



ITS INFORMATION
TECHNOLOGY
SCHOOL

VISOKA ŠKOLA STRUKOVNIH STUDIJA ZA IT

BAZE PODATAKA

Projekat

Projektovanje baze podataka za podrsku rada frizerskog salona

Predmetni nastavnik

Studenti

Katarina Jelic 370/22

Adrijan Mohora 419/22

Danijel Dejanovic 53/22

Datum predaje _____

Beograd

Januar 2024. Godine

Sadržaj

Sadržaj.....	2
1.Rezime.....	3
2. Poslovni domen – OPIS	4
2.1 Podsystem Nabavka.....	4
2.2 Podsystem Zakazivanje	4
2.3 Podsystem Prodaja	5
3. Podmodeli podataka	6
4.Integrisan model podataka(PMOV)	8
5. Prevodjenje PMOV u relacioni model	9
6. SQL naredbe za kreiranje tabela	11
7. SQL naredbe za umetanje podataka	16
8.Tabelarni prikaz sadržaj tabela	18
9.Specifikacija poslovnih transakcija i njihovih realizacija	21
9.1 Podsystem Nabavka.....	21
9.1.1 Definicija poslovne transakcije	21
9.1.2 Pogled	21
9.1.3 Bazna funkcija	22
9.1.4 Bazna procedura	22
9.2 Podsystem Zakazivanje	24
9.2.1 Definicija poslovne transakcije	24
9.2.2 Pogled	24
9.2.3 Bazna funkcija	25
9.2.4 Bazna procedura	25
9.3 Podsystem Prodaja	27
9.3.1 Definicija poslovne transakcije	27
9.3.2 Pogled	27
9.3.3 Bazna funkcija	28
9.3.4 Bazna procedura	28
10. Literatura.....	31

1.Rezime

U okviru ovog projekta, analiziran je poslovni domen frizerskog salona, sa fokusom na zadovoljenje potreba klijenata u oblastima nabavke, prodaje i zakazivanja usluga.

Kreiran je model podataka koji precizno opisuje sve relevantne aspekte poslovanja frizerskog salona. Implementacija ovog modela sprovedena je kroz kreiranje šeme relacione baze podataka, uz korišćenje SQL insert naredbi za instanciranje ove šeme baze podataka.

U sklopu projekta, pružen je tabelarni prikaz sadržaja tabela baze podataka, pružajući pregled organizacije podataka i omogućavajući efikasno upravljanje informacijama o nabavci, prodaji i zakazivanju usluga.

Za svaki podsistem frizerskog salona definisane su poslovne transakcije, pri čemu su dati i pogledi i bazne procedure koje omogućavaju sprovođenje ovih transakcija.

2. Poslovni domen – OPIS

Za potrebe frizerskog salona potrebno je isprojektovati bazu podataka sa neophodnim podacima. Poslovni domen je dekomponovan na tri celine (podsistema): Zakazivanje, Prodaja i Nabavka. U nastavku se daje kratak opis ovih celina.

2.1 Podsystem Nabavka (Danijel Dejanovic)

Frizerski salon uspostavlja kontakt sa dobavljačem od kog nabavlja artikle. Pri tome se evidentiraju osnovni podaci o dobavljaču a to su: id dobavljača, naziv, adresa i broj telefona. Svaki put kada se vrši nabavka artikala, dobavljač izrađuje fakturu kao zahtev za isplatu poručenih artikala. Faktura ima fakturu id, broj fakture, valutu plaćanja, vrednost bez pdv-a i vrednost sa pdv-om. Na fakturi se takodje nalaze stavke fakture, može se naci jedna ili više stavki u zavisnosti od toga koliko artikala je poruceno. Stavka fakture sadrži id stavke, cenu artikla, kolicinu, rabat i pdv. Artikli sadrže id artikla i naziv artikla. Frizerski salon na osnovu fakture vrši uplatu dobavljaču za porucene artikle. Pri uplati beleži se id uplate, datum i uplaćen iznos.

2.2 Podsystem Zakazivanje (Adrijan Mohora)

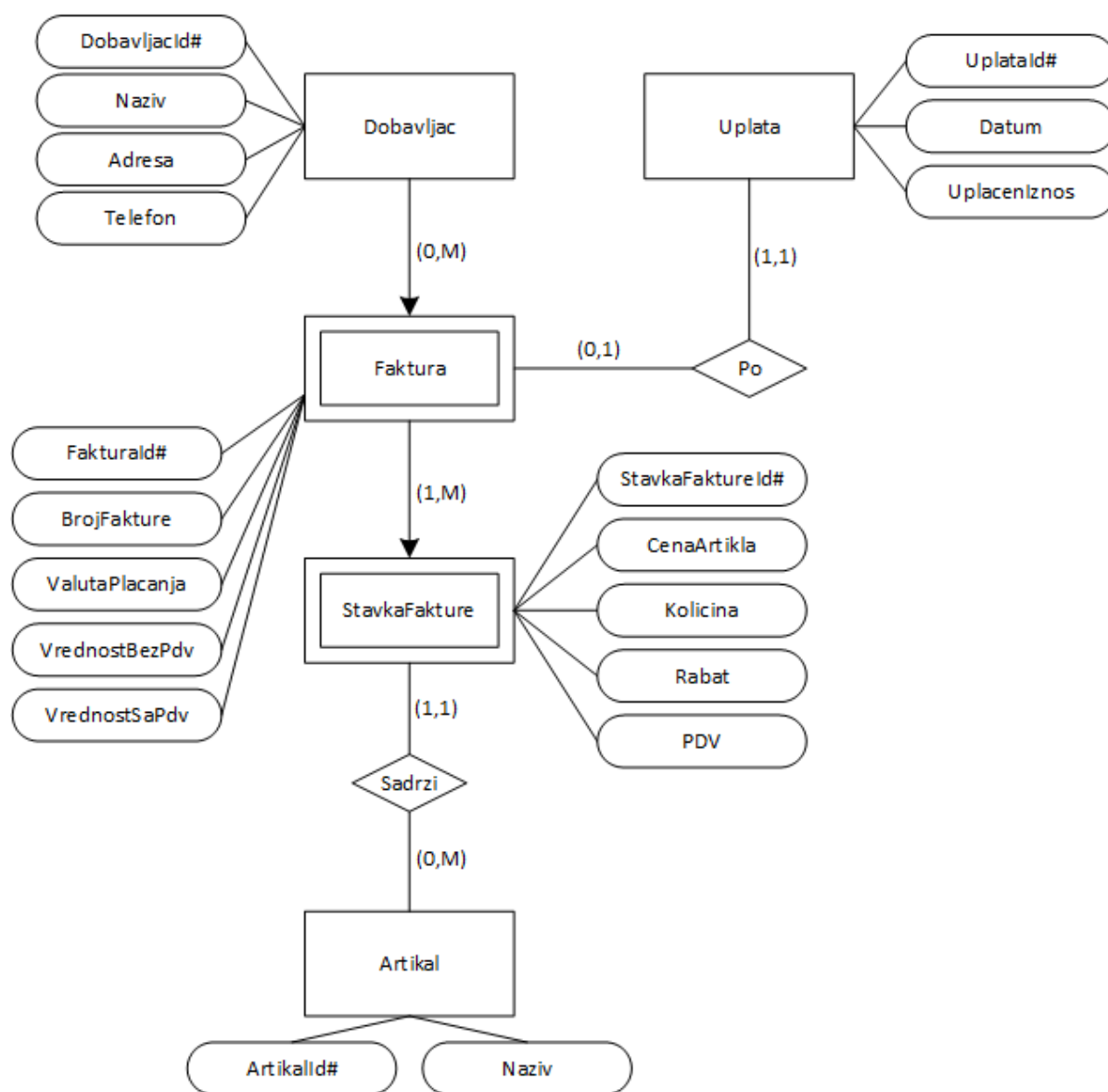
Frizerski salon, koji se može nalaziti na više lokacija u više gradova održava detaljnu evidenciju o svakom salonu, uključujući adrese na kojima se nalaze, id salona, naziv i broj telefona. Svaki salon zaposljava stručan tim frizera za koje se beleži id zaposlenog, ime, prezime, JMBG i broj telefona. Radi uspesnijeg poslovanja salona postoji menadzer koji je odgovoran za celokupan rad i organizaciju salona. Menadzer je nadređen ostalim zaposlenima u salonu. Zaposleni vrše zakazivanje termina, pri čemu se beleži datum i vreme zakazivanja. Svaka usluga koju salon pruža ima svoj id i za uslugu se beleži njeno vremensko trajanje, kao i naziv usluge. Pri zakazivanju klijenta se beleži id klijenta, ime, prezime, broj telefona klijenta i e-mail adresa.

2.3 Podsystem Prodaja (Katarina Jelic)

Osim usluga, frizerski salon nudi i mogućnost kupovine njihovih artikala. Frizerki salon ima svoj cenovnik za koji se beleži cenovnik id, datum od kad važi, datum do kad važi. Postoje i stavke cenovnika gde se mogu pronaći različiti artikli i usluge koje nudi salon. Za artikle beležimo njihov id i naziv artikala. Za usluge se beleži usluga id, naziv i vremensko trajanje te usluge. Stavke cenovnika sadrže stavku cenovnika id, popust i cenu artikala. Jedna stavka cenovnika vezana je za jedan artikal ili za jednu uslugu. Pri kupovini izdaje se račun klijentu gde se naplaćuju artikli i usluge. Na računu se beleži id računa, datum računa i ukupan iznos. Takođe se beleže stavke računa na kojim se beleži količina kupljenih artikala i iskoriscenih usluga koji su odredjeni u stavkama cenovnika. Beleži se i cena, rabat i id stavke računa. Na kraju klijent plaća svoj račun.

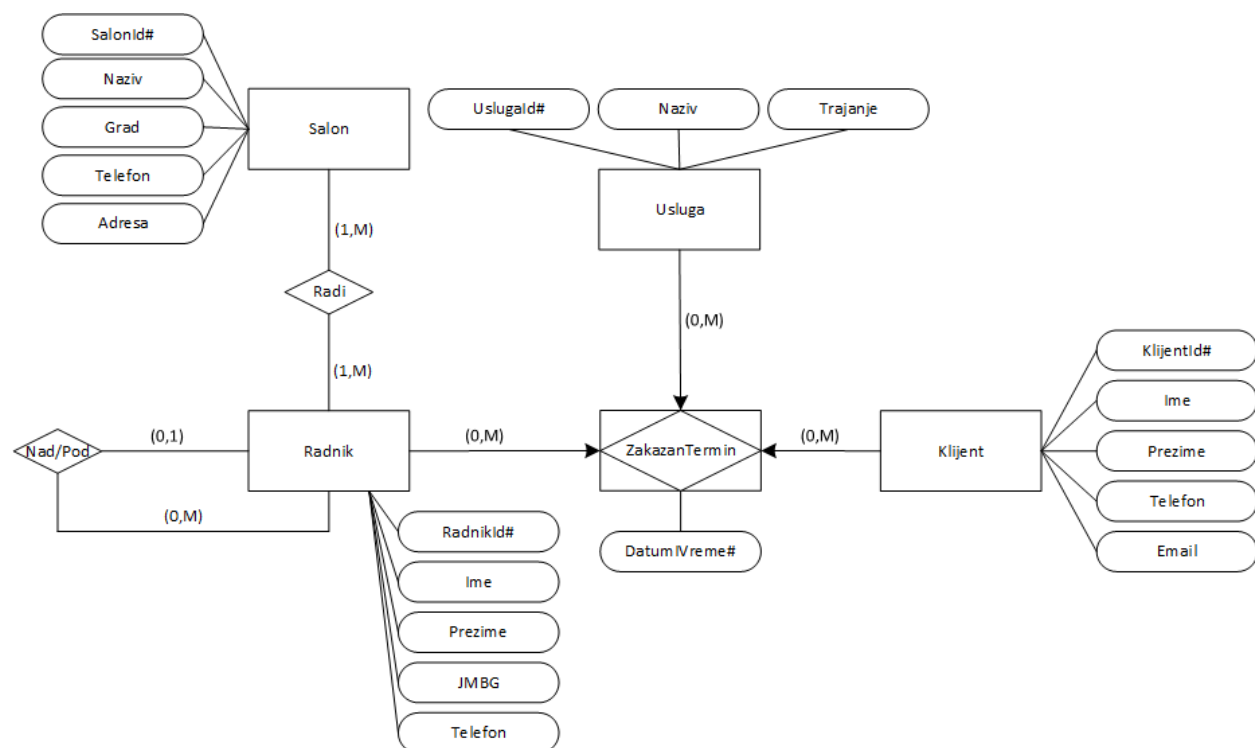
3. Podmodeli podataka

U nastavku su prikazani podmodeli podataka, napred opisanih celina, kroz podmodele PMOV.

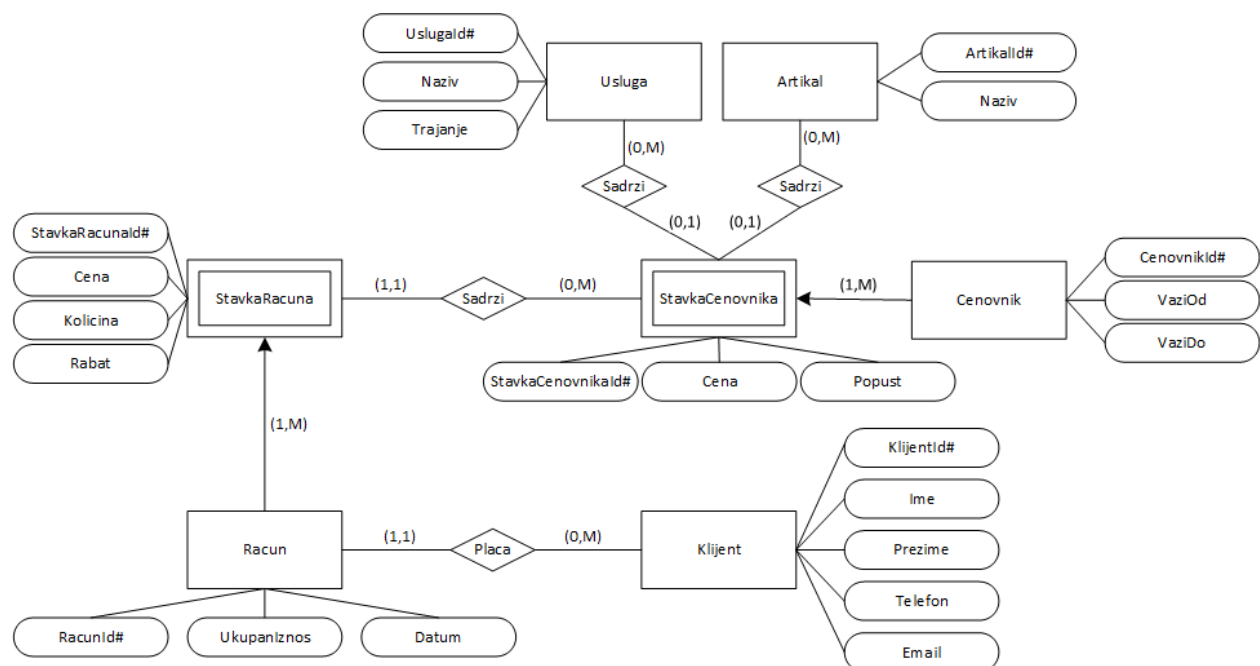


Slika 1.Podmodel PMOV-Nabavka

Projekat frizerski salon



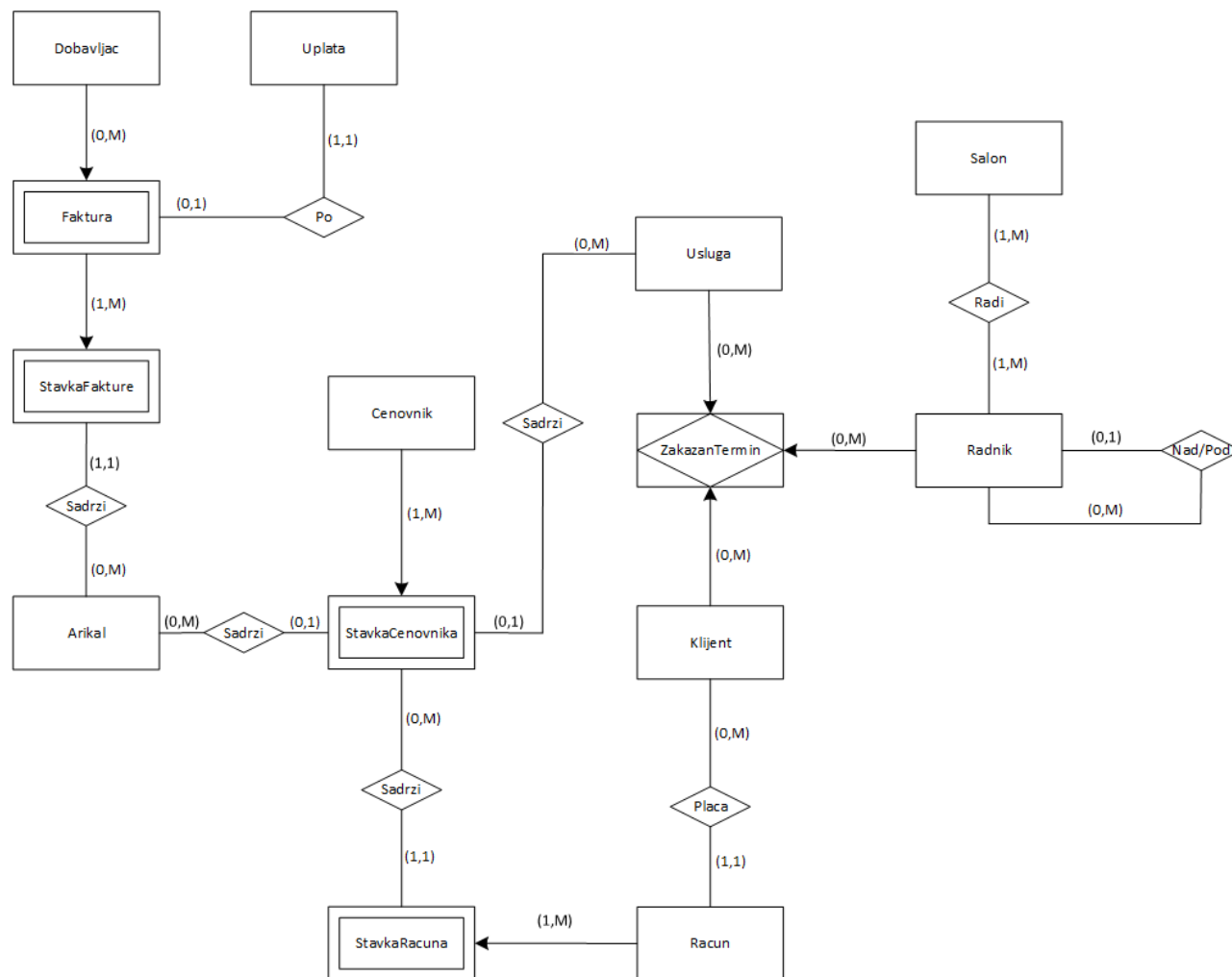
Slika 2. Podmodel PMOV-Zakazivanje



Slika 3. Podmodel PMOV-Prodaja

4.Integrisan model podataka(PMOV)

Na slici 4 je prikazan integrisan PMOV.



Slika 4.integrisan PMOV

5. Prevodjenje PMOV u relaciji model

Dobavljač(DobavljačId, Naziv, Adresa, Telefon)

Artikal(ArtikalId, Naziv)

Cenovnik(CenovnikId, VaziOd, VaziDo)

Klijent(KlijentId, Ime, Prezime, Telefon, Email)

Usluga(UslugaId, Naziv, Trajanje)

Salon(SalonId, Naziv, Grad, Telefon, Adresa)

Radnik(RadnikId, Ime, Prezime, JMBG, Telefon, NadredjenId)

Radnik(NadredjenId) references Radnik(RadnikId)

ZakazanTermin(DatumIVreme,RadnikId, UslugaId,KlijentId)

ZakazanTermin(RadnikId) references Radnik(RadnikId)

ZakazanTermin(UslugaId) references Usluga(UslugaId)

ZakazanTermin(KlijentId) references Klijent(KlijentId)

SalonRadnik(SalonId, RadnikId)

SalonRadnik(SalonId) references Salon(SalonId)

SalonRadnik(RadnikId) references Radnik(RadnikId)

Racun(RacunId, UkupanIznos, Datum, KlijentId)

Racun(KlijentId) references Klijent(KlijentId)

StavkaCenovnika(StavkaCenovnikaId,CenovnikId,Cena,Popust)

StavkaCenovnika(CenovnikId) references Cenovnik(CenovnikId)

ArtikalStavkaCenovnika(ArtikalId,StavkaCenovnikaId,CenovnikId)

ArtikalStavkaCenovnika (ArtikalId) references Artikal(ArtikalId)

ArtikalStavkaCenovnika(CenovnikId)references StavkaCenovnika(CenovnikId)

ArtikalStavkaCenovnika(StavkaCenovnikaId)references StavkaCenovnika (StavkaCenovnikaId)

UslugaStavkaCenovnika(UslugaId,CenovnikId,StavkaCenovnikaId)

UslugaStavkaCenovnika (UslugaId) references Usluga(UslugaId)

UslugaStavkaCenovnika(CenovnikId)references StavkaCenovnika(CenovnikId)

UslugaStavkaCenovnika(StavkaCenovnikaId)references StavkaCenovnika (StavkaCenovnikaId)

StavkaRacuna(StavkaRacunaId, RacunId, Cena, Kolicina, Rabat, CenovnikId, StavkaCenovnikId)

StavkaRacuna(RacunId) references Racun(RacunId)

StavkaRacuna(CenovnikId) references StavkaCenovnika(CenovnikId)

StavkaRacuna(StavkaCenovnikId) references StavkaCenovnika(StavkaCenovnikId)

Faktura(FakturaId, DobavljacId, BrojFakture, ValutaPlacanja, VrednostBezPdv, VrednostSaPdv)

Faktura(DobavljacId) references Dobavljac(DobavljacId)

Uplata(UplataId, Datum, UplacenIznos, DobavljacId, FakturaId)

Uplata(DobavljacId) references Faktura(DobavljacId)

Uplata(FakturaId) references Faktura(FakturaId)

StavkaFakture(StavkaFaktureId, DobavljacId, FakturaId, CenaArtikla, Kolicina, Rabat, PDV, ArtiklaId)

StavkaFakture(DobavljacId) references Faktura(DobavljacId)

StavkaFakture(FakturaId) references Faktura(FakturaId)

StavkaFakture(ArtiklaId) references Artikel(ArtiklaId)

6. SQL naredbe za kreiranje tabela

```
USE master;  
IF EXISTS (SELECT name FROM master.dbo.sysdatabases WHERE name = N'FrizerskiSalon')  
DROP DATABASE FrizerskiSalon;  
CREATE DATABASE FrizerskiSalon;  
USE FrizerskiSalon;
```

```
CREATE TABLE Dobavljac  
(  
    DobavljacId INT PRIMARY KEY,  
    Naziv VARCHAR(255) NOT NULL,  
    Adresa VARCHAR(255),  
    Telefon VARCHAR(15)  
);
```

```
CREATE TABLE Artikl  
(  
    ArtiklId INT PRIMARY KEY,  
    Naziv VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE Cenovnik  
(  
    CenovnikId INT PRIMARY KEY,  
    VaziOd DATE NOT NULL,  
    VaziDo DATE NOT NULL  
);
```

```
CREATE TABLE Klijent  
(  
    KlijentId INT PRIMARY KEY,  
    Ime VARCHAR(255) NOT NULL,  
    Prezime VARCHAR(255) NOT NULL,  
    Telefon VARCHAR(15) NOT NULL,  
    Email VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE Usluga  
(  
    UslugaId INT PRIMARY KEY,  
    Naziv VARCHAR(255) NOT NULL,  
    Trajanje INT NOT NULL  
);
```

```
CREATE TABLE Salon
```

```
(  
    SalonId INT PRIMARY KEY,  
    Naziv VARCHAR(255) NOT NULL,  
    Grad VARCHAR(255) NOT NULL,  
    Telefon VARCHAR(15) NOT NULL,  
    Adresa VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE Radnik
```

```
(  
    RadnikId INT PRIMARY KEY,  
    Ime VARCHAR(255) NOT NULL,  
    Prezime VARCHAR(255) NOT NULL,  
    JMBG VARCHAR(13) NOT NULL,  
    Telefon VARCHAR(15) NOT NULL,  
    NadredjenId INT,  
    FOREIGN KEY (NadredjenId) REFERENCES Radnik(RadnikId)  
);
```

```
CREATE TABLE ZakazanTermin
```

```
(  
    DatumVreme DATETIME,  
    RadnikId INT,  
    UslugaId INT,  
    KlijentId INT,  
    PRIMARY KEY (DatumVreme, RadnikId, UslugaId, KlijentId),  
    FOREIGN KEY (RadnikId) REFERENCES Radnik(RadnikId),  
    FOREIGN KEY (UslugaId) REFERENCES Usluga(UslugaId),  
    FOREIGN KEY (KlijentId) REFERENCES Klijent(KlijentId)  
);
```

```
CREATE TABLE SalonRadnik
```

```
(  
    SalonId INT,  
    RadnikId INT,  
    PRIMARY KEY (SalonId, RadnikId),  
    FOREIGN KEY (SalonId) REFERENCES Salon(SalonId),  
    FOREIGN KEY (RadnikId) REFERENCES Radnik(RadnikId)  
);
```

```
CREATE TABLE Racun
(
    RacunId INT PRIMARY KEY,
    UkupanIznos DECIMAL(10, 2) NOT NULL,
    Datum DATE NOT NULL,
    KlijentId INT,
    FOREIGN KEY (KlijentId) REFERENCES Klijent(KlijentId)
);

CREATE TABLE StavkaCenovnika
(
    StavkaCenovnikId INT,
    CenovnikId INT,
    Cena DECIMAL(10, 2) NOT NULL,
    Popust DECIMAL(5, 2) NOT NULL,
    PRIMARY KEY(StavkaCenovnikId, CenovnikId),
    FOREIGN KEY (CenovnikId) REFERENCES Cenovnik(CenovnikId)
);

CREATE TABLE ArtiklStavkaCenovnika
(
    ArtiklId INT,
    StavkaCenovnikId INT,
    CenovnikId INT,
    PRIMARY KEY (ArtiklId, StavkaCenovnikId, CenovnikId),
    FOREIGN KEY (ArtiklId) REFERENCES Artikl(ArtiklId),
    FOREIGN KEY (StavkaCenovnikId, CenovnikId) REFERENCES
    StavkaCenovnika(StavkaCenovnikId, CenovnikId)
);

CREATE TABLE UslugaStavkaCenovnika
(
    UslugaId INT,
    CenovnikId INT,
    StavkaCenovnikId INT,
    PRIMARY KEY (UslugaId, StavkaCenovnikId, CenovnikId),
    FOREIGN KEY (UslugaId) REFERENCES Usluga(UslugaId),
    FOREIGN KEY (StavkaCenovnikId, CenovnikId) REFERENCES
    StavkaCenovnika(StavkaCenovnikId, CenovnikId)
);
```

```
CREATE TABLE StavkaRacuna
(
    StavkaRacunId INT,
    RacunId INT,
    Cena DECIMAL(10, 2) NOT NULL,
    Kolicina INT NOT NULL,
    Rabat DECIMAL(5, 2) NOT NULL,
    StavkaCenovnikId INT,
    CenovnikId INT,
    PRIMARY KEY (StavkaRacunId, RacunId),
    FOREIGN KEY (RacunId) REFERENCES Racun(RacunId),
    FOREIGN KEY (StavkaCenovnikId, CenovnikId) REFERENCES
    StavkaCenovnika(StavkaCenovnikId, CenovnikId)
);
```

```
CREATE TABLE Faktura
(
    FakturalId INT,
    DobavljacId INT,
    BrojFakture VARCHAR(50) NOT NULL,
    ValutaPlacanja VARCHAR(50) NOT NULL,
    VrednostBezPdv DECIMAL(10, 2) NOT NULL,
    VrednostSaPdv DECIMAL(10, 2) NOT NULL,
    PRIMARY KEY (FakturalId, DobavljacId),
    FOREIGN KEY (DobavljacId) REFERENCES Dobavljac(DobavljacId)
);
```

```
CREATE TABLE Uplata
(
    UplataId INT PRIMARY KEY,
    Datum DATE NOT NULL,
    UplacenIznos DECIMAL(10, 2) NOT NULL,
    FakturalId INT,
    DobavljacId INT,
    FOREIGN KEY (FakturalId, DobavljacId) REFERENCES Faktura(FakturalId, DobavljacId)
);
```

```
CREATE TABLE StavkaFakture
(
    StavkaFaktureId INT,
    FakturalId INT,
    DobavljacId INT,
    CenaArtikla DECIMAL(10, 2) NOT NULL,
```

Projekat frizerski salon

```
Kolicina INT NOT NULL,  
Rabat DECIMAL(5, 2) NOT NULL,  
PDV DECIMAL(5, 2) NOT NULL,  
ArtikalId INT,  
PRIMARY KEY (StavkaFaktureId, Fakturald, DobavljacId),  
FOREIGN KEY (Fakturald, DobavljacId) REFERENCES Faktura(Fakturald, DobavljacId),  
FOREIGN KEY (ArtikalId) REFERENCES Artikel(ArtikalId)  
);
```

7. SQL naredbe za umetanje podataka

```
INSERT INTO Dobavljac (DobavljaId, Naziv, Adresa, Telefon)
VALUES (1, 'Balea', 'Vucka Milicevica 9', '0634554556'),
       (2, 'Loreal', 'Sumadijska 7', '0643344443');
```

```
INSERT INTO Artikal (ArtiklId, Naziv)
VALUES (1, 'Sampon'),
       (2, 'Gel');
```

```
INSERT INTO Cenovnik (CenovnikId, VaziOd, VaziDo)
VALUES (1, '2024-01-01', '2024-12-31');
```

```
INSERT INTO Klijent (KlijentId, Ime, Prezime, Telefon, Email)
VALUES (1, 'Marko', 'Markovic', '061231234', 'marko@gmail.com'),
       (2, 'Ana', 'Anic', '987654321', 'ana@gmail.com');
```

```
INSERT INTO Usluga (UslugaId, Naziv, Trajanje)
VALUES (1, 'Sisanje', 30),
       (2, 'Feniranje', 45);
```

```
INSERT INTO Salon (SalonId, Naziv, Grad, Telefon, Adresa)
VALUES (1, 'Butterfly', 'Beograd', '0636777889', 'Futoska 1'),
       (2, 'Beauty', 'Novi Sad', '068766545', 'Zlatiborska 2');
```

```
INSERT INTO Radnik (RadnikId, Ime, Prezime, JMBG, Telefon, NadredjenId)
VALUES (1, 'Nikola', 'Nikolic', '1234567890123', '0645634535', null),
       (2, 'Jovana', 'Jovanovic', '9876543210987', '0657898976', 1);
```

```
INSERT INTO ZakazanTermin (DatumIVreme, RadnikId, UslugaId, KlijentId)
VALUES ('2024-01-15 10:00:00', 1, 1, 1),
       ('2024-01-20 15:00:00', 2, 2, 2);
```

```
INSERT INTO SalonRadnik (SalonId, RadnikId)
VALUES (1, 1),
       (2, 2);
```

```
INSERT INTO StavkaCenovnika (StavkaCenovnikaId, CenovnikId, Cena, Popust)
VALUES (1, 1, 1000.00, 0.0),
       (2, 1, 1500.00, 0.0),
       (3, 1, 800.00, 0.0),
       (4, 1, 2000.00, 0.0);
```



```
INSERT INTO ArtikalStavkaCenovnika (ArtikalId, StavkaCenovnikId, CenovnikId)
VALUES (1, 1, 1),
       (2, 2, 1);
```

```
INSERT INTO UslugaStavkaCenovnika (UslugaId, StavkaCenovnikId, CenovnikId)
VALUES (1, 3, 1),
       (2, 4, 1);
```

```
INSERT INTO Racun (RacunId, UkupanIznos, Datum, KlijentId)
VALUES (1, 2350.00, '2024-01-15', 1),
       (2, 2000.00, '2024-01-20', 2);
```

```
INSERT INTO StavkaRacuna (StavkaRacunId, RacunId, Cena, Kolicina, Rabat, StavkaCenovnikId,
CenovnikId)
VALUES (1, 1, 1000.00, 1, 0.0, 1, 1),
       (2, 1, 1500.00, 1, 10.0, 2, 1),
       (3, 2, 2000.00, 1, 0.0, 4, 1);
```

```
INSERT INTO Faktura (Fakturald, DobavljaclId, BrojFakture, ValutaPlacanja, VrednostBezPdv,
VrednostSaPdv)
VALUES (1, 1, 'F2024001', 'din', 900.00,1000.00),
       (2, 2, 'F2024002', 'din', 1400.00, 1500.00),
       (3,2,'F2024003','din',1900,2000);
```

```
INSERT INTO Uplata (Uplatald, Datum, UplacenIznos, Fakturald, DobavljaclId)
VALUES (1, '2024-01-20', 1000.00, 1, 1),
       (2, '2024-01-25', 1500.00, 2, 2);
```

```
INSERT INTO StavkaFakture (StavkaFaktureId, Fakturald, DobavljaclId, CenaArtikla, Kolicina,
Rabat, PDV, ArtikalId)
VALUES (1, 1, 1, 1000.00, 2, 0.0, 0.1, 1),
       (2, 1, 1, 1500.00, 1, 0.0, 0.1, 2);
```

8. Tabelarni prikaz sadrzaj tabela

	Artiklald	Naziv
1	1	Sampon
2	2	Gel

Slika 1. Tabela-Artikal

	Artikalld	StavkaCenovnikald	Cenovnikld
1	1	1	1
2	2	2	1

Slika 2. Tabela-ArtikalStavkaCenovnika

	Cenovnikld	VaziOd	VaziDo
1	1	2024-01-01	2024-12-31

Slika 3. Tabela-Cenovnik

	Dobavljaclld	Naziv	Adresa	Telefon
1	1	Balea	Vucka Milicevica 9	0634554556
2	2	Loreal	Sumadijska 7	0643344443

Slika 4. Tabela-Dobavljac

	Fakturald	Dobavljaclld	BrojFakture	ValutaPlacanja	VrednostBezPdv	VrednostSaPdv
1	1	1	F2024001	din	900.00	1000.00
2	2	2	F2024002	din	1400.00	1500.00
3	3	2	F2024003	din	1900.00	2000.00

Slika 5. Tabela-Faktura

	Klijentld	Ime	Prezime	Telefon	Email
1	1	Marko	Markovic	061231234	marko@gmail.com
2	2	Ana	Anic	987654321	ana@gmail.com

Slika 6. Tabela-Klijent

	Racunld	UkupanIznos	Datum	Klijentld
1	1	2350.00	2024-01-15	1
2	2	2000.00	2024-01-20	2

Slika 7. Tabela-Racun

Projekat frizerski salon

	RadnikId	Ime	Prezime	JMBG	Telefon	NadredjenId
1	1	Nikola	Nikolic	1234567890123	0645634535	NULL
2	2	Jovana	Jovanovic	9876543210987	0657898976	1

Slika 8. Tabela-Radnik

	SalonId	Naziv	Grad	Telefon	Adresa
1	1	Butterfly	Beograd	0636777889	Futoska 1
2	2	Beauty	Novi Sad	068766545	Zlatiborska 2

Slika 9. Tabela-Salon

	SalonId	RadnikId
1	1	1
2	2	2

Slika 10. Tabela-SalonRadnik

	StavkaCenovnikId	CenovnikId	Cena	Popust
1	1	1	1000.00	0.00
2	2	1	1500.00	0.00
3	3	1	800.00	0.00
4	4	1	2000.00	0.00

Slika 11. Tabela-StavkaCenovnika

	StavkaFakturaId	FakturaId	DobavljaId	CenaArtikla	Kolicina	Rabat	PDV	ArtikalId
1	1	1	1	1000.00	2	0.00	0.10	1
2	2	1	1	1500.00	1	0.00	0.10	2

Slika 12. Tabela-StavkaFakture

	StavkaRacunId	RacunId	Cena	Kolicina	Rabat	StavkaCenovnikId	CenovnikId
1	1	1	1000.00	1	0.00	1	1
2	2	1	1500.00	1	10.00	2	1
3	3	2	2000.00	1	0.00	4	1

Slika 13. Tabela-StavkaRacuna

	UplataId	Datum	UplacenIznos	FakturaId	DobavljaId
1	1	2024-01-20	1000.00	1	1
2	2	2024-01-25	1500.00	2	2

Slika 14. Tabela-Uplata

	Uslugald	Naziv	Trajanje
1	1	Sisanje	30
2	2	Feniranje	45

Slika 15. Tabela-Usluga

	Uslugald	CenovnikId	StavkaCenovnikald
1	1	1	3
2	2	1	4

Slika 16. Tabela-UslugaStavkaCenovnika

	Datum/Vreme	RadnikId	Uslugald	KlijentId
1	2024-01-15 10:00:00.000	1	1	1
2	2024-01-20 15:00:00.000	2	2	2

Slika 17. Tabela-ZakazanTermin

9.Specifikacija poslovnih transakcija i njihovih realizacija

Na osnovu podsistema pokazuje se realizacija poslovnih transakcija, pogleda, baznih funkcija kao i baznih procedura. Testira se izvršavanje ove baze.

Poslovne transakcije se opisuju na način koji je lak za razumevanje.

Kada se ove transakcije sprovedu u bazi podataka, to znači da će se nešto promeniti u podacima.

Pogledi se definišu tako da prikažu bitne podatke relevantne za određenu poslovnu transakciju. Njihova svrha je olakšavanje pisanja osnovnih procedura koje sprovode tu transakciju, kao i osnovnih funkcija koje se koriste za testiranje izvršenja transakcije. Ove funkcije su podešene tako da se odnose direktno na delove baze podataka nad kojima se vrše promene, kako pre izvršenja osnovne procedure, tako i nakon njenog izvršenja.

9.1 Podsystem Nabavka

9.1.1 Definicija poslovne transakcije

Ova transakcija započinje kada frizerski salon identifikuje potrebe za određenim artiklima. Nakon identifikacije potreba, salon kontaktira dobavljače kako bi proverio dostupnost artikala i cene. Nakon što se postigne dogovor sa dobavljačem, pristupa se izdavanju fakture, koja sadrži detalje o artiklima. Kada artikli stignu u salon, zaposleni proveravaju isporučene artikle kako bi se osiguralo da su svi proizvodi prisutni i u dobrom stanju. Plaćanje za artikle se obavlja u skladu sa dogovorenim uslovima plaćanja navedenim na fakturi.

9.1.2 Pogled

Ovaj pogled prikazuje informacije o fakturama i uplatama dobavljača. Daje pregled ID-ova faktura, brojeva faktura, valute plaćanja, vrednosti bez PDV-a i sa PDV-om, ID-ova dobavljača, naziva dobavljača i broja uplata povezanih sa svakom fakturom.

```
CREATE VIEW pFakturaUplata
AS
SELECT
    F.Fakturald,
    F.BrojFakture,
    F.ValutaPlacanja,
    F.VrednostBezPdv,
    F.VrednostSaPdv,
    D.DobavljaclD,
    D.Naziv AS NazivDobavljacka,
    COUNT(U.Uplatald) AS BrojUplata
FROM
    Faktura F
JOIN
    Dobavljac D ON F.DobavljaclD = D.DobavljaclD
```

Projekat frizerski salon

JOIN

Uplata U ON F.Fakturald = U.Fakturald

GROUP BY

F.Fakturald, F.BrojFakture, F.ValutaPlacanja, F.VrednostBezPdv, F.VrednostSaPdv, D.DobavljaclId, D.Naziv;

--Pozivanje pogleda

SELECT *

FROM pFakturaUplata;

Sadržaj koji upit vraća dat je u sledećoj tabeli.

	Fakturald	BrojFakture	ValutaPlacanja	VrednostBezPdv	VrednostSaPdv	DobavljaclId	NazivDobavljacka	BrojUplata
1	1	F2024001	din	900.00	1000.00	1	Balea	1
2	2	F2024002	din	1400.00	1500.00	2	Loreal	1

9.1.3 Bazna funkcija

Ova funkcija omogućava dobavljanje informacija o fakturama za određenog dobavljača. Prima jedan ulazni parametar - ID dobavljača. Vraća tabelu koja sadrži informacije o fakturama za tog dobavljača, uključujući ID faktura, brojeve faktura, valutu plaćanja, vrednost bez PDV-a i sa PDV-om, kao i broj uplata vezanih za svaku fakturu.

CREATE FUNCTION kdfFaktureDobavljacka

(@DobavljaclId INT)

RETURNS TABLE

AS

RETURN

(SELECT p.DobavljaclId, p.NazivDobavljacka, p.Fakturald, p.BrojFakture, p.ValutaPlacanja,

p.VrednostBezPdv, p.VrednostSaPdv, p.BrojUplata

FROM pFakturaUplata p

WHERE p.DobavljaclId = @DobavljaclId);

--Pozivanje funkcije

SELECT *

FROM kdfFaktureDobavljacka(1);

Funkcija vraća vrednost u tabeli, i ta tabela je prikazana u nastavku.

	DobavljaclId	NazivDobavljacka	Fakturald	BrojFakture	ValutaPlacanja	VrednostBezPdv	VrednostSaPdv	BrojUplata
1	1	Balea	1	F2024001	din	900.00	1000.00	1

9.1.4 Bazna procedura

Ova procedura omogućava kreiranje uplate za određenu fakturu. Prima tri ulazna parametra: datum uplate, uplaćeni iznos i id fakture. Procedura prvo dobavlja identifikator dobavljača vezanog za fakturu, zatim generiše novi id uplate i unosi podatke o uplati u tabelu Uplata. Ukoliko proces uspe, transakcija se potvrđuje, a u suprotnom se poništava, uz odgovarajuću poruku o grešci.

CREATE PROCEDURE kdpKreiranjeUplata

Projekat frizerski salon

```
@Datum DATE,
@UplacenIznos DECIMAL(10, 2),
@Fakturald INT
AS
BEGIN TRY
    BEGIN TRANSACTION

    DECLARE @dobavljacId INT, @uplataId INT;

    SELECT @dobavljacId = DobavljacId
    FROM Faktura
    WHERE Fakturald = @Fakturald

    SELECT @uplataId = MAX(UplataId) + 1
    FROM Uplata

    INSERT INTO Uplata (UplataId ,Datum, UplacenIznos, Fakturald, DobavljacId)
    VALUES (@uplataId, @Datum, @UplacenIznos, @Fakturald, @dobavljacId)

    COMMIT TRANSACTION
END TRY
BEGIN CATCH
    PRINT 'DOŠLO JE DO GREŠKE!'
    PRINT '-----Proverite ulazne parametre-----'
    ROLLBACK TRANSACTION;
    PRINT 'Transakcija je poništena.'
END CATCH;

--Pozivanje procedure
EXEC kdpKreiranjeUplate
    @Datum ='2024-03-01',
    @UplacenIznos =2000;
    @Fakturald =3;
```

Provera izvršenja poslovne transakcije, tj. promena u bazi podataka koje su prouzrokovane pozivom bazne procedure.

	UplataId	Datum	UplacenIznos	FakturalId	DobavljaId
1	1	2024-01-20	1000.00	1	1
2	2	2024-01-25	1500.00	2	2
3	3	2024-03-01	2000.00	3	2

9.2 Podsystem Zakazivanje

9.2.1 Definicija poslovne transakcije

Zakazivanje termina za klijente u frizerskom salonu je ključni proces koji omogućava pružanje usluga u salonu. Ova transakcija započinje kada klijent kontaktira salon radi zakazivanja termina za usluge. Zaposleni u salonu beleže informacije o zakazivanju, kao i vrstu usluge koju klijent želi. Ovi podaci se unose u sistem za zakazivanje termina.

9.2.2 Pogled

Pogled pZakazaniTermini omogućava prikaz zakazanih termina u frizerskom salonu, uključujući datum i vreme, ime i prezime radnika, ime i prezime klijenta, kontakt telefon klijenta, naziv usluge, očekivano trajanje usluge u minutima, kao i identifikacione brojeve radnika, klijenta i usluge. Povezuje informacije iz tabela ZakazanTermin, Klijent, Radnik i Usluga. Ovaj pogled olakšava pristup relevantnim informacijama o zakazanim terminima, omogućavajući efikasnu organizaciju rada u salonu.

```
CREATE VIEW pZakazaniTermini
AS
SELECT ZT.DatumIVreme,
       R.Ime + ' ' + R.Prezime AS Radnik,
       K.Ime + ' ' + K.Prezime AS Klijent,
       K.Telefon AS KlijentTelefon,
       U.Naziv AS Usluga,
       U.Trajanje As OcekivanoTrajanjeUMinutima,
       ZT.RadnikId,
       ZT.KlijentId,
       ZT.UslugaId
FROM ZakazanTermin ZT
JOIN Klijent K
ON K.KlijentId = ZT.KlijentId
JOIN Radnik R
ON R.RadnikId = ZT.RadnikId
JOIN Usluga U
```


Projekat frizerski salon

ON U.Uslugald = ZT.Uslugald

--Pozivanje pogleda

SELECT *

FROM pZakazaniTermini;

Sadržaj koji upit vraća dat je u sledećoj tabeli.

	DatumIVreme	Radnik	Klijent	KlijentTelefon	Usluga	OcekivanoTrajanjeUMinutima	RadnikId	KlijentId	Uslugald
1	2024-01-15 10:00:00.000	Nikola Nikolic	Marko Markovic	061231234	Sisanje	30	1	1	1
2	2024-01-20 15:00:00.000	Jovana Jovanovic	Ana Anic	987654321	Feniranje	45	2	2	2

9.2.3 Bazna funkcija

Ova funkcija kdfZakazaniTerminiZaDatum prima datum kao ulazni argument i vraća tabelu sa informacijama o zakazanim terminima za taj određeni datum. Koristi se unapred definisani pogled pZakazaniTermini kako bi filtrirala podatke samo za željeni datum. Funkcija omogućava lak pristup informacijama o zakazanim terminima za određeni dan.

```
CREATE FUNCTION kdfZakazaniTerminiZaDatum
(@Datum DATE)
RETURNS TABLE
AS
RETURN
(SELECT *
FROM pZakazaniTermini p
WHERE CAST(p.DatumIVreme AS DATE) = @Datum);
--Pozivanje funkcije
SELECT *
FROM kdfZakazaniTerminiZaDatum('2024-01-15');
```

Funkcija vraća vrednost u tabeli, i ta tabela je prikazana u nastavku.

	DatumIVreme	Radnik	Klijent	KlijentTelefon	Usluga	OcekivanoTrajanjeUMinutima	RadnikId	KlijentId	Uslugald
1	2024-01-15 10:00:00.000	Nikola Nikolic	Marko Markovic	061231234	Sisanje	30	1	1	1

9.2.4 Bazna procedura

Ova procedura kdpZakazivanjeTermina omogućava zakazivanje termina za određeni datum i vreme, za određenog radnika, određenu uslugu i određenog klijenta. Prvo se proverava da li je termin već zauzet za dati datum i vreme i za datog radnika. Ako termin nije zauzet, tada se vrši unos novog zapisa u tabelu ZakazanTermin. Ako je termin već zauzet, ispisuje se poruka da termin nije slobodan.

```
CREATE PROCEDURE kdpZakazivanjeTermina
@DatumIVreme DATETIME,
@RadnikId INT,
@Uslugald INT,
@KlijentId INT
```

Projekat frizerski salon

```
AS
BEGIN TRY
    BEGIN TRANSACTION
    IF NOT EXISTS (
        SELECT 1
        FROM ZakazanTermin
        WHERE DatumIVreme = @DatumIVreme
        AND RadnikId = @radnikId
    )
    BEGIN
        INSERT INTO ZakazanTermin(DatumIVreme, RadnikId, UslugaId, KlijentId)
        VALUES (@DatumIVreme, @RadnikId, @UslugaId, @KlijentId)
    END
ELSE
    BEGIN
        PRINT 'Termin nije slobodan'
    END
    COMMIT TRANSACTION
END TRY
BEGIN CATCH
    PRINT 'DOŠLO JE DO GREŠKE!'
    PRINT '-----Proverite ulazne parametre-----'
    ROLLBACK TRANSACTION;
    PRINT 'Transakcija je poništena.'
END CATCH;
--Pozivanje procedure
EXEC kdpZakazivanjeTermina
    @DatumIVreme = '2024-01-27',
    @RadnikId = 1,
    @UslugaId = 2,
    @KlijentId = 1;
```

Provera izvršenja poslovne transakcije, tj. promena u bazi podataka koje su prouzrokovane pozivom baze procedure

	DatumIVreme	RadnikId	UslugaId	KlijentId
1	2024-01-15 10:00:00.000	1	1	1
2	2024-01-20 15:00:00.000	2	2	2
3	2024-01-27 10:00:00.000	1	2	1

9.3 Podsystem Prodaja

9.3.1 Definicija poslovne transakcije

Kada klijent završi sa frizerskim tretmanom ili odabere proizvode za kupovinu, zaposleni beleže sve usluge i artikle koji su korišćeni ili kupljeni. Zatim, koristeći informacije iz cenovnika salona, izračunavaju ukupan iznos koji se odnosi na usluge i artikle koje je klijent koristio ili kupio. Na osnovu toga, kreiraju račun koji detaljno prikazuje cene svake stavke, kao i ukupan iznos koji klijent treba da plati.

9.3.2 Pogled

`pStavkeCenovnika` je pogled koji prikuplja informacije o cenama proizvoda i usluga iz cenovnika. Za svaku stavku cenovnika prikazuje se naziv proizvoda ili usluge, cena, trajanje usluge kao i ID-jevi artikla, usluge i cenovnika koji su povezani sa tom stavkom

```
CREATE VIEW pStavkeCenovnika AS
SELECT SC.StavkaCenovnikId,
       COALESCE(A.Naziv, U.Naziv) AS Proizvod,
       SC.Cena,
       U.Trajanje,
       A.ArtiklId,
       U.UslugaId,
       SC.CenovnikId
FROM StavkaCenovnika SC
left JOIN ArtikalStavkaCenovnika ASCN
ON ASCN.StavkaCenovnikId = SC.StavkaCenovnikId
left JOIN Artikal A
ON A.ArtiklId = ASCN.ArtiklId
left JOIN UslugaStavkaCenovnika USCN
ON USCN.StavkaCenovnikId = SC.StavkaCenovnikId
left JOIN Usluga U
ON U.UslugaId = USCN.UslugaId
--Pozivanje pogleda
SELECT *
FROM pStavkeCenovnika;
```

Sadržaj koji upit vraća dat je u sledećoj tabeli.

	StavkaCenovnikald	Proizvod	Cena	Trajanje	Artiklald	Uslugald	Cenovnikld
1	1	Sampon	1000.00	NULL	1	NULL	1
2	2	Gel	1500.00	NULL	2	NULL	1
3	3	Sisanje	800.00	30	NULL	1	1
4	4	Feniranje	2000.00	45	NULL	2	1

9.3.3 Bazna funkcija

Ova funkcija, kdfStavkePoCenovniku, pruža korisnicima mogućnost da dobiju sve stavke cenovnika koje su povezane sa određenim cenovnikom. Koristeći ID cenovnika kao ulazni parametar, funkcija vrši filtriranje podataka i vraća sve relevantne stavke cenovnika koje su vezane za taj određeni cenovnik.

```
CREATE FUNCTION kdfStavkePoCenovniku
```

```
(@Cenovnikld INT)
```

```
RETURNS TABLE
```

```
AS
```

```
RETURN
```

```
(SELECT *
```

```
FROM pStavkeCenovnika sc
```

```
WHERE sc.Cenovnikld = @Cenovnikld);
```

```
--Pozivanje funkcije
```

```
SELECT *
```

```
FROM kdfStavkePoCenovniku(1);
```

Funkcija vraća vrednost u tabeli, i ta tabela je prikazana u nastavku.

	StavkaCenovnikald	Proizvod	Cena	Trajanje	Artiklald	Uslugald	Cenovnikld
1	1	Sampon	1000.00	NULL	1	NULL	1
2	2	Gel	1500.00	NULL	2	NULL	1
3	3	Sisanje	800.00	30	NULL	1	1
4	4	Feniranje	2000.00	45	NULL	2	1

9.3.4 Bazna procedura

Ova procedura dodaje novu uslugu u sistem frizerskog salona. Prvo se određuju ID za novu uslugu i stavku cenovnika. Zatim se pronalazi trenutni cenovnik za današnji datum. Nakon toga, nova usluga se dodaje u tabelu Usluga, a nova stavka cenovnika u tabelu StavkaCenovnika, povezana sa trenutnim cenovnikom.

```
CREATE PROCEDURE kdpDodavanjeUsluge
```

```
@Naziv VARCHAR(255),
```

```
@Trajanje INT,
```

```
@Cena DECIMAL(10,2),
```

```
@Popust DECIMAL(5,2)
```

```
AS
```

Projekat frizerski salon

```
BEGIN TRY
    BEGIN TRANSACTION

    DECLARE @uslugald INT
    SELECT @uslugald = MAX(Uslugald) + 1
    FROM Usluga

    DECLARE @stavkaCenovnikald INT
    SELECT @stavkaCenovnikald = MAX(StavkaCenovnikald) + 1
    FROM StavkaCenovnika

    DECLARE @cenovnikId INT
    SELECT @cenovnikId = CenovnikId
    FROM Cenovnik
    WHERE GETDATE() BETWEEN VaziOd AND VaziDo

    INSERT INTO Usluga(Uslugald, Naziv, Trajanje)
    VALUES(@uslugald, @Naziv, @Trajanje)

    INSERT INTO StavkaCenovnika(StavkaCenovnikald, CenovnikId, Cena, Popust)
    VALUES (@stavkaCenovnikald, @cenovnikId, @Cena, @Popust)

    INSERT INTO UslugaStavkaCenovnika(Uslugald, CenovnikId, StavkaCenovnikald)
    VALUES (@uslugald, @cenovnikId, @stavkaCenovnikald)

    COMMIT TRANSACTION
END TRY
BEGIN CATCH
    PRINT 'DOŠLO JE DO GREŠKE!'
    PRINT '-----Proverite ulazne parametre-----'
    ROLLBACK TRANSACTION;
    PRINT 'Transakcija je poništena.'
END CATCH;
--Pozivanje procedure
EXEC kdpDodavanjeUsluge
@Naziv = 'Keratin',
@Trajanje = 180,
@Cena = 7000,
@Popust = 0
```

Projekat frizerski salon

Provera izvršenja poslovne transakcije, tj. promena u bazi podataka koje su prouzrokovane pozivom bazne procedure.

	Uslugald	Naziv	Trajanje
1	1	Sisanje	30
2	2	Feniranje	45
3	3	Keratin	180

	StavkaCenovnikald	Cenovnikld	Cena	Popust
1	1	1	1000.00	0.00
2	2	1	1500.00	0.00
3	3	1	800.00	0.00
4	4	1	2000.00	0.00
5	5	1	7000.00	0.00

	Uslugald	Cenovnikld	StavkaCenovnikald
1	1	1	3
2	2	1	4
3	3	1	5

10. Literatura

1. Majstorović M.: Baze podataka, skripta, Visoka škola strukovnih studija za informacione tehnologije, Beograd, 2013.
2. M. Majstorović, Uvod u baze podataka, Visoka škola strukovnih studija za informacione tehnologije, Beograd, 2015, ISBN 978-86-89007-03-9.