

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Iva Melnjak

SQL u sustavu Microsoft Access 2016

ZAVRŠNI RAD

Varaždin, 2018.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Iva Melnjak

Matični broj: 44303/15-R

Studij: *Primjena informacijske tehnologije u poslovanju*

SQL u sustavu Microsoft Access 2016

ZAVRŠNI RAD

Mentor:

Prof. dr. sc. Kornelije Rabuzin

Varaždin, rujan 2018.

Iva Melnjak

Izjava o izvornosti

Izjavljujem da je moj završni/diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Razvojem tehnologije pohrana podataka u papirnatom obliku zamijenjena je raznim sustavima za upravljanje bazom podataka. Danas imamo veliku ponudu takvih sustava, pri čemu su neki od njih besplatni, a neki su ipak investicija. Jedan od jednostavnijih sustava za izradu baze podataka je Microsoft Access.

Microsoft Access nudi nekoliko načina izrade baze podataka, čime omogućuje jednostavnu prilagodbu korisnika različitih zanimanja, znanja i interesa. U ovom radu naglasak se stavlja na izravno pisanje SQL koda u sustav Microsoft Access 2016. SQL je najviše korišten jezik za rad s bazama podataka. Sastoji se od određenog broja naredbi, pri čemu svaku naredbu čini nekoliko klauzula navedenih i objašnjenih u ovom radu.

Ključne riječi: SQL; Microsoft Access; tablice; upiti; klauzule; operatori; funkcije

Sadržaj

1. Uvod	1
2. Microsoft Access 2016	2
2.1. Povijest sustava	2
2.2. Izgled i način rada u sustavu	3
3. SQL	5
3.1. Komponente i sintaksa SQL-a.....	5
3.2. Razlike u mogućnostima i načinu pisanja.....	6
3.3. Primjer baze podataka	7
4. Naredbe, klauzule, operatori i funkcije koje podržava Microsoft Access 2016	9
4.1. Naredbe	9
4.2. Klauzule, operatori i funkcije	9
5. Rad s tablicama	11
5.1. Kreiranje tablica	11
5.1.1. Tipovi podataka.....	11
5.1.2. Ograničenja.....	12
5.2. Mijenjanje strukture tablice.....	13
5.3. Upravljanje podacima.....	13
5.4. Indeksi	14
6. Upiti	15
6.1. Kreiranje upita.....	15
6.1.1. TRANSFORM naredba	16
6.1.2. Grupiranje i sortiranje redova	16
6.2. Spajanje tablica i unija	17
6.2.1. Spajanje korištenjem FROM i WHERE klauzula.....	17
6.2.2. Spajanje pomoću klauzule JOIN.....	17
6.2.3. Unija.....	18
6.3. Operatori.....	20
6.3.1. Aritmetički operatori i operatori uspoređivanja	20
6.3.2. Logički operatori.....	21
6.3.3. Operatori IN, BETWEEN i LIKE.....	22
6.4. Funkcije	24
6.5. Podupiti.....	26
7. Zaključak	27
Popis literature	28
Popis slika	30
Popis tablica	31

1. Uvod

Microsoft Access 2016 nudi nekoliko načina izrade baze podataka. Jednostavan način, namijenjen novim korisnicima ili korisnicima koji ne žele pisati kod je izrada objekata pomoću već ugrađenih predložaka, te pomoću čarobnjaka. Predmet ovog završnog rada je prikazati mogućnosti i karakteristike spomenutog sustava sa strane SQL-a i njegovih klauzula. Cilj je upoznati se s najnovijom verzijom Microsoft Accessa kroz opciju kreiranja baze i njenih objekata izravnim upisivanjem koda, odnosno SQL iskaza.

Rad ima teorijski dio i praktični dio, ali su oni spojeni u jednu cjelinu. Svaka točka započinje teorijom, a završava praktičnim primjerom koji ju potkrepljuje. Praktični dio temelji se na samostalno izrađenoj bazi podataka u sustavu Microsoft Access 2016.

Sve slike u ovom radu su autorske slike vlastite baze podataka.

Prvi dio ovog rada odnosi se na neke osnovne informacije o programu i jeziku koji će se koristiti, razlikama koje su uočene tijekom rada na praktičnom dijelu, opisom baze podataka koja se koristi kao primjer, te popisom naredba i klauzula koje Microsoft Access prihvaća.

Nakon upoznavanja s temom i osnovnim pojmovima rad se dijeli na dva velika dijela. Prvi dio odnosi se na tablice, njihovo kreiranje, upravljanje njihovom strukturom, unošenje i upravljanje podacima u tablicama, te na kreiranje indeksa. Drugi dio odnosi se na upite, njihovo kreiranje, spajanje tablica pomoću upita, operatore i funkcije koje koristimo u Accessu, te na stvaranje podupita.

2. Microsoft Access 2016

2.1. Povijest sustava

Microsoft Access je dio programskog paketa Microsoft Office, a prva verzija izašla je u listopadu 1992. godine. To je relacijski sustav za upravljanje bazom podataka koji sprema podatke u relacije, odnosno tablice koje su u pravilu povezane, pri čemu je baza podataka skup organiziranih podataka kojima se može upravljati te koji se mogu pretraživati.

Sve verzije spomenutog sustava su:

Tablica 1: Verzije sustava Access

Godina	Verzija	Broj verzije
1992	Access 1.1	1
1993	Access 2.0	2.0
1995	Access for Windows 95	7.0
1997	Access 97	8.0
2000	Access 2000	9.0
2002	Access 2002	10
2003	Access 2003	11
2007	Microsoft Office Access 2007	12
2010	Microsoft Office Access 2010	14
2013	Microsoft Office Access 2013	15
2016	Microsoft Office Access 2016	16

(Prema: NevVB, bez dat.)

Microsoft Access 2016 kompatibilan je s Windows 10, Windows 8.1 i Windows 7. Ima 2 glavne komponente, Microsoftov mehanizam Jet za bazu podataka i Rapid Application Development (RAD) alat. RAD se odnosi na brzi razvoj aplikacija pri kojem se developer može primarno fokusirati na brzinu razvoja obrazaca i izvještaja vezanih za bazu podataka (Vuk i Ciriković, 2015.).

„Skoro svi objekti u bazi podataka Access 2016 nalaze se unutar jedne datoteke s ekstenzijom ACCDB ili ACCDE. Access baze podataka također mogu imati ekstenziju MDB ili MDE.“ (Alexander i Kusleika, 2016, str. 5).

2.2. Izgled i način rada u sustavu

Aktivnosti u Accessu možemo podijeliti u dvije grupe, izradu baze podataka i obradu podataka. Izrada same baze podataka odnosi se na izradu tablica, upita, obrazaca, izvještaja, makronaredbi i drugih objekata. Obrada podataka odnosi se na upisivanje novih podataka u objekte, ispravljanje, brisanje i dodavanje, te pretraživanje postojećih podataka.

U Microsoft Accessu baza podataka sastoji se od 6 glavnih vrsta objekata (Microsoft (11), bez dat.):

- Tablica - služi za pohranu podataka
- Upit - za pronalaženje i dohvat podataka
- Obrazac - za unošenje podataka prikazanih u prilagođenom formatu
- Izvještaj - prikaz podataka za ispis
- Makronaredba - dodavanje funkcionalnosti bazi i automatiziranje koraka
- Modul - obuhvaća pisanje naredbi u programskom jeziku Visual Basic for Applications

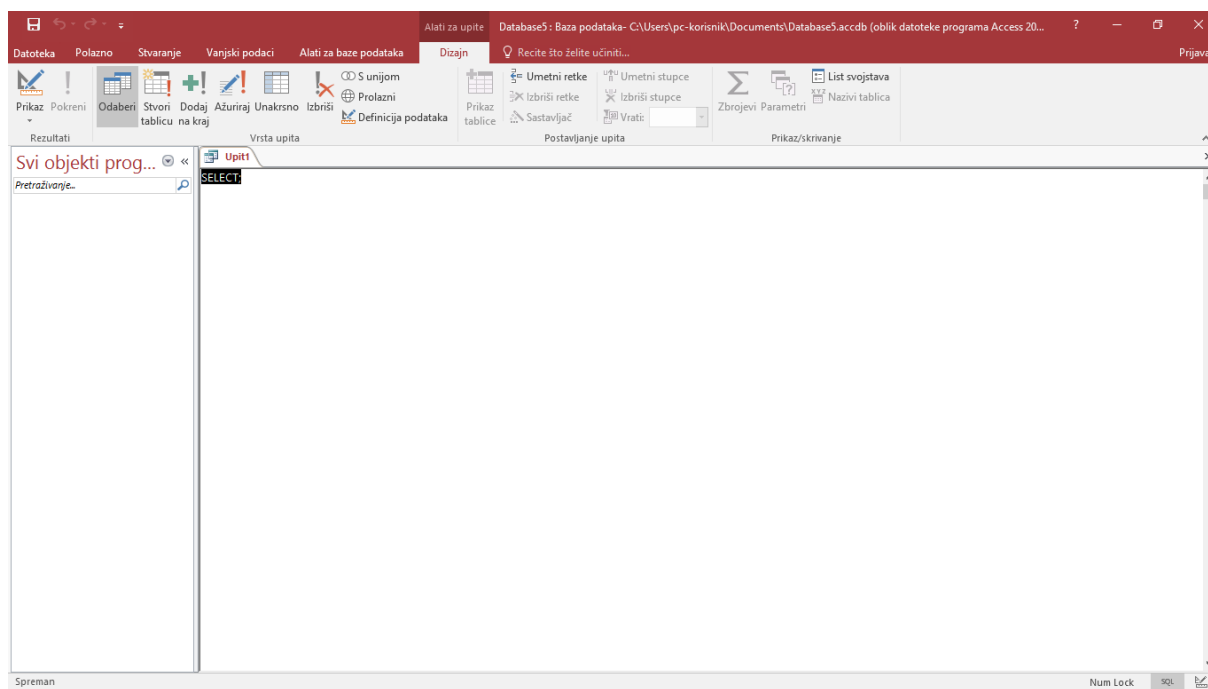
Početni prozor otvorene baze podataka u Microsoft Accessu sastoji se od 3 dijela:

1. Alatne trake koja se nalazi na vrhu i pokazuje moguće opcije iz odabranih izbornika
2. Stupca s lijeve strane prozora koji sadrži popis svih elemenata u trenutnoj bazi podataka
3. Najveći i središnji dio prozora služi za definiranje elemenata i karakteristika od kojih se sastoji baza

Alatna traka se sastoji od četiri osnovna izbornika:

- Polazno
- Stvaranje
- Vanjski podaci
- Alati za baze podataka

Zavisno o trenutnoj poziciji u programu, može se pojaviti i još neki izbornik. Na slici 1 možemo vidjeti izgled programa Microsoft Access 2016, pri čemu je u središnjem dijelu otvorena opcija kreiranja novog upita u SQL prikazu koji se koristi kroz cijeli rad. Prilikom otvaranja ove mogućnosti javlja se dodatni izbornik „Dizajn“ u kojem kad napišemo SQL iskaz klikom na Pokreni izvršavamo taj upit i dobivamo rezultat.



Slika 1: Izgled Microsoft Accessa 2016

U ovom programu osim SQL prikaza, omogućen je i takozvani prikaz dizajna, gdje u alatnoj traci odabiremo koji objekt želimo kreirati, te nakon toga u određenim poljima (ćelijama) definiramo i dodajemo potrebne elemente. Isto tako kod izrade objekata postoji čarobnjak koji uvelike olakšava postupak.

Microsoft Access nam omogućuje korištenje eksternih podataka, koji dolaze iz drugih izvora, kao što je druga Access baza ili Excel datoteka, tekstna datoteka, XML datoteka, HTML datoteka i slično.

3. SQL

SQL ili strukturirani upitni jezik je dio relacijskog sustava za upravljanje bazom podataka i deklarativan je, odnosno mi njime specificiramo što treba učiniti, a sustav određuje kako to optimalno izvršiti.

„SQL je postao standard Američkog Nacionalnog instituta za standarde (ANSI) 1986., te Međunarodne organizacije za standardizaciju (ISO) 1987. godine. Iako je SQL ANSI / ISO standard, postoje različite verzije SQL jezika. Međutim, kako bi bili u skladu s ANSI standardom, svi oni podržavaju barem glavne naredbe (kao što su SELECT, UPDATE, DELETE, INSERT, WHERE) na sličan način.“ (W3Schools, bez dat.)

Nakon spomenutih standarda se kroz godine, radi napretka i novih opcija razvilo još nekoliko standarda, a posljednja verzija je SQL:2016.

3.1. Komponente i sintaksa SQL-a

SQL ima dvije komponente, DDL (eng. Data Definition Language) je jezik za definiranje strukture podataka i DML (eng. Data Manipulation Language) je jezik za rukovanje podacima.

U DDL pripadaju naredbe CREATE, ALTER i DROP, odnosno naredbe za kreiranje, mijenjanje i brisanje postojećih objekata. Objekti mogu biti tablice, indeksi, domene, sheme, pogledi i slično, ali u Microsoft Accessu 2016 neke DDL naredbe i njihovi dijelovi nisu podržani.

DML obuhvaća naredbe INSERT, UPDATE i DELETE, odnosno naredbe koje omogućuju dodavanje podataka, ažuriranje i brisanje već postojećih podataka.

Kako bi poštivali pravila SQL-a moramo znati njegovu sintaksu. SQL naredbe sadrže ključne riječi, uvijek počinju glagolom koji označava što naredba radi, a nastavljaju objektom nad kojim se nešto radi. Isto tako neki dijelovi naredbe su obavezni, dok se neki ne moraju koristiti. „SQL naredbe nisu osjetljive na velika i mala slova, ali podaci u bazi jesu.“ „Identifikatori su nazivi objekata u bazi, oni bi trebali biti što kraći, i ne bi trebali sadržavati razmak.“ Kod Access-a ako identifikator ima razmak ili neki drugi znak najbolje je cijeli identifikator staviti u ugate zagrade. (Rabuzin, 2011).

SQL se sastoji od određenog broja naredbi, a svaka naredba se sastoji od određenih klauzula. U ovom radu SQL je podijeljen na upravljanje tablicama i upravljanje upitima, svaka cjelina ima nekoliko različitih skupina naredbi, odnosno klauzula. Neke od najbitnijih klauzula su CREATE DATABASE za kreiranje baze, CREATE TABLE za kreiranje tablica, te SELECT pomoću kojeg se definiraju upiti.

3.2. Razlike u mogućnostima i načinu pisanja

Postoji nekoliko razlika u mogućnostima kako u samom sustavu, tako i u SQL-u.

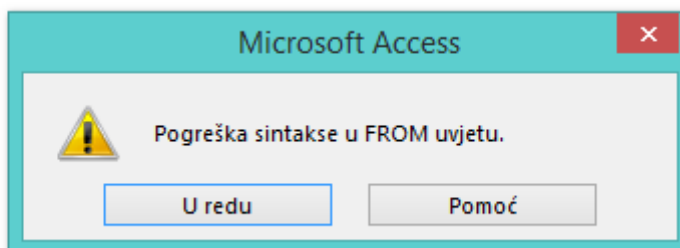
Objekti kao što su obrasci i izvještaji se mogu kreirati u samom Microsoft Accessu preko alatne trake i njezinih izbornika, ali se ne mogu kreirati preko SQL-a. Jedan od razloga je to što Microsoft Access ne podržava neke oblike DDL jezika.

U nekim programima je moguće pomoću naredbe INSERT odjednom dodati više redova u tablicu. Ta varijanta se temelji na mogućnosti da se zagrada s vrijednostima ponovi više puta. Ova opcija nije omogućena u Microsoft Accessu, već se svaki red dodaje zasebno.

Kod operatora LIKE koriste se drugačiji znakovi u Accessu, nego što se koriste u drugim programima. O tome detaljnije kod objašnjenja svih operatora.

Od operacija nad skupovima koje se mogu primijeniti na tablice, a to su unija, presjek i razlika u Microsoft Access-u moguće je koristiti samo uniju, pomoću operatora UNION. Pri pokušaju korištenja presjeka, odnosno operatora INTERSECT, Microsoft Access javlja grešku prikazanu na sljedećim slikama:

```
SELECT *  
FROM brojevi1  
INTERSECT  
SELECT *  
FROM brojevi3;
```



Slika 2: Pogreška kod presjeka 1.

```
SELECT *  
FROM brojevi1  
INTERSECT  
SELECT *  
FROM brojevi3;
```

Slika 3: Pogreška kod presjeka 2.

3.3. Primjer baze podataka

Još uvijek postoje takozvane ručne baze podataka kod kojih je glavno sredstvo papir. Papiri sadrže informacije i spremaju se u određene mape, koje držimo u ormaru poslovnog prostora, u različitim odjeljcima. Kako bi se pronašle potrebne informacije ponavlja se isti postupak otvaranja ormara, uzimanja točne mape i pronalaska odgovarajućeg papira na kojem se nalazi potrebna informacija. Podaci ulaze u sustav tako što ih određena osoba zapiše ili unese preko forme te ispiše dokument i sprema ga na odgovarajuće mjesto. Access baza je zapravo automatizirana verzija funkcija podnošenja i pronalaženja kod ručne baze podataka. Ona sprema informacije u pažljivo definiranu strukturu. (Alexander i Kusleika, 2016)

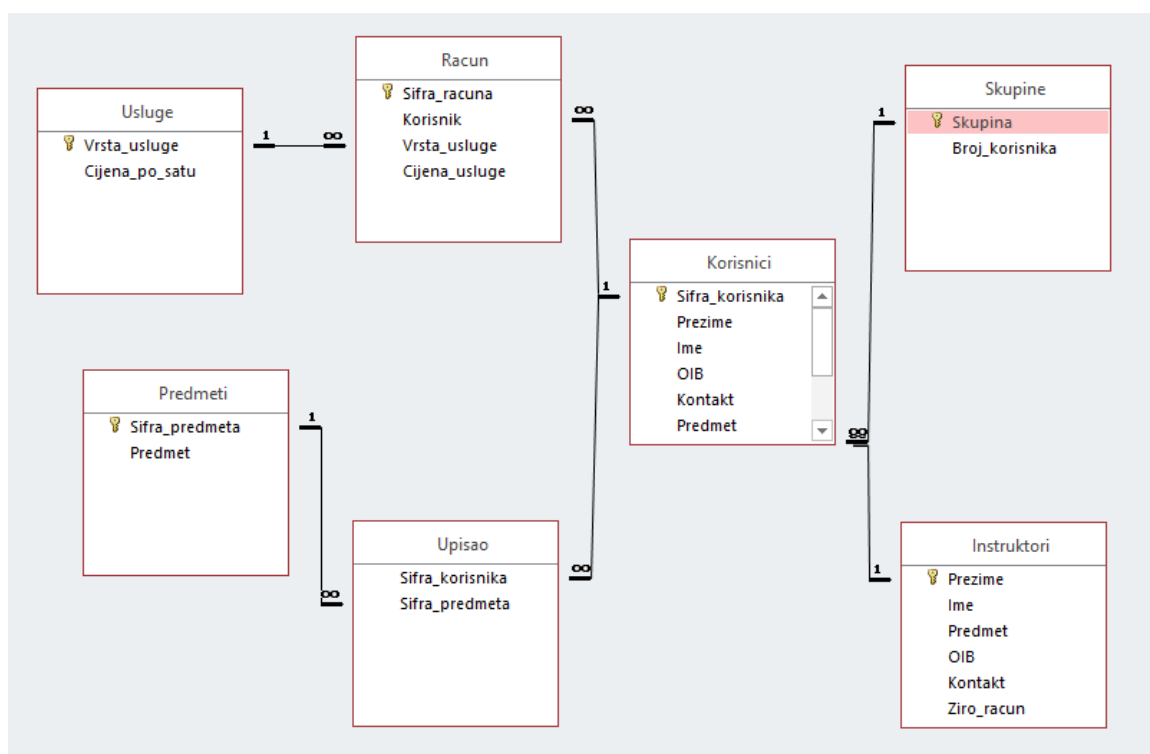
„Baza podataka je organizirani elektronički skup podataka, pripremljen tako da se može jednostavno koristiti. To znači da se podaci u bazi podataka mogu jednostavno koristiti, pregledavati, pretraživati, sortirati, uspoređivati, te također, mijenjati, dodavati i brisati. Podaci su u bazi podataka pohranjeni i sustavno upravljani.“ (Kaluža, 2008)

Baze podataka se koriste u bankama, bolnicama, školama, knjižnicama, tvrtkama i sličnim ustanovama, gdje se vodi evidencija o korisnicima, stvarima, rokovima, robi, materijalu i ostalim bitnim stavkama poslovanja. Primjena baze ovim ustanovama daje viši stupanj organiziranosti podataka, veću sigurnost, brzinu obrade i pretrage podataka, te kontrolu pristupa podacima.

Kao što je već rečeno kroz ovaj rad teorija je potkrijepljena autorskom bazom podataka, izrađenom u programu Microsoft Access 2016, pomoću SQL jezika, za uslugu pružanja instrukcija učenicima, iz različitih predmeta i različitih skupina. Baza podataka potrebna je radi lakšeg snalaženja i korištenja svih informacija potrebnih za uspješno i organizirano vođenje poslovanja.

Baza podataka „Instrukcije“ ima 8 tablica i više od 30 različitih vrsta upita, kao što su unakrsni upit, union upit, upit za stvaranje tablice, dodaj upit, select upit, te upit za definiciju podataka.

Najbitnija tablica je tablica „Korisnici“ jer ona spaja 4 različite tablice, to su „Racun“, „Skupine“, „Instruktori“ i „Upisao“. „Racun“ je dalje povezan s tablicom „Usluge“, a „Upisao“ je tablica stvorena radi prekida veze više naprema više između korisnika i predmeta. Tako su sve veze u bazi jedan naprema više, na primjer račun je namijenjen jednom korisniku, ali jedan korisnik može biti na više računa. Odnosi između tablica prikazani su na slici 4.



Slika 4: Odnosi u bazi podataka

4. Naredbe, klauzule, operatori i funkcije koje podržava Microsoft Access 2016

Ovaj sustav podržava samo određene SQL naredbe i klauzule, koje su nabrojene u nastavku. Neke naredbe koje su podržane u ranijim verzijama ovog sustava ili u drugim sustavima moguće je provesti samo pomoću ugrađenog Visual Basic for Applications (VBA) jezika. Primjer su CREATE VIEW, CREATE PROCEDURE, CREATE USER, TRANSACTION naredba i slično. „Sigurnosne značajke na razini korisnika nisu dostupne u Access web aplikacijama, web-bazama podataka ili bazama podataka koje koriste jedan od novih formata datoteka (.accdb, .accde, .accdc, .accdr).“ (Microsoft (14), bez dat.)

4.1. Naredbe

Naredbe za kreiranje objekata:

- CREATE (TABLE, INDEKS)
- ALTER (TABLE, COLUMN)
- DROP

Naredbe za manipuliranje podacima:

- INSERT (INTO)
- UPDATE
- DELETE
- TRANSFORM

Naredba za postavljanje upita:

- SELECT (INTO)

4.2. Klauzule, operatori i funkcije

Klauzule koje podržava Microsoft Access 2016 i koje su objašnjene u ovom radu su:

- FROM
- WHERE
- SET

- VALUES
- ADD
- INNER JOIN
- LEFT JOIN, RIGHT JOIN
- GROUP BY
- ORDER BY
- CONSTRAINT

Operatori su sljedeći:

- IN, BETWEEN, LIKE
- UNION
- AND, OR, NOT

Funkcije koje podržava spomenuti sustav su:

- COUNT()
- SUM()
- AVG()
- MAX(), MIN()
- FIRST(), LAST()
- VarP(), Var()
- StDevP(), StDev()

Microsoft Access 2016 još i podržava opciju DISTINCT, te podupite.

5. Rad s tablicama

5.1. Kreiranje tablica

Tablica je zbirka podataka organiziranih u redove i stupce, pri čemu je stupac skup podataka određenog tipa, a red je zapis u tablici. (Codecademy, bez dat.)

Naredba CREATE omogućuje kreiranje različitih objekata, a kako bi kreirali tablicu koristimo naredbu CREATE TABLE. Nakon te naredbe određujemo ime tablice, a unutar zagrada određujemo ime stupca i tip podatka svakog stupca. Također možemo odrediti ograničenja nad stupcima i dodijeliti ključeve. Na slici ispod prikazan je primjer kreiranja tablice „Korisnici“, koja ima 9 stupaca. Kako bi znali specificirati stupce moramo znati tipove podataka i da li želimo ili trebamo koje ograničenje.

```
CREATE TABLE Korisnici (  
Sifra_korisnika INTEGER NOT NULL PRIMARY KEY,  
Prezime TEXT(30),  
Ime TEXT(30),  
OIB TEXT(11) NOT NULL UNIQUE,  
Kontakt TEXT(10),  
Skupina TEXT(50) NOT NULL CONSTRAINT FK_Skupina REFERENCES Skupine(Skupina),  
Vrsta_usluge TEXT(30),  
Instruktori TEXT(30) NOT NULL CONSTRAINT FK_Instruktori REFERENCES Instruktori(Prezime),  
Placeno YESNO );
```

Slika 5: Kreiranje tablice

5.1.1. Tipovi podataka

Nakon naziva stupca prvo što se specificira je tip podatka. Tipovi podataka razlikuju se u svakom programu, Microsoft Access 2016 nudi njih 11, to su:

- Kratki tekst (Short Text) - alfanumerički znakovi; do 255 znakova
- Dugi tekst (Long Text) - alfanumerički znakovi; do 1 gigabajta
- Broj (Number) - numeričke vrijednosti; 1,2,4,8 ili 16 bajta
- Datum/vrijeme (Date/Time) - datum i vrijeme; 8 bajtova
- Valuta (Currency) - monetarni podaci, novac; 8 bajtova
- Samonumeriranje (AutoNumber) - jedinstvena vrijednost generirana za svaki novi zapis; 4 bajta
- Da/ne (Yes/No) - boolean podaci; 1 bit, sprema se numerička vrijednost 0 za ne (laž), a -1 za da (istinu)
- OLE objekt (OLE Object) - slike, grafikoni, zvuk, video; do 1 gigabajta

- Hiperveza (Hyperlink) - poveznica na datoteku na internetu; do 1 gigabajta znakova
- Privitak (Attachment) - polje koje omogućuje da se priloži posebna vanjska datoteka u bazu podataka
- Izračunato (Calculated) - prikazuje rezultate nekog izračuna

Ako imamo tip podatka Tekst dodavanjem broja u zagradi možemo specificirati željenu veličinu polja. Isto to možemo i za broj, samo što umjesto Number zapisujemo onu veličinu polja koju želimo. Tako u primjeru imamo tip Integer koji predstavlja cijeli broj, veličine 2 bajta. Imamo još i veličine Byte, Long Integer, Single, Double i Decimal.

(Microsoft (9), bez dat.; Vazler i Đumić, 2014-15)

5.1.2.Ograničenja

Kako bi pokušali spriječiti zapisivanje „krivih podataka“ u tablicu, postoje određena ograničenja koja kontroliraju što se upisuje. Prvo ograničenje koje možemo vidjeti na slici 5 je primarni ključ (PRIMARY KEY). Primarni ključ se koristi za jedinstveno identificiranje reda, odnosno pokušamo li unijeti vrijednost retka koja već postoji ovo ograničenje će to spriječiti. Dobar prijedlog za primarni ključ je onaj koji jedinstveno označava svaki redak, vrijednosti koje sadrži rijetko se mijenjaju i nikad nije prazno polje niti ima vrijednost NULL.

Kako bi spriječili da bilo koji stupac ima prazan red dodajemo ograničenje NOT NULL. U primjeru smo za stupac OIB dodali ograničenje UNIQUE, jer znamo da svaka osoba ima različit osobni identifikacijski broj te želimo spriječiti ponavljanje tog broja. Dakle, ovo ograničenje je jednako kao i primarni ključ, samo što jedna tablica može imati više stupaca s ograničenjem UNIQUE, a samo jedan s ograničenjem PRIMARY KEY.

Osim primarnog ključa, za određeni stupac možemo definirati i vanjski ključ (Foreign Key - FK). Vanjski ključ je stupac u drugoj tablici u kojoj pohranjujemo vrijednost(i) primarnog ključa tablice na koju se vanjski ključ referencira. To pravilo kojim osiguravamo da svaka vrijednost vanjskog mora odgovarati nekoj vrijednosti primarnog ključa, nazivamo referencijalni integritet (Maleković i Rabuzin, 2016).

U primjeru vanjski ključ se implementira kao ograničenje nad stupcem pomoću klauzule CONSTRAINT, dodjeljuje se ime vanjskog ključa, te se definira na koju tablicu i koji stupac se vanjski ključ referencira.

5.2. Mijenjanje strukture tablice

Nakon što su tablice kreirane može doći do potrebe da se promijeni njihova struktura, na primjer da dodamo stupac, obrišemo stupac ili promijenimo neko ograničenje, i slično. To nam omogućuje naredba ALTER TABLE. S ovom naredbom možemo koristiti klauzule ADD za dodavanje stupca, ALTER COLUMN za mijenjanje tipa i veličine podatka određenog stupca, DROP COLUMN za brisanje stupca, te ADD CONSTRAINT za dodavanje ograničenja na određeni stupac.

Na slici ispod možemo vidjeti primjer iz baze podataka, gdje je tip podatka u stupcu „Konacna_cijena“ promijenjen u valutu.

```
ALTER TABLE Popust  
ALTER COLUMN Konacna_cijena CURRENCY;
```

Slika 6: Struktura tablice

5.3. Upravljanje podacima

Za dodavanje podataka u tablicu koristimo naredbu INSERT. Podatke možemo dodati na dva načina, dodavanjem svakog reda posebno, ili dodavanjem više redova odjednom.

Za dodavanje red po red s naredbom INSERT INTO koristimo klauzulu VALUES nakon koje u zagradama ispisujemo vrijednosti, odvojene zarezom, koje želimo spremiti u bazu. Ako upisujemo podatke za svaki stupac nije potrebno posebno navesti imena stupaca u zagradi nakon naziva tablice. Ako pak upisujemo podatke samo za određene stupce, imena tih stupaca moramo navesti prije klauzule VALUES. Kod navođenja podataka, sve vrste podataka se pišu pod jednostrukim navodnicima, osim brojeva. Primjer ovakvog dodavanja prikazan je na sljedećoj slici 7.

```
INSERT INTO Korisnici  
VALUES (25, 'Fodor', 'Tina', 12875492563, 0998567425, 'Kemija', '7. razred osnovne škole', 'Grupna', 'Piskač', yes);
```

Slika 7: Dodavanje podataka - red po red

Kod dodavanja više redova odjednom koristimo klauzulu SELECT, pomoću koje dohvaćamo podatke iz stupaca koji se nalaze u drugim tablicama. Svaki SELECT ima klauzulu FROM koja govori iz kojih tablica dohvaćamo podatke.

```
INSERT INTO Upisao ( Sifra_korisnika, Sifra_predmeta )  
SELECT K.Sifra_korisnika, P.Sifra_predmeta  
FROM Korisnici AS K, Predmeti AS P;
```

Slika 8: Dodavanje podataka - više redova

Osim dodavanja, vrijednosti u tablici možemo i mijenjati pomoću klauzule UPDATE. Nakon te klauzule definiramo ime tablice u kojoj želimo nešto mijenjati, a zatim pomoću klauzule SET definiramo stupac i vrijednost koju mu želimo dodijeliti. Po potrebi pomoću klauzule WHERE možemo i specificirati koji točno red želimo promijeniti.

Isto tako podatke možemo i obrisati, i to pomoću klauzule DELETE. Možemo obrisati podatke jednog stupca ili cijele tablice, pri čemu se struktura tablice ne mijenja. Kod brisanja možemo koristiti samo ovu klauzulu ili možemo dodati klauzulu WHERE, kojom ćemo detaljnije definirati što želimo obrisati.

(Microsoft, 2017)

5.4. Indeksi

Microsoft Access koristi indekse kako bi sustav brže pronašao određene redove u tablicama, odnosno oni poboljšavaju upitne performanse. Neke indekse sustav sam kreira, kao što je to za primarni ključ, a nekad mi to moramo sami.

Indeksi su jako korisni jer smanjuju vrijeme pretrage. Iz tog razloga indekse stavljamo na one stupce čije ćemo vrijednosti puno pretraživati, sortirati i koristiti u drugim tablicama i objektima. Iako ubrzavaju navedene postupke, indeksi usporavaju postupak kod dodavanja podataka i ažuriranja.

Indeks kreiramo korištenjem naredbe CREATE INDEX, nakon koje dolazi ime indeksa, ključna riječ ON, ime tablice nad kojom se indeks kreira, te ime stupca. U sljedećem primjeru kreirat će se indeks „Ime“ nad istoimenim stupcem koji se nalazi u tablici „Instruktori“.

```
CREATE INDEX Ime ON Instruktori (Ime);
```

6. Upiti

6.1. Kreiranje upita

„Prema SQL standardu upit je operacija kojim se referenciramo na nula ili više tablica, a kao rezultat izvršavanja upita opet nastaje tablica.“ (Rabuzin, 2011., str.43)

Kreiranje upita i dohvat podataka iz baze omogućuje naredba SELECT. Svaki upit započinje tom naredbom. Osim SELECT obavezna klauzula kod definiranja upita je klauzula FROM. Ostale klauzule su opcionalne.

Kod definiranja upita nakon ključne riječi SELECT navodimo što želimo u odgovoru. To mogu biti stupci tablice, rezultati funkcije, izrazi i slično. U klauzuli FROM navodimo izvore podataka, koji mogu biti tablica (ili više njih) ili bilo koji objekt koji nam kao rezultat vraća skup redova. Ako želimo dohvatiti sve stupce iz određene tablice nakon SELECT stavljamo znak * (zvjezdicu), a ako želimo dohvatiti više stupaca, izraza, funkcija, i slično, iste odvajamo zarezom.

Na slici ispod možemo vidjeti kako izgleda jedan upit sa SELECT i FROM klauzulama, a u nastavku ćemo objasniti ostale dijelove ovog upita.

```
TRANSFORM Count(Placeno)
SELECT [Korisnici.Prezime]+" "+[Korisnici.Ime] AS Korisnici
FROM Instruktori INNER JOIN Korisnici ON Instruktori.Prezime = Korisnici.Instruktori
GROUP BY [Korisnici.Prezime]+" "+[Korisnici.Ime]
ORDER BY [Korisnici.Prezime]+" "+[Korisnici.Ime], [Instruktori.Prezime]+" "+[Instruktori.Ime]
PIVOT [Instruktori.Prezime]+" "+[Instruktori.Ime];
```

Slika 9: Unakrsni upit

Još jedna opcija kod upita je DISTINCT opcija. Ova opcija omogućava da se vrijednosti u rezultatu ne ponavljaju, odnosno kao rezultat nam daje samo različite vrijednosti. DISTINCT isto tako u obzir uzima NULL vrijednosti, pa ako dva ili više reda imaju NULL vrijednost, u rezultatu ostaje samo jedna. Iako je ova opcija korisna u nekim slučajevima, smatra se da zbog provjere i izbacivanja duplikata usporava rad.

Jednostavan primjer upita s ovom opcijom imamo na slici 10, gdje će rezultat biti popis svih različitih predmeta iz tablice Korisnici.

```
SELECT DISTINCT Predmet
FROM Korisnici;
```

Slika 10: Opcija DISTINCT

6.1.1.TRANSFORM naredba

Ova naredba omogućuje stvaranje unakrsnog upita koji daje kompaktniji oblik podataka. Sastoji se od tri dijela. Prvi dio se odnosi na zadanu funkciju čiji rezultat unakrsni upit grupira, u ovom slučaju Count(Placeno). Drugi dio odnosi se na naredbu SELECT koja određuje koja će se polja koristiti kao naslovi redaka, u ovom primjeru spojena su dva stupca kojima je pomoću opcije AS dodijeljeno smisljeno ime. A posljednji dio, takozvano PIVOT polje određuje nazive stupaca u upitu (Microsoft (12), bez dat.).

6.1.2.Grupiranje i sortiranje redova

„Za sortiranje redova u rezultatu koristimo klauzulu ORDER BY. Sortirati možemo prema vrijednostima nekog stupca, prema nekom izrazu ili npr. prema rezultatu neke funkcije.“ (Rabuzin, 2011). Isto tako možemo odrediti silazno sortiranje dodavanjem ključne riječi DESC (descending) ili uzlazno sortiranje dodavanjem ključne riječi ASC (ascending). U primjeru su vrijednosti sortirane uzlazno, jer je tako zadano u samom programu. Isto tako moguće je umjesto naziva stupca navesti redni broj stupca prema kojem želimo sortirati.

Za grupiranje redova u rezultatu koristimo klauzulu GROUP BY. Ova klauzula omogućuje dijeljenje podataka u logičke grupe, broj kreiranih grupa jednak je broju različitih vrijednosti u stupcu. Ako nemamo agregirajuće funkcije, kao u primjeru gore ova klauzula daje jednak rezultat kao i opcija DISTINCT. Zato se ova klauzula najčešće koristi s agregirajućim funkcijama, kao i u primjeru na slici 11, ali te funkcije će biti objašnjenje dalje u radu.

```
SELECT Instruktori, Count(Prezime) AS Broj_korisnika  
FROM Korisnici  
GROUP BY Instruktori;
```

Slika 11: Grupiranje redova

6.2. Spajanje tablica i unija

Postoje dva načina spajanja tablica. Na slikama 7 i 8 prikazani su primjeri ta dva načina spajanja koji daju iste rezultate, ali koriste različite klauzule kako bi to postigli. Ovi načini detaljnije su objašnjeni u točkama 6.2.1. i 6.2.2.

```
SELECT K.Prezime, K.Ime, P.Predmet  
FROM Korisnici AS K, Predmeti AS P  
WHERE P.Predmet = K.Predmet  
ORDER BY Prezime;
```

Slika 12: Spajanje klauzulama FROM i WHERE

```
SELECT Korisnici.Prezime, Korisnici.Ime, Predmeti.Predmet  
FROM Predmeti INNER JOIN Korisnici ON Predmeti.Predmet = Korisnici.Predmet  
ORDER BY Prezime;
```

Slika 13: Spajanje klauzulom JOIN

6.2.1. Spajanje korištenjem FROM i WHERE klauzula

Stariji način spajanja tablica je pomoću klauzula FROM i WHERE. U FROM klauzuli navodimo tablice iz kojih dohvaćamo podatke, a u WHERE klauzuli je potrebno te tablice spojiti na pravilan način. U pravilu se spajanje tablica provodi pomoću primarnih i vanjskih ključeva, tako da im u WHERE klauzuli izjednačimo vrijednosti.

Ako se u dvije tablice javlja stupac istog naziva, koristimo notaciju ime_tablice.ime_stupca. Isto tako kao i u prethodnom primjeru tablicama možemo dodijeliti druga, skraćena imena pomoću ključne riječi AS (Rabuzin, 2011).

6.2.2. Spajanje pomoću klauzule JOIN

Drugi, noviji način spajanja tablica je pomoću klauzule JOIN. Ovaj način spajanja možemo podijeliti u dvije skupine.

Prva skupina je INNER JOIN skupina. Njome se tablice spajaju tako da se odaberu samo oni zapisi koji u obje tablice imaju iste vrijednosti. Zapisi koji nemaju jednaku vrijednost su izostavljeni. Tako u primjeru sa slike 13, SQL odabire samo one zapise u kojima se naziv predmeta nalazi i u tablici Predmeti i u tablici Korisnici. Kod pisanja SQL izjave riječ INNER možemo i izostaviti.

Druga skupina je OUTER JOIN skupina. Ona se dijeli na LEFT i RIGHT JOIN klauzule. LEFT JOIN govori Accessu da dohvati sve zapise iz prve (lijeve) tablice, a samo one zapise iz druge (desne) tablice koji imaju jednaku vrijednost u pridruženim poljima. RIGHT JOIN govori Accessu da dohvati sve zapise iz druge tablice (desne), a samo one zapise iz prve (lijeve) tablice koji imaju jednaku vrijednost u pridruženim poljima (Alexander i Kusleika, 2016).

Ako kod ove vrste spajanja postoji red koji se ne može spojiti, istom će se dodijeliti vrijednost NULL. Na primjer želimo pomoću LEFT JOIN klauzule spojiti predmete s njihovom cijenom (FROM predmeti LEFT JOIN cijene_predmeta). Za jedan od tih predmeta nije definirana cijena, te će se za taj predmet ispisati red u kojem će u prvom stupcu biti naziv tog predmeta, a drugi stupac će biti prazan, točnije dodati će mu se vrijednost NULL.

Tako u sljedećem primjeru klauzula RIGHT JOIN dohvaća sve podatke iz tablice Racun i samo one podatke iz tablice Korisnici gdje su vrijednosti šifre korisnika jednake kao i u tablici Racun.

```
SELECT K.Predmet, SUM(R.Cijena_usluge) AS Ukupna_cijena  
FROM Korisnici AS K RIGHT JOIN Racun AS R ON K.Sifra_korisnika = R.Korisnik  
GROUP BY Predmet;
```

Slika 14: RIGHT JOIN klauzula

Postoji i potpuno vanjsko spajanje, odnosno FULL OUTER JOIN ili samo FULL JOIN. Ono vraća sve zapise iz spojenih tablica, čak i ako ne zadovoljavaju kriterij pridruživanja. Kao i kod prijašnjih vanjskih spajanja za red koji nema odgovarajuću vrijednost dodat će mu se NULL. Access ne podržava ovu klauzulu, ali je ovaj način spajanja moguće ostvariti drukčijim putem.

Jednak rezultat dobit ćemo spajanjem tri različita spoja unijom. Potrebno je izvršiti unutarnji spoj (INNER JOIN), te lijevi i desni vanjski spoj (LEFT/RIGHT JOIN). Nakon toga dobivene rezultate spojimo pomoću operatora UNION (Larsen, 2005).

Kao rezultat dobit ćemo tablicu koja sadrži zapise koji su jednaki u obje tablice, ali i zapise koje ima samo lijeva, te zapise koje ima samo desna tablica. Odnosno dobit ćemo jednak rezultat kao da smo koristili potpuno vanjsko spajanje.

6.2.3.Unija

U Microsoft Accessu omogućeno je spajanje dvije tablice u jednu pomoću unije, odnosno operatora UNION. Prilikom definiranja ove operacije najbitnija je struktura samih tablica, odnosno da bi se operacija mogla provesti, tablice koje koristimo moraju imati jednak broj stupaca, te ti stupci moraju imati jednake tipove podataka (Rabuzin, 2011).

Unija spaja dvije tablice, odnosno rezultat drugog upita se dodaje rezultatu prvog. Kao primjer iz baze podataka na slici 15 imamo uniju dvaju upita koji prikazuju ukupnu cijenu usluge, odnosno prosječnu cijenu usluge po predmetu.

```
SELECT K.Predmet, AVG(R.Cijena_usluge) AS Cijena
FROM Korisnici AS K RIGHT JOIN Racun AS R ON K.Sifra_korisnika = R.Korisnik
GROUP BY Predmet
UNION
SELECT K.Predmet, SUM(R.Cijena_usluge) AS Cijena
FROM Korisnici AS K RIGHT JOIN Racun AS R ON K.Sifra_korisnika = R.Korisnik
GROUP BY Predmet;
```

Slika 15: Unija

Svaki upit ima 2 stupca, pa će prema tome i rezultat unije imati dva stupca. Prilikom spajanja unija dodaje vrijednosti, ali i izbacuje duplikate. Tako u rezultatu ove unije možemo vidjeti da svaki predmet ima po dva reda, jedan za ukupnu, a jedan za prosječnu cijenu, osim Fizike. Fizika ima samo jedan red jer su njena ukupna cijena i prosječna cijena jednake, te je time duplikat izbačen iz rezultata.

Cijene_po_predmetima	
Predmet	Cijena
Fizika	200,00 kn
Kemija	120,00 kn
Kemija	240,00 kn
Matematika	134,00 kn
Matematika	670,00 kn
Računovodstvo	207,50 kn
Računovodstvo	830,00 kn
Statistika	100,00 kn
Statistika	300,00 kn

Slika 16: Rezultat unije

6.3. Operatori

6.3.1. Aritmetički operatori i operatori uspoređivanja

Aritmetički operatori koriste se za obavljanje numeričkih izračuna. Imamo sedam osnovnih operatora:

Tablica 2: Aritmetički operatori

+	Zbrajanje dvaju brojeva.
-	Traženje razlike ili označavanje negativne vrijednosti broja.
*	Množenje dvaju brojeva.
/	Dijeljenje prvog broja drugim.
\	Zaokruživanje dvaju brojeva na cijele brojeve, dijeljenje prvog broja drugim, i skraćivanje rezultata na cijeli broj.
^	Potenciranje broja na eksponent.
Mod	Dijeljenje prvog broja drugim i prikaz samo ostatka dijeljenja.

(Prema: Microsoft (13), bez dat.)

Primjer aritmetičkih operatora možemo vidjeti u sljedećem upitu, gdje se izračunava cijena s popustom, tako što se množenjem dobije iznos popusta na cijenu ($R.Cijena_usluge * 0.2$), a zatim se od trenutne cijene oduzme taj dobiveni iznos.

```
SELECT K.Sifra_korisnika, K.Ime, K.Prezime, R.Cijena_usluge, (R.Cijena_usluge-(R.Cijena_usluge)*0.2) AS Konacna_cijena INTO Popust
FROM Korisnici_za_popust AS K LEFT JOIN Racun AS R ON K.Sifra_korisnika = R.Korisnik
ORDER BY Sifra_korisnika;
```

Slika 17: Aritmetički operatori

Operatori uspoređivanja služe za usporedbu dviju vrijednosti, a kao rezultat vraćaju istinu (TRUE), laž (FALSE) ili NULL vrijednost. U slučaju usporedbe s NULL vrijednosti rezultat će uvijek biti NULL, bez obzira na operator. U Accessu se najčešće koriste:

Tablica 3: Operatori uspoređivanja

=	Vraća istinu ako su vrijednosti jednake.
<>	Vraća istinu ako su vrijednosti različite.
<	Vraća istinu ako je vrijednost s lijeve strane operatora manja od vrijednosti s desne strane.
<=	Vraća istinu ako je vrijednost s lijeve strane operatora manja ili jednaka vrijednosti s desne strane.
>	Vraća istinu ako je vrijednost s lijeve strane operatora veća od vrijednosti s desne strane.
>=	Vraća istinu ako je vrijednost s lijeve strane operatora veća ili jednaka vrijednosti s desne strane.

(Prema: Alexander i Kusleika, 2016; Vuk i Ciriković, 2015.)

6.3.2. Logički operatori

Logički (Boolean) operatori koriste se za stvaranje višestrukih uvjeta, te vraćaju istinu (TRUE), laž (FALSE) ili NULL vrijednost. Najčešće korišteni logički operatori su AND, OR i NOT.

AND operator vraća istinu samo ako su oba izraza (vrijednosti) istinita. U sljedećem primjeru će se ispisati predmeti i instruktori samo za one retke u kojima je predmet matematika, a instruktor Bezak:

```
SELECT Predmet, Instruktori
FROM Korisnici
WHERE Predmet = 'Matematika' AND Instruktori = 'Bezak';
```

OR operator vraća istinu kad je barem jedan od izraza istinit. U sljedećem primjeru će se ispisati predmeti ako je vrijednost pod stupcem Predmet jednaka „Matematika“ ili „Statistika“:

```
SELECT Predmet
FROM Korisnici
WHERE Predmet = 'Matematika' OR Predmet = 'Statistika';
```

NOT operator vraća istinu ako je izraz lažan. U sljedećem primjeru će se ispisati predmeti ako vrijednost pod stupcem Predmet nije „Matematika“, odnosno ispisat će se svi ostali predmeti:

```
SELECT Predmet  
  
FROM Korisnici  
  
WHERE NOT Predmet = 'Matematika';
```

6.3.3. Operatori IN, BETWEEN i LIKE

```
SELECT Sifra_korisnika, Ime, Prezime  
FROM Korisnici  
WHERE Sifra_korisnika IN (  
SELECT Korisnik  
FROM Racun  
WHERE Cijena_usluge BETWEEN 180 AND 500);
```

Slika 18: Operatori IN i BETWEEN

Na slici 18 prikazan je primjer podupita u kojem su korišteni operatori IN i BETWEEN. Operator IN provjerava nalazi li se vrijednost stupca u skupu vrijednosti koji smo definirali unutar zagrada. U ovom primjeru provjerava se da li su vrijednosti u tablici Racun pod stupcem Korisnik u stupcu Sifra_korisnika tablice Korisnici.

Operator BETWEEN omogućuje definiranje raspona koji vrijednosti moraju zadovoljavati. Vrijednosti mogu biti brojevi, tekst i datum. Ako provjeravamo brojeve uključuju se i granične vrijednosti, odnosno u primjeru se promatraju usluge čija je cijena jednaka ili veća od 180,00 kuna i jednaka ili manja od 500,00 kuna.

Operator LIKE traži specifičan uzorak u stupcu. Bitna razlika od drugih programa je što se u programu Microsoft Access umjesto "%" koristi "*" i umjesto "_" koristi "?". Za negaciju se koristi "!". Primjeri korištenja ovog operatora:

Tablica 4: Operator LIKE

SELECT Field 1 FROM Table1 WHERE Field1 LIKE "A*"	Odabire sve zapise u kojima Field1 počinje slovom A.
SELECT Field 1 FROM Table1 WHERE Field1 LIKE "**A"	Odabire sve zapise u kojima Field1 ima(uključuje) slovo A.
SELECT Field 1 FROM Table1 WHERE Field1 LIKE "*A"	Odabire sve zapise u kojima Field1 završava slovom A.
SELECT Field 1 FROM Table1 WHERE Field1 LIKE "???"	Odabire sve zapise u kojima je duljina Field1, 3 znaka.
SELECT Field 1 FROM Table1 WHERE Field1 LIKE "####"	Odabire sve zapise u kojima je Field1 broj duljine 3 znamenka.
SELECT Field 1 FROM Table1 WHERE Field1 LIKE "**[\$%!*]/**"	Odabire sve zapise u kojima Field1 uključuje bilo koji od specijalnih znakova prikazanih u iskazu.

(Prema: Alexander i Kusleika, 2016)

Primjer iz baze podataka provjerava koji korisnici koriste Simpu, odnosno ispisuje korisnike i njihove kontakte, kod kojih kontakt započinje brojevima 099 i ima sveukupno 10 znamenaka.

```
SELECT Ime, Prezime, Kontakt
FROM Korisnici
WHERE Kontakt LIKE "099????????";
```

Slika 19: Primjer operatora LIKE

6.4. Funkcije

U Microsoft Accessu imamo takozvane agregirajuće funkcije koje na temelju skupa vrijednosti vraćaju jednu vrijednost. To su:

- COUNT() - zbraja broj ne-praznih vrijednosti u stupcu
- SUM() - vraća sumu svih vrijednosti u stupcu
- AVG() - vraća prosječnu vrijednost stupca
- MAX() - vraća maksimalnu vrijednost u stupcu
- MIN() - vraća minimalnu vrijednost u stupcu
- FIRST() - vraća prvu vrijednost u stupcu
- LAST() - vraća posljednju vrijednost u stupcu
- VarP(), Var() - vraćaju procijenjenu varijancu populacije, odnosno uzorka populacije
- StDevP(), StDev() - vraćaju procijenjenu standardnu devijaciju populacije, odnosno uzorka populacije

COUNT(stupac) funkcija vraća broj redova nekog stupca i pritom ne uzima u obzir NULL vrijednosti, dok COUNT(*) koja vraća ukupan broj redova uzima u obzir i NULL. U primjeru iz baze podataka Instrukcije, ova funkcija omogućila je prikaz broja korisnika po svakoj vrsti usluge.

```
SELECT Vrsta_usluge, Count(Vrsta_usluge) AS Broj_korisnika  
FROM Korisnici  
GROUP BY Vrsta_usluge;
```

Slika 20: Funkcija COUNT()

SUM funkcija uzima ime stupca kao argument i vraća sumu svih vrijednosti u tom stupcu, a AVG funkcija radi isto, samo što vraća prosječnu vrijednost tog stupca. Kod obje funkcije tip podatka mora biti numerički kako bi se vrijednosti mogle zbrajati i dijeliti.

Primjer na slici 21 kao rezultat daje ukupnu cijenu usluge za svaki predmet posebno. Funkcija SUM u ovom primjeru zbraja sve vrijednosti stupca Cijena_usluge, pri čemu je pomoću GROUP BY klauzule određeno da se sumira vrijednost zasebno za svaki predmet.

```
SELECT K.Predmet, SUM(R.Cijena_usluge) AS Ukupna_cijena  
FROM Korisnici AS K RIGHT JOIN Racun AS R ON K.Sifra_korisnika = R.Korisnik  
GROUP BY Predmet;
```

Slika 21: Funkcija SUM()

Slika 22 prikazuje isti upit kao i slika 21, ali umjesto sumiranja imamo funkciju AVG, koja u ovom slučaju vraća prosječne cijene usluge za svaki predmet posebno.

```
SELECT K.Predmet, AVG(R.Cijena_usluge) AS Prosječna_cijena  
FROM Korisnici AS K RIGHT JOIN Racun AS R ON K.Sifra_korisnika = R.Korisnik  
GROUP BY Predmet;
```

Slika 22: Funkcija AVG()

Također postoje i funkcije MIN i MAX, koje vraćaju najmanju, odnosno najveću vrijednost u zadanom stupcu. Rezultat koji vraćaju je istog tipa kao i sam stupac. Obje funkcije zanemaruju pojavljivanje NULL znaka. Pomoću ovih funkcija, odnosno njihovom kombinacijom može se odrediti raspon vrijednosti, kao što je to u primjeru, za stupac Cijena_usluge.

```
SELECT MIN(Cijena_usluge) AS MIN, MAX(Cijena_usluge) AS MAX  
FROM Racun;
```

Slika 23: Funkcije MIN() i MAX()

FIRST() i LAST() funkcije vraćaju prvu, odnosno posljednju vrijednost zapisanu u odabranom stupcu. U sljedećem primjeru funkcija FIRST će vratiti ime i prezime korisnika koji se nalazi u prvom redu, odnosno koji je prvi zapisan u tablici Korisnici.

```
SELECT FIRST(Ime+" "+Prezime) AS Prvi_korisnik FROM Korisnici;
```

Jedan primjer kako bi ove funkcije mogle poslužiti u bazi podataka u realnom svijetu je ako bi došlo do problema s transakcijom, odnosno plaćanjem računa, funkcija LAST mogla bi nam na brz način dati informaciju o posljednjem izdanom računu, odnosno informacijama o transakciji koja nije uspjela.

Access podržava i funkcije za izračun varijance i standardne devijacije. Varijanca je aritmetička sredina kvadrata odstupanja vrijednosti numeričkog obilježja od njegove aritmetičke sredine. Standardna devijacija je drugi korijen iz varijance, a interpretira se kao prosječno odstupanje od prosječne vrijednosti. (Kero, Dobša i Bojanić-Glavica, 2008)

VarP i StDevP procjenjuju vrijednosti populacije, a Var i StDev vrijednosti uzorka populacije. Ako imamo manje od 2 zapisa navedene funkcije vraćaju NULL vrijednost, odnosno nije moguće izračunati vrijednost varijance i standardne devijacije.

Kao primjer imamo izračun varijance i standardne devijacije ocjena, odnosno povratnih informacija korisnika o rezultatu nakon instrukcija. Vrijednosti rezultata su male jer korisnici ostvaruju slične rezultate, odnosno odstupanja od prosječne ocjene nisu velika.

```
SELECT Var(Ocjena) AS Varijanca, StDev(Ocjena) AS Standardna_devijacija  
FROM Povratne_informacije;
```

Slika 24: Varijanca i standardna devijacija

Varijanca	Standardna_devijacija
0,285714285714286	0,534522483824849

Slika 25: Rezultat funkcija Var() i StDev()

6.5. Podupiti

Podupit je upit koji je sadržan unutar drugog upita, nalazi se u zagradama radi lakšeg prepoznavanja. Kod takve vrste upita prvo se provodi podupit čiji rezultat tada koristi glavni (vanjski) upit kao kriterij, izraz, parametar i slično.

Podupit mora minimalno imati SELECT naredbu i FROM klauzulu, mora biti u zagradama i može vraćati samo jednu vrijednost. Ako imamo iste tablice u upitu i podupitu moramo im dati različita imena pomoću AS.

```
SELECT Sifra_korisnika, Ime, Prezime  
FROM Korisnici  
WHERE Sifra_korisnika IN (  
SELECT Korisnik  
FROM Racun  
WHERE Cijena_usluge BETWEEN 180 AND 500);
```

Slika 26: Podupit

Na slici 26 imamo primjer podupita iz baze podataka. Taj podupit prvo daje kao rezultat vrijednost stupca Korisnik iz tablice Racun za koje vrijedi da je Cijena_usluge između 180,00 i 500,00 kuna. Nakon toga vanjski upit uzima taj rezultat i ispisuje vrijednosti stupaca Sifra_korisnika, Ime i Prezime za onu Sifru_korisnika iz tablice Korisnici koja je jednaka vraćenom rezultatu podupita.

7. Zaključak

Internet je pun informacija o Microsoft Access-u i SQL-u, te ih je u ovom radu trebalo ujediniti. Za provedbu praktičnog dijela najvažnije je bilo razlikovati SQL sustava Microsoft Access 2016 i SQL u drugim sustavima.

Sustav Microsoft Access 2016 je jednostavan sustav, prilagodljiv različitim korisnicima, te ga je lako shvatiti posebno s uputama i informacijama koje pruža sam Microsoft na službenim Internet stranicama. SQL se također može naučiti korištenjem izvora s interneta i konstantnim vježbanjem te testiranjem iskaza. W3Schools pruža jako kvalitetne smjernice za rad sa SQL-om, te svaku naredbu potkrepljuje primjerima. Još jedna stranica preko koje se može naučiti velik dio SQL-a je Codecademy. Stranica nudi besplatne tečajeve za početnike, a uz plaćanje su omogućeni napredni tečajevima za takozvane stručnjake. Osim tečaja u kojem korisnik sam testira pisanje koda, korisnik može provjeriti svoje trenutno znanje preko kviza.

Ovaj sustav može se koristiti u svakodnevnom poslovanju i vođenju zapisa. Olakšava rad organizacijama koje svakodnevno upravljaju i koriste veliku količinu podataka. Organizira podatke u cjelinu, omogućuje brz pronalazak potrebnih informacija, te lako ažuriranje trenutnih podataka. Daje organizacijama veću sigurnost i kontrolu nad pristupom podacima.

Popis literature

1. Alexander, M., i Kusleika, D. (2016). *Access 2016 bible*. Indianapolis: John Wiley & Sons.
2. Codecademy (bez dat.) *Learn SQL - Manipulation*. Preuzeto s <https://www.codecademy.com/learn/learn-sql>
3. Kaluža, M. (2008). *Sustavi baza podataka*. Preuzeto 12.09.2018. s https://www.veleri.hr/files/datotekep/nastavni_materijali/k_informatika_1/Predavanja_skripta.pdf
4. Kero, K, Dobša, J., i Bojanić-Glavica, B. (2008). *Statistika deskriptivna i inferencijalna i vjerojatnost*. Varaždin: Tiskara Varteks: Fakultet organizacije i informatike.
5. Rabuzin, K. (2014). *SQL : napredne teme*. Varaždin: Fakultet organizacije i informatike.
6. Rabuzin, K. (2011). *Uvod u SQL*. Varaždin: Fakultet organizacije i informatike.
7. Larsen, A. G. (2005). Implementing the Equivalent of a FULL OUTER JOIN in Microsoft Access. Preuzeto 12.09.2018. s <https://www.databasejournal.com/features/msaccess/article.php/3516561/Implementing-the-Equivalent-of-a-FULL-OUTER-JOIN-in-Microsoft-Access.htm>
8. Maleković, M., i Rabuzin, K. (2016). *Uvod u baze podataka*. Varaždin: Fakultet organizacije i informatike.
9. Microsoft (bez dat.) Data types for Access desktop databases. Preuzeto 25.08.2018. s <https://support.office.com/en-ie/article/data-types-for-access-desktop-databases-df2b83ba-cef6-436d-b679-3418f622e482>
10. Microsoft (2017). Insert, Update, and Delete Records From a Table Using Access SQL. Preuzeto 27.08.2018. s <https://docs.microsoft.com/en-us/office/vba/access/concepts/structured-query-language/insert-update-and-delete-records-from-a-table-using-access-sql>
11. Microsoft (bez dat.) *Learn the structure of an Access database*. Preuzeto 21.08.2018. s <https://support.office.com/en-us/article/learn-the-structure-of-an-access-database-001a5c05-3fea-48f1-90a0-cccaa57ba4af>
12. Microsoft (bez dat.) *Naredba TRANSFORM*. Preuzeto 31.08.2018. s <https://support.office.com/hr-hr/article/naredba-transform-c3a2361a-9f3f-4213-bc0a-65ed70031619>
13. Microsoft (bez dat.). *Tablica operatora*. Preuzeto 31.08.2018. s <https://support.office.com/hr-hr/article/tablica-operatora-e1bc04d5-8b76-429f-a252-e9223117d6bd>

14. Microsoft (bez dat.). *What happened to user-level security?* Preuzeto 12.09.2018. s <https://support.office.com/en-us/article/what-happened-to-user-level-security-69b362cd-503f-4e8a-a735-fb923ffb9fa3?ui=en-US&rs=en-US&ad=US>
15. NevVB (bez dat.). *Microsoft Access Database Version History*. Preuzeto 11.09.2018. s <https://www.nevnb.com.au/Database-Access-Versions.aspx>
16. Vazler, I., i Đumić, M. (2014-15). *Osnove baza podataka - Tipovi podataka*. Preuzeto 25.03.2018. s <https://www.mathos.unios.hr/~ivazler/obp/obp-3-tipovi.pdf>
17. Vuk, D., i Ciriković, E. (2015). *Priručnik za laboratorijske vježbe iz baza podataka* Preuzeto 21.08.2018. s <https://vsmti.hr/wp-content/uploads/2017/11/BP-Prirucnik-za-lab-vj-v1.01c.pdf>
18. W3Schools (bez dat.). *Introduction to SQL*. Preuzeto s https://www.w3schools.com/sql/sql_intro.asp

Popis slika

Slika 1: Izgled Microsoft Accessa 2016	4
Slika 2: Pogreška kod presjeka 1.	6
Slika 3: Pogreška kod presjeka 2.	6
Slika 4: Odnosi u bazi podataka	8
Slika 5: Kreiranje tablice	11
Slika 6: Struktura tablice	13
Slika 7: Dodavanje podataka - red po red.....	13
Slika 8: Dodavanje podataka - više redova.....	14
Slika 9: Unakrsni upit.....	15
Slika 10: Opcija DISTINCT	15
Slika 11: Grupiranje redova	16
Slika 12: Spajanje klauzulama FROM i WHERE.....	17
Slika 13: Spajanje klauzulom JOIN.....	17
Slika 14: RIGHT JOIN klauzula	18
Slika 15: Unija	19
Slika 16: Rezultat unije	19
Slika 17: Aritmetički operatori	20
Slika 18: Operatori IN i BETWEEN	22
Slika 19: Primjer operatora LIKE	23
Slika 20: Funkcija COUNT()	24
Slika 21: Funkcija SUM()	24
Slika 22: Funkcija AVG()	25
Slika 23: Funkcije MIN() i MAX().....	25
Slika 24: Varijanca i standardna devijacija.....	26
Slika 25: Rezultat funkcija Var() i StDev()	26
Slika 26: Podupit	26

Popis tablica

Tablica 1: Verzije sustava Access	2
Tablica 2: Aritmetički operatori	20
Tablica 3: Operatori uspoređivanja	21
Tablica 4: Operator LIKE	23