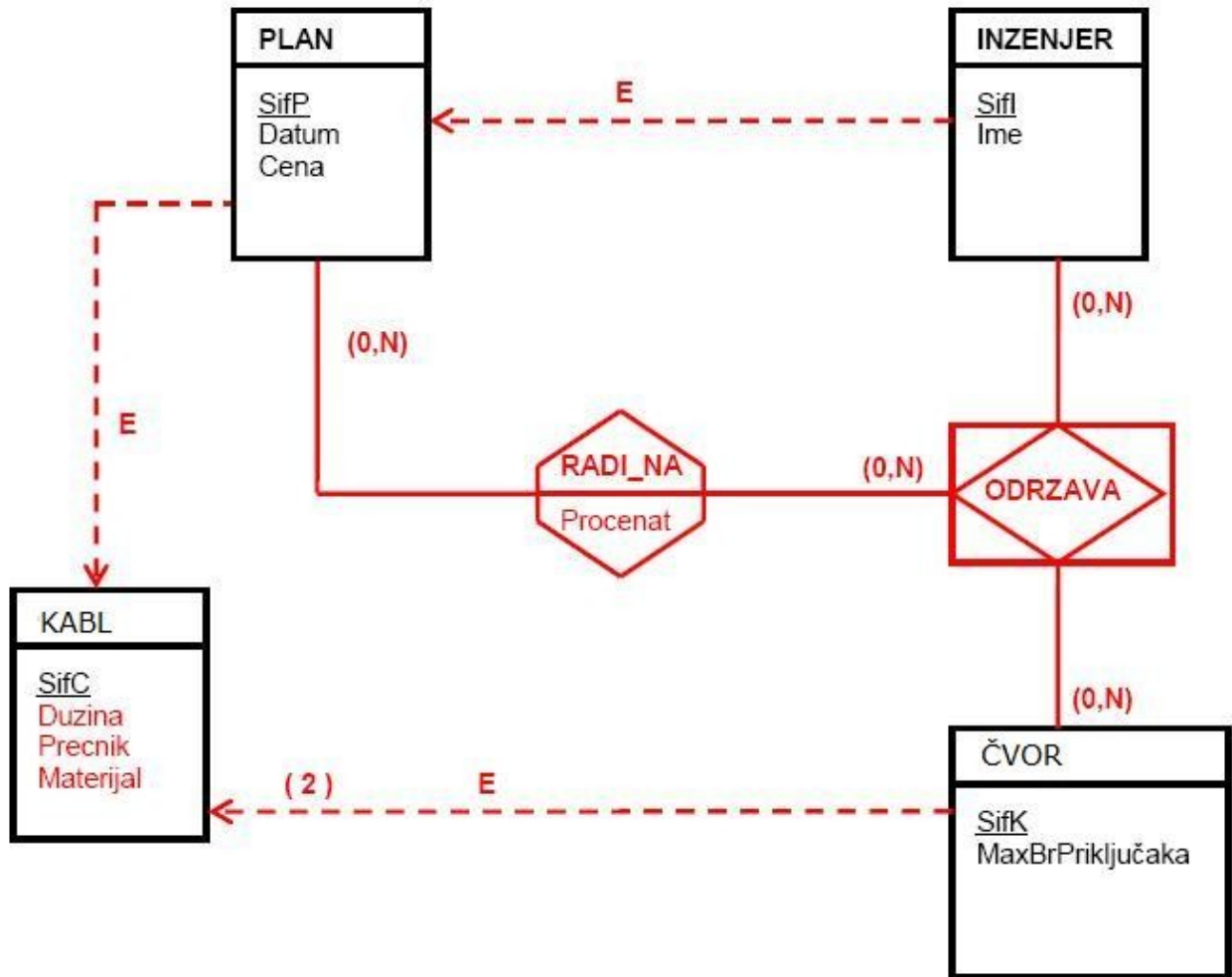


Objedinjeni rezultati (max 30 poena)

1. Predstaviti modelom entiteta i odnosa deo baze podataka jednog preduzeca. Vodi se evidencija o mrežnim kablovima, njihovoj dužini, prečniku, materijalu, kao i projektnom planu po kom je obavljeno montiranje. Kabl se prostire između dva mrežna čvora. Mrežni čvorovi mogu imati veći broj priključaka. Evidentira se maksimalni broj priključaka. Mrežne čvorove redovno održava jedan ili više inženjera. Projektni plan (datum donošenja, planirana cena) mora imati jednog glavnog inženjera, dok na njegovoj implementaciji može učestvovati više inženjera, ali samo onih koji održavaju bar jedan (proizvoljni) mrežni čvor. Inženjer koji radi na implementaciji plana može učestvovati sa 50%, 75%, 100% radnog vremena.



2. Data je šema relacione deo baze podataka fakulteta:

STUDENT (SifS, Ime, BrIndeksa);

PROFESOR (SifP, Ime, SifO);

ODSEK (SifO, Naziv);

KURS (SifK, Naziv, BrKredita, SifO)

UČIONICA (SifU, BrMesta);

POHAĐA(SifS, SifR)

PREDUSLOV (SifK, SifKP) POHAĐA(SifS, SifR);

RASPORED (SifR, SifP, SifK, SifU, Termin, Dan, Br.Prijavljenih);

Sastaviti SQL skript kojim se formira tabela RASPORED, ukoliko je poznato da samo jedan profesor može držati po rasporedu predavanje u jednom terminu u jednoj učionici.

Pri tom broj prijavljenih mora biti manji od broja mesta u učionici koja je rasporedu predviđena za taj kurs. Vrednost atributa termin mora biti celobrojna vrednost u opsegu od 1 do 7. Vrednost atributa dan iz skupa vrednosti {Pon, Uto, Sre, Cet, Pet}.

```
CREATE TABLE RASPORED
```

```
(  
  SifRINT PRIMARY KEY,  
  SifPINT NOT NULL REFERENCES PROFESOR ON UPDATE CASCADE,  
  SifKINT NOT NULL REFERENCES KURS ON UPDATE CASCADE,  
  SifUINT NOT NULL REFERENCES UCIONICA ON UPDATE CASCADE,  
  TerminINT NOT NULL CHECK (TerminBETWEEN 1 and 7),  
  Dan CHAR(3) NOT NULL CHECK (Dan IN ('PON', 'UTO', 'SRE', 'CET', 'PET')),  
  BrPrijavljenihINT NOT NULL,  
  UNIQUE (SifP, Termin, Dan),  
  UNIQUE (SifU, Termin, Dan)  
);
```

```
CREATE ASSERTION ProveraBrPrijavljenih
```

```
CHECK (NOT EXISTS (SELECT *  
                    FROM UCIONICA U, RASPORED R  
                    WHERE (U.SifU=R.SifU) AND ( R.BrPrijavljenih> (U.BrMesta))  
                    )  
);
```

3. Sastaviti SQL skript koji proverava kvalitet rasporeda studenata i za svakog studenta pored imena studenta i broja indeksa ispisuje i odgovarajuću poruku o kvalitetu rasporeda. Raspored studenta je LOS ukoliko u svom rasporedu bar jednog dana ima prekid u terminima u kojima prati predavanje, u suprotnom raspored studenta je DOBAR.

```
CREATE VIEW LosRaspored(SifS, Dan)
```

```
AS SELECT P.SifS, R.Dan
```

```
FROM POHADJA P, RASPORED R
```

```
WHERE P.SifR=R.SifR
```

```
GROUP BY P.SifS, R.Dan
```

```
HAVING COUNT(R.Termin) < (MAX(R.Termin)-MIN(R.Termin)+1);
```

```
SELECT S.Ime,S.BrIndeksa, 'LOS'
```

```
FROM STUDENT S
```

```
WHERE S.SifS IN (SELECT SifS FROM LosRaspored)
```

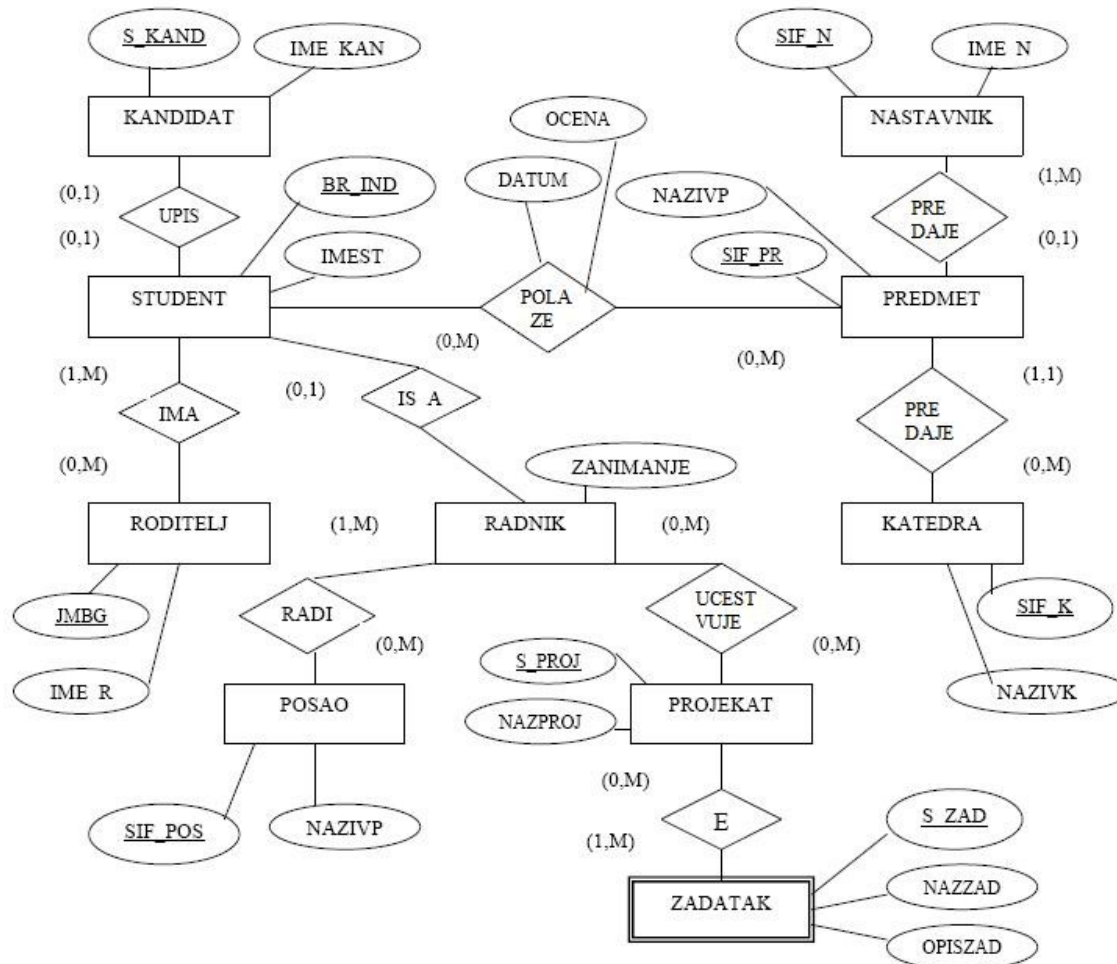
```
UNION
```

```

SELECT S.Ime,S.BrIndeksa, 'DOBAR'
FROM STUDENT S
WHERE S.SifS NOT IN (SELECT SifS FROM LosRaspored)
AND S.SifS IN (SELECT SifS FROM POHADJA);

```

4. Transformisati prikazani model entiteta i odnosa u shemu relacione baze podataka.



· Prevedimo prvo entitete.

Primenom pravila za entitete po kome svaki entitet postaje posebna šema relacije, dobijamo sledeće šeme relacija:

```

Kandidat({S_KAND, IME_KAN}, {S_KAND})
Student({BR_IND, IMEST}, {BR_IND})
Predmet({SIF_PR, NAZIVP}, {SIF_PR})
Nastavnik({SIF_N, IME_N}, {SIF_N})
Roditelj({JMBG, IMER}, {JMBG})
Katedra({SIF_K, NAZIVK}, {SIF_K})
Posao({SIF_POS, NAZIVP}, {SIF_POS})
Projekat({S_PROJ, NAZPROJ}, {S_PROJ})

```

Primenom pravila po kome svaki slab entitet postaje šema relacije (u našem primeru egzistencijalno zavisni tip entiteta ZADATAK), dobijamo još jednu šemu relacije:

```

Zadatak({S_ZAD, NAZZAD, OPISZAD, S_PROJ}, {S_ZAD, S_PROJ})

```

Primenom pravila po kome entitet podtip (u našem primeru RADNIK) dobijamo još jednu šemu relacije: Radnik({BR_IND, ZANIMANJE}, {BR_IND})

· Sada prelazimo na prevodjenje veza.

Prevedimo, prvo, tipove veza sa kardinalitetima 1:1. To je sledeća veza:

1. Upis

Prema pravilu, kreiraju se tri šeme relacija: po jedna šema relacije za svaki entitet u vezi (to smo već kreirali) i jedna šema relacije za poveznik, odnosno kreiramo *novu* šemu relacije:

Upis({BR_IND, S_KAND}, {BR_IND})

• Predjimo, zatim, na sledeće veze sa kardinalitetima 1:M. To su sledeći veznici:

2. Ima

Prema pravilu veznik ovoga tipa tj. (1,M) : (0,M) postaje posebna šema relacije, pa tako dobijamo *još jednu* šemu relacije: *Ima*({JMBG, BR_IND}, {JMBG, BR_IND})

3. Polaže

Kao i u prethodnom slučaju, prema pravilu veznik ovoga tipa tj. (0,M) : (0,M) postaje posebna šema relacije, pa tako dobijamo *još jednu* šemu relacije:

Polaze({BR_IND, SIF_PR, DATUM, OCENA}, {BR_IND, SIF_PR})

4. Predaje

Prema pravilu ovaj tip veznika se prevodi u posebnu šemu relacije, pa dobijamo, takodje, *novu šemu relacije*: *Predaje*({SIF_PR, SIF_N}, {SIF_PR})

5. Pripada

Prema pravilu ovaj tip veznika ne postaje posebna šema relacije, nego se veznik prevodi tako što ključ tipa entiteta sa strane za koju je gornja granica kardinaliteta preslikavanja M (u našem primeru to je entitet *KATEDRA*) postaje obeležje šeme relacije koja odgovara entitetu sa strane poveznika za koju je gornja granica kardinaliteta preslikavanja 1 (u našem primeru to je entitet *PREDMET*). To znači, da modifikujemo šemu relacije *Predmet* na sledeći način:

Predmet({SIF_PR, NAZIVP, SIF_K}, {SIF_PR})

6. Radi

Prema pravilu veznik ovoga tipa tj. (1,M) : (0,M) postaje posebna šema relacije, pa tako dobijamo *još jednu* šemu relacije: *Radi*({BR_IND, SIF_POS}, {BR_IND, SIF_POS})

7. Učestvuje

Kao i u slučaju 6. veznika *RADI*, i u ovom slučaju dobijamo još jednu šemu relacije, za ovaj veznik: *Učestvuje*({BR_IND, S_PROJ}, {BR_IND, S_PROJ})

Kao rezultat prevodjenja modela entiteta i poveznika sa slike dobijamo sledeći skup šema relacija:

Kandidat({S_KAND, IME_KAN}, {S_KAND})

Student({BR_IND, IMEST}, {BR_IND})

Predmet({SIF_PR, NAZIVP, SIF_K}, {SIF_PR})

Nastavnik({SIF_N, IME_N}, {SIF_N})

Roditelj({JMBG, IMER}, {JMBG})

Katedra({SIF_K, NAZIVK}, {SIF_K})

Posao({SIF_POS, NAZIVP}, {SIF_POS})

Projekat({S_PROJ, NAZPROJ}, {S_PROJ})

Zadatak({S_ZAD, NAZZAD, OPISZAD, S_PROJ}, {S_ZAD, S_PROJ})

Radnik({BR_IND, ZANIMANJE}, {BR_IND})

Upis({BR_IND, S_KAND}, {BR_IND})

Ima({JMBG, BR_IND}, {JMBG, BR_IND})

Polaze({BR_IND, SIF_PR, DATUM, OCENA}, {BR_IND, SIF_PR})

Predaje({SIF_PR, SIF_N}, {SIF_PR})

Radi({BR_IND, SIF_POS}, {BR_IND, SIF_POS})

Učestvuje({BR_IND, S_PROJ}, {BR_IND, S_PROJ})