



Zadatci za vježbu

#7

Previous step

Submit answer

Next step

Layout ▾

Tutorial content ▾

1/3

: Definiranje integritetskih ograničenja pomoću naredbe CREATE TABLE

Integritetska ograničenja mogu se definirati prilikom stvaranja relacije naredbom CREATE TABLE ili, za već postojeće relacije, naredbom ALTER TABLE.

Naredba ALTER TABLE koristi se za izmjenu sheme, integritetskih ograničenja i fizičkih svojstava postojeće relacije. Dio naredbe ALTER TABLE koji se odnosi na definiranje ograničenja nad postojećom relacijom možete naći u službenom podsjetniku.

Napomena vezana uz testiranje SQL naredbi iz rješenja.

SQL naredbe iz rješenja primjera u ovom i sljedećim koracima tutoriala možete izvoditi na dva načina:

1. kroz Edgara (npr. u Playgroundu)
2. u bazi podataka koju ste stvorili na vlastitoj instanci PostgreSQL sustava pomoću pgAdmina.

Preporučujemo drugi način jer ćete u tom slučaju moći obavljati naredbu po naredbu i lakše ćete razumjeti na koji način se pomoću SQL naredbi definira shema relacije i osigurava integritet podataka. Također, u rješenjima primjera se podrazumijevaju da na taj način obavljate testiranje.

Ako ćete naredbe obavljati u okruženju Edgara vodite računa o tome da su efekti naredbi koje obavite prilikom jednog pokretanja (**Run**) prisutni samo za vrijeme njihovog izvršavanja. Nakon toga se baza podataka vraća u stanje u kojem je bila u trenutku neposredno prije pokretanja (obavljanja) naredbi. Zbog toga je naredbu koja očekuje da su u bazi podataka sačuvani efekti neke druge, prethodne naredbe (ili više njih) potrebno obaviti zajedno sa svim tim naredbama.

Primjer 1.

Napisati naredbe kojima će se stvoriti relacije:

TIPVOZILO = {sifTipVozilo, nazTipVozilo}
PK_{TIPVOZILO} = {sifTipVozilo}

VOZILO = {sifVozilo, sifTipVozilo , nazVozilo, serijskiBroj, nosivost}
PK_{VOZILO} = {sifVozilo}
K2_{VOZILO} = {serijskiBroj}

Za attribute postavite navedene tipove podataka:

atribut	tip	duljina/raspon
Šifra tipa vozila	cijeli broj	
Naziv tipa vozila	niz znakova varijabilne duljine	max 50 znakova
Šifra vozila	cijeli broj	
Naziv vozila	niz znakova varijabilne duljine	max 50 znakova
Serijski broj vozila	niz znakova varijabilne duljine	max 20 znakova
Nosivost vozila u kilogramima	cijeli broj	[100, 50000]

- Pri tom osigurati:
- entitetski integritet i integritet ključa u obje relacije
 - integritet ključa K2 u relaciji vozilo(VOZILO).
 - nemogućnost pojave NULL vrijednosti za bilo koji atribut obje relacije
 - ograničenje nosivosti vozila na interval [100, 50000] (radi se o broju kilograma koje vozilo može nositi)
 - referencijski integritet pozivajuće relacije vozilo(VOZILO) s obzirom na pozivanu relaciju tipVozilo(TIPVOZILO)

Unesite nekoliko zapisa u obje tablice i pokušajte narušiti sva integritetska ograničenja koja ste definirali kako biste se upoznali s porukama koje SUBP javlja u slučaju pokušaja narušavanja integriteta.

rješenje

Feedback

```
CREATE TABLE TipVozilo(  
    sifTipVozilo SMALLINT CONSTRAINT pkTipVozilo PRIMARY KEY  
    , nazTipVozilo VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE Vozilo(  
    sifVozilo      INTEGER  
    , sifTipVozilo SMALLINT NOT NULL  
    , nazVozilo    VARCHAR(50) NOT NULL  
    , serijskiBroj VARCHAR(20) NOT NULL CONSTRAINT unqSerijskiBroj UNIQUE  
    , nosivost     INTEGER NOT NULL -- ne smije biti SMALLINT!  
    , CONSTRAINT pkVozilo PRIMARY KEY (sifVozilo)  
    , CONSTRAINT fkVoziloTipVozilo FOREIGN KEY (sifTipVozilo) REFERENCES TipVozilo  
    , CONSTRAINT chkNosivost CHECK (nosivost BETWEEN 100 AND 50000)  
);
```

Objekte relacije su prazne. Uvjerimo se u to:

```
SELECT * FROM TipVozilo;  
SELECT * FROM Vozilo;
```

Sljedeće naredbe obavljajte jednu po jednu (na vlastitom postgresu pomoću pgAdmina) i obratite pozornost na rezultat obavljanja:

```
INSERT INTO TipVozilo VALUES (1, 'Osobni automobil');
```

INSERT 0 1

Query returned successfully in 56 msec.

```
INSERT INTO TipVozilo VALUES (1, 'Kombi');
```

ERROR: duplicate key value violates unique constraint "pktipvozilo"

DETAIL: Key (siftipvozilo)=(1) already exists.

SQL state: 23505

```
INSERT INTO TipVozilo VALUES (2, 'Kombi');
```

INSERT 0 1

Query returned successfully in 44 msec.

```
INSERT INTO TipVozilo VALUES (3, 'Kamion');
```

INSERT 0 1

Query returned successfully in 43 msec.

Iako u relaciji tipVozilo sada postoje 3 ntorke (uvjerite se u to), u rezultatu sljedeće SQL naredbe se neće pojaviti nijedna ntorka. Zašto?

```
SELECT *  
    FROM Vozilo, TipVozilo  
    WHERE Vozilo.sifTipVozilo = TipVozilo.sifTipVozilo;
```

Zbog toga što u relaciji vozilo nema nijedne ntorka, pa nijedna ntorka iz tipVozilo neće zadovoljiti uvjet spajanja `Vozilo.sifTipVozilo = TipVozilo.sifTipVozilo`.

Unesite ntorka u vozilo i obratite pozornost na rezultat obavljanja:

```
INSERT INTO Vozilo VALUES (100, 1, 'V8 interceptor'    , 'MMX-123'    , 1200);
```

INSERT 0 1

Query returned successfully in 41 msec.

```
INSERT INTO Vozilo VALUES (100, 2, 'Kamion interceptor', 'KX-123'    , 12000);  
/* Greška: pkVozilo */
```

ERROR: duplicate key value violates unique constraint "pkvozilo"
DETAIL: Key (sifvozilo)=(100) already exists.
SQL state: 23505

```
INSERT INTO Vozilo VALUES (101, 3, 'Izmisljeni kamion' , 'KX-123-56', 62000);  
/* Greška: chkNosivost */
```

ERROR: new row for relation "vozilo" violates check constraint "chknosivost"
DETAIL: Failing row contains (101, 3, Izmisljeni kamion, KX-123-56, 62000).
SQL state: 23514

```
INSERT INTO Vozilo VALUES (102, 2, 'Caddy' , 'MMX-123' , 1500);  
/* Greška: unqSerijskiBroj */
```

ERROR: duplicate key value violates unique constraint "unqserijskibroj"
DETAIL: Key (serijskibroj)=(MMX-123) already exists.
SQL state: 23505

```
INSERT INTO Vozilo VALUES (103, 5, 'Caddy' , 'CDY-0-123', 1500);  
/* Greška: fkVoziloTipVozilo */
```

ERROR: insert or update on table "vozilo" violates foreign key constraint "fkvozilotipvozilo"
DETAIL: Key (siftipvozilo)=(5) is not present in table "tipvozilo".
SQL state: 23503

```
INSERT INTO Vozilo VALUES (104, 2, 'Caddy' , 'CDY-1-123', 1500);
```

INSERT 0 1
Query returned successfully in 50 msec.

```
INSERT INTO Vozilo VALUES (105, 2, NULL , 'CDY-2-123', 2500);  
/* Greška: naziv vozila ne smije biti NULL */
```

ERROR: null value in column "nazvozilo" of relation "vozilo" violates not-null constraint
DETAIL: Failing row contains (105, 2, null, CDY-2-123, 2500).
SQL state: 23502

Question

Playground

Ovaj zadatak se odnosi na bazu podataka studAdmin.

Osigurati referencijski integritet među relacijama **ispit** i **predmet**.

Napomena: Svi potrebni strani ključevi u bazi podataka studAdmin su već kreirani.

Strani ključ ispit -> predmet je nepotreban jer su definirani strani ključevi ispit->ispitni rok i ispitniRok -> predmet.

Ovaj primjer služi samo za vježbanje sintakse ALTER table naredbe.

rješenje

```
ALTER TABLE ispit  
ADD constraint fkIspitPredmet FOREIGN key (sifPredmet)  
REFERENCES predmet(sifPredmet);
```

1

2

3

4

Run

Save

Feedback

