



Zadatci za vježbu

#7

3/3

: Definiranje integritetskih ograničenja - kaskadno brisanje, kaskadna izmjena

Primjer 1.
Napisati naredbe kojima će se stvoriti relacije:

TECAJ = {sifTecaj, nazTecaj}
PK_{TECAJ} = {sifTecaj}

OSOBA= {sifOsoba, imeOsoba, prezOsoba}
PK_{OSOBA} = {sifOsoba}

POLAZNIK_TECAJ = {sifTecaj, sifOsoba, datumUpis, datumIspis}
PK_{POLAZNIK_TECAJ} = {sifTecaj, sifOsoba }

POLAZNIK_TECAJ_RATA = {sifTecaj, sifOsoba, datumRata, iznosRata}
PK_{POLAZNIK_TECAJ_RATA} = {sifTecaj, sifOsoba, datumRata }

Za attribute postavite navedene tipove podataka:

atribut	tip	duljina/raspon
Šifra tečaja	cijeli broj	
Naziv tečaja	niz znakova varijabilne duljine	max 50 znakova
Šifra osobe	cijeli broj	
Ime osobe	niz znakova varijabilne duljine	max 50 znakova
prezimeosobe	niz znakova varijabilne duljine	max 50 znakova
Datum upisa polaznika na tečaj	datum	
Datum ispisa polaznika s tečaja	datum	
Datum dospijeća rate	datum	
Iznos rate	decimalni broj	ukupno 7 mjesta od čega 2 decimalna

- Pri tom osigurati:
- samo atribut datumIspis relacije polaznik_tecaj(POLAZNIK_TECAJ) smije poprimiti NULL vrijednost
 - entitetski integritet i integritet ključa u obje relacije
 - referencijski integritet pozivajuće relacije polaznik_tecaj(POLAZNIK_TECAJ) s obzirom na pozivane relacije tecaj(TECAJ) i osoba(OSOBA), kao i referencijski integritet pozivajuće relacije polaznik_tecaj_rata(POLAZNIK_TECAJ_RATA) s obzirom na pozivanu relaciju polaznik_tecaj(POLAZNIK_TECAJ)
 - datum ispisa mora biti veći od datum upisa polaznika tečaja
 - iznos rate mora biti veći od 0
 - pri brisanju osobe ili tečaja automatski se brišu i svi zapisi iz relacije polaznik_tecaj(POLAZNIK_TECAJ) vezani za tu osobu ili tečaj, kao i zapisi iz relacije polaznik_tecaj_rata(POLAZNIK_TECAJ_RATA)
 - pri promjeni šifre osobe u relaciji OSOBA automatski se mijenja šifra osobe u relacijama polaznik_tecaj(POLAZNIK_TECAJ) i polaznik_tecaj_rata(POLAZNIK_TECAJ_RATA) na tu novu vrijednost

Unesite nekoliko zapisa u sve tablice, pokušajte brisati/mijenjati zapise relacija tecaj(TECAJ) i osoba(OSOBA), te analizirajte sadržaj relacija polaznik_tecaj(POLAZNIK_TECAJ) i polaznik_tecaj_rata(POLAZNIK_TECAJ_RATA) nakon obavljenih naredbi.

rješenje

```
CREATE TABLE tecaj(  
    sifTecaj INTEGER CONSTRAINT pkTecaj PRIMARY KEY  
    , nazTecaj VARCHAR(50) NOT NULL  
);  
  
CREATE TABLE osoba(  
    sifOsoba INTEGER CONSTRAINT pkOsoba PRIMARY KEY  
    , imeOsoba VARCHAR(50) NOT NULL  
    , prezOsoba VARCHAR(50) NOT NULL  
);  
  
CREATE TABLE polaznik_tecaj(  
    sifOsoba INTEGER  
    , sifTecaj INTEGER  
    , datumUpis DATE NOT NULL  
    , datumIspis DATE  
    , CONSTRAINT pkPolaznikTecaj PRIMARY KEY (sifOsoba, sifTecaj)  
    , FOREIGN KEY (sifOsoba) REFERENCES osoba (sifOsoba)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
    , FOREIGN KEY (sifTecaj) REFERENCES tecaj (sifTecaj)  
        ON DELETE CASCADE  
    , CHECK (datumIspis IS NULL or  
        (datumIspis IS NOT NULL and datumIspis > datumUpis))  
);  
  
CREATE TABLE polaznik_tecaj_rata(  
    sifOsoba INTEGER  
    , sifTecaj INTEGER  
    , datumRata DATE NOT NULL  
    , iznosRata DECIMAL (7,2) NOT NULL  
    , CONSTRAINT pkPolaznikTecajRata PRIMARY KEY (sifOsoba, sifTecaj, datumRata)  
    , FOREIGN KEY (sifOsoba, sifTecaj)  
        REFERENCES polaznik_tecaj (sifOsoba, sifTecaj)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
    , CHECK (iznosRata > 0)  
);
```

Obaviti naredbe za unos ntorki u relacije:

```
INSERT INTO tecaj VALUES (1, 'Baze podataka');  
INSERT INTO tecaj VALUES (2, 'Uvod u baze podataka');  
  
INSERT INTO osoba VALUES (1, 'Ante', 'Zenić');  
INSERT INTO osoba VALUES (2, 'Laura', 'Krolo');  
INSERT INTO osoba VALUES (3, 'Dino', 'Balić');  
  
INSERT INTO polaznik_tecaj VALUES (1, 1, '10.12.2007', NULL);  
INSERT INTO polaznik_tecaj VALUES (2, 2, '10.6.2007', NULL);  
INSERT INTO polaznik_tecaj VALUES (3, 1, '10.12.2007', '10.12.2009');  
INSERT INTO polaznik_tecaj VALUES (3, 2, '10.12.2007', '10.12.2009');  
  
INSERT INTO polaznik_tecaj_rata VALUES (3, 2, '10.12.2007', 100);  
INSERT INTO polaznik_tecaj_rata VALUES (2, 2, '10.12.2007', 150);  
INSERT INTO polaznik_tecaj_rata VALUES (3, 1, '10.12.2007', 200);  
INSERT INTO polaznik_tecaj_rata VALUES (3, 2, '5.12.2007', 600);
```

Primjeri SQL naredbi kod kojih će doći do izražaja kaskadno brisanje (ON DELETE CASCADE):

1. DELETE FROM osoba WHERE sifOsoba = 2;

Pored zapisa iz osoba obrisat će se i zapisi iz polaznik_tecaj zbog stranog ključa:

```
CREATE TABLE polaznik_tecaj(  
    ...  
    , FOREIGN KEY (sifOsoba) REFERENCES osoba (sifOsoba)  
        **ON DELETE CASCADE**  
        ON UPDATE CASCADE  
    ...
```

ali i zapisi iz polaznik_tecaj_rata zbog stranog ključa:

```
CREATE TABLE polaznik_tecaj_rata(  
    ...  
    FOREIGN KEY (sifOsoba, sifTecaj)  
        REFERENCES polaznik_tecaj (sifOsoba, sifTecaj)  
        **ON DELETE CASCADE**  
        ON UPDATE CASCADE  
    ...
```

Da su zapisi za osobu sa sifOsoba = 2 obrisani iz sve 3 relacije možemo provjeriti sljedećim SQL naredbama:

```
SELECT * FROM osoba WHERE sifOsoba = 2;  
SELECT * FROM polaznik_tecaj WHERE sifOsoba = 2;  
SELECT * FROM polaznik_tecaj_rata WHERE sifOsoba = 2;
```

2. DELETE FROM tecaj WHERE sifTecaj = 1;

Pored zapisa iz osoba obrisat će se i zapisi iz polaznik_tecaj zbog stranog ključa:

```
CREATE TABLE polaznik_tecaj(  
    ...  
    FOREIGN KEY (sifTecaj) REFERENCES tecaj (sifTecaj)  
        ON DELETE CASCADE  
    ...
```

ali i zapisi iz polaznik_tecaj_rata zbog stranog ključa:

```
CREATE TABLE polaznik_tecaj_rata(  
    ...  
    FOREIGN KEY (sifOsoba, sifTecaj)  
        REFERENCES polaznik_tecaj (sifOsoba, sifTecaj)  
        **ON DELETE CASCADE**  
        ON UPDATE CASCADE  
    ...
```

Da su zapisi za tečaj sa sifTecaj = 1 obrisani iz sve 3 relacije možemo provjeriti sljedećim SQL naredbama:

```
SELECT * FROM tecaj WHERE sifTecaj = 1;  
SELECT * FROM polaznik_tecaj WHERE sifTecaj = 1;  
SELECT * FROM polaznik_tecaj_rata WHERE sifTecaj = 1;
```

Primjer SQL naredbe kod koje će doći do izražaja kaskadna izmjena (ON UPDATE CASCADE):

```
UPDATE osoba SET sifOsoba = 5 WHERE sifOsoba = 3;
```

Pored zapisa iz osoba izmijenit će se sifOsoba za zapise iz polaznik_tecaj zbog stranog ključa:

```
CREATE TABLE polaznik_tecaj(  
    ...  
    , FOREIGN KEY (sifOsoba) REFERENCES osoba (sifOsoba)  
        ON DELETE CASCADE  
        **ON UPDATE CASCADE**  
    ...
```

ali i zapisi iz polaznik_tecaj_rata zbog stranog ključa:

```
CREATE TABLE polaznik_tecaj_rata(  
    ...  
    FOREIGN KEY (sifOsoba, sifTecaj)  
        REFERENCES polaznik_tecaj (sifOsoba, sifTecaj)  
        ON DELETE CASCADE  
        **ON UPDATE CASCADE**  
    ...
```

Da je zapisima sa sifOsoba = 5 promijenjena sifOsoba na vrijednost 5 u sve 3 relacije možemo provjeriti sljedećim SQL naredbama:

```
SELECT * FROM osoba WHERE sifOsoba IN (3, 5);  
SELECT * FROM polaznik_tecaj WHERE sifOsoba IN (3, 5);  
SELECT * FROM polaznik_tecaj_rata WHERE sifOsoba IN (3, 5);
```

Nema zapisa za sifOsoba = 3. Postoje samo zapisi za sifOsoba = 5.

Relacije možete obrisati sljedećim naredbama:

```
DROP TABLE polaznik_tecaj_rata;  
DROP TABLE polaznik_tecaj;  
DROP TABLE tecaj;  
DROP TABLE osoba;
```

[Question](#)

[Playground](#)

Ovaj zadatak se odnosi na bazu podataka studAdmin.

Osigurajte da atribut **evidencijaBoravka.datumVrijemeKraj** bude ili NULL ili veći od **evidencijaBoravka.datumVrijemePoc**.

rješenje

ALTER TABLE evidencijaBoravka ADD CONSTRAINT chkUlazIzlaz CHECK (datumVrijemeKraj IS NULL OR datumVrijemeKraj > datumVrijemePoc);

1

2

3

4

5

6

Run

Save