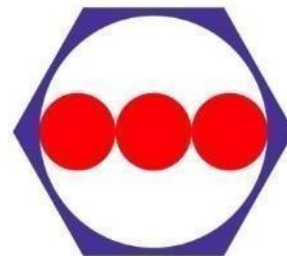




**Univerzitet u Novom Sadu**  
**FAKULTET TEHNIČKIH NAUKA**

Departman za industrijsko inženjerstvo i inženjerski menadžment  
Inženjerstvo informacionih sistema



**INFORMACIONI SISTEM ZA PODRŠKU RADA SLUŽBE**  
**ZA LJUDSKE RESURSE**  
**-SISTEMI BAZA PODATAKA-**

**Saša Ilić, IT4/2019**

**Novi Sad, 2022.**

## SADRŽAJ

UVOD .....	2
ANALIZA PROGRAMSKOG DOMENA.....	2
ER MODEL .....	4
ER MODEL PODŠEME .....	4
TABELARNI PRIKAZ OBELEŽJA I OGRANIČENJA.....	7
RELACIONI MODEL.....	12
DDL .....	13
SQL .....	15
OBJEKTI .....	16
UPITI .....	23
FUNKCIJE.....	26
PROCEDURE .....	30
TRIGERI.....	35
ZAKLJUČAK .....	38

## UVOD

HR služba čini važnu ulogu svake kompanije, budući da vodi računa o jednom od najvažnijih resursa svake kompanije - zaposlenima. S obzirom da se radi o delatnosti koja obuhvata veliki opseg aktivnosti i procesa, upotreba dobro organizovanog informacionog sistema koji će olakšati upravljanje ljudskim resursima je od ključnog značaja za svaku kompaniju koja želi da održi stabilnost i konkurentnost na tržištu. Projekat koji će biti opisan detaljnije u nastavku ove dokumentacije predstavlja razvoj baze podataka na osnovu dela isprojektovane šeme baze podataka za podršku poslovanju službe za ljudske resurse nekog odabranog realnog sistema, koja je bila kreirana na predmetu Projektovanje baza podataka. Projekat se bavi jednom od najvažnijih delova HR službe, a to su angažovanja zaposlenih. Pod angažovanjima se ne podrazumeva samo generalno angažovanje zaposlenog u kompaniji, već i individualna angažovanja svakog od zaposlenih na pojedinim projektima. Takođe, jako je bitno pratiti i angažovanja ljudi iz same HR službe prilikom organizovanja konkursa, te evidentiranja informacija o samom konkursu i o zaposlenima koji su angažovani na osnovu istog. Projekat obuhvata i aktivnost organizovanja evaluacionih testova od strane HR službe.

U nastavku dokumentacije biće prikazan ER model podataka HR službe, ER model izabrane podšeme pod nazivom Angažovanje koja će biti osnova realizacije ovog projekta. Takođe, u daljem tekstu će biti predstavljeni obeležja, ograničenja pojave tipa, integriteta entiteta i jedinstvenih vrednosti u tabelarnom formatu. Prevođenjem ER modela podšeme nastaje relacioni model podšeme o kojem će biti reči u daljem tekstu. Na osnovu relacionog modela podataka izvršena je implementacija šeme baza podataka. Prilikom izrade projekta, nad kreiranom bazom podataka su kreirani odgovarajuće sekvence, upiti, funkcije, procedure, kursori i trigeri, koji će biti priloženi projektu u obliku SQL skripti, a zatim i detaljno objašnjeni u okviru dokumentacije.

## ANALIZA PROGRAMSKOG DOMENA

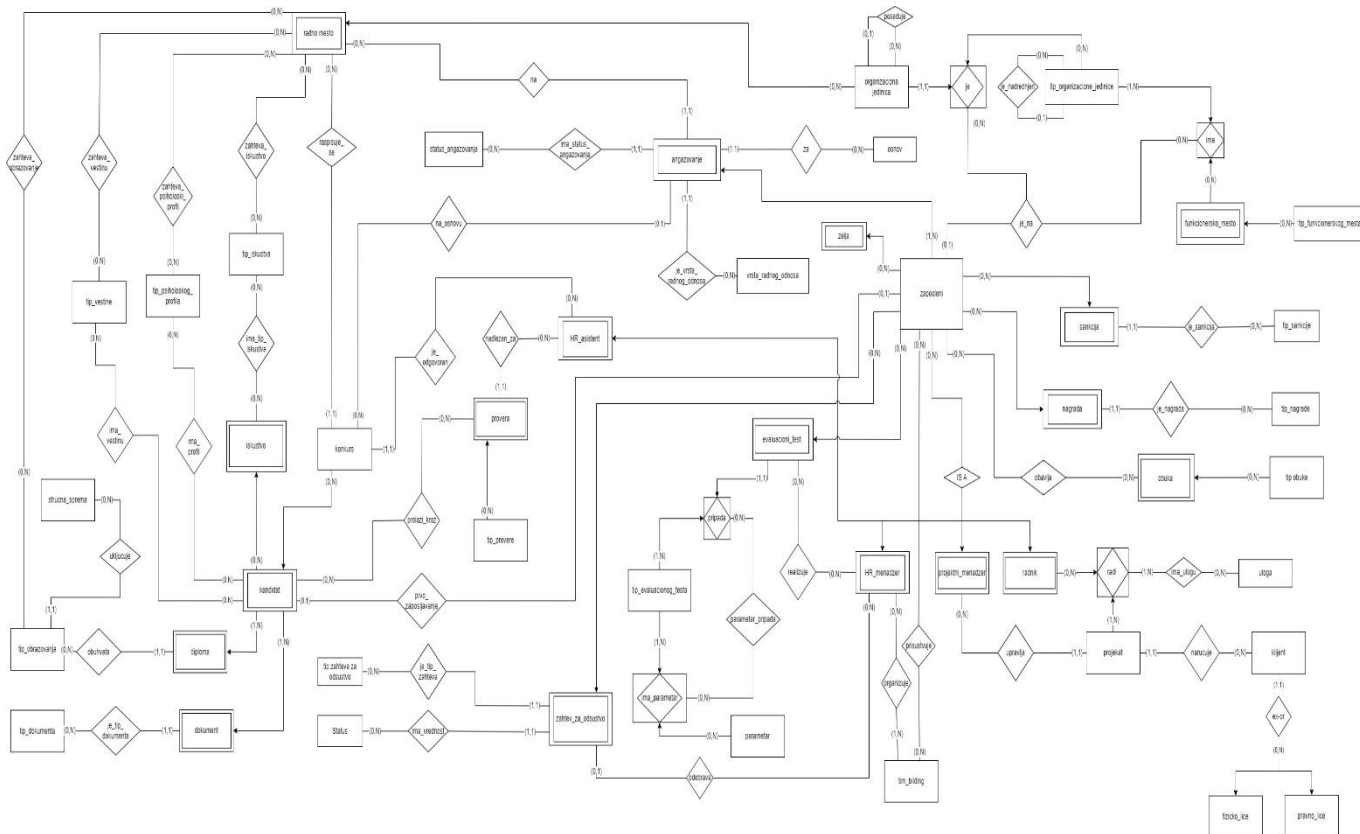
Cilj projekta, kao što je već i bilo reči u uvodnom delu, jeste olakšavanje i automatizacija rada službe za upravljanje ljudskim resursima. Značaj informacionog sistema za podršku rada službe za ljudske resurse odnosi se na povećanu efikasnost upravljanja ljudskim resursima kroz kvalitetnije metode zapošljavanja, bolju i bržu komunikaciju među zaposlenima i klijentima, kao i bržu, delotvorniju i efikasniju obradu podataka. Stoga, sa

razvojem digitalizacije i sve većeg korišćenja skladištenja i prezentovanja informacija na takav način, razvojem ovog projekta značajno se olakšava poslovanje HR službi tako što se omogućava skladištenje veće količine podataka, brz pristup podacima, a samim tim i vremensku uštedu koja je u svakom poslu od velikog značaja. Sam projekat treba da predstavlja fleksibilnu strukturu koja će se lako prilagoditi poslovanju službe za ljudske resurse bilo kog realnog sistema. S obzirom da broj zaposlenih u pojedinim kompanija premašuje cifru od nekoliko hiljada, poslovanje HR službe se ne može zamisliti bez adekvatne baze podataka koja će biti namenjena da podrži njeno funkcionisanje. Domen ovog projekta, najvećim delom se odnosi na evidentiranje angažovanja zaposlenih. Veoma je važno da se vodi evidencija o svim zaposlenim u kompaniji, kako o onima koji su trenutno angažovani, tako i onima koju se nekada bili angažovani u kompaniji.

Za svakog od zaposlenih, bilo da se radi o sadašnjem ili bivšem zaposlenom, neophodno je evidentirati podatke o njegovom radnom mestu, vrsti njegovog radnog odnosa, zatim o osnovu njegovog zaposlenja, koje može biti na osnovu konkursa organizovanih od strane HR službe, koji se takođe evidentiraju ili pak po nekom drugom osnovu. Takođe, projektom je predviđeno da se prate i evidentiraju, kao što je već i pomenuto, i drugi vidovi angažovanja zaposlenih. Podaci o aktivnostima zaposlenih na projektima, evaluacioni testovi i konkursi predstavljaju jedne od ključnih ulaznih podataka neophodnih za donošenje odluka i sprovođenje daljih aktivnosti od strane ljudi zaposlenih u HR službi. Sve te podatke je potrebno negde zabeležiti i upravo se svi ti podaci nalaze u bazi podataka. Cilj je omogućiti definiciju podataka, unos i izmenu podataka na bezbedan i konzistentan način, zatim implementirati određena ograničenja i mehanizme koji bi sprečili narušavanje konzistentnosti tih podataka.

## ER MODEL

Na slici 1 dat je prikaz kompletnog ER modela ,izmodelovanog prilikom rada na projektu iz predmeta Projektovanje baza podataka.

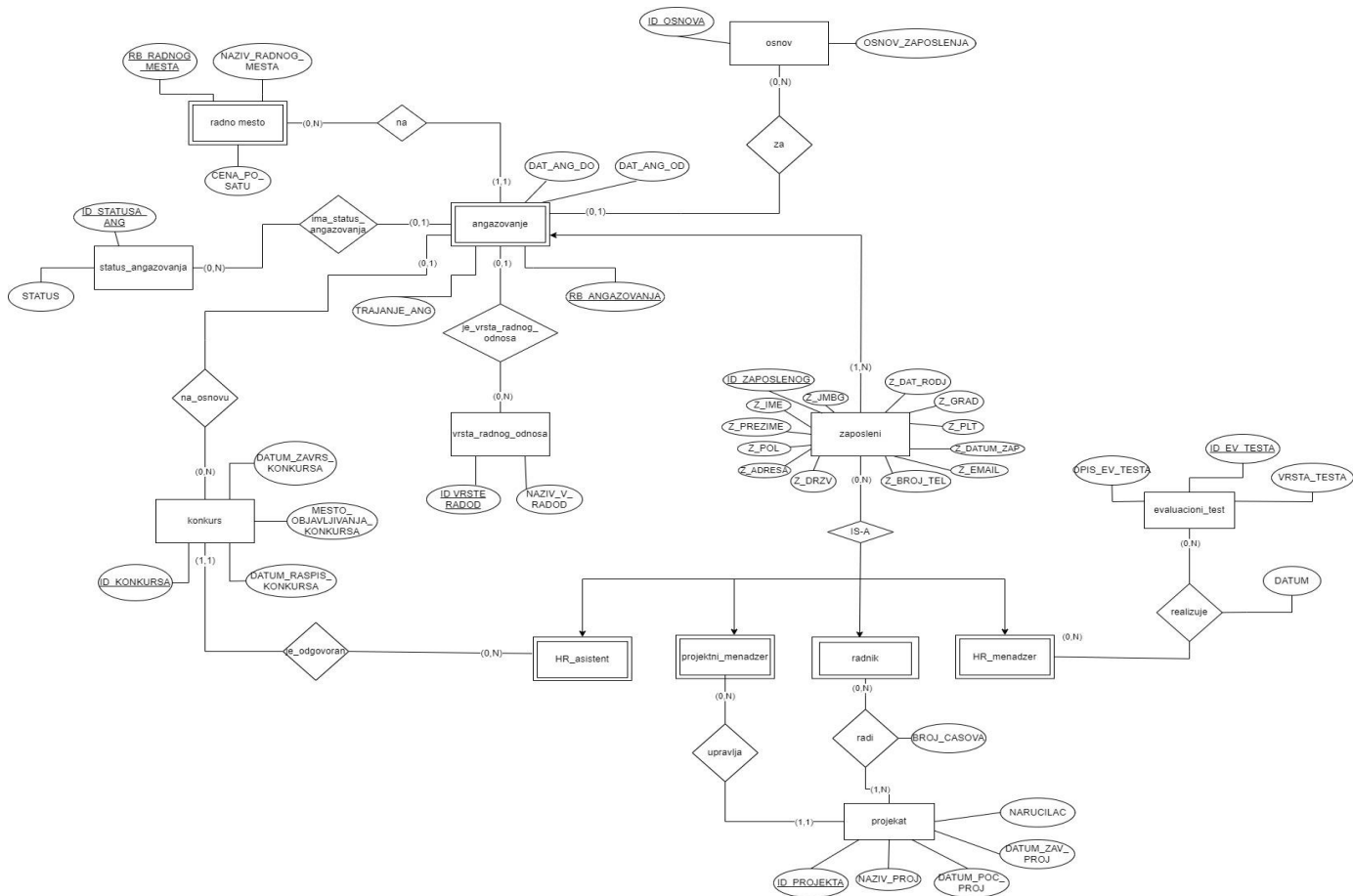


*Slika 1: Prikaz ER modela 1*

## ER MODEL PODŠEME

Na slici 2 prikazan je ER model izabrane podšeme Angažovanje. Naglasio bih da se verzija podšeme Angažovanje koja je prikazana na slici i na osnovu koje je rađen projekat i razvijena baza podataka, neznatno razlikuje od izvorne podšeme. Izmene nad podšemom su načinjene usled uočavanja sitnih nepravilnosti, a radi kreiranja optimalnog rešenja koje bi u potpunosti zadovoljavalo vođenje evidencije o angažovanjima zaposlenih. U nastavku teksta sledi objašnjenje date podšeme.

Angažovanje zaposlenog u kompaniji se modeluje tipom entiteta angazovanje. Tip entiteta angazovanje je identifikaciono zavisan od tipa entiteta zaposleni, jer se angažovanje beleži za nekog konkretnog zaposlenog i bez njega nema puno smisla i značenja. Svaki zaposleni ima može imati određene uloge u kompaniji, što se modeluje IS\_A hijerarhijom. Tip entiteta HR\_menadzer je deo IS\_A hijerarhije zaposleni. IS\_A hijerarhiji pripadaju i HR asistent, projektni menadzer i radnik. Oni su redom modelovani tipovima entiteta HR\_asistent, projektni\_menadzer i radnik. U konkretnom slučaju, radi se o parcijalnoj, presečnoj IS\_A hijerarhiji iz razloga što jedan zaposleni, primera radi, istovremeno može da bude i HR asistent i HR menadzer, odnosno jedan zaposleni može istovremeno i da rukovodi projektom, odnosno da bude projektni menadzer i da radi na njemu. Sa druge strane, IS\_A hijerarhija je parcijalna iz razloga što je ostavljena mogućnost da se pojavi neka druga, nova uloga u kompaniji a sve radi poboljšanja fleksibilnosti rešenja i njegovom prilagođavanja različitim realnim sistemima. Zaposleni može imati jedno ili više angažovanja. Na taj način je omogućeno da se višestruko zabeleže informacije o pojedinačnom zaposlenom, usled mogućnosti angažovanja nekog bivšeg zaposlenog, odnosno onog zaposlenog koji je nekada već radio u kompaniji i čije angažovanje imamo evidentirano. Na jednom radnom mestu može da se nađe više angažovanih, a ne mora ni jedan. Angažovan zaposleni se nalazi na tačno jednom radnom mestu, koje je u velikoj meri određeno njegovom ulogom u kompaniji. Angažovanje može imati najviše jedan status angažovanja, dok status angažovanja može imati više angažovanja. Status angažovanja označava da li je angažovanje aktivno, završeno ili pak pauzirano. Vrsta radnog odnosa se modeluje tipom entiteta vrsta\_radnog\_odnosa. Angažovanje može pripadati tačno jednoj vrsti radnog odnosa. Kod vrste radnog odnosa se beleži da li je na određeno, neodređeno, po nekom ugovoru. Osnov se modeluje tipom entiteta osnov. Angažovanje može imati najviše jedan osnov angažovanja, dok osnova za angažovanje može biti više. Angažovanje može biti na osnovu konkursa. S obzirom da je konkurs od velikog značaja kada se govori o ljudskim resursima, isti se modeluje tipom entiteta konkurs, radi beleženja dodatnih informacija o konkursu kao što su datum njegovog objavljivanja, datum njegovog završetka, platforma na kojoj je objavljen, kao i informacije o HR asistentu koji je bio zadužen za njegovo organizovanje. Takođe, vodi se evidencija o evaluacionim testovima, koji se modeluju tipom entiteta evaluacioni\_test. HR menadzer je zaposleni koji je zadužen za samu realizaciju evaluacionog testa i on može da realizuje više testova, a ne mora nijedan.



Slika 2: Podšema Angažovanje

## TABELARNI PRIKAZ OBELEŽJA I OGRANIČENJA

Tabela 1. Tip entiteta zaposleni

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
1.	ID_ZAPOS LENOG	Identifikacioni broj zaposlenog	INTEGER	8	⊥	$d > 0$
2.	Z_JMBG	Jedinstveni matični broj zaposlenog	VARCHAR	13	⊥	$d > 0$
3.	Z_IME	Ime zaposlenog	VARCHAR	20	⊥	$\Delta$
4.	Z_PREZIME	Prezime zaposlenog	VARCHAR	20	⊥	$\Delta$
5.	Z_POL	Pol zaposlenog	CHAR	1	T	$d = \{ 'M', 'Z' \}$
6.	Z_DAT_RODJ	Datum rođenja zaposlenog	DATE		⊥	$\Delta$
7.	Z_ADRESA	Ulica i broj adrese zaposlenog	VARCHAR	50	T	$\Delta$
8.	Z_GRAD	Grad u kojem živi zaposleni	VARCHAR	20	T	$\Delta$
9.	Z_DRZV	Država u kojoj živi zaposleni	VARCHAR	20	T	$\Delta$
10.	Z_BROJ_TEL	Broj telefona zaposlenog	VARCHAR	15	T	$\Delta$
12.	Z_DATUM_ZAP	Datum zapošljavanja	DATE		⊥	$\Delta$
13.	Z_PLT	Plata zaposlenog	NUMERIC	8	T	$\Delta$
Ključ	<b>K={ID_ZAPOSLENOG}</b>					
	Ne postoje dva zaposlena sa istim Z_JMBG					

Tabela 2. Tip entiteta HR\_menadzer

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
Ključ	<b>K={ID_HR_MENADZER}</b>					



Tabela 3. Tip entiteta HR\_asistent

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
Ključ	K={ID_HR_ASISTENTA}					

Tabela 4. Tip entiteta Projektni\_menadzer

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
Ključ	K={ID_PROJ_MENADZER}					

Tabela 5. Tip entiteta HR\_menadzer

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
Ključ	K={ID_RADNIK}					

Tabela 6. Tip entiteta projekat

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
1.	ID_PROJEKTA	Identifikacioni broj projekta	INTEGER	8	⊥	d>0
2.	NAZIV_PROJ	Naziv projekta	VARCHAR	80	⊥	Δ
3.	DAT_POC_PROJ	Datum početka projekta	DATE		⊥	Δ
4.	DAT_ZAVR_PROJ	Datum završetka projekta	DATE		T	Δ
5.	NARUCILAC	Ime ili naziv narucioca projekta	VARCHAR	30	T	Δ
Ključ	K={ID_PROJEKTA}					
DAT_POC_PROJ se navodi pre DAT_ZAVR_PROJ						

Tabela 7. Tip entiteta radi

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
1.	BROJ_CASOVA	Broj radnih casova koje je radnik ostvario radeci na projektu	NUMERIC	4	T	d>=0
Ključ	K={ID_PROJEKTA+ID_RADNIK}					

Tabela 8. Tip entiteta evaluacioni\_test

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
1.	ID_EV_TESTA	Identifikacioni broj evaluacionog testa	SMALLINT	5	⊥	d>0
2.	VRSTA_EV_TESTA	Vrsta evaluacionog testa koji se sprovodi	VARCHAR	50	⊥	Δ
3.	OPIS_EV_TESTA	Opis evaluacionog testa koji se sprovodi	VARCHAR	200	T	Δ
Ključ	K={ID_EV_TESTA}					

Tabela 9. Tip entiteta realizuje

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
1.	DATUM	Datum realizacije konkretnog evaluacionog testa	DATE		⊥	Δ
Ključ	K={ID_HR_MENADZER + ID_EV_TESTA}					

Tabela 10. Tip entiteta radno mesto

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
1.	ID_RADNOG_MESTA	Identifikacioni broj radnog mesta	INTEGER	8	⊥	d>0
2.	NAZIV_RADNOG_MESTA	Naziv radnog mesta	VARCHAR	30	⊥	Δ
3.	CENA_PO_SATU	Cena rada po satu na tom radnom mestu	NUMERIC	4	T	Δ
Ključ	K={ID_RADNOG_MESTA}					

Tabela 11. Tip entiteta status\_angazovanja

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
1.	ID_STATUSA_ANG	Identifikacioni broj statusa angazovanja	TINYINT	3	⊥	d>0
2.	NAZIV_STATUSA_ANG	Naziv statusa angazovanja	VARCHAR	30	⊥	Δ
Ključ	K={ID_STATUSA_ANG }					

Tabela 12. Tip entiteta vrsta\_radnog\_odnosa

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
1.	ID_VRSTA_RADOD	Identifikacioni broj vrste radnog odnosa	TINYINT	3	⊥	d>0
2.	NAZIV_VRSTE_RADOD	Naziv vrste radnog odnosa	VARCHAR	50	⊥	Δ
Ključ	K={ID_VRSTA_RADOD}					

Tabela 13. Tip entiteta osnov

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
1.	ID_OSNOVA	Identifikacioni broj osnova zaposlenja	TINYINT	3	⊥	d>0
2.	OSNOV_ZAPOSL ENJA	Naziv osnova zaposlenja	VARCHAR	40	⊥	Δ
Ključ	K={ID_OSNOVA}					

Tabela 14. Tip entiteta konkurs

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
1.	ID_KONKURSA	Identifikacioni broj konkursa	INTEGER	8	⊥	d>0
2.	DATUM_RASPIS_KONKURSA	Datum raspisivanja konkursa	DATE		⊥	Δ
3.	DATUM_ZAVR_KONKURSA	Datum završetka konkursa	DATE		T	Δ
4.	MESTO_OBJAVLJ IVANJA_KONKURSA	Mesto na kojem je konkurs objavljen	VARCHAR	30	T	Δ
Ključ	K={ID_KONKURS}					

Tabela 15. Tip entiteta angazovanje

Redni broj	Naziv obeležja	Opis obeležja	Tip podataka	Dužina podataka	Null	Uslov
1.	RB_ANGAZOVAN JA	Identifikacioni broj angažovanja	INTEGER	8	⊥	d>0
2.	DATUM_ANG_OD	Datum početka angažovanja	DATE		⊥	Δ
3.	DATUM_ANG_DO	Datum završetka angažovanja	DATE		T	Δ
4.	TRAJANJE_ANG	Vremensko trajanja angažovanja izraženo u mesecima	NUMERIC	5	T	Δ
Ključ	K={RB_ANGAZOVANJA}					

## RELACIONI MODEL

Relacioni model za upravljanje bazama podataka je jedna vrsta modela podataka, koji se zasniva na predikatskom računu i logici i teoriji skupova i relacija. Osnovna ideja relacionog modela je korišćenje relacija i jasno razdvajanje logičkog i fizičkog aspekta prikazivanja podataka. U razvoju je od 1970-ih godina, a njegova komercijalna upotreba počinje 1990-ih godina. Danas je jedan od najkorišćenijih modela podataka. Motiv razvoja relacionog modela podataka jeste bilo otklanjanje nedostataka klasičnih modela podataka i insistiranje na matematičkim osnovama u razvoju modela podataka. Nedostaci klasičnih modela podataka su bili čvrsta povezanost logičkih i fizičkih aspekata, strukturalna kompleksnost i korišćenje navigacionog jezika za manipulaciju sa podacima. Samim tim, prilikom razvoja relacionog modela podataka, postavljena su sledeća 3 cilja, čije ostvarivanje bi trebalo da reši probleme klasičnih modela, a to su:

1. Nezavisnost programa od podataka - Uvođenje jasne granice između logičkih i fizičkih aspekata baze podataka, kako u domenu projektovanja tako i u domenu korišćenja. Kako bi se to omogućilo, uvodi se, već gore spomenuta, relacija. Relacija je jedan od fundamentalnih matematičkih koncepata i predstavlja osnovu reprezentacije logičkih struktura podataka u relacionom modelu podataka. Relacija je skup torki. Ona ne sadrži nikakve informacije o fizičkoj organizaciji podataka. Uobičajena je tabelarna reprezentacija relacije, tako da se jedna relacija može poistovetiti sa jednom tabelom, ali tabelom koja je napunjena podacima.
2. Strukturalna jednostavnost - Obezbeđivanje da struktura modela bude veoma jednostavna, prihvatljiva i razumljiva korisniku. Koristi se relacija – "homogena" i "uniformna" struktura koja je lako razumljiva korisnicima podataka. Takođe, za razliku od ranijih modela podataka kod kojih su se koristile fizičke (relativne i apsolutne adrese) kod relacionog modela podataka se koriste simboličke adrese, koje su zapravo najčešće vrednost primarnog ključa. Svaki podatak u relaciji se pronalazi na osnovu naziva relacije, zadatih obeležja i vrednosti ključa
3. Upotreba deklarativnog jezika – Proceduralni jezik se zamenjuje deklarativnim jezikom koji se temelji na primenjenim tehnikama povezivanja podataka sa prostiranjem ključa. I dok proceduralne jezike karakteriše zadavanje ŠTA i KAKO treba uraditi, kod deklarativnih jezika je dovoljno navesti samo ŠTA je potrebno uraditi. Selekcija kod proceduralnih jezika koja

se vršila putem indikatora aktuelnosti, ili putem odnosa između podataka, kod deklarativnih jezika sada se vrši, kao što je već spomenuto, putem vrednosti obeležja.

Relaciona baza podataka je poseban tip baze podataka kod kojeg se organizacija podataka zasniva na relacionom modelu. Podaci se u ovakvim bazama organizuju u skup relacija između kojih se definišu određene veze. Relacija se definiše kao skup n-torki sa istim atributima, definisanih nad istim domenima iz kojih mogu da uzimaju vrednosti. U relacionim bazama podataka, svaka relacija mora da ima definisan primarni ključ, koji predstavlja atribut pomoću kojeg se jedinstveno identifikuje svaka n-torka. Relacija opciono može da poseduje i strani ključ, preko kojeg ostvaruje vezu sa drugim relacijama. Upravljanje ovakvim bazama podataka se realizuje preko sistema za upravljanje relacionim bazama podataka (SUBP). Za potrebe izrade ovog projekta korišćen je Microsoft-ov sistem za upravljanje bazama podataka, poznatiji kao Microsoft SQL Server.

## DDL

DDL (*eng. Data Definition Language*) je jezik za definisanje podataka, odnosno opis podataka. DDL se koristi za kreiranje i modifikaciju specifikacija šeme baze podataka, fizičke strukture baze podataka, prava pristupa i zaštite baze podataka i novih tipova operacija (programa) za upravljanje podacima. Naredbe DDL-a su, konkretno u ovom projektu, korišćene za kreiranje tabela, kolona, njihovih tipova, zatim sekvenci i slično. Neke od najpoznatijih i najkorišćenijih DDL naredbi su:

1. CREATE - kreiranje objekata u bazi podataka (najčešće tabela)
2. ALTER - promena objekta - npr. kada u gotovu tabelu želimo da dodamo novu kolonu, ograničenje
3. DROP - brisanje objekta - pomoću ove naredbe možemo ukloniti tabelu ili neki drugi objekat iz baze
4. TRUNCATE - brisanje podataka iz tabele - praktično "prazni" tabelu

Kratki primeri DDL naredbi slede u nastavku.

1. Osnovna sintaksa za kreiranje tabele korišćenjem DDL naredbe CREATE:

```
CREATE TABLE schema_name.naziv_tabele
(naziv_obelezja data_type(size) ogranicenje,
naziv_obelezja1,
naziv_obelezja2,
ogranicenje
);
```

2. Osnovna sintaksa za brisanje tabele korišćenjem DDL naredbe DROP:

```
DROP TABLE schema_name.naziv_tabele
```

3. Osnovna sintaksa za dodavanje nove kolone putem modifikacije već postojeće tabele korišćenjem DDL naredbe ALTER

```
ALTER TABLE schema_name.naziv_tabele
ADD naziv_kolone type(size) NOT NULL;
```

4. Osnovna sintaksa za brisanje podataka iz tabele korišćenjem DDL naredbe TRUNCATE

```
TRUNCATE TABLE schema_name.naziv_tabele
```

## DML

DML(eng. *Data Manipulation Language*) je jezik za manipulisanje,odnosno ažuriranje podataka,koji omogućava izmenu stanja baze podataka,odnosno njeno ažuriranje a sve u cilju praćenja izmena stanja podataka u realnom sistemu. DML naredbe omogućavaju čitanje,upisivanje i menjanje podataka u tabelama. Najkorišćenije DML naredbe su:

1. SELECT - čitanje podataka iz baze
2. INSERT - unos podataka u tabelu
3. UPDATE - promena postojećih podataka u tabeli
4. DELETE - brisanje podataka iz tabele

Kratki primeri DML naredbi slede u nastavku.

1. Osnovna sintaksa za čitanje podataka iz baze korišćenjem SELECT naredbe. U konkretnom slučaju \* označa da želimo iščitati sve podatke iz navedene tabele. Naravno,umesto nje mogu se zadati konkretni nazivi kolona koje želimo iščitati,kao i različiti kriterijumi po kojima želimo da izlistamo podatke,koji se mogu navesti u okviru WHERE klauzule.

```
SELECT * FROM schema_name.naziv_tabele
```

2. Osnovna sintaksa za unos podataka u tabelu korišćenjem INSERT naredbe

```
INSERT INTO schema_name.naziv_tabele(naziv_obelezja1,naziv_obelezja2) VALUES ('Primer','Takodje primer')
```

3. Osnovna sintaksa za ažuriranje podata u odredjenoj tabeli korišćenjem UPDATE naredbe

```
UPDATE schema_name.naziv_tabele  
SET naziv_obelezja1 = 'Primer update-a'  
where naziv_obelezja1 = 'Primer'
```

4. Osnovna sintaksa za brisanje podataka iz tabele. U konkretnom primeru pokazan je slučaj kada želimo obrisati sve podatke iz tabele. Naravno,moguće je zadati posebne uslove po kojima želimo brisati u okviru WHERE klauzule.

```
DELETE FROM schema_name.naziv_tabele
```

## SQL

SQL(eng. *Structured Query Language*) je relacioni upitni jezik zasnovan na ANSO i ISO standardima. Relacije se kreiraju jednom naredbom i odmah su dostupne, što ga čini jednostavnim za korišćenje. Uniforman je, jer se svi podaci i rezultati operacija prikazuju u vidu tabele i omogućava interaktivno i klasično programiranje. Većina relacionih baza podataka jedino preko SQL-a dozvoljava pristup podacima. Ovo je i jedan od razloga potrebe da većina korisnika baza podataka upozna komande SQL-a. SQL omogućava korisnicima pristup podacima u sistemima za upravljanje relacionim bazama podataka (npr. Oracle, Microsoft SQL Server). SQL je neproceduralni jezik tj. serveru se samo kaže šta da uradi, ali ne i kako to da uradi. Server baze podataka prevodi komande SQL-a u interne procedure koje obrađuju zahtev. SQL je jednostavan zato što sakriva sve detalje obrade podataka. SQL naredbe se mogu razvrstati u sledećih 7 kategorija:

1. naredbe za šemu baze podataka - za kreiranje, izmenu i izbacivanje šema i objekata šema (CREATE, ALTER, DROP)
2. naredbe za podatke - za prikaz i ažuriranje podataka baze (SELECT, INSERT, UPDATE, DELETE)
3. naredbe za transakcije (SQL-transaction statements) - za startovanje, završavanje i postavljanje parametara za transakcije (COMMIT,



## ROLLBACK)

4. naredbe za kontrolu – koriste se za kontrolu izvršavanja sekvence SQL naredbi (CALL, RETURN)
5. naredbe za konekcije - za uspostavljanje i prekidanje SQL konekcije (CONNECT, DISCONNECT)
6. naredbe za sesije - za postavljanje default vrednosti i drugih parametara SQL sesije (SET)
7. naredbe za dijagnostiku - signalizuju izuzetke u SQL rutinama (GET DIAGNOSTIC)

Naredba SELECT je jedna od najkompleksnijih naredbi SQL-,pa će ovde,i ako je već u prethodnom tekstu prilikom objašnjavanja DML naredbi bilo reči o SELECT naredbi, biti dato detaljnije objašnjenje,te će se samim tim kreirati i adekvatna podloga za kasnije objašnjavanje upita koji su napravljeni nad kreiranom bazom podataka u ovom projektu. Uključuje veći broj ključnih reči i klauzula:

1. SELECT - definiše listu kolona koje će biti uključene u rezultujuću tabelu
2. FROM - definiše tabele iz kojih se pribavljaju podaci za potrebe generisanja rezultujuće tabele. Klauzula FROM može da uključi jednu ili više opcionih JOIN klauzula za povezivanje tabela na osnovu kriterijuma zadatih od strane korisnika.
3. WHERE - definiše predikat na osnovu koga se ograničava broj vrsta u rezultujućoj tabeli. Ova klauzula iz rezultata eliminiše sve vrste za koje specificirani predikat ne vraća vrednost TRUE.
4. GROUP BY - grupiše vrste koje u određenim kolonama imaju identične vrednosti.
5. HAVING - definiše predikat na osnovu koga se elimiše vrste nakon što je klauzula
6. GROUP BY primenjena na rezultujuću tabelu.
7. ORDER BY - koristi se za sortiranje rezultujuće tabele. Korisnici specificiraju kolone po kojima se vrši sortiranje kao i smer sortiranja.

## OBJEKTI

Objekti koji su bili korišćeni prilikom izrade ovog projekta će biti objašnjeni detaljnije u nastavku teksta.

## Šema

Šema predstavlja kolekciju svih objekata koji dele isti prostor imenovanja. Šema može sadržati jednu ili više tabela, procedura, sekvenci, funkcija, trigera i ostalih objekata, a svaka tabela, odnosno bilo koji drugi objekat može pripadati logički tačno jednoj šemi. Šema se kreira naredbom CREATE SCHEMA [naziv šeme]. Kreiranje šeme omogućava logičko grupisanje objekata baze podataka. Svi objekti baze podataka pripadaju nekoj šemi. Prilikom kreiranja objekata šemu možemo ili navesti eksplicitno ili će ona biti implicitno podrazumevana u suprotnom. Implicitno podrazumevano objekat će biti smešten u defaultnu dbo šemu. Prilikom izrade ovog projekta kreirana je šema Angazovanje čiji je naziv u skladu sa izabranom podšemom. Svi nadalje kreirani objekti bili su kreirani isključivo u okviru ove šeme. Sam postupak kreiranja šeme Angazovanje dat je na slici 3.

```
if schema_id ('Angazovanje') is not null
    drop schema Angazovanje;
go
create schema Angazovanje;
go
```

*Slika 3: Postupak kreiranja šeme Angazovanje*

Kao što se može videti, najpre se u skladu sa tehnikom pisanja čistog koda, proverava da li takva šema već postoji, te se nakon toga kreira šema na gore spomenuti način.

## Tabela

Podaci se u SQL-u čuvaju se i prikazuju u obliku tabela. Svaka kolona tabele ima ime i tip podatka ili domen. Red je najmanja jedinica podatka koja se može uneti u tabelu ili izbrisati iz nje. Reprezentacija podataka u tabelarnom obliku je lako razumljiva za korisnike baze podataka. Svaka tabela mora imati različito ime od drugih tabela u bazi podataka ali isto tako i svaka kolona u tabeli mora imati različito ime kolone u jednoj tabeli. Kao što je već i bilo reči prilikom objašnjavanja DDL-a, ta kreiranje tabela i kasnije eventualne promene u strukturi, u SQL se koriste dve naredbe, a to su CREATE TABLE i ALTER TABLE. Na slici 4 dat je prikaz postupka kreiranja jedna od kreiranih tabela tokom projekta, a reč je o tabeli Zaposleni. Analogno ovoj tabeli, bile su kreirane i druge tabele u okviru ovog projekta.

```

create table Angazovanje.Zaposleni (
id_zaposlenog int not null constraint DFT_Zaposleni_id_zaposlenog default(next value for Angazovanje.SEQ_id_zaposlenog),
z_jmbg varchar(13) not null,
z_ime varchar(20) not null,
z_prezime varchar(20) not null,
z_pol char(1) not null,
z_dat_rodj date not null,
z_adresa varchar (50),
z_grad varchar(20),
z_broj_tel varchar(15),
z_drzava varchar(20),
z_datum_zap date not null constraint DFT_Zaposleni_datum_zap default (sysdatetime()),
z_plt numeric(8),

constraint PK_Zaposleni primary key(id_zaposlenog),
constraint CHK_Zaposleni_jmbg check (len(z_jmbg)=13),
constraint UQ_Zaposleni_jmbg unique (z_jmbg),
constraint CHK_Zaposleni_pol check (z_pol in ('m','z')),
constraint CHK_Zaposleni_adresa check (((z_adresa is not null) and z_grad is not null and z_drzava is not null) or
((z_adresa is null) and (z_grad is not null or z_grad is null) and (z_drzava is not null or z_drzava is null))),
constraint CHK_Zaposleni_plt check ((z_plt is not null and len(z_plt) between 5 and 8) or z_plt is null ),
constraint CHK_jmbg_dat_rodj check (z_jmbg like (concat(iif(datepart(day,z_dat_rodj)<10,concat('0',datepart(day,z_dat_rodj)),cast(datepart(day,z_dat_rodj) as varchar)
iif(datepart(month,z_dat_rodj)<10,concat('0',datepart(month,z_dat_rodj)),cast(datepart(month,z_dat_rodj) as varchar)),
cast(right(datepart(year,z_dat_rodj),3) as varchar),'%'))
);
go

```

*Slika 4: Postupak kreiranja tabele Zaposleni*

## Sekvenca

Sekvenca je deljeni (shareable) objekat koji automatski generiše jedinstvene brojeve. Najčešće se koriste za generisanje vrednosti primarnog ključa, te se samim tim izbegava ručno unošenje vrednosti primarnog ključa prilikom unosa podataka u tabelu. Sekvenca brojeva je objekat koji se smešta i generiše nezavisno od tabela pa se zato ista sekvenca može koristiti za više tabela, što se naravno ne preporučuje, već je bolja tehnika kreirati po jednu, ili više sekvenci ako je potrebno za svaku od tabela, te je tako i učinjeno prilikom izrade ovog projekta. Za kreiranje sekvenci se koristi naredba CREATE SEQUENCE, u okviru koje je moguće definisati dodatne parametre, a to su:

- INCREMENT BY – za koliko se uvećava vrednost sekvence,
- MINVALUE – minimalna vrednost sekvence,
- MAXVALUE – maksimalna vrednost sekvence,
- START WITH – od kog broja počinje sekvenca,
- CYCLE – specificira da li se sekvenca restartuje kada dostigne maksimalnu ili minimalnu vrednost.

Postupak kreiranja sekvence, koja će se koristiti za unos vrednosti primarnog ključa u tabeli zaposleni, dat je na slici 5.

```

~
create sequence Angazovanje.SEQ_id_zaposlenog as int
start with 1
increment by 1
no cycle

```

*Slika 5: Postupak kreiranja sekvence*

## Funkcija

Funkcije su objekti koji prihvataju ulazne parametre, izvršavaju nad njima neku logiku i vraćaju određeni rezultat. U zavisnosti od toga šta vraćaju kao rezultat, razlikuje se nekoliko vrsta funkcija:

1. Skalarne funkcije - funkcije koje mogu primiti jedan ili više parametara, a kao rezultat uvek vraćaju jednu vrednost,
2. Inline Table Valued funkcije - tretiraju se kao pogledi, tj. parametrizovani pogledi. To su funkcije koje mogu primiti više parametara a povratna vrednost im je tabela koja nastaje kao rezultat upita u return bloku,
3. Multi statment Table Valued - funkcije koje mogu primiti više parametara. Povratna vrednost im je tabela koja nastaje kao rezultat upita u begin-end bloku. Neophodno je definisati i strukturu tabele koja se vraća kao rezultat funkcije i neophodno je eksplicitno napuniti definisanu tabelu.

Za kreiranje funkcije koristi se naredba CREATE FUNCTION, a opšti postupak kreiranja funkcije dat je na slici 6.

```

CREATE FUNCTION [dbo].[FunctionName]
(
    @param1 int,
    @param2 int
)
RETURNS INT
AS
BEGIN

    RETURN @param1 + @param2

END

```

*Slika 6: Postupak kreiranja funkcije*

## Procedura

Procedure su objekti baze podataka koji sadrže kod koji se izvršava kao zasebna

celina. Procedure slično kao i funkcije primaju parametre. Moguće je kreirati i procedure koje ne primaju nikakve parametre. Procedurama je dozvoljeno da menjaju podatke u bazi što obično i rade dok funkcijama to nije dozvoljeno. Procedure obično imaju zadatak da promene vrednost u tabeli ali mogu i da vrate vrednost obično u formi 1 ili 0 u zavisnosti da li je zadatak uređen ili nije. Procedure se pozivaju komandom exec a gore opisane funkcije su deo select, delete ili update iskaza. Za kreiranje procedure koristi se naredba CREATE PROCEDURE, a opšti postupak kreiranja procedure dat je na slici 7.

```
CREATE PROCEDURE dbo.Sample_Procedure
    @param1 int = 0,
    @param2 int
AS
    SELECT @param1, @param2
RETURN 0
```

*Slika 7: Postupak kreiranja procedure*

## Trigeri

Triger (okidač, engl. *trigger*) je mehanizam kojim se za određenu operaciju modifikovanja podataka (insert, update ili delete), nad određenom tabelom, definiše skup akcija (SQL iskaza) koji se „aktivira“, odnosno izvrši svaki put kad se izvrši operacija. Specifikacija trigera se sastoji iz:

1. oblast aktiviranja (tabela nad kojom se triger definiše),
2. specifikacija operacija koje ga pokreću,
3. uslovi pod kojima se triger aktivira,
4. vreme aktiviranja (neposredno pre ili posle same operacije ili umesto same operacije),
5. frekvencija aktiviranja (jednom za celu operaciju, ili za svaku torku, koja je predmet operacije, pojedinačno),
6. aktivnost (procedura) koju triger treba da realizuje.

Postoje dve vrste trigera: AFTER triger koji se aktivira nakon izvršene DML naredbe i INSTEAD OF triger koji se izvršava umesto same DML naredbe. Za kreiranje trigera koristi se naredba CREATE TRIGGER, a opšti postupak kreiranja trigera dat je na slici 8.

```
CREATE TRIGGER TriggerName
ON [dbo].[TableName]
AFTER DELETE, INSERT, UPDATE
AS
BEGIN
SET NOCOUNT ON
END
```

*Slika 8: Postupak kreiranja trigera*

## **Kursor**

Kursor vraća skup podataka, ali omogućava programskim jezicima da pozovu samo jednu vrstu iz tog skupa u jednom trenutku. Puna funkcionalnost kursora je obezbeđena preko četiri SQL naredbe: DECLARE CURSOR, OPEN, FETCH i CLOSE. Ove naredbe se primenjuju u navedenom redosledu u kursoru. Naredba DECLARE CURSOR deklariše kursor definisanjem njegovog imena, njegovih karakteristika i upita koji se poziva kada se kursor otvori. Znači, u osnovi kursora je validna SELECT naredba. Naredba OPEN otvara kursor i poziva naredbu SELECT i omogućava da rezultat bude raspoloživ za naredbu FETCH. Naredba FETCH vadi podatke iz tabela baze podataka i smešta ih u promenljive koje su prethodno deklarisle. Naredba CLOSE zatvara kursor i tada se ne može pristupiti podacima iz deklaracije kursora. Kako bi se bolje razumelo funkcionisanje kursora, na slici 9 će biti kao primer prikazan kursor koji je korišćen u jednoj od procedura kreiranih u ovom projektu.

```

declare cursor_zaposleni cursor fast_forward read_only for
select z_ime,z_prezime,z_plt,naziv_vrste_radod,osnov_zaposlenja,datediff(month,dat_ang_od,isnull(dat_ang_do,sysdatetime())) ,dat_ang_do
from Angazovanje.Zaposleni zap inner join Angazovanje.Angazovanje ang on (ang.id_zaposlenog=zap.id_zaposlenog)
inner join Angazovanje.Osnov o on (o.id_osnova=ang.id_osnova)
inner join Angazovanje.Vrsta_radnog_odnosa vrs on (vrs.id_vrsta_radod=ang.id_vrsta_radod)
where ang.id_statusa_ang=(select id_statusa_ang from Angazovanje.Status_Angazovanja where naziv_statusa_ang=@status)
order by dat_ang_od asc;

open cursor_zaposleni;

fetch next from cursor_zaposleni into @ime,@prezime,@plata,@vrstaRadnogOdnosa,@osnov,@duzinaAngazovanja,@datumAngazovanjaDo;
while(@@FETCH_STATUS=0)
begin
if(@datumAngazovanjaDo is null)
begin
print concat(@redniBroj,'. ',@ime,' ',@prezime,' sa platom ',isnull(cast(@plata as varchar),'koja nije poznata'),
' i vrstom radnog odnosa ',@vrstaRadnogOdnosa,' je angazovan po osnovu ',@osnov,
'. Angazovanje traje ',@duzinaAngazovanja,' meseci. ');
end
else
begin
print concat(@redniBroj,'. ',@ime,' ',@prezime,' sa platom ',isnull(cast(@plata as varchar),'koja nije poznata'),
' i vrstom radnog odnosa ',@vrstaRadnogOdnosa,' je bio angazovan po osnovu ',@osnov,
'. Angazovanje je trajalo ',@duzinaAngazovanja,' meseci. ');
end
fetch next from cursor_zaposleni into @ime,@prezime,@plata,@vrstaRadnogOdnosa,@osnov,@duzinaAngazovanja,@datumAngazovanjaDo;
set @redniBroj=@redniBroj+1;
end
close cursor_zaposleni;
deallocate cursor_zaposleni;

```

*Slika 9:Primer kursora*

## UPITI

Prilikom izrade ovog projekta, kreirano je 5 SQL upita različite složenosti. Upiti se nalaze u prilogu fajla upiti, a u nastavku je dat opis i objašnjenje tih upita redosledom kojim su navedeni u fajlu.

### 1.Prvi upit

Prvi upit ima za cilj da za svako radno mesto na kom su angažovani zaposleni, prikaže naziv radnog mesta, ime, prezime i jedinstveni matični broj zaposlenog sa najdužim angažovanjem od svih zaposlenih na tom radnom mestu. Takođe, prikazuje se početak angažovanja tog zaposlenog u formatu naziv meseca i godina i informacija o tome da li je angažovanje i dalje aktivno. Pomoću ugrađene funkcije `iif` se proverava da li kolona `dat_ang_do`, koje govori o datumu do kog je zaposleni angažovan, ima null vrednost i u slučaju da ima, to znači da je zaposleni i dalje angažovan, i u tom slučaju se ispisuje da je angažovanje još uvek aktivno, a u suprotnom, ispisuje se datum do kog je zaposleni bio angažovan. Pored ovih informacija, upit ima zadatak da pokaže i trajanje angažovanja. U slučaju da se angažovanje već završilo, pomoću `DATEDIFF()` funkcije izračunava se trajanje angažovanja od datuma početka angažovanja do datuma završetka angažovanja, izraženo u danima. U suprotnom slučaju, biće izračunato trajanje angažovanja do trenutka pokretanja upita. Rezultati upita su sortirani po dužini trajanja angažovanja opadajuće. Rezultat izvršavanja upita dat je na slici 10.

	Naziv radnog mesta	Ime i prezime zaposlenog	Jmbg zaposlenog	Trajanje angažovanja	Pocetak angazovanja	Kraj angazovanja
1	HR asistent	Marko Markovic	3003997123456	1604	January 2018	Angazovanje je jos uvek aktivno
2	Projektni menadzer	Aleksa Aleksic	1308983632547	1584	January 2018	Angazovanje je jos uvek aktivno
3	Administrator baza podataka	Marija Milosevic	2504994321587	1584	January 2018	Angazovanje je jos uvek aktivno
4	Backend developer	Enio Kurtesi	2303999321478	1579	February 2018	Angazovanje je jos uvek aktivno
5	QA analiticar	Ivana Ivanovic	0802994231654	1473	January 2018	2022-02-05
6	HR menadzer	Milan Milanovic	1201987789632	711	June 2020	Angazovanje je jos uvek aktivno
7	Frontend Developer	Petar Petrovic	2108977321987	52	April 2022	Angazovanje je jos uvek aktivno

Slika 10: Rezultat prvog upita

### 2.Drugi upit

Drugi upit za cilj ima da prikaže ime, prezime i adresu zaposlenih u punom obliku, prikazujući podatak o adresi, gradu i državi zaposlenog za one zaposlene



koji su iz Novog Sada ili Beograda, njihovu platu, kao i datum angažovanja u formatu dd.MM.yyyy. Upit prikazuje samo zaposlene čiji je status angažovanja aktivan i koji imaju veću platu od prosečne plate svih zaposlenih. Za pribavljanje podatka o prosečnoj plati zaposlenih kao i za pribavljanje id-a statusa angažovanja koji je aktivan, a sve u cilju izbegavanja bespotrebnog spajanja više tabela, korišćeni su podupiti u where klauzuli. Rezultati upita su sortirani po visini plate opadajuće, a po datumu angažovanja rastuće. Rezultat izvršavanja upita dat je na slici 11.

	Ime i prezime zaposlenog	Adresa zaposlenog	Plata zaposlenog	Naziv radnog mesta	Datum angažovanja
1	Enio Kurtesi	Bulevar Oslobođenja 77, Novi Sad, Srbija	120000	Backend developer	03.02.2018
2	Mitra Mitrovic	Cerska 65, Beograd, Srbija	84000	HR asistent	01.03.2021

Slika 11: Rezultat drugog upita

### 3. Treći upit

Treći upit ima za cilj da izlista id, ime i prezime HR asistenta, broj konkursa koje je on organizovao, i broj angažovanih zaposlenih na osnovu tih konkursa koje je taj HR asistent organizovao. Upit u obzir uzima samo one konkurse organizovane 2021. i 2022. godine koji su raspisani za pozicije programera i developera. U slučaju da HR asistent nije organizovao konkurs ispisuje se poruka "HR asistent nije organizovao nijedan konkurs". Prikazuju se samo oni HR asistenti koji su organizovali manje od 5 konkursa, i na osnovu kojih je zaposlen bar jedan radnik. Za prebrojavanje broja organizovanih konkursa i broja zaposlenih na osnovu konkursa koristi se funkcija COUNT, te se rezultati upita na kraju grupišu prema id, imenu i prezimenu HR asistenta. Rezultati upita su sortirani prema broju organizovanih konkursa opadajuće. Rezultat izvršavanja upita dat je na slici 12.

	ID	Ime	Prezime	Broj organizovanih konkursa	Broj zaposlenih na osnovu konkursa
1	3	Petar	Peric	3	2
2	1	Marko	Markovic	1	1

Slika 12: Rezultat trećeg upita

### 4. Četvrti upit

Četvrti upit ima za cilj da za svaki projekat prikaže id projekta, naziv projekta i naručioca. Takođe, upit prikazuje ime, prezime i godine zaposlenog sa najmanje godina od svih zaposlenih koji rade na tom projektu, odnosno prikazuje najmlađeg radnika na tom projektu. Upit ima zadatak da prikaže i broj časova provedenih tokom rada na projektu i u dodatnoj koloni broj dana provedenih radeći na projektu izveden na osnovu broja časova (jedan dan sadrži 8 radnih časova). U

slučaju da radnik nije radio,ili još uvek nije počeo da radi na projektu u koloni broj časova ispisuje se informacija da Radnik nije radio na projektu,a u koloni broj dana će u tom slučaju pisati 0. Osim toga,upit prikazuje broj telefona zaposlenog u formatu +3816xx. Upit u obzir uzima one samo radnike koje rade na projektima ciji su naručioci FTN,NIS i NS i kojima je status angažovanja aktivan.Rezultati upita su sortirani prema godinama zaposlenih opadajuće. Rezultat izvršavanja upita dat je na slici 13.

	Id projekta	Naziv projekta	Narucilac	Ime zaposlenog	Prezime zaposlenog	Broj casova	Broj dana	Broj godina	Broj telefona
1	5	Refaktoring softvera za nadgledanje poslovanja	NS	Marija	Milosevic	21	2	28	+38163214963
2	9	Novo	FTN	Ivana	Ivanovic	20	2	28	+38169421115
3	10	Novo	FTN	Ivana	Ivanovic	20	2	28	+38169421115
4	3	Gas	NIS	Marko	Markovic	Radnik jos nije radio na projektu	0	25	+38164136632
5	2	Izrada tehnicke dokumentacije za naucno-tehnolos...	FTN	Enio	Kurtesi	65	8	23	+381641993023
6	7	Implementacija platforme za online učenje	FTN	Enio	Kurtesi	55	6	23	+381641993023

Slika 13:Rezultat četvrtog upita

## 5.Peti upit

Peti upit ima za cilj da izlista ime,prezime zaposlenog, njegovu ukupnu zaradu ostvarenu tokom angažovanja u kompaniji i pol zaposlenog u formatu muški/ženski.

Takođe prikazuje prosečno angažovanje svakog od zaposlenih na projektima,a u slučaju da zaposleni nije bio angazovan na projektima ispisuje poruku 'Zaposleni nije učestvovao na projektima'. Upit ima zadatak da prikaže samo one zaposlene čija je plata veća od plate zaposlenih čije ime počinje na Ma. Pored toga prikazuje i osnov njegovog zaposlenja. Rezultati upita su sortirani po ukupnoj zaradi opadajuće,a po imenu i prezimenu rastuće. Rezultat izvršavanja upita dat je na slici 14.

	Ime	Prezime	Pol zaposlenog	Ukupna zarada zaposlenog	Prosecno angazovanje na projektu	Osnov zaposlenja
1	Enio	Kurtesi	Muški	6240000	42.50	Zaposlenje na osnovu stipendije
2	Ana	Petrovic	Ženski	3850000	26.75	Zaposlenje na osnovu stipendije
3	Petra	Petrovic	Ženski	3003000	Zaposleni nije ucestvovao na projektima	Zaposlenje na osnovu unapredjenja
4	Mitra	Mitrovic	Ženski	1260000	Zaposleni nije ucestvovao na projektima	Zaposlenje na osnovu konkursa
5	Sasa	Ilic	Muški	1155000	Zaposleni nije ucestvovao na projektima	Zaposlenje na osnovu konkursa
6	Danilo	Danilovic	Muški	368000	Zaposleni nije ucestvovao na projektima	Zaposlenje na osnovu konkursa
7	Jovana	Jovanovic	Muški	337500	Zaposleni nije ucestvovao na projektima	Zaposlenje na osnovu konkursa

Slika 14: Rezultat petog upita

## FUNKCIJE

Prilikom izrade ovog projekta, kreirane su 2 funkcije različite vrste. Funkcije se nalaze u prilogu fajla funkcije, a u nastavku je dat opis i objašnjenje tih funkcija redosledom kojim su navedeni u fajlu.

### 1.Prva funkcija

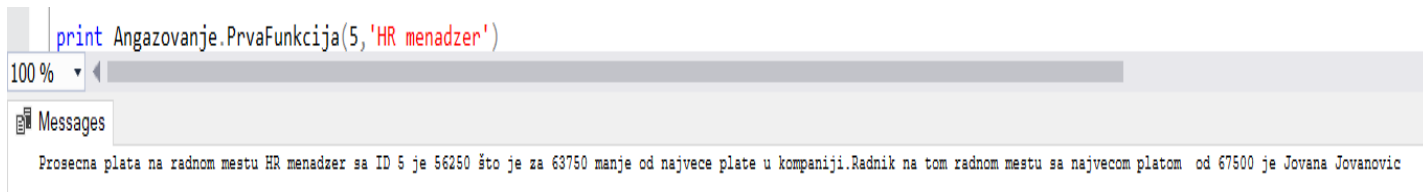
Prva funkcija predstavlja skalarnu funkciju koja za prosledjeni id i naziv radnog mesta vraća prosečnu platu radnika na tom radnom mestu, informaciju o tome koliko je prosečna plata na tom radnom mestu manja od maksimalne plate u kompaniji i ime i prezime radnika sa najvećom platom na tom radnom mestu. S obzirom da funkcija vraća veći broj informacija, povratni tip podatka je varchar, dužine 200. Parametri funkcije su opcioni, te se funkcija može pozivati na više načina, prosledjivanjem samo naziva radnog mesta, prosledjivanjem ID radnog mesta, ili prosledjivanjem i naziva i ID-a radnog mesta. U slučaju pogrešno zadatih parametara, a sve u zavisnosti od toga koji parametri su prosleđeni funkcija će vratiti odgovarajuću informaciju, prema narednom šablonu:

1. U slučaju da se funkcija poziva zadavanjem oba parametra, u slučaju pogrešno zadatih parametara biće vraćena poruka 'Pogrešno zadati parametri funkcije! Ne postoji radno mesto sa takvom kombinacijom parametara!'
2. U slučaju da se funkcija poziva zadavanjem samo ID-a radnog mesta, a naziv nije poznat, biće vraćena poruka 'Pogrešno zadati parametri funkcije! Ne postoji radno mesto sa takvim ID!'
3. U slučaju da se funkcija poziva zadavanjem samo naziva radnog mesta, a ID nije poznat, biće vraćena poruka 'Pogrešno zadati parametri funkcije! Ne postoji radno mesto sa takvim nazivom!'

Kako bi se obezbedilo adekvatno izvršavanje funkcije, uz pomoć if uslova su obrađeni svi mogući slučajevi prosleđivanja parametara, te se u okviru tih uslova zapravo proverava koji od prosleđenih parametara ima null vrednost, te se na taj način zna koji parametar nije zadat prilikom pozivanja funkcije. Logika izvršavanja funkcije u svim mogućim slučajevima je jako slična i svodi se na izvlačenje informacija o prosečnoj plati na odgovarajućem radnom mestu, zatim informacija o maksimalnoj plati u kompaniji, te imenu i prezimenu zaposlenog sa najvećom platom na tom radnom mestu. Nakon izvlačenja iz odgovarajućih tabela ovi podaci se smeštaju u prethodno deklarisanu promenljivu, te se nakon toga vrši konkatenacija sa odgovarajućim tekstom radi adekvatnog ispisa u vraćenom rezultatu.

U nastavku slede primeri pozivanja funkcije u različitim situacijama, odnosno biće prikazan rad funkcije u zavisnosti od toga koji parametri joj se proslede.

Na slici 15 prikazano je ponašanje funkcije za prosleđen ID radnog mesta 5 i naziv radnog mesta HR menadzer.



```
print Angazovanje.PrvaFunkcija(5, 'HR menadzer')
```

100 %

Messages

Prosečna plata na radnom mestu HR menadzer sa ID 5 je 56250 što je za 63750 manje od najveće plate u kompaniji. Radnik na tom radnom mestu sa najvećom platom od 67500 je Jovana Jovanovic

*Slika 15:Rezultat prve funkcije, slučaj 1*

Na slici 16 prikazano je ponašanje funkcije za prosleđen ID radnog mesta 5 i ne prosledjen naziv radnog mesta.



```
print Angazovanje.PrvaFunkcija(5, null)
```

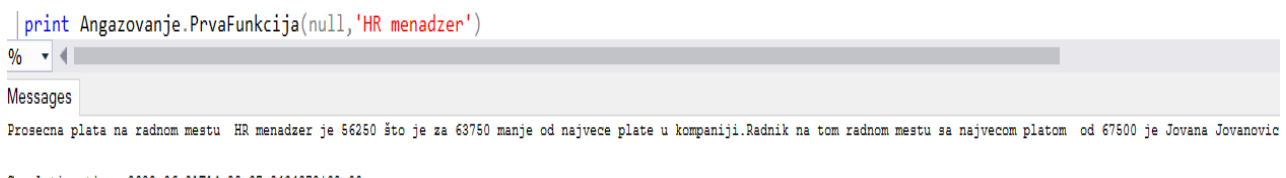
100 %

Messages

Prosečna plata na radnom mestu sa ID 5 je 56250 što je za 63750 manje od najveće plate u kompaniji. Radnik na tom radnom mestu sa najvećom platom od 67500 je Jovana Jovanovic

*Slika 16:Rezultat prve funkcije, slučaj 2*

Na slici 17 prikazano je ponašanje funkcije za neprosleđen ID radnog mesta i prosleđen naziv radnog mesta HR menadzer.



```
print Angazovanje.PrvaFunkcija(null, 'HR menadzer')
```

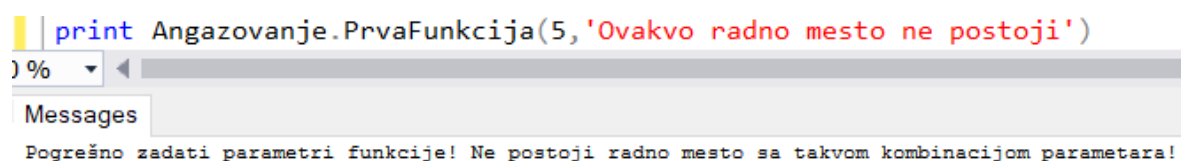
100 %

Messages

Prosečna plata na radnom mestu HR menadzer je 56250 što je za 63750 manje od najveće plate u kompaniji. Radnik na tom radnom mestu sa najvećom platom od 67500 je Jovana Jovanovic

*Slika 17:Rezultat prve funkcije, slučaj 3*

Na slici 18 prikazano je ponašanje funkcije u slučaju kada se proslede pogrešni parametri.



```
print Angazovanje.PrvaFunkcija(5, 'Ovakvo radno mesto ne postoji')
```

100 %

Messages

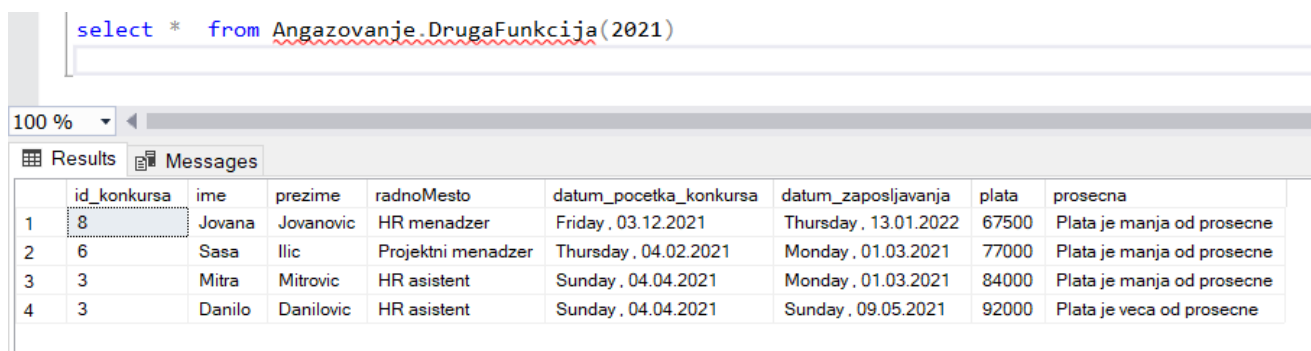
Pogrešno zadati parametri funkcije! Ne postoji radno mesto sa takvom kombinacijom parametara!

*Slika 18:Rezultat prve funkcije, slučaj 4*

## 2. Druga funkcija

Druga funkcija predstavlja Multi statment Table-Valued funkciju kojoj se kao parametar prosleđuje proizvoljna godina, pa je shodno tome ovaj parametar tipa int. Ova funkcija ima za cilj da za prosleđenu godinu u obliku tabele vrati informacije o konkursima održanim u toj godini. Pod relevantnim informacijama o konkursima podrazumeva se id konkursa kao i datum objavljivanja konkursa. Pored toga, u toj novoj tabeli se nalaze i informacije o zaposlenima koji su se zaposlili na osnovu tog konkursa. Za svakog od zaposlenih prikazuje se podaci o njegovom imenu, prezimenu, datumu zapošljavanja, plati, kao i informacija o tome kakva je njegova plata u odnosu na prosečnu platu svih zaposlenih te godine, bez obzira na to da li su oni zaposleni na osnovu konkursa, ili su se pak zaposlili po nekom drugom osnovu. S obzirom da se radi Multi statment Table-Valued funkciji, njena povratna vrednost, kao što je već spomenuto, jeste tabela koja nastaje kao rezultat SQL upita u begin-end bloku. Pre toga, neophodno je definisati i samu strukturu tabele koja se vraća kao rezultat funkcije a zatim je neophodno istu tu tabelu eksplicitno napuniti podacima vraćenim iz upita u begin-end bloku.

Na slici 19 i 20 dati su primeri ponašanja funkcije za različite prosleđene godine.



```
select * from Angazovanje.DrugaFunkcija(2021)
```

	id_konkursa	ime	prezime	radnoMesto	datum_pocetka_konkursa	datum_zaposljavanja	plata	prosekna
1	8	Jovana	Jovanovic	HR menadzer	Friday , 03.12.2021	Thursday , 13.01.2022	67500	Plata je manja od prosečne
2	6	Sasa	Ilic	Projektni menadzer	Thursday , 04.02.2021	Monday , 01.03.2021	77000	Plata je manja od prosečne
3	3	Mitra	Mitrovic	HR asistent	Sunday , 04.04.2021	Monday , 01.03.2021	84000	Plata je manja od prosečne
4	3	Danilo	Danilovic	HR asistent	Sunday , 04.04.2021	Sunday , 09.05.2021	92000	Plata je veca od prosečne

Slika 19: Rezultat druge funkcije za prosleđenu godinu 2021

```
select * from Angazovanje.DrugaFunkcija(2022)
```

100 %								
Results Messages								
	id_konkursa	ime	prezime	radnoMesto	datum_pocetka_konkursa	datum_zaposljavanja	plata	prosecna
1	4	Petar	Petrovic	Frontend Developer	Monday , 07.03.2022	Sunday , 10.04.2022	60000	Plata je manja od prosečne

*Slika 20:Rezultat druge funkcije za prosleđenu godinu 2022*

U slučaju da nema podataka o konkursima organizovanih u godini koja je prosleđena kao parametar,funkcija će vratiti praznu tabelu,kao što je to prikazano na slici 21,gde se može videti ponašanje funkcije za prosleđenu godinu 2017,u kojoj nije organizovan nijedan konkurs.

```
select * from Angazovanje.DrugaFunkcija(2017)
```

100 %								
Results Messages								
	id_konkursa	ime	prezime	radnoMesto	datum_pocetka_konkursa	datum_zaposljavanja	plata	prosecna

*Slika 21:Rezultat druge funkcije za prosleđenu godinu 2017*

## PROCEDURE

Prilikom izrade ovog projekta, kreirane su 2 procedure. Procedure se nalaze u prilogu fajla procedure, a u nastavku je dat opis i objašnjenje tih procedura redosledom kojim su navedeni u fajlu.

### 1. Prva procedura

Prva procedura ima za cilj da za prosleđen naziv statusa angažovanja izlista sve podatke o zaposlenima sa tim statusom angažovanja. U slučaju da postoje zaposleni sa takvim statusom angažovanja u zaglavlju ispisa procedure se ispisuje naredni tekst:

' Status angažovanja <status> imaju sledeći zaposleni: '.

Nakon tog ispisa sledi spisak zaposlenih koji imaju taj status angažovanja. Za svakog od zaposlenih sa tim statusom angažovanja se ispisuje redni broj, ime i prezime zaposlenog, njegova plata, a u slučaju da plata nije poznata ispisuje se da plata nije poznata, zatim vrsta radnog odnosa i osnov po kom je zaposleni angažovan. Pored toga ispisuje se i podatak koliko traje/ili je trajalo angažovanje u mesecima. U proceduri je obezbeđeno da u slučaju da angažovanje i dalje traje, u ispisu bude ispisano da angažovanje traje, a u slučaju da je završeno da u ispisu piše da je angažovanje trajalo određen broj meseci. Nakon liste svih zaposlenih sa prosleđenim statusom angažovanja ispisuje se informacija o tome koliko ima zaposlenih sa tim statusom angažovanja.

U slučaju da za prosleđeni naziv statusa angažovanja nema zaposlenih sa takvim statusom angažovanja, nakon pozivanja procedure ispisaće se naredni tekst: 'Ne postoje zaposleni sa datim statusom angažovanja - <status> '. Takođe, može se desiti da se prilikom poziva procedure prosledi naziv statusa angažovanja koji zapravo ne postoji, pa će u tom slučaju procedura ispisati poruku 'Ne postoji status angažovanja <status> '.

U nastavku slede primeri ponašanja procedure za različite prosleđene vrednosti parametra naziv statusa angažovanja.

Na slici 22 prikazan je primer ponašanja procedure za prosleđeni status angažovanja 'Aktivno angažovanje'.

```
exec Angazovanje.PrvaProcedura 'Aktivno angazovanje'

100 %

Messages

Status angazovanja Aktivno angazovanje imaju sledeci zaposleni
1. Marko Markovic sa platom 55000.00 i vrstom radnog odnosa Radni odnos na neodređeno je angazovan po osnovu Zaposlenje na osnovu strucne prakse. Angazovanje traje 53 meseci.
2. Aleksa Aleksić sa platom 45000.00 i vrstom radnog odnosa Radni odnos na neodređeno je angazovan po osnovu Zaposlenje na osnovu preporuke. Angazovanje traje 53 meseci.
3. Marija Milosevic sa platom 63000.00 i vrstom radnog odnosa Radni odnos na neodređeno je angazovan po osnovu Zaposlenje na osnovu strucne prakse. Angazovanje traje 53 meseci.
4. Enio Kurtesi sa platom 120000.00 i vrstom radnog odnosa Radni odnos sa nepunim radnim vremenom je angazovan po osnovu Zaposlenje na osnovu stipendije. Angazovanje traje 52 meseci.
5. Milan Milanovic sa platom 45000.00 i vrstom radnog odnosa Radni odnos na neodređeno je angazovan po osnovu Zaposlenje na osnovu preporuke. Angazovanje traje 24 meseci.
6. Sasa Ilic sa platom 77000.00 i vrstom radnog odnosa Radni odnos na neodređeno je angazovan po osnovu Zaposlenje na osnovu konkursa. Angazovanje traje 15 meseci.
7. Mitra Mitrovic sa platom 84000.00 i vrstom radnog odnosa Radni odnos na neodređeno je angazovan po osnovu Zaposlenje na osnovu konkursa. Angazovanje traje 15 meseci.
8. Jovana Jovanovic sa platom 67500.00 i vrstom radnog odnosa Radni odnos sa pripravnikom; je angazovan po osnovu Zaposlenje na osnovu konkursa. Angazovanje traje 5 meseci.
9. Petar Petrovic sa platom 60000.00 i vrstom radnog odnosa Radni odnos sa rad van prostorija poslodavca je angazovan po osnovu Zaposlenje na osnovu konkursa. Angazovanje traje 2 meseci.
Ukupno zaposlenih sa statusom angazovanja Aktivno angazovanje: 9

Completion time: 2022-06-01T15:48:02.3497029+02:00
```

Slika 22: Primer ponašanja prve procedure

Na slici 23 prikazan je primer ponašanja procedure za prosleđeni status angažovanja 'Prekinuto angažovanje'.

```
exec Angazovanje.PrvaProcedura 'Prekinuto angazovanje'

00 %

Messages

Status angazovanja Prekinuto angazovanje imaju sledeci zaposleni
1. Ana Petrovic sa platom 77000.00 i vrstom radnog odnosa Radni odnos na određeno je bio angazovan po osnovu Zaposlenje na osnovu stipendije. Angazovanje je trajalo 50 meseci.
2. Petar Peric sa platom koja nije poznata i vrstom radnog odnosa Radni odnos na neodređeno je bio angazovan po osnovu Zaposlenje na osnovu preporuke. Angazovanje je trajalo 21 meseci.
Ukupno zaposlenih sa statusom angazovanja Prekinuto angazovanje: 2

Completion time: 2022-06-01T15:52:40.1969440+02:00
```

Slika 23: Primer ponašanja prve procedure

Na slici 24 prikazan je primer ponašanja procedure za prosleđeni status angažovanja koji ne postoji.

```
exec Angazovanje.PrvaProcedura 'Ovakvo angazovanje ne postoji'

0 %

Messages

Ne postoji status angazovanja Ovakvo angazovanje ne postoji

Completion time: 2022-06-01T15:54:41.8485959+02:00
```

Slika 24: Primer ponašanja prve procedure

## 2.Druga procedura



Druga procedura ima za cilj da za prosledjeni id zaposlenog prikaže informacije o tom zaposlenom. Pod informacijama se podrazumeva ime i prezime zaposlenog, njegova adresa prebivališta u punom obliku što podrazumeva ispisivanje adrese, grada i države zaposlenog, ukoliko su ti podaci naravno transparentni i postoje. Nakon toga, prikazuje se plata zaposlenog, a u slučaju da plata nije poznata, ispisuje se informacija da plata nije poznata. Zatim, ispisuje se radno mesto na kom je zaposleni angažovan. U zavisnosti od toga na kom radnom mestu je zaposleni angažovan, biće prikazane dodatne informacije o njemu. Pod dodatnim informacijama se podrazumevaju sledeći stvari:

1. U slučaju da je zaposleni angažovan na radnom mestu HR menadžer, biće prikazan broj evaluacionih testova koje je zaposleni, radeći na tom radnom mestu realizovao
2. U slučaju da je zaposleni angažovan na radnom mestu HR asistent, biće prikazan broj konkursa za koje je zaposleni, radeći na tom radnom mestu bio odgovoran
3. U slučaju da je zaposleni angažovan na radnom mestu Projektni menadžer, biće prikazan broj projekata na kojima je zaposleni, radeći na tom radnom mestu bio projektni menadžer
4. U slučaju da je zaposleni angažovan na nekom drugom radnom mestu, to će ukazati na to da se radi o zaposlenom koji se tretira kao radnik i koji učestvuje na projektima, te će za njega biti prikazan broj projekata na kojima je učestvovao zaposleni, radeći na tom radnom mestu

Primeri ponašanja procedure za različite prosledene vrednosti id-a zaposlenog, dati su na slikama 25, 26, 27, 28. Na slici 25 može se videti da je proceduri prosleđen id zaposlenog koji je HR menadžer, zatim na slici 26 se može videti da je proceduri prosleđen id zaposlenog koji je HR asistent, a na slici 27 da je proceduri prosleđen id zaposlenog koji je Projektni menadžer. Na samom kraju, na slici 28, prikazano je ponašanje procedure u slučaju kada joj se prosledi id zaposlenog koji je običan radnik i koji radi na projektima.

```
exec Angazovanje.DrugaProcedura 3
```

00 %

Messages

Informacije o zaposlenom sa ID-jem 3  
Ime i prezime: Petar Peric  
Adresa i kontakt: Beograd,Srbija, 062181585  
Plata: nije poznata  
Zaposlen je na radnom mestu HR menadzer  
Broj evaluacionih testova koji je realizovao ovaj zaposleni je 4

Completion time: 2022-06-01T16:19:05.3062142+02:00

Slika 25: Primer ponašanja druge procedure

```
exec Angazovanje.DrugaProcedura 1
```

%

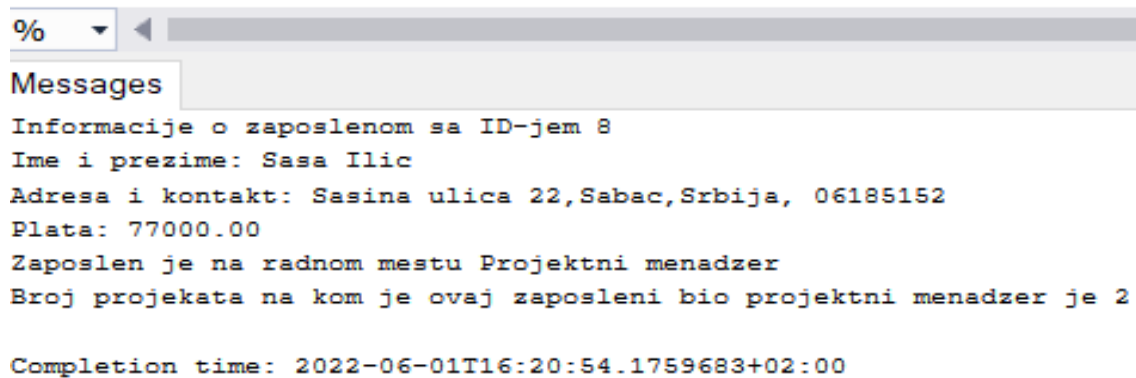
Messages

Informacije o zaposlenom sa ID-jem 1  
Ime i prezime: Marko Markovic  
Adresa i kontakt: Bulevar Oslobođenja 5,Novi Sad,Srbija, 064136632  
Plata: 55000.00  
Zaposlen je na radnom mestu HR asistent  
Broj konkursa za koje je bio odgovoran ovaj zaposleni je 3

Completion time: 2022-06-01T16:19:50.0772709+02:00

Slika 26: Primer ponašanja druge procedure

`exec Angazovanje.DrugaProcedura 8`

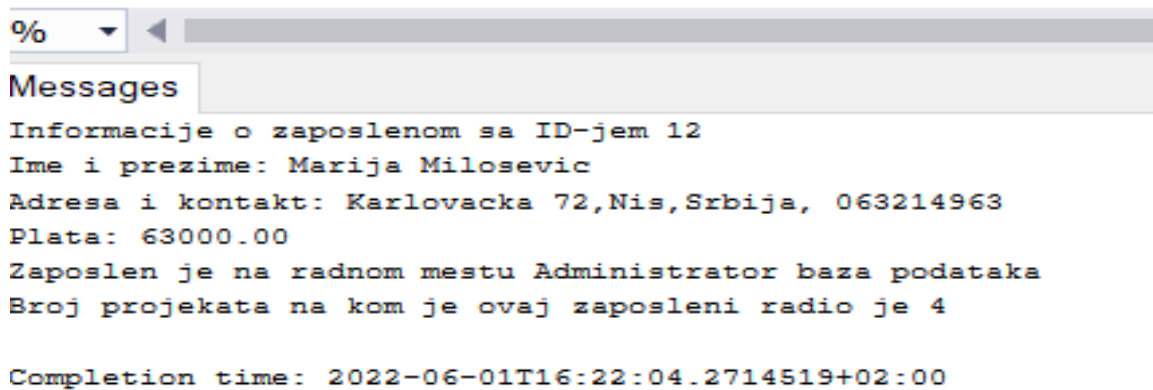


The screenshot shows a database client window with a title bar containing a dropdown menu set to '%'. Below the title bar is a tab labeled 'Messages'. The main area displays the output of a SQL procedure call. The text is as follows:

```
Informacije o zaposlenom sa ID-jem 8  
Ime i prezime: Sasa Ilic  
Adresa i kontakt: Sasina ulica 22,Sabac,Srbija, 06185152  
Plata: 77000.00  
Zaposlen je na radnom mestu Projektni menadzer  
Broj projekata na kom je ovaj zaposleni bio projektni menadzer je 2  
  
Completion time: 2022-06-01T16:20:54.1759683+02:00
```

*Slika 27: Primer ponašanja druge procedure*

`exec Angazovanje.DrugaProcedura 12`



The screenshot shows a database client window with a title bar containing a dropdown menu set to '%'. Below the title bar is a tab labeled 'Messages'. The main area displays the output of a SQL procedure call. The text is as follows:

```
Informacije o zaposlenom sa ID-jem 12  
Ime i prezime: Marija Milosevic  
Adresa i kontakt: Karlovacka 72,Nis,Srbija, 063214963  
Plata: 63000.00  
Zaposlen je na radnom mestu Administrator baza podataka  
Broj projekata na kom je ovaj zaposleni radio je 4  
  
Completion time: 2022-06-01T16:22:04.2714519+02:00
```

*Slika 28: Primer ponašanja druge procedure*

## TRIGERI

Prilikom izrade ovog projekta, kreirana su 2 trigeri. Trigeri se nalaze u prilogu fajla triggeri, a u nastavku je dat opis i objašnjenje tih trigeri redosledom kojim su navedeni u fajlu.

### 1.Prvi trigger

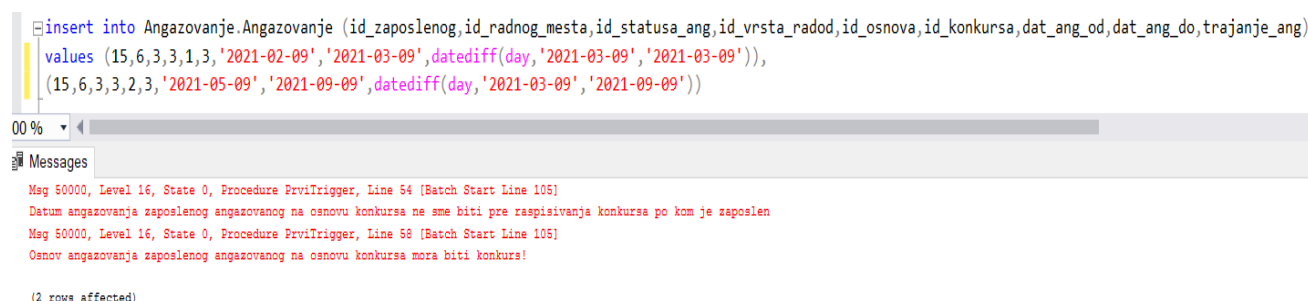
Prvi trigger je kreiran nad tabelom Angazovanje i ima za cilj da implementira poslovno pravilo da datum početka angažovanja zaposlenog koji je angažovan na osnovu nekog od konkursa raspisanih od strane kompanije, ne sme biti pre datuma objavljivanja samog konkursa po kom je zaposlen. Takođe, trigger zabranjuje unos id osnova u tabelu Angazovanje, čiji je naziv različit od 'Zaposlenje na osnovu konkursa' u slučaju da je zaposlen angažovan na osnovu konkursa. Kreirani trigger je vrste INSTEAD OF i okida se prilikom izmene nekog od gore spomenutih obeležja, ili prilikom unosa cele torke za radnika koji je angažovan na osnovu konkursa. Ova vrsta triggera je izabrana kako bi se zasigurno osiguralo da pogrešni podaci neće biti upisani u tabelu Angazovanje. U slučaju da je narušeno neko od ovih 2 pravila, trigger će vratiti grešku sa adekvatnom porukom i to:

1. U slučaju da je za zaposlenog angažovanog na osnovu konkursa pokušano da se unese, ili da se za već postojećeg zaposlenog izmeni vrednost kolone `dat_ang_od` na vrednost datuma koja je pre datuma objavljivanja konkursa po kojem je zaposleni angažovan poruka greške će biti 'Datum angazovanja zaposlenog angazovanog na osnovu konkursa ne sme biti pre raspisivanja konkursa po kom je zaposlen!'
2. U slučaju da je za zaposlenog angažovanog na osnovu konkursa pokušano da se unese, ili da se za već postojećeg zaposlenog izmeni vrednost kolone `id_osnova` na vrednost id-a čiji je naziv osnova u tabeli Osnov različit od 'Zaposlenje na osnovu konkursa' poruka greške će biti 'Osnov angazovanja zaposlenog angazovanog na osnovu konkursa mora biti konkurs!'

Kako bi se obezbedio adekvatan rad triggera u slučaju pokušaja unosa ili izmene više od jedne torke, u okviru triggera su korišćeni kursori za prolazak kroz sve torke koje se mogu naći u deleted, odnosno inserted tabeli, koje predstavljaju privremene tabele i koje su od velikog značaja prilikom rada sa triggerima. U slučaju da nije narušeno nijedno od gore navedenih pravila, svi podaci će biti pokupljeni iz inserted tabele, smešteni u prethodno deklarisanu promenljivu te zatim prosleđeni u tabelu Angazovanje, bilo da se radi o unosu nove torke ili ažuriranju. Prilikom ažuriranja, takođe je potrebno proslediti sve podatke, jer

postoji mogućnost da je korisnik ažurirao neko obeležje koje nije obuhvaćeno triggerom, te bi se na taj način prosleđivanjem samo izmene obeležja koje je pokriveno triggerom došli do situacije gubitka ažuriranja, što svakako narušava konzistentnost baze podataka.

Na slici 29 prikazano je okidanje triggera. Kao što se može videti, pokušano je insertovanje više od jedne torke, pri čemu jedna torka ne zadovoljava prvo pravilo, a to je da datum početka angažovanja mora biti pre datuma objavljivanja konkursa, a druga torka ne zadovoljava drugo pravilo, a to je nekorektan unos id osnova.



```
insert into Angazovanje.Angazovanje (id_zaposlenog,id_radnog_mesta,id_statusa_ang,id_vrsta_radod,id_osnova,id_konkursa,dat_ang_od,dat_ang_do,trajanje_ang)
values (15,6,3,3,1,3,'2021-02-09','2021-03-09',datediff(day,'2021-03-09','2021-03-09')),
(15,6,3,3,2,3,'2021-05-09','2021-09-09',datediff(day,'2021-03-09','2021-09-09'))
```

00 %

Messages

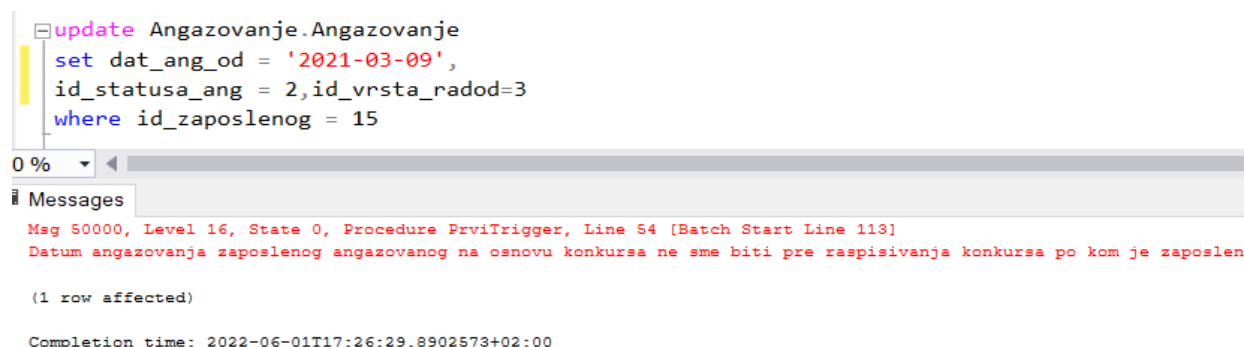
Msg 50000, Level 16, State 0, Procedure PrviTrigger, Line 54 [Batch Start Line 105]  
Datum angazovanja zaposlenog angazovanog na osnovu konkursa ne sme biti pre raspisivanja konkursa po kom je zaposlen

Msg 50000, Level 16, State 0, Procedure PrviTrigger, Line 58 [Batch Start Line 105]  
Osnov angazovanja zaposlenog angazovanog na osnovu konkursa mora biti konkurs!

(2 rows affected)

Slika 29: Primer okidanja triggera prilikom inserta

Na slici 30 prikazano je okidanje triggera prilikom pokušaja ažuriranja torke, i unosa nekorektne vrednosti za datum početka angažovanja.



```
update Angazovanje.Angazovanje
set dat_ang_od = '2021-03-09',
id_statusa_ang = 2,id_vrsta_radod=3
where id_zaposlenog = 15
```

0 %

Messages

Msg 50000, Level 16, State 0, Procedure PrviTrigger, Line 54 [Batch Start Line 113]  
Datum angazovanja zaposlenog angazovanog na osnovu konkursa ne sme biti pre raspisivanja konkursa po kom je zaposlen

(1 row affected)

Completion time: 2022-06-01T17:26:29.8902573+02:00

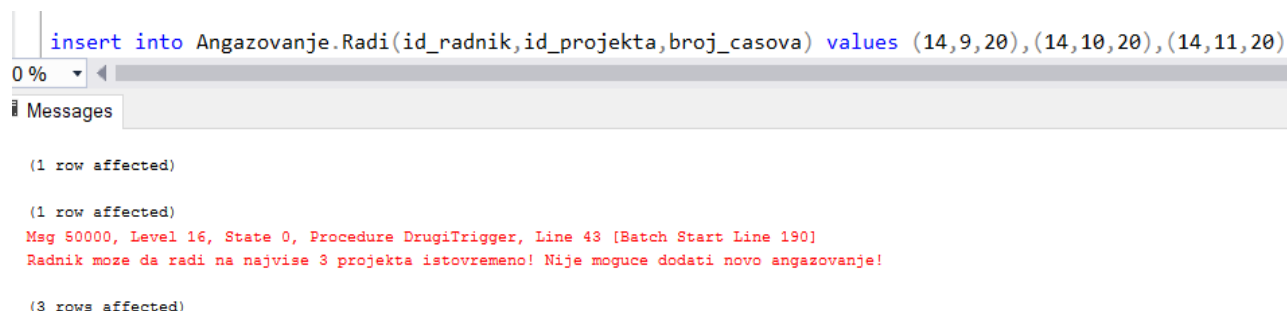
Slika 30: Primer okidanja triggera prilikom update-a

## 2. Drugi trigger

Drugi trigger je kreiran nad tabelom Radi i implementira poslovno pravilo da

radnik ne sme istovremeno da radi na više od 3 projekta koja su aktivna. S obzirom da se u tabeli Radi nalazi jedna svojevrsna evidencija angažmana zaposlenih na projektima, trigger ima za cilj da pronade sve moguće angažmane na projektima zaposlenog koji se unosi i da proveriti da li su projekti na kojima on radi zaista aktivni. Uz pomoć kursora, prolazi se kroz sve projekte na kojima je zaposleni radio/ili radi i vrši se prebrojavanje onih koji su aktivni, odnosno onih projekata kod kojih datum završetka ima null vrednost, što ukazuje na to da oni nisu završeni. Kreirani trigger je vrste INSTEAD OF i okida se prilikom pokušaja unosa nove torke, odnosno novih torki u tabelu Radi. U slučaju da je narušeno poslovno pravilo, i da se ustanovi da zaposleni radi na više od 3 projekta koja su trenutno aktivna, trigger će vratiti grešku sa adekvatnom porukom a to je 'Radnik može da radi na najviše 3 projekta istovremeno! Nije moguće dodati novo angazovanje!'. Takođe, trigger proverava još jedno poslovno pravilo a to je da nije moguće evidentirati novi angažman radnika na projektu za projekat koji je već završen. U tom slučaju, ukoliko se prilikom unosa nove torke zada vrednost id projekta onog projekta koji je završen, trigger će vratiti grešku sa porukom 'Nije moguće dodati novo angazovanje za radnika na projektu koji je već završen!'. Slično kao i kod prvog triggera, kako bi se obezbedio adekvatan rad triggera u slučaju pokušaja unosa, u okviru triggera su korišćeni kursori za prolazak kroz sve torke koje se mogu naći u inserted tabeli.

Na slici 31 prikazano je okidanje triggera prilikom inserta 3 nove torke u tabelu Radi. Naime, u sve 3 torke unosi se zaposleni sa id 14, za kojeg u tabeli Radi već postoji evidentirano da radi/ili je radio na 5 projekata. Međutim, od tih 5 projekata, zapravo je samo jedan aktivan, te se s toga trigger neće aktivirati prilikom insertovanja prve dve torke. Međutim, kako se s unosom prve dve torke, povećava broj aktivnih projekata na kojem zaposleni trenutno radi sa 1 na 3, trigger zabranjuje unos treće torke zbog narušavanja poslovnog pravila i prijavljuje grešku.



```

insert into Angazovanje.Radi(id_radnik,id_projekta,broj_casova) values (14,9,20),(14,10,20),(14,11,20)
0 %
Messages

(1 row affected)

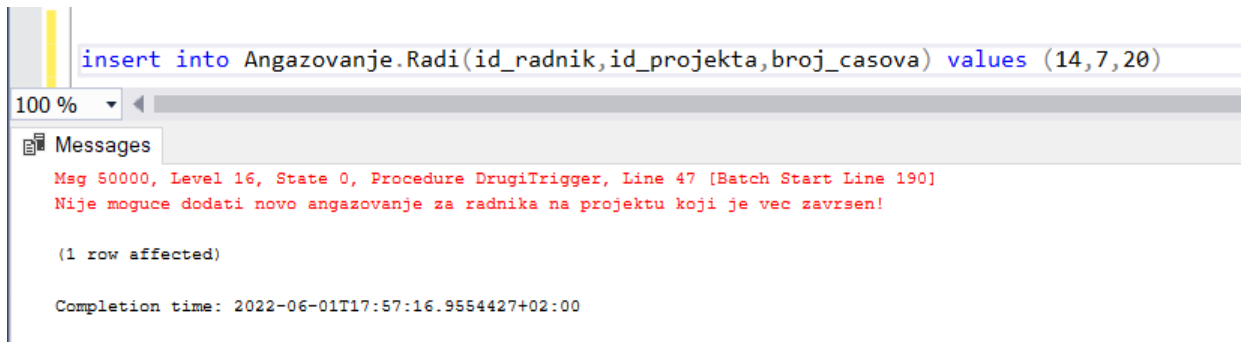
(1 row affected)
Msg 50000, Level 16, State 0, Procedure DrugiTrigger, Line 43 [Batch Start Line 190]
Radnik može da radi na najviše 3 projekta istovremeno! Nije moguće dodati novo angazovanje!

(3 rows affected)

```

*Slika 31:Primer okidanja drugog triggera*

Na slici 32, može se videti primer okidanja trigger prilikom pokušaja dodavanja angažmana zaposlenog na projekat, koji je već završen.



```
insert into Angazovanje.Radi(id_radnik,id_projekta,broj_casova) values (14,7,20)
```

100 %

Messages

Msg 50000, Level 16, State 0, Procedure DrugiTrigger, Line 47 [Batch Start Line 190]  
Nije moguće dodati novo angazovanje za radnika na projektu koji je već završen!

(1 row affected)

Completion time: 2022-06-01T17:57:16.9554427+02:00

*Slika 32: Primer okidanja drugog trigera*

## ZAKLJUČAK

Iskustvo rada na projektima iz predmeta Projektovanje baza podataka i Sistema baza podataka smatram neprocenjivim. Radeći na projektu, uspeo sam da složim kompletnu sliku kako izgleda jedan proces projektovanja baze podataka i njena kasnija implementacija u praksi. Svaki problem na koji sam naišao u toku rad nad projektom smatrao sam izazovom i drago mi je što sam uspeo sve da uspešno privedem kraju. Mislim da će mi znanje koje sam stekao ova dva semestra nesumnjivo pomoći u mom profesionalnom razvoju.