

Sveučilište Josipa Jurja Strossmayera u Osijeku
Fakultet elektrotehnike, računarstva i informacijskih tehnologija

Seminarski rad iz kolegija: Baze podataka
Poljoprivredna ljekarna

Student: Antun, Derek, AI4538

Osijek, 2018

SADRŽAJ

1. UTVRĐIVANJE I ANALIZA ZAHTJEVA.....	1
2. KONCEPTUALNO OBLIKOVANJE.....	3
3. LOGIČKO OBLIKOVANJE.....	4
4. FIZIČKO OBLIKOVANJE.....	6
5. UNOS, UKLANJANJE I PRETRAŽIVANJE PODATAKA.....	22
6. ZAKLJUČAK.....	25

1. UTVRĐIVANJE I ANALIZA ZAHTJEVA

Specifikacija:

Svaki poslovni partner postavlja račun za kupovinu određenih artikala u zahtijevanim količinama. Svaki poslovni partner može imati više računa. Račun ispunjava jedan djelatnik koji je ujedno i šef. Račun treba sadržavati popis robe i datum izdavanja. Popis robe sadrži sve artikle i njihovu količinu koji su vezani za određeni račun. Cijene proizvoda se čuvaju kao cijene bez PDV-a, dok se cijena računa poslovnih partnera prikazuje kao cijena narudžbe + PDV i cijena bez PDV-a. Baza treba imati mogućnost dodavanja i uklanjanja artikla, djelatnika, dobavljača, poslovnih partnera i računa. Unosom određenog artikla u račun odnosno popis robe potrebno je smanjiti ukupnu količinu artikla. Svaki od djelatnika, dobavljača, poslovnih partnera moraju imati status koji nam označavaju da li su u još uvijek aktivni u poslu. Artikli trebaju imati stanje da li su u prodaji ili ne. Svaki artikl treba imati informacije o dobavljaču.

Analizom specifikacije su otkriveni sljedeći entiteti, veze i atributi.

Entiteti: ARTIKL, POPIS ROBE, RAČUN, POSLOVNI PARTNERI, DOBAVLJAČ, DJELATNIK.

Veze: INFORMIRA između DOBAVLJAČ i ARTIKL, PREDAJE između ARTIKL i POPIS ROBE, SADRŽI između POPIS ROBE i RAČUN, POSTAVLJA između POSLOVNI PARTNER i RAČUN, ISPUNJAVA između DJELATNIK i RAČUN.

Atributi: ŠIFRA ARTIKLA, ŠIFRA ROBE, ŠIFRA RAČUNA, OIB DOBAVLJAČA, OIB DJELATNIKA, OIB PARTNERA, NAZIV, MJESTO, PBR, ADRESA, BROJ TELEFONA, IME, PREZIME, NAZIV ARTIKLA, NAZIV DOBAVLJAČA, ULOGA, PLAĆA, AKTIVAN, IBAN, JEDINICA MJERE, KOLIČINA, CIJENA, U PRODAJI, DATUM.

Dodjela atributa entitetima.

Entitet DJELATNIK ima sljedeće attribute: OIB DJELATNIKA, IME, PREZIME, ULOGA, PLAĆA, BROJ TELEFONA, AKTIVAN

Entitet POSLOVNI PARTNER ima sljedeće attribute: OIB PARTNERA, IME, PREZIME, PBR, MJESTO, ADRESA, BROJ TELEFONA, IBAN, AKTIVAN.

Entitet DOBAVLJAČ ima sljedeće attribute: OIB DOBAVLJAČA, NAZIV DOBAVLJAČA, PBR, MJESTO, ADRESA, BROJ TELEFONA, IBAN, AKTIVAN.

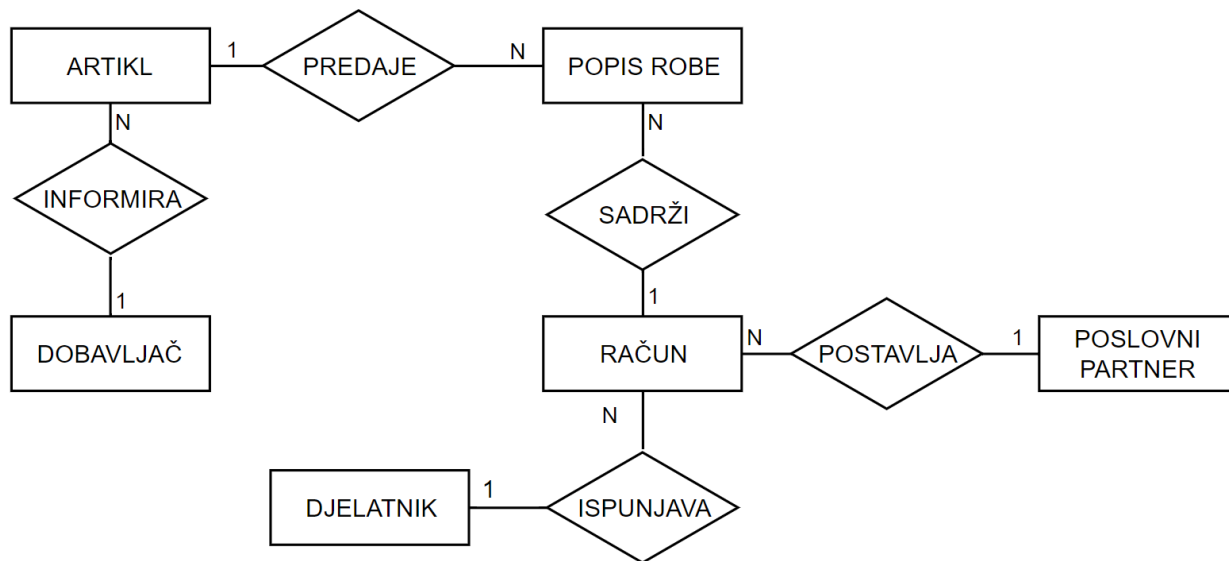
Entitet ARTIKL ima sljedeće attribute: ŠIFRA ARTIKLA, NAZIV ARTIKLA, JEDINICA MJERE, KOLIČINA, CIJENA, U PRODAJI.

Entitet POPIS ROBE ima sljedeće attribute: ŠIFRA ROBE, KOLIČINA.

Entitet RAČUN ima sljedeće attribute: ŠIFRA RAČUNA, DATUM.

2. KONCEPTUALNO OBLIKOVANJE

Prema utvrđivanju i analizi zahtjeva korisnika izradimo E/R dijagram koji je prikazan na slici 1. Na slici 1 su prikazani entiteti i veze između njih, te odgovarajuće funkcionalnosti veza.



Slika 1. E/R dijagram za bazu podataka poljoprivredne ljekarne

U tablici 1 su navedeni pripadni atributi odgovarajućih entiteta i veza. Podvučeni atributi predstavljaju kandidate za ključ.

Tablica 1. Pripadni atributi entiteta za dijagram sa slike 1.

Entitet	Atribut
DOBAVLJAČ	<u>OIB DOBAVLJAČA</u> , NAZIV DOBAVLJAČA, PBR, MJESTO, ADRESA, BROJ TELEFONA, IBAN, AKTIVAN
ARTIKL	<u>ŠIFRA ARTIKLA</u> , NAZIV ARTIKLA, JEDINICA MJERE, KOLIČINA, CIJENA, UPRODAJI
POPIS ROBE	<u>ŠIFRA ROBE</u> , KOLIČINA
RAČUN	<u>ŠIFRA RAČUNA</u> , DATUM
POSLOVNI PARTNER	<u>OIB PARTNERA</u> , IME, PREZIME, PBR, MJESTO, ADRESA, BROJ TELEFONA, IBAN, AKTIVAN
DJELATNIK	<u>OIB DJELATNIKA</u> , IME, PREZIME, ULOGA, PLAĆA, BROJ TELEFONA, AKTIVAN

3. LOGIČKO OBLIKOVANJE

Provođenjem E/R modela (slika 1 i tablica 1) u relacijski model dobivena je sljedeća relacijska shema baze podataka poljoprivredne ljekarne kao skup relacija prikazanih na slici 2.

DOBAVLJAČ(OIB DOBAVLJAČA, NAZIV DOBAVLJAČA, PBR, MJESTO, ADRESA, BROJ TELEFONA, IBAN, AKTIVAN)

ARTIKL(ŠIFRA ARTIKLA, NAZIV ARTIKLA, JEDINICA MJERE, KOLIČINA, CIJENA, OIB DOBAVLJAČA, U PRODAJI)

POPIS ROBE(ŠIFRA ROBE, ŠIFRA RAČUNA, ŠIFRA ARTIKLA, KOLIČINA)
RAČUN(ŠIFRA RAČUNA, OIB DJELATNIKA, OIB PARTNERA, DATUM)

POSLOVNI PARTNER(OIB PARTNERA, IME, PREZIME, PBR, MJESTO, ADRESA, BROJ TELEFONA, IBAN, AKTIVAN)

DJELATNIK(OIB DJELATNIKA, IME, PREZIME, ULOGA, PLAĆA, BROJ TELEFONA, AKTIVAN)

Slika 2. Relacijska shema baze podataka poljoprivredne ljekarne

Rječnik podataka na bazu podataka poljoprivredne ljekarne, odnosno na relacijsku shemu sa slike 2, je dan u tablici 2.

Ime atributa	Tip	Opis
OIB DOBAVLJAČA	Niz od točno 11 znakova	OIB dobavljača
NAZIV DOBAVLJAČA	Niz znakova	Naziv dobavljača
PBR	Niz od točno 5 znakova	Pošanski broj mjesta
MJESTO	Niz znakova	Mjesto osobe
ADRESA	Niz znakova	Adresa osobe
BROJ TELEFONA	Niz od maksimalno 10 znakova	Broj telefona osobe
IBAN	Niz od točno 21 znaka	IBAN račun
AKTIVAN	Niz od točno 2 znaka	Da li osoba još uvijek posluje ili radi
ŠIFRA ARTIKLA	Niz od maksimalno 5 znakova	Šifra artikla
NAZIV ARTIKLA	Niz znakova	Naziv artikla
JEDINICA MJERE	Niz od točno tri znaka	Jedinica mjere artikla
KOLIČINA	Cijeli broj	Količina artikla
CIJENA	Decimalni broj	Cijena artikla

Tablica 2. Nastavlja se

Ime atributa	Tip	Opis
U PRODAJI	Niz od točno 2 znaka	Da li je artikl trenutno u prodaji
ŠIFRA ROBE	Niz od maksimalno 5 znakova	Šifra popisa robe
ŠIFRA RAČUNA	Niz od maksimalno 5 znakova	Šifra računa
DATUM	Datum	Datum ispostavljanja računa
OIB PARTNERA	Niz od točno 11 znakova	OIB poslovnog partnera
IME	Niz znakova	Ime osobe
PREZIME	Niz znakova	Prezime osobe
OIB DJELATNIKA	Niz od točno 11 znakova	OIB djelatnika

Tablica 2. Rječnik podataka za bazu podataka poljoprivredne ljekarne

4. FIZIČKO OBLIKOVANJE

Fizička shema (SQL naredbe za implementaciju baze podataka) baze podataka poljoprivredne ljekarne je prikazana na slici 3, te je dobivena na osnovu relacijskih shema baze podataka poljoprivredne ljekarne.

```
CREATE TABLE djelatnik(  
    OIB CHAR(11),  
    ime NVARCHAR(30) NOT NULL,  
    prezime NVARCHAR(30) NOT NULL,  
    uloga NCHAR(17) DEFAULT 'RADNIK',  
    placa DECIMAL(7, 2) NOT NULL,  
    broj_telefona CHAR(10),  
    aktivan CHAR(2) DEFAULT 'DA',  
    CONSTRAINT pk_djelatnik PRIMARY KEY (OIB),  
    CONSTRAINT chk_djelatnik CHECK (uloga IN ('ŠEF', 'RADNIK', 'PRIVREMENI  
    RADNIK')),  
    CONSTRAINT chk_status CHECK (aktivan IN ('DA', 'NE'))  
);  
  
CREATE TABLE poslovni_partner(  
    OIB CHAR(11),  
    ime NVARCHAR(30) NOT NULL,  
    prezime NVARCHAR(30) NOT NULL,  
    pbr CHAR(5),  
    mjesto NVARCHAR(30),  
    adresa NVARCHAR(50),  
    broj_telefona CHAR(10),  
    IBAN CHAR(21),  
    aktivan CHAR(2) DEFAULT 'DA',  
    CONSTRAINT pk_poslPartner PRIMARY KEY (OIB),  
    CONSTRAINT chk_poslPartner CHECK (aktivan IN ('DA', 'NE'))  
);  
  
CREATE TABLE dobavljac(  
    OIB CHAR(11),  
    naziv NVARCHAR(100) NOT NULL,  
    pbr CHAR(5),  
    mjesto NVARCHAR(30),  
    adresa NVARCHAR(50),  
    broj_telefona CHAR(10),  
    IBAN CHAR(21),  
    aktivan CHAR(2) DEFAULT 'DA',  
    CONSTRAINT pk_dobavljac PRIMARY KEY (OIB),  
    CONSTRAINT chk_dobavljac CHECK (aktivan IN ('DA', 'NE'))  
);
```

Slika 3. *Nastavlja se*


```

CREATE TABLE artikl(
    sifra CHAR(5),
    naziv NVARCHAR(50),
    jedinicaMjere CHAR(3),
    kolicina INT,
    cijena DECIMAL(7, 2) NOT NULL,
    dobavljac CHAR(11),
    uprodaji CHAR(2) DEFAULT 'DA',
    CONSTRAINT chk_jedinicaMjere CHECK (jedinicaMjere IN ('kom', 'kg',
'm', 'l')),
    CONSTRAINT chk_uprodaji CHECK (uprodaji IN ('DA', 'NE')),
    CONSTRAINT pk_artikl PRIMARY KEY (sifra),
    CONSTRAINT fk_dobavljac FOREIGN KEY (dobavljac) REFERENCES
dobavljac(OIB)
);

CREATE TABLE racunR2(
    sifra CHAR(5),
    djelatnik CHAR(11),
    poslovni_partner CHAR(11),
    datum_izdavanja DATETIME,
    CONSTRAINT pk_racunR2 PRIMARY KEY (sifra),
    CONSTRAINT fk_djelatnik FOREIGN KEY (djelatnik) REFERENCES
djelatnik(OIB),
    CONSTRAINT fk_poslovniPartner FOREIGN KEY (poslovni_partner)
REFERENCES poslovni_partner(OIB)
);

CREATE TABLE popis_robe(
    sifra INT IDENTITY(1,1),
    racun CHAR(5),
    artikl CHAR(5),
    kolicina INT,
    CONSTRAINT chk_kolicina CHECK (kolicina > 0),
    CONSTRAINT pk_sifra PRIMARY KEY (sifra),
    CONSTRAINT fk_racun FOREIGN KEY (racun) REFERENCES racunR2(sifra)
);

```

Slika 3. *Fizička shema baze podataka poljoprivredne ljekarne*

U nastavku slijede funkcije, pogledi, okidači i procedure koje su također sastavni dio fizičke sheme baze podataka.

Procedura za ispis traženog računa dana je na slici 4. Procedura kao vrijednost prima šifru računa za koji se žele znati detalji.

```
CREATE PROC nadi_racunR2 @sifra_racuna CHAR(5)
AS
BEGIN
SELECT DISTINCT racunR2.sifra AS 'Sifra racuna',
    djelatnik.ime + ' ' + djelatnik.prezime AS 'Izdavatelj računa',
    poslovni_partner.ime + ' ' + poslovni_partner.prezime AS 'Izdano',
    racunR2.datum_izdavanja AS 'Datuma'
    FROM djelatnik, racunR2, poslovni_partner
    WHERE racunR2.djelatnik = djelatnik.OIB
    AND racunR2.poslovni_partner = poslovni_partner.OIB
    AND racunR2.sifra = @sifra_racuna;

SELECT DISTINCT artikl.naziv AS 'Naziv artikla',
    artikl.jedinicaMjere AS 'Jedinica mjere',
    popis_robe.kolicina AS 'Kolicina',
    artikl.cijena AS 'Cijena kn',
    popis_robe.kolicina * artikl.cijena AS 'Iznos kn'
    FROM racunR2, artikl, popis_robe
    WHERE artikl.sifra = popis_robe.artikl
    AND popis_robe.racun = racunR2.sifra
    AND racunR2.sifra = @sifra_racuna;

SELECT DISTINCT    dbo.ukupan_iznos_racuna(@sifra_racuna) AS 'Ukupan iznos
u kn',
    dbo.ukupan_iznos_racuna_PDV(@sifra_racuna) -
    dbo.ukupan_iznos_racuna(@sifra_racuna) AS 'PDV (25%) u kn',
    dbo.ukupan_iznos_racuna_PDV(@sifra_racuna) AS 'Ukupan iznos u kn s
PDV(25%)'
    FROM racunR2, artikl, popis_robe;
END;
```

Slika 4. *Procedura za ispis traženog računa*

Navedena procedura ispisuje tri tablice. Prva predstavlja informacije o računu, druga popis robe, te treća iznos računa. Ova procedura sadrži i dvije funkcije koje nam izračunaju ukupnu cijenu i ukupnu cijenu s PDV-om.

Na slici 5 je prikazana funkcija koja izračuna ukupnu cijenu računa bez PDV-a, a slika 6 prikazuje ukupnu cijenu računa s PDV-om.

```

CREATE FUNCTION ukupan_iznos_racuna (@racun_sifra CHAR(5))
RETURNS FLOAT
AS
BEGIN
    DECLARE @ukupni_iznos FLOAT
    SELECT @ukupni_iznos = SUM(artikl.cijena * popis_robe.kolicina)
    FROM artikl, popis_robe, racunR2
    WHERE artikl.sifra = popis_robe.artikl
    AND popis_robe.racun = @racun_sifra
    AND @racun_sifra = racunR2.sifra
    GROUP BY popis_robe.racun;
    RETURN @ukupni_iznos
END;

```

Slika 5. Funkcija za izračun ukupne cijene računa bez PDV-a

```

CREATE FUNCTION ukupan_iznos_racuna_PDV (@racun_sifra CHAR(5))
RETURNS FLOAT
AS
BEGIN
    DECLARE @ukupni_iznos FLOAT
    DECLARE @pdv DECIMAL (3, 2)
    SET @pdv = '1.25'
    SELECT @ukupni_iznos = SUM(artikl.cijena * popis_robe.kolicina) * @pdv
    FROM artikl, popis_robe, racunR2
    WHERE artikl.sifra = popis_robe.artikl
    AND popis_robe.racun = @racun_sifra
    AND @racun_sifra = racunR2.sifra;
    RETURN @ukupni_iznos
END;

```

Slika 6. Funkcija za izračun ukupne cijene računa s PDV-om

Na slikama 7, 8, 9, 10, 11 se nalaze pogledi vezani za artikle. Pogled na slici 11 vraća artikle čija je količina manja od 6.

```

CREATE VIEW stanje_skladista
AS
SELECT artikl.sifra AS 'Sifra',
       artikl.naziv AS 'Naziv',
       artikl.jedinicaMjere AS 'Jedinica mjere',
       artikl.kolicina AS 'Količina',
       artikl.cijena AS 'Cijena u kn',
       artikl.cijena * artikl.kolicina AS 'Ukupni iznos u kn',
       artikl.uprodaji AS 'U prodaji',
       dobavljac.naziv AS 'Dobavljač'
FROM artikl, dobavljac
WHERE artikl.dobavljac = dobavljac.OIB;

```

Slika 7. Pogled na sve artikle pod nazivom stanje_skladišta

```

CREATE VIEW artikli_uprodaji
AS
SELECT artikl.sifra AS 'Sifra',
       artikl.naziv AS 'Naziv',
       artikl.jedinicaMjere AS 'Jedinica mjere',
       artikl.kolicina AS 'Količina',
       artikl.cijena AS 'Cijena u kn',
       artikl.cijena * artikl.kolicina AS 'Ukupni iznos u kn',
       artikl.uprodaji AS 'U prodaji',
       dobavljac.naziv AS 'Dobavljač'
FROM artikl, dobavljac
WHERE artikl.dobavljac = dobavljac.OIB
AND artikl.uprodaji = 'DA';

```

Slika 8. *Pogled na sve artikle koji su trenutačno u prodaji*

```

CREATE VIEW artikli_nisu_uprodaji
AS
SELECT artikl.sifra AS 'Sifra',
       artikl.naziv AS 'Naziv',
       artikl.jedinicaMjere AS 'Jedinica mjere',
       artikl.kolicina AS 'Količina',
       artikl.cijena AS 'Cijena u kn',
       artikl.cijena * artikl.kolicina AS 'Ukupni iznos u kn',
       artikl.uprodaji AS 'U prodaji',
       dobavljac.naziv AS 'Dobavljač'
FROM artikl, dobavljac
WHERE artikl.dobavljac = dobavljac.OIB
AND artikl.uprodaji = 'NE';

```

Slika 9. *Pogled na sve artikle koji trenutačno nisu u prodaji*

```

CREATE VIEW artikli_kojih_nema
AS
SELECT artikl.sifra AS 'Sifra',
       artikl.naziv AS 'Naziv',
       artikl.jedinicaMjere AS 'Jedinica mjere',
       artikl.kolicina AS 'Količina',
       artikl.cijena AS 'Cijena u kn',
       artikl.cijena * artikl.kolicina AS 'Ukupni iznos u kn',
       artikl.uprodaji AS 'U prodaji',
       dobavljac.naziv AS 'Dobavljač',
       dobavljac.broj_telefona AS 'Broj telefona'
FROM artikl, dobavljac
WHERE artikl.dobavljac = dobavljac.OIB
AND artikl.kolicina = 0;

```

Slika 10. *Pogled na artikle koji trenutačno nema*

```

CREATE VIEW artikli_pri_kraju
AS
SELECT artikl.sifra AS 'Sifra',
       artikl.naziv AS 'Naziv',
       artikl.jedinicaMjere AS 'Jedinica mjere',
       artikl.kolicina AS 'Količina',
       artikl.cijena AS 'Cijena u kn',
       artikl.cijena * artikl.kolicina AS 'Ukupni iznos u kn',
       artikl.uprodaji AS 'U prodaji',
       dobavljac.naziv AS 'Dobavljač',
       dobavljac.broj_telefona AS 'Broj telefona'
FROM artikl, dobavljac
WHERE artikl.dobavljac = dobavljac.OIB
AND artikl.kolicina < 6;

```

Slika 11. *Pogled na artikle koji su pri kraju*

Na slikama 12, 13, 14 se nalaze pogledi vezani za dobavljače.

```

CREATE VIEW svi_dobavljac
AS
SELECT naziv AS 'Naziv',
       OIB AS 'OIB',
       pbr AS 'PBR',
       mjesto AS 'Mjesto',
       adresa AS 'Adresa',
       broj_telefona AS 'Broj telefona',
       IBAN AS 'IBAN',
       aktivan AS 'Aktivan'
FROM dobavljac;

```

Slika 12. *Pogled na sve dobavljače*

```

CREATE VIEW aktivni_dobavljac
AS
SELECT naziv AS 'Naziv',
       OIB AS 'OIB',
       pbr AS 'PBR',
       mjesto AS 'Mjesto',
       adresa AS 'Adresa',
       broj_telefona AS 'Broj telefona',
       IBAN AS 'IBAN'
FROM dobavljac
WHERE aktivan = 'DA';

```

Slika 13. *Pogled na aktivne dobavljače*

```

CREATE VIEW neaktivni_dobavljac
AS
SELECT naziv AS 'Naziv',
       OIB AS 'OIB',
       pbr AS 'PBR',
       mjesto AS 'Mjesto',
       adresa AS 'Adresa',
       broj_telefona AS 'Broj telefona',
       IBAN AS 'IBAN'
FROM dobavljac
WHERE aktivan = 'NE';

```

Slika 14. *Pogled na neaktivne dobavljače*

Na slikama 15, 16, 17, 18, 19, 20 nalaze se pogledi za djelatnike. Pogledi na slikama 18, 19 i 20 ispisuju djelatnike ovisno o njihovoj ulozi.

```

CREATE VIEW svi_djelatnici
AS
SELECT ime + ' ' + prezime AS 'Djelatnik',
       OIB AS 'OIB',
       UPPER(uloga) AS 'Uloga',
       placa AS 'Plaća',
       broj_telefona AS 'Broj telefona',
       aktivan AS 'Aktivan'
FROM djelatnik;

```

Slika 15. *Pogled na sve djelatnike*

```

CREATE VIEW aktivni_djelatnici
AS
SELECT ime + ' ' + prezime AS 'Djelatnik',
       OIB AS 'OIB',
       UPPER(uloga) AS 'Uloga',
       placa AS 'Plaća',
       broj_telefona AS 'Broj telefona'
FROM djelatnik
WHERE aktivan = 'DA';

```

Slika 16. *Pogled na aktivne djelatnike*

```

CREATE VIEW neaktivni_djelatnici
AS
SELECT ime + ' ' + prezime AS 'Djelatnik',
       OIB AS 'OIB',
       UPPER(uloga) AS 'Uloga',
       placa AS 'Plaća',
       broj_telefona AS 'Broj telefona'
FROM djelatnik
WHERE aktivan = 'NE';

```

Slika 17. *Pogled na neaktivne djelatnike*

```

CREATE VIEW sef
AS
SELECT ime + ' ' + prezime AS 'Djelatnik',
       OIB AS 'OIB',
       UPPER(uloga) AS 'Uloga',
       placa AS 'Plaća',
       broj_telefona AS 'Broj telefona',
       aktivan AS 'Aktivan'
FROM djelatnik
WHERE uloga = 'ŠEF';

```

Slika 18. *Pogled na ulogu šef*

```

CREATE VIEW radnici
AS
SELECT ime + ' ' + prezime AS 'Djelatnik',
       OIB AS 'OIB',
       UPPER(uloga) AS 'Uloga',
       placa AS 'Plaća',
       broj_telefona AS 'Broj telefona',
       aktivan AS 'Aktivan'
FROM djelatnik
WHERE uloga = 'RADNIK';

```

Slika 19. *Pogled na ulogu radnik*

```

CREATE VIEW privremeni_radnici
AS
SELECT ime + ' ' + prezime AS 'Djelatnik',
       OIB AS 'OIB',
       UPPER(uloga) AS 'Uloga',
       placa AS 'Plaća',
       broj_telefona AS 'Broj telefona',
       aktivan AS 'Aktivan'
FROM djelatnik
WHERE uloga = 'PRIVREMENI RADNIK';

```

Slika 20. *Pogled na ulogu privremeni radnik*

Na slikama 21, 22, 23 nalaze se pogledi za poslovne partnere.

```
CREATE VIEW svi_poslovni_partneri
AS
SELECT ime + ' ' + prezime AS 'Ime i prezime',
       OIB AS 'OIB',
       pbr AS 'PBR',
       mjesto AS 'Mjesto',
       adresa AS 'Adresa',
       broj_telefona AS 'Broj telefona',
       IBAN AS 'IBAN',
       aktivan AS 'Aktivan'
FROM poslovni_partner;
```

Slika 21. *Pogled na sve poslovne partnere*

```
CREATE VIEW aktivni_poslovni_partneri
AS
SELECT ime + ' ' + prezime AS 'Ime i prezime',
       OIB AS 'OIB',
       pbr AS 'PBR',
       mjesto AS 'Mjesto',
       adresa AS 'Adresa',
       broj_telefona AS 'Broj telefona',
       IBAN AS 'IBAN'
FROM poslovni_partner
WHERE aktivan = 'DA';
```

Slika 22. *Pogled na aktivne poslovne partnere*

```
CREATE VIEW neaktivni_poslovni_partneri
AS
SELECT ime + ' ' + prezime AS 'Ime i prezime',
       OIB AS 'OIB',
       pbr AS 'PBR',
       mjesto AS 'Mjesto',
       adresa AS 'Adresa',
       broj_telefona AS 'Broj telefona',
       IBAN AS 'IBAN'
FROM poslovni_partner
WHERE aktivan = 'NE';
```

Slika 23. *Pogled na neaktivne poslovne partnere*

Slika 24 prikazuje pogled na sve račune, tako da daje informacije o šifri računa, iznosu računa bez PDV-a, ime djelatnika koji izdaje račun, ime poslovnog partnera kojemu je izdan račun, datum izdavanja .

```
CREATE VIEW racuniR2
AS
SELECT DISTINCT racunR2.sifra AS 'Sifra racuna',
    dbo.ukupan_iznos_racuna(racunR2.sifra) AS 'Ukupan iznos u kn',
    djelatnik.ime + ' ' + djelatnik.prezime AS 'Izdavatelj računa',
    poslovni_partner.ime + ' ' + poslovni_partner.prezime AS 'Izdano',
    racunR2.datum_izdavanja AS 'Datuma'
FROM djelatnik, racunR2, poslovni_partner, artikl, popis_robe
WHERE artikl.sifra = popis_robe.artikl
AND popis_robe.racun = racunR2.sifra
AND racunR2.djelatnik = djelatnik.OIB
AND racunR2.poslovni_partner = poslovni_partner.OIB;
```

Slika 24. *Pogled na sve račune*

Na slikama 25, 26 se nalaze procedure za pretraživanje artikala pomoću šifre artikla, te naziva artikla.

```
CREATE PROC nadi_artikl_po_sifri @sifra_artikla CHAR(5)
AS
BEGIN
SELECT artikl.sifra AS 'Sifra',
    artikl.naziv AS 'Naziv',
    artikl.jedinicaMjere AS 'Jedinica mjere',
    artikl.kolicina AS 'Količina',
    artikl.cijena AS 'Cijena u kn',
    artikl.cijena * artikl.kolicina AS 'Ukupni iznos u kn',
    artikl.uprodaji AS 'U prodaji',
    dobavljac.naziv AS 'Dobavljač',
    dobavljac.broj_telefona AS 'Broj telefona'
FROM artikl, dobavljac
WHERE artikl.dobavljac = dobavljac.OIB
AND @sifra_artikla = artikl.sifra;
END;
```

Slika 25. *Procedura za ispisivanje artikla pomoću šifre artikla*

```

CREATE PROC nadi_artikl_po_nazivu @naziv_artikla NVARCHAR(50)
AS
BEGIN
SELECT artikl.sifra AS 'Sifra',
      artikl.naziv AS 'Naziv',
      artikl.jedinicaMjere AS 'Jedinica mjere',
      artikl.kolicina AS 'Količina',
      artikl.cijena AS 'Cijena u kn',
      artikl.cijena * artikl.kolicina AS 'Ukupni iznos u kn',
      artikl.uprodaji AS 'U prodaji',
      dobavljac.naziv AS 'Dobavljač',
      dobavljac.broj_telefona AS 'Broj telefona'
FROM artikl, dobavljac
WHERE artikl.dobavljac = dobavljac.OIB
AND @naziv_artikla = artikl.naziv;
END;

```

Slika 26. *Procedura za ispisivanje artikla pomoću naziva*

Na slici 27 se nalazi procedura za povećanje količine artikla, tako da upišemo šifru željenog artikla i količinu za koju ćemo dodati artiklu.

```

CREATE PROC povecaj_kolicinu_artikla @sifra_artikla CHAR(5), @kolicina INT
AS
BEGIN
      UPDATE artikl SET kolicina = kolicina + @kolicina
      WHERE sifra = @sifra_artikla;
END;

```

Slika 27. *Procedura za povećanje količine određenog artikla*

Na slikama 28, 29, 30 se nalaze procedure za pretraživanje djelatnika, poslovnog partnera i dobavljača, pomoću OIB-a osobe.

```

CREATE PROC nadi_djelatnika_po_OIB @OIB_djelatnika CHAR(11)
AS
BEGIN
SELECT ime + ' ' + prezime AS 'Djelatnik',
       OIB AS 'OIB',
       UPPER(uloga) AS 'Uloga',
       placa AS 'Plaća',
       broj_telefona AS 'Broj telefona',
       aktivan AS 'Aktivan'
FROM djelatnik
WHERE @OIB_djelatnika = OIB;
END;

```

Slika 28. *Procedura za ispisivanje djelatnika pomoću OIB-a*

```

CREATE PROC nadi_poslPartnera_po_OIB @OIB_poslPartnera CHAR(11)
AS
BEGIN
SELECT ime + ' ' + prezime AS 'Ime i prezime',
       OIB AS 'OIB',
       pbr AS 'PBR',
       mjesto AS 'Mjesto',
       adresa AS 'Adresa',
       broj_telefona AS 'Broj telefona',
       IBAN AS 'IBAN',
       aktivan AS 'Aktivan'
FROM poslovni_partner
WHERE @OIB_poslPartnera = OIB;
END;

```

Slika 29. *Procedura za ispisivanje poslovnog partnera pomoću OIB-a*

```

CREATE PROC nadi_dobavljacka_po_OIB @OIB_dobavljacka CHAR(11)
AS
BEGIN
SELECT naziv AS 'Naziv',
       OIB AS 'OIB',
       pbr AS 'PBR',
       mjesto AS 'Mjesto',
       adresa AS 'Adresa',
       broj_telefona AS 'Broj telefona',
       IBAN AS 'IBAN',
       aktivan AS 'Aktivan'
FROM dobavljac
WHERE OIB = @OIB_dobavljacka;
END;

```

Slika 30. *Procedura za ispisivanje dobavljača pomoću OIB-a*

Na slikama 31, 32, 33, 34, 35, 36 nalaze se procedure za uklanjanje podataka iz tablica.

```
CREATE PROC obrisi_racunR2 @sifra_racuna CHAR(5)
AS
BEGIN
    DELETE popis_robe WHERE racun = @sifra_racuna;
    DELETE racunR2 WHERE sifra = @sifra_racuna;
END;
```

Slika 31. *Procedura za brisanje računa ovisno o unesenoj šifri računa*

```
CREATE PROC obrisi_artikl @sifra_artikla CHAR(5)
AS
BEGIN
    DELETE popis_robe WHERE artikl IN (SELECT sifra FROM artikl WHERE
sifra = @sifra_artikla);
    DELETE artikl WHERE sifra = @sifra_artikla;
END;
```

Slika 32. *Procedura za brisanje računa ovisno o unesenoj šifri artikla*

```
CREATE PROC obrisi_djelatnika @OIB_djelatnika CHAR(11)
AS
BEGIN
    DELETE popis_robe WHERE racun IN (SELECT sifra FROM racunR2 WHERE
djelatnik = @OIB_djelatnika);
    DELETE racunR2 WHERE djelatnik = @OIB_djelatnika;
    DELETE djelatnik WHERE OIB = @OIB_djelatnika;
END;
```

Slika 33. *Procedura za brisanje djelatnika ovisno o unesenom OIB-u djelatnika*

```
CREATE PROC obrisi_poslPartnera @OIB_poslPartner CHAR(11)
AS
BEGIN
    DELETE popis_robe WHERE racun IN (SELECT sifra FROM racunR2 WHERE
poslovni_partner = @OIB_poslPartner);
    DELETE racunR2 WHERE poslovni_partner = @OIB_poslPartner;
    DELETE poslovni_partner WHERE OIB = @OIB_poslPartner;
END;
```

Slika 34. *Procedura za brisanje poslovnog partnera ovisno o OIB-u poslovnog partnera*

```

CREATE PROC obrisi_dobavljacka @OIB_dobavljacka CHAR(11)
AS
BEGIN
    DELETE popis_robe WHERE artikl IN (SELECT sifra FROM artikl WHERE
dobavljac = @OIB_dobavljacka);
    DELETE artikl WHERE dobavljac = @OIB_dobavljacka;
    DELETE dobavljac WHERE OIB = @OIB_dobavljacka;
END;

```

Slika 35. *Procedura za brisanje dobavljača ovisno o OIB-u dobavljača*

Umjesto navedenih procedura za brisanje bolje je koristiti UPDATE naredbu kako bi očuvali integritet ostalih tablica. Primjer korištenja UPDATE naredbe je dan na slici 36. U primjeru samo zamijenimo vrijednosti ime_tablice, OIB_željene_osobe, sifra_željenog_artikla s željenim vrijednostima.

```

UPDATE ime_tablice SET stanje = 'NE' WHERE OIB = OIB_osobe_željene_osobe;
UPDATE artikl SET uprodaji = 'NE' WHERE sifra = sifra_željenog_artikla;

```

Slika 36. *Primjer korištenja UPDATE naredbe umjesto korištenja procedura za uklanjanje*

Okidači su prikazani na slikama 37 i 38.

Okidač na slici 37 nam služi umjesto INSERT naredbe na tablici popis_robe. Radi na način da provjerava unešenu količinu s trenutnom količinom artikla. Ako unešena količina artikala nadmašuje trenutačnu, onda procedura ispisuje „Navedena količina artikla nije dostupna”. U suprotnom oduzimamo navedenu količinu artikala od trenutačne. Zatim provjeravamo da li se navedeni artikl za zadani račun već nalazi u popisu_robe, ako da onda samo dodajemo količinu na njega, u suprotnom vršimo naredbu INSERT u tablicu popis_robe.

```

CREATE TRIGGER unos_popisa_robe
ON popis_robe
INSTEAD OF INSERT
AS
    DECLARE @kolicina_skladiste INT
    DECLARE @sifra_artikla CHAR(5)
    DECLARE @sifra_racuna CHAR(5)
    DECLARE @kolicina INT

    SELECT @kolicina_skladiste=artikl.kolicina, @sifra_artikla = i.artikl
    FROM inserted i, artikl
    WHERE artikl.sifra = i.artikl;
    SELECT @sifra_racuna = i.racun FROM inserted i;
    SELECT @kolicina = i.kolicina FROM inserted i;

    IF @kolicina <= @kolicina_skladiste AND @kolicina > 0
    BEGIN
        UPDATE artikl SET kolicina = kolicina - @kolicina WHERE
        artikl.sifra = @sifra_artikla;
        IF @sifra_racuna = (SELECT popis_robe.racun FROM popis_robe
        WHERE popis_robe.artikl = @sifra_artikla AND @sifra_racuna =
        popis_robe.racun)
            UPDATE popis_robe SET kolicina = kolicina + @kolicina
            WHERE popis_robe.artikl = @sifra_artikla AND popis_robe.racun =
            @sifra_racuna;

        ELSE
            INSERT INTO popis_robe VALUES (@sifra_racuna, @sifra_artikla,
            @kolicina);

        END

    ELSE
    BEGIN
        PRINT 'Navedena količina artikla nije dostupna';
    END

```

Slika 37. Okidač na popis_robe umjesto INSERT naredbe

Okidač na slici 38 služi umjesto INSERT naredbe na tablicu artikl. Radi na način da provjerava da li navedeni artikl postoji u tablici artikla, ako postoji onda zovemo UPDATE naredbu i povećamo količinu artikla. U suprotnom okidač vrši INSERT unešenih vrijednosti u tablicu artikl.

```
CREATE TRIGGER unos_robe_u_skladiste
ON artikl
INSTEAD OF INSERT
AS
    DECLARE @sifra CHAR(5)
    DECLARE @naziv NVARCHAR(50)
    DECLARE @jed_mjere CHAR(3)
    DECLARE @kolicina INT
    DECLARE @cijena DECIMAL(7,2)
    DECLARE @dobavljac CHAR(11)

    SELECT @sifra = i.sifra FROM inserted i;
    SELECT @naziv = i.naziv FROM inserted i;
    SELECT @jed_mjere = i.jedinicaMjere FROM inserted i;
    SELECT @kolicina = i.kolicina FROM inserted i;
    SELECT @cijena = i.cijena FROM inserted i;
    SELECT @dobavljac = i.dobavljac FROM inserted i;

    IF @naziv IN (SELECT artikl.naziv FROM artikl)
    BEGIN
        UPDATE artikl SET kolicina = kolicina + @kolicina
        WHERE @naziv IN (SELECT artikl.naziv FROM artikl);
    END
    ELSE
    BEGIN
        INSERT INTO artikl VALUES (@sifra, @naziv, @jed_mjere,
        @kolicina, @cijena, @dobavljac, 'DA');
    END;
END;
```

Slika 38. Okidač na artikl umjesto INSERT naredbe

5. UNOS, UKLANJANJE I PRETRAŽIVANJE PODATAKA

Podaci koje unosimo u bazu podataka se nalaze na slikama 39, 40, 41, 42, 43, 44. Njih unosimo redom pomoću SQL naredbe prikazane na slici 45.

```
981728169XX,Ivo,Ivić,Šef,6000,0XXXXXXXX,DA
```

Slika 39. Podaci o djelatniku

```
715716313XX,Horvat,Horvatić,32100,Vinkovci,Obrtnička  
ul.,0XXXXXXXX,HRXXXXXXXXXXXXXXXXXXXX,
```

```
378321826XX,Iva,Ivanković,32272,Cerna,Ul. Kralja  
Tomislava,0XXXXXXXX,HRXXXXXXXXXXXXXXXXXXXX,
```

```
304769545XX,Pero,Perić,32272,Cerna,Strossmayerova  
ul.,0XXXXXXXX,HRXXXXXXXXXXXXXXXXXXXX,
```

```
262894341XX,Tomo,Tomić,32270,Županja,Ul. Alozija  
Stepinca,0XXXXXXXX,HRXXXXXXXXXXXXXXXXXXXX,
```

```
716129438XX,Marija,Marić,32273,Gradište,Ul. Vladimira  
Nazora,0XXXXXXXX,HRXXXXXXXXXXXXXXXXXXXX,
```

```
632731006XX,Antonio,Antić,32271,Andrijaševci,Ul. Marka  
Marulića,0XXXXXXXX,HRXXXXXXXXXXXXXXXXXXXX,
```

Slika 40. Podaci o poslovnim partnerima

```
03834418154,Zito  
d.o.o.,31000,Osijek,Đakovština,031228921,HRXXXXXXXXXXXXXXXXXXXX,
```

```
12875096243,PPK VALPOVO d.o.o.,31550,Valpovo,A.B. Šimića  
27,031656500,HRXXXXXXXXXXXXXXXXXXXX,
```

```
35063921066,Amarilis d.o.o.,32100,Vinkovci,Ul. kralja Zvonimira  
99,032308462,HRXXXXXXXXXXXXXXXXXXXX,
```

```
00721719381,Vinkoprom d.o.o.,32100,Vinkovci,H.V.Hrvatinića  
108,032348500,HRXXXXXXXXXXXXXXXXXXXX,
```

Slika 41. Podaci o dobavljačima


```

1,KAN25,kg,1200,1.92,03834418154,
2,Harmony,kom, 10,500,12875096243,
3,Axial 50 EC 1/1,kom,10,499,35063921066,
4,Hranilica za perad,kom,5,30,00721719381,
5,Folija prozirna 4000*0.15,m,53,10,00721719381,
6,Karate zeon 1/1,kom,11,449,03834418154,
7,Valpomin za nesilice,kom,25,13,12875096243,
8,Valpovo S-35 10/1 DKS za svinje,kom,16,45,12875096243,
9,WD-40 spray 200 ml,kom,3,26,00721719381,
10,Nurelle D 1/1,kom,24,284,03834418154,

```

Slika 42. Podaci o artiklima

```

1,981728169XX,715716313XX,6/13/2018
2,981728169XX,262894341XX,6/14/2018
3,981728169XX,378321826XX,6/18/2018
4,981728169XX,262894341XX,6/18/2018
5,981728169XX,632731006XX,6/19/2018

```

Slika 43. Podaci o računima

```

,1,10,4
,1,5,4
,1,1,200
,2,4,2
,2,7,5
,2,9,1
,3,8,2
,3,1,50
,3,6,1
,3,5,5
,4,3,2
,5,2,1
,5,8,1

```

Slika 44. Podaci o popisu robe

```

BULK
INSERT ime_tablice
FROM 'putanja_do_datoteke'
WITH (
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n'
);

```

Slika 45. SQL naredba za unos podataka pomoću datoteke

Navedene podatke možemo unositi i pojedinačno prikazano na slici 46. tako da zamijenimo naziv_tablice s nazivom tablice u koju želimo unijeti vrijednosti i unijeti_potrebne_vrijednosti s vrijednostima koje želimo unijeti.

```
INSERT INTO naziv_tablice VALUES (unijeti_potrebne_vrijednosti);
```

Slika 46. *INSERT naredba za pojedinačan unos podataka*

Slika 47 prikazuje uklanjanje vrijednosti iz tablica korištenjem procedura.

```
EXEC obrisi_racunR2 '1';  
EXEC obrisi_artikl '1';  
EXEC obrisi_djelatnika '981728169XX';  
EXEC obrisi_poslPartnera '715716313XX';  
EXEC obrisi_dobavljacka '03834418154';
```

Slika 47. *Primjer brisanja određenih vrijednosti iz tablica pomoću procedura*

Slika 48 prikazuje pretraživanje vrijednosti u tablicama korištenjem procedura.

```
EXEC nadi_racunR2 '1';  
EXEC nadi_artikl_po_sifri '1';  
EXEC nadi_artikl_po_nazivu 'harmony';  
EXEC nadi_djelatnika_po_OIB '981728169XX';  
EXEC nadi_dobavljacka_po_OIB '03834418154';  
EXEC nadi_poslPartnera_po_OIB '715716313XX';
```

Slika 48. *Primjer pretraživanja željene tablice korištenjem procedura*

6. ZAKLJUČAK

Ovim seminarom je bilo potrebno napraviti bazu podataka za malu poljoprivrednu ljekarnu, koja sadrži podatke o dobavljačima, djelatnicima, poslovnim partnerima, artiklima, računima. Baza je napravljena na način da postoje osobe koje još uvijek posluju i koje možemo jednostavno pronaći koristeći OIB osobe za pretraživanje. Slično stanje sadrži artikl koji predstavlja artikle koji su još u prodaji ili nisu. Tablica artikala također sadrži informacije o dobavljaču radi lakšeg pristupa u kontakt s dobavljačem u slučaju nedostatka ili upita vezanog za artikl. Tablica za djelatnike ne sadržava previše informacija jer je namijenjena malom poduzeću koje ne zahtjeva puno djelatnika. Račun je prikazan pomoću tri tablice radi bolje preglednosti. Prva tablica sadrži šifru računa, datum izdavanja, ime djelatnika koji je izdao račun i ime poslovnog partnera kojemu je račun izdan. Druga tablica popis robe i treća ukupnu cijenu računa s PDV-om i bez PDV-a.