

SQL (Structured Query Language)

SQL je 1986 godine postao američki nacionalni standard (ANSI X3:135 - 1986).

Počeo je da se razvija u IBM-u, ranih sedamdesetih godina, pod imenom SEQUEL.

Zbog toga što je postao standard, danas je SQL jezik koji se koristi u velikoj većini relacionih baza podataka. Originalno se koristi u relacionim sistemima IBM-a (SQL/DS i DB2) i ORACLE, a naknadno je dodat i mnogim drugim koji ga nisu originalno koristili.

SQL je homogeno relacioni jezik, što znači da može da se koristi i interaktivno, sa terminala i u okviru standardnih programskih jezika (COBOL, Basic, C++ i drugih), za komunikaciju sa bazom podataka.

Pored konstrukcija analognih relacionoj algebri ili relacionom računu SQL sadrži i konstrukcije za opisivanje baze podataka, operacije koje omogućuju vezivanje SQL-a sa nekim standardnim jezikom.

Kontrolne konstrukcije za upravljanje konkurentnim obradama, oporavak baze podataka, definisanje zaštite podataka i integriteta baze podataka.

U SQL terminologija je slijedeća:

Tabela – umjesto izraza relacija

Red tabele – umjesto n-torka relacije

Kolona tabele – umjesto atribut relacije

SQL jezik podržava tri osnovne funkcije u radu sa bazama podataka:

-definicija baze podataka: definisanje strukture, koje tabele postoje, koji atributi postoje u tabelama i kog su tipa, koja ograničenja postoje u tabelama.

-manipulacija bazom podataka: pored upita nad bazom podataka, kojima dobijamo željene informacije, neophodno je obezbjediti i ažuriranje baze podataka, odnosno unos, izmjenu i brisanje podataka.

- kontrola pristupa podacima: kontrola koji korisnici imaju pristup kojim podacima i šta mogu da rade sa tim podacima.

6.2 SQL za definiciju baze podataka

Definicija neke baze podataka podrazumjeva i mogućnost naknadne izmjene te definicije.

U standardnom SQL jeziku se to postiže sa svega dvije naredbe:

- **CREATE**: služi za kreiranje nekog objekta (tabele, pogleda ili indeksa) u bazi podataka
- **DROP**: služi za uklanjanje definicije nekog objekta iz baze podataka.

6.2.1 SQL tipovi podataka

Neophodno je da se definiše kog su tipa kolone u tabeli.

- **INTEGER** ili skraćeno INT: cijeli brojevi od 32 bita
- **SMALLINT**: 16 bitni cio broj
- **DECIMAL (m,n)** ili skraćeno DEC (m,n): decimalni broj sa ili bez predznaka i sa ukupno m cifara od čega su n decimalne.
- **CHARACTER (n)** ili skraćeno CHAR (n)
- **LOGICAL**
- **DATE**
- **TIME**

6.2.2 Naredbe kreiranja i uklanjanja tabele

Osnovni oblik iskaza za kreiranje bazne tabele je

```
CREATE TABLE  ImeTabele  
    ( ImeKolone TipKolone Ograničenje Kolone ..  
    {, ImeKolone TipKolone Ograničenje Kolone ... }  
[OgraničenjeTabele]
```

Ime tabele je identifikator (maksimalne dužine 18 znakova u DB2), čiji je prvi znak slovo a ostali znaci su slova, cifre ili podvlaka ('_').

Ovako definisano ime je *nekvalifikovano*, a može se kvalifikovati navođenjem lokacije (scheme) i vlasnika tabele.

Definicija kolone *def-kolone* ima oblik:

ime-kolone tip-podataka [NOT NULL]
[[WITH] DEFAULT [*vrednost*]]

Ograničenje kolone i njihova značenja su:

- NOT NULL

u koloni nije dozvoljena NULL vrijednost

- UNIQUE

u koloni nije dozvoljeno ponavljanje iste vrijednosti

- PRIMARY KEY

kolona je primarni ključ i u njoj nije dozvoljena NULL vrijednost niti ponavljanje vrijednosti

- CHECK (Predikat)

svaka vrijednost u koloni mora da zadovoljava uslov zadat logičkim izrazom koji smo označili sa Predikat i u kome se ne mogu navoditi druge kolone.

- DEFAULT = Konstanta

ako se prilikom unošenja jednog reda podataka u tabelu za kolonu ne zada vrijednost, preuzima se podrazumjevana vrijednost Konstanta.

Za statičku specifikaciju referencijalnog integriteta za jednu kolonu:

REFERENCES ImeTabele [(Ime Kolone)]

kolona je strani ključ u odnosu na tabelu ImeTabele (u kojoj se nalazi taj strani ključ)

i uvijek mora imati ili vrijednost primarnog ključa u toj tabeli ili NULL vrijednost (ako nema NOT NULL ograničenja)

Za cijelu tabelu se mogu zadati: ni jedno, jedno ili više ograničenja. To je neophodno u situacijama kada ograničenja važe za skup od dvije ili više kolona.

Osnovne klauzule OgraničenjeKolone za više tabela su

- UNIQUE (**ListKolona**)

u koloni nije dozvoljeno ponavljanje istih kombinacija vrijednosti kolona čija su imena navedena u **ListKolona**.

- PRIMARY KEY (**ListKolona**)

kolone navedene u (**ListKolona**) su primarni ključ i u njoj nije dozvoljena NULL vrijednost niti ponavljanje kombinacija njihovih vrijednosti.

Sledeća konstrukcija služi za definisanje da više kolona predstavlja strani ključ i za dinamičku specifikaciju referencijalnog integriteta:

-FOREIGN KEY (*ListaKolona* ,..)
REFERENCES *ImeTabele* [(*ListaKolona* ,
...)]

Navedene kolone su zajedno strani ključ u odnosu na tabelu *IMETABELE* i one smiju ili sve biti NULL vrijednosti ili u kombinaciji imati vrijednost primarnog ključa u toj tabeli.

U predhodnim definicijama sa ListaKolona se označava konstrukcija oblika

ImeKolone {,ImeKolone} ...

odnosno navođenje jedne kolone ili više kolona odvojenih zarezom.

Slijedeće dvije klauzule služe za dinamičku specifikaciju referencijalnog integriteta u slučaju brisanja reda u cijeloj tabeli **ImeTabele** ili izmjene vrijednosti PK u cijeloj tabeli:

- **DELETE OF *ImeTabele***
{RESTRICTED/CASCADE/NULLS}

specifikacija šta se dešava u slučaju pokušaja uklanjanja jednog reda u cijeloj tabeli.

- **UPDATE OF *ImeTabele***
{RESTRICTED/CASCADE/NULLS}

specifikacija šta se dešava u slučaju pokušaja promjene vrijednosti PK u jednom redu u cijeloj tabeli.

Sva ograničenja navedena u CREATE TABLE definiciji neke tabele aktivna su u svakom trenutku postojanja te tabele.

Primjer:

```
CREATE TABLE Oblast (SifO CHAR(2) PRIMARY KEY,  
                     Naziv CHAR(20) NOT NULL UNIQUE);
```

```
CREATE TABLE Naslov (SifN CHAR(4) PRIMARY KEY,  
                     Naziv CHAR(20) NOT NULL,  
                     SifO CHAR(2) NOT NULL  
                     REFERENCES Oblast  
                     UPDATE OF Oblast CASCADES,  
                     DELETE OF Oblast RESTRICTED );
```

```
CREATE TABLE Knjiga (SifK CHAR(3) PRIMARY KEY,  
                     SifN CHAR(4) NOT NULL REFERENCES Naslov,  
                     UPDATE OF Naslov CASCADES,  
                     DELETE OF Naslov RESTRICTED );
```

```
CREATE TABLE JeAutor (SifA CHAR(3) REFERENCES Autor,  
                      SifN CHAR(4) REFERENCES Naslov,  
                      Koji INT NOT NULL CHECK (Koji > 0)  
                      PRIMARY KEY (SifA, SifN),  
  
                      UPDATE OF Autor CASCADES,  
  
                      DELETE OF Autor RESTRICTED,  
  
                      UPDATE OF Naslov CASCADES,  
  
                      DELETE OF Naslov RESTRICTED );
```

Izbor dinamičkih specifikacija integriteta za slučaj ulanjanja predstavlja delikatnu odluku.

Ako uvijek nepromišljeno koristimo klauzulu **RESTRICTED**, namećemo vrlo krut režim rada sa bazom podataka koji može dovesti i do toga da ne možemo da uklonimo pogrešno unijete podatke iz tabele.

Naredba uklanjanja definicije tabele iz baze podataka je jednostavna. Njena sintaksa je:

DROP TABLE *ImeTabele*;

Tabela koja se uklanja mora biti prazna. U suprotnom, sistem za upravljanje bazom podataka neće izvršiti tu naredbu.

6.2.3 Indeksi

Indeks je pomoćna datoteka koja treba da ubrza pristup podacima u nekoj osnovnoj datoteci.

Indeks se može posmatrati kao uređeni skup pokazivača na vrste bazne tabele, fizički odvojen od podataka u tabeli. Svaki indeks baziran je na vrednostima podataka jedne ili više kolona tabele.

Indeks je veoma koristan mehanizam koji ubrzava pristup vrstama tabele.

Kao što je u pronalaženju nekog pojma u knjizi koristan indeks, koji eliminiše potrebu za prelistavanjem knjige stranu po stranu, tako je i u pretraživanju vrsta tabele po nekom uslovu mnogo efikasnije koristiti indeks po koloni koja učestvuje u uslovu, nego pretraživati tabelu sekvencijalno, vrstu po vrstu.

Nad indeksima se mogu izvršavati samo SQL iskazi za definisanje, tj. iskazi za kreiranje i uklanjanje indeksa.

Iskaz kreiranja indeksa je oblika

```
CREATE [UNIQUE] INDEX ime-indeksa  
ON Imetabele  
( ime-kolone [redosled] {, ime-kolone  
[redosled]})  
[ drugi-parametri]
```

Opcija **UNIQUE** precizira da dve razne vrste u baznoj tabeli nad kojom se kreira indeks ne mogu uzeti istovremeno istu vrednost na indeksnoj koloni ili njihovoj kombinaciji.

Redosled uz svaku kolonu – komponentu indeksa je ili ASC (rastući) ili DESC (opadajući); ASC se podrazumeva ako se redosled ne navede.

Iskaz za uklanjanje indeksa ima oblik

DROP INDEX *ime-indeksa*

Izvršavanjem ovog iskaza opis indeksa se uklanja iz kataloga.

Primjer

CREATE INDEX JeAutor1 ON JeAutor (SIFA)

CREATE INDEX JeAutor2 ON JeAutor (SIFN)

Jeautor1		Jeautor		Jeautor2		
<u>SIFA</u>		<u>SIFA</u>	<u>SIFN</u>	KOJI		<u>SIFN</u>
AP0		AP0	RBPO	1		PJC0
AP1		JN0	RBPO	2		PJCO
DM0		DM0	RK00	1		PPOO
DM0		ZP0	PPOO	1		PPOO
IT0		DM0	PPOO	2		PPOO
JN0		AP1	PJC0	1		RBPO
ZP0		IT0	PPOO	3		RBPO
ZP0		ZP0	PJCO	2		RK00

3.2.4 Pogledi

Pogledi su virtuelne tabele koje nisu predstavljene fizičkim podacima, već je njihova definicija (u terminima drugih tabela) zapamćena u katalogu.

Iskaz za kreiranje pogleda u SQL-u je oblika:

```
CREATE VIEW ime-pogleda [( kolonaf,  
kolonag)] AS puni upitni blok  
[WITH CHECK OPTION]
```

CHECK opcija se odnosi na primenu ograničenja navedenih u logičkom izrazu WHERE linije upitnog bloka, pri unošenju i izmeni podataka u pogledu.

Na primer,

```
CREATE VIEW JUGIZD AS SELECT I SIF,  
NAZIV FROM I WHERE DRZAVA='Jugoslavija'
```

Kada se izvrši ovaj CREATE iskaz, podupit SELECT-FROM-WHERE se ne izvršava, već se samo definicija upita sačuva u katalogu.

Pogled JUGIZD je “prozor”, i to dinamički, na tabelu I. Izmene nad pogledom JUGIZD automatski se izvršavaju nad tabelom I, a izmene nad tabelom I automatski se odražavaju (“vide”) na pogled JUGIZD (ako zadovoljavaju uslove koji definišu taj pogled).

Korisnik može ali i ne mora biti svjestan da pogled nije bazna tabela, jer se nad pogledom mogu postavljati upiti (gotovo svi) kao i nad baznom tabelom.

Iskaz za uklanjanje pogleda ima oblik

DROP VIEW *ime-pogleda*

Na primer

DROP VIEW JUGIZD

Uklanjanje pogleda nema nikakvog efekta na osnovne tabele iz upita.

Definiciju pogleda nije moguće menjati.

