

Računarska gimnazija

Maturski rad

iz predmeta

Modeli i baze podataka

Baza podataka za advokatsku kancelariju

Učenik

Milica Zoranović

Mentor

Stanka Matković

Beograd, maj 2020.

Sadržaj

| | |
|---|----|
| Relacione baze podataka | 3 |
| Istorijat o relacionim bazama podataka..... | 3 |
| Osnovno o relacionim bazama podataka | 3 |
| Model baze podataka za advokatsku kancelariju | 4 |
| Opis poslovanja..... | 4 |
| Opis modela..... | 5 |
| Mapiranje | 7 |
| Uzorak podataka | 8 |
| Jezik SQL i programski jezik PL/SQL | 9 |
| Jezik SQL | 9 |
| 1. Naredbe za definisanje podataka..... | 10 |
| 2. Naredbe za rukovanje podacima | 11 |
| 3. Naredbe za kontrolne funkcije..... | 15 |
| Pretraga baze podataka pomoću jezika PL/SQL | 15 |
| Jezik PL/SQL..... | 15 |
| Naredba SELECT INTO | 15 |
| Kursori | 17 |
| Kursor sa parametrom | 19 |
| Procedure | 22 |
| Funkcije | 23 |
| Zaključak..... | 24 |
| Literatura..... | 24 |

Relacione baze podataka

Istorijat o relacionim bazama podataka

Početak sedamdesetih godina prošlog veka počinje da se pojavljuje pojam relacionog modela, a njegov osnivač je Edgar Kod. On je 1970. godine objavio veliki broj kratkih dela o relacionom modelu, to je bio uvod u njegovo veliko delo „Relacioni model za organizaciju velike količine podataka“. U delu je prikazan model, po kojem bi se podaci čuvali u tabelama umesto u povezanim listama, koje su predstavljale osnove navigacionog modela. Navigacioni model je bio vrlo neefikasan. Kod je u svom delu naveo i velike prednosti relacionog modela u odnosu na navigacioni model, koje se odnose na smeštanje, čitanje podataka, brisanje iz baze. Kod je predložio jezik koji se kasnije razvio u SQL.

Osnovno o relacionim bazama podataka

Baza podataka predstavlja skup različitih tabela koje su međusobno povezane određenim relacijama. Dijagram relacione baze podataka sačinjen je od skupa entiteta povezanih različitim vezama. Svaki entitet sadži skup atributa koji ga bliže opisuje. U jednom entitetu možemo da imamo više atributa koji mogu da budu označeni kao obavezan atribut i kao opcioni atribut. U zavisnosti od toga šta nam je potrebno u tabeli tako i označavamo attribute. Atribut koji je označen kao obavezan, ispred naziva atributa ima *, pri kreiranju mora da se napise da je not null, i obavezno je popuniti to polje prilikom popunjavanja tabele. Kada je atribut označen kao opcioni podatak, on ispred naziva atributa ima o, tako da pri kreiranju tabele nije obavezno popuniti taj podatak. Kada u nekom redu tabele nećemo da popunimo podatak unosemo samo reč null, i tako napisano će u tabeli biti prikazano kao -. Jedan od atributa se izabere kao jedinstveni identifikator, primarni ključ i on se označava #.

| Advokati | | | | |
|-------------|---------|------------|-----------|--|
| Id_advokata | ime | prezime | telefon | mejl |
| 1 | Slavica | Mastilovic | 069548638 | slavica@gmail.com |
| 2 | Miljana | Petrovic | 065214896 | - |
| 3 | Milica | Cvetkovic | 069447512 | milica@gmail.com |
| 4 | Anja | Djuric | 061758462 | djuric@gmail.com |
| 5 | Andrea | Corak | 065455584 | - |

Naziv tabele mora biti u množini

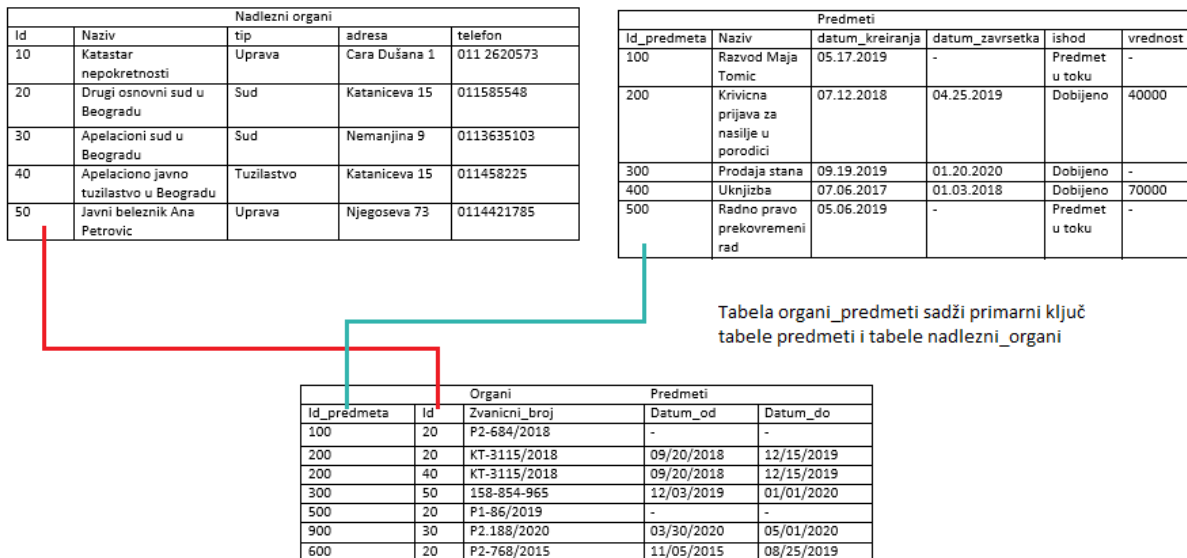
U jednom redu su podaci o jednom advokatu

Kolona koja je primarni ključ

Kolona gde svi podaci moraju biti popunjeni

Kolona gde je opciono popunjavanje podataka

Kada u dijagramu imamo vezu više na prema više, tada dodajemo novi entitet koji putem barovanih veza spajamo sa drugim entitetima. Prilikom spajanja novi entitet povlači primarne ključeve drugih entiteta, u njegovoj tabeli su prikazani kao strani i primarni ključ. Nova tabela može ali i ne mora da sadrži dodatne podatke, potrebno je samo da ima primarne ključeve susednih entiteta. U slučaju da nam entitet sadrži i druge atribute, za njih možemo odrediti da li će biti opcioni ili obavezni atributi. Pri kreiranju tabele možemo napisati da li je podatak opcioni ili obavezni, i na osnovu toga možemo popuniti tabelu.



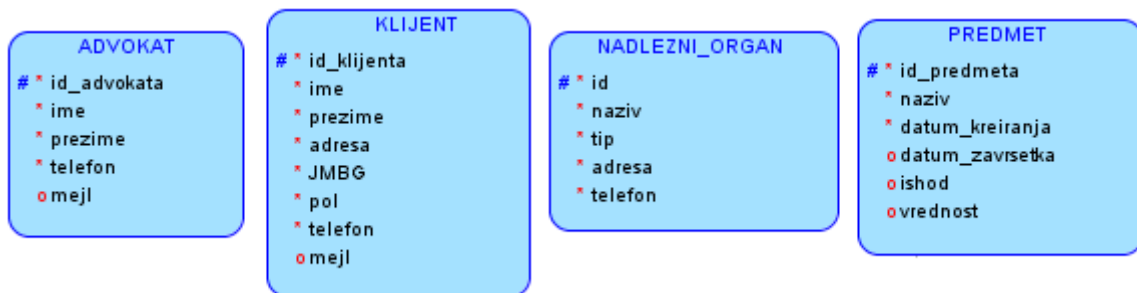
Model baze podataka za advokatsku kancelariju

Opis poslovanja

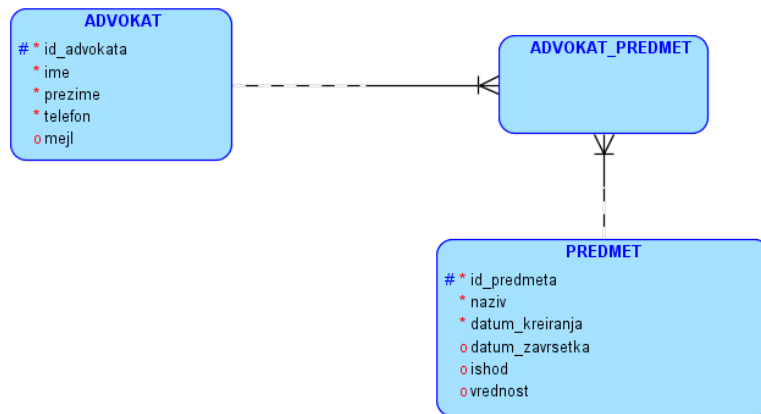
Advokatska kancelarija pruža stručnu pravnu pomoć u različitim oblastima i situacijama primene i ostvarivanja pravnog sistema, pravo zastupanja građana i pravih lica. U svakodnevnom radu sa klijentima na njihovim predmetima advokati u savremeno doba počinju da koriste sve više programe koji im omogućavaju lakšu organizaciju poslovanja, čuvanja i pretragu podataka i podelu posla među advokatima. Kada klijent odluči da agažuje advokatsku kancelariju prvo se određuje koji advokat/i će biti zadužen/i za taj predmet i nakon pravne analize doneće se odluka koju vrstu postupka i pred kojom ustanovom postupak treba voditi. Postupak se pokreće podnošenjem odgovarajućeg podneska i to se evidentira kao datum početka. Po prijemu podneska nadležni organ dodeljuje jedinstveni službeni broj predmeta. Ovaj broj je važan jer pomoću njega se pristupa online zvaničnoj bazi podataka za svaki predmet koji se vodi pred sudom što olakšava praćenje predmeta (<https://tpson.portal.sud.rs/tposvs/>). Po završenom postupku unosi se datum završetka postupka. Po naplati advokatske nagrade završen predmet će biti spreman za arhivu.

Opis modela

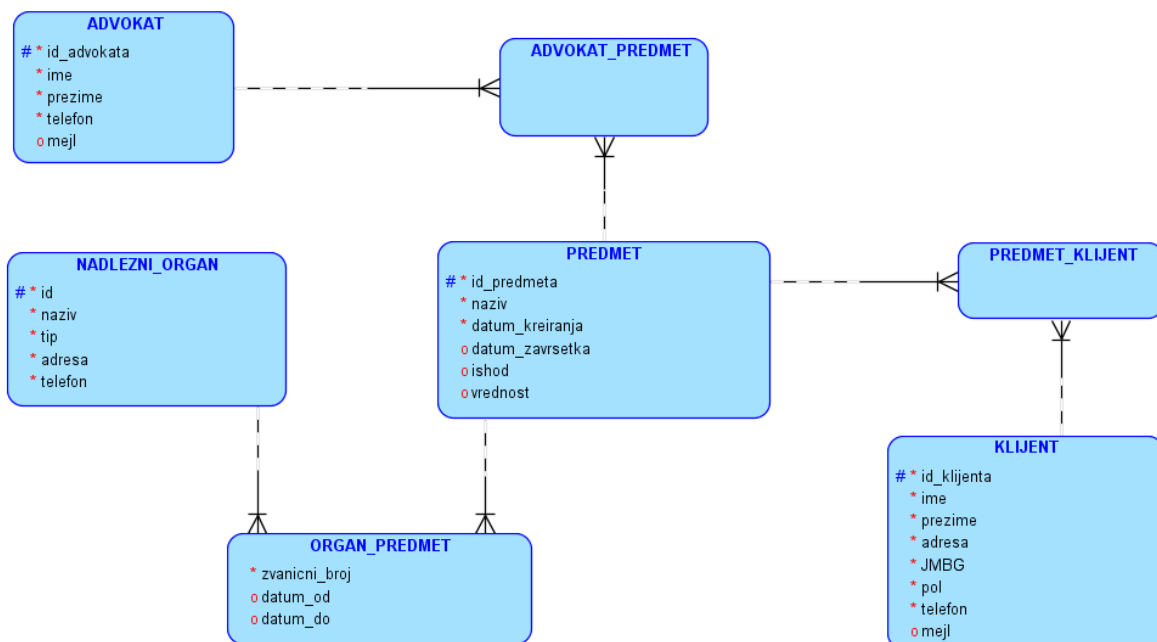
Pravimo bazu podataka za advokatsku kancelariju. U kancelariji radi više advokata. Prvo kreiramo entitete koji su nezavisni od drugih entiteta u dijagramu. Imamo entitete ADVOKAT, KLIJENT, NADLEZNI_ORGAN i PREDMET koji su samostalni entiteti u dijagramu. Entitet ADVOKAT sadrži atribute kao što su id advokata koji je obavezan atribut, ali on je i primarni ključ tog entiteta, tako da ispred imena ima # i time je označen kao primarni ključ. Takođe entitet sadrži i druge potrebne atribute kao što su ime, prezime i telefon koju su obavezni atributi u entitetu i oni sadrže * ispred naziva. Imamo i atribut mejl koji je označen kao opcioni i ispred naziva sadrži o. Entitet KLIJENT nam daje detaljnije informacije o klijentu, a to su id klijenta koji je primarni ključ ovog entiteta, sadrži i druge atribute kao što su ime, prezime, adresa, JMBG, pol klijenta i telefon koji su obavezni atributi. Jedini opcioni atribut u ovom entitetu je mejl. Entitet NADLEZNI_ORGAN sadrži potrebne podatke o tome gde se vode predmeti. Sadrži id koji je primarni ključ i obeležen je sa #. U ovom entitetu svi ostali podaci, naziv ustanove u kojoj se vodi postupak, tip ustanove, adresu i telefon ustanove, su označeni kao obavezni podaci, tako da se ispred svakog atributa nalazi *. Entitet PREDMET sadrži atribute potrebne za detaljniji opis predmeta, a to su id predmeta koji je primarni ključ, naziv, datum kreiranja predmeta i to su atributi koji su označeni kao obavezni u entitetu. Dok su datum završetka, ishod i vrednost vođenog predmeta opcioni atributi.



Advokat može da radi na jednom ili više predmeta, ali i na jednom predmetu može da radi više advokata. Taj problem veze više na prema više smo rešili dodavanjem novog entiteta ADVOKAT_PREDMET koji sadrži barovane veze i sadrži primarne ključeve entiteta ADVOKAT i PREDMET. U kancelariju dolaze klijenti, jedan klijent može da angažuje advokatsku kancelariju za više predmeta. Jedan predmet može da ima više klijenta koji stoje iza tog predmeta. Tako da smo taj problem veze više na prema više rešili dodavanjem novog entiteta pod nazivom PREDMET_KLIJENT, koja sadrži primarne ključeve entiteta PREDMET i KLIJENT. Većina predmeta zahteva nekoliko koraka koji se vode pred različitim nadležnim organima. Na primer jedan predmet mora da se vodi i kod javnog beležnika i u sudu. Tu situaciju smo isto rešili barovanom vezom i dodavanjem novog entiteta ORGAN_PREDMET koji sadrži neke od potrebnih podataka. Entitet OGRAN_PREDMET sadrži kao obavezan atribut zvaničan broj predmeta, ali sadrži i dodatne atribute koji su opcioni, a to su datumi od i do kad traje neki određen postupak. Na taj način je rešen problem rada jedne advokatske kancelarije koja je sačinjena od nekoliko advokata.

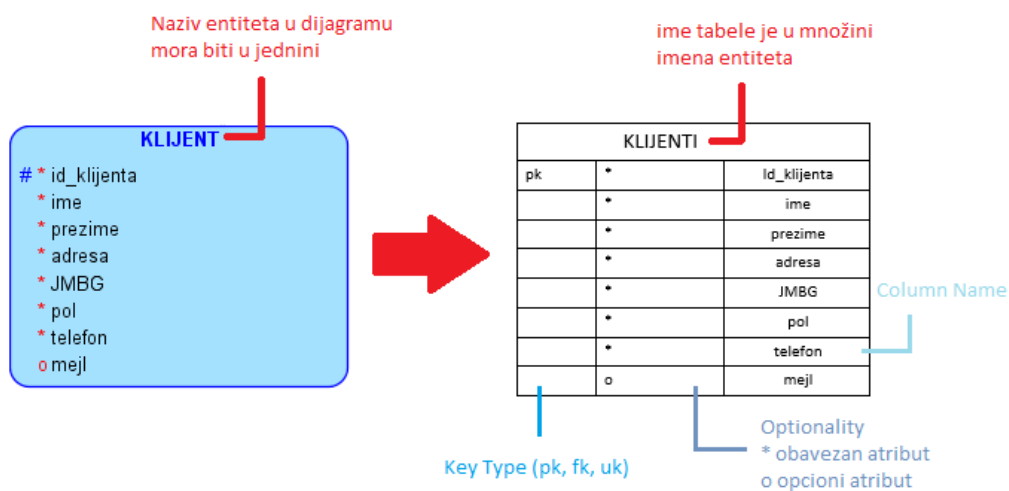


Prikaz celog dijagrama

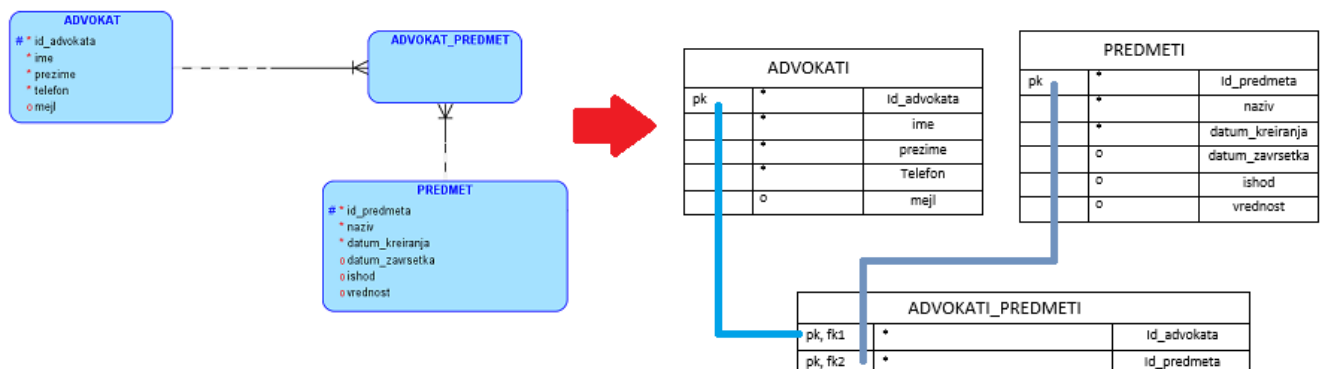


Mapiranje

Posle kreiranja dijagrama potrebno je uraditi mapiranje tabela. Za svaki entitet potrebno je napraviti tabelu koja će sadržati naziv tabele koji je u množini i tri kolone. Prva kolona pod nazivom Key Type biće popunjena samo na mestima gde je prikazan strani i primarni ključ. Ukoliko je polje primarni ključ, koji ima svaka tabela, to polje je popunjeno sa oznakom pk, a u slučaju da je polje strani ključ onda će to polje biti popunjeno sa oznakom fk. Druga kolona pod nazivom Optionality biće popunjena na svim poljima po uzoru na dijagram. Ona može da bude popunjena sa dve oznake, to je * ukoliko je atribut obavezan i o ukoliko je atribut opcioni. Treća kolona pod nazivom Column Name sadržace nazive svih atributa iz entiteta.



Entitet koji razrešava vezu više na prema više, prilikom mapiranja tabela će sadržati primarne ključeve susjednih entiteta, i u njoj će biti prikazani kao primarni i strani ključ. Ako su entiteti povezani samo putem veze jedan na prema više, onda će taj primarni ključ u njegovoj tabeli biti prikazan samo kao strani ključ i imaće oznaku fk. Ukoliko je povezan putem barovanih veza onda će taj id u tabeli biti i primaran i strani ključ i imaće oznaku pk i fk.



Uzorak podataka

| Advokati | | | | |
|-------------|---------|------------|-----------|--|
| Id_advokata | ime | prezime | telefon | mejl |
| 1 | Slavica | Mastilovic | 069548638 | slavica@gmail.com |
| 2 | Miljana | Petrovic | 065214896 | - |
| 3 | Milica | Cvetkovic | 069447512 | milica@gmail.com |
| 4 | Anja | Djuric | 061758462 | diuric@gmail.com |
| 5 | Andrea | Corak | 065455584 | - |

| Klijenti | | | | | | | |
|----------|------|-----------|----------------|---------------|--------|------------|--|
| id | ime | pezime | Adresa | JMBG | pol | telefon | mejl |
| 11 | Ana | Tomasevic | Takovska 11 | 1125487632584 | zensko | 0695584732 | - |
| 22 | Bane | Zoranovic | Djusina 9 | 0215477485214 | musko | 0614475212 | bane@gmail.com |
| 33 | Pera | Peric | Nusiceva 5 | 1524893632111 | musko | 0655254188 | pera@gmail.com |
| 44 | Hana | Bakarevic | Hilandarska 14 | 0104599999412 | zensko | 0659875562 | hana@gmail.com |
| 55 | Maja | Tomic | Svetogorska 20 | 0215488852132 | zensko | 0665847565 | - |

| Nadlezni organi | | | | |
|-----------------|--|------------|---------------|-------------|
| id | naziv | tip | adresa | telefon |
| 10 | Katastar nepokretnosti | Uprava | Cara Dušana 1 | 011 2620573 |
| 20 | Drugi osnovni sud u Beogradu | Sud | Kataniceva 15 | 011585548 |
| 30 | Apelacioni sud u Beogradu | Sud | Nemanjina 9 | 0113635103 |
| 40 | Apelaciono javno tuzilastvo u Beogradu | Tuzilastvo | Kataniceva 15 | 011458225 |
| 50 | Javni beleznik Ana Petrovic | Uprava | Njegoseva 73 | 0114421785 |

| Predmeti | | | | | |
|----------|--|-----------------|-----------------|----------------|----------|
| Id | Naziv | datum_kreiranja | datum_zavrsetka | ishod | vrednost |
| 100 | Razvod Maja Tomic | 05.17.2019 | - | Predmet u toku | - |
| 200 | Krivicna prijava za nasilje u porodici | 07.12.2018 | 04.25.2019 | Dobijeno | 40000 |
| 300 | Prodaja stana | 09.19.2019 | 01.20.2020 | Dobijeno | - |
| 400 | Uknjizba | 07.06.2017 | 01.03.2018 | Dobijeno | 70000 |
| 500 | Radno pravo prekovremeni rad | 05.06.2019 | - | Predmet u toku | - |
| 600 | Razvod Pera Peric | 10.17.2015 | 09.12.2019 | Izgubljeno | 100000 |
| 700 | Uknjizba | 12.25.2019 | - | Predmet u toku | - |
| 800 | Krivicna prijava za nasilje u porodici | 10.04.2019 | - | Predmet u toku | - |
| 900 | Utvrđivanje ocinstva | 03.20.2020 | 05.10.2020 | Dobijeno | 30000 |

| Advokati predmeti | |
|-------------------|-------------|
| Id_advokata | Id_predmeta |
| 4 | 700 |
| 1 | 200 |
| 3 | 200 |
| 2 | 500 |
| 2 | 800 |
| 3 | 800 |
| 5 | 800 |
| 1 | 100 |
| 5 | 600 |
| 4 | 300 |
| 2 | 400 |
| 4 | 900 |

| Predmeti klijenti | |
|-------------------|-------------|
| Id_predmeta | Id_klijenta |
| 100 | 55 |
| 800 | 11 |
| 300 | 44 |
| 300 | 22 |
| 200 | 11 |
| 400 | 33 |
| 500 | 44 |
| 600 | 33 |
| 700 | 22 |
| 900 | 33 |

| Organi | | | Predmeti | |
|-------------|----|--------------------------|------------|------------|
| Id_predmeta | Id | Zvanicni_broj | Datum_od | Datum_do |
| 100 | 20 | P2-684/2018 | - | - |
| 200 | 20 | KT-3115/2018 | 09/20/2018 | 12/15/2019 |
| 200 | 40 | KT-3115/2018 | 09/20/2018 | 12/15/2019 |
| 300 | 50 | 158-854-965 | 12/03/2019 | 01/01/2020 |
| 500 | 20 | P1-86/2019 | - | - |
| 900 | 30 | P2.188/2020 | 03/30/2020 | 05/01/2020 |
| 600 | 20 | P2-768/2015 | 11/05/2015 | 08/25/2019 |
| 700 | 10 | 952-12-5-339-307 25/2019 | - | - |
| 700 | 50 | 952-12-5-339-307 25/2019 | - | - |
| 800 | 20 | KT-4215/2019 | - | - |
| 800 | 40 | KT-4215/2019 | - | - |
| 400 | 10 | 951-07-4-449-308 06/2017 | 10/10/2017 | 12/15/2017 |
| 400 | 50 | 951-07-4-449-308 06/2017 | 10/10/2017 | 12/15/2017 |

Jezik SQL i programski jezik PL/SQL

Jezik SQL

SQL je nastao u Kaliforniji 1974. godine u istraživačkoj laboratoriji kompanije IMB u San Hoseu. Njegov osnivač je Donald Chamberlin. Nastao je na istom mestu gde je i Kod 1970. godine definisao osnove relacionog modela podataka.

SQL naredbe svrstane su u tri kategorije:

1. Naredbe za definisanje podataka
2. Naredbe za rukovanje podacima
3. Naredbe za kontrolne funkcije

1. Naredbe za definisanje podataka

Naredba CREATE TABLE

Kada kreiramo tabelu potrebno je uneti ključnu reč CREATE TABLE i naziv tabele po našem izboru. U ostatku naredbe za kreiranje tabele unosimo nazive kolona koje odgovaraju atributima iz dijagrama i pored svakog naziva upisujemo kojeg je tipa podatak koji smo uneli. Za podatke koji su obavezni atributi pored tipa podataka ćemo napisati not null, i taj podatak je obavezno popuniti. Ukoliko je podatak opcioni, pored tipa podataka nećemo ništa napisati.

```
CREATE TABLE klijenti
(id_klijenta number(10) primary key,
ime varchar2(30) not null,
prezime varchar2(40) not null,
adresa varchar2(100) not null,
JMBG varchar2 (13) not null,
pol varchar2(10) not null,
telefon varchar2 (20) not null,
mejl varchar2(100))
```

```
CREATE TABLE predmeti
(id_predmeta number(10) primary key,
naziv varchar2(50) not null,
datum_kreiranja date not null,
datum_zavrsetka date,
ishod varchar2(20) not null,
vrednost number(20))
```

Kada kreiramo tabelu za entitet koji je u dijagramu povezan putem barovanih veza sa drugim entitetima, tad su nam id tih tabela i strani i primarni ključ. Upisujemo naziv kolone u kojoj ćemo čuvati primarni ključ tabele sa kojom je naša tabela povezana, njem tip i ključnu reč REFERENCES potom naziv tabele iz koje uzimamo primarni ključ i u zagradama unosimo naziv primarnog ključa te tabele. Ovako unet primarni ključ druge tabele je u našoj tabeli prikazan kao strani ključ. On u nasoj tabeli postaje primarni ključ unošenjem ključne reči PRIMARY KEY i u zagradi nazive primarnih ključeva druge tabele.

```
CREATE TABLE predmeti_klijenti
(id_predmeta number(10) references predmeti(id_predmeta),
id_klijenta number(10) references klijenti(id_klijenta),
primary key (id_predmeta, id_klijenta))
```

Naredba ALTER TABLE

Pri kreiranju tabela često dodje do grske. Neke od grešaka su da pogrešan tip podataka stavimo pri kreiranju. Za promenu tipa podataka možemo da koristimo ALTER TABLE. Pomoću ove naredbe možemo i obrisati neku kolonu iz tabele, kao i da dodamo neku kolonu u tabelu.

```
ALTER TABLE table_name  
DROP COLUMN column_name;
```

```
alter table advokati  
modify (telefon varchar2(20))
```

```
ALTER TABLE table_name  
ADD column_name datatype;
```

```
alter table klijenti  
modify (JMBG varchar2(13))
```

Naredba DROP TABLE

Za brisanje tabele iz baze koju smo kreirali, koristimo DROPE TABLE. Kada pišemo naredbu za brisanje tabele koristimo ključnu reč DROPE TABLE u nastavku naredbe napišemo naziv tabele koju želimo da izbrišemo, pri pokretanju naredbe tabela će biti izbrisana.

```
DROP TABLE table_name;
```

2. Naredbe za rukovanje podacima

Naredba SELECT

Da bi prikazali sadržaj relacione baze podataka, koristimo SELECT. Kad prikazujemo sadržaj cele tabele onda ćemo napisati SELECT * FROM potom naziv tabele čiji sadržaj želimo da prikažemo.

```
SELECT * FROM predmeti
```

Ako prikazujemo sadržaj samo nekih kolona tabele onda ćemo posele SELECT napisati nazive kolona koje želimo da prikažemo, potom FROM i naziv tabele čiji sadržaj prikazujemo.

```
SELECT naziv, datum_kreiranja, ishod, vrednost
FROM predmeti
```

| NAZIV | DATUM_KREIRANJA | ISHOD | VREDNOST |
|--|-----------------|----------------|----------|
| Razvod Pera Peric | 10/17/2015 | Izgubljeno | 100000 |
| Krivicna prijava za nasilje u porodici | 10/04/2019 | Predmet u toku | - |
| Radno pravo prekovremeni rad | 05/06/2019 | Predmet u toku | - |
| Utvrđivanje ocinstva | 03/20/2020 | Dobijeno | 30000 |

Kada želimo da prikazemo samo neke redove tabele, koristimo WHERE deo. U SELECT delu napisemo nazive svih kolona koje želimo da prikazemo, dok u FROM delu samo napisemo iz koje tabele uzimamo podatke. U WHERE delu navedemo uslov po kojem će da se izdvoje samo neki redovi. U ovom primeru imamo prikazano da samo hteli da izdvojimo samo predmete koji su u toku.

```
SELECT naziv, datum_kreiranja, ishod
FROM predmeti
WHERE UPPER(ishod) = 'PREDMET U TOKU';
```

| NAZIV | DATUM_KREIRANJA | ISHOD |
|--|-----------------|----------------|
| Krivicna prijava za nasilje u porodici | 10/04/2019 | Predmet u toku |
| Radno pravo prekovremeni rad | 05/06/2019 | Predmet u toku |
| Uknjizba | 12/25/2019 | Predmet u toku |
| Razvod Maja Tomic | 05/17/2019 | predmet u toku |

Kada prikazujemo sadržaj iz više tabela, potrebno je napisati SELECT i nazive svih kolona koje želimo da prikazemo. U FROM delu koda potrebno je spojiti tabele da bi nam sadržaj bio prikazan. Tabele spajamo pomoću JOIN funkcije. Prvo napišemo naziv jedne tabele zatim ključnu reč JOIN i naziv druge tabele potom pišemo USING, gde upisujemo naziv primarnog ključa jedne tabele. Piše se id koji je u jednoj tabeli primarni, a u drugoj tabeli strani ključ.

```
SELECT naziv, tip, zvanicni_broj
FROM nadlezni_organ_i JOIN organi_predmeti USING(id)
```

| NAZIV | TIP | ZVANICNI_BROJ |
|--|------------|--------------------------|
| Apelaciono javno tuzilastvo u Beogradu | Tuzilastvo | KT-3115/2018 |
| Katastar nepokretnosti | Uprava | 952-12-5-339-307 25/2019 |
| Apelaciono javno tuzilastvo u Beogradu | Tuzilastvo | KT-4215/2019 |
| Drugi osnovni sud u Beogradu | Sud | P2-684/2018 |

Kada želimo da prikazemo samo neke redove iz tabele koristimo WHERE. U slučaju da u dve različite tabele imamo kolone sa istim imenom, pri pisanju SELECT upita te kolone sa istim imenom moramo označiti iz koje su tabele, u suprotnom nam upit neće raditi. Ispred naziva kolone napišemo naziv tabele u kojoj se nalazi taj kolone.

```
SELECT nadlezni_organ_i.naziv, tip, zvanicni_broj, predmeti.naziv
FROM nadlezni_organ_i JOIN organi_predmeti USING(id) JOIN predmeti USING(id_predmeta)
WHERE UPPER(tip)='TUZILASTVO';
```

| NAZIV | TIP | ZVANICNI_BROJ | NAZIV |
|--|------------|---------------|--|
| Apelaciono javno tuzilastvo u Beogradu | Tuzilastvo | KT-3115/2018 | Krivicna prijava za nasilje u porodici |
| Apelaciono javno tuzilastvo u Beogradu | Tuzilastvo | KT-4215/2019 | Krivicna prijava za nasilje u porodici |

Naredba UPDATE

Za izmenu vrednosti kolona tabele koristimo UPDATE. Možemo da izmeni sve podatke u tabeli. Kada želimo neki podatak da izmenimo prvo upisujemo UPDATE pa naziv tabele gde želimo da izmenimo podatak, u SET delu napišemo naziv kolone koji želimo da promenimo i stavljamo novi podatak koji želimo da piše, u WHERE delu upisujemo kako je taj podatak napisan u našoj tabeli.

```
UPDATE advokati
SET mejl='m.slavica@gmail.com'
WHERE mejl='slavica@gmail.com'
```

Naredba DELETE

Za brisanje podataka iz baze koristimo DELETE. Pri brisanju nekog reda iz baze napisaćemo ključnu reč DELETE i potom naziv tabele iz koje želimo da izbrišemo neki red, u WHERE delu napisaćemo id reda koji želimo da obrišemo. Pri pokretanju ove naredbe, red će biti obrisan iz tabele.

```
DELETE predmeti
WHERE id_predmeta=600
```

Naredba INSERT

Tabelu popunjavamo pomoću INSERT naredbe. Svaki novi red u tabeli dodajemo pomoću iste komande. Kada popunjavamo tabelu koja ne sadrži strane ključeve drugih tabela, onda se polja popunjavaju samo na osnovu atributa u entitetu. Prvo napišemo INSERT INTO i naziv tabele gde želimo da dodamo novi red, potom u VALUES unosimo podatke istim redom kao kad smo kreirali tabelu. Ako nam je polje tipa VARCHAR2 onda podatak pišemo između apostrofa, kada nam je podatak opcioni i to polje ukoliko ne želimo da popunimo onda upisujemo reč null. Kada nam je polje tipa NUMBER onda samo upisujemo određen broj koji nam treba. Kada nam je polje tipa DATE, datum se upisuje između apostrofa.

```
INSERT INTO klijenti
VALUES (11, 'Ana', 'Tomasevic', 'Takovska 11', '1125487632584', 'zensko', '0695584732',
null)
```

Kada popunjavamo novi red u tabeli koja sadrži strane ključeve drugih tabela, onda unosimo vrednost primarnog ključa jedne tabele i vrednost primarnog ključa druge tabele koje želimo da povežemo, ukoliko tabelela sadrži i druge atribute njih unosimo u zavisnosti od tipa podataka.

```
INSERT INTO organi_predmeti — naziv tabele
VALUES (200, 40, 'KT-3115/2018', '09/20/2018', '12/15/2019')
```

id_predmeta id_organa

3. Naredbe za kontrolne funkcije

Postoje još neke naredbe koje koristimo za kontrolne funkcije.

- GRANT
- REVOKE
- COMMIT
- ROLLBACK

Pretraga baze podataka pomoću jezika PL/SQL

Jezik PL/SQL

PL/SQL ima mnogo više opcija za obradu baze podataka od samog jezika SQL. On predstavlja proceduralni jezik koji je proširenje za SQL za relacione baze podataka kompanije Oracle. Naziv je nastao kao skraćenica za: „Procedural Language for SQL“. Pomoću PL/SQL mogu se kreirati neimenovani blokovi, funkcije, procedure i okidači. Omogućuje deklaraciju i korišćenje konstanti i promenljivih. Kada pišemo programe pomoću ovog jezika koristimo naredbe jezika SQL za direktan rad sa bazom podataka, ali imamo na raspolaganju i druge elemente proceduralnih jezika kao što su:

- grananja: IF i CASE
- ciklusi: LOOP, FOR i WHILE

Naredba SELECT INTO

1. Prikazati broj predmetima koji su u toku.

```
DECLARE
  v_broj number;
BEGIN
  SELECT count(*) INTO v_broj
  FROM predmeti WHERE UPPER(ishod)=UPPER('predmet u toku');
  DBMS_OUTPUT.PUT_LINE('Broj predmeta koji su u toku: '||v_broj);
END;
```

```
Broj predmeta koji su u toku: 4
```

2. Za klijenta datog imena i prezimena, prikazati broj predmeta koje vode u advokatskoj kancelariji.

U ovom primeru koristimo naredbu SELECT INTO. Ime i prezime unosimo sa tastature, i za tog klijenta će nam biti prikazan broj predmeta koje vodi u advokatskoj kancelariji. Deklarišemo ime, prezime i broj. Ime i prezime se unose sa tastature. Pomoću funkcije COUNT brojimo predmete jednog određenog klijenta. Na kraju prikazujemo ime i prezime klijenta, kao i broj predmeta.

```
DECLARE
  v_ime VARCHAR(30);
  v_prezime VARCHAR2(30);
  v_broj NUMBER;
BEGIN
  v_ime:=:Ime;
  v_prezime:=:Prezime;
  SELECT COUNT(*) INTO v_broj
  FROM klijenti JOIN predmeti_klijenti USING(id_klijenta)
  JOIN predmeti using(id_predmeta)
  WHERE UPPER(ime)||UPPER(prezime)
  LIKE '%'||UPPER(v_ime)||UPPER(v_prezime)||'%';
  DBMS_OUTPUT.PUT_LINE('Klijent: '||' '||v_ime||' '||v_prezime);
  DBMS_OUTPUT.PUT_LINE('Broj predmeta koje vodi u advokatskoj
  kancelariji: '||' '||v_broj);
END;
```

Kada se programski kod pokrene, kao rezultat nam prikazuje informacije za klijenta čije smo ime i prezime uneli sa tastature. Prikazuje nam ime i prezime klijenta kao i broj predmeta koje klijent vodi u advokatskog kancelariji.

```
Klijent: Pera Peric
Broj predmeta koje vodi u advokatskoj kancelariji: 3
```


Kursori

3. Prikazati sve informacije o nadležnim ustanovama gde se vode predmeti.

U ovom primeru koristimo CURSOR. Prvo deklariramo sve podatke koje želimo da prikazemo, deklarirali smo naziv, tip, adresu i telefon. Pri deklarisanju možemo uneti kog je tipa podatak, u našem slučaju naziv je tipa VARCHAR2 i unosimo broj karaktera koji nam je potreban za prikaz. Umesto tipa podatka, možemo napisati tačan naziv tabele i naziv kolone koji želimo da prikazemo, na kraju stavljamo % TYPE. Koristimo jedan kursor, gde izdvajamo sve podatke iz jedne tabele, NADLEZNI_ORGANI. Pomoću DBMS_OUTPUT.PUT_LINE prikazujemo svaki podatak pojedinačno i u novom redu.

```
DECLARE
  v_naziv varchar2(50);
  v_tip varchar2(20);
  v_adresa nadlezni_organ.adresa%type;
  v_telefon nadlezni_organ.telefon%type;
  CURSOR kursor_nadlezni IS
    SELECT naziv, tip, adresa, telefon
    FROM nadlezni_organ;
BEGIN
  OPEN kursor_nadlezni;
  LOOP
    FETCH kursor_nadlezni INTO v_naziv, v_tip, v_adresa, v_telefon;
    EXIT WHEN kursor_nadlezni%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('Naziv: ' || v_naziv);
    DBMS_OUTPUT.PUT_LINE('Tip ustanove: ' || v_tip);
    DBMS_OUTPUT.PUT_LINE('Adresa: ' || v_adresa);
    DBMS_OUTPUT.PUT_LINE('Broj telefona: ' || v_telefon);
    DBMS_OUTPUT.PUT_LINE('-----');
  END LOOP;
  CLOSE kursor_nadlezni;
END;
```

Kada se programski kod pokrene, kao rezultat nam prikazuje informacije o nadležnim organima gde se vode predmeti. Prikazuje naziv, tip ustanove, adresu i broj telefona. Svaki podatak je prikazan u novom redu.

```
Naziv:  Katastar nepokretnosti
Tip ustanove:  Uprava
Adresa:  Cara Dušana 1
Broj telefona:  011 2620573
-----
```

```
Naziv:  Drugi osnovni sud u Beogradu
Tip ustanove:  Sud
Adresa:  Kataniceva 15
Broj telefona:  011585548
-----
```

4. Prikazati informacije o svim advokatima ove advokatske kancelarije. Ime i prezime prikazati spojeno.

```
DECLARE
  v_advokat varchar2(50);
  v_telefon advokati.telefon%TYPE;
  v_mejl advokati.mejl%TYPE;
  CURSOR kursor_advokat IS
    SELECT ime || ' ' || prezime, telefon, mejl
    FROM advokati;
BEGIN
  OPEN kursor_advokat;
  LOOP
    FETCH kursor_advokat INTO v_advokat,v_telefon, v_mejl;
    EXIT WHEN kursor_advokat%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('Ime i prezime advokata: ' || v_advokat);
    DBMS_OUTPUT.PUT_LINE('Broj telefona: ' || v_telefon);
    DBMS_OUTPUT.PUT_LINE('Mejl adresa: ' || v_mejl);
    DBMS_OUTPUT.PUT_LINE('-----');
  END LOOP;
  CLOSE kursor_advokat;
END;
```

5. Za svakog advokata prikazati ime, prezime, telefon i broj predmeta na kojima radi.

```
DECLARE
CURSOR kursor_adv IS
SELECT id_advokata, ime, prezime,telefon, COUNT(*) AS broj_predmeta
FROM advokati JOIN advokati_predmeti USING(id_advokata)
JOIN predmeti USING(id_predmeta)
GROUP BY id_advokata, ime, prezime, telefon;
BEGIN
  FOR slog IN kursor_adv LOOP
    DBMS_OUTPUT.PUT_LINE('Id advokata: '||slog.id_advokata||' '||
      'Ime: '||slog.ime||' '||'Prezime: '||slog.prezime);
    DBMS_OUTPUT.PUT_LINE('  Broj telefona: '||slog.telefon||' '||
      'Broj predmeta: '||slog.broj_predmeta);
    DBMS_OUTPUT.PUT_LINE('-----');
  END LOOP;
END;
```

Kursor sa parametrom

6. Unosimo id predmeta sa tasature, i za taj predmet nam ispiše sve nadležne organe gde se taj predmet vodi.

U ovom primeru koristimo jedan kursor sa parametrom, kao parametar ima id predmeta gde izdvaja određene podatke za id predmeta koji smo uneli sa tasature. Za određen predmet nam izdvaja naziv kao i tip ustanove gde se predmet vodi i zvanični broj predmeta.

```
DECLARE
  v_id_p predmeti.id_predmeta%TYPE;
  CURSOR kursor_pred (p_id_predmeta predmeti.id_predmeta%TYPE)
  IS SELECT naziv, tip, zvanicni_broj
  FROM nadlezni_organ_i NATURAL JOIN organ_i_predmeti
  WHERE id_predmeta=p_id_predmeta;
BEGIN
  v_id_p := :id_predmeta;
  FOR v_red_p IN kursor_pred(v_id_p) LOOP
    DBMS_OUTPUT.PUT_LINE('Naziv: ' || v_red_p.naziv);
    DBMS_OUTPUT.PUT_LINE('Tip ustanove: : ' || v_red_p.tip);
    DBMS_OUTPUT.PUT_LINE('Zvanicni broj: ' || v_red_p.zvanicni_broj);
    DBMS_OUTPUT.PUT_LINE('-----');
  END LOOP;
END;
```

| Bind Variable | Value |
|---------------|----------------------------------|
| :ID_PREDMETA | <input type="text" value="200"/> |

Submit

```
Naziv: Apelaciono javno tuzilastvo u Beogradu
Tip ustanove: : Tuzilastvo
Zvanicni broj: KT-3115/2018
-----
Naziv: Drugi osnovni sud u Beogradu
Tip ustanove: : Sud
Zvanicni broj: KT-3115/2018
-----
```

7. Za svaki predmet prikazati sve informacije, koji advokati rade na predmetu kao i informacije o klijentima koji su uključeni u predmet.

U ovom primeru koristimo tri kursora: KURSOR_PREDMETI, KURSOR_ADVOKATI, KURSOR_KLIJENTI.

```
CURSOR kursor_predmeti IS
  SELECT id_predmeta, naziv, datum_kreiranja, datum_zavrsetka
  FROM predmeti;
CURSOR kursor_advokati(p_id_predmeta number) IS
  SELECT ime||' '||prezime
  FROM advokati natural join advokati_predmeti WHERE id_predmeta=p_id_predmeta;
CURSOR kursor_klijenti(p_id_predmeta number) IS
  SELECT ime||' '||prezime
  FROM predmeti_klijenti natural join klijenti WHERE id_predmeta=p_id_predmeta;
```

Prvi kursor je KURSOR_PREDMETI pomoću kog izdvajamo podatke o predmetu, a to su id predmeta, naziv, datum kreiranja i datum završetka predmeta. Svi podaci se nalaze u jednoj tabeli, tako da sve podatke dobijamo iz tabele PREDMETI. Drugi kursor pod nazivom KURSOR_ADVOKATI je kursor sa parametrom, pomoću njega izdvajamo advokate koji rade na predmetu određenog id predmeta. Za svaki predmet ćemo imati prikazano koji advokati rade na njemu. Kursor nam izdvaja ime i prezime advokata, prilikom pokretanja programa ime i prezime biće prikazano spojeno. Treći kursor pod nazivom KURSOR_PREDMETI je takođe kursor sa parametrom, gde za svaki predmet nam ispisuje ko su klijenti koji stoje iza tog predmeta. Prikazuje nam ime i prezime klijenta, prilikom pokretanja programa ime i prezime biće prikazano spojeno. Pomoću ova tri kursora, za svaki predmet biće prikazan naziv predmeta, datum kreiranja, datum završetka, spisak svih advokata, kao spisak svih klijenata.

```
DECLARE
  v_id_pred predmeti.id_predmeta%type;
  v_naziv predmeti.naziv%type;
  v_dat_k predmeti.datum_kreiranja%type;
  v_dat_z predmeti.datum_zavrsetka%type;
  v_advokat varchar2(40);
  v_klijent varchar2(40);
  CURSOR kursor_predmeti IS
    SELECT id_predmeta, naziv, datum_kreiranja, datum_zavrsetka
    FROM predmeti;
  CURSOR kursor_advokati(p_id_predmeta number) IS
    SELECT ime||' '||prezime
    FROM advokati NATURAL JOIN advokati_predmeti WHERE
id_predmeta=p_id_predmeta;
```

```

        CURSOR kursor_klijenti(p_id_predmeta number) IS
            SELECT ime||' '||prezime
            FROM predmeti_klijenti NATURAL JOIN klijenti WHERE
id_predmeta=p_id_predmeta;
BEGIN
    OPEN kursor_predmeti;
    LOOP
        FETCH kursor_predmeti INTO v_id_pred, v_naziv, v_dat_k, v_dat_z;
        EXIT WHEN kursor_predmeti%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Naziv predmeta: ' || v_naziv);
        DBMS_OUTPUT.PUT_LINE('Datum kreiranja: ' || v_dat_k);
        DBMS_OUTPUT.PUT_LINE('Datum zavrsetka: ' || v_dat_z);
        DBMS_OUTPUT.PUT_LINE('Advok/ti: ');
        OPEN kursor_advokati(v_id_pred);
        LOOP
            FETCH kursor_advokati INTO v_advokat;
            EXIT WHEN kursor_advokati%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE(v_advokat);
        END LOOP;
        CLOSE kursor_advokati;
        DBMS_OUTPUT.PUT_LINE('Klijent/i: ');
        OPEN kursor_klijenti(v_id_pred);
        LOOP
            FETCH kursor_klijenti INTO v_klijent;
            EXIT WHEN kursor_klijenti%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE(v_klijent);
        END LOOP;
        CLOSE kursor_klijenti;
        DBMS_OUTPUT.PUT_LINE('-----');
    END LOOP;
    CLOSE kursor_predmeti;
END;

```

Kada se programski kod pokrene, kao rezultat nam prikazuje informacije o predmetu: naziv predmeta, datum kreiranja, datum zavrsetka, koji advokati su angažovani za rad na predmetu i klijente.

```

-----
Naziv predmeta: Krivicna prijava za nasilje u porodici
Datum kreiranja: 10/04/2019
Datum zavrsetka:
Advok/ti:
Miljana Petrovic
Milica Cvetkovic
Andrea Corak
Klijent/i:
Ana Tomasevic
-----

```

Procedure

8. Za jednog advokata ispisati spisak svih aktivnih predmeta i za svaki predmet spisak svih nadležnih organa, pomoću procedure.

Za advokatsku kancelariju ovakva procedura je vrlo korisna, ona može svakom advokatu da prikaže sve predmete na kojima trenutno radi i u kojim ustanovama se ti predmeti vode. Procedura uzima kao parametar id advokata.

```
CREATE OR REPLACE PROCEDURE  spisak_aktivnih_predmeta(p_id_advokata NUMBER)
IS
  CURSOR k_predmet IS
    SELECT id_predmeta, naziv
    FROM predmeti JOIN advokati_predmeti USING(id_predmeta)
    WHERE id_advokata=p_id_advokata AND UPPER(ishod)='PREDMET U TOKU';
  CURSOR k_nadlezni (p_id_predmeta number)IS
    SELECT naziv, tip, zvanicni_broj
    FROM nadlezni_organ_i NATURAL JOIN organi_predmeti
    WHERE id_predmeta=p_id_predmeta;
BEGIN
  FOR v_red_p IN k_predmet LOOP
    DBMS_OUTPUT.PUT_LINE('Naziv predmeta: '||v_red_p.naziv);
    FOR v_red_n in k_nadlezni(v_red_p.id_predmeta) LOOP
      DBMS_OUTPUT.PUT_LINE('    Naziv ustanove: '||v_red_n.naziv);
      DBMS_OUTPUT.PUT_LINE('    Tip ustanove: '||v_red_n.tip);
      DBMS_OUTPUT.PUT_LINE('    Zvanicni broj predmeta:'
        ||v_red_n.zvanicni_broj);
    END LOOP;
  END LOOP;
  DBMS_OUTPUT.PUT_LINE('*****');
  END LOOP;
END;
```

Sledeći primer prikazuje spisak predmeta advokata čiji je id jednak 2.

```
Naziv predmeta: Krivicna prijava za nasilje u porodici
  Naziv ustanove: Apelaciono javno tuzilastvo u Beogradu
  Tip ustanove: Tuzilastvo
  Zvanicni broj predmeta: KT-4215/2019
  Naziv ustanove: Drugi osnovni sud u Beogradu
  Tip ustanove: Sud
  Zvanicni broj predmeta: KT-4215/2019
*****
Naziv predmeta: Radno pravo prekovremeni rad
  Naziv ustanove: Drugi osnovni sud u Beogradu
  Tip ustanove: Sud
  Zvanicni broj predmeta: P1-86/2019
*****
```

Funkcije

9. Za id advokata prebrojati sve dobijene predmete, pomoću funkcije.

Advokatskoj kancelariji je ova funkcija vrlo bitna, da bi pratila uspešnost advokata. Za savki id advokata može da se proveri koliko je predmeta dobio.

```
CREATE OR REPLACE FUNCTION broj_dobijenih_predmeta(p_id_adv NUMBER)
RETURN NUMBER IS v_broj NUMBER;
BEGIN
    SELECT COUNT(*) into v_broj
    FROM advokati_predmeti NATURAL JOIN predmeti
    WHERE id_advokata=p_id_adv AND UPPER(ishod)='DOBIJENO';
    RETURN v_broj;
END;

BEGIN
    DBMS_OUTPUT.PUT_LINE(broj_dobijenih_predmeta(4));
END;
```

10. Za id advokata ispisati vrednost svih završenih predmeta, pomoću funkcije.

Advokatskoj kancelariji je ova funkcija vrlo bitna, da bi pratila zaradu advokata. Za savki id advokata može da se proveri kolika je vrednost svakog završenog predmeta.

```
CREATE OR REPLACE FUNCTION vrednost_završenih_predmeta(p_id_adv NUMBER)
RETURN NUMBER IS v_broj NUMBER;
BEGIN
    SELECT sum(vrednost) INTO v_broj
    FROM advokati_predmeti NATURAL JOIN predmeti
    WHERE id_advokata=p_id_adv AND UPPER(ishod)!='PREDMET U TOKU';
    RETURN v_broj;
END;
```

Zaključak

Za uspješni rad jedne advokatske kancelarije vrlo je bitno da ima preglednu bazu podataka. Jedna baza može da sadrži sve informacije o advokatima koji rade za advokatsku kancelariju, kao i sve bitne informacije o klijentima i predmetima koje vodi advokatska kancelarija. Važno je napraviti preglednu i dobro popunjenu bazu podataka gde su svi podaci dobro organizovani. Sve podatke unosimo u tabele koje je potrebno dobro povezati da bi prilikom kreiranja upita za ispis nekih informacija, baza dala tačne podatke. Ovakva jedna baza podataka dosta olakšava rad pri pretraživanju podataka o nekom određenom predmetu. Baza sadrži sve potrebne informacije o nadležnim organima gde se predmeti šalju kao i sve informacije o predmetima, klijentima i advokatima. Pomoću ovakve jedne baze, advokati imaju lakši pregled svih predmeta i u kojim ustanovama se predmeti vode.

U bazi podataka lako mogu da se izmene podaci koji se trenutno u njoj nalaze, kao i da se dodaju novi podaci. Broj predmeta u advokatskoj kancelariji se stalno menja, kao i informacije kad je neki predmet koji je bio u toku završen. Sve te izmene se vrlo lako unose u jednu ovakvu bazu podataka što dosta olakšava rad advokatima.

Prikazana baza podataka u ovom radu je radjena u alatu APEX koji je relacioni sistem za upravljanje bazama podataka kompanije Oracle. Programiranje u ovom alatu se realizuje pomoću jezika PL/SQL. On predstavlja proceduralni jezik koji je proširenje za SQL.

Za svakog advokata, pomoću vrlo jednostavnih upita, funkcija i procedura možemo da napravimo određen izveštaj sa podacima iz baze. Mogu da se izvuku i vrlo bitne informacije za svakog advokata: koliko je advokat imao uspešno završenih predmeta u godini ili kolika je bila vrednost tih predmeta. Pomoću samo nekih vrlo bitnih funkcija može da se dobije pregledan izveštaj o uspešnosti rada određenih advokata: koliko su imali uspešno završenih predmeta, ukupan broj predmeta na kojima rade, koliko predmeta je u toku, kao i mnoge druge informacije koje će poboljšati rad jedne advokatske kancelarije.

Literatura

- https://sr.wikipedia.org/sr-el/Релационе_базе_података
- <https://edukacija.rs/it/baze-podataka/standardni-upitni-jezik-sql>
- <https://sr.wikipedia.org/wiki/SQL>
- <https://en.wikipedia.org/wiki/PL/SQL>
- <https://apex.oracle.com/en/>
- Skripta za modele i baze podataka, Računarska gimnazija, Stanka Matković, šk. 2019/20. god
- Materijali kursa „Database Programming with PLSQL“, Oracle Academy
- <https://ilearning.oracle.com/>