

-BAZE PODATAKA- 2010

SADRŽAJ

1. TEORIJA	2
2. BCNF Normalizacija	19
3. PREVOĐENJE IZ PMOV-a U RELACIONI I ODL NAREDBE...	23
4. XML	25
5. TRANSAKCIJE	30
6. SQL	31
7. PMOV	34

1. TEORIJA

Ovde su izvučena sva moguća pitanja koja se mogu naći na njihovim rokovima okačenim na sajtu, dakle 2001-2009. Pošto rokovi iz 2010 nisu dostupni na sajtu Baza u PDF formatu, slike rokova su okačene na ovom [linku](#).

Odgovori na sva pitanja se mogu naći u knjizi, a ostaviću odgovore na neka od njih koja su (barem meni) bila problematična.

6. Oporavak baze podataka. (10 poena)

7. Definisane složene tipove podataka u objektno-relacionim sistemima. Navesti konstruktore složenih tipova i objasniti. (10 poena)

7. Ograničenja u relacionom modelu. (8 poena)

8. Definisane složene tipove podataka u objektno-relacionim sistemima. Navesti konstruktore složenih tipova i objasniti. (8 poena)

9. Objasniti vremensko označavanje transakcija (timestamping). (8 poena)

7. Navesti i ukratko objasniti vrste lokota i odgovarajuće protokole. (7 poena)

8. Definisane složene tipove podataka u objektno-relacionim sistemima. Navesti konstruktore složenih tipova i objasniti. (8 poena)

9. Objasniti vremensko označavanje transakcija (timestamping). (7 poena)

6. Posmatrajmo sledeće dve transakcije:

T1: READ A;

READ B;

IF A = 0 THEN B := B + 1;

T2: READ B;

READ A;

IF B = 0 THEN A := A + 1;

a) Koja su moguća konzistentna stanja baze (vrednost podataka A i B) posle izvršenja ovih transakcija ako su početne vrednosti $A=0$; $B=0$; (5 poena)

b) Dodati LOCK i UNLOCK instrukcije u transakcije T1 i T2 tako da se ostvari "dvofazni protokol zaključavanja". Može li izvršenje ovako konstruisanih transakcija da dovede do "mrtvog čvora"? Pokazati da dvofazni protokol zaključavanja garantuje serijabilnost izvršenja ovih transakcija. (5 poena)

7. Objasniti koncept pogleda u relacionom modelu? Prikazati i objasniti SQL:1999 sintaksu naredbe za kreiranje pogleda. (10 poena)

8. Osnovne karakteristike objektno-relacionih sistema. (10 poena)

7. Dato je kokurentno izvršenje S1 dve transakcije T1 i T2.

S1: LS1(X), R1(X), LS2(Y), R2(Y), LS2(X), R2(X), UN2(X), LE1(X), W1(X) COMMIT1, LE2(Y), W2(Y), COMMIT2 ;

Sa X i Y su označeni objekti baze podataka kojima transakcije pristupaju. Operacije, čiji su argumenti objekti baze podataka, se označavaju na sledeći način:

LS() – postavljanje deljivog lokota

LE() – postavljanje ekskluzivnog lokota

R() - čitanje

W() – upisivanje

UN – skidanje lokota

Indeks uz operaciju označava transakciju koja obavlja posmatranu transakciju.

Ispitati serijabilnost izvršenja transakcija i obrazložiti odgovor.

Kako se dodavanjem ili brisanjem odogovarajućih operacija može omogućiti serijabilno tj. ne serijabilno izvršavanje datih transakcija.

(10 poena)

8. Objektne karakteristike objektno-relacionih sistema. Navesti ih i objasniti.

(8 poena)

9. Navesti sve dodatne operacije relacije algebre koje su uvedene zbog postojanja nula vrednosti u bazi podataka i dati primere. *(8 poena)*

3. Navesti i ukratko objasniti vrste lokota i odgovarajuće protokole. *(8 poena)*

4. Oporavak baze podataka. *(8 poena)*

5. Objasniti i dati primer za sledeće OQL koncepte:

a) Definisane ("iteratorske") promenljive

b) OQL-putanja

c) Struktura kao rezultat izvršavanja OQL upita

(10 poena)

5. Definisane sopstvenih osnovnih tipova podataka u objektno-relacionim sistemima. Prikazati primer. *(15 poena)*

6. Koje vrste nasledjivanja postoje u objektnim bazama (ODMG standard). Navesti ih, objasniti i uporediti. *(7 poena)*

7. Relacioni račun domena. *(7 poena)*

8. Objasniti koncept "atomnosti" i serijabilnosti transakcija. *(6 poena)*

6. Opštenamenski sistem pravila u objektno-relacionim sistemima. Prikazati primer. *(15 poena)*

7. Ažuriranje relacionih pogleda. *(7 poena)*

Odgovor: Nabranje na dnu 151. strane

8. Navesti, objasniti i dati primer za osnovne operacije relacije algebre. *(6 poena)*

9. Navesti i ukratko objasniti vrste lokota i odgovarajuće protokole. *(7 poena)*

6. Koje vrste nasledjivanja postoje u objektnim bazama (ODMG standard). Navesti ih, objasniti i uporediti. *(7 poena)*

7. Objasniti koncept "atomnosti" i serijabilnosti transakcija. (Navesti primer) *(7 poena)*

5. Definisane sopstvenih osnovnih tipova podataka u objektno-relacionim sistemima. Prikazati primer. *(12 poena)*

6. Koje vrste nasledjivanja postoje u objektnim bazama (ODMG standard). Navesti ih, objasniti i uporediti. *(7 poena)*

7. Ažuriranje relacionih pogleda. *(7 poena)*

8. Navesti i ukratko objasniti vrste lokota i odgovarajuće protokole. *(7 poena)*

6. Koje vrste nasljedjivanja postoje u objektnim bazama (ODMG standard). Navesti ih, objasniti i uporediti. (7 poena)
7. Navesti, objasniti i dati primer za osnovne operacije relacione algebre. (7 poena)
8. Navesti i ukratko objasniti vrste lokota i odgovarajuće protokole. (7 poena)

6. Koje vrste nasljedjivanja postoje u objektnim bazama (ODMG standard). Navesti ih, objasniti i uporediti. (7 poena)
7. Definisati koncepte i algoritam za oporavak baze podataka. (7 poena)
8. Dvofazni commit protokol. (6 poena)

3. Operacija deljenja relacione algebre: dati definiciju, primer i postupak izvođenja na osnovu drugih operacija relacione algebre. (10 poena)
4. Dvofazni Commit protokol. (9 poena)
5. Objektne karakteristike objektno-relacionih sistema. Navesti ih i objasniti. (10 poena)

3. Dinamicka pravila integriteta u relacionom modelu (9 poena)
4. Oporavak baze podataka. (10 poena)
5. Konstruisani tipovi u objektno-relacionim sistemima. Ukratko objasniti i ilustrovati primerom. (10 poena)

5. Oporavak baze podataka. (10 poena)
6. Definisane slozenih tipova podataka u objektno-relacionim sistemima. Navesti konstruktore slozenih tipova i objasniti. (10 poena)
7. Navesti i ukratko objasniti vrste lokota i odgovarajuće protokole. (7 poena)

3. Operacija deljenja relacione algebre: dati definiciju, primer i postupak izvođenja na osnovu drugih operacija relacione algebre. (9 poena)
4. Objasniti vremensko oznacavanje transakcija (timestamping). (10 poena)
5. Korisnicki definisani tipovi podataka u objektno-relacionim sistemima. Prikazati sintaksu i ilustrovati primerom. (10 poena)

3. Spoljna unija. Objasniti operaciju i dati primer. (9 poena)
4. Navesti i ukratko objasniti vrste lokota i odgovarajuće protokole. (10 poena)
5. Navesti i objasniti moguće načine povezivanja podataka u objektno-relacionim sistemima. Dati primer za svaki način povezivanja.

Odgovor na 5. pitanje: pri vrhu 192. strane

3. Operacija deljenja relacione algebre: dati definiciju, primer i postupak izvođenja na osnovu drugih operacija relacione algebre. (8 poena)
4. Dvofazni Commit protokol. (7 poena)
5. Navesti i objasniti moguće načine povezivanja podataka u objektno-relacionim sistemima. Dati primer za svaki način povezivanja. (10 poena)

3. Navesti sve dodatne operacije relacione algebre koje su uvedene zbog postojanja nula vrednosti u bazi podataka i dati primere. (9 poena)
4. Oporavak baze podataka.

3. Uporediti sledeće termine:

- (a) kandidat za ključ; determinanta,
 - (b) nepotpuna funkcionalna zavisnost; tranzitivna zavisnost,
 - (c) relacija; tabela,
 - (d) integritet entiteta; referencijalni integritet.
- (8 poena)

5. Posmatrajmo sledeće dve transakcije:

```
T1: READ A;  
READ B;  
IF A = 0 THEN B := B + 1;  
T2: READ B;  
READ A;  
IF B = 0 THEN A := A + 1;
```

a) Koja su moguća konzistentna stanja baze (vrednost podataka A i B) posle izvršenja ovih transakcija ako su početne vrednosti $A=0$; $B=0$;

b) Dodati LOCK i UNLOCK instrukcije u transakcije T1 i T2 tako da se ostvari "dvofazni protokol zaključavanja". Može li izvršenje ovako konstruisanih transakcija da dovede do "mrtvog čvora"? Pokazati da dvofazni protokol zaključavanja garantuje serijabilnost izvršenja ovih transakcija.

(10 poena)

6. Navesti i objasniti moguće načine povezivanja podataka u objektno-relacionim sistemima. Dati primer za svaki način povezivanja. (10 poena)

(2003)

To su bila pitanja iz perioda 2001-2003 a zadaci koji se javljaju u ovim pitanjima su malo konfuzni i ne javljaju se u kasnijim rokovima tako da ne bih savetovao da se bacete njima.

3. Katalog baze podataka. (9 poena)

4. Ograničenja u relacionom modelu. (10 poena)

Odgovor: 152-158 strane.

5. Objasniti definicije i dati primer za ulazne tačke, definisanje promenljivih i osnovni upitni blok OQL-a. Može se koristiti ODL šema rešenja zadatka 2b. (10 poena)

3. Prikazati i objasniti ANSI/SPARC arhitekturu sistema za upravljanje bazom podataka. (9 poena)

4. Protokoli zaključavanja. (10 poena)

5. Osnovne karakteristike i klasifikacija SQL:1999 trigeri. (10 poena)

3. Objasniti koncept pogleda u relacionom modelu. Prikazati i objasniti SQL:1999 sintaksu naredbe za kreiranje pogleda.

(9 poena)

4. Protokoli zaključavanja. (10 poena)

5. Nasleđivanje u objektnim bazama podataka. Navesti primere i objasniti ih. (10 poena)

3. Spoljna unija. Objasniti operaciju i dati primer. (9 poena)

4. Vremensko označavanje transakcija. (10 poena)

5. Nasleđivanje u objektnim bazama podataka. (10 poena)

3. Konstruisani tipovi objektno-relacionog modela. (9 poena)

4. Katalog baze podataka; model objekti-veze za relacioni katalog. (10 poena)

5. Osnovne karakteristike i klasifikacija SQL:1999 trigeri. (10 poena)

3. Navesti sve dodatne operacije relacione algebre koje su uvedene zbog postojanja nula vrednosti u bazi podataka i OBAVEZNO dati primere. (9 poena)

4. Vremensko označavanje transakcija. (10 poena)

5. Osnovne karakteristike i klasifikacija SQL:1999 trigeri. (10 poena)

3. Spoljna unija. Objasniti operaciju i OBAVEZNO dati primer. (9 poena)

4. Protokoli zaključavanja. (10 poena)

5. Objasniti definicije i dati primer za ulazne tačke, definisanje promenljivih i osnovni upitni blok OQL-a. Može se koristiti ODL šema rešenja zadatka 2b. (10 poena)

3. Operacija deljenja relacione algebre: dati definiciju, primer i postupak izvođenja na osnovu drugih operacija relacione algebre. (9 poena)

4. Objasniti pojmove „život“ i „mrtvog“ lokota i opisati načine razrešavanja. (10 poena)

5. Osnovne karakteristike i klasifikacija SQL:1999 trigeri. (10 poena)

3. Fizičko projektovanje relacionih baza podataka. (9 poena)

4. Oporavak u distribuiranim bazama podataka. (10 poena)

5. Nasleđivanje u objektnim bazama podataka. (10 poena)

(2004)

3. Objasniti sličnosti i razlike između sledećih koncepata:

a) Relaciona algebra; Relacioni račun

b) Vrednosna ograničenja; Strukturna ograničenja u Modelu objekti-veze

c) Operacija spajanja; Operacija unije u Relacionom modelu

d) Distinkt tip; Strukturirani tip u Objektno-relacionom modelu

e) Deljivi lokot; Ekskluzivni lokot

(10 poena)

4. Šta je Transakcija? Objasniti skup osobina koje transakcija mora da poseduje. (10 poena)

5. Objasniti pojam „Rečnik podataka“. Rečnik baze podataka u SQL okruženju. MOV za relacioni rečnik. (11 poena)

3. Nacrtati sliku i objasniti ANSI/SPARC trionivosku arhitekturu. (9 poena)

4. Rečnik podataka. (10 poena)

5. Objasniti sledeće koncepte:

a) Determinanta; Kandidat za ključ

b) Potpuna funkcionalna zavisnost; Tranzitivna zavisnost

c) Uslovi koje treba tabela da zadovolji da bi bila relacija

d) Integritet entiteta; Referencijalni integritet

e) Objasniti dvofazni protokol zaključavanja (10 poena)

3. Navesti sve dodatne operacije relacione algebre koje su uvedene zbog postojanja nula vrednosti u bazi podataka i OBAVEZNO dati primere. (10 poena)
4. Objasniti pojmove „živog lokota“ i „mrtvog lokota“. Navesti i objasniti tehnike za razrešavanje „mrtvih lokota“. (10 poena)
5. Osnovne karakteristike i klasifikacija SQL:1999 trigeri. (11 poena)

3. Vrste ograničenja u relacionom modelu. Pokazati na primeru definisanje ograničenja. Kada se proveravaju ograničenja? (10 poena)
4. Katalog baze podataka. (10 poena)
5. Vremensko označavanje transakcija. (11 poena)

3. Vrste ograničenja u relacionom modelu. Pokazati na primeru definisanje ograničenja. Kada se proveravaju ograničenja? (9 poena)
4. Prikazati šemu komponenti Sistema za upravljanje bazama podataka i opisati ulogu svake komponente. (10 poena)
5. Objasniti sličnosti i razlike između sledećih koncepata:
 - a) Relaciona algebra; Relacioni račun
 - b) Vrednosna ograničenja; Strukturna ograničenja u Modelu objekti-veze
 - c) Operacija spajanja; Operacija unije u Relacionom modelu
 - d) Distinkt tip; Strukturirani tip u Objektno-relacionom modelu
 - e) Deljivi lokot; Ekskluzivni lokot(12 poena)

3. Navesti sve dodatne operacije relacione algebre koje su uvedene zbog postojanja nula vrednosti u bazi podataka i OBAVEZNO dati primere. (10 poena)
4. Objasniti pojam „Rečnik podataka“. Rečnik baze podataka u SQL okruženju. MOV za relacioni rečnik. (11 poena)
5. Šta je Transakcija? Objasniti skup osobina koje transakcija mora da poseduje. (10 poena)

3. Vrste ograničenja u relacionom modelu. Pokazati na primeru definisanje ograničenja. Kada se proveravaju ograničenja? (10 poena)
4. Prikazati šemu komponenti Sistema za upravljanje bazama podataka i opisati ulogu svake komponente. (10 poena)
5. Objasniti pojmove „živog lokota“ i „mrtvog lokota“. Navesti i objasniti tehnike za razrešavanje „mrtvih lokota“. (11 poena)

(2005)

- 3.** Konstruisani tipovi objektno-relacionog modela (dati primer). (10 poena)
- 4.** Osnovna struktura XML dokumenta (dati primer). Koje uslove mora da zadovolji dobro-iformljen XML dokument? Šta je validan XML dokument? (11 poena)
- 5.** Prikazati šemu komponenti Sistema za upravljanje bazama podataka i opisati ulogu svake komponente. (10 poena)
- 3.** Navesti sve Konvencionalne skupovne operacije i Specijalne relacione operacije relacione algebre. **OBAVEZNO** dati primere za svaku vrstu operacije. (10 poena)
- 4.** Prikazati šemu komponenti Sistema za upravljanje bazama podataka i opisati ulogu svake komponente. (11 poena)
- 5.** Semantika i obrada ECA pravila. ---
- 3.** Relacioni račun n-torki. Objasniti i dati primer. (10 poena)
- 4.** Semantika i obrada ECA pravila. (11 poena)
- 5.** Katalog baza podataka. Model objekti-veze za relacioni katalog. ---
- 3.** Prikazati arhitekturu objektnih SUBP i dati opis svih komponenti. (10 poena)
- 4.** Navesti operacije sa XML dokumentima. Prikazati osnovni oblik ovih operacija. (11 poena)
- 5.** Objasniti pojmove „živog lokota“ i „mrtvog lokota“. Navesti i objasniti tehnike za razrešavanje „mrtvih lokota“.
- 3.** Vrste ograničenja u relacionom modelu. Pokazati na primeru definisanje ograničenja. Kada se proveravaju ograničenja? (10 poena)
- 4.** Osnovne karakteristike i klasifikacija SQL:1999 trigera. (10 poena)
- 5.** Vremensko označavanje transakcija. (11 poena)

(2006)

3. Vremensko označavanje transakcija. (10 poena)

4. Date su tabele $R(P,A)$ i $S(F,B)$. Uz pretpostavku da postoji referencijalno ograničenje: $S.F$ je spoljni ključ koji se referencira na primarni $R.P$, napisati trigger na nivou naredbe kojim se implementira dinamičko pravilo integriteta "on update default" i „on delete nullifies“. (10 poena)

Odgovor na 4. pitanje:

```
Create trigger on_update_default
After update on R
referencing old table as oldr
for each statement
begin atomic
update s
set f = default
where f in (select p from oldr)
end;
```

```
Create trigger on_delete_nullifies
After delete on R
referencing old table as oldr
for each statement
begin atomic
update s
set f = null
where f in (select p from oldr)
end;
```

5.a) Navesti i objasniti osnovne elemente grafa prethođenja transakcija. Dati definiciju kada T_i prethodi T_j . (4 poena)

b) Proveriti da li postoji konflikt-serijabilnost izvršenja S_1 skupa transakcija, a posle i S_2 . Dati obrazloženje.

S_1 : $r_3(X), r_3(Y), w_3(X), r_1(Y), w_1(Y), r_2(X), w_2(X), w_2(Y)$

S_2 : $r_3(X), r_3(Y), r_1(Y), w_1(Y), r_2(X), w_2(X), w_2(Y), w_3(X)$

3. Spoljno spajanje. Objasniti svaku vrstu spoljnog spajanja i OBAVEZNO dati primere. (10 poena)

4. Vremensko označavanje transakcija. (10 poena)

5.a) Navesti i objasniti osnovne elemente grafa prethođenja transakcija. Dati definiciju kada T_i prethodi T_j . (4 poena)

b) Proveriti da li postoji konflikt-serijabilnost izvršenja S_1 skupa transakcija, a posle i S_2 . Dati obrazloženje.

S_1 : $r_3(X), r_3(Y), w_3(X), r_1(Y), w_1(Y), r_2(X), w_2(X), w_2(Y)$

S_2 : $r_3(X), r_3(Y), r_1(Y), w_1(Y), r_2(X), w_2(X), w_2(Y), w_3(X)$ (6 poena)

3. Spoljno spajanje. Objasniti svaku vrstu spoljnog spajanja i OBAVEZNO dati primere. (10 poena)

4. Date su tabele $R(P,A)$ i $S(F,B)$. Uz pretpostavku da postoji referencijalno ograničenje: $S.F$ je spoljni ključ koji se referencira na primarni $R.P$, napisati trigger na nivou naredbe kojim se implementira dinamičko pravilo integriteta "on update default" i „on delete nullifies“. (10 poena)

5.a) Navesti i objasniti osnovne elemente grafa prethođenja transakcija. Dati definiciju kada T_i prethodi T_j . (4 poena)

b) Proveriti da li postoji konflikt-serijabilnost izvršenja S_1 skupa transakcija, a posle i S_2 . Dati obrazloženje.

S_1 : $r_3(X), r_3(Y), w_3(X), r_1(Y), w_1(Y), r_2(X), w_2(X), w_2(Y)$

S_2 : $r_3(X), r_3(Y), r_1(Y), w_1(Y), r_2(X), w_2(X), w_2(Y), w_3(X)$

(6 poena)

3. Navesti sve dodatne operacije relacione algebre koje su uvedene zbog postojanja nula vrednosti u bazi podataka i OBAVEZNO dati primere

(10 poena)

4. Date su tabele $R(P,A)$ i $S(F,B)$. Uz pretpostavku da postoji referencijalno ograničenje: $S.F$ je spoljni ključ

koji se referencira na primarni $R.P$, napisati trigger na nivou naredbe kojim se implementira dinamičko pravilo integriteta "on update default" i „on delete nullifies“.

(10 poena)

5.a) Proveriti da li postoji konflikt-serijabilnost izvršenja S1 skupa transakcija. Dati obrazloženje.

S1: $r1(X), r2(X), r2(Y), w2(X), r3(Y), w3(Y), r3(Z), w1(Z)$.

(4 poena)

b) Poređati operacije tako da novodobijeno izvršenje bude konflikt-serijabilno, poštujući pravila o zameni mesta

operacija. Nacrtati graf prethođenja za novo izvršenje.

(6 poena)

3. Napisati SQL naredbe kojima se kreira objektno-relaciona tabela *Prodaja* sa atributima knjižara, knjiga i cena, kao i potrebni tipovi. Svi atributi su definisani preko navedenih objektnih tipova.

- Knjižaru definisati kao *tip vrsta* sačinjenu od polja naziv i lokacija.

- Knjigu definisati kao *struktuirani tip* sa poljima naslov, autor i godinaizdanja.

- Cenu definisati preko imenovanog *distinct tipa* sa mogućnošću implicitne konverzije u INTEGER predefinisani tip. (10 poena)

Odgovor:

```
CREATE TYPE TipCena AS INTEGER FINAL  
CAST (DISTINCT AS SOURCE) WITH INTEGER;
```

```
CREATE TYPE TipKnjiga AS (naslov VARCHAR(20),  
autor VARCHAR(20),  
godinaizdanja DATETIME)
```

```
NOT FINAL;
```

```
CREATE TABLE Prodaja (knjizara ROW (naziv VARCHAR(20),  
lokacija VARCHAR(20)),  
knjiga TipKnjiga,  
cena TipCena);
```

4. Date su tabele $R(P,A)$ i $S(F,B)$. Uz pretpostavku da postoji referencijalno ograničenje: $S.F$ je spoljni ključ koji se referencira na primarni $R.P$, napisati trigger na nivou naredbe kojim se implementira dinamičko pravilo integriteta "on delete cascade". (10 poena)

5.a) Navesti i objasniti osnovne elemente grafa prethođenja transakcija. Dati definiciju kada T_i prethodi T_j . (4 poena)

b) Proveriti da li postoji konflikt-serijabilnost izvršenja S1 skupa transakcija, a posle i S2. Dati obrazloženje.

S1: $r3(X), r3(Y), w3(X), r1(Y), w1(Y), r2(X), w2(X), w2(Y)$

S2: $r3(X), r3(Y), r1(Y), w1(Y), r2(X), w2(X), w2(Y), w3(X)$

(6 poena)

3. Objasniti sličnosti i razlike između sledećih koncepata:

a) Relaciona algebra; Relacioni račun

b) Vrednosna ograničenja; Strukturna ograničenja u Modelu objekti-veze

- c) Operacija spajanja; Operacija unije u Relacionom modelu
- d) Distinkt tip; Strukturirani tip u Objektno-relacionom modelu
- e) Deljivi lokot; Ekskluzivni lokot

(10 poena)

4. Date su tabele Kartica(Proizvod, Datum, TipPromene, Kolicina) i Stanje(Proizvod, KolStanje). Atribut TipPromene uzima vrednosti iz skupa ('Ulaz', 'Izlaz').

a) napisati triger kojim se pri brisanju promena u kartici azurira ukupno stanje u tabeli Stanje.

b) napisati triger koji se zabranjuje da atribut KolStanje ima negativnu vrednost.

(10 poena)

odgovor:

uzas je napisao:

```
CREATE OR REPLACE TRIGGER ZABRANA_BRISANJA_I_AZURIRANJA
BEFORE DELETE OR UPDATE OF Proizvod ON Kartica
FOR EACH ROW
DECLARE
broj INTEGER;
greska EXCEPTION;
CURSOR pronadji (PR NUMBER) IS
    SELECT Proizvod FROM Stanje WHERE Proizvod = PR;
BEGIN
OPEN pronadji (:old.Proizvod);
FETCH pronadji INTO broj;
IF pronadji%FOUND THEN
RAISE greska;
END IF;
CLOSE pronadji;
END;
```

5.a) Navesti i objasniti osnovne elemente grafa prethodjenja transakcija. Dati definiciju kada Ti prethodi Tj.

(4 poena)

b) Proveriti da li postoji konflikt-serijabilnost izvršenja S1 skupa transakcija, a posle i S2. Dati obrazloženje.

S1: r1(X), r1(Y), w1(X), r3(Y), w3(Y), r2(X), w2(X), w2(Y)

S2: r1(X), r1(Y), r3(Y), w3(Y), r2(X), w2(X), w2(Y), w1(X)

(6 poena)

3. Objasniti sličnosti i razlike između sledećih koncepata:

- a) Relaciona algebra; Relacioni račun
- b) Vrednosna ograničenja; Strukturna ograničenja u Modelu objekti-veze
- c) Operacija spajanja; Operacija unije u Relacionom modelu
- d) Distinkt tip; Strukturirani tip u Objektno-relacionom modelu
- e) Deljivi lokot; Ekskluzivni lokot(10 poena)

4. Date su tabele Kartica(Proizvod, Datum, TipPromene, Kolicina) i Stanje(Proizvod, KolStanje). Atribut TipPromene uzima vrednosti iz skupa ('Ulaz', 'Izlaz').

a) napisati triger kojim se pri unosu promene u karticu ažurira ukupno stanje u tabeli Stanje.

b) napisati triger koji se zabranjuju operacije brisanja i ažuriranja u tabeli Kartica.(10 poena)

5.a) Navesti i objasniti osnovne elemente grafa prethođenja transakcija. Dati definiciju kada Ti prethodi Tj.(4 poena)

b) Proveriti da li postoji konflikt-serijabilnost izvršenja S1 skupa transakcija, a posle i S2. Dati obrazloženje.

S1: r2(X), r2(Y), w2(X), r1(Y), w1(Y), r3(X), w3(X), w3(Y)

S2: r2(X), r2(Y), r1(Y), w1(Y), r3(X), w3(X), w3(Y), w2(X)(6 poena)

3. Objasniti sličnosti i razlike između sledećih koncepata:

a) Relaciona algebra; Relacioni račun

b) Vrednosna ograničenja; Strukturna ograničenja u Modelu objekti-veze

c) Operacija spajanja; Operacija unije u Relacionom modelu

d) Distinkt tip; Strukturirani tip u Objektno-relacionom modelu

e) Deljivi lokot; Ekskluzivni lokot(10 poena)

4. Date su tabele Kartica(Proizvod, Datum, TipPromene, Kolicina) i Stanje(Proizvod, KolStanje). Atribut TipPromene uzima vrednosti iz skupa ('Ulaz', 'Izlaz').

a) napisati trigger kojim se pri unosu promene u karticu ažurira ukupno stanje u tabeli Stanje.

b) napisati trigger koji se zabranjuje operacije brisanja i ažuriranja u tabeli Kartica.(10 poena)

5.a) Navesti i objasniti osnovne elemente grafa prethođenja transakcija. Dati definiciju kada Ti prethodi Tj.(4 poena)

b) Proveriti da li postoji konflikt-serijabilnost izvršenja S1 skupa transakcija, a posle i S2. Dati obrazloženje.

S1: r2(X), r2(Y), w2(X), r1(Y), w1(Y), r3(X), w3(X), w3(Y)

S2: r2(X), r2(Y), r1(Y), w1(Y), r3(X), w3(X), w3(Y), w2(X)(6 poena)

3. Spoljno spajanje. Objasniti svaku vrstu spoljnog spajanja i OBAVEZNO dati primer.(10 poena)

4. Osnovna struktura XML dokumenta (dati primer). Koje uslove mora da zadovolji dobro-oformljen XML dokument? Šta je validan XML dokument?(11 poena)

5. Vremensko označavanje transakcija.(10 poena)

5. Dat je DTD koji prikazuje učešće studenata na projektima.

```
<!DOCTYPE SP [
  <!ELEMENT SP (Projekat*)>
  <!ELEMENT Projekat (Naslov, Student+)>
  <!ATTLIST Projekat ProjNum ID>
  <!ELEMENT Naslov (#PCDATA)>
  <!ELEMENT Student>
  <!ATTLIST Student BrInd ID Ime CDATA> ]>
```

a) Napraviti UML dijagram klasa ili model objekti-veze kojim se najbolje prikazuje dati DTD. (4 poena)

b) Napisati primer XML dokumenta koji je validan u skladu sa datim DTD-om. (4 poena)

c) Napisati XPath izraz kojim se prikazuju svi projekti u kojima učestvuje bar jedan student sa imenom "Marko". (3 poena)

* predstavlja kardinalnost 0,m

+ predstavlja kardinalnost 1,m

3. Spoljno spajanje. Objasniti svaku vrstu spoljnog spajanja i **OBAVEZNO** dati primer. (10 poena)

4. Osnovna struktura XML dokumenta (dati primer). Koje uslove mora da zadovolji dobro-iformljen XML dokument? Šta je validan XML dokument? (11 poena)

5. Vremensko označavanje transakcija. (10 poena)

3. Posmatrajmo sledeće tri transakcije koje se izvršavaju nad relacijom `studenti(ime, ocena)`.

Transakcija A:

```
INSERT INTO studenti VALUES ('F. Student', 5);
INSERT INTO studenti VALUES ('A. Student', 10);
INSERT INTO studenti VALUES ('C. Student', 8);
```

Transakcija B: `UPDATE studenti SET ocena=ocena+1 WHERE ocena<11 AND ocena>5;`

Transakcija C: `UPDATE studenti SET ocena=5 WHERE ocena=6;`

a) Predpostavimo da se transakcije izvršavaju simultano sa nivoom izolovanosti `READ COMMITTED` i da je tabela `studenti` inicijalno prazna. Prikazati sva rešenja njihovog serijabilnog izvršavanja u slučaju da su sve transakcije potvrđene (tj. nijedna transakcija nije uradila rollback). Rešenja prikazati preko ekstenzije relacije `studenti`. (5 poena)

b) Proveriti da li postoji konflikt-serijabilnost izvršenja skupa transakcija kojem pripadaju transakcije B i C. (5 poena)

4. Objasniti sledeće koncepte:
a) Determinanta; Kandidat za ključ b) Potpuna funkcionalna zavisnost; Tranzitivna zavisnost
c) Uslovi koje treba tabela da zadovolji da bi bila relacija d) Integritet entiteta; Referencijalni integritet

5. Dat je DTD koji prikazuje učešće studenata na projektima.

```
<!DOCTYPE SP [
<!ELEMENT SP (Projekat*)>
<!ELEMENT Projekat (Naslov, Student+)>
<!ATTLIST Projekat ProjNum ID>
<!ELEMENT Naslov (#PCDATA)>
<!ELEMENT Student>
<!ATTLIST Student BrInd ID Ime CDATA> ]>
```

a) Napraviti UML dijagram klasa ili model objekti-veze kojim se najbolje prikazuje dati DTD. (4 poena)

b) Napisati primer XML dokumenta koji je validan u skladu sa datim DTD-om. (4 poena)

c) Napisati XPath izraz kojim se prikazuju svi projekti u kojima učestvuje bar jedan student sa imenom "Marko". (3 poena)

(2007)

3. Objasniti osnove relacionog upitnog jezika QBE (Query by Example). (10 poena)

4. Objasniti i ilustrovati primerom vrste nasledjivanja u ODMG-u? (10 poena)

5. Šta je Transakcija? Objasniti skup osobina koje transakcija mora da poseduje. (10 poena)

6. Za dati XML dokument:

```
<?xml version="1.0"?>
<portfolio xmlns:dt="urn:schemas-microsoft-com:datatypes" xml:space="preserve">
<stock exchange="nyse">
<name>zacx corp</name>
<symbol>ZCXM</symbol>
<price dt:dt="number">28.875</price>
<share dt:dt="number">1000</share>
</stock>
<stock exchange="nasdaq">
<name>zaffymat inc</name>
<symbol>ZFFX</symbol>
<price dt:dt="number">92.250</price>
<share dt:dt="number">1500</share>
</stock>
<stock exchange="nasdaq">
<name>zysmergy inc</name>
<symbol>ZYSZ</symbol>
<price dt:dt="number">20.313</price>
<share dt:dt="number">2000</share>
</stock>
</portfolio>
```

a) Napisati XML shemu posmatrajući dati dokument kao templejt uz sledeće pretpostavke: svi atributi su obavezni; redosled oznaka je bitan; atribut „exchange“ može imati samo vrednosti nyse, nasdaq, i amx.

Odgovor:

```
<xs:attribute name="exchange" use="required">
  <xs:simpleType>
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="nyse"/>
      <xs:enumeration value="nasdaq"/>
      <xs:enumeration value="amx"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
```

b) Uz pretpostavku da se podaci o validnim XML dokumentima skladiste u relacionoj bazi podataka, definisati relacije kojima se omogućava njihovo skladistenje sa najmanje redudansi.

c) Napisati XQuery izraz kojim se prikazuju sva portfolija koja sadrže deonice (stocks) kompanije „zysmergy inc.“.

(15 poena)

3. Objasniti osnove relacionog upitnog jezika QBE (Query by Example). *(10 poena)*

4. Šta je Transakcija? Objasniti skup osobina koje transakcija mora da poseduje. *(10 poena)*

5. Objasniti i ilustrovati primerom vrste nasledjivanja u ODMG-u? *(10 poena)*

3. Poslovna pravila integriteta relacionog modela. Dati za svaki podtip ovih ograničenja primer. (10 poena)
4. Spoljno spajanje. Objasniti svaku vrstu spoljnog spajanja i OBAVEZNO dati primere. (10 poena)
5. a) Navesti i objasniti osnovne elemente grafa prethođenja transakcija. Dati definiciju kada Ti prethodi Tj. (4 poena)
 b) Proveriti da li postoji konflikt-serijabilnost izvršenja S1 skupa transakcija, a posle i S2. Dati obrazloženje.
 S1: r2(A), r2(B), w2(A), r1(B), w1(B), r3(A), w3(A), w3(B)
 S2: r2(A), r2(B), r1(B), w1(B), r3(A), w3(A), w3(B), w2(A) (6 poena)

3. Navesti sve dodatne operacije relacione algebre koje su uvedene zbog postojanja nula vrednosti u bazi podataka i OBAVEZNO dati primere. (10 poena)
4. Date su tabele $R(P,A)$ i $S(F,B)$. Uz pretpostavku da postoji referencijalno ograničenje: $S.F$ je spoljni ključ koji se referencira na primarni $R.P$, napisati trigger na nivou naredbe kojim se implementira dinamičko pravilo integriteta "on update default" i „on delete nullifies“. (10 poena)
5. Semantika i obrada ECA pravila. (10 poena)

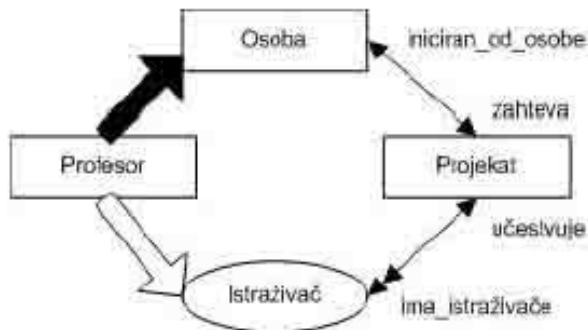
3. Spoljno spajanje. Objasniti svaku vrstu spoljnog spajanja i OBAVEZNO dati primer. (10 poena)
4. Navesti, opisati i dati primer za složene konstruisane tipove u objektno-relacionom bp (SQL-1999 standard). Za svaki tip dati opštu definiciju i način njihovog korišćenja u SQL naredbama. (10 poena)
5. Vremensko označavanje transakcija. (10 poena)

3. Spoljno spajanje. Objasniti svaku vrstu spoljnog spajanja i OBAVEZNO dati primere. (10 poena)
4. Objasniti i ilustrovati primerom vrste nasledjivanja u ODMG-u? (10 poena)
5. Semantika i obrada ECA pravila. (10 poena)

3. Poslovna pravila integriteta relacionog modela. Dati za svaki podtip ovih ograničenja primer. (10 poena)
4. Navesti, opisati i dati primer za složene konstruisane tipove u objektno-relacionom bp (SQL-1999 standard). Za svaki tip dati opštu definiciju i način njihovog korišćenja u SQL naredbama. (10 poena)
5. a) Navesti i objasniti osnovne elemente grafa prethođenja transakcija. Dati definiciju kada Ti prethodi Tj. (4 poena)
 b) Proveriti da li postoji konflikt-serijabilnost izvršenja S1 skupa transakcija, a posle i S2. Dati obrazloženje.
 S1: r2(A), r2(B), w2(A), r1(B), w1(B), r3(A), w3(A), w3(B)
 S2: r2(A), r2(B), r1(B), w1(B), r3(A), w3(A), w3(B), w2(A) (6 poena)

3. Navesti i u po jednoj rečenici objasniti koje su dve osnovne prednosti korišćenja pogleda. (10 poena)

4. Dat je objektni model. Podrazumeva se da svi tipovi imaju atribute ID i Naziv.



a) Dati ODL naredbe za kreiranje sheme objektna baze.

b) Napisati OQL upit kojim se prikazuju svi projekti na kojima učestvuje i profesor koji je inicirao taj projekat.

(10 poena)

5. Vremensko označavanje transakcija.

(10 poena)

Odgovor na 4. pitanje:

```
class Osoba (extent osobe)
{
attribute short id;
attribute string naziv;
relationship Projekat zahteva inverse Projekat::iniciran_od_osobe;
};
class Projekat (extent projekti)
{
attribute short id;
attribute string naziv;
relationship Osoba iniciran_od_osobe inverse Osoba::zahteva;
relationship set<Istrazivac> ima_istrzivace inverse Istrazivac::ucestvuje;
};
interface Istrazivac {
relationship Projekat ucestvuje inverse Projekat::ima_istrzivace;
};
class Profesor extends Osoba : Istrazivac (extent profesori) {};
b)
select x
from projekti x
where x.iniciran_od_osobe in
(select y from profesori y where y.ucestvuje = x);
```

3. Pravila integriteta relacionog modela. Primere ograničenja opisati preko operacija relacione algebre. (10 poena)
4. Navesti, opisati i dati primer za složene konstruisane tipove u objektno-relacionom bp (SQL-1999 standard). Za svaki tip dati opštu definiciju i način njihovog korišćenja u SQL naredbama. (11 poena)
5. a) Navesti i objasniti osnovne elemente grafa prethođenja transakcija. Dati definiciju kada Ti prethodi Tj. (4 poena)
- b) Proveriti da li postoji konflikt-serijabilnost izvršenja S1 skupa transakcija, a posle i S2. Dati obrazloženje.
 S1: r2(X), r2(Y), w2(X), r1(Y), w1(Y), r3(X), w3(X), w3(Y)
 S2: r2(X), r2(Y), r1(Y), w1(Y), r3(X), w3(X), w3(Y), w2(X)
- (6 poena)

(2008)

3. Spoljno spajanje. Objasniti svaku vrstu spoljnog spajanja i OBAVEZNO dati primer. (10 poena)
4. Navesti, opisati i dati primer za složene konstruisane tipove u objektno-relacionom bp (SQL-1999 standard). Za svaki tip dati opštu definiciju i način njihovog korišćenja u SQL naredbama. (10 poena)
5. Prikazati šemu komponenti Sistema za upravljanje bazama podataka i opisati ulogu svake komponente. (10 poena)
3. Navesti sve Konvencionalne skupovne operacije i Specijalne relacione operacije relacione algebre. OBAVEZNO dati primere za svaku vrstu operacije. (10 poena)
4. Objasniti šta je pogled. Koje su osnovne prednosti u korišćenju pogleda. Dati uslove koje treba da ispuni pogled da bi mogao da posluži za ažuriranje baze. (10 poena)
5. Ukratko objasniti vrste korisničkih i konstruisanih tipova. Kreirati objektno-relacioni model kojim se omogućava skladištenje kolekcije studenata sa jednoznačnim atributima BrojIndeksa, ImePrezime, višeznačnim atributom Položenispiti i referentnom kolonom koja ukazuje na NastavniPlan. NastavniPlan ima attribute ŠifraPlana i DatumOd. (10 poena)

Odgovor:

```
CREATE TABLE Studenti (BrojIndeksa CHAR (10), ImePrezime CHAR (30), Polozenilspiti CHAR (30) ARRAY [44], REF IS NastavniPlan DERIVED);
```

```
CREATE TYPE NastavniPlan AS (SifraPlana CHAR (20), DatumOd DATE) NOT FINAL;
```

3. Navesti sve Konvencionalne skupovne operacije i Specijalne relacione operacije relacione algebre. OBAVEZNO dati primere za svaku vrstu operacije. (10 poena)
4. Prikazati šemu komponenti Sistema za upravljanje bazama podataka i opisati ulogu svake komponente. (10 poena)
5. Vremensko označavanje transakcija. (10 poena)
3. Navesti sve Konvencionalne skupovne operacije i Specijalne relacione operacije relacione algebre. OBAVEZNO dati primere za svaku vrstu operacije. (10 poena)
4. Ukratko opisati 4 osnovne komponente svakog modela podataka. (10 poena)

3. Ukratko objasniti vrste korisničkih i konstruisanih tipova. Kreirati objektno-relacioni model kojim se omogućava skladištenje kolekcije studenata sa jednoznačnim atributima BrojIndeksa, ImePrezime, višeznačnim atributom Položenispiti i referentnom kolonom koja ukazuje na NastavniPlan. NastavniPlan ima attribute ŠifraPlana i DatumOd. (10 poena)

4. Šta je Transakcija? Objasniti skup osobina koje transakcija mora da poseduje. (10 poena)

5. Date su tabele T1(A,B) i T2(C,D). Uz pretpostavku da postoji referencijalno ograničenje: T2.C je spoljni ključ koji se referencira na primarni T1.A, napisati triger kojim se implementira dinamičko pravilo integriteta "on insert default" na nivou naredbe i „on delete cascades“ na nivou reda. (10 poena)

Odgovor:

```
CREATE TRIGGER on_delete_cascade
AFTER DELETE ON T1
REFERENCING
    OLD AS StariRed
FOR EACH ROW (jer je na nivou reda)
BEGIN ATOMIC
DELETE from T2
WHERE C = StariRed.A
END;
```

3. Navesti sve dodatne operacije relacione algebre koje su uvedene zbog postojanja nula vrednosti u bazi podataka i OBAVEZNO dati primere. (10 poena)

4. Ukratko opisati 4 osnovne komponente svakog modela podataka. (10 poena)

Iz razloga što je ovo jedan obiman predmet, koji po mom mišljenju, zahteva minimum dva semestra da bi se celokupno gradivo, od početka do kraja, savladalo kako treba,

odlučio sam da skupim sve materijale koji su potrebni da bi se Baze podataka položile sa ocenom većom od 6. Moguće je da će neki materijali vremenom nestati sa foruma, a i mene je lično nerviralo dok sam pretražio sve informacije i objašnjenja koja su razbacana na tri teme,

Tako da se ovde, na jednom mestu, mogu naći svi materijali koje sam pokupio sa teme od pre dve godine, prošlogodišnje i ovogodišnje:

2. BCNF normalizacija

Normalizacija je postupak koji se koristi da bi eliminisali sve nedostatke u održavanju baze podataka, a pre svega redudansu podataka. Normalizacijom se olakšava ažuriranje baze jer npr. ne moramo da za hiljadu radnika, svakog pojedinačno, brišemo neki podatak iz tabele, već će se promena željenih podataka (npr. naziv odeljenja) može uraditi u nekoj drugoj, referenciranoj, tabeli za par sekundi.

Pre nego što počnete sa daljim čitanjem pogledajte [video tutorijal](#) koji je napravio kolega Jessieblueyu, zaslužan za razjašnjenje mnogih stvari.

Jessieblueyu je napisao:

Evo mojih materijala iz proslogodisnje teme o Bazama podataka koji su apsolutno aktuelni i za ovu godinu, pa reko' da ih prebacim i ovde:

(Sve video tutorijale mozete gledati uz pomoc VLC player-a. Hvala Lanee-tu za info.

Evo linka: <http://www.videolan.org/vlc/>)

BCNF urađeni zadaci:

<http://rapidshare.com/files/241618207/BCNF.rar.html>

Postoje dva tipa zadataka kod normalizacije:

- 1) Direktnim korišćenjem BCNF
- 2) Normalizacija sa 3NF

Normalizacija direktnim korišćenjem BCNF se pojavljuje u 99% rokova, ali ko želi da pogleda i sa 3NF ima jedan zadatak koji je objasnio kolega Strag.

Normalizacija direktnim korišćenjem BCNF je prikazana na zadatku iz 17.09.2010

Data je relacija ProjektniTim (ŠifraProjekta, JMBG, NazivProjekta, ImePrezime, PeriodAngažovanja, ŠifraSektora, NazivSektora)

I sledeće funkcionalne zavisnosti:

ŠifraProjekta, JMBG --> PeriodAngažovanja, ŠifraSektora, NazivSektora, NazivProjekta, ImePrezime

ŠifraProjekta --> NazivProjekta

JMBG --> ImePrezime, ŠifraSektora, NazivSektora

ŠifraSektora --> NazivSektora

NazivSektora --> ŠifraSektora

Pre nego što se počne sa rešavanjem zadatka, par napomena i objašnjenja:

-Kod ovog zadatka je potrebno da se pre rešavanja napišu 3 definicije, za kandidat za ključ, determinantu i BCNF:

- **Kandidat za ključ je atribut koji određuje sve druge attribute u određenoj relaciji.**
- **Determinanta je onaj atribut, ili skup atributa, koji u potpunosti funkcionalno određuje drugi atribut.**
- **Relacija je u BCNFu ako i samo su sve determinante ujedno i kandidati za ključ.**

E sad, determinante su oni atributi koji se nalaze sa leve strane funkcionalne zavisnosti (sa leve strane strelice), tako da su to u našem slučaju

- ŠifraProjekta,JMBG
- ŠifraProjekta
- JMBG
- ŠifraSektora
- NazivSektora

Dakle, 5 determinanti. Treba obratiti pažnju da u nekim zadacima može da stoji nešto ovako:

ŠifraProjekta,JMBG --> NazivProjekta,ImePrezime,PeriodAngažovanja

ŠifraProjekta,JMBG --> ŠifraSektora,NazivSektora

To se onda neće posmatrati kao dve funkcionalne zavisnosti već se spaja u jednu FZ.

Po gore navedenoj definiciji kandidat za ključ je (složeni) atribut $KK=\{\text{ŠifraProjekta},\text{JMBG}\}$ jer određuje sve ostale attribute u početnoj relaciji ProjektniTim, a to se vidi iz funkcionalne zavisnosti 1.

Na osnovu funkcionalne zavisnosti 2 imamo:

Projekat(ŠifraProjekta,NazivProjekta)

ProjektniTim0(ŠifraProjekta,JMBG,ImePrezime,PeriodAngažovanja,ŠifraSektora,NazivSektora).

Početnu relaciju, ProjektniTim, smo dekomponovali na ProjektniTim0 i Projekat. Iz relacije ProjektniTim smo obrisali NazivProjekta, a ŠifruProjekta je bitno ostaviti jer je to deo ključa relacije ProjektniTim pa je samim tim ne treba brisati, a i preko nje će ta relacija biti povezana sa novonastalom relacijom Projekat.

Dalje posmatramo da li su novonastale relacije u BCNF:

Relacija Projekat jeste u BCNF jer od svih determinanti koje imamo za tu relaciju važi samo FZ2, a kandidat za ključ je jedan jedini (ŠifraProjekta) tako da imamo situaciju 1D 1KK pa posmatrana relacija jeste u BCNF:

Projekat (ŠifraProjekta,NazivProjekta) BCNF

Druga relacija koju smo dobili je ProjektniTim0 i ona nije u BCNF zato što imamo da važe FZ1 FZ3 FZ4 FZ5 (da bi neka funkcionalna zavisnost važila, potrebno je da se atributi na levoj strani funkcionalne zavisnosti nalaze i u posmatranoj relaciji, dok atributi sa desne strane FZ ne moraju svi da budu prisutni u datoj relaciji) a samo 1 KK pa data relacija nije u BCNF pa se dalje dekomponuje.

Zadatak se radi do kraja na potpuno isti način – atributi sa leve strane određene funkcionalne zavisnosti postaju primarni ključevi u novonastalim reakcijama dok atributi sa desne strane iste te funkcionalne zavisnosti postaju neključni atributi.

Na osnovu funkcionalne zavisnosti 3 imamo:

Projektant(JMBG,ImePrezime,ŠifraSektora,NazivSektora)Nije BCNF: FZ3,4,5 i 1KK

ProjektniTim1 (ŠifraProjekta, JMBG, PeriodAngažovanja) FZ1 i 1KK => BCNF

Na osnovu funkcionalne zavisnosti 4 imamo:

Sektor(ŠifraSektora,NazivSektora)

Ovo gore je malo poseban slučaj, imamo dve determinante FZ4,FZ5 i dva kandidata za ključ ŠifraSektora i NazivSektora jer nam te dve determinante govore da bilo koji od ova dva atributa može biti ključ pa se pravi relacija po želji:

Sektor (ŠifraSektora, NazivSektora) BCNF

ili

Sektor (NazivSektora, ŠifraSektora) BCNF

Projektant1 (JMBG, ImePrezime, ŠifraSektora) FZ3 i 1KK => jeste BCNF

To je to.

Drugi oblik zadatka sa 3NF koga zanima:

Strag je napisao:

Definicije:

-Relacija je u prvoj normalnoj formi (1NF) ako sve vrednosti njenih atributa atomske.

-Relacija je u 2NF ako je u 1NF i ako svi njeni neključni atributi potpuno funkcionalno zavise od primarnog ključa (koji može biti prost ili složen)

-Relacija je u 3NF ako je u 2NF i ako su svi njeni neključni atributi netranzitivno zavisni od primarnog ključa te relacije.

+Iteracija 1:

PlasmanFormula1_1([u]VozacID,TImID,Sezona,RbPlasmana[/u],ImeVozaca,NazivTima,Rang,StazaID,NazivStaze,)

+Diskusija:

Za kandidat za ključ uzimam onaj par atributa koji od kojih svi neključni atributi početne relacije funkcionalno zavise (za sada nebitno o kojoj funkcionalnoj zavisnosti se radi).

Dakle kandidat za ključ će biti $KK=\{VozacID,TImID,Sezona,RbPlasmana\}$, na osnovu Fz1 formiram

PlasmanFormula1_1 relaciju.

Ispitujemo da li je početna relacija u 1NF, kako su sve vrednosti atributa atomske, tj nema tabele u tabeli zaključujemo da je relacija u 1NF

Ispitujemo da li je početna relacija u 2NF, zaključujemo da nije jer postoje nepotpune funkcionalne zavisnosti, dakle potrebna je dekompozicija.

+Iteracija 2:

Vozac([u]VozalD[/u],ImeVozaca)

Tim([u]TimID[/u],ImeTima)

PlasmanFormula1_2([u]VozacID,TImID,Sezona,RbPlasmana[/u],Rang,StazaID,NazivStaze)

+Diskusija:

Na osnovu Fz2 a radi dekompozicije relacije PlasmanFormula1_1, dobijam relaciju Vozac, čiji je primarni ključ VozacID. Data relacija je u 1NF i 2NF. Isto važi i za relaciju Tim dobijenu na osnovu relacije Fz3.

Relacija PlasmanFormula1_2 je dekomponovana relacija relacije PlasmanFormula1_1. Novo nastala relacija u sebi ne sadrži attribute ImeVozaca i ImeTima da bi se otklonila nepotpuna funkcionalna zavisnost. Relacija PlasmanFormula1_2 je u 1NF i 2NF. Relacija nije u 3NF jer postoji tranzitivna funkcionalna zavisnost.

Dekompozicija se nastavlja. +Iteracija 3:

Vozac([u]VozalD[/u],ImeVozaca)

Tim([u]TimID[/u],ImeTima)

Staza([u]StazaID[/u],NazivStaze)

PlasmanFormula1_3([u]VozacID,TImID,Sezona,RbPlasmana[/u],Rang,StazaID)

+Diskusija:

Relacije Vozac, Tim i Staza (koja je dobijena radi dalje dekompozicije relacije PlasmanFormula1_2, a na osnovu Fz4) su u 1NF, 2NF i 3NF

Relacija PlasmanFormula1_3 je dekomponovana relacija relacije PlasmanFormula1_2. Atribut NazivStaze nije sadržan u novonastaloj relaciji sa ciljem da se poništi tranzitivna zavisnost, da se podaci nebi izgubili pravi se nova relacija Staza. Data relacija je sada i u 1NF, 2NF i 3NF.

Sve relacije su u 3NF

+Pregled:

Vozac([u]VozalD[/u],ImeVozaca)

Tim([u]TimID[/u],ImeTima)

Staza([u]StazaID[/u],NazivStaze)

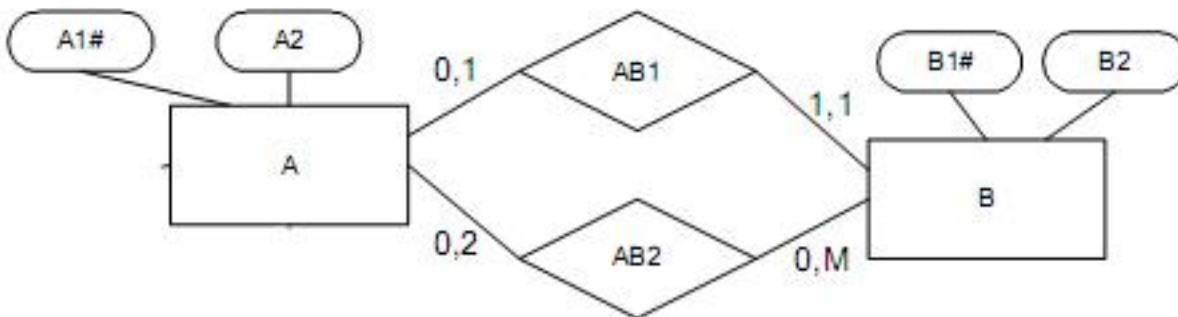
PlasmanFormula1_3([u]VozacID,TImID,Sezona,RbPlasmana[/u],Rang,StazaID)

3. PREVOĐENJE PMOV U RELACIONI

Drugi zadatak je lepo objašnjen u video tutorijalu koji je napravio već pomenuti kolega Jessieblueyu i može se naći na ovoj [adresi](#).

Međutim postoji par detalja koje bi trebalo napomenuti:

Kada imamo vezu koja sa obe strane ima gornju granicu kardinalnosti $GG=M$ to se rešava na dva načina, kao što je pokazano u tutorijalu. Mislim da je najbolji način da se takva veza radi tako što se pretvara u novu relaciju jer su šanse za grešku tada manje. Dakle samo u slučaju kada je kardinalnost sa obe strane M pravi se nova relacija i radi se tako što ta novonastala relacija preuzima ključeve objekata. U ovom slučaju sa slike imaćemo novu relaciju $AB2$ ($A1\#,B1\#$). Po mom mišljenju ovo je najjednostavniji način rada. Nove relacije uvodite samo u tim slučajevima i nećete moći da se zbunite.



Ostale sitnice koje treba zapamtiti kod ovog zadatka:

Svaka veza koja za GG ima neki broj koji je veći od 1 posmatra se u stvari kao $GG=M$

Objekat sa čije strane je kardinalnost 1,1 preuzima ključ od drugog objekta sa kojim gradi vezu (u slučaju sa gornje slike $B(B1\#,B2,A1\#)$)

Kod agregacija teoretski postoje dve opcije, kao što piše u knjizi na 425. strani pod d:

Primarni ključ agregacije postaje:

ključ objekta koji prema agregaciji gradi vezu sa kardinalnošću $GG=1$ dok ostali objekti koji grade vezu sa $GG=M$ samo predaju svoje primarne ključeve koji će u toj agregaciji biti neključni atributi,

ili

složeni atribut sačinjen od identifikatora objekata koji grade vezu ka agregaciji sa $GG=M$.

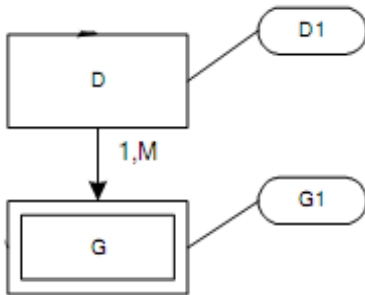
Međutim, u rešenim rokovima nisam naleteo na drugi slučaj tako da agregaciju treba raditi samo tako da joj primarni ključ bude od objekta koji ima vezu $GG=1$.

ODL naredbe

-Specijalizacija se pravi tako što se u objektu , npr B, koji je specijalizacija nekog objekta, npr. A, samo napiše `Object B extends A (extent Bs)` i samim tim se podrazumeva da postoji veza specijalizacije između ova dva objekta koju dakle nije potrebno posebno definisati sa `relationship A BA ...itd.`

-Ključevi su opcioni u ODLu tako da ih nije neophodno eksplicitno navesti ukoliko to ne želite, pa tako kod agregacije i slabih objekata dovoljno je samo da se napiše ime klase (npr. `Class G`) pa se odmah zatim deklariraju atributi i veze sa drugim objektima. To je ono što je možda ostalo nerazjašnjeno u tutorijalu, ali nije neka velika stvar..

Veza jakog i slabog objekta se može prikazati na dva(I i II) načina



Class D

```
{
...

```

```
Attribute set<G> slab_objekat; (ako se ovako definiše slab objekat, u objektu G ne mora ništa više da se piše što se tiče ove veze)
Relationship set<G> D_G inverse G::G_D; }
```

Class G

```
{
...

```

```
Relationship D G_D inverse D::D_G;
}
```

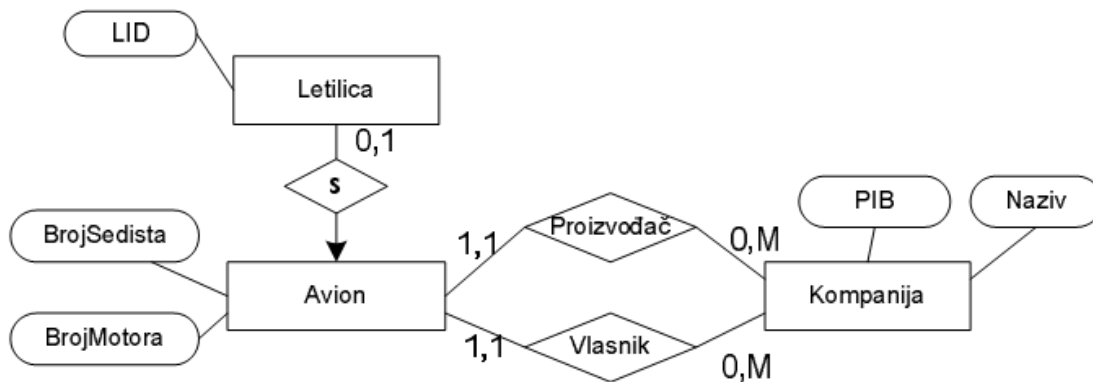
Primer urađenog zadatka [24.09.2007](#) koji je uradio Slađan Babarogić, dovoljan je da se shvati većina stvari iz ove oblasti.

4. XML

Postoje tri tipa zadataka kod XMLa:

1) *Napraviti XML model na osnovu PMOVa (90% rokova)*

Za dati model objekti-veze:



Napisati XML model zapisan pomoću XML Sheme, kojim se najbolje odslikava dati PMOV.

Napisati primer validnog XML dokumenta koji sadrži sve elemente date u XML shemi.

Napisati XPath ili XQuery kojim se prikazuju svi avioni koji su u vlasništvu kompanije koja ih je i

2) *Napraviti XML model na osnovu Relacionog modela*

Za dati relacioni model:

Dobavljač (SifraDob, NazivDob, AdresaDob)

Narudžbenica (BrojNarDob, SifraDob)

StavkaNar (BrojNarDob, RedniBroj, Proizvod, Kolicina, Vrednost, Valuta)

- napraviti XML šemu
- dati primer ekstenzije relacija datog relacionog modela i primer XML dokumenta na osnovu napravljene XML šeme
- napisati XQuery kojim se prikazuju sve narudžbenice kod kojima se nalaze naručeni i proizvod "A" i proizvod "B" od dobavljača "XYZ"

Ovaj tip zadatka se najlaše rešava tako što se relacioni model prevede u PMOV, a to se onda svodi na prvi tip zadatka

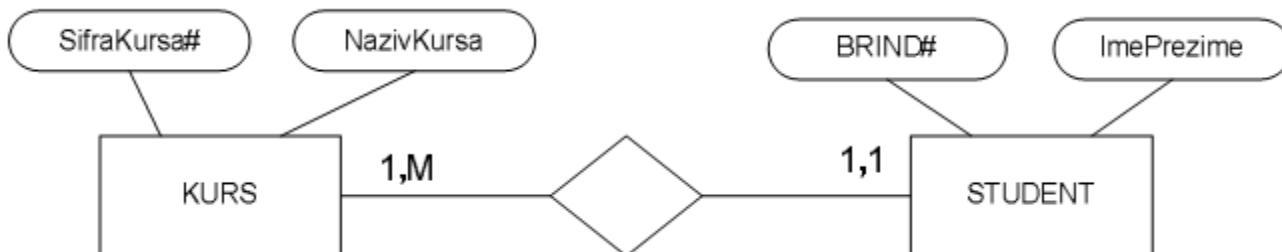
3) Na osnovu template-a

Za dati XML dokument:

```
<?xml version="1.0"?>
<portfolio xmlns:dt="urn:schemas-microsoft-com:datatypes" xml:space="preserve">
  <stock exchange="nyse">
    <name>zack corp</name>
    <symbol>ZCXM</symbol>
    <price dt:dt="number">28.875</price>
    <share dt:dt="number">1000</share>
  </stock>
  <stock exchange="nasdaq">
    <name>zaffymat inc</name>
    <symbol>ZFFX</symbol>
    <price dt:dt="number">92.250</price>
    <share dt:dt="number">1500</share>
  </stock>
  <stock exchange="nasdaq">
    <name>zysmergy inc</name>
    <symbol>ZYSZ</symbol>
    <price dt:dt="number">20.313</price>
    <share dt:dt="number">2000</share>
  </stock>
</portfolio>
```

- Napisati XML schemu posmatrajući dati dokument kao templejt uz sledeće pretpostavke: svi atributi su obavezni; redosled oznaka je bitan; atribut „exchange“ moze imati samo vrednosti nyse, nasdaq, i amx.
- Uz pretpostavku da se podaci o validnim XML dokumentima skladiste u relacionoj bazi podataka, definisati relacije kojima se omogucava njihovo skladistenje sa najmanje redudansi.
- Napisati XQuery izraz kojim se prikazuju sva portfolija koja sadrže deonice (stocks) kompanije „zysmergy inc.“.

Na vežbama se obrađuje samo prvi tip zadatka pa u skladu sa tim najčešće dolazi na ispitnim rokovima.



Takav tip zadatka može da se radi na dva načina prikazana na gornjem primeru:

1) Kako se navode elementi tako se i definiše njihova složenost (complexType)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs=http://...>
  <xs:element name="Model">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="Kurs">
          <xs:complexType //sad sledi definicija elementa Kurs
          <xs:sequence>
            <xs:element name="NazivKursa" type="xs:string"/>
          </xs:sequence>
          <xs:attribute name="SifraKursa" type="xs:ID"
use="required"/>
        </xs:complexType>
      </xs:element>

      <xs:element name="Student">
        <xs:complexType //definicija elementa Student
        <xs:sequence>
          <xs:element name="ImePrezime" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="BrInd" type="xs:string" use="required"/>

        <xs:attribute name="SifraKursa" type="xs:IDREF" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:choice>
</xs:complexType>
</xs:element>
</xs:schema>
```

2) Prvo se navedu svi elementi u modelu i nazivi njihovih kompleksnih tipova (complexType) a zatim se definišu svi ti tipovi

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs=http://...>
  <xs:element name="Model">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="Kurs" type="KursType"/>
        <xs:element name="Student" type="StudentType"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="KursType">
    <xs:sequence>
      <xs:element name="NazivKursa" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="SifraKursa" type="xs:ID" use="required"/>
  </xs:complexType>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ImePrezime" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="BrInd" type="xs:string" use="required"/>
    <xs:attribute name="SifraKursa" type="xs:IDREF" use="required"/>
  </xs:complexType>
</xs:schema>
```

Zadaci sa specijalizacijom se moraju raditi na drugi način.

U gornjem zadatku se može videti da se SAMO primarni ključ navodi kao atribut. Obavezno pored toga mora da stoji use="required".

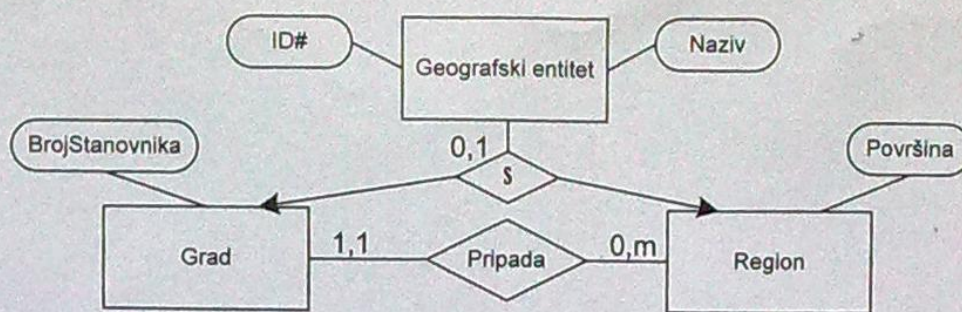
Veza 1,1 – 0/1,M se prikazuje u XML modelu na taj način što se u okviru elementa sa čije strane je kardinalnost 1,1 dodaje atribut sa svojstvom IDREF (u gornjem primeru Student dobija atribut,tj. Primarni ključ od Kursa, a to je ŠifraKursa.

Potrebno je na još neke detalje obratiti pažnju:

Kada se predstavlja veza JAK-SLAB objekat, ni kod jednog ni kod drugog se ne stavlja type="xs:ID" već type="xs:int" (ili odgovarajući) i uz to obavezno use="required".

Kod specijalizacije se samo u objektu koji se specijalizuje stavlja type="xs:ID" dok u objektima koji predstavljaju proširenje specijalizovanog objekta se to ne stavlja.

Zadatak sa specijalizacijom iz septembra 2010:



- Napisati XML model zapisan pomoću XML šeme, kojim se najbolje odlikava dati PMOV.
- Napisati primer validnog XML dokumenta koji sadrži sve elemente date u XML šemi.
- Napisati XQuery kojim se prikazuju svi gradovi sa više od 50000 stanovnika koji pripadaju regionu Šumadije.

uzas je napisao:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">

  <xs:element name="model">
    <xs:complexType>
      <xs:sequence maxOccurs="2">
        <xs:element name="Grad" type="GradTip" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="Region" type="RegionTip" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="GeografskiEntitet">
    <xs:attribute name="ID" type="xs:ID" use="required"/>
    <xs:attribute name="Naziv" type="xs:string"/>
  </xs:complexType>

  <xs:complexType name="GradTip">
    <xs:complexContent>
      <xs:extension base="GeografskiEntitet">
        <xs:attribute name="BrojStanovnika" type="xs:int"/>
        <xs:attribute name="Pripada" type="xs:IDREF" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="RegionTip">
    <xs:complexContent>
      <xs:extension base="GeografskiEntitet">
        <xs:attribute name="Povrsina" type="xs:int"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

5. TRANSAKCIJE

Jessieblueyu je napisao:

1) Nacrtate onoliko kruzica koliko ima transakcija i obelezite ih (1,2,3,...)

2) Kada transakcija 1 vrsi komandu upisivanja ($W1(X)$) u atribut iz koga je neka druga (npr. transakcija 2) VEC citala ($R2(X)$), povlacite liniju usmerenu od 2 ka 1. *Ovo cete najvise koristiti i uglavnom je dovoljno, a ako nije... idemo dalje na 3)!

3) Kada transakcija 1 vrsi komandu upisivanja ($W1(X)$) u atribut u koji je neka druga (npr. transakcija 2) VEC upisala nesto ($W2(X)$), povlacite liniju usmerenu od 2 ka 1. *ako i ovo nije dovoljno, poslednji korak je:

4) Kada transakcija 1 vrsi komandu citanja ($R1(X)$) iz atributa u koji je neka druga (npr. transakcija 2) VEC upisala nesto ($W2(X)$), povlacite liniju usmerenu od 2 ka 1.

Kad na grafu nema ciklusa, izvršenje je konflikt serijabilno u suprotnom nije.

6. SQL

Kod SQL-a nema puno priče, potrebno je uraditi mnogo zadataka da bi se on savladao i potrebno je dobro navići mozak na 'SQL način razmišljanja' da bi tačno rešili te zadatke jer i zahtev koji se sastoji od samo par reči može biti izuzetno komplikovan.

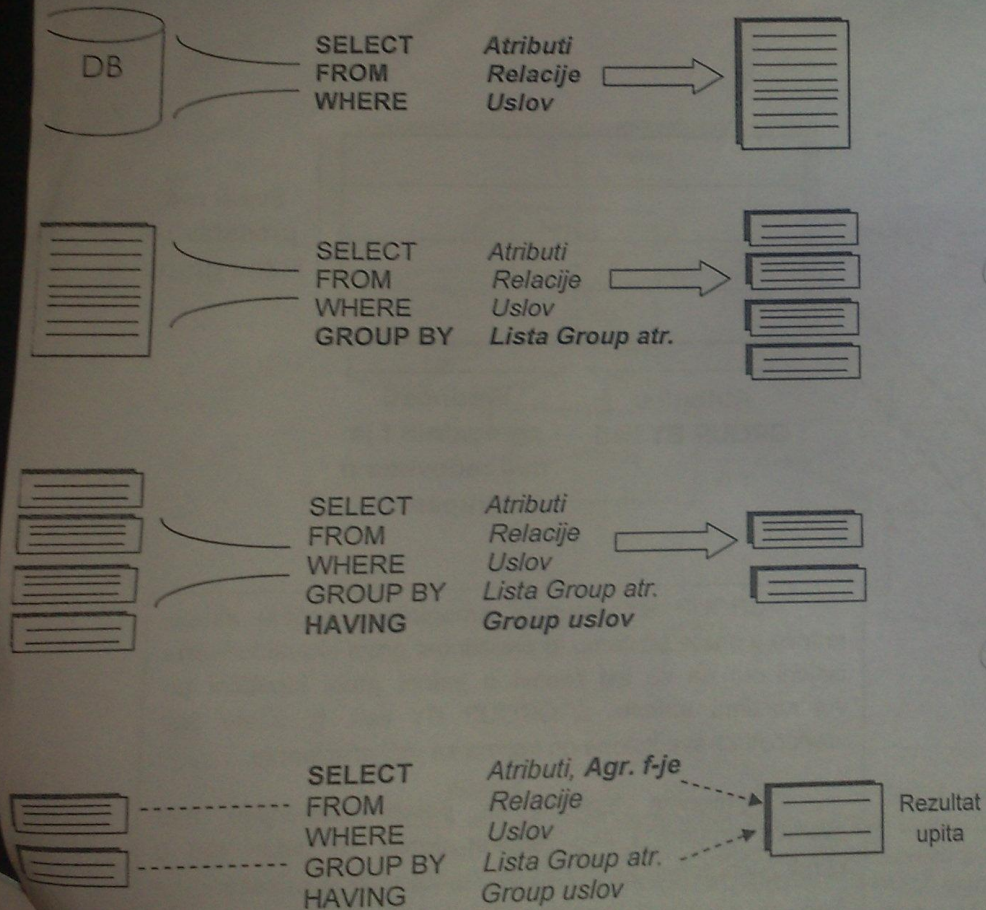
SQL možete vežbati u programu [Oracle Database 10g Express Edition](#).

Treba znati da:

- WHERE može da stoji u SELECT (kao podupit) i tada ključna reč WHERE definiše uslove koji redovi tabele moraju da zadovolje da bi se pojavili u rezultatu. Agregatne funkcije (SUM,COUNT,AVG...) ne mogu da stoje u WHERE.
- Primena agregatnih funkcija zahteva da redovi tabele, na koju se agregatna funkcija primenjuje, budu **grupisani** na neki način. Svaka agregatna funkcija generiše samo PO JEDAN red za svaku grupu redova tabele (npr. radnik sa najvećom platom iz odeljenja sa šifrom 110, radnik sa najvećom platom iz odeljenja sa šifrom 120, radnik sa najvećom platom iz odeljenja sa šifrom 130 itd. ako je naredba grupisana po šiframa odeljenja). Dakle, GROUP BY daje jedan red za svaku različitu vrednost kolone po kojoj se vrši grupisanje.
- Kada se koristi count(*) bez GROUP BY onda se u rezultatu pojavljuje broj redova koji ispunjava WHERE uslov.
Kada count(*) stoji sa GROUP BY onda prebrojava koliko ima redova sa istim atributima koji su definisani u okviru GROUP BY klauzule.
- WHERE i GROUP BY mogu da stoje zajedno, a sintaksa je takva da WHERE ide pre GROUP BY. U tom slučaju se sa WHERE selektuju prvo pojedinačni redovi pa se onda grupišu.
- HAVING određuje kriterijume za selekciju grupa koje su određene prethodno group by uslovom.
- Ako ne postoje agregatne funkcije tabela neće moći da se grupiše.
- Ako imate nedoumicu koje kolone treba staviti u GROUP BY postoji pravilo da sve KOLONE koje nisu agregatne funkcije moraju da stoje u okviru GROUP BY klauzule.

SQL objašnjenja predmetnih asistenata:

Evaluacija upita sa agregatnim funkcijama



Korak 1. Evaluira se FROM klauzula. Kao rezultat dobija se Dekartov proizvod tabela navedenih u njoj.

Korak 2. Evaluira se WHERE klauzula. Svaki red tabele dobijene u koraku 1 se pojedinačno procesuiru i evaluira u odnosu na definisane uslove. Tabela dobijena iz WHERE klauzule sadrži samo one redove kod kojih su navedeni uslovi evaluirani u true. Korak 1 i korak 2 su prikazani u prvom delu slike.

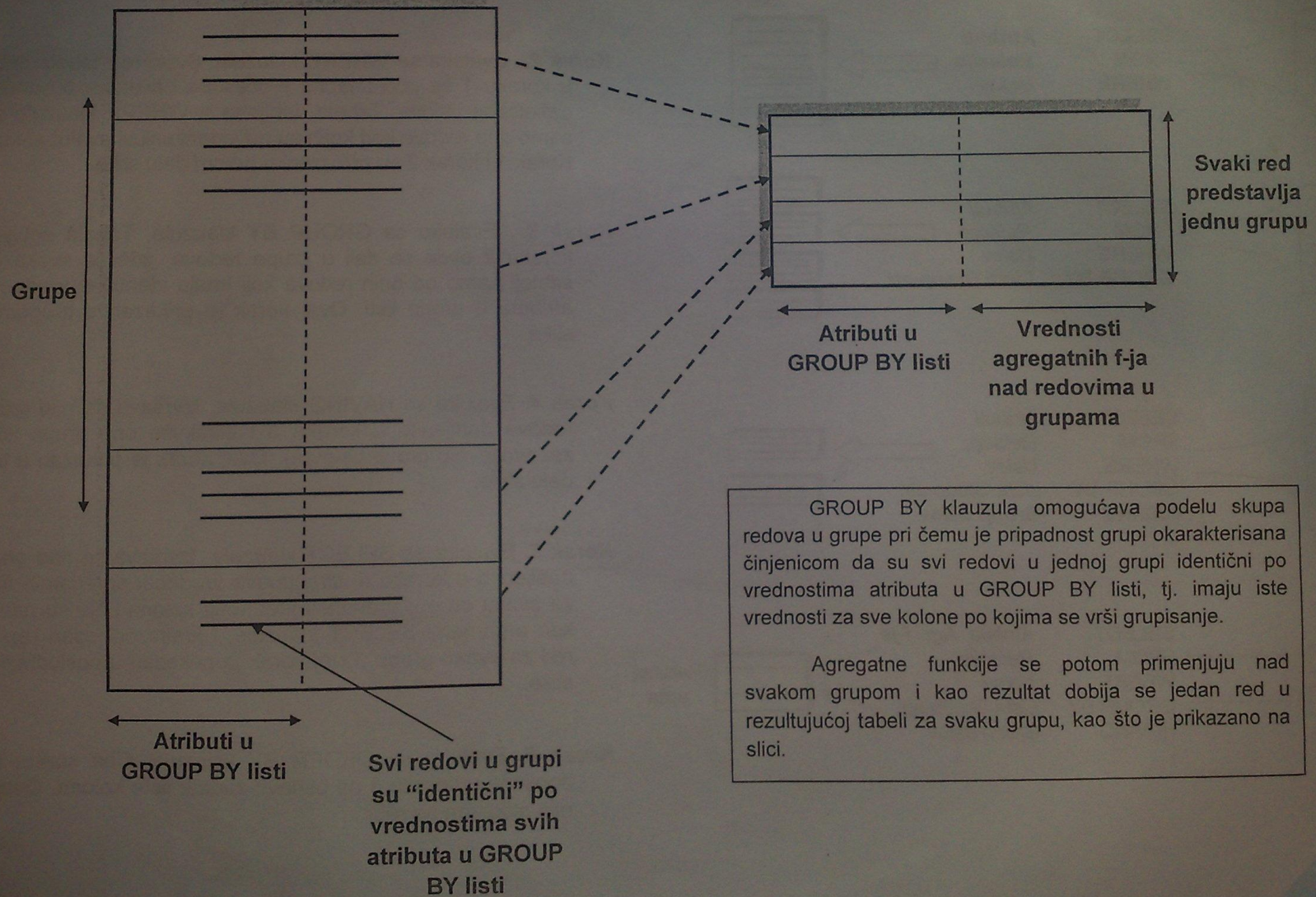
Korak 3. Evaluira se GROUP BY klauzula. Tabela dobijena u koraku 2 ovde se deli u grupe redova, gde se svaka grupa sastoji samo od onih redova koji imaju identične vrednosti za attribute u group listi. Ovaj korak je prikazan u drugom delu slike.

Korak 4. Evaluira se HAVING klauzula. Izvršava se nad grupama redova dobijenim u koraku 3 i eliminiše one grupe koje ne zadovoljavaju group uslov(e). Ovaj korak je prikazan u trećem delu slike.

Korak 5. Evaluira se SELECT klauzula. Izvršava se nad grupama dobijenim u koraku 4, izračunava vrednost agregatnih funkcija za svaku grupu, pronalazi vrednosti kolona koje su navedene kao argumenti SELECT klauzule, i kreira po jedan rezultujući red za svaku grupu. Ovaj korak je prikazan u poslednjem delu slike.

Korak 6. Evaluira se ORDER BY klauzula. Ona sortira redove dobijene u koraku 5 na osnovu zadate liste kolona. Ovaj korak nije prikazan na slici.

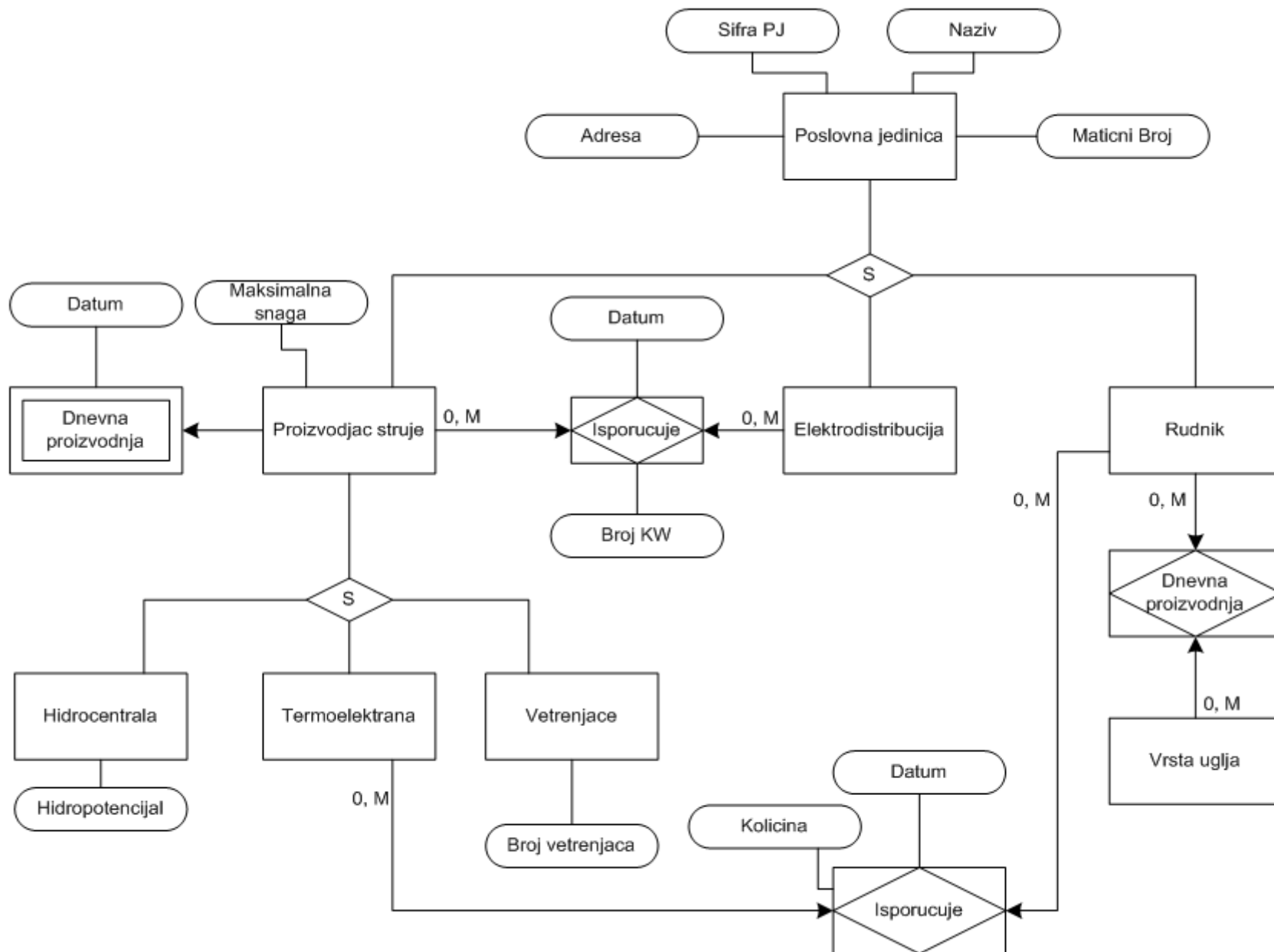
GROUP BY



7. PMOV

8. Napraviti model objekti-veze za deo informacionog sistema Elektro-privrednog sistema. Potrebno je voditi evidenciju o poslovnim jedinicama EPS-a sa sledećim osnovnim atributima: ŠifraPJ, Naziv, Adresa, MatičniBroj. Poslovna jedinica može biti: proizvođač struje ili elektrodistribucija ili rudnik. Proizvođač struje se može klasifikovati na hidroelektrane, termoelektrane i sisteme vetrenjača. Proizvođač struje ima osnovno obeležje maksimalnu snagu, a pored toga potrebno je beležiti i dnevnu proizvodnju u KW (kilovatima). Za hidrocentrale potrebno je pamtit i hidropotencijal, za sistem vetrenjača broj vetrenjača, a za termoelektranu dnevnu isporuku uglja iz svakog rudnika. Za svaku elektrodistribuciju treba voditi evidenciju o dnevnoj isporuci struje u KW (kilovatima) od strane proizvođača struje. Za svaki rudnik obezbediti evidentiranje dnevne proizvodnje uglja po vrstama uglja.

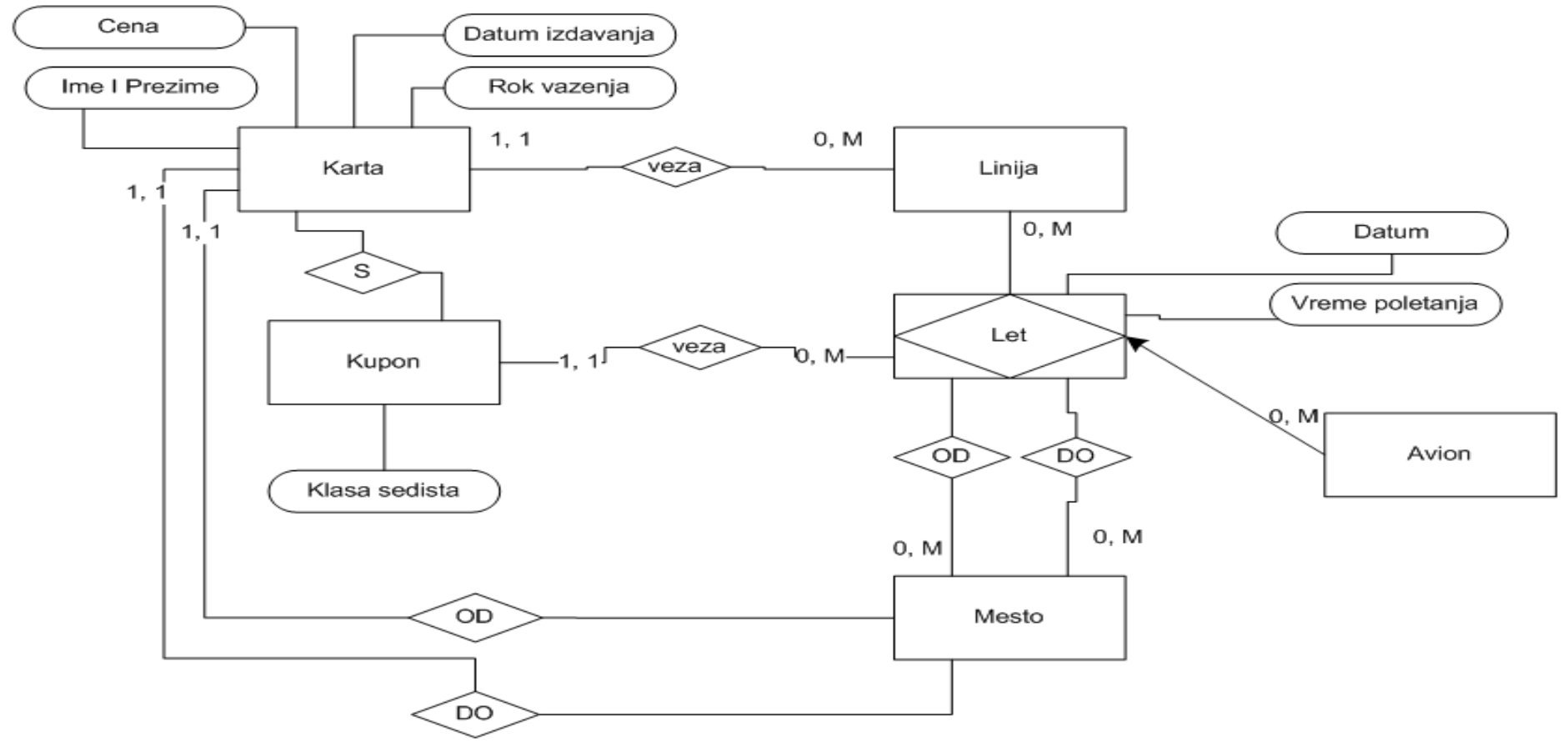
(10 poena)



8. Nacrtati dijagram objekti-veze za deo sistema praćenja avionskog saobraćaja opisan sledećim tekstom:
 „Avionska karta za jednu standardnu avio-liniju može biti sastavljena od više kupona. Jedna linija može da uključi više letova na relaciji izmedju mesta polaska i mesta krajnjeg odredišta. Svaki avion obično ima nekoliko letova u toku dana (let je identifikovan preko datuma i vremena poletanja aviona). Karta sadrži podatke o avionskoj liniji, prezimenu i imenu putnika, mestu polazišta, mestu krajnjeg odredišta, datumu izdavanja, roku važenja i ceni. Kuponi karte sadrže identične podatke kao i karta, kao i podatke o pojedinačnim letovima izmedju polazišta i odredišta: mesto poletanja, mesto sletanja, broj leta, klasa sedišta, datum i vreme poletanja”.

(11 poena)

Za ovaj zadatak je inače Slađan Babarogić rekao da je konfuzan pa da takav ne dolazi na kolokvijumu i ispitu.

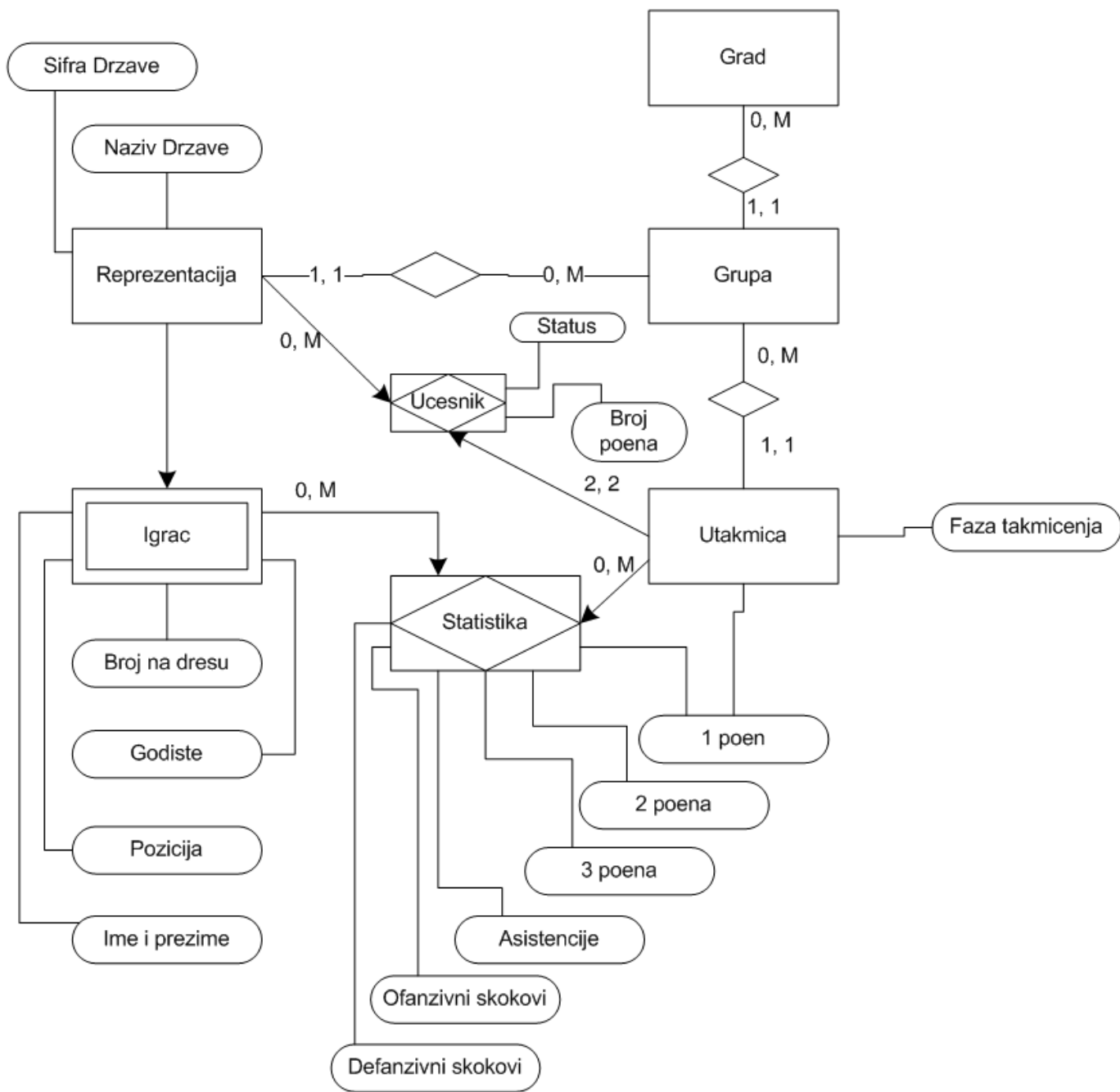


8. Napraviti model objekti-veze za deo informacionog sistema Evropskog prvenstva u košarci. Potrebno je voditi evidenciju o reprezentacijama učesnicama takmičenja sa osnovnim atributima ŠifraDržave, NazivDržave, kao i o njenim igračima sa sledećim osnovnim atributima: BrojNaDresu, ImePrezime, Pozicija, Godište. Svaka reprezentacija je raspoređena u jednu i samo jednu kvalifikacionu grupu. Sve utakmice jedne kvalifikacione grupe se igraju u samo jednom gradu. Potrebno je modelovati utakmice, gde se tačno zna koja je reprezentacija u ulozi domaćina a koja u ulozi gosta. Za svaku utakmicu evidentirati broj postignutih poena domaće i gostujuće reprezentacije. Za svaku utakmicu je potrebno znati i u kojoj fazi takmičenja se odigrala (kvalifikacije u grupi, baraž, četvrtfinale, polufinale, finale ...). Za svakog igrača je potrebno voditi statistiku učinka na svakoj odigranoj utakmici, gde učinak podrazumeva evidenciju broja pogodaka za 1 poen, broja pogodaka za 2 poena, broja pogodaka za 3 poena, broja ofanzivnih skokova, broja defanzivnih skokova i broja asistencija.

(11 poena)

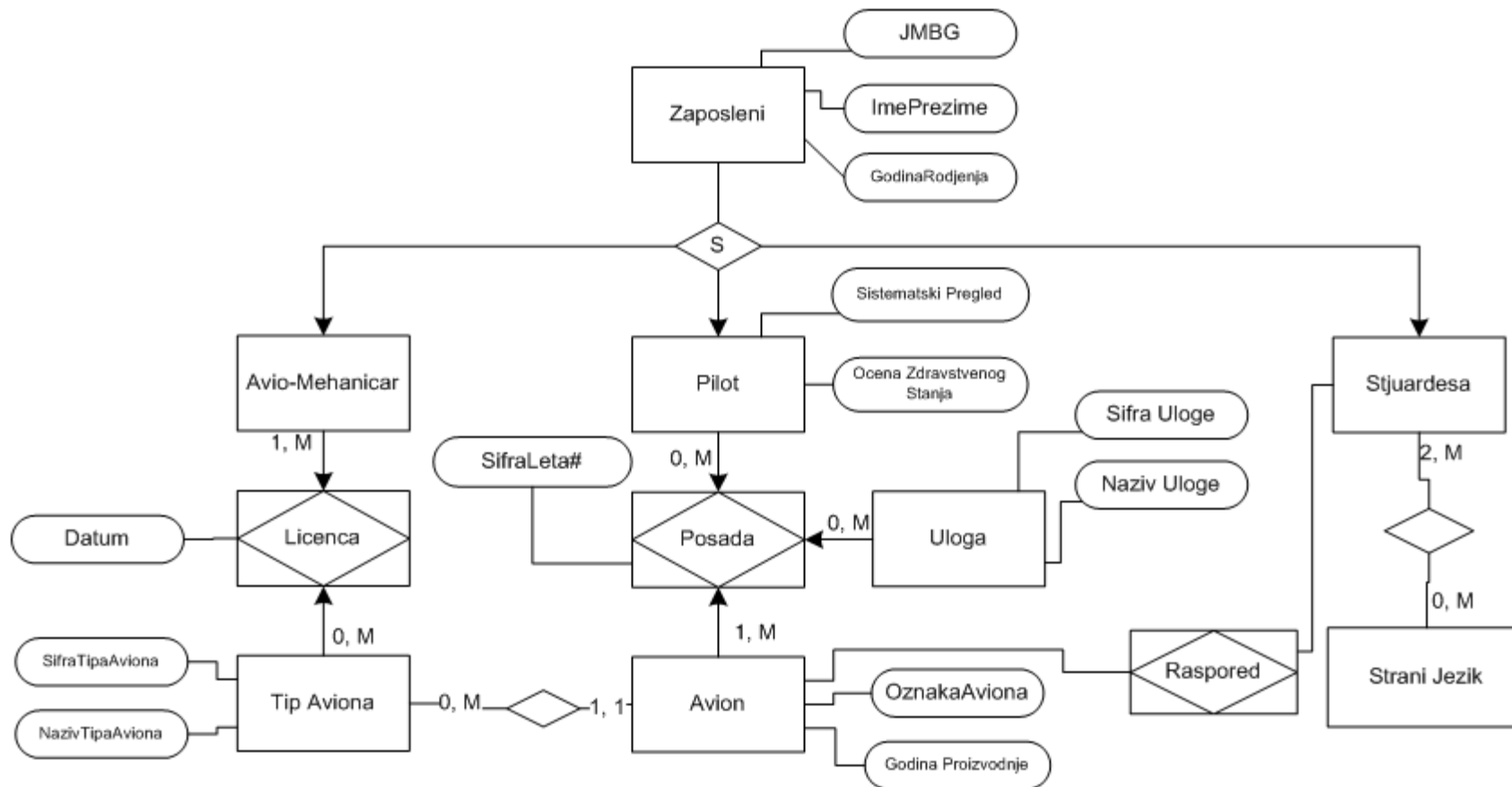
8. Napraviti model objekti-veze za deo informacionog sistema Svetskog prvenstva u fudbalu. Potrebno je voditi evidenciju o reprezentacijama učesnicama takmičenja sa osnovnim atributima ŠifraDržave, NazivDržave, kao i o njenim igračima sa sledećim osnovnim atributima: BrojNaDresu, ImePrezime, Pozicija, Godište. Svaka reprezentacija je raspoređena u jednu i samo jednu kvalifikacionu grupu. Omogućiti beleženje grada u kojem se igrala utakmica. Potrebno je modelovati utakmice, gde se tačno zna koja je reprezentacija u ulozi domaćina a koja u ulozi gosta. Za svaku utakmicu evidentirati broj postignutih golova domaće i gostujuće reprezentacije. Za svaku utakmicu je potrebno znati i u kojoj fazi takmičenja se odigrala (kvalifikacije u grupi, četvrtfinale, polufinale, finale). Za svakog igrača je potrebno voditi statistiku učinka na svakoj odigranoj utakmici, gde učinak podrazumeva evidenciju broja golova, broja udaraca u okvir gola, kao i da li je dobio žuti ili crveni karton.

(11 poena)



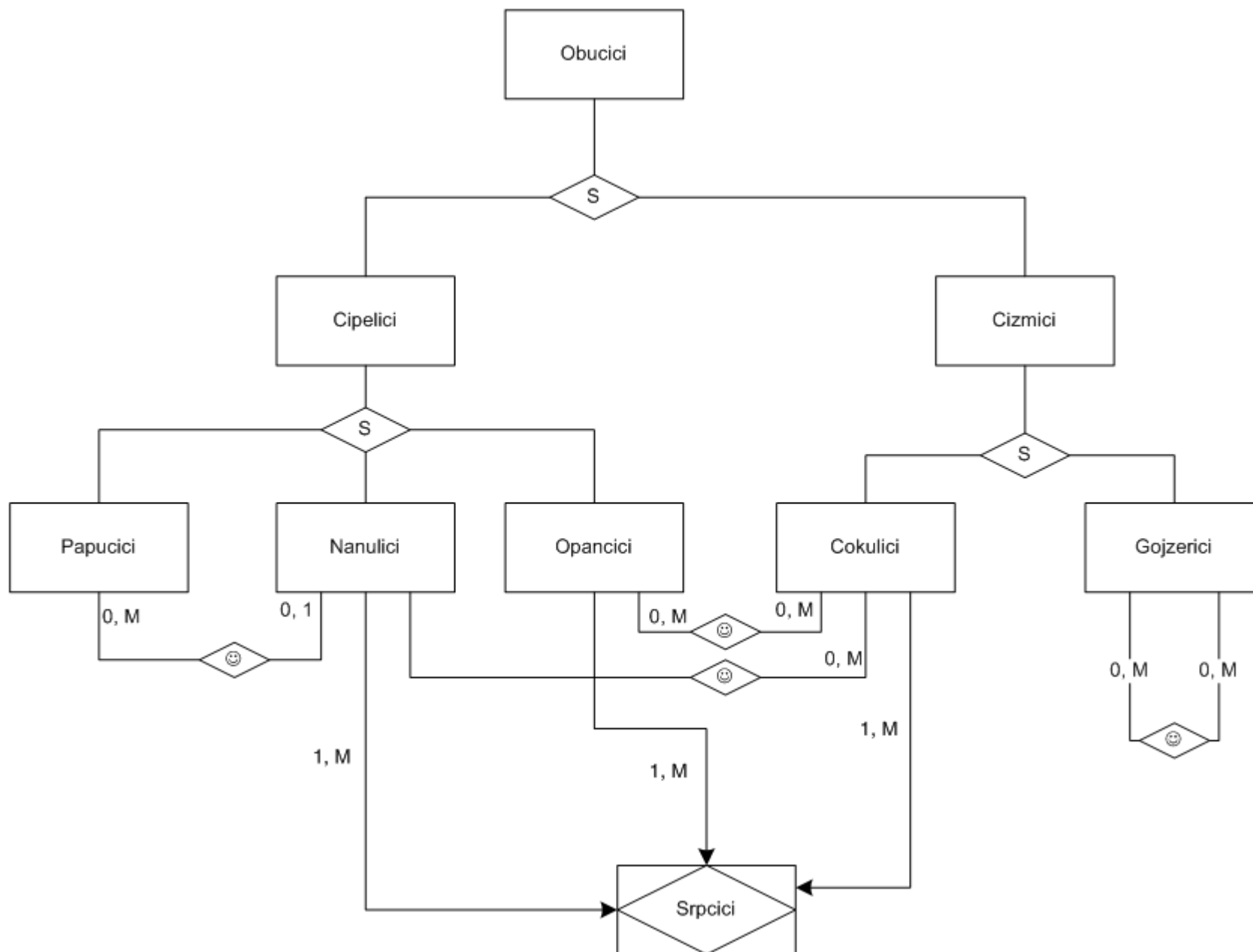
8. Nacrtati dijagram objekti-veze za deo kadrovske evidencije u vazduhoplovnoj kompaniji. Potrebno je voditi evidenciju o zaposlenima sa sledećim zajedničkim atributima: JMBG, ImePrezime, GodinaRođenja. Svaki zaposleni je ili Avio-mehaničar ili Pilot ili Stjuardesa(Stjuard). Za svakog avio-mehaničara potrebno je znati datum dobijanja licence za određeni tip aviona (ŠifraTipaAviona, NazivTipaAviona). Jedan avio-mehaničar može imati licence za više tipova aviona. Osnovna svojstva pilota su: DatumPoslednjegSistematskogPregleda i OcenaZdravstvenogStanja. Svaki pilot je dodeljen nekom konkretnom avionu (OznakaAviona, GodProizvodnje) i u toj posadi ima svoju ulogu (ŠifraUloge, NazivUloge). U modelu obezbediti da jedan pilot može biti dodeljen istom avionu više puta bez obzira na ulogu. Voditi evidenciju i o rasporedu stjuardesa (stjuarda) po avionima. Potrebno je omogućiti pamćenje poznavanja stranih jezika za svaku stjuardesu (stjuarda), moraju znati bar dva strana jezika.

(11 poena)



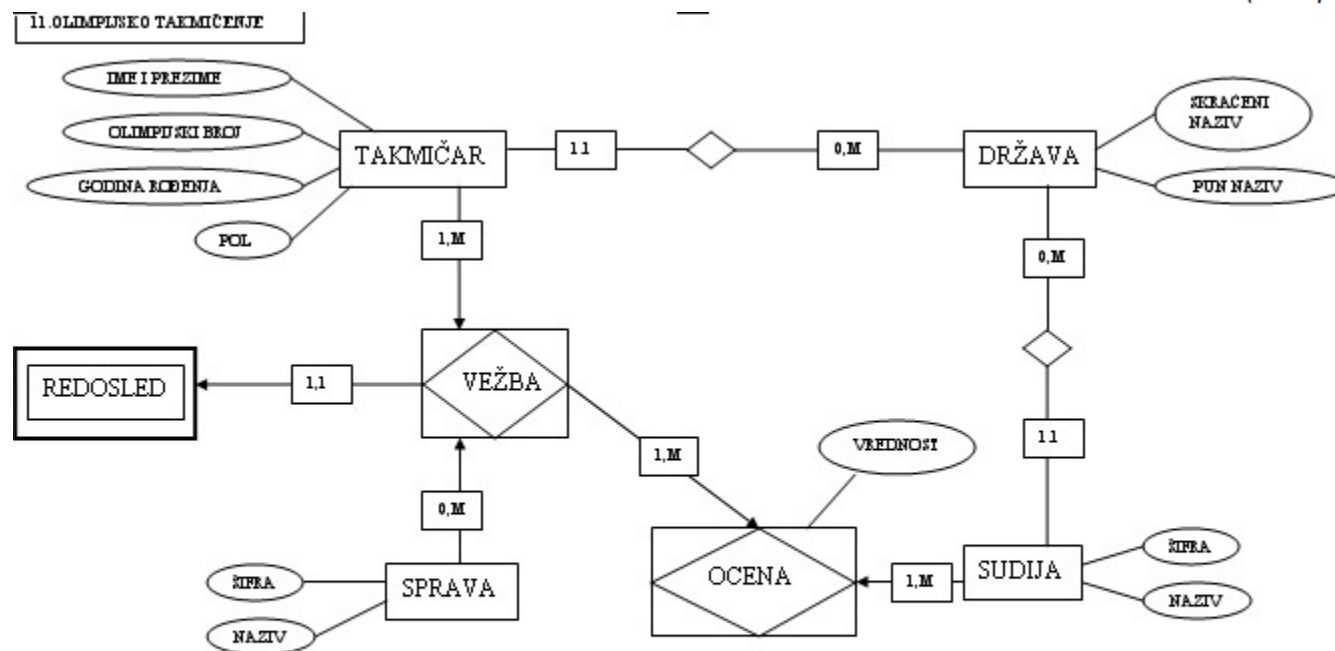
8. Napraviti model objekti-veze za zadati verbalni opis. U cipelgradu žive cipelići i čizmići. I za cipeliće i za čizmiće može se reći da su obućići. Cipelići mogu biti papučići ili nanulići ili opančići. Čizmići se dele na cokuliće i gojzeriće. Cokulići mogu da se druže sa svim cipelićima osim sa papučićima. Nanulić može ali ne mora da se druži sa najviše jednim papučićem, dok jedan papučić može da se druži sa više nanulića. Gojzerići su dovoljni sami sebi tako da se druže međusobno. Kada se udruže cokulići, opančići i nanulići onda se to naziva srpčići.

(11 poena)

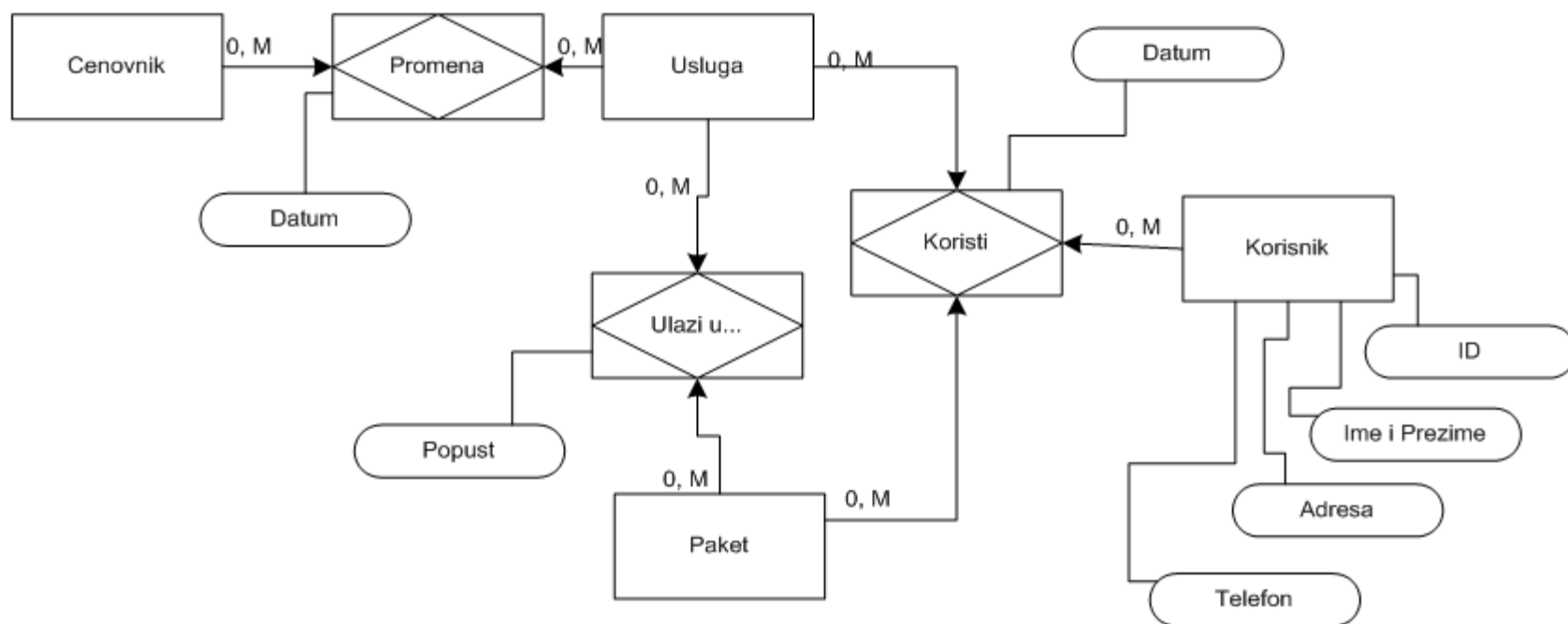


8. Napraviti model objekti-veze za praćenje olimpijskog takmičenja gimnastičkog višeboja. Potrebno je voditi evidenciju o takmičarima sa sledećim osnovnim atributima: OlimpijskiBroj, Prezime, GodinaRođenja, Pol. Svaki takmičar je iz jedne i samo jedne države za koju je potrebno znati Skraćeni naziv i Pun naziv države. Svaki takmičar vežba na više sprava (Šifra sprave, Naziv sprave). Unapred je određen Redosled vežbanja posmatranog takmičara na određenoj spravi. Svaki takmičar za svoju vežbu na određenoj spravi dobija po jednu ocenu od svakog postavljenog sudije. Potrebno je voditi evidenciju sudija sa osnovnim obeležjima kao što su ŠifraSudije, NazivSudije i država iz koje dolaze. Pored ovoga modelirati listu sudija za ocenjivanje vežbi na spravi, t.j. postavljanje jednog ili više sudija za svaku spravu.

(11 poena)

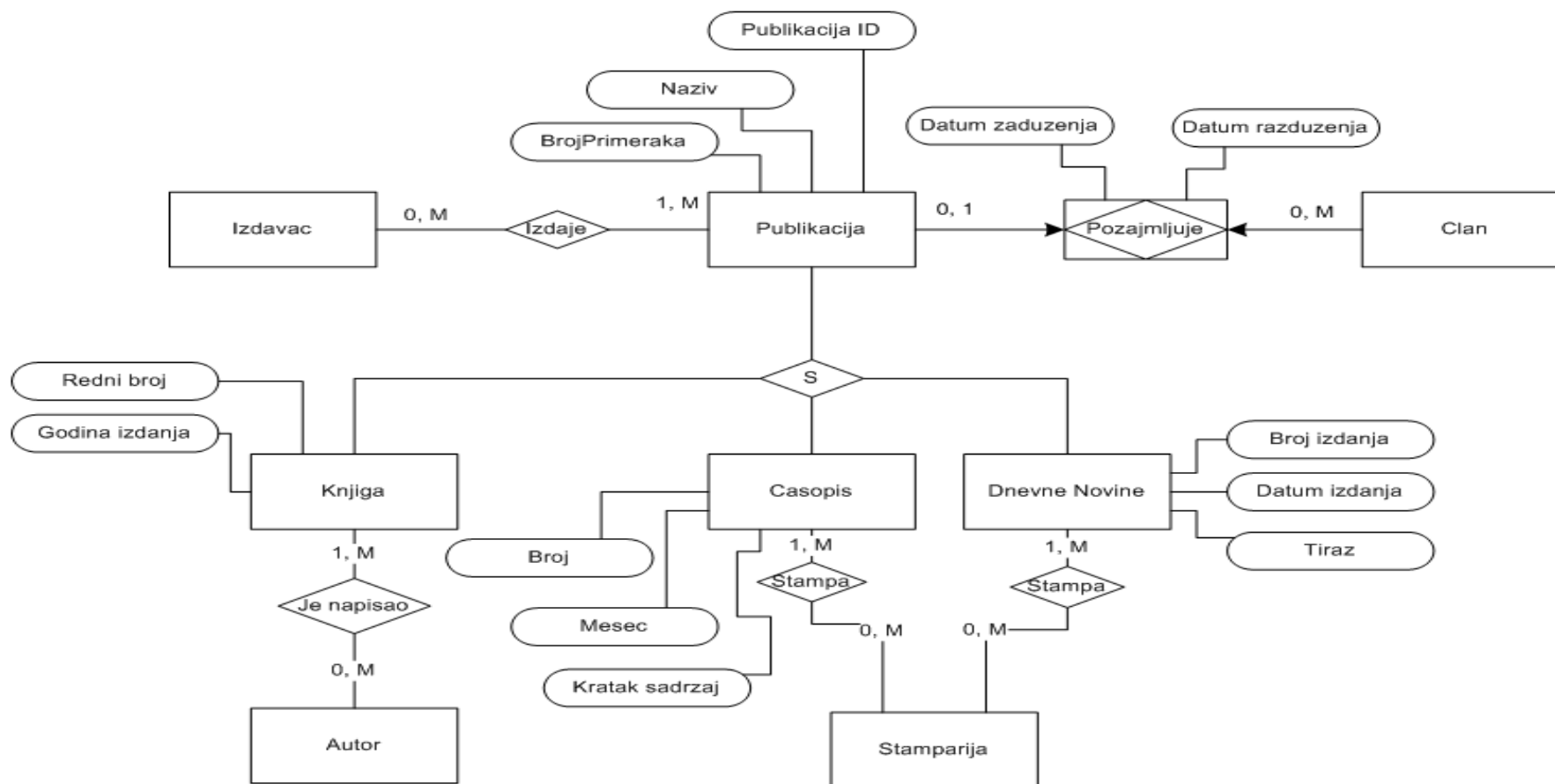


8. Napraviti model objekti-veze za deo informacionog sistema SPA centra. SPA centar nudi razne vrste usluga. Postoje pojedinačne usluge (bazen, sauna, đakuzi, fitnes, masaža, akupunktura ...) i paketi usluga. Paket usluga objedinjuje već postojeće pojedinačne usluge. Potrebno je voditi evidenciju korisnika usluga Centra (ID, ImePrezime, Adresa, KontaktTelefon). Takođe, potrebno je voditi evidenciju korišćenja usluga za svakog korisnika sa tačnim datumom korišćenja bilo da je pojedinačna usluga ili paket usluga. Modelovati cenovnik usluga tako da se prati istorija promene cena za svaku pojedinačnu uslugu. Paket usluga nema definisanu cenu već je samo definisan popust za svaku pojedinačnu uslugu koja je uključena u paket. *(11 poena)*



8. Napraviti model objekti-veze za deo informacionog sistema biblioteke. Potrebno je voditi evidenciju o publikacijama sa sledećim osnovnim atributima: PublikacijaID, Naziv, BrojPrimeraka. Publikacija može biti: knjiga ili časopis ili dnevne novine. Knjiga ima osnovna obeležja redni broj izdanja i godinu izdanja, a pored toga potrebno je beležiti i sve autore. Za časopis potrebno je pamtitii broj, mesec i kratak sadržaj, za dnevne novine broj izdanja, datum izdanja i tiraž. I za časopise i za dnevne novine potrebno je znati štampariju koja je štampala ta izdanja. I za knjigu i za časopis i za dnevne novine treba voditi evidenciju o izdavaču. Za svaku publikaciju obezbediti evidentiranje datuma zaduženja i datuma razduženja od strane člana biblioteke.

(10 poena)



Tvoj momak je sin ljubavnice mog muža? -Ne vredajte mi majku!

```

classDiagram
    class Muž
    class Žena
    class Ljubavnica
    class Sin
    class Devojka
    class Majka
    class Uvreda

    Muž --> Ljubavnica : ima
    Muž --> Sin : ima
    Ljubavnica --> Sin : ima
    Žena --> Sin : ima
    Žena --> Uvreda : ima
    Sin --> Devojka : ima
    Devojka --> Majka : ima
    Majka --> Uvreda : ima
  
```

PMOV
 Najzad nečemu koristi!