIBUT1, ATRIBUT2, ...)

AS SELECT ...

[WITH CHECK OPTIONS]

- **Primjer** (pogled pomoću kojeg možemo vidjeti sve trgovce koji su voditelji grupa)

❑ CREATE VIEW VODITELJ

AS SELECT * FROM TRGOVAC

WHERE MB=MB_VG;

- ☞ Ako upotrijebimo SELECT * FROM VODITELJ dobijemo sljedeći ispis

MB	IME_PREZIME	DATUM_RODJENJA	MB_VG	PLACA	DODATAK
1	Ivan Perić	1948-11-26	1	120	0
11	Maja Markić	1939-12-01	11	120	0

- **Primjer** (pogled koji prikazuje matični broj voditelja grupe, broj članova grupe, prosječnu plaću i dodatak te ukupni prihod svih članova grupe (ukupni prihod=placa+dodatak svih članova*3.5))

```
❑ CREATE VIEW GRUPA (MB_VG, BR_CLANOVA,  
PR_PL, PR_DOD, PRIHOD)  
AS SELECT MB_VG, COUNT(*), AVG(PLACA),  
AVG(DODATAK), SUM(PLACA+DODATAK)*3.5  
FROM TRGOVAC  
GROUP BY MB_VG;
```

- Ispisom iz tog pogled dobijemo sljedeće:
- SELECT * FROM GRUPA

MB	BR_CLANOVA	PR_PL	PR_DOD	PRIHOD
1	3	100	25	1312.5
11	2	115	5	840.0

- Isti pogled možemo koristiti kod postavljanja nekog novog upita kao npr.
- `SELECT *FROM GRUPA`
- `WHERE MB_VG=11;`
- Za ispis tada dobivamo

MB	BR_CLANOVA	PR_PL	PR_DOD	PRIHOD
11	2	115	5	840.0

• **Primjer** (*pogled koji prikazuje naziv i adresu poslovnog partnera te ime i prezime trgovca koji s njim radi*)

```

❑ CREATE VIEW PARTNER_TRGOVAC
      AS SELECT PARTNER_NAZIV,
                PARTNER_ADRESA, IME_PREZIME
      FROM PARTNER, TRGOVAC
      WHERE PARTNER.MB=TRGOVAC.MB

```

- zatražiti ispis svih poslovnih partnera s kojima radi određeni trgovac, npr. 'Josip Antić'
- `SELECT * FROM PARTNER_TRGOVAC`
- `WHERE IME_PREZIME=Josip Antić;`

PARTNER_NAZIV	PARTNER_ADRESA	IME_PREZIME
Horizont d.d.	Sesvete	Josip Antić
Kompakt d.o.o.	Osijek	Josip Antić

- Pomoću naredbe `DROP VIEW` brišemo pogled iz baze podataka, a pošto on ne sadrži vlastite podatke briše se samo njegova definicija iz baze podataka
- Npr. `DROP VIEW PARTNER_TRGOVAC;`

CREATE INDEX i DROP INDEX

- Pomoću CREATE INDEX stvaramo indeks pomoću kojega možemo ubrzati pristup n-torkama i tako ubrzati selekciju
- Pomoću njih možemo definirati i primarne ključeve
- **Primjer** (indeks za brzo pronalaženje po atributu PARTNER_NAZIV relacije PARTNER)
- ❑ CREATE INDEX PART_NAZ_INDEKS ON PARTNER
(PARTNER_NAZIV);
- **Primjer** (indeks pomoću koga definiramo primarni ključ)
- ❑ CREATE UNIQUE INDEKS MB_INDEX ON
TRGOVAC (MB)
- DROP INDEX-brisanje indeksa
- Npr. DROP INDEX PART_NAZ_INDEKS;

GRANT i REVOKE

- Naredbom GRANT možemo dozvoliti određena prava korištenja korisniku u relaciji
- Naredbom REVOKE ukidamo prava korištenja
- Primjeri:
 1. GRANT ALL PRIVILEGES ON PARTNER TO PUBLIC;
 2. GRANT INSERT, UPDATE ON TRGOVAC TO User6
WITH GRANT OPTION;
 3. REVOKE ALL ON PARTNER FROM PUBLIC;
 4. REVOKE UPDATE ON TRGOVAC FROM User5,
User6;

KRITIKE SQL-a

- Implementacije su nekonzistentne i, obično, nespojive između distributera.
- Sintaksa datuma i vremena ,niz ulančanih podataka, nule, u usporednim slučajevima osjetljivo variraju od proizvođača do proizvođača.
- Jeziku je lako napraviti Kartezijev produkt (spajanje svih mogućih kombinacija), što rezultira u izlazu netočne rezultate kada su WHERE naredbe pogresno napisane
- Moguce je da kriva konstrukcija WHERE za promjenu ili brisanje,utjece na vise redova u tablici nego sto je zeljeno.
- Gramatika SQL-a je mozda nepotrebno složena.

- C.J.Date, dugogodišnji suradnik E.J.Codda, poznati znanstvenik na području relacijskog modela i RDBMS sustava napisao je 2005.

SQL je (relativno) loš relacijski jezik, nastao stihijski

dobar relacijski jezik (i RDBMS općenito):

- ne bi smio podržavati NULL vrijednosti
- ne bi smio podržavati duple retke u tablici
- mora bolje podržati ažuriranje pogleda (view-ova)
- mora bolje podržati integritetna ograničenja
- mora omogućiti bolju optimizaciju
- mora biti (više) simetričan...

ZAKLJUČAK

- U masi jezika za rad sa relacijskom bazom podataka (SQL, QUEL, QBE....) SQL je trenutno najbolji, najrasprostranjeniji ,najčešće korišteni jezik.
- Hoće li ga se u budućnosti uspjeti u potpunosti usavršiti ili će se pojaviti neki bolji jezik možemo samo pretpostaviti!!!

Literatura

- Mladen Varga, Baze podataka – Konceptualno, logičko i fizičko modeliranje podataka, Zagreb, 2004.