



**Univerzitet u Novom Sadu Fakultet tehničkih nauka**

*Departman za industrijsko inženjerstvo i menadžment*



## **INFORMACIONI SISTEM ZA PODRŠKU PRAĆENJA PREDMETA ADVOKATSKE KANCELARIJE**

**-SISTEMI BAZA PODATAKA-**

**Filip Ilić IT7/2020**

**Novi Sad, 2023.**

## SADRŽAJ

UVOD.....	4
1. ANALIZA PROGRAMSKOG DOMENA .....	5
2. ER MODEL .....	6
3. ER MODEL PODŠEME .....	7
4. TABELARNI PRIKAZ OBELEŽJA I OGRANIČENJA .....	9
5. RELACIONI MODEL .....	22
6. DDL.....	27
7. DML.....	30
8. SQL .....	33
9. OBJEKTI .....	34
10. UPITI.....	38
11. FUNKCIJE .....	43
12. PROCEDURE .....	49
13. TRIGERI.....	55
14. ZAKLJUČAK .....	59

## SADRŽAJ SLIKA

Slika 1 ER šema cele baze podataka (na nivou naziva tipova) .....	6
Slika 2 ER podšema Uplata.....	7
Slika 3 Kreiranje tabele DDL komandom CREATE .....	27
Slika 4 Dodavanje kolone DDL komandom ALTER .....	28
Slika 5 Brisanje tabele DDL komandom DROP .....	28
Slika 6 Uklanjanje torki iz tabele DDL komandom TRUNCATE.....	29
Slika 7 Izlistavanje torki iz tabele DML komandom SELECT .....	30
Slika 8 Modifikacija torke DML komandom UPDATE .....	31
Slika 9 Dodavanje torke u tabelu DML komandom INSERT INTO .....	31
Slika 10 Brisanje torke iz tabele DML komandom DELETE FROM.....	32
Slika 11 Primer SQL upita .....	33
Slika 12 Primer kreiranja i brisanja šeme baze podataka .....	34
Slika 13 Primer kreiranja i brisanja tebele baze podataka .....	35
Slika 14 Primer kreiranja i brisanja sekvence baze podataka .....	35
Slika 15 Primer kreiranja i brisanja kursora baze podataka.....	37
Slika 16 Rezultat prvog upita .....	38
Slika 17 Rezultat drugog upita .....	39
Slika 18 Rezultat trećeg upita .....	40
Slika 19 Rezultat četvrtog upita .....	41
Slika 20 Rezultat petog upita .....	42
Slika 21 Pozitivan rezultat prve funkcije .....	45
Slika 22 Negativan rezultat prve funkcije.....	45
Slika 23 Negativan rezultat prve funkcije.....	45
Slika 24 Pozitivan rezultat druge funkcije .....	48
Slika 25 Pozitivan rezultat druge funkcije .....	48
Slika 26 Pozitivan rezultat druge funkcije .....	48
Slika 27 Negativan rezultat druge funkcije .....	48
Slika 28 Negativan rezultat druge funkcije .....	48
Slika 29 Pozitivan rezultat prve procedure .....	51
Slika 30 Negativan rezultat prve procedure .....	51
Slika 31 Negativan rezultat prve procedure .....	52
Slika 32 Pozitivan rezultat druge procedure .....	54
Slika 33 Negativan rezultat druge procedure .....	54
Slika 34 Rezultat prvog trigera - uspešno izvršenje .....	56
Slika 35 Rezultat prvog trigera - neuspešno izvršenje .....	56
Slika 36 Rezultat drugog trigera - uspešno izvršenje .....	58
Slika 37 Rezultat drugog trigera - neuspešno izvršenje .....	58

## SADRŽAJ TABELA

Tabela 1 <i>Tip entiteta zaposleni</i> .....	9
Tabela 2 <i>Tip entiteta racunovodja</i> .....	10
Tabela 3 <i>Tip entiteta banka</i> .....	11
Tabela 4 <i>Tip entiteta racun</i> .....	11
Tabela 5 <i>Tip poveznika je_u (racun - banka)</i> .....	11
Tabela 6 <i>Tip entiteta predmet</i> .....	12
Tabela 7 <i>Tip entiteta spis</i> .....	13
Tabela 8 <i>Tip entiteta pravosnazna_sudska_presuda</i> .....	13
Tabela 9 <i>Tip poveznika vezan_za (racun - pravosnazna_sudska_presuda)</i> .....	14
Tabela 10 <i>Tip entiteta faktura</i> .....	14
Tabela 11 <i>Tip poveznika sadrzi (racun - faktura)</i> .....	15
Tabela 12 <i>Tip entiteta uplata</i> .....	15
Tabela 13 <i>Tip entiteta uplata_po_sudskom_resenju</i> .....	15
Tabela 14 <i>Tip entiteta uplata_po_fakturi</i> .....	16
Tabela 15 <i>Tip poveznika zavodi (racunovodja - uplata)</i> .....	16
Tabela 16 <i>Tip poveznika izda (racunovodja - faktura)</i> .....	16
Tabela 17 <i>Tip poveznika na_osnovu (uplata_po_sudskom_resenju - pravosnazna_sudska_presuda)</i> .....	16
Tabela 18 <i>Tip poveznika po (faktura – uplata_po_fakturi)</i> .....	17
Tabela 19 <i>Tip entiteta institucije</i> .....	17
Tabela 20 <i>Tip poveznika je_upucena (uplata_po_fakturi - institucije)</i> .....	18
Tabela 21 <i>Tip poveznika poslata_je (uplata_po_sudskom_resenju - institucije)</i> .....	18
Tabela 22 <i>Tip entiteta fizicko_lice</i> .....	19
Tabela 23 <i>Tip entiteta pravno_lice</i> .....	20
Tabela 24 <i>Tip entiteta stranka</i> .....	21
Tabela 25 <i>Tip entiteta klijent</i> .....	21
Tabela 26 <i>Tip poveznika placa (klijent - uplata_po_fakturi)</i> .....	21

## UVOD

Projekat predstavljen u ovoj dokumentaciji podrazumeva razvoj baze podataka za potrebe rada jednog dela advokatske kancelarije, odnosno razvoj baze podataka koja će korisnicima pružiti mogućnost praćenja svih finansijskih transakcija i ostalih računovodstvenih poslova u okviru jedne advokatske kancelarije.

Kako u svim javnim i privatnim ustanovama postoje odeljenja koja se bave finansijskim poslovima, tako i u okviru advokatskih kancelarija postoji odeljenje koje se bavi tim poslovima. Svesni smo razvoja informacionih tehnologija u finansijskom svetu, pa je zbog toga potrebno pratiti trendove i ići u korak sa svetom. Zbog toga je potrebno automatizovati finansijsko poslovanje u okviru advokatskih kancelarija koje će pomoći i ubrzati rad ostalih zaposlenih kako u okviru advokatske kancelarije tako i eksternih korisnika koji koriste usluge advokatske kancelarije.

U daljem tekstu, priložen je ER model podataka, čijim prevođenjem je kreiran relacioni model podataka. Takođe, priložen je i tabelarni prikaz ograničenja pojave tipa, integriteta tipa i jedinstvenih vrednosti. Na samom kraju, na osnovu relacionog modela podataka, izvršena je implementacija šeme baze podataka, gde su kreirane tabele i ograničenja, kao i funkcije, kursori, procedure i trigeri.

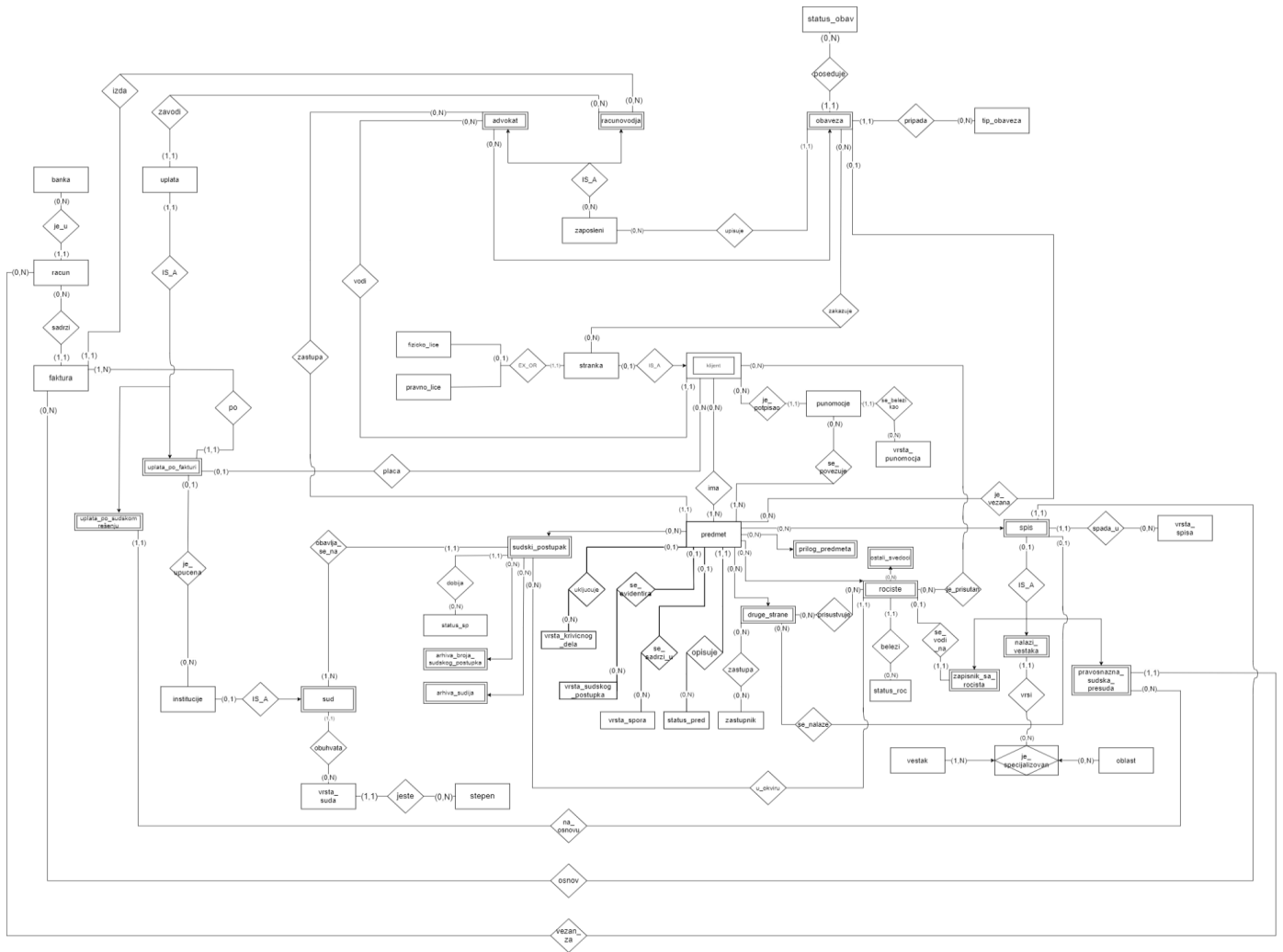
## 1. ANALIZA PROGRAMSKOG DOMENA

Svrha finansijskog odeljenja u advokatskoj kancelariji je da upravlja finansijskim resursima i pruži podršku u finansijskim operacijama specifičnim za advokatsku profesiju. Ovo odeljenje obavlja različite zadatke kako bi se osigurala efikasna finansijska uprava operacijama i održavanje finansijske stabilnosti advokatske kancelarije. Neki od glavnih zadataka finansijskog odeljenja u advokatskoj kancelariji uključuju:

- Računovodstvo: Voditi evidenciju o prihodima i troškovima kancelarije, pripremiti finansijske izveštaje i pratiti finansijsko stanje kancelarije.
- Naplata usluga: Upravljanje procesom fakturisanja klijentima za pružanje pravne usluge i osiguravanje pravovremene naplate.
- Upravljanje klijentskim računima: Praćenje i upravljanje novčanim tokom od klijenata, upravljanje depozitima, praćenje salda klijentskih računa i pridržavanje relevantnih zakonskih propisa.
- Finansijska analiza: Analiziranje finansijskih podataka, ocenjivanje profitabilnosti pravnih slučajeva, identifikacija trendova i pružanje relevantnih informacija za donošenje odluka.

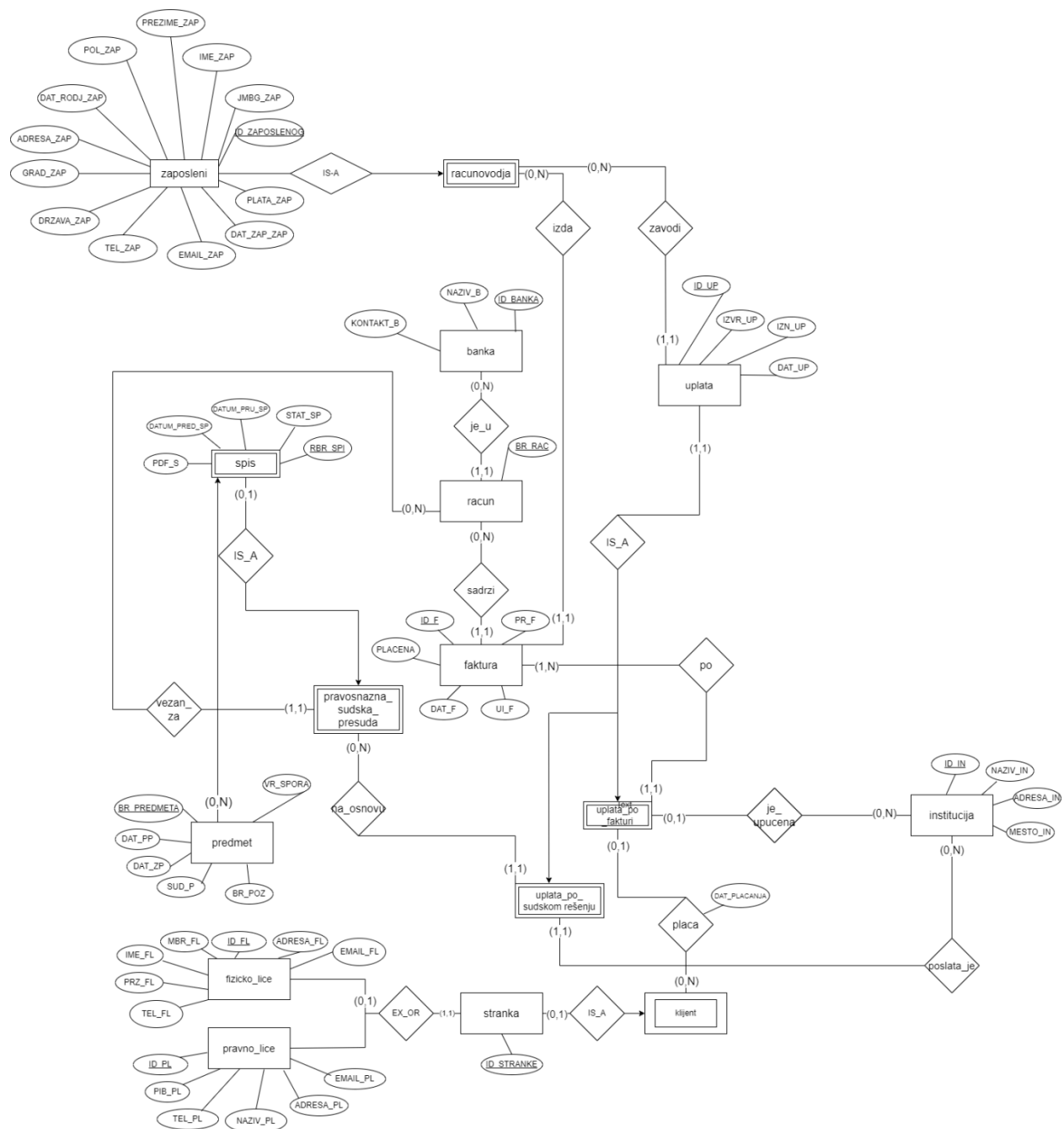
Finansijsko odeljenje u advokatskoj kancelariji igra ključnu ulogu u pružanju podrške advokatima i menadžmentu kancelarije u finansijskom upravljanju, planiranju i donošenju odluka. Njihov rad pomaže u osiguravanju efikasnog upravljanja finansijama kancelarije, pridržavanju zakonskih propisa i obezbeđivanju finansijske održivosti i uspeha kancelarije.

## 2. ER MODEL



**Slika 1 ER šema cele baze podataka (na nivou naziva tipova)**

### 3. ER MODEL PODŠEME



Slika 2 ER podšema Uplata



U okviru sistema vodi se evidencija izlaznih faktura i uplata. Advokatska kancelarija može imati račune u raznim bankama. Za svaku fakturu se navodi broj fakture, primalac fakture, datum fakturisanja, status fakture i ukupan iznos fakture koji je potrebno platiti advokatskoj kancelariji. S druge strane, uplata se može izvršiti na osnovu izdate fakture ili na osnovu izvršenja pravosnažne sudske presude. Zbog toga je kreirana IS-a hijerarhija od tipa entiteta *uplata* koja kao podtipove ima tipove entiteta *uplata\_po\_sudskom\_resenju* i *uplata\_po\_fakturi*. Pošto faktura može biti upućena nekoj instituciji ili klijentu, zbog toga je uplata po fakturi povezana sa klijentom i institucijom, pri čemu jedna faktura ne može biti istovremeno namenjena i instituciji i klijentu. Tip entiteta *uplata\_po\_sudskom\_resenju* je povezan sa tipom entiteta *pravosnazna\_sudska\_presuda* koja predstavlja spis koji je vezan za određeni predmet advokatske kancelarije na osnovu kog se može izvršiti uplata. Uplatu po sudskom rešenju može izvršiti neka od institucija. Na svakoj pravosnažnoj sudskoj presudi se navodi i broj računa na koji se može izvršiti uplata. Takođe, računovodstveni radnik advokatske kancelarije je zadužen za evidenciju izlaznih faktura i uplata što je i prikazano na podšemi. Stranka može biti fizičko ili pravno lice. Stranka postaje klijent samo ukoliko je potpisala odgovarajuće punomoćje.

## 4. TABELARNI PRIKAZ OBELEŽJA I OGRANIČENJA

Tabela 1 Tip entiteta zaposleni

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
1.	ID_ZAPOSLENOG	Identifikacioni broj zaposlenog	INTEGER	2	⊥	d >= 1
2.	JMBG_ZAP	Jedinstveni matični broj zaposlenog	VARCHAR	13	⊥	d >= 1
3.	IME_ZAP	Ime zaposlenog	VARCHAR	20	⊥	Δ
4.	PREZIME_ZAP	Prezime zaposlenog	VARCHAR	20	⊥	Δ
5.	POL_ZAP	Pol zaposlenog	VARCHAR	1	T	d ∈ {'M', 'Z'}
6.	DAT_RODJ_ZAP	Datum rođenja zaposlenog	DATE		⊥	Δ
7.	ADRESA_ZAP	Adresa zaposlenog (ulica i broj)	VARCHAR	50	⊥	Δ
8.	GRAD_ZAP	Grad u kom zaposleni živi	VARCHAR	20	⊥	Δ

9.	DRZAVA_ZAP	Država u kojoj zaposleni živi	VARCHAR	20	⊥	Δ
10.	TEL_ZAP	Broj telefona zaposlenog	VARCHAR	20	⊥	Δ
11.	EMAIL_ZAP	Email zaposlenog	VARCHAR	30	⊥	Δ
12.	DAT_ZAP_ZAP	Datum zapošljavanja zaposlenog	DATE		⊥	Δ
13.	PLATA_ZAP_ZAP	Plata zaposlenog	REAL	20	⊥	Δ
Ključ	K = {ID_ZAPOSLENOG}					
Ne postoje dva zaposlena sa istim JMBG-om. (Formalno iskazano u relacionom modelu)						

Tabela 2 Tip entiteta *racunovodja*

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

**Tabela 3** Tip entiteta *banka*

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
1.	<i>ID_BANKA</i>	Identifikacioni broj banke	INTEGER	2	⊥	d >= 1
2.	<i>NAZIV_B</i>	Naziv banke	VARCHAR	20	⊥	Δ
3.	<i>KONTAKT_B</i>	Kontakt banke	VARCHAR	20	T	Δ
Ključ	K = { <i>ID_BANKA</i> }					

**Tabela 4** Tip entiteta *racun*

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
1.	<i>BR_RAC</i>	Broj računa	VARCHAR	16	⊥	d >= 1
Ključ	K = { <i>BR_RAC</i> }					

**Tabela 5** Tip poveznika *je\_u* (*racun* - *banka*)

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
Ključ	K = { <i>BR_RAC</i> }					

Tabela 6 Tip entiteta *predmet*

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
1.	BR_PREDMETA	Identifikacioni broj predmeta	INTEGER	2	⊥	d >= 1
2.	DAT_PP	Datum pokretanja predmeta advokatske kancelarije	DATE		⊥	d <= datuma upisa podataka o pokretanju predmeta
3.	DAT_ZP	Datum zatvaranja predmeta advokatske kancelarije	DATE		⊥	d <= datuma upisa podataka o zatvaranju predmeta
4.	SUD_P	Predmet pokrenut pred sudom	BOOLEAN		⊥	Δ
5.	VR_SPORA	Vrednost spora	REAL	8	T	d > 0
6.	BR_POZ	Broj pozicije na kojoj se nalazi advokatska kancelarija u sudskom postupku	INTEGER	1	⊥	d ∈ {1,2,3}
Ključ	K = {BR_PREDMETA}					
SUD_P = FALSE => BR_POZ = NULL DAT_ZP ne sme biti pre DAT_PP						

Tabela 7 Tip entiteta spis

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
1.	<i>RBR_SPI</i>	Redni broj spisa	VARCHAR	4	⊥	$d \geq 1$
2.	<i>STAT_SP</i>	Status spisa	BOOLEAN		⊥	$\Delta$
3.	<i>DATUM_PRIJ_SP</i>	Datum prijema spisa	DATE		T	$\Delta$
4.	<i>DATUM_PRED_SP</i>	Datum predaje spisa	DATE		T	$\Delta$
5.	<i>PDF_S</i>	PDF verzija spisa	BLOB		T	$\Delta$
Ključ	$K = \{BR\_PREDMETA + RBR\_SPI\}$					

Tabela 8 Tip entiteta pravosnazna\_sudska\_presuda

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
Ključ	$K = \{BR\_PREDMETA + RBR\_SPI\}$					

Tabela 9 Tip poveznika **vezan\_za** (racun - pravosnazna\_sudska\_presuda)

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
Ključ	$K = \{BR\_PREDMETA + RBR\_SPI\}$					

Tabela 10 Tip entiteta **faktura**

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
1.	<i>ID_F</i>	Identifikacioni broj fakture	INTEGER	3	⊥	$d \geq 1$
2.	<i>PR_F</i>	Primalac fakture	VARCHAR	20	⊥	$\Delta$
3.	<i>DAT_F</i>	Datum fakturisanja	DATE		⊥	$d \leq$ datuma unosa podataka
4.	<i>UI_F</i>	Ukupan iznos fakture	INTEGER	10	⊥	$\Delta$
5.	<i>PLACENA</i>	Status fakture	BOOLEAN		⊥	$\Delta$
Ključ	$K = \{ID\_F\}$					

**Tabela 11** Tip poveznika *sadrzi* (racun - faktura)

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
Ključ	K = {ID_F}					

**Tabela 12** Tip entiteta *uplata*

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
1.	ID_UP	Identifikacioni broj uplate	VARCHAR	10	⊥	d >= 1
2.	IZVR_UP	Izvršilac uplate	VARCHAR	50	⊥	Δ
3.	IZN_UP	Iznos uplate	INTEGER	10	⊥	d > 0
4.	DAT_UP	Datum uplate	DATE		⊥	d <= datuma unosa podataka
Ključ	K = {ID_UP}					

**Tabela 13** Tip entiteta *uplata\_po\_sudskom\_resenju*

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
------------	----------------	---------------	-------------	----------------	------	-------



Ključ	K = {ID_UP}
-------	-------------

**Tabela 14** Tip entiteta *uplata\_po\_fakturi*

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
Ključ	K = {ID_UP}					

**Tabela 15** Tip poveznika *zavodi* (racunovodja - uplata)

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
Ključ	K = {ID_UP}					

**Tabela 16** Tip poveznika *izda* (racunovodja - faktura)

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
Ključ	K = {ID_F}					

**Tabela 17** Tip poveznika *na\_osnovu* (uplata\_po\_sudskom\_resenju - pravosnazna\_sudska\_presuda)

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
Ključ	K = {ID_UP}					

Tabela 18 Tip poveznika *po* (faktura – uplata\_po\_fakturi)

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
Ključ	K = {ID_UP}					

Tabela 19 Tip entiteta *institucije*

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
1.	ID_IN	Identifikacioni broj institucije	INTEGER	2	⊥	d >= 1
2.	NAZIV_IN	Naziv institucije	VARCHAR	40	⊥	Δ
3.	ADRESA_IN	Adresa institucije	VARCHAR	50	⊥	Δ
4.	MESTO_IN	Mesto u kom se nalazi institucija	VARCHAR	30	⊥	Δ
Ključ	K = {ID_IN}					

**Tabela 20** Tip poveznika *je\_upucena* (uplata\_po\_fakturi - institucije)

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
Ključ	K = {ID_UP}					

**Tabela 21** Tip poveznika *poslata\_je* (uplata\_po\_sudskom\_resenju - institucije)

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
Ključ	K = {ID_UP}					

Tabela 22 Tip entiteta *fizicko\_lice*

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
1.	ID_FL	Identifikacioni broj fizičkog lica	INTEGER	2	⊥	d >= 1
2.	MBR_FL	Matični broj fizičkog lica	INTEGER	13	T	Δ
3.	IME_FL	Ime fizičkog lica	VARCHAR	20	⊥	Δ
4.	PRZ_FL	Prezime fizičkog lica	VARCHAR	20	⊥	Δ
5.	TEL_FL	Kontakt fizičkog lica	VARCHAR	20	T	Δ
6.	EMAIL_FL	Email fizičkog lica	VARCHAR	40	T	Δ
7.	ADRESA_FL	Adresa fizičkog lica	VARCHAR	50	T	Δ
Ključ	K = {ID_FL}					
Ne sme postojati fizičko lice sa istim matičnim brojem, email-om ili telefonom.						

Tabela 23 Tip entiteta *pravno\_lice*

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
1.	ID_PL	Identifikacioni broj pravnog lica	INTEGER	3	⊥	d >= 1
2.	PIB_PL	PIB pravnog lica	INTEGER	20	⊥	Δ
3.	NAZIV_PL	Naziv pravnog lica	VARCHAR	30	⊥	Δ
4.	TEL_FL	Kontakt pravnog lica	VARCHAR	20	T	Δ
5.	EMAIL_FL	Email pravnog lica	VARCHAR	40	T	Δ
6.	ADRESA_FL	Adresa pravnog lica	VARCHAR	50	T	Δ
Ključ	K = {ID_PL}					
Ne sme postojati pravno lice sa istim nazivom, PIB-om, telefonom ili email-om.						

**Tabela 24** Tip entiteta *stranka*

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
1.	<i>ID_STRANKE</i>	Identifikacioni broj stranke	VARCHAR	5	⊥	d >= 1
Ključ	K = { <i>ID_STRANKE</i> }					

**Tabela 25** Tip entiteta *klijent*

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
Ključ	K = { <i>ID_STRANKE</i> }					

**Tabela 26** Tip poveznika *placa* (klijent - uplata\_po\_fakturi)

Redni broj	Naziv obeležja	Opis obeležja	Tip podatka	Dužina podatka	Null	Uslov
Ključ	K = { <i>ID_UP</i> }					

## 5. RELACIONI MODEL

Relacioni model podataka u ovom projektu je sledeći:

**zaposleni** ( $\{ID\_ZAPOSLENOG, JMBG\_ZAP, IME\_ZAP, PREZIME\_ZAP, POL\_ZAP, DAT\_RODJ\_ZAP, ADRESA\_ZAP, GRAD\_ZAP, DRZAVA\_ZAP, TEL\_ZAP, EMAIL\_ZAP, DAT\_ZAP\_ZAP, PLATA\_ZAP\}, \{ID\_ZAPOSLENOG\}$ )

$\text{dom}(ID\_ZAPOSLENOG\_R) \subseteq \text{dom}(ID\_ZAPOSLENOG)$

**racunovodja** ( $\{ID\_ZAPOSLENOG\_R\}, \{ID\_ZAPOSLENOG\_R\}$ )

“IS-A zaposleni”

$\text{racunovodja}[ID\_ZAPOSLENOG\_R] \subseteq \text{zaposleni}[ID\_ZAPOSLENOG]$

**banka** ( $\{ID\_BANKA, NAZIV\_B, KONTAKT\_B\}, \{ID\_BANKA\}$ )

**racun** ( $\{BR\_RAC, ID\_BANKA\}, \{BR\_RAC\}$ )

“je u”

$\text{racun}[ID\_BANKA] \subseteq \text{banka}[ID\_BANKA]$

$\text{Null}(\text{racun}, ID\_BANKA) = \perp$

**predmet** ( $\{BR\_PREDMETA, DAT\_PP, DAT\_ZP, SUD\_P, VR\_SPORA, BR\_POZ, ID\_ZAPOSLENOG\_A, ID\_STATP, ID\_VSPOR, ID\_VSP, ID\_KD\}, \{BR\_PREDMETA\}$ )

**spis** ( $\{RBR\_SPI, STAT\_SP, DATUM\_PRIJ\_SP, DATUM\_PRED\_SP, PDF\_S, BR\_PREDMETA, ID\_VS, ID\_F, RBR\_DS\}, \{BR\_PREDMETA + RBR\_SPI\}$ )

“identifikaciona zavisnost predmet - spis”

$$\text{spis } [BR\_PREDMETA] \subseteq \text{predmet } [BR\_PREDMETA]$$

$$\text{dom } (RBR\_SPI\_PSP) \subseteq \text{dom } (RBR\_SPI)$$

$$\text{dom } (BR\_PREDMETA\_PSP) \subseteq \text{dom } (BR\_PREDMETA)$$

**pravosnazna\_sudska\_presuda** ( $\{RBR\_SPI\_PSP, BR\_PREDMETA\_PSP, BR\_RAC\}$ ,  
 $\{BR\_PREDMETA\_PSP + RBR\_SPI\_PSP\}$ )

“IS-A spis”

$$\text{pravosnazna\_sudska\_presuda } [BR\_PREDMETA\_PSP + RBR\_SPI\_PSP] \subseteq \\ \text{spis } [BR\_PREDMETA + RBR\_SPI]$$

“vezan za”

$$\text{pravosnazna\_sudska\_presuda } [BR\_RAC] \subseteq \text{racun } [BR\_RAC]$$

$$\text{Null } (\text{pravosnazna\_sudska\_presuda}, BR\_RAC) = \perp$$

**faktura** ( $\{ID\_F, PR\_F, DAT\_F, UI\_F, PLACENA, ID\_ZAPOSLENOG\_R, BR\_RAC\}$ ,  
 $\{ID\_F\}$ )

“sadrzi”

$$\text{faktura } [BR\_RAC] \subseteq \text{racun } [BR\_RAC]$$

$$\text{Null } (\text{faktura}, BR\_RAC) = \perp$$

**uplata** ( $\{ID\_UP, IZVR\_UP, IZN\_UP, DAT\_UP, ID\_ZAPOSLENOG\_R\}$ ,  $\{ID\_UP\}$ )

$$\text{dom } (ID\_UPS) \subseteq \text{dom } (ID\_UP)$$

$$\text{dom } (ID\_UPF) \subseteq \text{dom } (ID\_UP)$$



***uplata\_po\_fakturi*** ( $\{ID\_UPF, ID\_F, ID\_STRANKE\_K, ID\_IN\}, \{ID\_UPF\}$ )

$$(ID\_IN = \omega \wedge ID\_STRANKE\_K \neq \omega) \vee (ID\_IN \neq \omega \wedge ID\_STRANKE\_K = \omega)$$

***uplata\_po\_sudskom\_resenju*** ( $\{ID\_UPS, BR\_PREDMETA\_PSP, RBR\_SPI\_PSP\}, \{ID\_UPS\}$ )

“IS-A uplata”

$$uplata\_po\_fakturi [ID\_UPF] \subseteq uplata [ID\_UP]$$

$$uplata\_po\_sudskom\_resenju [ID\_UPS] \subseteq uplata [ID\_UP]$$

$$uplata\_po\_sudskom\_resenju [ID\_UPS] \cap uplata\_po\_fakturi [ID\_UPF] = \emptyset$$

$$uplata [ID\_UP] \subseteq uplata\_po\_sudskom\_resenju [ID\_UPS] \cup uplata\_po\_fakturi [ID\_UPF]$$

“zavodi”

$$uplata [ID\_ZAPOSLENOG\_R] \subseteq racunovodja [ID\_ZAPOSLENOG\_R]$$

$$\text{Null}(uplata, ID\_ZAPOSLENOG\_R) = \perp$$

“izda”

$$faktura [ID\_ZAPOSLENOG\_R] \subseteq racunovodja [ID\_ZAPOSLENOG\_R]$$

$$\text{Null}(faktura, ID\_ZAPOSLENOG\_R) = \perp$$

“po”

$$uplata\_po\_fakturi [ID\_UPF] \subseteq faktura [ID\_F]$$

$$faktura [ID\_F] \subseteq uplata\_po\_fakturi [ID\_UPF]$$

$$\text{Null}(uplata\_po\_fakturi, ID\_F) = \perp$$

“na osnovu”

$uplata\_po\_sudskom\_resenju [BR\_PREDMETA\_PSP + RBR\_SPI\_PSP] \subseteq$   
 $pravosnazna\_sudska\_presuda [BR\_PREDMETA\_PSP + RBR\_SPI\_PSP]$

$Null (uplata\_po\_sudskom\_resenju, BR\_PREDMETA\_PSP) = \perp$

$Null (uplata\_po\_sudskom\_resenju, RBR\_SPI\_PSP) = \perp$

**institucije** ( $\{ID\_IN, NAZIV\_IN, ADRESA\_IN, MESTO\_IN\}, \{ID\_IN\}$ )

“je upucena”

$uplata\_po\_fakturi [ID\_IN] \subseteq institucije [ID\_IN]$

“poslata je”

$institucije [ID\_IN] \subseteq uplata\_po\_sudskom\_resenju [ID\_IN]$

$Null (uplata\_po\_sudskom\_resenju, ID\_IN) = \perp$

**fizicko\_lice** ( $\{ID\_FL, MBR\_FL, IME\_FL, PRZ\_FL, TEL\_FL, EMAIL\_FL, ADRESA\_FL\},$   
 $\{ID\_FL\}$ )

**pravno\_lice** ( $\{ID\_PL, PIB\_PL, TEL\_PL, EMAIL\_PL, ADRESA\_PL, NAZIV\_PL\}, \{ID\_PL\}$ )

**stranka** ( $\{ID\_STRANKE, ID\_FL, ID\_PL\}, \{ID\_STRANKE\}$ )

“kategorizacija stranka”

$stranka [ID\_FL] \subseteq fizicko\_lice [ID\_FL]$

$stranka [ID\_PL] \subseteq pravno\_lice [ID\_PL]$

$(ID\_PL = \omega \wedge ID\_FL \neq \omega) \vee (ID\_PL \neq \omega \wedge ID\_FL = \omega)$

$dom (ID\_STRANKE\_K) \subseteq dom (ID\_STRANKE)$

**klijent** ( $ID\_STRANKE\_K, ID\_ZAPOSLENOG\_A\}, \{ID\_STRANKE\_K\}$ )

“IS-A stranka”

$klijent [ID\_STRANKE\_K] \subseteq stranka [ID\_STRANKE]$

“placa”

$uplata\_po\_fakturi [ID\_STRANKE\_K] \subseteq klijent [ID\_STRANKE\_K]$

## 6. DDL

Jezik za definisanje podataka (DDL) je računarski jezik koji se koristi za kreiranje i modifikaciju objekata u bazi podataka kao što su pogledi, šeme, tabele, indeksi, kolone, funkcije i uskladištene procedure.

Komande koje se najčešće koriste u okviru DDL-a jesu:

- **CREATE**,
- **ALTER**,
- **DROP** i
- **TRUNCATE**.

U nastavku će biti prikazani primeri korišćenja ovih komandi u okviru rada sa tabelom.

### Create

Ova komanda kreira novu tabelu uz pomoć unapred definisane sintakse.

Sintaksa komande CREATE je:

```
CREATE TABLE Tablename  
(  
  Columns1 datatype,  
  Columns2 datatype  
);
```

Na slici 3 prikazan je primer kreiranja tabele DDL komandom CREATE.

```
create table Projekat.Fizicko_lice  
(  
  ID_FL NUMERIC(5) NOT NULL,  
  MBR_FL NUMERIC(13),  
  IME_FL VARCHAR(20) NOT NULL,  
  PRZ_FL VARCHAR(20) NOT NULL,  
  TEL_FL VARCHAR(20),  
  EMAIL_FL VARCHAR(40),  
  ADRESA_FL VARCHAR(50),  
  CONSTRAINT PK_FIZ_LICE PRIMARY KEY (ID_FL)  
);
```

Slika 3 Kreiranje tabele DDL komandom CREATE

## Alter

Ova komanda menja postojeću tabelu baze podataka. Takođe, pomoću nje je moguće dodati nove kolone, izbaciti postojeće kolone ili promeniti tip podataka kolona u okviru tabele baze podataka.

Sintaksa ALTER komande za dodavanje nove kolone je:

```
ALTER TABLE tablename  
ADD columns1 datatype;
```

Na slici 4 prikazan je primer dodavanje nove kolone u tabelu pomoću DDL naredbe komandom ALTER.

```
alter table Projekat.Uplata  
add IZVR_UP varchar(50) not null
```

Slika 4 Dodavanje kolone DDL komandom ALTER

## Drop

Ova komanda se koristi za brisanje objekata u bazi podataka. Naredba komande DROP se ne može vratiti, tako da kada se objekat jednom uništi, više ga nije moguće vratiti.

Sintaksa DROP komande za brisanje tabele je:

```
DROP TABLE Tablename;
```

Na slici 5 prikazano je brisanje tabele pomoću DDL komande DROP.

```
if object_id('Projekat.Uplata_po_sudskom_resenju') is not null  
drop table Projekat.Uplata_po_sudskom_resenju
```

Slika 5 Brisanje tabele DDL komandom DROP

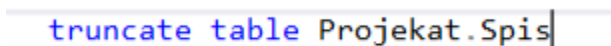
## Truncate

Komanda TRUNCATE se koristi za uklanjanje svih torki iz tabele. Za razliku od komande DROP, komanda TRUNCATE očuvava strukturu tabele.

Sintaksa TRUNCATE komande je:

**TRUNCATE TABLE** Tablename;

Na slici 6 prikazano je uklanjanje svih torki iz tabele DDL komandom TRUNCATE.

A screenshot of a text editor showing the SQL command 'truncate table Projekat.Spis' in a blue monospace font. A vertical cursor is positioned at the end of the command, after the period. The text is highlighted with a light blue background.

```
truncate table Projekat.Spis|
```

Slika 6 Uklanjanje torki iz tabele DDL komandom TRUNCATE

## 7. DML

Jezik za manipulaciju podacima (DML) je računarski jezik koji uključuje komande koje dozvoljavaju korisnicima manipulisanje podacima u bazi podataka. Manipulacija podacima podrazumeva insertovanje podataka u tabele baze podataka, izlistavanje postojećih podataka, kao i brisanje podataka iz postojećih tabela i modifikaciju istih. DML je uglavnom ugrađen u SQL baze podataka.

Komande koje se koriste u okviru DML jezika su:

- **SELECT,**
- **UPDATE,**
- **INSERT INTO i**
- **DELETE FROM.**

### Select

Komanda SELECT se koristi za izlistavanje redova iz tabela baze podataka.

Sintaksa SELECT komande je:

```
SELECT column name(s)  
FROM tablename  
WHERE conditions;
```

Na slici 7 prikazano je izlistavanje redova iz tabele DML komandom SELECT.

```
select ID_BANKA, NAZIV_B, KONTAKT_B from Projekat.Banka
```

Slika 7 Izlistavanje torki iz tabele DML komandom SELECT

### Update

Komanda UPDATE se koristi za modifikovanje postojećih podataka u tabeli baze podataka.

Sintaksa UPDATE komande je:

**UPDATE** tablename  
**SET** column name = value  
**WHERE** conditions;

Na slici 8 prikazan je primer modifikovanja tabele DML komandom UPDATE.

```
update Projekat.Banka  
set KONTAKT_B='062/5985487'  
where ID_BANKA=3
```

Slika 8 Modifikacija torke DML komandom UPDATE

### Insert into

Komanda INSERT INTO dodaje jedan ili više zapisa u tabelu baze podataka.  
Sintaksa komande INSERT INTO je:

**INSERT INTO** tablename  
**VALUES** (value1, value2, value3...);

Na slici 9 prikazan je primer unosa torke u tabelu DML komandom INSERT INTO.

```
insert into Projekat.Banka  
values ('4', 'Komercijalna banka', '062/7895648')
```

Slika 9 Dodavanje torke u tabelu DML komandom INSERT INTO

### Delete from

Komanda DELETE FROM uklanja jedan ili više zapisa iz tabele u zavisnosti od postavljenih uslova.

Sintaksa komande DELETE FROM je:



**DELETE FROM** tablename  
[**WHERE** condition]

Na slici 10 prikazan je primer brisanja torke iz tabele DML komandom DELETE FROM.

```
delete from Projekat.Banka  
where ID_BANKA =1
```

Slika 10 Brisanje torke iz tabele DML komandom DELETE FROM

## 8. SQL

SQL (Structured Query Language) je standardni relacioni upitni jezik (ANSI i ISO standard) koji se zasniva na relacionoj algebri i relacionom računu. SQL je implementiran u praktično sve komercijalne SUBP, i mada svaki proizvođač ima svoj dijalekt SQL-a rad sa svim verzijama je gotovo identičan.

Osnovni nedostatak SQL paketa ogleda se u činjenici da sve njegove mogućnosti i opcije ne mogu savladati korisnici – laici u smislu projektovanja, eksploatacije, obrade i neposrednog korišćenja informacionog sistema. SQL, naime, zahteva od korisnika poznavanje konfiguracije informacionog sistema (logički model baze, odnosno spisak relacija, veza među njima, kao i raspored atributa u njima), te poznavanje osnova tehnike programiranja.

Na slici 11 prikazan je jedan primer SQL upita.

```
--select k.ID_STRANKE_K as 'ID klijenta', f1.MBR_FL as 'Maticni broj klijenta', f1.IME_FL + ' ' +f1.PRZ_FL as 'Ime i prezime klijenta', f1.TEL_FL as 'Broj telefona',
isnull(f1.EMAIL_FL, 'Email klijenta nije poznat') as 'Email klijenta', f1.ADRESA_FL as 'Adresa klijenta',
avg(u.IZN_UP) as 'Prosecan iznos uplata', count(upf.ID_STRANKE_K) as 'Ukupan broj svih uplata po fakturi'
from Projekat.Fizicko_lice fl join Projekat.Stranka s on fl.ID_FL=s.ID_FL
join Projekat.Klijent k on s.ID_STRANKE = k.ID_STRANKE_K
join Projekat.Uplata_po_fakturi upf on k.ID_STRANKE_K=upf.ID_STRANKE_K
join Projekat.Uplata u on upf.ID_UPF=u.ID_UP
group by k.ID_STRANKE_K, f1.MBR_FL, f1.IME_FL, f1.PRZ_FL, f1.TEL_FL, f1.EMAIL_FL, f1.ADRESA_FL
having count(upf.ID_STRANKE_K) > 1
```

Slika 11 Primer SQL upita

## 9. OBJEKTI

Objekat baze podataka u relacionoj bazi podataka je struktura podataka koja se koristi za skladištenje ili referenciranje podataka. Najčešći objekat sa kojim korisnici komuniciraju su tabele.

U okviru projekta korišćeni su objekti poput šeme, tabela, sekvenci, funkcija, procedura, trigeri i kursora. U nastavku je predstavljen kratak opis svakog objekta uz primer.

**Šema** baze podataka definiše kako su podaci organizovani u okviru relacione baze podataka. Šema zapravo ne sadrži podatke već logička ograničenja kao što su imena tabela, polja, tipovi podataka i odnosi između ovih entiteta.

Na slici 12 prikazan je primer kreiranja i brisanja šeme baze podataka.

```
if SCHEMA_ID('Projekat') is not null
drop schema Projekat
go

create schema Projekat
go
```

Slika 12 Primer kreiranja i brisanja šeme baze podataka

**Tabele** su osnovne komponente relacione baze podataka, koje predstavljaju više redova strukturiranih elemenata podataka. Svaka pojedinačna stavka unesena u relacionu bazu podataka je predstavljena vrednošću u koloni.

Na slici 13 je prikazan primer kreiranja i brisanja tabele baze podataka.

```

if object_id('Projekat.Zaposleni') is not null
drop table Projekat.Zaposleni
go
create table Projekat.Zaposleni
(
    ID_ZAPOSLENOG numeric(2) not null,
    JMBG_ZAP varchar(13) constraint UC_JMBG unique (JMBG_ZAP) not null,
    IME_ZAP VARCHAR(20) NOT NULL,
    PREZIME_ZAP VARCHAR(20) NOT NULL,
    POL_ZAP VARCHAR(1) constraint CHK_POL_ZAPOSLENOG check (POL_ZAP IN ('M','Z')),
    DAT_RODJ_ZAP DATE NOT NULL,
    ADRESA_ZAP VARCHAR(50) NOT NULL,
    GRAD_ZAP VARCHAR(20) NOT NULL,
    DRZAVA_ZAP VARCHAR(20) NOT NULL,
    TEL_ZAP VARCHAR(20) NOT NULL,
    EMAIL_ZAP VARCHAR(30) NOT NULL,
    DAT_ZAP_ZAP DATE NOT NULL,
    PLATA_ZAP numeric(20) NOT NULL,
    CONSTRAINT PK_ID_ZAP PRIMARY KEY (ID_ZAPOSLENOG)
);

```

Slika 13 Primer kreiranja i brisanja tebele baze podataka

**Sekvence** su objekti baze podataka koji generišu jedinstvene brojeve u bazi podataka. One se koriste za automatsko generisanje vrednosti, kao što su identifikatori ili ključevi. Sekvence omogućavaju obezbeđivanje jedinstvenih vrednosti čak i u okruženjima sa više korisnika ili više tabela.

Na slici 14 prikazan je primer kreiranja i brisanja sekvence baze podataka.

```

if object_id('Projekat.SekvencaFaktura') is not null
drop sequence Projekat.SekvencaFaktura
go
create sequence Projekat.SekvencaFaktura as numeric
start with 1
minvalue 1
increment by 1
no cycle

```

Slika 14 Primer kreiranja i brisanja sekvence baze podataka

**Korisnički definisane funkcije** u SQL-u su funkcije koje se mogu kreirati i prilagoditi prema potrebama korisnika. One omogućavaju programerima da definišu sopstvene funkcije sa specifičnom logikom koji mogu koristiti u svojim upitima.

Postoji dve vrste korisnički definisanih funkcija:

1. Skalarne funkcije: vraćaju pojedinačnu vrednost na osnovu zadatih parametara.
2. Tabelarne funkcije: vraćaju rezultat u obliku tabele.

U okviru dela projekta pod nazivom Funkcije prikazano je kreiranje i brisanje funkcije baze podataka.

**Uskladištene procedure** su blokovi SQL naredbi u sistemu za upravljanje relacionim bazama podataka i obično ih kreiraju programer, administrator baze podataka, analitičar podataka, a čuva se i ponovo koristi za više programa. Dve najvažnije uskladištene procedure koje se nalaze u bilo kom RDBMS-u su korisnički definisane uskladištene procedure i sistemski uskladištene procedure.

U okviru dela projekta pod nazivom Procedure prikazano je kreiranje i brisanje procedure baze podataka.

**Kursori** su objekti baze podataka koji omogućavaju programerima da iteriraju kroz rezultate upita i manipulišu tim rezultatima red po red. Kursori se mogu koristiti u okviru uskladištenih procedura, trigeri ili direktno u SQL upitima. Kada se koristi kursor, treba voditi računa o otvaranju, pomeranju kroz redove, izvršavanju operacija nad redovima i zatvaranju kursora kako bi se oslobodili resursi.

Na slici 15 prikazan je primer kreiranja i brisanja kursora.

```

open procedura_cursor
set @counter=1
fetch next from procedura_cursor into @idUp, @iznos, @datUp, @idFak, @datFak, @brRac, @banka
while @@fetch_status = 0
begin
    print convert(varchar, @counter) + '. ID uplate: ' + convert(varchar, @idUp) + ', iznos: ' + convert(varchar, @iznos) + ', datum uplate: ' + convert(varchar, @datUp) + ', ID fakture: ' + convert(varchar, @idFak) + ', datum fakturisanja: ' + convert(varchar, @datFak) + ', broj racuna: ' + @brRac + ', banka: ' + @banka
    set @sum=@sum+@iznos
    fetch next from procedura_cursor into @idUp, @iznos, @datUp, @idFak, @datFak, @brRac, @banka
    set @counter=@counter+1
end
close procedura_cursor
deallocate procedura_cursor

```

Slika 15 Primer kreiranja i brisanja kursora baze podataka

**Trigeri** su objekti baze podataka koji automatski reaguju na određene događaje ili akcije u bazi podataka. Oni se koriste za postavljanje posebnih operacija koje se izvršavaju automatski kada se određeni događaj desi, kao što je unos, ažuriranje ili brisanje podataka u tabeli. Kada se aktivira odgovarajući događaj, trigger može izvršiti niz SQL instrukcija ili pokrenuti korisnički definisanu logiku koja može obuhvatiti kompleksne poslovne procese.

U okviru dela projekta pod nazivom Trigeri prikazano je kreiranje i brisanje triggera baze podataka.

Uz dokumentaciju priloženi su:

- script za kreiranje i brisanje šema, sekvenci i tabela projekta,
- script za unos podataka projekta,
- script sa SQL upitima projekta,
- script za kreiranje procedura projekta,
- script za kreiranje funkcija projekta i
- script za kreiranje triggera projekta.

## 10. UPITI

### Prvi upit

Upit izlistava za svakog klijenta koji je fizičko lice (ID, matični broj, ime i prezime, broj telefona, email i adresu) broj njegovih uplata po fakturi (samo ako je taj broj veći od 1) i prosečan iznos njegovih uplata po fakturi.

```
select k.ID_STRANKE_K as 'ID klijenta', fl.MBR_FL as 'Maticni broj klijenta', fl.IME_FL + ' ' + fl.PRZ_FL as 'Ime i prezime klijenta', fl.TEL_FL as 'Broj telefona', isnull(fl.EMAIL_FL, 'Email klijenta nije poznat') as 'Email klijenta', fl.ADRESA_FL as 'Adresa klijenta', avg(u.IZN_UP) as 'Prosecan iznos uplata', count(upf.ID_STRANKE_K) as 'Ukupan broj svih uplata po fakturi' from Projekat.Fizicko_lice fl join Projekat.Stranka s on fl.ID_FL=s.ID_FL join Projekat.Klijent k on s.ID_STRANKE = k.ID_STRANKE_K join Projekat.Uplata_po_fakturi upf on k.ID_STRANKE_K=upf.ID_STRANKE_K join Projekat.Uplata u on upf.ID_UPF=u.ID_UP group by k.ID_STRANKE_K, fl.MBR_FL, fl.IME_FL, fl.PRZ_FL, fl.TEL_FL, fl.EMAIL_FL, fl.ADRESA_FL having count(upf.ID_STRANKE_K) > 1
```

	ID klijenta	Maticni broj klijenta	Ime i prezime klijenta	Broj telefona	Email klijenta	Adresa klijenta	Prosecan iznos uplata	Ukupan broj svih uplata po fakturi
1	1	2403977456522	Milija Rosic	064/7774445	milijarosic1@advokatiNS.com	Puskinova 22, Novi Sad	4666.666666	3
2	13	2811968565512	Milena Stojicevic	064/8778475	milenaast@gmail.com	Kopernikova 16, Novi Sad	12900.000000	2
3	3	2709952768998	Sava Markovic	064/5555554	Email klijenta nije poznat	Fruskogorska 11, Novi Sad	5750.000000	2
4	6	110985765666	Aleksandar Savkovic	061/899854	savkovic123@gmail.com	Doza Djerdja 3, Novi Sad	13333.333333	3

Slika 16 Rezultat prvog upita

### Drugi upit

Upit izlistava sve fakture (ID, primalac, ukupan iznos i datum) iz poslednjeg kvartala 2022. godine čiji je iznos veći od prosečnog iznosa svih faktura u tom periodu, u jednoj koloni ispisuje da li je primalac fakture klijent ili institucija.

```
select fak.ID_F as 'ID fakture', fak.PR_F as 'Primalac fakture', fak.UI_F as 'Ukupan iznos fakture', fak.DAT_F as 'Datum fakturisanja', case when upf.ID_IN is not null then 'Institucija' when upf.ID_STRANKE_K is not null then 'Klijent' end as 'Vrsta primaoca' from Projekat.Faktura fak
```

```

join Projekat.Uplata_po_fakturi upf on fak.ID_F=upf.ID_F
where DATEPART(year, fak.DAT_F) = 2022 and DATEPART(MONTH, fak.DAT_F)
in (9,10,11,12)
and fak.UI_F>(select AVG(UI_F) from PROJEKAT.Faktura fak where
DATEPART(year, fak.DAT_F) = 2022 and DATEPART(MONTH, fak.DAT_F) in
(9,10,11,12))

```

	ID fakture	Primalac fakture	Ukupan iznos fakture	Datum fakturisanja	Vrsta primaoca
1	6	Preksajni sud u Novom Sadu	40000	2022-12-01	Institucija
2	1	Osnovni sud u Novom Sadu	45000	2022-12-21	Institucija
3	18	Inzenjering-Mirkovic	55000	2022-12-22	Klijent
4	5	Preksajni sud u Novom Sadu	120000	2022-11-21	Institucija

Slika 17 Rezultat drugog upita

### Treći upit

Upit izlistava za sve klijente koji su fizička lica (ID, matični broj, ime i prezime, broj telefona, email i adresu) ukupne iznose njihovih uplata po fakturi čija vrednost nije manja od 7000, sortirano rastuće po imenu. U jednoj koloni ispisuje da li je ukupna vrednost uplata veća od 30000, manja od 15000 ili se nalazi između.

```

select k.ID_STRANKE_K as 'ID klijenta', f1.MBR_FL as 'Maticni broj
klijenta', f1.IME_FL + ' ' +f1.PRZ_FL as 'Ime i prezime klijenta',
f1.TEL_FL as 'Broj telefona',
isnull(f1.EMAIL_FL, 'Email klijenta nije poznat') as 'Email klijenta',
f1.ADRESA_FL as 'Adresa klijenta', sum(u.IZN_UP) as 'Ukupan iznos svih
uplata po fakturi',
case
when sum(u.IZN_UP) >= 30000 then 'Ukupan iznos je veci od 30 000'
when sum(u.IZN_UP) <= 15000 then 'Ukupan iznos je manji od 15 000'
when sum(u.IZN_UP) < 30000 and sum(u.IZN_UP) > 15000 then 'Ukupan
iznos je manji od 30 000, a veci od 15 000'
end as 'Odnos uplata'
from Projekat.Fizicko_lice f1 join Projekat.Stranka s on
f1.ID_FL=s.ID_FL
join Projekat.Klijent k on s.ID_STRANKE = k.ID_STRANKE_K
join Projekat.Uplata_po_fakturi upf on k.ID_STRANKE_K=upf.ID_STRANKE_K
join Projekat.Uplata u on upf.ID_UPF=u.ID_UP
group by k.ID_STRANKE_K, f1.MBR_FL, f1.IME_FL, f1.PRZ_FL, f1.TEL_FL,
f1.EMAIL_FL, f1.ADRESA_FL
having sum(u.IZN_UP)>7000
order by f1.IME_FL asc

```



	ID klijenta	Matični broj klijenta	Ime i prezime klijenta	Broj telefona	Email klijenta	Adresa klijenta	Ukupan iznos svih uplata po fakturi	Odnos uplata
1	6	110985765666	Aleksandar Savkovic	061/899854	savkovic123@gmail.com	Doza Djerdja 3, Novi Sad	40000	Ukupan iznos je veci od 30 000
2	12	1201970854689	Jovan Stevic	060/7645584	stevicjovan@gmail.com	Pavla Papa 15, Novi Sad	25000	Ukupan iznos je manji od 30 000, a veci od 15 000
3	13	2811968565512	Milena Stojicevic	064/8778475	mlenast@gmail.com	Kopernikova 16, Novi Sad	25800	Ukupan iznos je manji od 30 000, a veci od 15 000
4	1	2403977456522	Milija Rosic	064/7774445	milijarosc1@advokatINS.com	Puskinova 22, Novi Sad	14000	Ukupan iznos je manji od 15 000
5	3	2709952768998	Sava Markovic	064/5555554	Email klijenta nije poznat	Fruskogorska 11, Novi Sad	11500	Ukupan iznos je manji od 15 000

Slika 18 Rezultat trećeg upita

## Četvrti upit

Upit za svakog klijenta (fizičko lice) koji je izvršio neku uplatu, prikazuje ID klijenta, njegovo ime i prezime, zatim ID uplate, datum i iznos njegove najveće uplate. Pored toga prikazuje broj računa i naziv banke u kojoj je izvršena uplata. Sve je sortirano rastuće po iznosu uplate.

```
select k.ID_STRANKE_K as 'ID klijenta', fl.IME_FL + ' ' + fl.PRZ_FL as
'Ime i prezime klijenta', up.ID_UP as 'ID uplate',
up.IZN_UP as 'Iznos uplate', try_convert(date, up.DAT_UP, 105) as
'Datum uplate', r.BR_RAC as 'Broj racuna', banka.NAZIV_B as 'Naziv
banke'
from Projekat.Fizicko_lice fl join Projekat.Stranka s on
fl.ID_FL=s.ID_FL
join Projekat.Klijent k on s.ID_STRANKE = k.ID_STRANKE_K
join Projekat.Uplata_po_fakturi upf on k.ID_STRANKE_K=upf.ID_STRANKE_K
join Projekat.Faktura f on upf.ID_F=f.ID_F
join Projekat.Racun r on f.BR_RAC=r.BR_RAC
join Projekat.Banka banka on r.ID_BANKA=banka.ID_BANKA
join Projekat.Uplata up on upf.ID_UPF=up.ID_UP
join (select k.ID_STRANKE_K, max(up.IZN_UP) maks from Projekat.Klijent
k join Projekat.Uplata_po_fakturi upf on
k.ID_STRANKE_K=upf.ID_STRANKE_K
join Projekat.Uplata up on upf.ID_UPF=up.ID_UP group by
k.ID_STRANKE_K) uplata
on k.ID_STRANKE_K=uplata.ID_STRANKE_K
where up.IZN_UP=uplata.maks
group by k.ID_STRANKE_K, fl.IME_FL, fl.PRZ_FL, up.DAT_UP, up.ID_UP,
up.IZN_UP, r.BR_RAC, banka.NAZIV_B
order by up.IZN_UP asc
```

	ID klijenta	Ime i prezime klijenta	ID uplate	Iznos uplate	Datum uplate	Broj racuna	Naziv banke
1	2	Darko Lazic	45687	1200	2022-04-29	2051475848965844	Komercijalna banka
2	14	Filip Bogdanovic	32112	6000	2023-03-30	1707457896554178	UniCredit banka
3	3	Sava Markovic	99878	7500	2023-04-10	2051475848965844	Komercijalna banka
4	1	Milija Rosic	14254	7500	2023-01-17	2051475848965844	Komercijalna banka
5	13	Milena Stojicevic	11114	15500	2022-10-07	1707457896554178	UniCredit banka
6	6	Aleksandar Savkovic	12774	20000	2022-08-08	2051475848965844	Komercijalna banka
7	12	Jovan Stevic	44441	25000	2022-09-30	1707457896554178	UniCredit banka

Slika 19 Rezultat četvrtog upita

## Peti upit

Upit izlistava dve uplate po fakturi (ID, izvršilac uplate, datum, iznos) od kojih jedna predstavlja najveću uplatu koju je izvršio klijent koji je fizičko lice, dok druga predstavlja najveću uplatu koju je izvršio klijent koji je pravno lice. U jednoj koloni ispisuje se da li je klijent fizičko ili pravno lice.

```

select u.ID_UP as 'ID uplate', u.IZVR_UP as 'Izvršilac uplate',
u.DAT_UP as 'Datum uplate', u.IZN_UP as 'Iznos uplate',
case
when s.ID_FL is null then 'Pravno lice'
when s.ID_PL is null then 'Fizicko lice'
end as 'Vrsta klijenta'
from Projekat.Uplata u
join Projekat.Uplata_po_fakturi upf on u.ID_UP=upf.ID_UPF
join Projekat.Klijent k on upf.ID_STRANKE_K=k.ID_STRANKE_K
join Projekat.Stranka s on k.ID_STRANKE_K=s.ID_STRANKE
JOIN
((select top(1) k.ID_STRANKE_K, max(u.IZN_UP) maxizn from
Projekat.Uplata u join Projekat.Uplata_po_fakturi upf on
u.ID_UP=upf.ID_UPF
join Projekat.Klijent k on upf.ID_STRANKE_K=k.ID_STRANKE_K
join Projekat.Stranka s on k.ID_STRANKE_K=s.ID_STRANKE
join Projekat.Fizicko_lice f on s.ID_FL=f.ID_FL
group by k.ID_STRANKE_K
order by maxizn desc)
union
(
select top(1) k.ID_STRANKE_K, max(u.IZN_UP) maxizn from
Projekat.Uplata u join Projekat.Uplata_po_fakturi upf on
u.ID_UP=upf.ID_UPF
join Projekat.Klijent k on upf.ID_STRANKE_K=k.ID_STRANKE_K
join Projekat.Stranka s on k.ID_STRANKE_K=s.ID_STRANKE

```

```

join Projekat.Pravno_lice p on s.ID_PL=p.ID_PL
group by k.ID_STRANKE_K
order by maxizn desc
)) maxPlata on k.ID_STRANKE_K= maxPlata.ID_STRANKE_K
where u.IZN_UP=maxizn

```

	ID uplate	Izvršilac uplate	Datum uplate	Iznos uplate	Vrsta klijenta
1	44441	Jovan Stevic	2022-09-30	25000	Fizicko lice
2	44578	Inzenjering-Mirkovic	2022-12-30	55000	Pravno lice

Slika 20 Rezultat petog upita

## 11. FUNKCIJE

### Prva funkcija

Funkcija na osnovu prosleđenog ID-ja klijenta ispisuje ukupan broj njegovih uplata po fakturi i iznos njegove najveće uplate. Ukoliko klijent nije imao nijednu uplatu ispisuje odgovarajuću poruku. Takođe, ukoliko ne postoji klijent sa prosleđenim ID-jem ispisuje odgovarajuću poruku.

```
if OBJECT_ID('Projekat.Funkcija1','FN') is not null
drop function Projekat.Funkcija1
go
create function Projekat.Funkcija1
(
    @id numeric(3)
)
returns varchar(500)
as
begin
    declare @brUplata as numeric(2)
    declare @iznos as numeric(10)
    declare @idKlij as numeric(3)
    declare @output as varchar(500)
    declare @ime as varchar(20)
    declare @prz as varchar(25)
    declare @fl as numeric(3)
    declare @pl as numeric(3)
    declare @naziv as varchar(40)

    set @idKlij =(select ID_STRANKE_K from Projekat.Klijent where
ID_STRANKE_K=@id)

    if(@idKlij is null)
    begin
        set @output = 'Klijent sa ID-jem ' + convert(varchar, @id) + ' ne
postoji.'
        return @output
    end
    else
    begin
        set @fl= (select s.ID_FL from Projekat.Stranka s join
Projekat.Klijent k on s.ID_STRANKE=k.ID_STRANKE_K where
k.ID_STRANKE_K=@id)
```

```

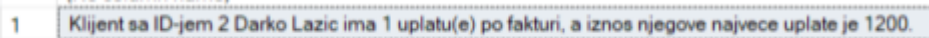
        set @pl= (select s.ID_PL from Projekat.Stranka s join
Projekat.Klijent k on s.ID_STRANKE=k.ID_STRANKE_K where
k.ID_STRANKE_K=@id)
        set @brUplata= (select count(ID_STRANKE_K) from
Projekat.Uplata_po_fakturi where ID_STRANKE_K=@id)
        if(@brUplata != 0)
        begin
            set @iznos=(select max(u.IZN_UP) from Projekat.Uplata u
join Projekat.Uplata_po_fakturi upf on u.ID_UP=upf.ID_UPF
WHERE upf.ID_STRANKE_K=@id)
            if(@f1 is not null)
            begin
                set @ime = (select f.IME_FL from Projekat.Fizicko_lice
f join Projekat.Stranka s on f.ID_FL=s.ID_FL join Projekat.Klijent k
on s.ID_STRANKE=K.ID_STRANKE_K where k.ID_STRANKE_K=@id)
                set @prz = (select f.PRZ_FL from Projekat.Fizicko_lice
f join Projekat.Stranka s on f.ID_FL=s.ID_FL join Projekat.Klijent k
on s.ID_STRANKE=K.ID_STRANKE_K where k.ID_STRANKE_K=@id)
                set @output= 'Klijent sa ID-jem ' + convert(varchar,
@id) + ' ' + convert(varchar,@ime) + ' ' + convert(varchar,@prz) + '
ima '
                    + convert(varchar, @brUplata) + ' uplatu(e) po
fakturi, a iznos njegove najveće uplate je '
                    + convert(varchar, @iznos) + '.'
                return @output
            end
        else if(@pl is not null)
        begin
            set @naziv = (select p.NAZIV_PL from
Projekat.Pravno_lice p join Projekat.Stranka s on p.ID_PL=s.ID_PL join
Projekat.Klijent k on s.ID_STRANKE=K.ID_STRANKE_K where
k.ID_STRANKE_K=@id)
            set @output= 'Klijent sa ID-jem ' + convert(varchar,
@id) + ' ' + @naziv + ' ima '
                    + convert(varchar, @brUplata) + ' uplatu(e) po
fakturi, a iznos njegove najveće uplate je '
                    + convert(varchar, @iznos) + '.'
            return @output
        end
    end
else
begin
    if(@f1 is not null)
    begin

```

```

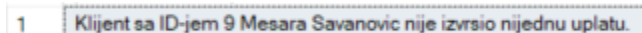
        set @ime = (select f.IME_FL from Projekat.Fizicko_lice
f join Projekat.Stranka s on f.ID_FL=s.ID_FL join Projekat.Klijent k
on s.ID_STRANKE=K.ID_STRANKE_K where k.ID_STRANKE_K=@id)
        set @prz = (select f.PRZ_FL from Projekat.Fizicko_lice
f join Projekat.Stranka s on f.ID_FL=s.ID_FL join Projekat.Klijent k
on s.ID_STRANKE=K.ID_STRANKE_K where k.ID_STRANKE_K=@id)
        set @output='Klijent sa ID-jem ' + convert(varchar,
@id) + ' ' + convert(varchar,@ime) + ' ' + convert(varchar,@prz) + '
nije izvršio nijednu uplatu.'
        return @output
    end
    else if(@pl is not null)
    begin
        set @naziv = (select p.NAZIV_PL from
Projekat.Pravno_lice p join Projekat.Stranka s on p.ID_PL=s.ID_PL join
Projekat.Klijent k on s.ID_STRANKE=K.ID_STRANKE_K where
k.ID_STRANKE_K=@id)
        set @output='Klijent sa ID-jem ' + convert(varchar,
@id) + ' ' + @naziv + ' nije izvršio nijednu uplatu.'
        return @output
    end
end
end
return @output
end
go

```



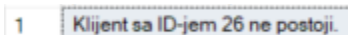
1	Klijent sa ID-jem 2 Darko Lazic ima 1 uplatu(e) po fakturi, a iznos njegove najveće uplate je 1200.
---	---

Slika 21 Pozitivan rezultat prve funkcije



1	Klijent sa ID-jem 9 Mesara Savanovic nije izvršio nijednu uplatu.
---	---

Slika 22 Negativan rezultat prve funkcije



1	Klijent sa ID-jem 26 ne postoji.
---	----------------------------------

Slika 23 Negativan rezultat prve funkcije

## Druga funkcija

Funkcija na osnovu prosleđenog broja fakture, proverava da li je izvršena uplata po toj fakturi, ukoliko nije ispisuje odgovarajuću poruku. Ukoliko jeste izvršena, ispisati ko je izvršilac uplate na osnovu fakture sa prosleđenim ID-jem i

da li je on pravno ili fizičko lice ili je institucija. Ukoliko faktura sa prosleđenim ID-jem ne postoji, ispisuje odgovarajuću poruku.

```
if OBJECT_ID('Projekat.FunkcijaII', 'FN') is not null
drop function Projekat.FunkcijaII
GO
create function Projekat.FunkcijaII
(
    @id as numeric(3)
)
returns varchar(500)
as
begin
    declare @poruka as varchar(500)
    declare @idF as numeric(3)
    declare @status as bit
    declare @klij as numeric(3)
    declare @inst as numeric(3)
    declare @ime as varchar(20)
    declare @prz as varchar(25)
    declare @fl as numeric(3)
    declare @pl as numeric(3)
    declare @in as numeric(3)
    declare @nazivPL as varchar(40)
    declare @nazivIns as varchar(40)

    set @idF = (select ID_F from Projekat.Faktura where ID_F=@id)
    set @status = (select PLACENA from Projekat.Faktura where
ID_F=@id)
    set @klij = (select ID_STRANKE_K from Projekat.Uplata_po_fakturi
where ID_F=@id)
    set @inst = (select ID_IN from Projekat.Uplata_po_fakturi where
ID_F=@id)

    if(@idF is null)
    begin
        set @poruka = 'Faktura sa ID-jem ' +convert(varchar, @id) +
' ne postoji.'
        return @poruka
    end
    else
    begin
        if(@status= 1)
        begin
```

```

        if(@klij is null and @inst is not null)
        begin
            set @nazivIns = (select distinct i.NAZIV_IN from
Projekat.Institucije i join Projekat.Uplata_po_fakturi upf on
i.ID_IN=upf.ID_IN where upf.ID_IN=@inst)
            set @poruka = 'ID fakture: ' + convert(varchar,
@id) + ', Institucija: ' + @nazivIns + '.'
            return @poruka
        end
        else
        begin
            set @f1 = (select distinct s.ID_FL from
Projekat.Stranka s join Projekat.Klijent K on
s.ID_STRANKE=K.ID_STRANKE_K join Projekat.Uplata_po_fakturi upf on
k.ID_STRANKE_K=upf.ID_STRANKE_K
                                where upf.ID_STRANKE_K=@klij)
            set @p1 = (select distinct s.ID_PL from
Projekat.Stranka s join Projekat.Klijent K on
s.ID_STRANKE=K.ID_STRANKE_K join Projekat.Uplata_po_fakturi upf on
k.ID_STRANKE_K=upf.ID_STRANKE_K
                                where upf.ID_STRANKE_K=@klij)

            if(@f1 is null and @p1 is not null)
            begin
                set @nazivPL = (select p1.NAZIV_PL from
Projekat.Pravno_lice p1 join Projekat.Stranka s on p1.ID_PL=s.ID_PL
where s.ID_PL= @p1)
                set @poruka = 'ID fakture: ' +
convert(varchar, @id) + ', Pravno lice: ' + @nazivPL + '.'
                return @poruka
            end
            else
            begin
                set @ime = (select f.IME_FL from
Projekat.Fizicko_lice f join Projekat.Stranka s on f.ID_FL=s.ID_FL
where s.ID_FL= @f1)
                set @prz = (select f.PRZ_FL from
Projekat.Fizicko_lice f join Projekat.Stranka s on f.ID_FL=s.ID_FL
where s.ID_FL= @f1)
                set @poruka = 'ID fakture: ' +
convert(varchar, @id) + ', Fizicko lice: ' + @ime + ' ' + @prz + '.'
                return @poruka
            end
        end
    end
end
else

```



```

begin
    set @poruka = 'Nije izvršena nijedna uplata na osnovu
fakture sa ID-jem ' + convert(varchar,@id) + '.'
    return @poruka
end
end
return @poruka
end
go

```

1	ID fakture: 1, Institucija: Osnovni sud u Novom Sadu.
---	---

Slika 24 Pozitivan rezultat druge funkcije

1	ID fakture: 2, Pravno lice: Eksport-import.
---	---

Slika 25 Pozitivan rezultat druge funkcije

1	ID fakture: 3, Fizicko lice: Milija Rosic.
---	--

Slika 26 Pozitivan rezultat druge funkcije

1	Faktura sa ID-jem 33 ne postoji.
---	----------------------------------

Slika 27 Negativan rezultat druge funkcije

1	Nije izvršena nijedna uplata na osnovu fakture sa ID-jem 23.
---	--

Slika 28 Negativan rezultat druge funkcije

## 12. PROCEDURE

### Prva procedura

Procedura na osnovu prosleđenog ID-ja klijenta izlistava sve njegove uplate po fakturi. Takođe, u ispisu se nalazi i broj fakture na osnovu koje je izvršena uplata kao i datum fakturisanja. Ispisuje se i broj računa i ime banke u kojoj je izvršena uplata. Na kraju ispisuje ukupan broj uplata klijenta sa prosleđenim ID-jem, kao i ukupan iznos svih njegovih uplata. (U okviru ove procedure korišćen je kursor koji prolazi kroz uplate koje je klijent izvršio.)

```
if object_id ('Projekat.Procedura1', 'P') is not null
drop proc Projekat.Procedura1
go
create procedure Projekat.Procedura1
    @id as numeric(3)
as
begin
    declare @idKlij as numeric(3)
    declare @ime as varchar(20)
    declare @prz as varchar(25)
    declare @naziv as varchar(25)
    declare @idUp as varchar(10)
    declare @iznos as numeric(10)
    declare @datUp as date
    declare @idFak as numeric(3)
    declare @datFak as date
    declare @brRac as varchar(16)
    declare @banka as varchar(30)
    declare @counter as int
    declare @sum as numeric(10) = 0
    set @idKlij = (select distinct ID_STRANKE_K from
Projekat.Uplata_po_fakturi where ID_STRANKE_K = @id)
    if (@id not in (select ID_STRANKE_K from Projekat.Klijent))
    begin
        print 'Klijent sa ID-jem ' + convert(varchar, @id) + ' ne
postoji.'
    end
    else
    begin
        if(@idKlij is null)
        begin
```

```

        print 'Klijent sa ID-jem ' + convert(varchar,@id) + '
nije izvršio nijednu uplatu.'
    end
    else
    begin
        declare @f1 as int = (select distinct s.ID_FL from
Projekat.Stranka s join Projekat.Klijent K on
s.ID_STRANKE=K.ID_STRANKE_K join
Projekat.Uplata_po_fakturi upf
on k.ID_STRANKE_K=upf.ID_STRANKE_K
where upf.ID_STRANKE_K=@id)
        declare @p1 as int= (select distinct s.ID_PL from
Projekat.Stranka s join Projekat.Klijent K on
s.ID_STRANKE=K.ID_STRANKE_K join
Projekat.Uplata_po_fakturi upf
on k.ID_STRANKE_K=upf.ID_STRANKE_K
where upf.ID_STRANKE_K=@id)
        if(@f1 is null and @p1 is not null)
        begin
            set @naziv = (select pl.NAZIV_PL from
Projekat.Pravno_lice pl join Projekat.Stranka s on pl.ID_PL=s.ID_PL
where s.ID_PL= @p1)
            print 'Klijent ' + @naziv + ' sa ID-jem ' +
convert(varchar,@id) + ' je izvršio sledece uplate:'
        end
        else
        begin
            set @ime = (select f.IME_FL from
Projekat.Fizicko_lice f join Projekat.Stranka s on f.ID_FL=s.ID_FL
where s.ID_FL= @f1)
            set @prz = (select f.PRZ_FL from
Projekat.Fizicko_lice f join Projekat.Stranka s on f.ID_FL=s.ID_FL
where s.ID_FL= @f1)
            print 'Klijent ' + @ime + ' ' +@prz + ' sa
ID-jem ' + convert(varchar,@id) + ' je izvršio sledece uplate:'
        end
        set @counter=0

        declare procedura_cursor CURSOR fast_forward for
        select u.ID_UP, u.IZN_UP, U.DAT_UP, f.ID_F,
f.DAT_f, r.BR_RAC, b.NAZIV_B
        from Projekat.Uplata u
        join Projekat.Uplata_po_fakturi upf on
u.ID_UP=upf.ID_UPF
        join Projekat.Faktura f on upf.ID_F=f.ID_F
        join Projekat.Racun r on f.BR_RAC=r.BR_RAC

```

```

join Projekat.Banka b on
r.ID_BANKA=b.ID_BANKA
where upf.ID_STRANKE_K=@id
open procedura_cursor
set @counter=1
fetch next from procedura_cursor into @idUp, @iznos,
@datUp, @idFak, @datFak, @brRac, @banka
while @@fetch_status = 0
begin
print convert(varchar, @counter) + '. ID
update: ' + convert(varchar, @idUp) + ', iznos: ' + convert(varchar,
@iznos) + ', datum update: ' + convert(varchar, @datUp)
+ ', ID fakture: ' + convert(varchar,
@idFak) + ', datum fakturisanja: ' + convert(varchar, @datFak) + ',
broj racuna: ' + @brRac + ', banka: ' + @banka
set @sum=@sum+@iznos
fetch next from procedura_cursor into
@idUp, @iznos, @datUp, @idFak, @datFak, @brRac, @banka
set @counter=@counter+1
end
close procedura_cursor
deallocate procedura_cursor

print 'Ukupno uplata: ' + convert(varchar,@counter-1)
print 'Ukupan iznos svih uplata: ' +
convert(varchar,@sum)

end
end
go

```

```

Klijent Aleksandar Savkovic sa ID-jem 6 je izvrrio sledece uplate:
1. ID uplate: 12774, iznos: 20000, datum update: 2022-08-08, ID fakture: 8, datum fakturisanja: 2022-08-02, broj racuna: 2051475848965844, banka: Komercijalna banka
2. ID uplate: 65455, iznos: 4000, datum update: 2022-02-08, ID fakture: 12, datum fakturisanja: 2022-02-03, broj racuna: 2051475848965844, banka: Komercijalna banka
3. ID uplate: 74565, iznos: 16000, datum update: 2022-10-16, ID fakture: 13, datum fakturisanja: 2022-10-10, broj racuna: 2051475848965844, banka: Komercijalna banka
Ukupno uplata: 3
Ukupan iznos svih uplata: 40000

Completion time: 2023-05-17T00:22:15.4750125+02:00

```

Slika 29 Pozitivan rezultat prve procedure

```

Klijent sa ID-jem 5 nije izvrrio nijednu uplatu.

Completion time: 2023-05-17T00:22:37.8329862+02:00

```

Slika 30 Negativan rezultat prve procedure

```
Klijent sa ID-jem 50 ne postoji.
```

```
Completion time: 2023-05-17T00:22:32.4331741+02:00
```

Slika 31 Negativan rezultat prve procedure

## Druga procedura

Procedura na osnovu prosleđenog ID-ja klijenta, klijentu kreira mejl koji će biti korišćen u radu sa advokatskom kancelarijom. Mejl je u formatu <naziv klijenta><ID klijenta>@advokatiNS.com (na primer [filipilic1@advokatiNS.com](mailto:filipilic1@advokatiNS.com)). Sva slova u mejlu pre ekstenzije @advokatiNS.com su mala.

```
if object_id('Projekat.Procedura2', 'P') is not null
drop proc Projekat.Procedura2
go
create procedure Projekat.Procedura2
    @id as numeric(3)
as
begin
    declare @ekstenzija as varchar(15)
    declare @email as varchar(60)
    set @ekstenzija = '@advokatiNS.com'
    if (@id not in (select ID_STRANKE_K from Projekat.Klijent))
    begin
        print 'Klijent sa ID-jem ' + convert(varchar, @id) + ' ne
postoji.'
    end
    else
    begin
        declare @p1 as numeric(3) = (select s.ID_PL from
Projekat.Stranka s join Projekat.Klijent k on
s.ID_STRANKE=k.ID_STRANKE_K where k.ID_STRANKE_K=@id)
        declare @f1 as numeric(3) = (select s.ID_FL from
Projekat.Stranka s join Projekat.Klijent k on
s.ID_STRANKE=k.ID_STRANKE_K where k.ID_STRANKE_K=@id)
        if(@p1 is null and @f1 is not null)
        begin
            declare @ime as varchar(20) = (select f.IME_FL from
Projekat.Fizicko_lice f join Projekat.Stranka s on f.ID_FL=s.ID_FL
where s.ID_FL= @f1)
```

```

        declare @prz as varchar(25) = (select f.PRZ_FL from
Projekat.Fizicko_lice f join Projekat.Stranka s on f.ID_FL=s.ID_FL
where s.ID_FL= @f1)
        declare @imeLower as varchar(20) = (select
lower(@ime))
        declare @przLower as varchar(25) = (select
lower(@prz))
        declare @emailStari as varchar(60) = (select EMAIL_FL
from Projekat.Fizicko_lice where ID_FL=@f1)
        set @email =
@imeLower+@przLower+convert(varchar,@id)+@ekstenzija

        update Projekat.Fizicko_lice
        set EMAIL_FL= @email
        where ID_FL= @f1

        print 'Klijent ' + convert(varchar, @id) + ':'
        print 'Stari mejl je ' + @emailStari + '.'
        print 'Novi mejl je ' + @email + '.'
    end
    else
    begin
        declare @naziv as varchar(25) = (select p.NAZIV_PL
from Projekat.Pravno_lice p join Projekat.Stranka s on p.ID_PL=s.ID_PL
where s.ID_PL= @p1)
        declare @nazivLower as varchar(25) = (select
lower(@naziv))
        declare @nazivReplace as varchar(25) = (select
replace(@nazivLower, ' ', '_'))
        declare @mailStari as varchar(60) = (select EMAIL_PL
from Projekat.Pravno_lice where ID_PL=@p1)
        set @email =
@nazivReplace+convert(varchar,@id)+@ekstenzija

        update Projekat.Pravno_lice
        set EMAIL_PL = @email
        where ID_PL=@p1

        print 'Klijent ' + convert(varchar, @id) + ':'
        print 'Stari mejl je ' + @mailStari + '.'
        print 'Novi mejl je ' + @email + '.'
    end
end
end

```

go

```
(1 row affected)
Klijent 12:
Stari mejl je stevicjovan@gmail.com.
Novi mejl je jovanstevic12@advokatINS.com.

Completion time: 2023-05-17T00:30:26.5777923+02:00
```

Slika 32 Pozitivan rezultat druge procedure

```
Klijent sa ID-jem 111 ne postoji.

Completion time: 2023-05-17T00:29:25.2586331+02:00
```

Slika 33 Negativan rezultat druge procedure

## 13. TRIGERI

### Prvi trigger

Trigger koji se aktivira nakon upisivanja u tabelu Uplata po fakturi i proverava da li su iznos uplate i iznos fakture jednaki. Takođe nakon upisivanja, status fakture (obeležje PLACENA) se postavlja na 1 što znači da je faktura plaćena. Ukoliko su svi uslovi ispunjeni upisati torku i promeniti status fakture, ukoliko nisu ispisati grešku u konzoli.

```
if object_id('Projekat.Triger1') is not null
drop trigger Projekat.Triger1
go
create trigger Projekat.Triger1
on Projekat.Uplata_po_fakturi
after insert
as
begin
    if @@ROWCOUNT=0
        return;

    declare @ID_UPF as VARCHAR(10)=(select ID_UPF from inserted)
    declare @ID_STRANKE_K as VARCHAR(5)=(select ID_STRANKE_K from
inserted)
    declare @ID_F as NUMERIC(3)=(select ID_F from inserted)
    declare @ID_IN as NUMERIC(2)=(select ID_IN from inserted)

    declare @iznUplate as numeric(8) = (select IZN_UP from
Projekat.Uplata where ID_UP=@ID_UPF)
    declare @iznFakture as numeric(8) = (select f.UI_F from
Projekat.Faktura f join Projekat.Uplata_po_fakturi upf on
f.ID_F=upf.ID_F where upf.ID_UPF=@ID_UPF)

    if (@iznFakture!=@iznUplate)
    begin
        declare @msg2 as varchar(200)
        set @msg2=formatmessage('Iznos uplate '+
convert(varchar,@iznUplate)+ ' se razlikuje od iznosa fakture
'+convert(varchar,@iznFakture)+
' Ne moze unos.')
        raiserror(@msg2,16,0)
```



```

delete from Projekat.Uplata_po_fakturi
where ID_UPF=@ID_UPF
delete from Projekat.Uplata
where ID_UP=@ID_UPF
end
else
begin
    print 'Iznos fakture jednak je iznosu uplate. Promena
statusa fakture.'

    update Projekat.Faktura
    set PLACENA= 1
    where ID_F=@ID_F
end
end
go

```

Iznos fakture jednak je iznosu uplate. Promena statusa fakture.

(1 row affected)

(1 row affected)

Completion time: 2023-05-25T19:47:07.3924328+02:00

Slika 34 Rezultat prvog trigeru - uspešno izvršenje

Msg 50000, Level 16, State 0, Procedure Triger1, Line 23 [Batch Start Line 73]  
Iznos uplate 15000 se razlikuje od iznosa fakture 10000. Ne može unos.

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2023-05-25T19:46:51.8498762+02:00

Slika 35 Rezultat prvog trigeru - neuspešno izvršenje

## Drugi trigger

Trigger se aktivira prilikom upisivanja nove torke u tabelu Uplata po sudskom resenju i proverava da li je datum uplate manji od datuma pokretanja predmeta na osnovu čije pravosnažne sudske presude je izvršena uplata po sudskom rešenju. Ukoliko je datum validan uspešno izvršiti unos, a ukoliko nije onda ispisati grešku u konzoli.

```
if object_id('Projekat.Triger2') is not null
drop trigger Projekat.Triger2
go
create trigger Projekat.Triger2
on Projekat.Uplata_po_sudskom_resenju
instead of insert
as
begin
    if @@ROWCOUNT=0
        return;
    declare @ID_UPS as VARCHAR(10)=(select ID_UPS from inserted)
    declare @brPredmeta varchar(10) = (select BR_PREDMETA_PSP from
inserted)
    declare @rbrSpisa as numeric(2) = (select RBR_SPI_PSP from
inserted)
    declare @datumUplate as date=(select DAT_UP from Projekat.Uplata
where ID_UP=@ID_UPS)
    declare @datumPokretanja as date =(select p.DAT_PP from
Projekat.Predmet p WHERE p.BR_PREDMETA=@brPredmeta)

    if(@datumUplate<@datumPokretanja)
    begin
        declare @msg as varchar(150) = 'Uplata ne sme biti izvršena
pre pokretanja predmeta na osnovu cije pravosnazne sudske presude je
izvršena uplata! Greska!'
        raiserror(@msg,16,0)

        delete from Projekat.Uplata
        where ID_UP=@ID_UPS
    end
    else
    begin
        insert into Projekat.Uplata_po_sudskom_resenju
        values(@ID_UPS,@brPredmeta,@rbrSpisa, (select ID_IN from
inserted))
    end
end
```

```
print 'Uspesno izvršen unos u tabelu uplata po sudskom  
resenju.'  
end  
end  
go
```

```
(1 row affected)  
Uspesno izvršen unos u tabelu uplata po sudskom resenju.  
  
(1 row affected)  
  
Completion time: 2023-05-25T19:47:50.7131921+02:00
```

Slika 36 Rezultat drugog trigeru - uspešno izvršenje

```
Msg 50000, Level 16, State 0, Procedure Triger2, Line 17 [Batch Start Line 171]  
Uplata ne sme biti izvršena pre pokretanja predmeta na osnovu cije pravosnazne sudske presude je izvršena uplata! Greska!  
  
(1 row affected)  
  
(1 row affected)  
  
Completion time: 2023-05-25T19:48:08.3126169+02:00
```

Slika 37 Rezultat drugog trigeru - neuspešno izvršenje

## 14. ZAKLJUČAK

Prilikom realizacije projekta iz predmeta Sistemi baza podataka korišćena je šema baze podataka projekta sa predmeta Projektovanje baza podataka. Prilikom rada na tom projektu susretali smo se sa mnoštvom problema, ali smo na kraju uspjeli da postignemo svoj cilj misleći da smo dobili zadovoljavajući model na kojem će moći dalje da se razvija sistem.

Međutim, prilikom rada na ovom projektu, može se zaključiti da tadašnja znanja nisu bila dovoljna da se u kompletu razume sistem kako bi se efikasno izmodelovali svi delovi. Radivši na jednoj od podšema, bilo je prilike posvetiti se konkretnim delovima i zbog toga se neretko javljala potreba izmene istih.

Na kraju izrade ovog projekta jasno se može doneti zaključak o tome da je sada u potpunosti zaokružena priča o projektu koji je rađen u protekla dva semestra u pogledu baza podataka.