

Baze podataka

Vježbe 1

21. februar 2023.

Kreiranje tabela, unošenje podataka u tabele i prikaz tabela; Tipovi podataka, Ograničenja vrijednosti, not null i unique; Brisanje tabela. Oracle baza podataka.

1. Napraviti tabelu STUDENT koja treba da sadrži sljedeće kolone: IND (broj indeksa, string maksimalne dužine 6 karaktera), IME (string, maksimalne dužine 50 karaktera), SS (šifra smjera, numeričko polje, cio broj, sastavljeno od maksimalno 5 cifara), GS (godina studija, numeričko polje, sastavljeno od maksimalno 1 cifre). Unijeti podatke prikazane u Tabeli 1 u kreiranu tabelu.

Tabela 1: Tabela STUDENT

IND	IME	SS	GS
25/09	Marko Marković	12001	4
13/07	Petar Petrović	12003	4

2. Napraviti tabelu SMJEROVI koja treba da sadrži sljedeće kolone: SS (šifra smjera, numeričko polje od maksimalno 5 cifara), NAZIV (string maksimalne dužine 50 karaktera), SO (šifra odsjeka, numeričko polje od maksimalno 2 cifre). Unijeti podatke prikazane u Tabeli 2 u kreiranu tabelu.

Tabela 2: Tabela SMJEROVI

SS	NAZIV	SO
12001	Elektronika	12
12002	Telekomunikacije	12
12003	Računari	12
12004	Mikrotalasna tehnika	12
13001	Elektroenergetski sistemi	13
13002	Automatika	13

3. Napraviti tabelu ODSJECI, koja treba da sadrži sljedeće kolone: SO (šifra odsjeka, numeričko polje sastavljeno od maksimalno 2 cifre) i NAZIV (string sastavljen od maksimalno 50 karaktera). Unijeti podatke prikazane u Tabeli 3 u kreiranu tabelu.
4. Napraviti tabelu PREDMET koja treba da sadrži spisak predmeta koji se proučavaju na različitim odsjecima, smjerovima, i godinama studija na Elektrotehničkom fakultetu. Tabela treba da sadrži sljedeća polja: SO, SS, GS i NAZIV, kolone definisane na isti način kao u prethodnim zadacima. Podatke iz ilustracione Tabele 5 potrebno je unijeti u kreiranu tabelu.
5. Obratite pažnju na strukturu kreiranih tabela. Razmislite i diskutujte o načinu povezivanja podataka iz tabela. Šta ćete uraditi da biste za zadanog studenta pročitali na kojem je smjeru?

Tabela 3: Tabela ODSJECI

SO	NAZIV
12	Elektronika, telekomunikacije, računari
13	Energetika i automatika
14	Studije primijenjenog računarstva

Tabela 4: Tabela PREDMET

SO	SS	GS	NAZIV
12	12003	4	Baze podataka
13		1	Osnovi računarstva II
14		3	Web design

6. Kreirati tabelu Srednje škole (case sensitive naziv tabele, i kolona) koja sadrži kolone:

- Identifikator škole– numeričko polje, cio broj u opsegu od 0 do 55
- Tip škole – string, sa najviše 50 karaktera
- Naziv škole– string, sa najviše 50 karaktera, neka se naziv unosi pod apostrofima
- Koeficijent – decimalan broj, od 2 decimale, broj mora biti veći od 5 a manji od 99

Unijeti dvije vrste u kreiranu tabelu. Pokušati unos podataka koji ne zadovoljavaju postavljena ograničenja polja. Naziv škole treba da bude unešen sa apostrofima. Tabela 5 ilustruje traženu tabelu.

Tabela 5: Tabela **Srednje škole**

Identifikator škole	Tip škole	Naziv škole	Koeficijent
13	Gimnazija	'Petar I Petrović Njegoš'	7.22
17	Elektroehnička	'Vaso Aligrudić'	8.19

7. Napraviti tabelu PITAGORINA_TEOREMA koja ima tri kolone sa podacima tipa number: A , B i C . Obezbijediti da prilikom unosa podataka u tabelu uvijek bude zadovoljeno da je $A^2 + B^2 = C^2$. Prilikom unosa podataka, nijedno polje ne smije biti prazno.

Predlog rješenja

1. Kreiranje tabele:

```
CREATE TABLE student (  
    ind VARCHAR2(6),  
    ime VARCHAR2(50),  
    ss NUMBER(5),  
    gs NUMBER(1)  
)
```

Prikaz sadržaja tabele:

```
select * from student
```

SQL nije case sensitive, pa može i:

```
SELECT * FROM Student
```

Unesimo podatke i prikažimo rezultat:

```
insert into student values ('25/09', 'Marko Marković', 12001, 4)  
insert into student (ind, ss, gs, ime)  
    values ('13/10', 13003, 4, 'Petar Petrović')  
insert into student(ind,ime) values ('05/08', 'Nepotpun red')  
insert into student values ('06/08', 'Nepotpun red 2', null, null)  
SELECT * FROM student
```

Uočite da kod posljednja dva unosa imamo prazna polja. Prikazali smo dva načina unosa u tom slučaju.

2. Kreirajmo tabelu:

```
CREATE TABLE smjerovi (  
    ss NUMBER(5),  
    naziv VARCHAR2(50),  
    so NUMBER(2)  
)
```

Unesimo podatke:

```
INSERT INTO smjerovi VALUES (12002, 'Računari', 12)  
INSERT INTO smjerovi VALUES (12002, 'Telekomunikacije', 12)  
INSERT INTO smjerovi values (12001, 'Elektronika', 12)  
INSERT INTO smjerovi VALUES (12004, 'Mikrotalasna tehnika', 12)  
INSERT INTO smjerovi VALUES (13001, 'Elektroenergetski sistemi', 13)  
INSERT INTO smjerovi VALUES (13002, 'Automatika', 13)
```

3. Kreiranje tabele:

```
CREATE TABLE odsjeci (  
    so NUMBER(2),  
    naziv VARCHAR2(50)  
)
```

Unos podataka:

```
INSERT INTO odsjeci VALUES (12, 'ETR')  
INSERT INTO odsjeci VALUES (13, 'EA')  
INSERT INTO odsjeci VALUES (14, 'SPR')
```

4. Kreiranje tabele:

```
CREATE TABLE predmet (
  so NUMBER(2),
  ss NUMBER(5),
  gs NUMBER(1),
  naziv VARCHAR2(50)
)
```

Unos podataka u tabelu:

```
INSERT INTO predmet (so, ss, gs, naziv) VALUES (12, 12003, 4, 'Baze podataka')
INSERT INTO predmet (so, gs, naziv) VALUES (13, 1, 'Osnovi računarstva II')
INSERT INTO predmet VALUES (14, NULL, 3, 'Web design')
```

5. Možemo uočiti da imamo 4 međusobno povezane tablele. Važno je napomenuti da struktura ovih tabela nije pažljivo modelovana – naime, tabela **PREDMETI** može praviti probleme prilikom definisanja upita kojima se povezuje sa drugim tabelama (npr. sa tabelom **STUDENT**).

Međutim, možemo diskutovati na koji način se kombinuju podaci iz različitih tabela. Ako, na primjer, želimo da se za datog studenta prikaže odgovarajući smjer, pogledaćemo i uporediti odgovarajuće vrijednosti u tabelama **STUDENT** i **SMJEROVI**. Na primjer:

- Student **Marko Marković** u tabeli **STUDENT** ima **SS** jednak **12001**. Iz tabele **STUDENT** ćemo pročitati koji smjer ima **SS = 12001**. Zaključujemo da je to smjer **Elektronika**.
- Student **Petar Petrović** u tabeli **STUDENT** ima **SS** jednak **12003**. Iz tabele **STUDENT** ćemo pročitati koji smjer ima **SS = 12003**. Zaključujemo da je to smjer **Računari**.

Da bi ovakvo povezivanje vrijednosti bilo konzistentno, neophodno je uvesti **ograničenja kolona SMJEROVI u obje tablele**. Naime, u tabeli **SMJEROVI**, kolona **SS** treba da ima ograničenje **PRIMARY KEY** (kojim se obezbjeđuje da je svaki unos jedinstven i da nema vrijednost **null**). Kolona **SS** u tabeli **STUDENT** treba da ima ograničenje **FOREIGN KEY**. To znači da vrijednosti u koloni **SS** tabele **STUDENT** moraju da budu iz skupa vrijednosti kolone **SS** tabele **SMJEROVI**.

6. Kreirajmo ovu case sensitive tabelu:

```
CREATE TABLE "Srednje škole" (
  "Identifikator škole" NUMBER(2)
  CHECK("Identifikator škole" >= 0 AND "Identifikator škole" <= 55),
  /* "Identifikator škole" NUMBER(2)
  CHECK("Identifikator škole" BETWEEN 0 AND 55), */
  "Tip škole" VARCHAR2(50),
  "Naziv škole" VARCHAR2(50),
  "Koeficijent" NUMBER(4,2) CHECK("Koeficijent" > 5 AND "Koeficijent" < 99)
  /* "Koeficijent" NUMBER(4,2) CHECK("Koeficijent" BETWEEN 5.01 AND 98.99 ) */
)
```

Probajmo prvo zabranjeni unos:

```
INSERT INTO "Srednje škole" VALUES
(13, 'Gimnazija', '''Petar I Petrović Njegoš''', 1.22)
```

Očekivano, javlja se greška. Sada jedan regularan unos:

```
INSERT INTO "Srednje škole" VALUES
(13, 'Gimnazija', '''Petar I Petrović Njegoš''', 7.22)
```

Prikažimo sadržaj tablele:

```
SELECT * FROM "Srednje škole"
```

7. Kreiranje tabele:

```
CREATE TABLE pitagorina_teorema (  
  a NUMBER NOT NULL,  
  b NUMBER NOT NULL,  
  c NUMBER NOT NULL,  
  CHECK (POWER(a,2) + POWER(b,2) = POWER(c,2))  
)
```

Primjer nedozvoljenog unosa:

```
INSERT INTO pitagorina_teorema VALUES (3, 3, 3)
```

Primjer dozvoljenog unosa:

```
INSERT INTO pitagorina_teorema VALUES (3, 4, 5)
```