

Baze podataka - kolokvijum

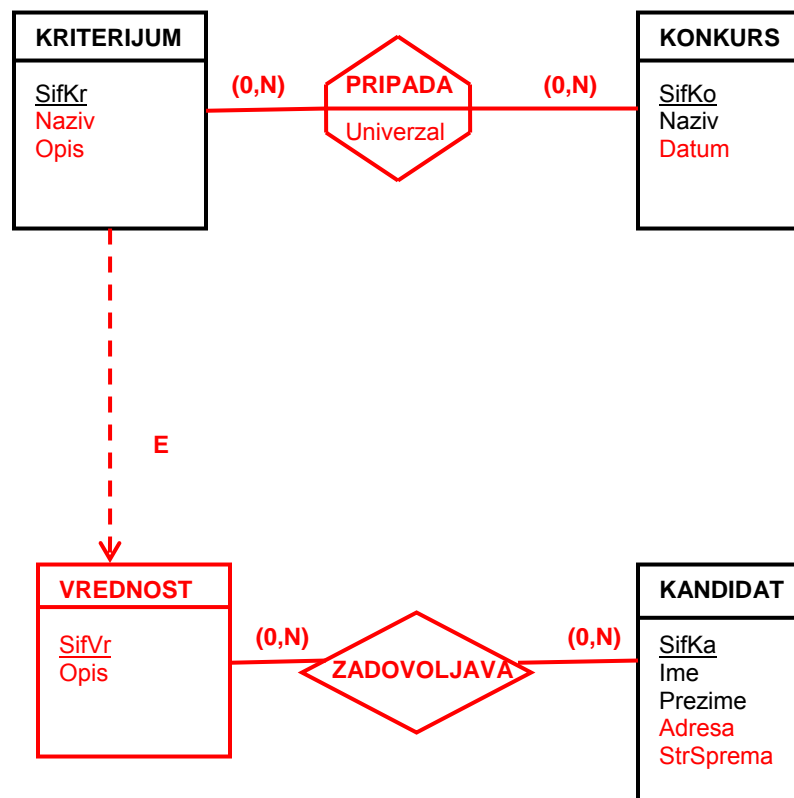
Kolokvijum traje **120** minuta

Ime prezime i broj indeksa studenta	Potpis dežurnog	Broj poena

Napomena: Nije dozvoljena upotreba literature.

1. Posmatra se deo baze podataka za potrebe ocenjivanja kandidata na konkursima za poslove. U sistemu se vodi evidencija o konkursima, njihovim nazivima i datumima objavljivanja. Svaki konkurs može imati veći broj kriterijuma koji se ocenjuju. Pored naziva i opisa za kriterijum postoje i moguće vrednosti i njihovi opisi. Kriterijum može biti univerzalan, u kom slučaju može pripadati većem broju konkursa. Kandidati se evidentiraju nezavisno od konkursa i za njih se prate ime, prezime, adresa i nivo stručne spreme (mogu biti nisko, srednje ili visoko kvalifikovani). Za kandidate mogu biti evidentirane vrednosti koje zadovoljavaju za bilo koji od kriterijuma. Za opisani sistem sastaviti model entiteta i odnosa tako što treba:

a. (4) Ukloniti suvišne (odnosno dodati potrebne) attribute/entitete, a potom dodati neposredne i posredne odnose (slabe entitete nije potrebno duplo zaokruživati).



b. (4) Da li je model u tački pod a) moguće, uvođenjem odgovarajućih odnosa i naznačavanjem primarnih ključeva, postići da kandidat ne može imati veći broj vrednosti za bilo koji od kriterijuma. Ako je moguće, onda odgovoriti kako.

Odgovor:

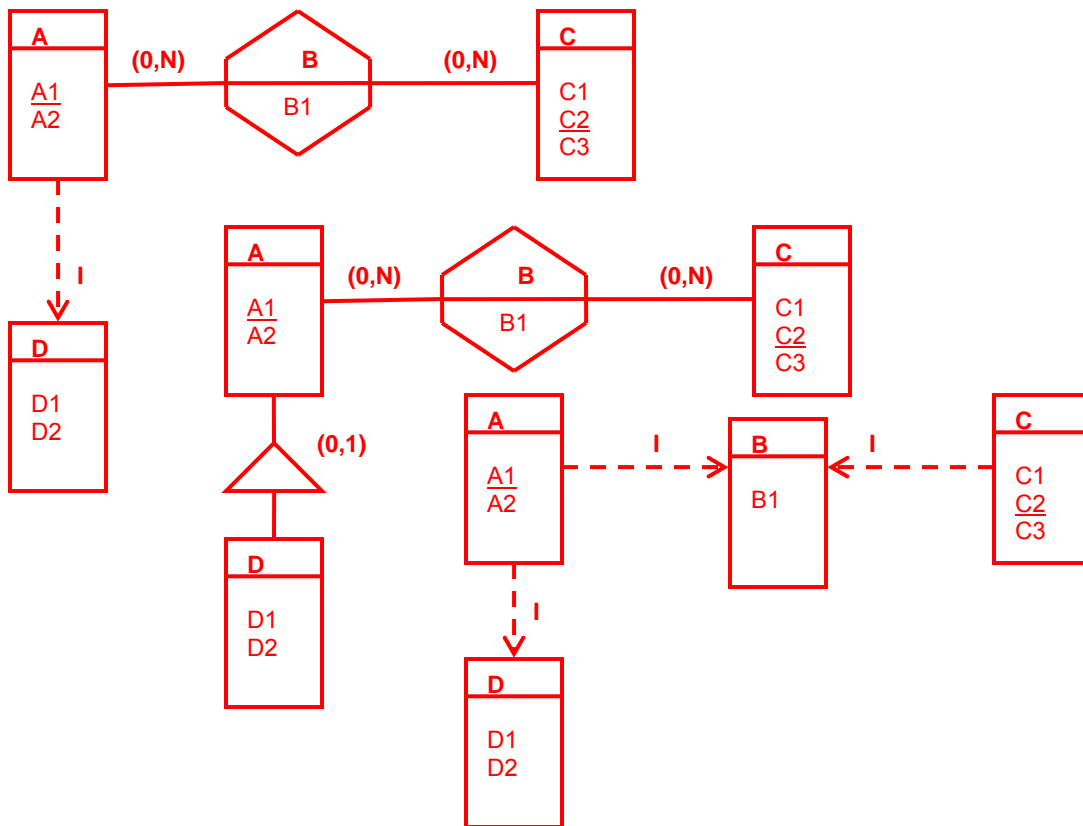
Postoji više mogućih rešenja. Npr. da se u modelu u tački pod a), egzistencijalna zavisnost između Kriterijuma i Vrednosti pretvori u identifikacionu (atribut SifVr se ne uklanja jer on ima ulogu diskriminatora). Potom se u vezi Zadovoljava kao primarni ili kao kandidat ključ odaberu SifKa i SifKr.

Još jedno moguće rešenje bilo bi upotreba veze između Kriterijuma i Vrednosti (umesto egzistencijalne zavisnosti). Potom, ta veza kao agregacija ušla u vezu Zadovoljava sa entitetom Kandidat. Odabir primarnog ili kandidat ključa za vezu Zadovoljava bio bi kao u prethodnom komentaru (SifKa i SifKr).

2.(3) Dat je deo šeme relacione baze podataka, prikazati tri moguća modela entiteta i odnosa kojima data šema odgovara. Primarni ključevi su podvučeni a strani ključevi zaokruženi. (razlike postići uvođenjem različitih tipova odnosa, posrednih i neposrednih, a ne promenom kardinalnosti nad istim odnosima)

A(A1, A2); B(A1, C2, B1); C(C1, C2, C3); D(A1, D1, D2)

Odgovor:



3. Dat je deo šeme relacione baze podataka telekomunikacione kompanije za potrebe evidentiranja prometa u toku jedne godine za prepaid korisnike mobilnih telefona. Datumi se predstavljaju kao celobrojne vrednosti u opsegu od 1 do 365. Svaki red u tabeli Razgovori predstavlja ukupnu cenu razgovora određenog datuma.

PRETPLATNICI (SifP, Ime, BrLK, BrTelefona)

UPLATE (SifU, SifP, DatumU, Iznos)

RAZGOVORI (SifR, SifP, DatumR, CenaR)

PORUKE (SifS, SifP, DatumP, CenaP)

a.(2) Napisati SQL upit koji vraća isti rezultat kao i sledeći upit relacione algebre:

$$\pi_{\text{Ime}} (\pi_{\text{SifP}} (\text{Razgovori} \underset{\substack{\text{Razgovori.SifP=Uplate.SifP} \\ \text{DatumR=DatumU} \\ \text{CenaR} \geq \text{Iznos}}}{\bowtie} \sigma_{\text{Iznos} > 200} (\text{Uplate})) \underset{*}{\bowtie} \text{Pretplatnici})$$

Odgovor:

```
SELECT DISTINCT P.Ime
FROM Razgovori R, Uplate U, Pretplatnici P
WHERE R.SifP = U.SifP
      AND R.DatumR = U.DatumU
      AND R.CenaR >= U.Iznos
      AND U.Iznos > 200
      AND R.SifP = P.SifP
```

b. (2) Sastaviti SQL skript kojim se za sve razgovore čija je cena veća i jednaka 25, cena smanjuje za 10%, a svih čija je cena manja od 25, povećava za 15% .

Odgovor:

```
UPDATE Razgovori
SET CenaR = CenaR * 1.80
WHERE CenaR >= 25;
```

```
UPDATE Razgovori
SET CenaR = CenaR * 1.15
WHERE CenaR < 25;
```

```
UPDATE Razgovori
SET CenaR = CenaR * 0.5
WHERE CenaR >= 45;
```

c. (5) Sastaviti SQL skript koji vraća šifre i brojeve telefona pretplatnika koji su (u toku posmatrane godine) imali najduži period u kome je funkcija cene razgovora po datumu ($f_{CenaR}(DatumR)$) bila monotono neopadajuća.

Odgovor:

```
CREATE VIEW Periodi (SifP, DatumP, DatumK)
AS SELECT R1.SifP, R1.DatumR, MAX(R2.DatumR)
FROM Razgovori R1, Razgovori R2
WHERE R1.DatumR < R2.DatumR AND R1.CenaR < R2.CenaR, AND R1.SifP = R2.SifP
AND NOT EXISTS ( SELECT *
                  FROM Razgovori R3
                  WHERE R3.DatumR BETWEEN R1.DatumR AND R2.DatumR
                  AND R3.SifP = R1.SifP
                  AND R3.CenaR < ANY ( SELECT R4.CenaR
                                      FROM Razgovori R4
                                      WHERE R4.SifP = R1.SifP
                                      AND R4.DatumR BETWEEN R1.DatumR AND R3.DatumR
                                      )
                )
GROUP BY R1.SifP, R1.DatumR;

SELECT P.SifP, P.BrTelefona
FROM Pretplatnici P
WHERE P.SifP IN ( SELECT R.SifP
                 FROM Periodi R
                 WHERE (R.DatumK - R.DatumP) = ( SELECT MAX(DatumK - DatumP)
                                                  FROM Periodi
                                                  )
                )
);
```