

PROJEKAT iz predmeta “BAZE PODATAKA 2”

IME I PREZIME		Đorđe Krstović
Broj indeksa		
Studijski modul		Poslovna inteligencija
Stečena diploma osnovnih akademskih studija		Diplomirani inženjer softvera
NAZIV PROJEKTA		Sistem za evidenciju radnog vremena preduzeća „Infrastruktura železnice Srbije“ A.D.
Mentor		Prof. Dr. Nenad Aničić
Napomena		
Verzija	Datum promene	Opis promene
v1.1	19/11/2020	Dorađen model podataka
v1.2	28/11/2020	Dorađen koeficijent radnog mesta, i isplatni list + dodata denormalizacija
v1.3	10/12/2020	Trigeri, distinkt i struktuirani tip
v1.4	23/12/2020	Implementacija trigeru u SQL Developeru, distinkt i struktuirani tip, Optimizacija baze podataka

SADRŽAJ

OPIS PROJEKTA.....	3
Opis podsistema za koji se projektuje baza podataka	3
Poslovna dokumenta koja se koriste u procesu	3
PROJEKTOVANJE BAZE PODATAKA	4
Model podatka (PMOV)	4
Relacioni model.....	7
DENORMALIZACIJA RELACIJA.....	8
Denormalizacija relacija uz narušavanje 2NF.....	8
Pre-joining.....	8
Denormalizacija relacija uz narušavanje 3NF.....	10
Pre-joining.....	10
Short-circuit keys	11
KORISNIČKI DEFINISANI TIPOVI I TRIGERI	12
Definicija korisničkom tipa i njegovo korišćenje.....	12
Distinkt tip:.....	12
Strukturirani tip:	12
Realizacija proceduralne logike uz pomoć trigera	12
OPTIMIZACIJA BAZE PODATAKA	18
Definisanje i analiza indeksa	18
B-TREE indeks:.....	18
FUNCTION-BASED indeks:.....	19
Horizontalno particionisanje.....	20
Vertikalno particionisanje	21
Primena drugih optimizacionih tehnika.....	23
PMOV pre primene storing derivable values tehnike:.....	23
PMOV nakon primene storing derivable values tehnike:	23
TEHNOLOGIJE ZA IMPLEMENTACIJU PROJEKTA	26
SUBP korišćen za implementaciju baze podatka (kratak opis)	26
Programsko okruženje za razvoj korisničkog interfejsa (kratak opis).....	26

OPIS PROJEKTA

Opis podsistema za koji se projektuje baza podataka

Preduzeće „Infrastruktura železnice Srbije“ A.D. se bavi upravljanjem i održavanjem infrastrukturnih objekata „Železnice Srbije“. U fokusu ovog rada biće podsistem za evidenciju radnog vremena u preduzeću „Infrastruktura železnice Srbije A.D.“. Sadrži podatke o zaposlenima, izvode njihovog ličnog dohotka, radnim mestima, podatke o lekarskim pregledima, godišnjem odmoru, dežurstvu, broju radnih sati, kao i koeficijent svakog radnog mesta u preduzeću na osnovu kog se vrši obračun.

Poslovna dokumenta koja se koriste u procesu

Za potrebe razvoja ovog informacionog sistema korišćeni su poslovni dokumenti koji se koriste svakodnevno u preduzeću „Infrastruktura železnice Srbije A.D.“. Ovaj sistem predstavlja studentski projekat na master studijama Fakulteta organizacionih nauka za predmet Baze podataka 2. U daljem razvoju projekta biće implementiran samo deo koji se odnosi na radno vreme zaposlenih.

Korišćeni dokumenti za razvoj ovog sistema su:

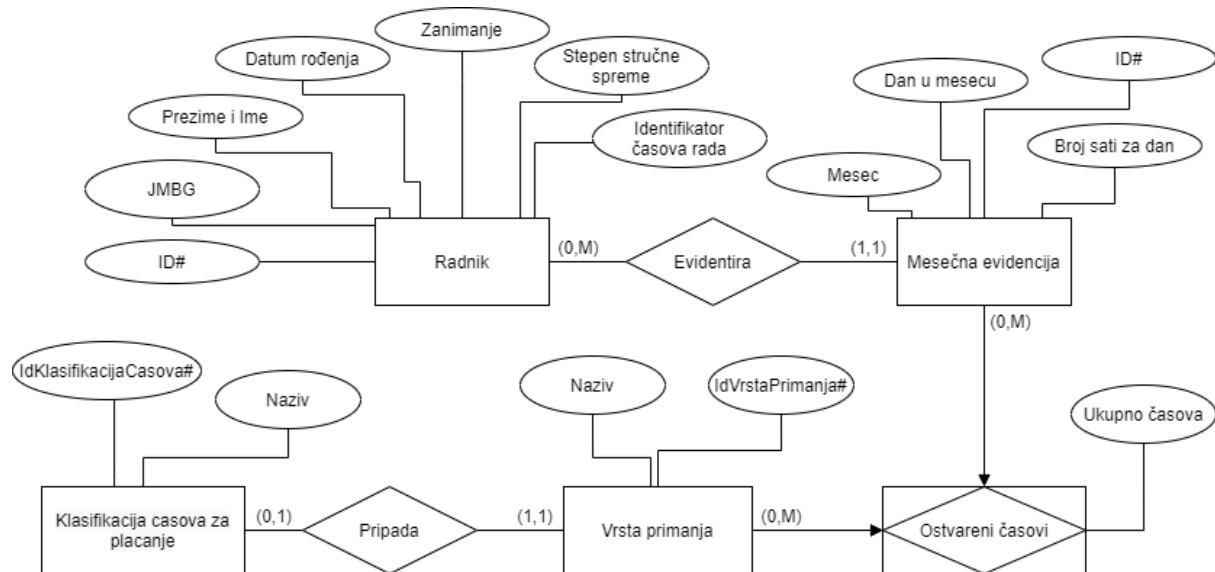
- Evindencija radnog vremena
- Koeficijenti radnih mesta po metodologiji
- Isplatni list zaposlenih
- Lekarski pregled zaposlenih
- Godišnji odmor zaposlenih
- Dežurstva zaposlenih

NAPOMENA: Fotografije dokumenata su izbačene iz projekta u verziji koja je dostupna online, i u toku razvoja izvršene su promene za potrebe prilagođavanja sistema.

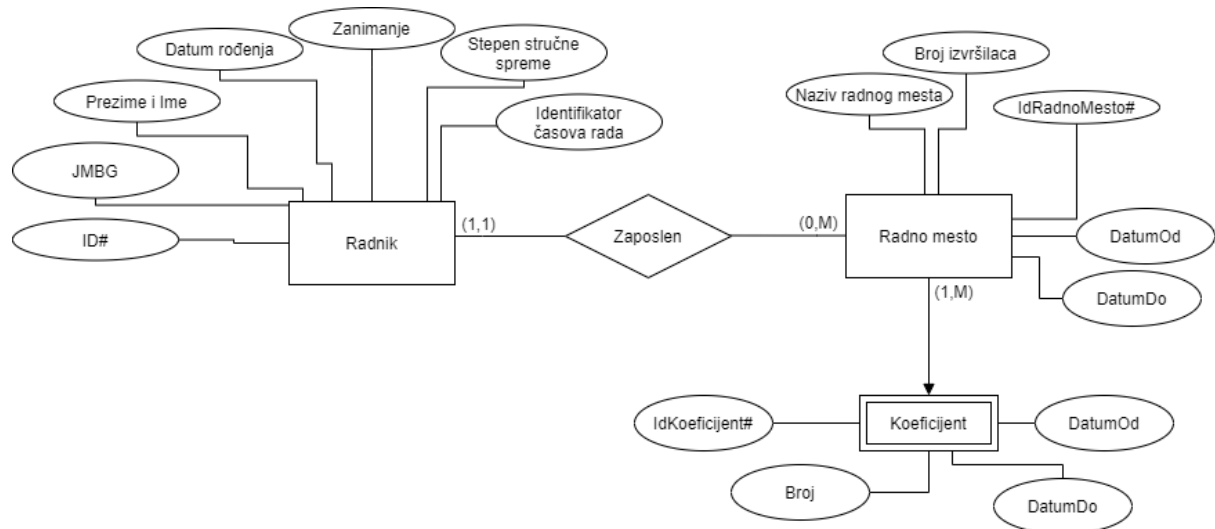
PROJEKTOVANJE BAZE PODATAKA

Model podatka (PMOV)

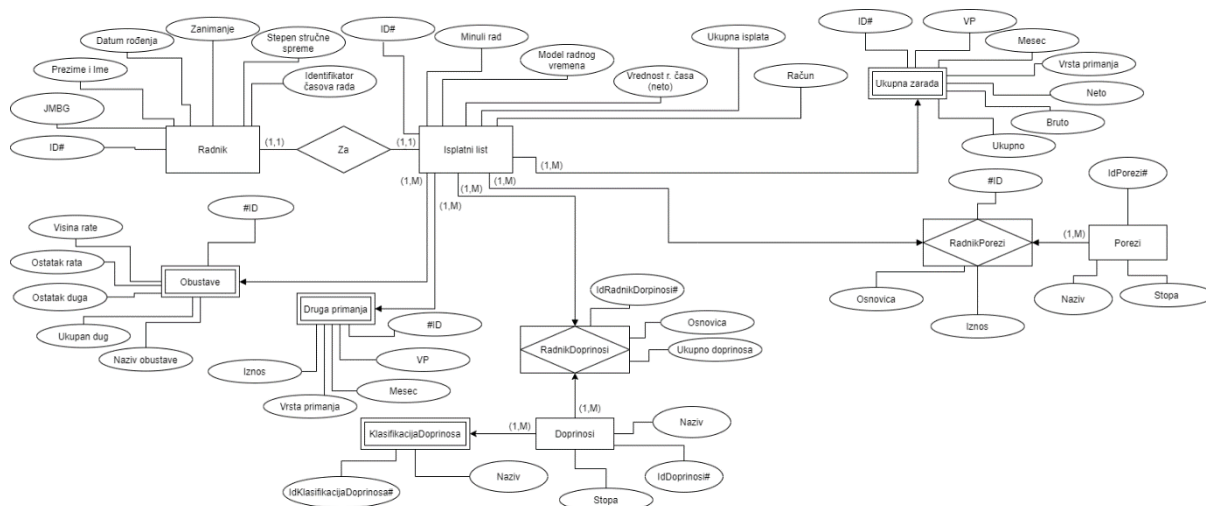
Evidencija radnog vremena



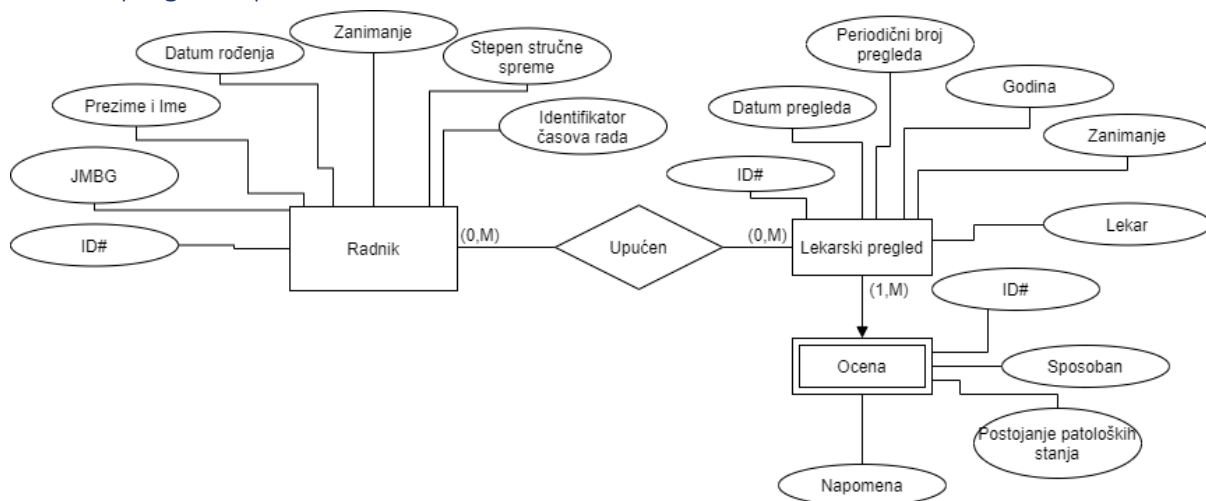
Koeficijent radnih mesta



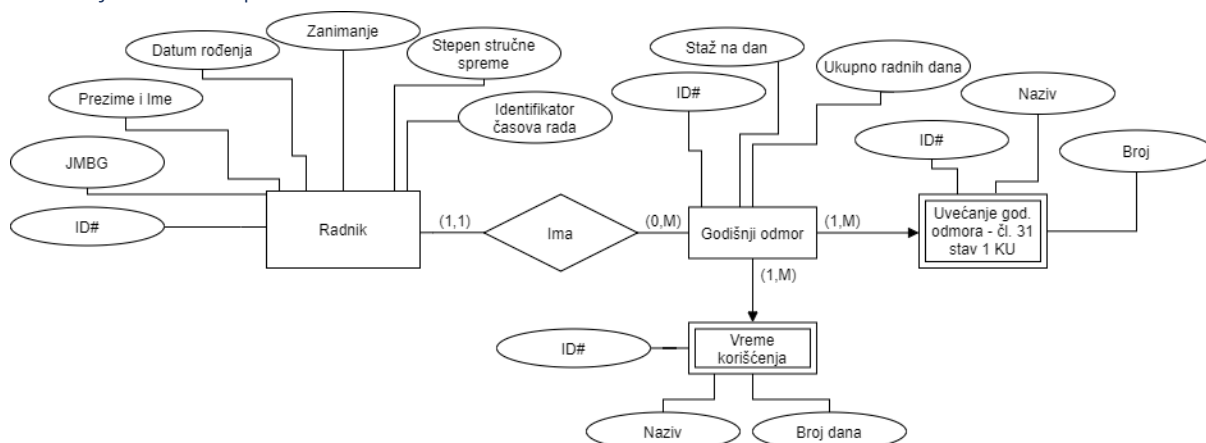
Isplatni list zaposlenih



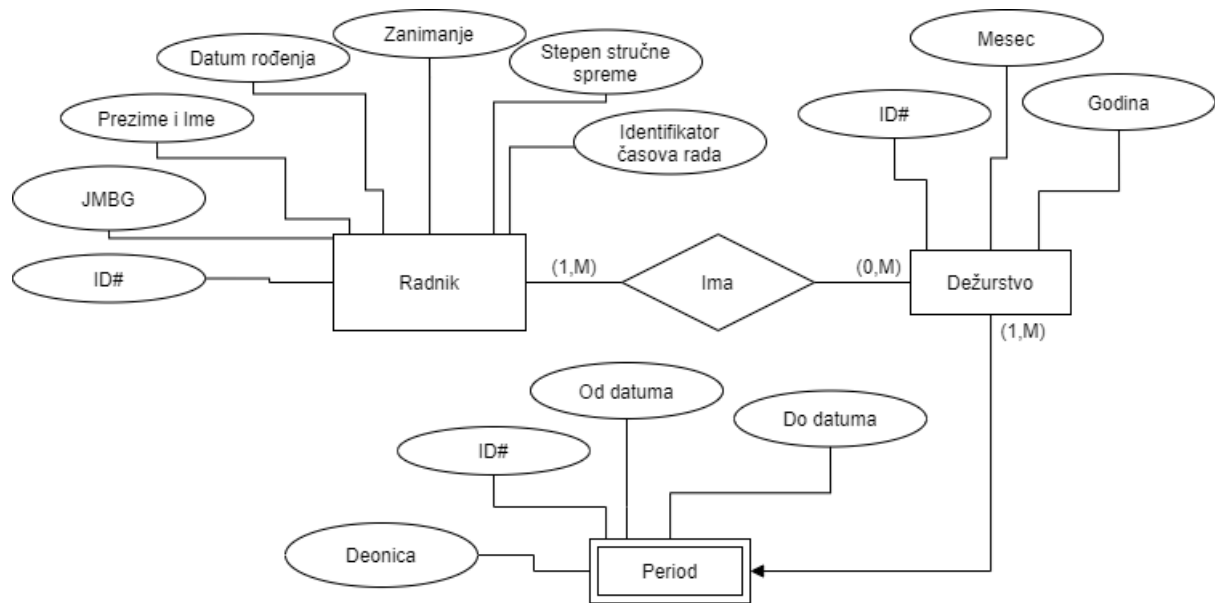
Lekarski pregled zaposlenih



Godišnji odmor zaposlenih



Dežurstva zaposlenih



Relacioni model

Radnik (**IdRadnik**, JMBG, PrezimeIme, DatumRodjenja, Zanimanje, StepenStrucneSpreme, IdentifikatorCasovaRada, *IdRadnoMesto*)

MesecnaEvidencija(**IdMesecnaEvidencija**, Mesec, DanUMesecu, BrojSatiZaDan, *IdRadnik*)

OstvareniCasovi(**IdVrstaPrimanja**, **IdMesecnaEvidencija**, UkupnoCasova)

VrstaPrimanja(**IdVrstaPrimanja**, Naziv, *IdKlasifikacijaCasova*)

KlasifikacijaCasova(**IdKlasifikacijaCasova**, Naziv)

RadnoMesto(**IdRadnoMesto**, Naziv, BrojIzvršilaca, DatumOd, DatumDo, *IdKoeficijent*)

Koeficijent(**IdKoeficijent**, Broj, DatumOd, DatumDo)

IsplatniList(**IdIsplatniList**, MinRad, ModRadVr, VrednostRCasa, UkupnoIsplata, Racun, *IdRadnik*)

Obustave(**IdObustave**, Rata, OstatakRata, OstatakDuga, UkupanDug, Naziv, *IdIsplatniList*)

DrugaPrimanja(**IdDrugaPrimanja**, Iznos, VrstaPrimanja, Mesec, VP, *IdIsplatniList*)

RadnikDoprinosi(**IdRadnikDoprinosi**, **IdIsplatniList**, **IdDoprinosi**, Ukupno, Osnovica)

Doprinosi(**IdDoprinosi**, Stopa, Naziv)

KlasifikacijaDoprinosa(**IdKlasifikacijaDoprinosa**, Naziv, *IdDoprinosi*)

RadnikPorezi(**IdRadnikPorezi**, **IdIsplatniList**, **IdPorezi**, Iznos, Osnovica)

Porezi(**IdPorezi**, Stopa, Naziv)

LekarskiPregled(**IdLekarskiPregled**, DatumPr, PeriodicniBrojPr, Godina, Zanimanje, Lekar, *IdRadnik*)

Ocena(**IdOcena**, Sposoban, PostojanjePatStanja, Napomena, *IdLekarskiPregled*)

GodisnjiOd(**IdGodisnjiOd**, StazNaDan, UkupnoRadDana, *IdRadnik*)

UvecanjeGodOd(**IdUvecanjeGodOd**, Naziv, Broj, *IdGodisnjiOd*)

VremeKoriscenja(**IdVremeKoriscenja**, Naziv, BrojDana, *IdGodisnjiOd*)

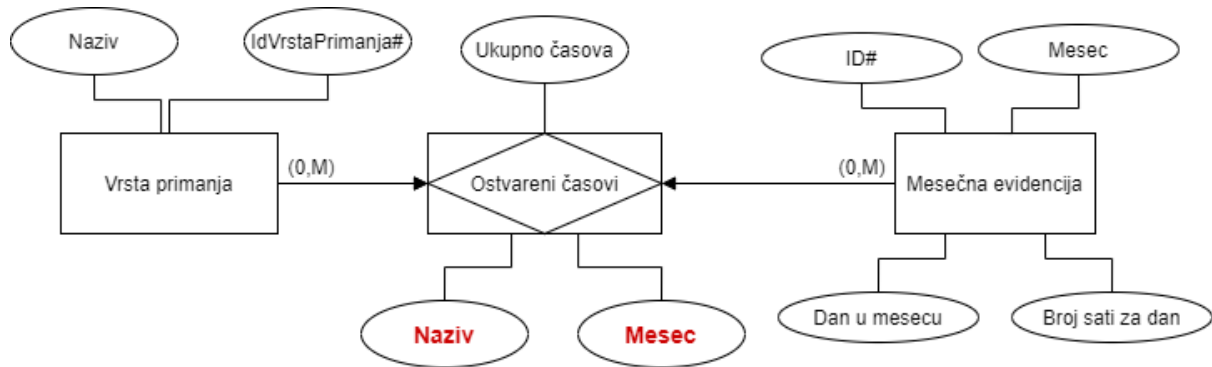
Dezurstvo(**IdDezurstvo**, Mesec, Godina, *IdRadnik*)

Period(**IdPeriod**, OdDatuma, DoDatuma, Deonica, *IdDezurstvo*)

DENORMALIZACIJA RELACIJA

Denormalizacija relacija uz narušavanje 2NF

Pre-joining



MesecnaEvidencija(IdMesecnaEvidencija, Mesec, DanUMesecu, BrojSatiZaDan, IdRadnik)

OstvareniCasovi(IdVrstaPrimanja, IdMesecnaEvidencija, UkupnoCasova, **Naziv**, **Mesec**)

VrstaPrimanja(IdVrstaPrimanja, Naziv, IdKlasifikacijaCasova)

Funkcionalne zavisnosti:

IdVrstaPrimanja, IdMesecnaEvidencija -> UkupnoCasova, **Naziv**, **Mesec**

IdVrstaPrimanja -> Naziv

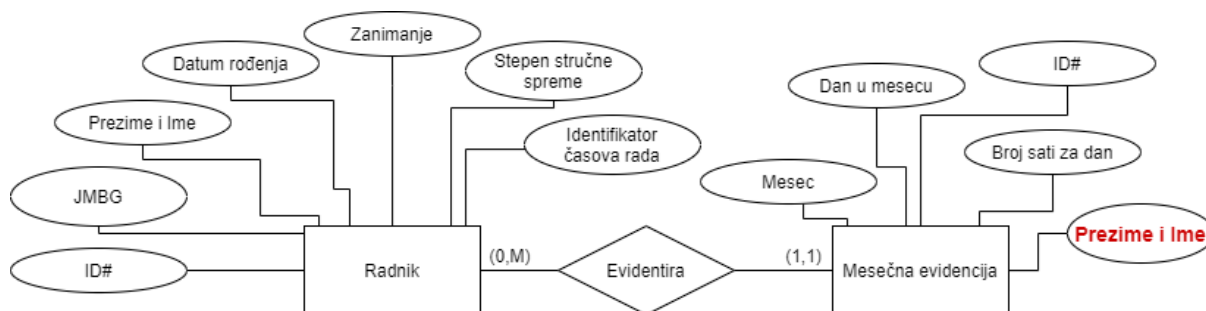
IdMesecnaEvidencija -> Mesec

Tabela 1 - Specifikacija trigera 2NF – Pre-joining

Tabela	Tip trigera	Kolona	Potreb an	Šta treba da uradi?
MesecnaEvidencija	Insert		NE	
	Update	Mesec	DA	Prilikom izmene vrednosti kolone Mesec u tabeli MesecnaEvidencija , pokreće se triger koji izmenjenu vrednost ažurira u tabeli OstvareniCasovi .
	Delete		NE	
OstvareniCasovi	Insert		DA	Triger ažurira vrednosti kolona Mesec i Naziv , na osnovu unetih vrednosti kolona IdVrstaPrimanja i IdMesecnaEvidencija .
	Update	IdVrstaPrimanja	DA	Triger vrši ažuriranje kolone Naziv na osnovu izmenjene vrednosti kolone IdVrstaPrimanja .
		Naziv	DA	Zabraniti direktno ažuriranje ove kolone.
		IdMesecnaEvidencija	DA	Triger vrši ažuriranje kolone Mesec na osnovu izmenjene vrednosti kolone IdMesecnaEvidencija .
		Mesec	DA	Zabraniti direktno ažuriranje ove kolone.
	Delete		NE	
VrstaPrimanja	Insert		NE	
	Update	Naziv	DA	Prilikom izmene vrednosti kolone Naziv u tabeli VrstaPrimanja , pokreće se triger koji izmenjenu vrednost ažurira u tabeli OstvareniCasovi .
	Delete		NE	

Denormalizacija relacija uz narušavanje 3NF

Pre-joining



Radnik (**IdRadnik**, JMBG, PrezimeIme, DatumRodjenja, Zanimanje, StepenStrucneSpreme, IdentifikatorCasovaRada)

MesecnaEvidencija(**IdMesecnaEvidencija**, Mesec, DanUMesecu, BrojSatiZaDan, *IdRadnik*, **PrezimeIme**)

Funkcionalne zavisnosti:

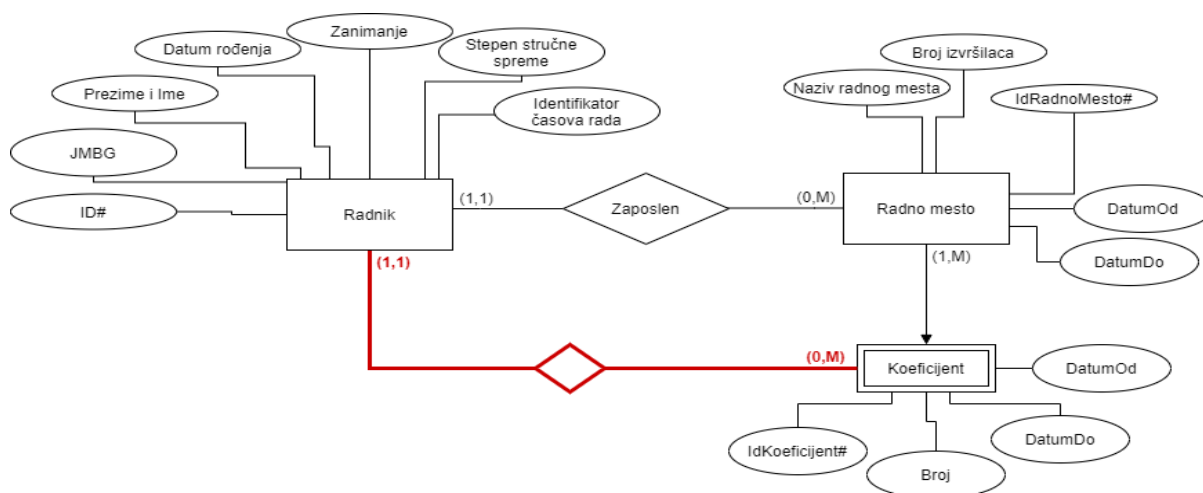
IdMesecnaEvidencija -> Mesec, DanUMesecu, BrojSatiZaDan, IdRadnik, **PrezimeIme**

IdRadnik -> **PrezimeIme**

Tabela 2 - Specifikacija trigera 3NF – Pre-Joining

Tabela	Tip trigera	Kolona	Potreban	Šta treba da uradi?
Radnik	Insert		NE	
	Update	PrezimeIme	DA	Prilikom izmene vrednosti kolone PrezimeIme u tabeli Radnik , pokreće se triger koji izmenjenu vrednost ažurira u tabeli MesecnaEvidencija .
	Delete		NE	
MesecnaEvidencija	Insert		DA	Triger ažurira vrednost kolone PrezimeIme na osnovu unete vrednosti kolone IdRadnik .
	Update	IdRadnik	DA	Triger vrši ažuriranje kolone PrezimeIme na osnovu izmenjene vrednosti kolone IdRadnik .
		PrezimeIme	DA	Zabraniti direktno ažuriranje ove kolone.
	Delete		NE	

Short-circuit keys



Radnik (**IdRadnik**, JMBG, PrezimeIme, DatumRodjenja, Zanimanje, StepenStrucneSpreme, IdentifikatorCasovaRada, **IdRadnoMesto**, **IdKoeficijent**)

RadnoMesto(**IdRadnoMesto**, Naziv, BrojIzvršilaca, DatumOd, DatumDo, **IdKoeficijent**)

Koeficijent(**IdKoeficijent**, Broj, DatumOd, DatumDo)

Funkcionalne zavisnosti:

IdRadnik -> JMBG, PrezimeIme, DatumRodjenja, Zanimanje, StepenStrucneSpreme, IdentifikatorCasovaRada, **IdKoeficijent**

IdRadnoMesto -> **IdKoeficijent**

Tabela 3 - Specifikacija triggera 3NF – Short-Circuit keys

Tabela	Tip triggera	Kolona	Potreban	Šta treba da uradi?
RadnoMesto	Insert		NE	
	Update	IdKoeficijent	DA	Ukoliko postoji Radnik za RadnoMesto, ažurira IdKoeficijent u tabeli Radnik.
	Delete		NE	
Radnik	Insert		DA	U kolonu IdKoeficijent upisuje odgovarajuću vrednost za unetu vrednost IdRadnoMesto.
	Update	IdRadnoMesto	DA	U kolonu IdKoeficijent upisuje odgovarajuću vrednost na osnovu nove vrednosti za IdRadnoMesto.
		IdKoeficijent	DA	Sprečava direktnu izmenu.
	Delete		NE	

KORISNIČKI DEFINISANI TIPOVI I TRIGERI

Definicija korisničkom tipa i njegovo korišćenje

Distinkt tip:

1. **CREATE** OR **REPLACE** TYPE SS_STEPEN **AS** OBJECT (stepen VARCHAR2(100)) FINAL;

Strukturirani tip:

1. **CREATE** OR **REPLACE** TYPE PI_RADNIK **AS** OBJECT (prezime VARCHAR2(50), ime VARCHAR2(50),
2. MEMBER **FUNCTION** get_prezime **RETURN** VARCHAR2,
3. MEMBER **FUNCTION** get_ime **RETURN** VARCHAR2)
4. INSTANTIABLE NOT FINAL;
- 5.
6. **CREATE** OR **REPLACE** TYPE BODY PI_RADNIK **AS**
7. MEMBER **FUNCTION** get_prezime **RETURN** VARCHAR2 **IS**
8. **BEGIN**
9. RETURN SELF.prezime;
10. **END**;
11. MEMBER **FUNCTION** get_ime **RETURN** VARCHAR2 **IS**
12. **BEGIN**
13. RETURN SELF.ime;
14. **END**;
15. **END**;

Za potrebe testiranja **Distinkt** i **Strukturiranog** tipa kreirana je kopija tabele RADNIK:

1. **CREATE TABLE** RADNIK_TYPES (ID_RADNIK **INTEGER**, JMBG NUMBER, PREZIME_IME PI_RADNIK, DATUM_RODZENJA **DATE**, ZANIMANJE VARCHAR2(100), STEPEN_STRUCNE_SPREME SS_STEPEN, IDENTIFIKATOR_CASOVA_RADA NUMBER, ID_KOEFICIJENT NUMBER, ID_RADNO_MESTO NUMBER);

Insertovanje novih vrednosti u tabelu RADNIK_COPY uz upotrebu korisnički definisanih tipova:

1. **INSERT INTO** RADNIK_TYPES **VALUES** (1, 2303003444555, PI_RADNIK('Sonja', 'Savic'), '24-MAR-03', 'Dipl. pravnik', SS_STEPEN('7.1a'), 2, 1, 2);
2. **INSERT INTO** RADNIK_TYPES **VALUES** (2, 2304003444555, PI_RADNIK('Mika', 'Mikic'), '24-APR-03', 'Dipl. ekonomista', SS_STEPEN('7.1b'), 3, 2, 3);
3. **INSERT INTO** RADNIK_TYPES **VALUES** (3, 2305003444555, PI_RADNIK('Pera', 'Peric'), '24-MAY-03', 'Dipl. informaticar', SS_STEPEN('7.1a'), 4, 3, 4);

Prikaz podataka upotrebom get metoda korisnički definisanih tipova:

1. **SELECT** r.id_radnik, r.jmbg, r.prezime_ime.get_prezime() **"Prezime"**, r.prezime_ime.get_ime() **"Ime"**, r.datum_rodjenja, r.zanimanje,
2. r.stepen_strucne_spreme.stepen **"Stepen"**, r.identifikator_casova_rada, r.id_koeficijent, r.id_radno_mesto **from** radnik_types r;

RADNIK_COPY – SELECT:

ID_RADNIK	JMBG	Prezime	Ime	DATUM_RODZENJA	ZANIMANJE	Stepen	IDENTIFIKATOR_CASOVA_RADA	ID_KOEFICIJENT	ID_RADNO_MESTO
1	2303003444555	Sonja	Savic	24-MAR-03	Dipl. pravnik	7.1a	2	1	2
2	2304003444555	Mika	Mikic	24-APR-03	Dipl. ekonomista	7.1b	3	2	3
3	2305003444555	Pera	Peric	24-MAY-03	Dipl. informaticar	7.1a	4	3	4

Realizacija proceduralne logike uz pomoć trigera

MesecnaEvidencija(IdMesecnaEvidencija, Mesec, DanUMesecu, BrojSatiZaDan, IdRadnik)

OstvareniCasovi(IdVrstaPrimanja, IdMesecnaEvidencija, UkupnoCasova, **Naziv**, **Mesec**)

VrstaPrimanja(IdVrstaPrimanja, Naziv, *IdKlasifikacijaCasova*)

MESECNA_EVIDENCIJA update MESEC – ažuriranje u tabeli OSTVARENI_CASOVI:

```
1. CREATE OR REPLACE TRIGGER UM_MESECNA_EVIDENCIJA
2. AFTER UPDATE OF MESEC ON MESECNA_EVIDENCIJA
3. FOR EACH ROW
4. DECLARE
5.     pragma AUTONOMOUS_TRANSACTION;
6.     v_novi_mesec VARCHAR2(9);
7.     v_id_mesecne_evidencije NUMBER;
8. BEGIN
9.     v_novi_mesec := :new.MESEC;
10.    v_id_mesecne_evidencije := :new.ID_MESECNA_EVIDENCIJA;
11.    EXECUTE IMMEDIATE 'ALTER TRIGGER UM_OSTVARENI_CASOVI DISABLE';
12.    UPDATE OSTVARENI_CASOVI
13.    SET MESEC = v_novi_mesec
14.    WHERE ID_MESECNA_EVIDENCIJA = v_id_mesecne_evidencije;
15.    EXECUTE IMMEDIATE 'ALTER TRIGGER UM_OSTVARENI_CASOVI ENABLE';
16. END;
```

OSTVARENI_CASOVI insert – ažuriranje MESEC i NAZIV na osnovu ID_VP/ME vrednosti:

```
1. create or replace TRIGGER BI_OSTVARENI_CASOVI
2. BEFORE INSERT ON OSTVARENI_CASOVI
3. FOR EACH ROW
4. DECLARE
5.     v_mesec VARCHAR2(9);
6.     v_naziv VARCHAR2(100);
7. BEGIN
8.     SELECT MESEC INTO v_mesec
9.     FROM MESECNA_EVIDENCIJA
10.    WHERE id_mesecna_evidencija = :NEW.id_mesecna_evidencija;
11.    :NEW.MESEC := v_mesec;
12.
13.    SELECT NAZIV INTO v_naziv
14.    FROM vrsta_primanja
15.    WHERE id_vrsta_primanja = :NEW.id_vrsta_primanja;
16.    :NEW.NAZIV := v_naziv;
17. END;
```

OSTVARENI_CASOVI update – NAZIV na osnovu ID_VRSTA_PRIMANJA:

```
1. create or replace TRIGGER UIVP_OSTVARENI_CASOVI
2. BEFORE UPDATE OF ID_VRSTA_PRIMANJA ON OSTVARENI_CASOVI
3. FOR EACH ROW
4. DECLARE
5.     pragma AUTONOMOUS_TRANSACTION;
6.     v_naziv VARCHAR2(100);
7. BEGIN
8.     SELECT NAZIV INTO v_naziv
9.     FROM vrsta_primanja
10.    WHERE id_vrsta_primanja = :NEW.id_vrsta_primanja;
11.    :NEW.NAZIV := v_naziv;
12. END;
```

OSTVARENI_CASOVI zabrana direktne izmene kolone NAZIV:

```
1. create or replace TRIGGER UN_OSTVARENI_CASOVI
```

```

2. BEFORE UPDATE OF NAZIV ON OSTVARENI_CASOVI
3. FOR EACH ROW
4. BEGIN
5.     RAISE_APPLICATION_ERROR(20000, 'Zabranjeno direktno ažuriranje naziva u tabeli
6.     ostvareni casovi!');
7. END;

```

OSTVARENI_CASOVI update – MESEC na osnovu ID_MESECNA_EVIDENCIJA:

```

1. create or replace TRIGGER UIME_OSTVARENI_CASOVI
2. BEFORE UPDATE OF ID_MESECNA_EVIDENCIJA ON OSTVARENI_CASOVI
3. FOR EACH ROW
4. DECLARE
5.     pragma AUTONOMOUS_TRANSACTION;
6.     v_mesec VARCHAR2(9);
7. BEGIN
8.     SELECT MESEC INTO v_mesec
9.     FROM MESECNA_EVIDENCIJA
10.    WHERE id_mesecna_evidencija = :NEW.id_mesecna_evidencija;
11.    :NEW.MESEC := v_mesec;
12. END;

```

OSTVARENI_CASOVI zabrana direktne izmene kolone MESEC:

```

1. create or replace TRIGGER UM_OSTVARENI_CASOVI
2. BEFORE UPDATE OF MESEC ON OSTVARENI_CASOVI
3. FOR EACH ROW
4. BEGIN
5.     RAISE_APPLICATION_ERROR(20000, 'Zabranjeno direktno ažuriranje meseca u tabeli
6.     ostvareni casovi!');
7. END;

```

VRSTA_PRIMANJA update – NAZIV ažuriranje u tabeli OSTVARENI_CASOVI:

```
1. create or replace TRIGGER UN_VRSTA_PRIMANJA
2. BEFORE UPDATE OF NAZIV ON VRSTA_PRIMANJA
3. FOR EACH ROW
4. DECLARE
5.     pragma AUTONOMOUS_TRANSACTION;
6.     v_naziv VARCHAR2(100);
7.     v_id_vrsta_primanja NUMBER;
8. BEGIN
9.     v_naziv := :new.NAZIV;
10.    v_id_vrsta_primanja := :new.ID_VRSTA_PRIMANJA;
11.    EXECUTE IMMEDIATE 'ALTER TRIGGER UN_OSTVARENI_CASOVI DISABLE';
12.    UPDATE OSTVARENI_CASOVI
13.    SET NAZIV = v_naziv
14.    WHERE ID_VRSTA_PRIMANJA = v_id_vrsta_primanja;
15.    EXECUTE IMMEDIATE 'ALTER TRIGGER UN_OSTVARENI_CASOVI ENABLE';
16. END;
```

Radnik (**IdRadnik**, JMBG, PrezimeIme, DatumRodjenja, Zanimanje, StepenStrucneSpreme, IdentifikatorCasovaRada)

MesecnaEvidencija(**IdMesecnaEvidencija**, Mesec, DanUMesecu, BrojSatiZaDan, *IdRadnik*, **PrezimeIme**)

RADNIK update – PREZIME_IME ažuriranje u tabeli MESECNA_EVIDENCIJA:

```
1. create or replace TRIGGER UPI_RADNIK
2. BEFORE UPDATE OF PREZIME_IME ON RADNIK
3. FOR EACH ROW
4. DECLARE
5.     pragma AUTONOMOUS_TRANSACTION;
6.     v_prezimeIme VARCHAR2(100);
7.     v_id_radnik NUMBER;
8. BEGIN
9.     v_prezimeIme := :new.PREZIME_IME;
10.    v_id_radnik := :new.ID_RADNIK;
11.    EXECUTE IMMEDIATE 'ALTER TRIGGER UPI_MESECNA_EVIDENCIJA DISABLE';
12.    UPDATE MESECNA_EVIDENCIJA
13.    SET PREZIME_IME = v_prezimeIme
14.    WHERE ID_RADNIK = v_id_radnik;
15.    EXECUTE IMMEDIATE 'ALTER TRIGGER UPI_MESECNA_EVIDENCIJA ENABLE';
16. END;
```

MESECNA_EVIDENCIJA insert – ažuriranje PREZIME_IME na osnovu ID_RADNIK vrednosti:

```
1. create or replace TRIGGER BI_MESECNA_EVIDENCIJA
2. BEFORE INSERT ON MESECNA_EVIDENCIJA
3. FOR EACH ROW
4. DECLARE
5.     v_prezimeIme VARCHAR2(100);
6. BEGIN
7.     SELECT prezime_ime INTO v_prezimeIme
8.     FROM RADNIK
9.     WHERE id_radnik = :NEW.id_radnik;
10.    :NEW.prezime_ime := v_prezimeIme;
11.
12. END;
```

MESECNA_EVIDENCIJA update – ID_RADNIK ažuriranje PREZIME_IME na osnovu ID_RADNIK vrednosti:

```
1. create or replace TRIGGER UIR_MESECNA_EVIDENCIJA
2. BEFORE UPDATE OF ID_RADNIK ON MESECNA_EVIDENCIJA
3. FOR EACH ROW
4. DECLARE
5.     pragma AUTONOMOUS_TRANSACTION;
6.     v_prezimeIme VARCHAR2(100);
7. BEGIN
8.     SELECT PREZIME_IME INTO v_prezimeIme
9.     FROM RADNIK
10.    WHERE id_radnik = :NEW.id_radnik;
11.    :NEW.PREZIME_IME := v_prezimeIme;
12. END;
```

MESECNA_EVIDENCIJA zabrana direktne izmene kolone PREZIME_IME:

```
1. create or replace TRIGGER UPI_MESECNA_EVIDENCIJA
2. BEFORE UPDATE OF PREZIME_IME ON MESECNA_EVIDENCIJA
3. FOR EACH ROW
4. BEGIN
5.     RAISE_APPLICATION_ERROR(-
6.     20000, 'Zabranjeno direktno ažuriranje prezimena i imena u tabeli ostvareni casovi!
7.     ');
8. END;
```

Radnik (**IdRadnik**, JMBG, PrezimeIme, DatumRodjenja, Zanimanje, StepenStrucneSpreme, IdentifikatorCasovaRada, *IdRadnoMesto*, **IdKoeficijent**)

RadnoMesto(**IdRadnoMesto**, Naziv, BrojIzvršilaca, DatumOd, DatumDo, *IdKoeficijent*, *IdRadnik*)

Koeficijent(**IdKoeficijent**, Broj, DatumOd, DatumDo)

RADNO_MESTO update – ID_KOEFICIJENT ažuriranje ID_KOEFICIJENT u tabeli RADNIK:

```
1. create or replace TRIGGER UIK_RADNO_MESTO
2. AFTER UPDATE OF ID_KOEFICIJENT ON RADNO_MESTO
3. FOR EACH ROW
4. DECLARE
5.     PRAGMA AUTONOMOUS_TRANSACTION;
6. BEGIN
7.     EXECUTE IMMEDIATE 'ALTER TRIGGER UIK_RADNIK DISABLE';
8.     UPDATE RADNIK
9.     SET ID_KOEFICIJENT = :NEW.ID_KOEFICIJENT
10.    WHERE ID_RADNO_MESTO = :NEW.ID_RADNO_MESTO;
11.     EXECUTE IMMEDIATE 'ALTER TRIGGER UIK_RADNIK ENABLE';
12. END;
```


RADNIK insert – ažuriranje ID_KOEFICIJENT na osnovu unete vrednosti ID_RADNO_MESTO:

```
1. create or replace TRIGGER BI_RADNIK
2. BEFORE INSERT ON RADNIK
3. FOR EACH ROW
4. DECLARE
5.     v_id_koeficijent NUMBER(38);
6. BEGIN
7.     SELECT ID_KOEFICIJENT INTO v_id_koeficijent
8.     FROM RADNO_MESTO
9.     WHERE ID_RADNO_MESTO = :new.ID_RADNO_MESTO;
10.    :NEW.ID_KOEFICIJENT := v_id_koeficijent;
11. END;
```

RADNIK update – ID_RADNO_MESTO ažuriranje ID_KOEFICIJENT na osnovu ID_RADNO_MESTO vrednosti:

```
1. CREATE OR REPLACE TRIGGER UIRM_RADNIK
2. BEFORE UPDATE OF ID_RADNO_MESTO ON RADNIK
3. FOR EACH ROW
4. DECLARE
5.     pragma AUTONOMOUS_TRANSACTION;
6.     v_id_koeficijent NUMBER;
7. BEGIN
8.     SELECT ID_KOEFICIJENT INTO v_id_koeficijent
9.     FROM RADNO_MESTO
10.    WHERE ID_RADNO_MESTO = :NEW.ID_RADNO_MESTO;
11.    :NEW.ID_KOEFICIJENT := v_id_koeficijent;
12. END;
```

RADNIK update zabrana direktne izmene kolone ID_KOEFICIJENT:

```
1. create or replace TRIGGER UIK_RADNIK
2. BEFORE UPDATE OF ID_KOEFICIJENT ON RADNIK
3. FOR EACH ROW
4. BEGIN
5.     RAISE_APPLICATION_ERROR(-20000, 'Zabranjeno direktno ažuriranje!');
6. END;
```

STATEMENT TRIGGER – broj sati za mesec AVGUST – tabela UKUPNO_SATI_AVGUST:

```
1. CREATE OR REPLACE TRIGGER ST_BRSATI_MESEVI
2. AFTER INSERT OR UPDATE ON MESECNA_EVIDENCIJA
3. DECLARE
4.     br_sati number:=0;
5. BEGIN
6.     select sum(broj_sati_za_dan) into br_sati from mesecna_evidencija where mesec L
   IKE 'Avgust';
7.     insert into ukupno_sati_avgust values(br_sati, SYSDATE);
8. END;
```

OPTIMIZACIJA BAZE PODATAKA

Definisanje i analiza indeksa

B-TREE indeks:

Ubrzava **SELECT** upite nad kolonama pristupajući kolonama koje su obuhvaćene indeksom. Omogućava bržu pretragu prezimena i imena radnika. Za potrebe bržeg pristupa koloni **PREZIME_IME** kreiran je sledeći indeks:

1. **CREATE INDEX** prezime_ime_radnik **on** radnik(prezime_ime);

Pretraga po indeksu koji je kreiran se vrši na sledeći način:

1. **select** /*+ **INDEX**(r prezime_ime_radnik)*/prezime_ime **from** radnik r **where** prezime_ime like '%Mikic%';

Testiranje indeksirane kolone I kolone koja nije indeksirana upotrebom funkcije “*explain plan for*”:

Indeksirana kolona – B-TREE:

```
56 explain plan for select /*+ INDEX(r prezime_ime_radnik)*/prezime_ime from radnik r where prezime_ime like '%Pera%';
57 select * from table(dbms_xplan.display)
```

Script Output x Query Result x

SQL | All Rows Fetched: 13 in 0.027 seconds

PLAN_TABLE_OUTPUT									
1	Plan hash value: 3499286085								
2									
3	-----								
4	Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time		
5	-----								
6	0	SELECT STATEMENT		1	12	1 (0)	00:00:01		
7	* 1	INDEX FULL SCAN	PREZIME_IME_RADNIK	1	12	1 (0)	00:00:01		
8	-----								
9									
10	Predicate Information (identified by operation id):								
11	-----								
12									
13	1	- filter("PREZIME_IME" LIKE '%Pera%')							

Kolona bez indeksa:

```
59 explain plan for select mesec from mesecna_evidencija where mesec like 'Avgust';
60 select * from table(dbms_xplan.display)
```

Script Output x Query Result x

SQL | All Rows Fetched: 13 in 0.023 seconds

PLAN_TABLE_OUTPUT									
1	Plan hash value: 3621261425								
2									
3	-----								
4	Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time		
5	-----								
6	0	SELECT STATEMENT		2	16	3 (0)	00:00:01		
7	* 1	TABLE ACCESS FULL	MESECNA_EVIDENCIJA	2	16	3 (0)	00:00:01		
8	-----								
9									
10	Predicate Information (identified by operation id):								
11	-----								
12									
13	1	- filter("MESEC"='Avgust')							

FUNCTION-BASED indeks:

Ovaj metod skeniranja zasnovan je na funkcijama, omogućava brži pristup vrednosti indeksa.

```
1. CREATE INDEX vp_naziv ON vrsta_primanja(lower(naziv));
```

Pretraga po indeksu koji je kreiran se vrši na sledeći način:

```
1. SELECT /*+ INDEX(vrsta_primanja vp_naziv) */ naziv from vrsta_primanja where lower(naziv) like '%rad%';
```

Testiranje indeksirane kolone I kolone koja nije indeksirana upotrebom funkcije “*explain plan for*”:

Indeksirana kolona – Function-based:

```
69 EXPLAIN PLAN FOR SELECT /*+ INDEX(vrsta_primanja vp_naziv) */ naziv from vrsta_primanja where lower(naziv) like 'rad';
70 select * from table(dbms_xplan.display)
```

Script Output x Query Result x

SQL | All Rows Fetched: 14 in 0.049 seconds

PLAN_TABLE_OUTPUT							
1 Plan hash value: 1757453693							
2							
3							

4	Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time

6	0	SELECT STATEMENT		1	22	2 (0)	00:00:01
7	1	TABLE ACCESS BY INDEX ROWID BATCHED	VRSTA_PRIMANJA	1	22	2 (0)	00:00:01
8	* 2	INDEX RANGE SCAN	VP_NAZIV	1		1 (0)	00:00:01

11 Predicate Information (identified by operation id):							
12 -----							
13							
14 2 - access(LOWER("NAZIV")='rad')							

Kolona bez indeksa:

```
72 explain plan for select mesec from mesecna_evidencija where lower(mesec) like 'Avgust';
73 select * from table(dbms_xplan.display)
```

Script Output x Query Result x

SQL | All Rows Fetched: 13 in 0.023 seconds

PLAN_TABLE_OUTPUT							
1 Plan hash value: 3621261425							
2							
3							

4	Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time

6	0	SELECT STATEMENT		1	8	3 (0)	00:00:01
7	* 1	TABLE ACCESS FULL	MESECNA_EVIDENCIJA	1	8	3 (0)	00:00:01

10 Predicate Information (identified by operation id):							
11 -----							
12							
13 1 - filter(LOWER("MESEC")='Avgust')							

Horizontalno particionisanje

Usled potencijalne sistematizacije i otvaranja novih radnih mesta javlja se potreba za podelom koeficijenata u nekoliko kategorija. Za potrebe testiranja trenutnog stanja sistema kreirane su dve particije, modifikacijom tabele **KOEFICIJENT**. **KP_OSTALI** particija se odnosi na koeficijente niže od **5.7**, a particija **KP_RUKOVODIOCI** se odnosi na koeficijente u opsegu od **5.7** do **10.0**.

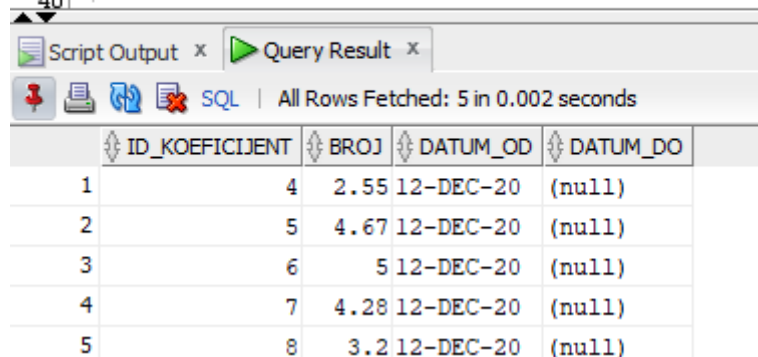
```
1. ALTER TABLE KOEFICIJENT
2. MODIFY PARTITION BY RANGE(broj)
3. (PARTITION KP_OSTALI VALUES LESS THAN(5.7),
4. PARTITION KP_RUKOVODIOCI VALUES LESS THAN(10.0)
5. );
```

Testiranje vrednosti iz opsega:

Do 5.7:

```
1. -- TESTIRANJE PARTICIJE KP_OSTALI --
2. select * from KOEFICIJENT PARTITION(KP_OSTALI);
```

```
38 | -- TESTIRANJE PARTICIJE KP_OSTALI --
39 | select * from KOEFICIJENT PARTITION(KP_OSTALI);
40 |
```



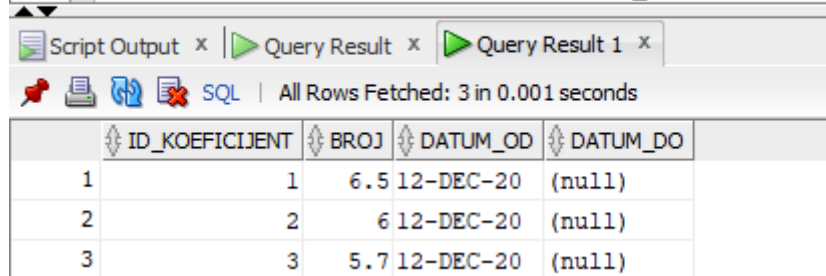
The screenshot shows a SQL query result window with a table containing 5 rows. The columns are ID_KOEFICIJENT, BROJ, DATUM_OD, and DATUM_DO. The data is as follows:

ID_KOEFICIJENT	BROJ	DATUM_OD	DATUM_DO
1	4	2.55 12-DEC-20	(null)
2	5	4.67 12-DEC-20	(null)
3	6	5 12-DEC-20	(null)
4	7	4.28 12-DEC-20	(null)
5	8	3.2 12-DEC-20	(null)

Od 5.7 do 10.0:

```
1. -- TESTIRANJE PARTICIJE KP_RUKOVODIOCI --
2. select * from KOEFICIJENT PARTITION(KP_RUKOVODIOCI);
```

```
41 | -- TESTIRANJE PARTICIJE KP_RUKOVODIOCI --
42 | select * from KOEFICIJENT PARTITION(KP_RUKOVODIOCI);
```



The screenshot shows a SQL query result window with a table containing 3 rows. The columns are ID_KOEFICIJENT, BROJ, DATUM_OD, and DATUM_DO. The data is as follows:

ID_KOEFICIJENT	BROJ	DATUM_OD	DATUM_DO
1	1	6.5 12-DEC-20	(null)
2	2	6 12-DEC-20	(null)
3	3	5.7 12-DEC-20	(null)

Vertikalno partitionisanje

Pre vertikalnog partitionisanja:

RADNIK (**ID_RADNIK**, JMBG, PREZIME_IME, DATUM_RODJENJA, ZANIMANJE, STEPEN_STRUCNE_SPREME, IDENTIFIKATOR_CASOVA_RADA, ID_KOEFICIJENT, ID_RADNO_MESTO)

Za potrebe vertikalnog partitionisanja kreirane su dve nove tabele – **RADNIK_COPY** i **RADNIK_COPY_DET**:

RADNIK_COPY(**ID_RADNIK**, PREZIME_IME)

RADNIK_COPY_DET(**ID_RADNIK**, JMBG, DATUM_RODJENJA, ZANIMANJE, STEPEN_STRUCNE_SPREME, IDENTIFIKATOR_CASOVA_RADA, ID_KOEFICIJENT, ID_RADNO_MESTO)

Kreiran je pogled(**VIEW**) kojim su objedinjene informacije iz dve tabele **RADNIK_COPY** i **RADNIK_COPY_DET**, tako da kreiraju strukturu polazne relacije **RADNIK**.

```
1. CREATE OR REPLACE VIEW SVE_O_RADNIKU AS
2. SELECT rc.id_radnik, rc.prezime_ime, rcd.jmbg, rcd.datum_rodjenja, rcd.zanimanje, rcd.stepen_strucne_spreme, rcd.identifikator_casova_rada, rcd.id_koeficijent, rcd.id_radno_mesto
3. FROM RADNIK_COPY rc, RADNIK_COPY_DET rcd
4. WHERE rc.id_radnik = rcd.id_radnik;
```

Testiranje pogleda:

```
1. -- SELECT NAD POGLEDOM
2. SELECT * FROM SVE_O_RADNIKU;
```

INSTEAD OF TRIGGER - pogled_radnik_copy insert – pokreće se prilikom dodavanja novog radnika u pogled:

Ovim triggerom je obezbeđeno insertovanje podataka u obe tabele koje su objedinjene pogledom.

```
1. CREATE OR REPLACE TRIGGER pogled_radnik_copy
2. INSTEAD OF INSERT ON sve_o_radniku
3. REFERENCING NEW AS NOVI
4. FOR EACH ROW
5. BEGIN
6. INSERT INTO radnik_copy(id_radnik, prezime_ime)
7. VALUES (:NOVI.id_radnik, :NOVI.prezime_ime);
8. INSERT INTO radnik_copy_det(id_radnik, jmbg, prezime_ime, datum_rodjenja, zanimanje, stepen_strucne_spreme, identifikator_casova_rada, id_koeficijent, id_radno_mesto)
9. VALUES (:NOVI.id_radnik, :NOVI.jmbg, :NOVI.prezime_ime, :NOVI.datum_rodjenja, :NOVI.zanimanje, :NOVI.stepen_strucne_spreme, :NOVI.identifikator_casova_rada, :NOVI.id_koeficijent, :NOVI.id_radno_mesto);
10. END;
```

Insertovanje nove n-torke:

```
1. insert into SVE_O_RADNIKU VALUES (7, PI_RADNIK('Zivic', 'Zivorad'), 1232654777888,
   '24-SEP-20', 'Dipl. ing.', SS_STEPEN('7'), 2, 2,2);
```

INSTEAD OF TRIGGER – *pogled_radnik_copy_upd* – pokreće se prilikom ažuriranja pogleda SVE_O_RADNIKU:

```
1. CREATE OR REPLACE TRIGGER pogled_radnik_copy_upd
2. INSTEAD OF UPDATE ON sve_o_radniku
3. REFERENCING NEW AS NOVI
4. FOR EACH ROW
5. BEGIN
6. UPDATE radnik_copy SET prezime_ime = :NOVI.prezime_ime
7. WHERE radnik_copy.id_radnik = :NOVI.id_radnik;
8. UPDATE radnik_copy_det SET jmbg = :NOVI.jmbg, datum_rodjenja= :NOVI.datum_rodjenja,
   zanimanje = :NOVI.zanimanje,
9. stepen_strucne_spreme = :NOVI.stepen_strucne_spreme, identifikator_casova_rada = :NOVI.identifikator_casova_rada,
10. id_koeficijent = :NOVI.id_koeficijent, id_radno_mesto = :NOVI.id_radno_mesto,
   prezime_ime = :NOVI.prezime_ime
11. WHERE radnik_copy_det.id_radnik = :NOVI.id_radnik;
12. END;
```

Za testiranje ažuriranja pogleda:

```
1. UPDATE SVE_O_RADNIKU SET PREZIME_IME = 'Ankic Anka' WHERE ID_RADNIK = 12;
2. SELECT * FROM SVE_O_RADNIKU;
```

INSTEAD OF TRIGGER – *pogled_radnik_copy_del* – pokreće se prilikom brisanja podataka iz pogleda SVE_O_RADNIKU:

```
1. CREATE OR REPLACE TRIGGER pogled_radnik_copy_del
2. INSTEAD OF DELETE ON sve_o_radniku
3. BEGIN
4. DELETE FROM RADNIK_COPY WHERE RADNIK_COPY.ID_RADNIK = :OLD.ID_RADNIK;
5. DELETE FROM RADNIK_COPY_DET WHERE RADNIK_COPY_DET.ID_RADNIK = :OLD.ID_RADNIK;
6. END;
```

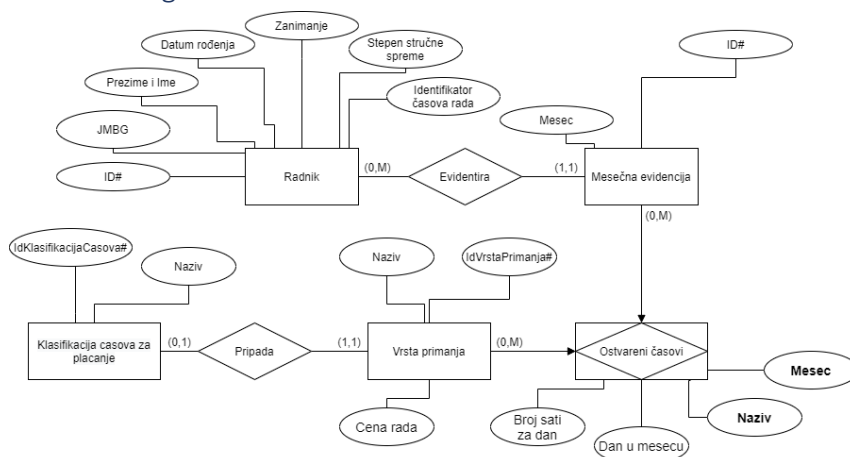
Testiranje instead of triggera za brisanje:

```
1. SELECT * FROM SVE_O_RADNIKU;
2. DELETE FROM SVE_O_RADNIKU WHERE id_radnik=3;
3. SELECT * FROM SVE_O_RADNIKU;
```

Primena drugih optimizacionih tehnika

Storing Derivable Values – tehnika izabrana za potrebe ovog informacionog sistema, za potrebe izračunavanja procenjene zarade u tabeli **MESECNA_EVIDENCIJA** što predstavlja informaciju koja je često potrebna. Kako bi se izbeglo često pisanje upita za ovu informaciju kreirana je dodatna kolona **PREDVIDJENA_ZARADA** u tabeli **MESECNA_EVIDENCIJA**.

PMOV pre primene storing derivable values tehnike:



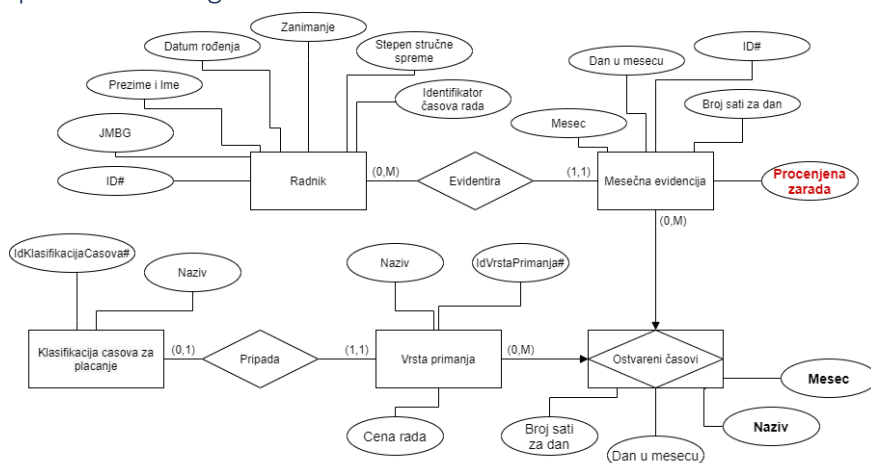
Relacioni model samo za tabele koje su neophodne:

VRSTA_PRIMANJA(ID_VRSTA_PRIMANJA, NAZIV, CENA_RADA, ID_KLASIFIKACIJA_CASOVA)

OSTVARENI_CASOVI(ID_MESECNA_EVIDENCIJA, ID_VRSTA_PRIMANJA, NAZIV, MESEC, DAN_U_MESECU, BROJ_SATI_ZA_DAN)

MESECNA_EVIDENCIJA(ID_MESECNA_EVIDENCIJA, MESEC, PREZIME_IME, ID_RADNIK)

PMOV nakon primene storing derivable values tehnike:



Relacioni model samo za tabele koje su neophodne:

VRSTA_PRIMANJA(ID_VRSTA_PRIMANJA, NAZIV, CENA_RADA, ID_KLASIFIKACIJA_CASOVA)

OSTVARENI_CASOVI(ID_MESECNA_EVIDENCIJA, ID_VRSTA_PRIMANJA, NAZIV, MESEC, DAN_U_MESECU, BROJ_SATI_ZA_DAN)

MESECNA_EVIDENCIJA(ID_MESECNA_EVIDENCIJA, MESEC, **PROCENJENA_ZARADA**, PREZIME_IME, ID_RADNIK)

Tabela	Tip trigera	Kolona	Potreban	Šta treba da uradi?
MESECNA EVIDENCIJA	Insert		NE	
	Update	PROCENJENA ZARADA	DA	Sprečava direktnu izmenu kolone PROCENJENA_ZARADA .
	Delete		NE	
OSTVARENI CASOVI	Insert		DA	Prilikom unosa novih ostvarenih časova ažurira se kolona PROCENJENA_ZARADA tabeli MESECNA_EVIDENCIJA .
	Update	BROJ SATI ZA DAN	DA	Prilikom ažuriranja novih ostvarenih časova ažurira se kolona PROCENJENA_ZARADA u tabeli MESECNA_EVIDENCIJA .
	Delete		DA	Brisanjem ostvarenih časova ažurira se kolona PROCENJENA_ZARADA u tabeli MESECNA_EVIDENCIJA .

Triger za zabranu direktne izmene kolone **PROCENJENA_ZARADA** u tabeli **MESECNA_EVIDENCIJA**:

```

1. CREATE OR REPLACE TRIGGER BU_proc_zar_mes_evi
2. BEFORE UPDATE OF PROCENJENA_ZARADA ON MESECNA_EVIDENCIJA
3. FOR EACH ROW
4. BEGIN
5.     RAISE_APPLICATION_ERROR(-
6.         20000, 'Zabranjeno ažuriranje kolone procenjena zarada u tabeli mesecna evidencija.
7.     ');
8. END;
```

Testiranje trigera:

```

1. update mesecna_evidencija set procenjena_zarada = 22000 where id_mesecna_evidencija
   = 2;
```

Za potrebe kreiranja sledećih trigera potrebno je kreirati paket sa deklarisanom promenljivom koja će se koristiti u trigerima, tj koja će im biti zajednička što se postiže deklaracijom kroz paket.

```

1. CREATE OR REPLACE PACKAGE evidencija AUTHID DEFINER AS
2.     mes_evi number;
3. end evidencija;
```


Triger koji vrednosti id_mesecna_evidencija iz tabele MESECNA_EVIDENCIJA upisuje u mes_evi varijablu kreiranog paketa evidencija, za svaki red:

```
1. CREATE OR REPLACE TRIGGER "ZBIR"
2. BEFORE INSERT OR UPDATE OR DELETE ON OSTVARENI_CASOVI
3. FOR EACH ROW
4. BEGIN
5. IF (INSERTING OR UPDATING)
6. THEN
7. BEGIN
8.     evidencija.mes_evi :=:NEW.id_mesecna_evidencija;
9. END;
10. ELSE
11. BEGIN
12.     evidencija.mes_evi :=:OLD.id_mesecna_evidencija;
13. END;
14. END IF;
15. END;
```

Procedura koja izračunava *procenjena_zarada* prema ID_MESECNA_EVIDENCIJA:

```
1. CREATE OR REPLACE PROCEDURE "ZBIR_OSTVARENIH_CASOVA" (id_mes_evi in NUMBER) AS
2. zbir NUMBER := 0;
3. PRAGMA AUTONOMOUS_TRANSACTION;
4. BEGIN
5.     SELECT SUM(oc.broj_sati_za_dan * vp.cena_rada) INTO zbir
6.     FROM ostvareni_casovi oc JOIN vrsta_primanja vp ON (oc.id_vrsta_primanja = vp.i
7.     d_vrsta_primanja)
8.     WHERE id_mesecna_evidencija = id_mes_evi;
9.     EXECUTE IMMEDIATE 'ALTER TRIGGER BU_proc_zar_mes_evi DISABLE';
10.    UPDATE MESECNA_EVIDENCIJA SET procenjena_zarada = zbir
11.    WHERE id_mesecna_evidencija = id_mes_evi;
12.    EXECUTE IMMEDIATE 'ALTER TRIGGER BU_proc_zar_mes_evi ENABLE';
13. END;
```

Triger za poziv na izvršavanje procedure:

```
1. CREATE OR REPLACE TRIGGER "ZBIR_PROC"
2. AFTER INSERT OR UPDATE OR DELETE ON OSTVARENI_CASOVI
3. DECLARE broj INTEGER := evidencija.mes_evi;
4. BEGIN
5.     ZBIR_OSTVARENIH_CASOVA(broj);
6. END;
```

Testiranje izračunavanja procenjene zarade:

```
1. update ostvareni_casovi set broj_sati_za_dan = 10, id_vrsta_primanja = 4 where id_m
   esecna_evidencija = 8;
```

TEHNOLOGIJE ZA IMPLEMENTACIJU PROJEKTA

SUBP korišćen za implementaciju baze podatka (kratak opis)

Programsko okruženje za razvoj korisničkog interfejsa (kratak opis)