



Računarstvo i informatika

Katedra za računarstvo

Elektronski fakultet u Nišu

Baze podataka (Računske vežbe) SQL DDL naredbe

Letnji semestar 2017/2018



Sadržaj

- SQL – Osnovni koncepti
- Pregled Oracle SQL naredbi
- CREATE TABLE naredba
- DROP TABLE naredba
- ALTER TABLE naredba



SQL – Osnovni koncepti

- **SQL** (Structured Query Language) je standardni upitni jezik za relacione baze podataka.
- Prvu verziju SQL-a razvio je IBM 1974. godine.
- Originalni naziv **SEQUEL**.
- SQL je razvijen pod uticajem **Codd**-ovog rada (relacioni model) a tvorci su: **Donald D. Chamberlin** i **Raymond F. Boyce**
- Svi relacioni DBMS-ovi podržavaju SQL: ORACLE, IBM DB2, MS SQL Server, MySQL, PostgreSQL, Sybase, Access, ...



SQL – Osnovni koncepti

- **Važeći standardi:**

- **SQL-86** - 1986. godine od strane Američkog nacionalnog instituta za standarde (ANSI) i 1987. godine Internacionalne organizacije za standardizaciju (ISO)
 - **SQL-89** - 1989. godine
 - **SQL-92** – 1992. godine
 - **SQL:1999** – 1999. godine (dodata podrška za trigere, neskalarne tipove i objekte).
 - **SQL:2003**- 2003. godine (dodata podrška za XML, sekvence i kolone čije se vrednosti generišu automatski).
- Gotovo svi komercijalni DBMS-ovi podržavaju SQL ali većina njih ima sopstvenu verziju SQL-a (PL/SQL, SQL PL, T-SQL, ...).
 - U cilju poštovanja ANSI i ISO standarda svaki proizvođač DBMS-a koji podržava SQL mora da implementira određeni skup ključnih reči (SELECT, INSERT, UPDATE, DELETE, WHERE, ...) koje se ponašaju na gotovo identičan način u svim proizvodima.
 - Ovaj skup ključnih reči je u većini slučajeva sasvim dovoljna za sve standardne operacije sa podacima u relacionoj bazi.



SQL – Osnovni koncepti

- **Jednostavnost** - Relacije se kreiraju jednom naredbom i odmah su dostupne za korišćenje.
- **Uniformnost** - Svi podaci i rezultati operacija se prikazuju u vidu tabele.
- **Interaktivno i klasično programiranje** - SQL ili ugrađeni SQL u C, C++, Java, PL/I, Pascal, FORTRAN, COBOL, ...
- **Neproceduralnost (deklarativni jezik)** - Specificira **ŠTA**, a ne i **KAKO** nešto treba uraditi.
- Podskupovi SQL naredbi:
 - Data Definition Language (DDL)
 - View Definition Language (VDL)
 - Data Manipulation Language (DML)
 - Data Control Language (DCL)



Pregled Oracle SQL naredbi

- DDL naredbe
 - **CREATE TABLE** – kreiranje tabela
 - **ALTER TABLE** – izmena tabela
 - **DROP TABLE** – brisanje tabela



Pregled Oracle SQL naredbi

- DML naredbe
 - **SELECT** – naredba za kreiranje upita
 - Aritmetičke funkcije
 - Funkcije za rad sa stringovima
 - Spoj
 - Operacije nad tabelama sa strukturom stabla
 - **INSERT INTO** - dodavanje novih podataka
 - **DELETE FROM** – brisanje podataka iz tabele
 - **UPDATE** – ažuriranje postojećih podataka
 - **TRUNCATE** – brisanje podataka iz tabele



Pregled Oracle SQL naredbi

- VDL naredbe
 - **CREATE VIEW** – naredba za kreiranje pogleda
 - **DROP VIEW** – naredba za brisanje pogleda



Pregled Oracle SQL naredbi

- Naredbe za rad sa indeksima
 - CREATE INDEX
 - DROP INDEX
- Naredbe za rad sa transakcijama
 - START TRANSACTION (BEGIN WORK)
 - COMMIT
 - ROLLBACK
- DCL naredbe
 - GRANT
 - REVOKE
 - AUDIT



DDL naredbe

- **CREATE TABLE** – naredba za kreiranje tabele.
- **DROP TABLE** – naredba za brisanje tabele.
- **ALTER TABLE** – naredba za modifikacijo obstoječe tabele.



CREATE TABLE

- Osnovni oblik CREATE TABLE naredbe za ORACLE DBMS:

CREATE TABLE <ime_tabele>

(

<lista_deklaracija_kolona>

[,<lista_deklaracija_ograničenja_tabele>]

);

- Za deklaraciju kolona se koristi sledeći format:
<ime_kolone> <tip_podatka>[(<dužina>)] [<ograničenje_kolone>],



CREATE TABLE

- Deklaracija kolone sadrži ime kolone, tip podatka kolone, opciono dužinu (ako je neophodna, u zavisnosti od tipa podatka) i opciono, ograničenja za kolonu. Deklaracije kolona se razdvojaju zapetama.
- Ograničenja za kolonu su opciona.
- Nakon deklaracije svih kolona navode se ograničenja koja važe za celu tabelu.
- Navedena sintaksa naredbe CREATE TABLE nije kompletna. SQL standard obezbeđuju korišćenje velikog broja dodatnih klauzula koje korisnicima omogućavaju preciznu kontrolu procesa kreiranja tabele. Osim toga proizvođači RDBMS-ova proširuju CREATE TABLE naredbu klauzulama koje su posledica specifičnih osobina njihovih proizvoda.



CREATE TABLE

Imena tabela, pogleda, atributa i drugih objekata baze podataka moraju da poštuju sledeća pravila:

1. Maksimalna dužina imena je 30 znakova.
2. Ime ne sme da sadrži znak pitanja (?)
3. Nema razlike između malih i velikih slova
4. Prvi znak mora biti slovo
5. Dozvoljeni znaci su A-Z, 0-9, _, \$ i #.
6. Kao imena se ne smeju koristiti rezervisane reči.
7. Imena se ne smeju ponavljati.



CREATE TABLE

- Oracle podržava veliki broj ugrađenih tipova podataka:
 - Znakovni tipovi podataka
 - CHAR
 - VARCHAR, VARCHAR2
 - NCHAR, NVARCHAR
 - Numerički tipovi podataka
 - NUMBER,
 - INTEGER, SMALL INT, INT
 - FLOAT, REAL, DOUBLE PRECISION
 - Tipovi podataka za predstavljanje datuma
 - DATE
 - TIMESTAMP
 - Tipovi podataka za predstavljanje velikih objekata (LOB)
 - BLOB
 - CLOB, NCLOB
 - BFILE



CREATE TABLE

- **CHAR**

- Znakovni podaci fiksne dužine
- Obavezna se mora specificirati dužina kolone (u bajtovima ili karakterima) od 1 do 2000 bajtova. Podrazumevana dužina je 1 bajt. Dužina se podrazumevano zadaje kao broj bajtova.
- Ukoliko je podatak u koloni kraći od specificirane dužine on se dopunjuje blanko znacima sa desne strane.
- Ukoliko je podataka veći od specificirane dužine Oracle DBMS vraća grešku.
- Prilikom poređenja CHAR vrednosti uzimaju se u obzir i dodati blanko znaci.



CREATE TABLE

- **VARCHAR i VARCHAR2**

- Znakovni podaci promenljive dužine
- Obavezna se mora specificirati maksimalna dužina kolone (u bajtovima ili karakterima) od 1 do 4000 bajtova. Podrazumevana dužina je 1 bajt. Maksimalna dužina se podrazumevano zadaje kao broj bajtova.
- Ukoliko je podataka veći od specificirane maksimalne dužine Oracle DBMS vraća grešku.
- U koloni se pamti podatak bez dopunjavanja blanko znacima.
- VARCHAR i VARCHAR2 su sinonimi (preporučuje se korišćenje VARCHAR2 tipa da bi se izbegli problemi sa budućom kompatibilnošću)



CREATE TABLE

- **NCHAR i NVARCHAR**

- UNICODE znakovni podaci (UTF8 ili ALI6UTFI6 kodni raspored)
- NCHAR pamti UNICODE znakovne podatke fiksne dužine
- NVARCHAR pamti UNICODE znakovne podatek promenljive dužine
- Dužina se uvek specificira kao broj karaktera.
- Maksimalna dužina za NCHAR kolone je 2000 karaktera a za NVARCHAR kolone je 4000 karaktera.
- NCHAR i NVARCHAR omogućava memorisanje UNICODE karaktera bez obzira na kodni raspored Oracle baze podataka.



CREATE TABLE

- **NUMBER**

- Za predstavljanje brojeva i u fiksnom i u ppokretno zarezu.
- Mogu se predstaviti sledeće vrednosti:
 - Pozitivni brojevi 1×10^{-130} do $9.99...9 \times 10^{125}$ sa 38 značajnih cifara
 - Negativni brojevi -1×10^{-130} do $-9.99...9 \times 10^{125}$ sa 38 značajnih cifara
 - Nula
 - Pozitivna i negativna beskonačna vrednost
- column_name NUMBER
- column_name NUMBER(p, s) – specificira se precision (ukupna broj cifara) i scale (broj cifara desno od decimalne tačke)
- column_name NUMBER(*, s) – precision je 38 a scale se održava
- Negativna vrednost za scale nalaže Oracle-u da izvrši zaokruživanje numeričke vrednosti na specificirani broj pozicija levo od decimalne tačke.



CREATE TABLE

- **INTEGER, INT, SMALL INT**
 - Za celobrojne podatke
- **FLOAT, REAL, BINARY_FLOAT, BINARY_REAL**
 - Za razlomljene podatke sa pokretnim zarezom
- **LOB tipovi podataka**
 - Za pamćenje velikih blokova nestruktuiranih podataka
 - BLOB se koristi za pamćenje binarnih podataka
 - CLOB i NCLOB se koriste za pamćenje znakovnih podataka
 - BFILE omogućava pamćenje podataka van baze podataka u file sistemu
 - Maksimalna veličina je 128TB



CREATE TABLE

- **DATE, TIMESTAMP**

- Za pamćenje informacija o datumu i vremenu
- Pamti se informacija o godini, mesecu, danu, satu, minuti i sekundama.
- Standardni format za datume je DD-MMM-YYYY (Primer: '13-NOV-92')
- Standardni format za pamćenje vremena je HH:MI:SS. (Primer: '13-NOV-92 11:12:32')
- Ukoliko se navede samo datum podrazumevani podatak za vreme je 00:00:00 odnosno ponoć.
- Ukoliko se navede samo vreme podrazumevani datum je prvi dan tekućeg meseca.
- TIMESTAMP može da uključuje i delove sekundi i informaciju o vremenskoj zoni.



CREATE TABLE

```
CREATE TABLE RADNIK  
(  
    LIME VARCHAR(15),  
    SSLOVO CHAR(1),  
    PREZIME VARCHAR(15),  
    MATBR NUMBER(10),  
    DATRODJ DATE,  
    POL CHAR(1),  
    PLATA NUMBER,  
    ADRESA VARCHAR(30),  
    MATBROJS NUMBER(10),  
    BRSEK NUMBER(2)  
);
```

```
CREATE TABLE SEKTOR  
(  
    NAZIV VARCHAR(25),  
    SBROJ NUMBER(2),  
    MATBRR NUMBER(10),  
    DATPOST DATE  
);
```



CREATE TABLE

- Ograničenja kolone:
 - **NULL** ili **NOT NULL** –definiše da kolona može ili ne može imati NULL vrednosti.
 - **UNIQUE** – definiše da kolona ima jedinstvene vrednosti (kandidati za ključeve).
 - **PRIMARY KEY** – definiše da kolona predstavlja primarni ključ tabele (može da se primeni na samo jednu kolonu u tabeli).
 - **CHECK** –definiše ograničenje za proveru vrednosti kolone (koristi se kod upisa ili ažuriranja vrednosti).
 - **DEFAULT** – definiše podrazumevanu vrednost za kolonu (kolona uzima ovu vrednost, ako vrednost kolone nije navedena).
 - **REFERENCES** –definiše da kolona predstavlja spoljni ključ tabele.



CREATE TABLE

- Za vrednost kolone se mogu specificirati ograničenja NULL ili NOT NULL čime se dozvoljava ili zabranjuje NULL vrednost kolone.
- Strože ograničenje je UNIQUE, koje ne dozvoljava ponavljanje vrednosti u koloni.
- Za kolonu podrazumevano ograničenje je NULL vrednost. To znači da navodimo samo ograničenje NOT NULL, ako je definisano za konkretnu kolonu.
- Kod navođenja PRIMARY KEY podrazumevaju se UNIQUE i NOT NULL ograničenje za tu kolonu, tako da se ne moraju posebno navoditi.



CREATE TABLE

- Pri kreiranju tabele u kojoj je atribut A primarni ključ, a atribut B spoljni ključ koristi se sledeći oblik naredbe:

```
CREATE TABLE <imeTabele>
```

```
(
```

```
  A <tip_podatka> PRIMARY KEY,
```

```
  B <tip_podatka> REFERENCES <ime_ref_tabele>(<ime_ref_kolone>),
```

```
  ostali atributi
```

```
);
```

- Kod deklaracije spoljnog ključa tabele iza ključne reči **REFERENCES** navodi se ime referencirane tabele i opciono, u maloj zagradi, ime referencirane kolone u toj tabeli. Oracle SQL očekuje da je referencirana kolona primarni ključ (kolona deklarirana sa **PRIMARY KEY**) u referenciranoj tabeli.
- Atribut iza koga stoji klauzula **REFERENCES** definisan je kao spoljni ključ u odnosu na primarni ključ tabele čije je ime navedeno iza klauzule **REFERENCES**.



CREATE TABLE

CREATE TABLE RADNIK

(

LIME VARCHAR(15) **NOT NULL**,

SSLOVO CHAR(1),

PREZIME VARCHAR(15) **NOT NULL**,

MATBR NUMBER(10) **PRIMARY KEY**,

DATRODJ DATE,

POL CHAR(1) **CHECK** (POL IN ('M', 'Ž')),

PLATA NUMBER **CHECK** (PLATA > 0),

ADRESA VARCHAR(30) **NOT NULL**,

MATBROJS NUMBER(10) **REFERENCES** RADNIK(MATBR),

BRSEK NUMBER(2) **REFERENCES** SEKTOR(SBROJ)

);



CREATE TABLE

- Za definisanje ograničenja koja važe za tabelu u celini možete koristiti:
 - **PRIMARY KEY** – definiše koja kolona ili koje kolone čine primarni ključ tabele.
 - **UNIQUE** – definiše koja kolona ili koje kolone imaju jedinstvene vrednosti (kandidati za ključeve).
 - **FOREIGN KEY** – definiše koja kolona ili koje kolone čine spoljni ključ tabele.
 - **CHECK** – definiše ograničenja vrednosti kolone ili kolona koje DBMS proverava kod upisa ili ažuriranja vrednosti te ili tih kolona.



CREATE TABLE

- Za kreiranje tabele u kojoj je skup atributa (a,b,c) primarni ključ možete koristiti sledeći oblik CREATE TABLE naredbe.

```
CREATE TABLE < ime_tabele >
```

```
(
```

```
    <atributi i njihovi tipovi podataka>,
```

```
    PRIMARY KEY (a,b,c)
```

```
);
```

- Attribute a, b i c treba deklarirati u sekciji < atributi i njihovi tipovi podataka >.



CREATE TABLE

```
CREATE TABLE RADI_NA
```

```
(
```

```
  MBR INTEGER PRIMARY KEY,
```

```
  BRPR INTEGER PRIMARY KEY,
```

```
  SATI NUMBER(2,0)
```

```
);
```



- Ovo nije validan pristup. Ograničenje kolone se može iskoristiti za kreiranje primarnog ključa za slučaj da primarni ključ čini samo jedna kolona tabele.
- Za slučaj kompozitnog primarnog ključa mora se iskoristiti ograničenje tabele.



CREATE TABLE

```
CREATE TABLE RADI_NA  
(  
    MBR INTEGER NOT NULL,  
    BRPR INTEGER NOT NULL,  
    SATI NUMBER(2,0),  
    CONSTRAINT RADI_NA_PK PRIMARY KEY (MBR, BRPR)  
);
```

```
CREATE TABLE RADI_NA  
(  
    MBR INTEGER NOT NULL,  
    BRPR INTEGER NOT NULL,  
    SATI NUMBER(2,0),  
    PRIMARY KEY (MBR, BRPR)  
);
```




CREATE TABLE

- U deklaraciji kolone ili tabele, nakon klauzule **REFERENCES**, mogu se navesti klauzule **ON DELETE** ili **ON UPDATE** koje specificiraju aktivnosti u slučaju narušavanja integriteta.
 - **ON DELETE** omogućava specifikaciju aktivnosti nad torkama relacije, odnosno vrstama tabele u kojoj je navedeno ovo ograničenje, u slučaju brisanja torki u tabeli na koju se referencirate u **REFERENCES** klauzuli.
 - **ON UPDATE** omogućava specifikaciju aktivnosti nad torkama u tabeli gde je **REFERENCES** ograničenje specificirano u slučaju ažuriranja (promene vrednosti) podataka u referenciranoj tabeli . Oracle ne podržava klauzulu **ON UPDATE**.
- U oba slučaja, iza ovih klauzula se navodi jedna od klauzula koja definiše aktivnost koja će se izvršiti nad torkama u slučaju brisanja ili ažuriranja vrednosti torke u tabeli u kojoj je ograničenje navedeno:
 - **NO ACTION** – nema akcije (ovo je predefinisana vrednost).
 - **CASCADE** – kaskadno izvršenje aktivnosti brisanja (brišu se torke u tabeli koja sadrži ograničenje) kod **ON DELETE**.
 - **SET NULL** – vrednosti atributa stranog ključa se postavljaju na NULL.



CREATE TABLE

```
CREATE TABLE RADI_NA  
(  
    MBR INTEGER NOT NULL,  
    BRPR INTEGER NOT NULL,  
    SATI NUMBER(2,0),  
    CONSTRAINT RADI_NA_PK PRIMARY KEY (MBR, BRPR),  
    CONSTRAINT RADNIK_FK FOREIGN KEY (MBR)  
        REFERENCES RADNIK(MATBR),  
    CONSTRAINT PROJEKAT_FK FOREIGN KEY (BRPR)  
        REFERENCES PROJEKAT(BROJ)  
        ON DELETE CASCADE  
);
```



ALTER TABLE

- Naredba ALTER TABLE menja definiciju tabele.

ALTER TABLE <ime tabele>

ADD <definicija atributa> |

ADD <definicija ograničenja> |

DROP <ime atributa> |

DROP CONSTRAINT <ime ograničenja> |

MODIFY < definicija atributa > |

DROP PRIMARY KEY

DROP UNIQUE <ime atributa>



ALTER TABLE

Tabela za testiranje:

```
CREATE TABLE PROBA
```

```
(
```

```
  X INT,
```

```
  Y INT,
```

```
  CONSTRAINT XYcheck check(X>y)
```

```
);
```

Dodavanje novih kolona:

```
ALTER TABLE PROBA
```

```
  ADD (Z FLOAT, W CHAR(1));
```



ALTER TABLE

Definisanje primarnog ključa:

ALTER TABLE PROBA

ADD PRIMARY KEY(X);

Dodavanje ograničenja:

ALTER TABLE PROBA

ADD CONSTRAINT YZunique UNIQUE(Y, Z);

Izmena ograničenja:

ALTER TABLE PROBA

MODIFY (W VARCHAR(2) DEFAULT 'AM' CONSTRAINT
Wnotnull NOT NULL);



ALTER TABLE

Dodavanje novog ograničenja:

```
ALTER TABLE PROBA
```

```
ADD CHECK (W IN ('AM', 'PM'));
```

Brisanje ograničenja:

```
ALTER TABLE PROBA
```

```
DROP CONSTRAINT YZunique;
```

Brisanje primarnog ključa:

```
ALTER TABLE PROBA
```

```
DROP PRIMARY KEY;
```

Brisanje atributa:

```
ALTER TABLE PROBA
```

```
DROP COLUMN Z;
```



DROP TABLE

- Naredba za brisanje tabele:

**DROP TABLE <ime_tabele> [CASCADE CONSTRAINTS]
[PURGE]**

- **CASCADE CONSTRAINTS** – brisanjem tabele automatski briše i sva ograničenja koja referenciraju tabelu koja se briše.

Primer:

DROP TABLE RADNIK;

DROP TABLE PROJEKAT CASCADE CONSTRAINTS;