

UNIVERZITET MB
Poslovni i pravni fakultet
Beograd

SEMINARSKI RAD

FORMIRANJE BAZE PODATAKA

Predmet: Baze podataka

Mentor:

Dr Borivoje Milošević

Student:

SADRŽAJ

1. UVOD	3
2. MICROSOFT ACCESS	4
3. OSNOVNI ELEMENTI KREIRANJA BAZE U ACCESU.....	4
4. OTVARANJE BAZE I IZRADA TABELA	5
I DEO.....	5
5. ORGANIZACIJA BAZE U MS ACCESS_U:.....	7
5.1.1 PLANIRANJE TABELA	8
6. POVEZIVANJE TABELA	16
6.1 KEYS - KLJUČEVİ	17
6.1.1 PRIMARNI KLJUČ:.....	17
6.1.2 SEKUNDARNI - STRANI KLJUČ:	17
6.1.3 PREDNOSTI KORIŠTENJA RELACIJA	19
7. UNOŠENJE PRVIH ZAPISA	20
7.1 USPOSTAVLJANJE RELACIJA VRŠIMO VEĆ OPISANIM POSTUPKOM:.....	22
8. UPITI	23
9. FORME.....	26
10. IZVEŠTAJI	28
11. MAKRO PROGRAM	30
12. BAZA PODATAKA – REALIZACIJA PREKO SQL DRUGI DEO	32
13. POPUNJAVANJE TABELA	40
14. IZVODJENJE RELACIJA	43
15. UPITI NAD BAZOM PODATAKA	48
16. NEKA OBJAŠNJENJA O TOME KAKO VAMP SERVER FUNKCIIONIŠE:.....	55
17. ZAKLJUČAK.....	64

1. UVOD

Baze podataka su svojevrsne organizovane "elektronske kartice", kolekcija informacija vezanih za određeni subjekt, namenu ili pojavu, kao što su praćenje narudžbina kupaca, sportskog kluba, vođenje stanja magacina ili održavanje muzičke kolekcije u prividnom tabelarnom izgledu. Bazu podataka ne čine samo organizovane tabele kao nosioci podataka, već je to kompletan program koji može proračunavati, filtrirati podatke, štampati izveštaje itd. Programe sličnog tipa možemo najčešće sresti u preduzećima, bankama, poštama, arhivama, i praktično rečeno - svuda.

Brigu o podacima, vezama među podacima, ispravnosti podataka isvemu ostalom na sebe preuzima sistem za upravljanje podacima u bazi podataka (DBMS – Database Management Sistem). Svi moderni sistemi zaupravljanje bazama podataka koriste relacioni model podataka što znači da korisnik vidi bazu kao skup tabele sa sloganima, gde je i rezultat svake operacije nad sadržajem baze podataka takođe tabela. Pored toga relacioni model koristi veze, odnosno relacije (relationships) za povezivanje datoteka(tabela) baze podataka u strukturiranu celinu.

Microsoft Access je sistem, odnosno program za upravljanje bazama podataka. Access omogućuje skupljanje, slaganje i organizovanje podataka kao stvaranje izveštaja koji vode do poslovnih odluka. Glavne funkcionalne mogućnosti Access-a su sledeće:

- Unos podataka direktno u bazu podataka ili uvoz podataka iz drugih programa.
- Sortiranje, indeksiranje i organizacija podataka na način koji se želi.
- Brzo stvaranje izveštaja ili priprema ispis nalepnica za pisma upotrebotom svih ili samo dela podataka.

U ovom radu u Microsoft Access-u odrađena je baza podataka pod nazivom "Servis računara, televizora, video i audio uređaja". Osnovna namena ove baze je da čuva podatke o klijentima koji su doneli uređaje na servis, podatke o servisiranim udeđajima, cenu servisa, kao i cenu i katalog delova koji su dostavljeni od dobavljača servisu.

2. MICROSOFT ACCESS

Access je sistem za upravljanje bazama podataka. Koristi se za skladištenje i učitavanje podatke, prikazivanje, a takođe i za automatizaciju poslova koji se često ponavljaju. Predstavlja odličnu kombinaciju produktivnosti programa za upravljanje bazom podataka i lakoće korišćenja Windows-a. Kao zaokružena aplikacija, Access obezbeđuje potpuno upravljanje relacionom bazom podataka i integritet na nivou same mašine na kojoj se nalazi baza. Tabelama je moguće pridružiti pravila koja proveravaju ispravnost i validnost podataka i sprečavaju unos neispravnih, a svako polje tabele može imati svoj format i podrazumevanu vrednost što znatno povećava produktivnost.

Access 2010 omogućava korišćenje svojih aplikacija za rad na Internetu. Izborom određene opcije omogućava se prebacivanje tabela, upita, izveštaja i prikaza u tabelarnom obliku u HTML format. Pomoću Data Access Pages mogu da se prave složeni obrazci na Internetu koji su povezani sa podacima u konkretnoj mreži. Za razliku od drugih programa za rad sa bazama podataka kod kojih se svaka tabela nalazi u posebnoj datoteci, kompletna Access baza se nalazi u tačno jednoj datoteci. Tabele, upiti, formulari, izveštaji, stranice za rad sa podacima, makroi i moduli zajedno čine Accessovu bazu podataka. Svaki od ovih objekata ima specijalnu namenu. Accessova aplikacija takođe sadrži nekoliko različitih objekata, uključujući zavisnosti, komandne linije, svojstva baze podataka i uvozne/izvozne specifikacije. Pomoću ovih objekata možemo napraviti moćne, prijateljske, integrisane aplikacije.

3. OSNOVNI ELEMENTI KREIRANJA BAZE U ACCESU

Baza podataka kreirana u Access-u sastoji se od sledećih elemenata odnosno objekata:

- tabele
- forme
- upiti
- izveštaji
- makroi

Pored osnovnih elemenata Access baza može da sadrži i Data Access strane, makroe i/ili VBA module (Visual Basic for Application - programski jezik za programiranje operacija u Office aplikacijama). Svrha ovih elemenata je nadogradnja osnovnog oblika baze u cilju maksimalne moguće prilagodjenosti baze ka njenom korisniku i ispunjenja posebnih zahteva korisnika.

4. OTVARANJE BAZE I IZRADA TABELA

I deo

Izgradnja kuće počiva od temelja. Izgradnje baze počinje od osmišljavanja tabela i njihovog povezivanja. Osnovni elemenat svake baze podataka je tabela. Svaka tabela sarži kolone i redove a u njihovom preseku nalazi se polje ili ćelija u koje Accesssmešta karakterističan podatak. Red tabele predstavlja jedan slog koji sasvojim poljima daje grupu podataka čiji je karakter određen kolonamatabele. Definisanjem kolone, odnosno karaktera podatka koji će se slagati u koloni, polja u koloni se definišu i svrstavaju u određen tip, koji može biti:tekst, broj, datum/vreme, logičko polje, memo polje (polje u koga se može uneti tekst proizvoljne veličine), polje za unos brojeva u novčanom formatu, posebno polje u kome se automatski po započetom unosu sloga inkrementira broj (polje AutoNumber), hiperlink polje (pokazivač na adresu na Internetu), OLE object polje i Look Up polje.

Za potrebe izrade baze “Servis računara, televizora, video i audio uređaja” za početak ćemo osmotriti celine - podatke koje treba grupisati u zasebnim tabelama. Na osnovu projekta koji je sastavila grupa za planiranje i projektovanje baze, baza će sadžati šest tabela i to: **klijenti, serviser, uredjaji, delovi, dobaljaci i delovi-dobavljac**. U projektu su na osnovu korisničkih zahteva date i relacije izmedju tabela, dati primarni i sekundarni ključevi koji imaju za zadatak da povežu tabele i omoguće brže i tačnije prikazivanje izveštaja - **Report**.

Klijenti	
	Field Name
ID_klijenta	
Ime	
Prezime	
Adresa	
Telefon	

Tabela “klijenti”

serviser	
	Field Name
ID_servisera	
Prezime	
Ime	
RadniSat	

Tabela “serviser”

Field Name
ID_uredjaja
Naziv
Tip
Proizvodjac
ID_klijenta
Opis problema
DatumDostavke
DatumUgradnje
DatumZavrsetka
VremeRada
Kolicina
ID_servisera

Tabela "uredjaji"

Field Name
ID_dobavljača
Naziv dobavljača
Adresa
Telefon
e-mail
Grad

Tabela "dobavljadi"

Field Name
ID_dela
Naziv dela
Opis dela
Cena dela
ID_uredjaja

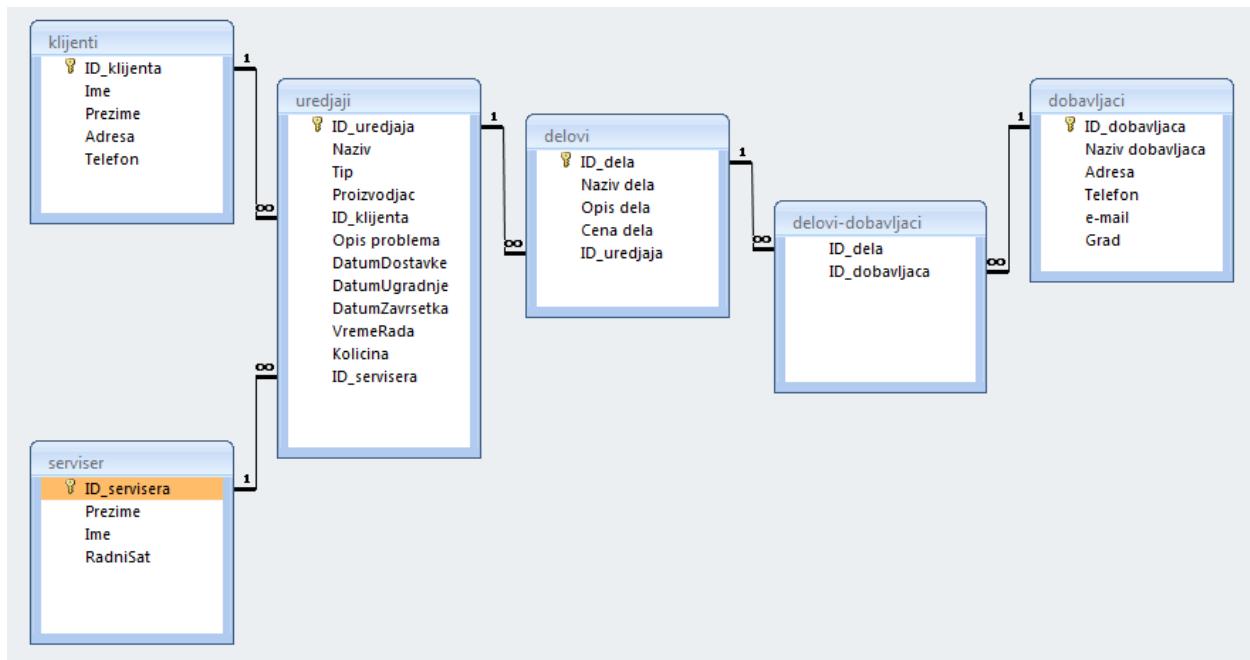
Tabela "delovi"

Field Name
ID_dela
ID_dobavljača

Tabela "delovi-dobavljadi"

U projektu su prikazane sve tabele koje treba da čine bazu podataka, njihovi atributi, veze i primarni i sekundarni ključevi. Razmotrićemo sve postavljene relacije !!!

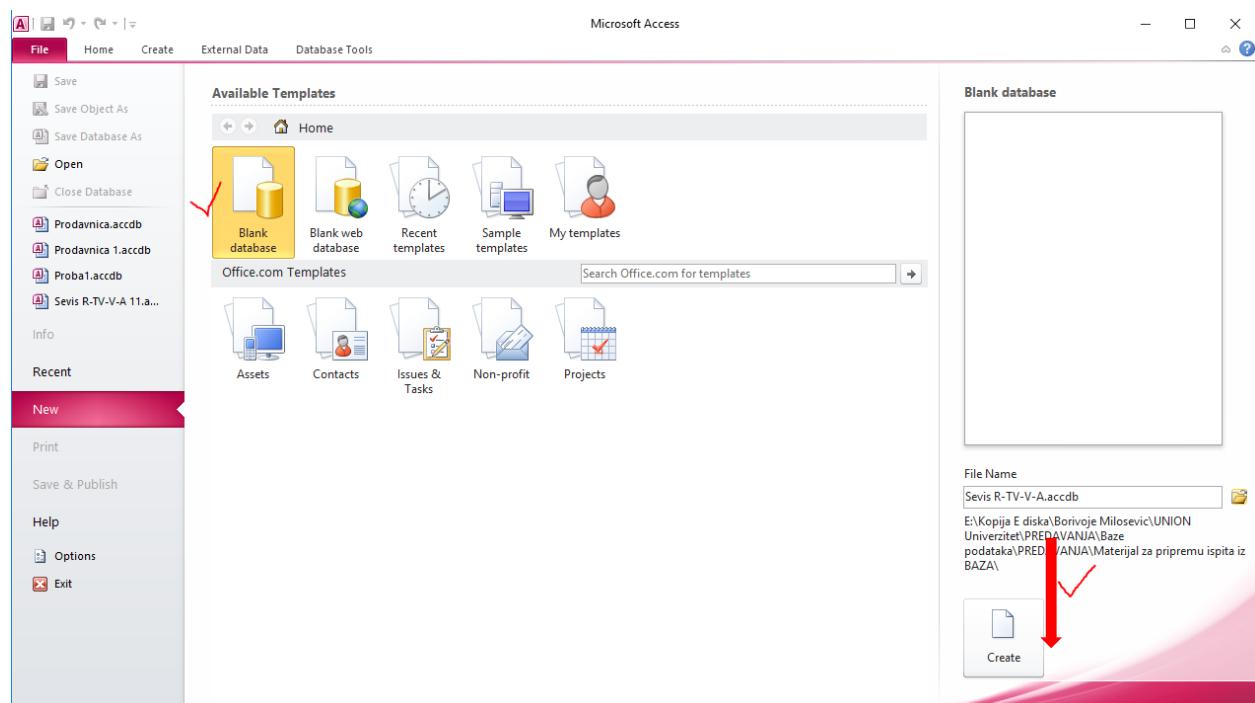
U ovom trenutku programer baze podataka poseduje sve potrebne elemente pa može sastaviti konceptualni model baze podataka koji će mu pomoći da lakše sagleda organizaciju baze i pristupi njenom programiranju.



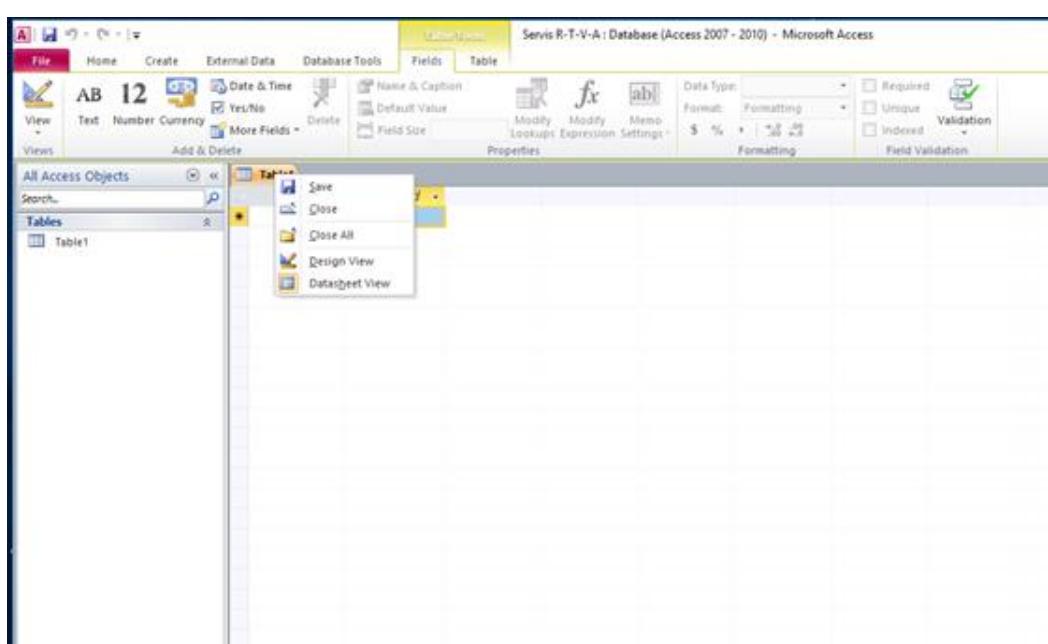
5. ORGANIZACIJA BAZE U MS ACCESS_U:

Otvorićemo bazu podataka MS Access kao **Blank database** koja će ćemo zapamtiti na tekućem direktorijumu (ili na onom koga odaberemo) i u polje uneti naziv baze:

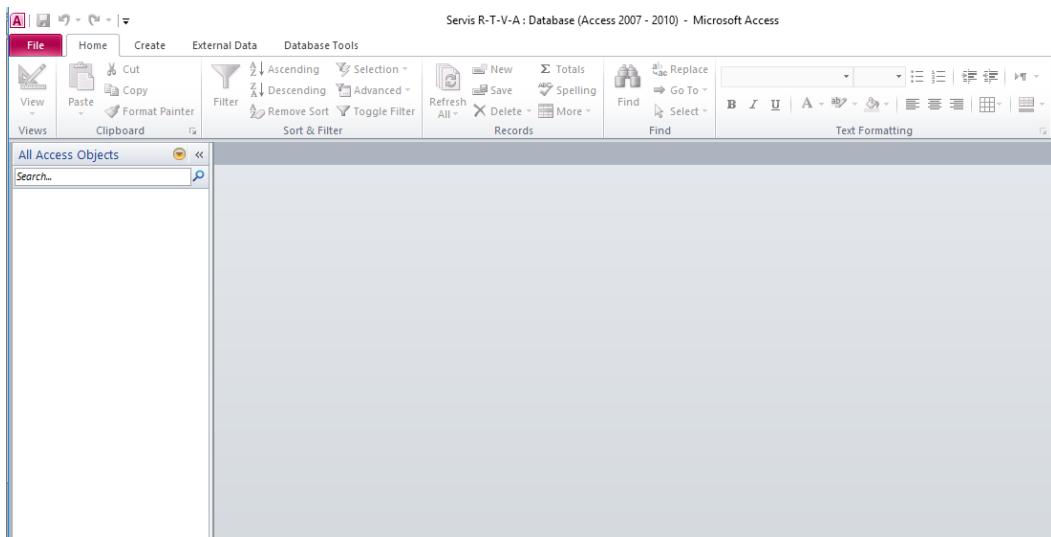
Servis R-T-V-A



Zatvorićemo ponudjenu tabelu Table1, desnom tasterom na njen jezičak i izborom iz padajućeg menija opcije Close.



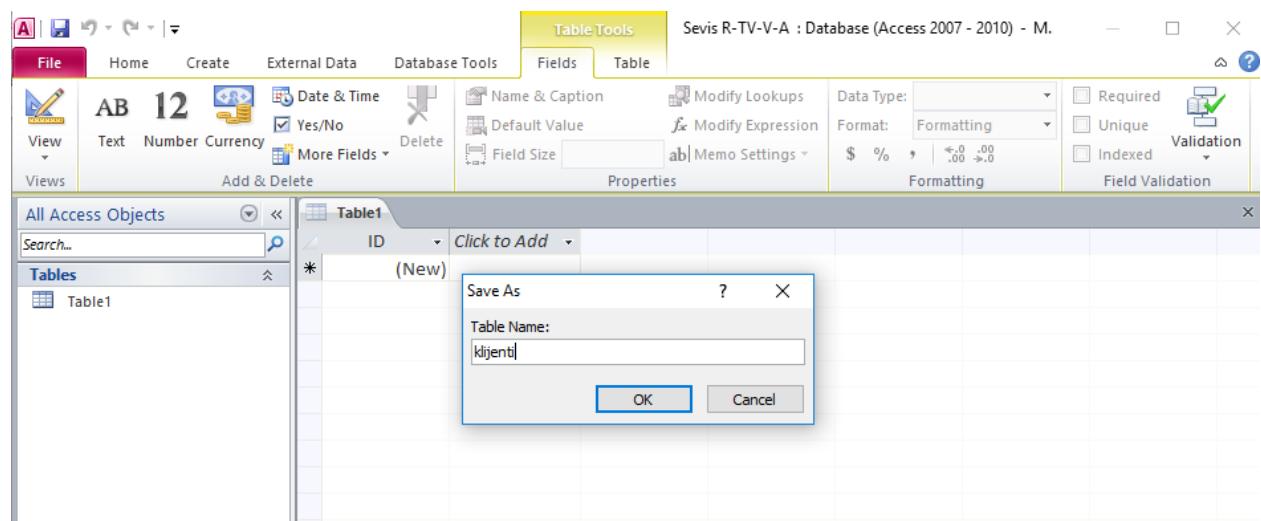
Sada nam se otvara prazan prostor za rad sa našom bazom i njenu organizaciju:

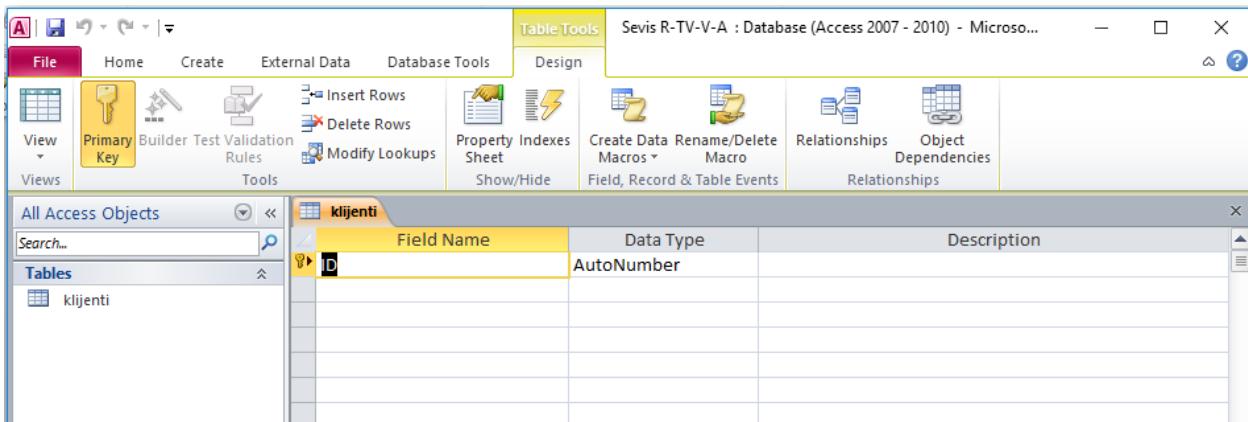


5.1.1 PLANIRANJE TABELA

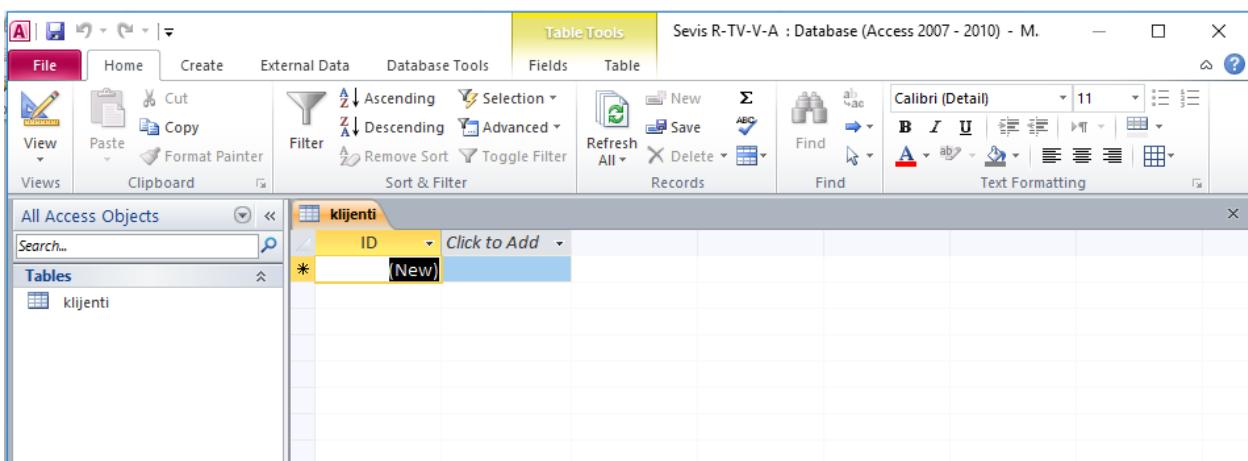
Pošto smo stvorili sliku o potrebnom broju tabela i njihovoj nameni, možemo preći na sledeći korak - **definisanje tabela**. Za svaku tabelu valja definisati polja i adekvatne tipove podataka za atributе koji se mogu u njih unositi.

Iz opcije glavnog menija **VIEW** izabraćemo iz padajućeg menija opciju **Design View** i u prozoru otkucati ime prve tabele: **klijenti**.





U padajućem meniju VIEW čekiramo opciju Datasheet View gde se u glavnom prozoru otvaraju polja za definisanje prvog (i ostalih) atributa **ID_klijenta** kao primarnog ključa tabele klijenti:



Smatramo da na osnovu projekta tabela klijenti mora da ima sledeće attribute: **ID_klijenta, Ime, Prezime, Adresa, Telefon.**

Tada u poljima View/Design View definišemo parametre prvog atributa **ID_klijenta** koji se tiču **Data Type – int**, njegove veličine polja - **Field Size (Long integer)**, u ovom slučaju smatramo da bi ovaj atribut mogao da ima primarni ključ.

NAPOMENA: Primarni ključ u ovom slučaju mogao bi da bode npr. Matični broj klijenta

The screenshot shows the Microsoft Access 2010 interface. The 'Table Tools Design' ribbon tab is active. A table named 'klijenti' is selected. In the datasheet, the first column 'ID_klijenta' is highlighted. The 'Field Properties' pane is open, showing the 'General' tab with settings for 'Field Size' (Long Integer) and 'New Values' (Increment). A tooltip provides information about generating values for the field.

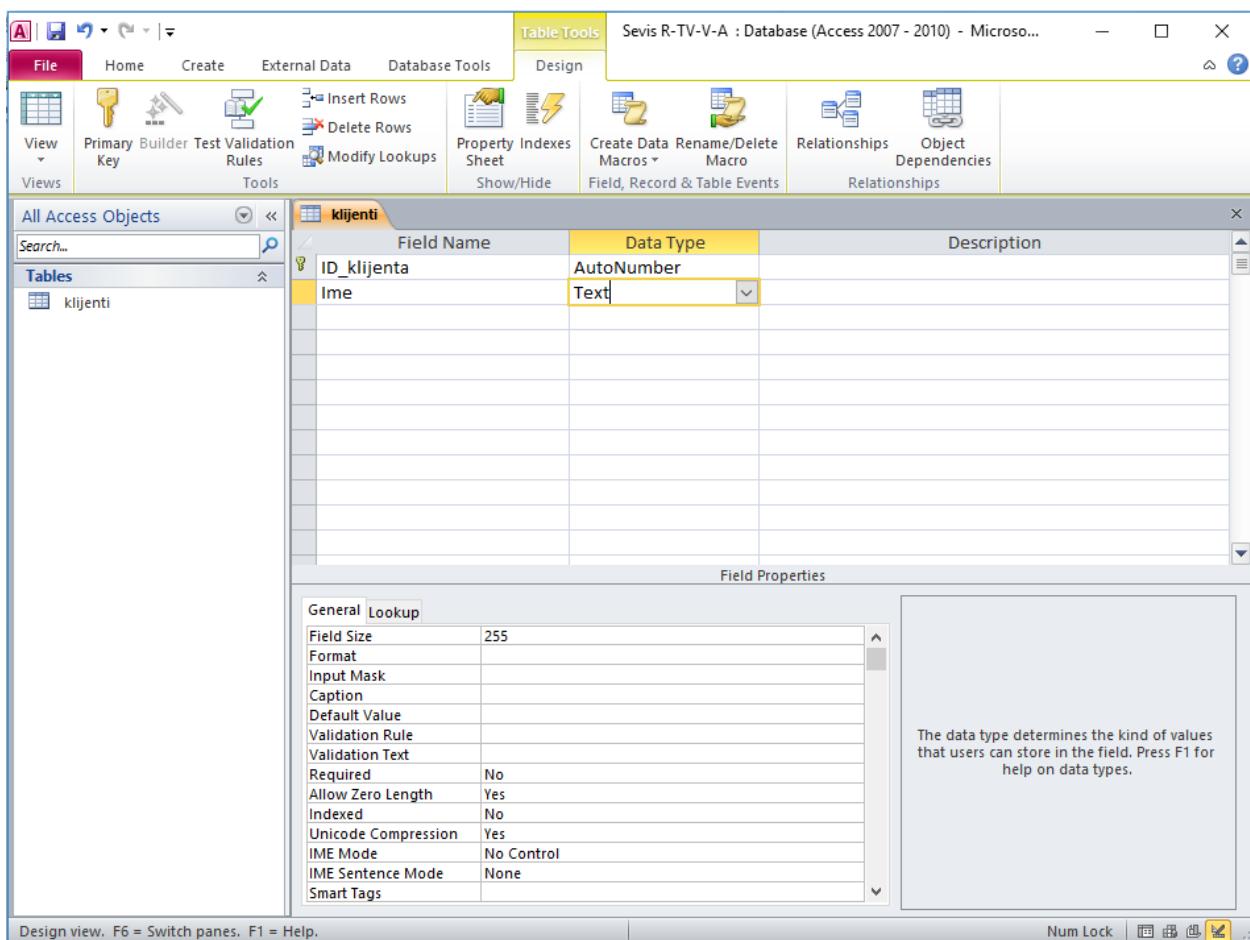
Kada definišemo sve, za bazu potrebne podatke atributa **ID_klijenta** prelazimo na definisanje ostalih atributa, u ovom slučaju **Ime**:

Vraćamo se na opciju glavnog menija **VIEW/Datasheet view** i u polju **Click to Add** dodajemo novi atribut

The screenshot shows the Microsoft Access 2010 interface with the 'View' ribbon tab selected. The 'Datasheet View' mode is active. A table named 'klijenti' is open in the datasheet. A new row is being added, indicated by the '(New)' placeholder in the second column. The ribbon tabs include 'File', 'Home', 'Create', 'External Data', 'Database Tools', 'Fields', and 'Table'.

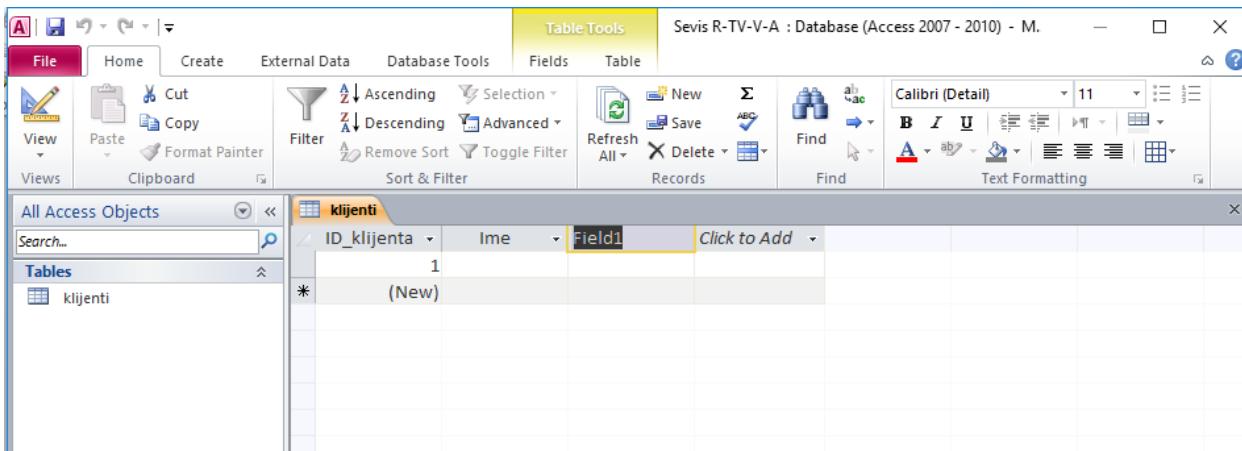
Vraćamo se na opciju glavnog menija **VIEW/Design View** i u polju **Field1** dodajemo naziv drugog atributa **Ime**. Odredujemo tip podatka **Data Type** iz padajućeg menija pa postavljamo da je to tekstualni podatak **Text**. U ostalim poljima za dodefinisanje odabiramo

veličinu polja **Field Size** (maksimalno do 255 karaktera), za zahtevani podatak **Required** odaberemo **NO**, dozvoljavamo nultu dužinu podatka **Allow Zero Length – Yes**, **Indexed – No**, kompresija karaktera **Unicode Compression – NO**, **IME Mode – No Control**, **IME Sentence Mode – None**.

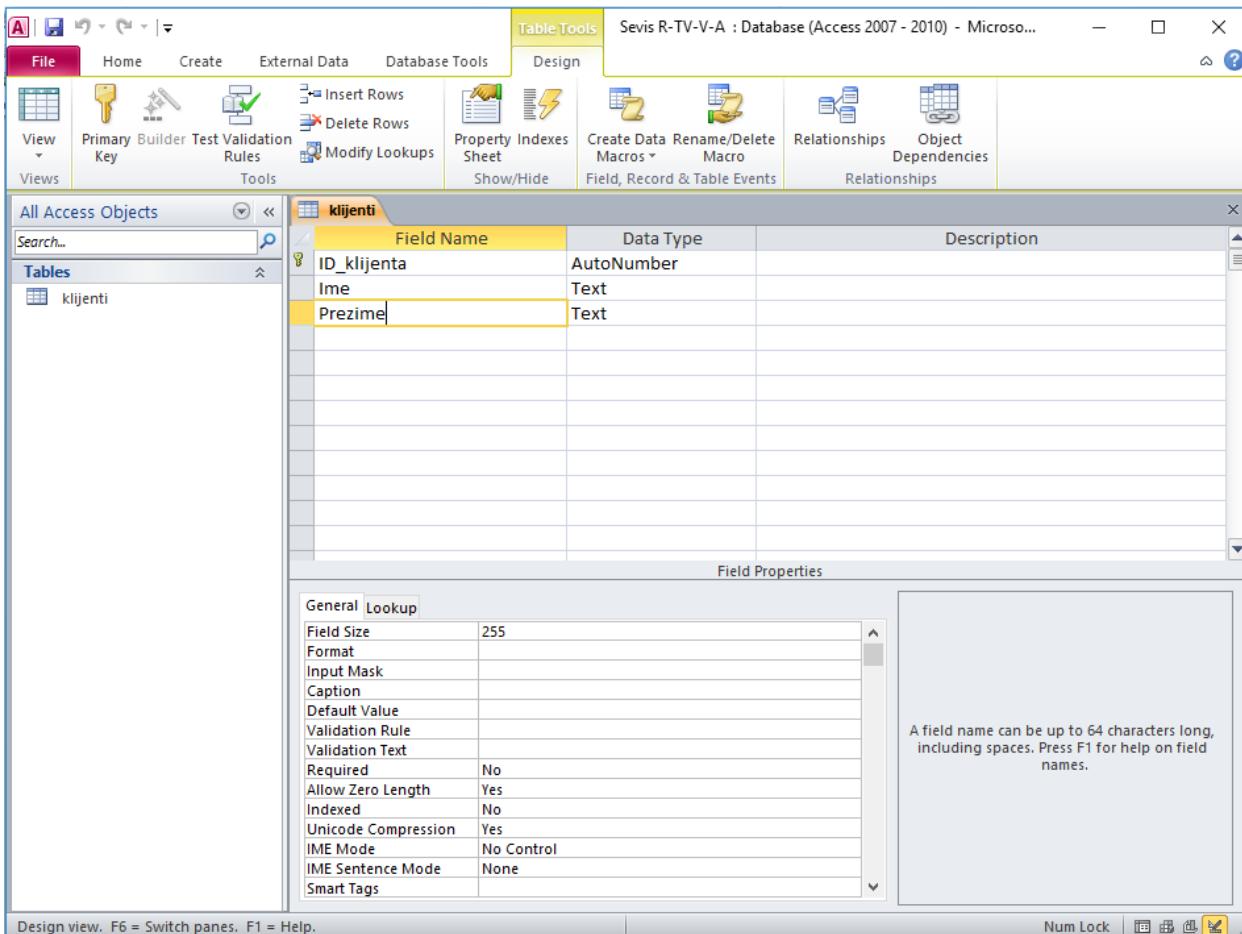


Kada definišemo sve, za bazu potrebne podatke atributa **Ime** prelazimo na definisanje ostalih atributa, u ovom slučaju **Prezme**:

Vraćamo se na opciju glavnog menija **VIEW/Datasheet view** i u polju **Click to Add** dodajemo novi atribut

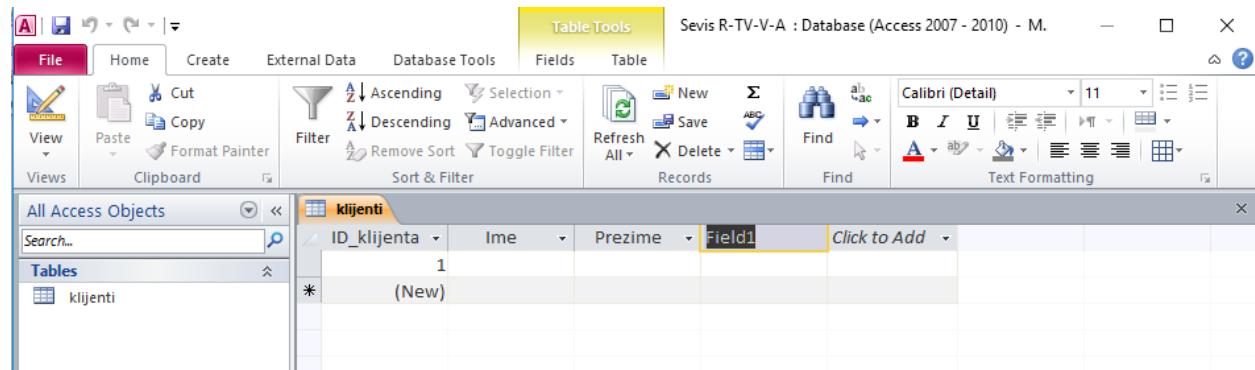


Vraćamo se na opciju glavnog menija **VIEW/Design View** i u polju polju **Field1** dodajemo naziv drugog atributa **Prezime**. Odredujemo tip podatka **Data Type** iz padajućeg menija pa postavljamo da je to tekstualni podatak **Text**. U ostalim poljima za dodefinisanje odabiramo veličinu polja **Field Size** (maksimalno do 255 karaktera), za zahtevani podatak **Required** odaberemo **NO**, dozvoljavamo nultu dužinu podatka **Allow Zero Length – Yes**, **Indexed – No**, kompresija karaktera **Unicode Compression – Yes**, **IME Mode – No Control**, **IME Sentence Mode – None**.

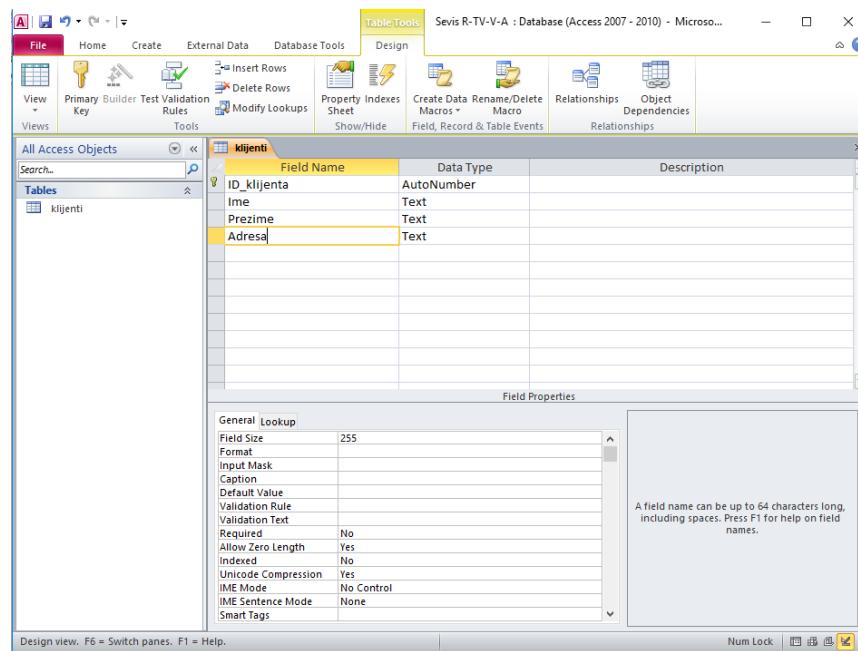


Kada definišemo sve, za bazu potrebne podatke atributa **Prezime** prelazimo na definisanje ostalih atributa, u ovom slučaju **Adresa**:

Vraćamo se na opciju glavnog menija **VIEW/Datasheet view** i u polju **Click to Add** dodajemo novi atribut

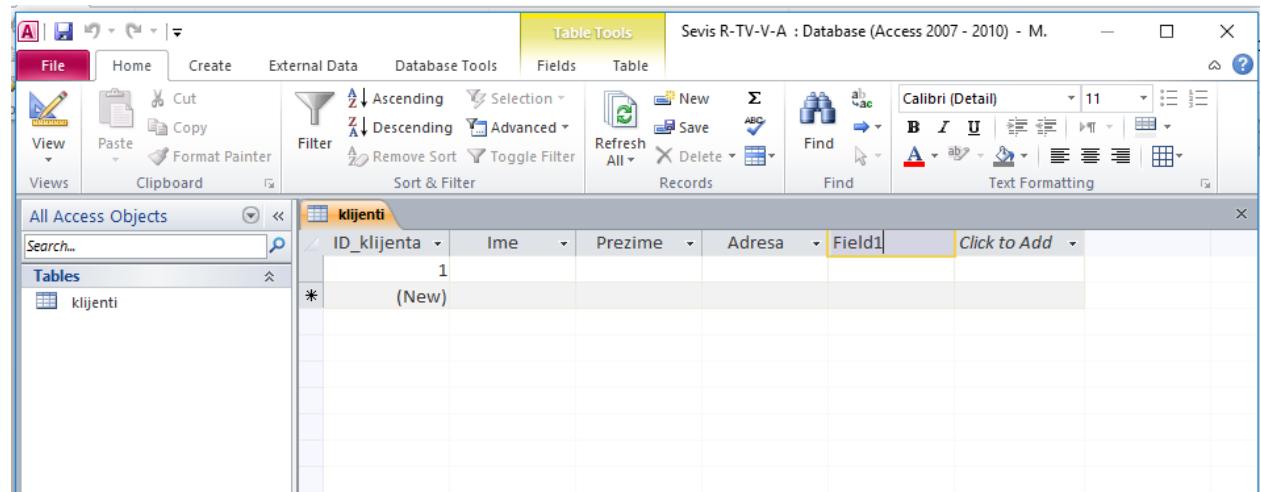


Vraćamo se na opciju glavnog menija **VIEW/Design View** i u polju polju **Field1** dodajemo naziv drugog atributa **Adresa**. Odredujemo tip podatka **Data Type** iz padajućeg menija pa postavljamo da je to tekstualni podatak **Text**. U ostalim poljima za dodefinisanje odabiramo veličinu polja **Field Size** (maksimalno do 255 karaktera), za zahtevani podatak **Required** odaberemo **NO**, dozvoljavamo nultu dužinu podatka **Allow Zero Length – Yes**, **Indexed – No**, kompresiju karaktera **Unicode Compression – Yes**, **IME Mode – No Control**, **IME Sentence Mode – None**.

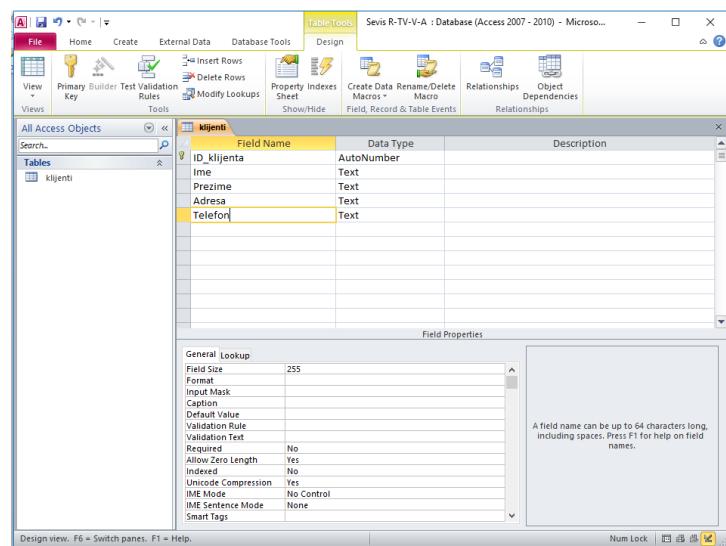


Kada definišemo sve, za bazu potrebne podatke atributa **Adresa** prelazimo na definisanje ostalih atributa, u ovom slučaju **Telefon**:

Vraćamo se na opciju glavnog menija **VIEW/Datasheet view** i u polju **Click to Add** dodajemo novi atribut.



Vraćamo se na opciju glavnog menija **VIEW/Design View** i u polju polju **Field1** dodajemo naziv drugog atributa **Telefon**. Odredujemo tip podatka **Data Type** iz padajućeg menija pa postavljamo da je to tekstualni podatak **Text**. U ostalim poljima za dodefinisanje odabiramo veličinu polja **Field Size** (maksimalno do 255 karaktera), za zahtevani podatak **Required** odaberemo **NO**, dozvoljavamo nultu dužinu podatka **Allow Zero Length – Yes**, **Indexed – No**, kompresija karaktera **Unicode Compression – Yes**, **IME Mode – No Control**, **IME Sentence Mode – None**.



Na kraju, definisali smo sve za bazu potrebne podatke tabele **klijenti** pa prelazimo na opciju glavnog menija **VIEW/Design view** i sagledavamo sve atribute tabele **klijenti**:

Tabela “*klijenti*”

Field Name	Data Type	Description
ID_klijenta	AutoNumber	
Ime	Text	
Prezime	Text	
Adresa	Text	
Telefon	Text	

Prateći ovaj postupak unećemo u bazu podataka i ostale tabele sa svojim osobinama i atributima:

Tabela “*serviser*”

Field Name	Data Type	Description
ID_servisera	AutoNumber	
Prezime	Text	
Ime	Text	
RadniSat	Number	

Tabela “*uredjaji*”

Field Name	Data Type	Description
ID_uredjaja	AutoNumber	
Naziv	Text	
Tip	Text	
Proizvodjac	Text	
ID_klijenta	Number	
Opis problema	Text	
DatumDostavke	Date/Time	
DatumUgradnje	Date/Time	
DatumZavrsetka	Date/Time	
VremeRada	Currency	
Kolicina	Number	
ID_servisera	Number	

Tabela "dobavljac"

Field Name	Data Type	Description
ID_dobavljaca	AutoNumber	
Naziv dobavljaca	Text	
Adresa	Text	
Telefon	Text	
e-mail	Hyperlink	
Grad	Text	

Tabela "delovi"

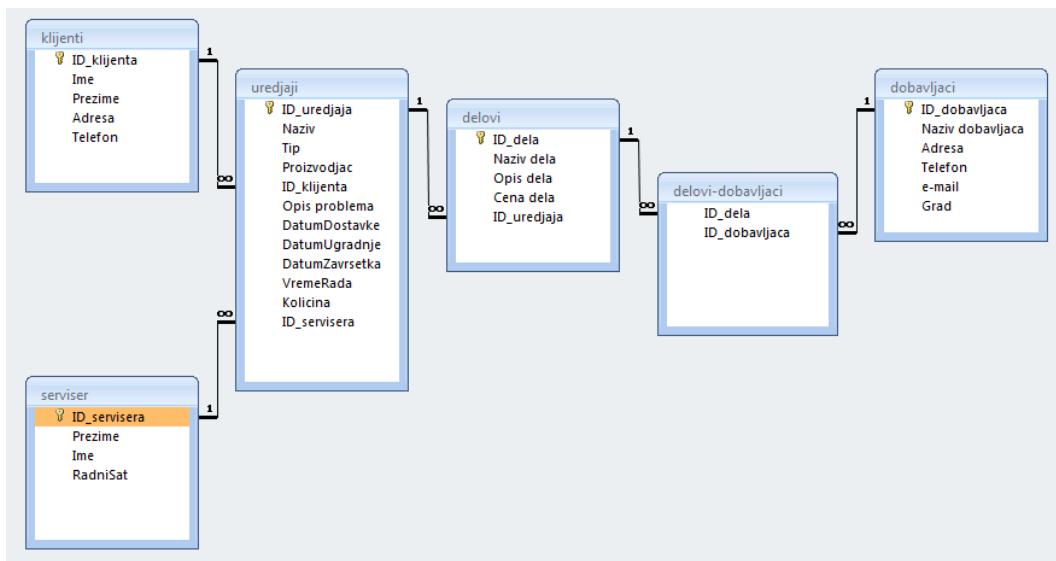
Field Name	Data Type	Description
ID_dela	AutoNumber	
Naziv dela	Text	
Opis dela	Text	
Cena dela	Currency	
ID_uredjaja	Number	

Tabela "delovi-dobavljaci"

Field Name	Data Type	Description
ID_dela	Number	
ID_dobavljaca	Number	

6. POVEZIVANJE TABELA

Iskoristićemo postojeće, kreirane tabele i povezaćemo ih međusobno u relacije. Punu snagu Access ostvaruje iz relacionih tabela (otuda naziv relacione baze podataka). Iz relacionih tabela podatke je lako filtrirati i na njih primeniti razne kriterijume bez obzira iz koliko tabela podaci potiču. Na slici 6.1. prikazan je prozor Relationship kada smo povezali sve tabele.



6.1. Prozor "Relationships"

Relacije izmadiju tabelu:

Iako svaka tabela pamti podatke o različitom subjektu (attribute), tabele u bazi podataka obično pamte podatke o atributima koji su medjusobno povezani. Na primer, baza podataka može sadržati

:

- ✓ Tabelu kupaca koja navodi kupce naše fabrike i njihove adrese.
- ✓ Tabela proizvoda koja sadrži proizvode koje prodajemo, uključujući cene i slike za svaku stavku.
- ✓ Tabela narudžbi koja prati narudžbe kupaca.

Budući da podatke o različitim subjektima pamtimo u zasebnim tabelama, trebamo na neki način povezati podatke kako bi lako mogli kombinovati povezane podatke iz tih zasebnih tabela. Da bi povezali podatke pohranjene u različitim tabelama, stvaramo relacije - odnose. Relacija je logička veza izmedju dve tabele koja određuje polja koja su zajednička u tim tabelama.

6.1 Keys - Ključevi

6.1.1 Primarni ključ:

Tabela može imati samo jedan primarni ključ. Primarni ključ sastoji se od jednog ili više polja koja jedinstveno identifikuju svaki zapis koji pamtimo u tabeli. Često postoji jedinstveni identifikacijski broj, poput ID broja, serijskog broja ili koda, koji služi kao primarni ključ. Na primer, možda imamo tabelu Kupci u kojoj svaki kupac ima jedinstveni ID broj kupca. Polje korisničkog ID-a primarni je ključ tabele Kupci. Kada primarni ključ sadrži više od jednog polja, obično se sastoji od već postojećih polja koja, zajedno uzeta, daju jedinstvene vrednosti. Na primer, kao primarni ključ za tabelu o ljudima možete koristiti kombinaciju prezimena, imena i datuma rođenja.

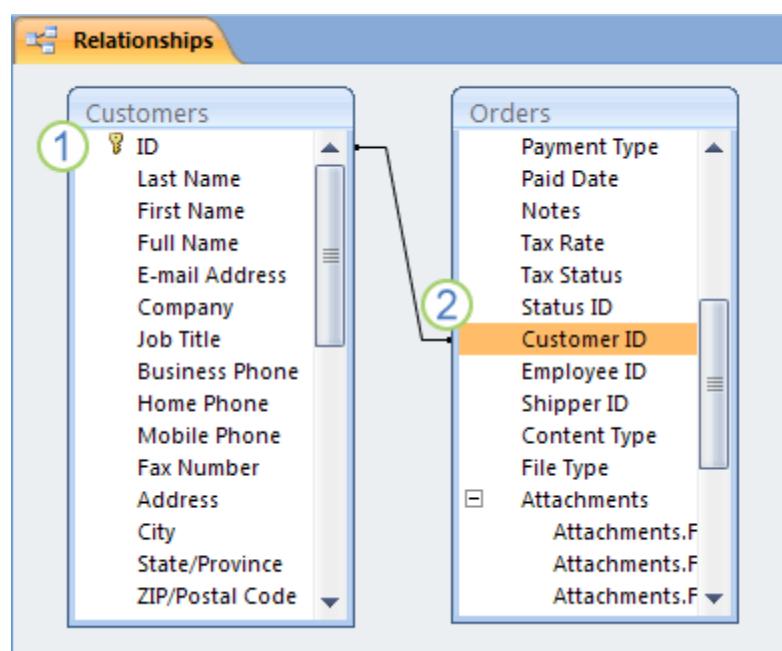
6.1.2 Sekundarni - Strani ključ:

Tabela takođe može imati jedan ili više stranih ključeva. Strani ključ sadrži vrednosti koje odgovaraju vrednostima u primarnom ključu druge tabele. Na primer, možda imamo tabelu

Narudžbe u kojoj svaka narudžba ima ID broja kupca koji odgovara zapisu u tabeli Kupci. Polje ID-a kupca sekundarni je ključ tabele Narudžbe.

Korespondencija vrednosti između ključnih polja čini osnovu odnosa tabela. Odnos tabela koristimo za kombinovanje podataka iz povezanih tabela. Na primer, pretpostavimo da imamo tabelu kupaca i tabelu narudžbi. U tabeli kupaca svaki je zapis identifikovan poljem primarnog ključa, ID.

Da bi svaku narudžbu povezali sa kupcem, u tabelu Narudžbe dodajemo polje stranog ključa koje odgovara polju ID tabele Kupci, a zatim stvaramo odnos izmedju dva ključa. Kada u tabeli Narudžbe dodamo zapis, koristimo vrednost za ID kupca koja dolazi iz tabele Kupci. Kad god želimo pregledati bilo kakve informacije o kupcu narudžbe, koristimo odnos da bi utvrdili koji podaci iz tabele Kupci odgovaraju kojim zapisima u tabeli Narudžbe.



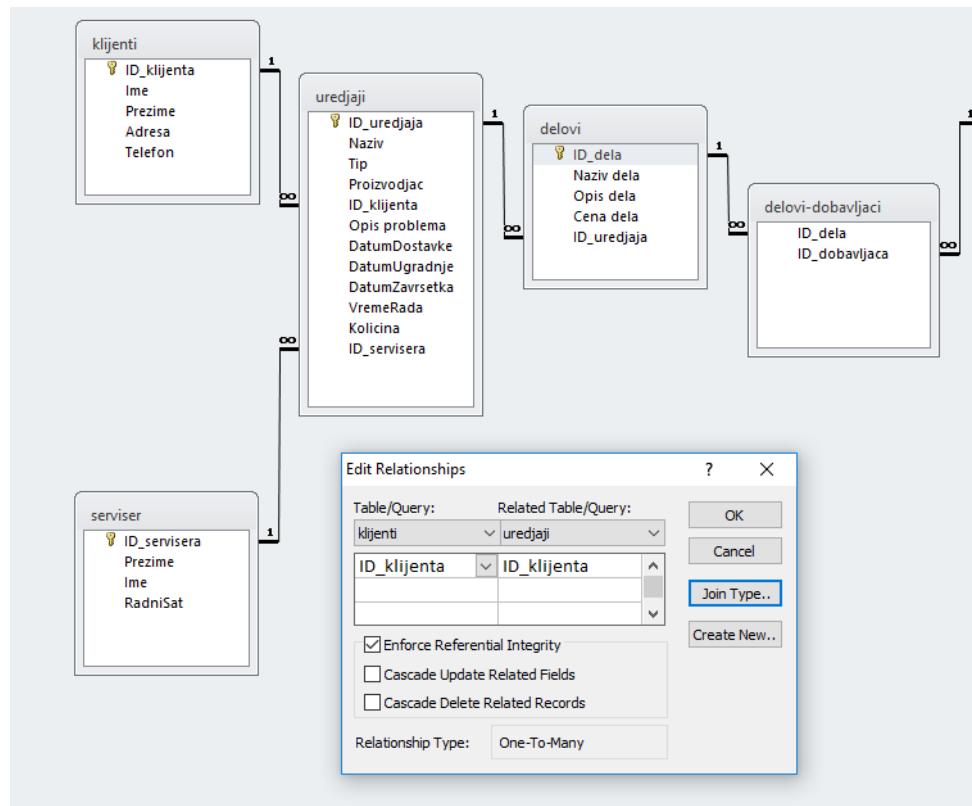
Relacija izmedju tabela prikazuje se u Relationships window.

- 1** Primarni ključ, identificuje ga ikonica ključa pored imane polja
- 2** Sekundarni ključ - A foreign key — imajte na umu odsutnost ikone ključa.

6.1.3 Prednosti korištenja relacija

Odvajanje podataka u povezane tablice donosi sledeće prednosti:

- ✓ **Doslednost:** Budući da se svaka stavka podataka beleži samo jednom, u jednoj tabeli manje je mogućnosti za dvosmislenost ili nedoslednost. Na primer, ime kupca pamtimo samo jednom, u tabelu o kupcima, umesto da ga više puta (i potencijalno nedosledno) pamtimo u tabeli koja sadrži podatke o narudžbi.
- ✓ **Raspoloživost:** Snimanje podataka na samo jednom mestu znači da koristimo manje prostora na disku. Štaviše, manje tabele obično pružaju podatke brže od većih tabela. Napokon, ako ne koristimo zasebne tabele za odvojene predmete, u svoje ćemo tabele uvesti null vrednosti (nedostatak podataka) i suvišnost, pa obe mogu izgubiti prostor i otežati izvodjenje.
- ✓ **Razumljivost:** Dizajn baze podataka lakše je razumeti ako su predmeti pravilno odvojeni u tabele.

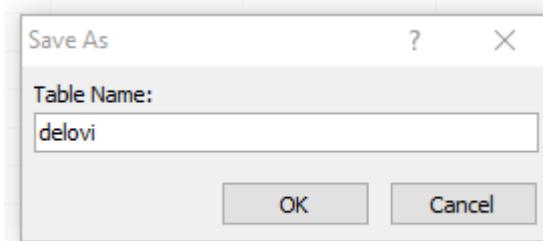
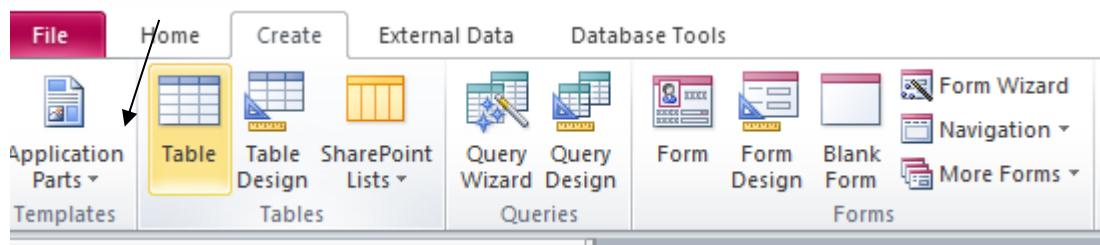


SAVET:

Planirajte svoje tabele imajući na umu odnose. Pomoću čarobnjaka za pretraživanje možemo stvoriti polje stranog ključa ako tabela koja sadrži odgovarajući primarni ključ već postoji. Čarobnjak pretraživanja Lookup Wizard stvara odnos za nas.

7. UNOŠENJE PRVIH ZAPISA

Pre nego što pređemo na kreiranje formulara za rad sa podacima, iskoristićemo priliku da unesemo podatke direktno u tabele. Način na koji Access smešta podatke u tabele je naizgled veoma jednostavan, a njegovo razumevanje je veoma važno za kasnije dizajniranje formulara i izveštaja. Od tabele koje smo kreirali otvorimo za unos u novu tabelu. Dvaklik na naziv tabele uz pritisnut taster CTRL otvara tabelu u dizajn pogledu (ili iz menija biramo design view). U sledećim primerima uneti su podaci u tabele koje sam prethodno kreirao.



Moramo zapamtitи novu tabelу !!!

Primarni ključ za tabelu delovi je polje ID i označava se u imenu polja ikonicom ključ.

The screenshot shows the Microsoft Access ribbon with the 'Design' tab selected. In the 'Views' section, the 'Primary Key' icon is highlighted. Below the ribbon, the 'Tables' section displays a table named 'delovi'. A new field 'Nova1' is being added, with its 'Field Name' set to 'ID' and 'Data Type' set to 'AutoNumber'.

Nastavljamo dalje sa organizacijom novih tabela i njihovim popunjavanjem sa potrebnim podacima izborom iz glavnog menija Datasheet View.

The screenshot shows the Microsoft Access ribbon with the 'Fields' tab selected. The 'Datasheet View' is selected in the left pane. The 'delovi' table is displayed in the main area, showing data for various parts such as '1/4W 51K', 'TRIMER VL 33K', 'PREKIDAC TS 503', 'ANTENA 2M 5/8', 'DUGME 11313', and 'KABL LICNA'. The table includes columns for 'ID_dela', 'Naziv dela', 'Opis dela', 'Cena dela', and 'ID_uredjaja'.

Tabela “*klijenti*”

	ID_klijenta	Ime	Prezime	Adresa	Telefon	Add New Field
*	1	Ivan	Manić	Vozdova 12	018125656	
*	2	Jovan	Cvetković	Pantelejska 45	018123456	
*	(New)					

Tabela “*serviser*”

	ID_servisera	Prezime	Ime	RadniSat	Add New Field
*	1	Ćirić	Miljan	200,00	
*	2	Pečić	Bratislav	200,00	
*	(New)				

Tabela “*uredjaji*”

	ID_uredjaja	Naziv	Tip	Proizvodjac	ID_klijenta	Opis problem	DatumDosta
*	1	TV	123vc	Sony	1	Nece da se upa	24.5.2015
*	2	DVD	010vc	Panasonic	2	Ne cita disk	24.5.2015
*	6						28.5.2015
*	(New)						

DatumUgrad	DatumZavrsi	VremeRada	Kolicina	ID_servisera	Add New Field
24.5.2015	24.5.2015	1	1	1	
24.5.2015	24.5.2015	2	1	2	

Tabela "dobavljac"

	ID_dobavlja	Naziv dobav	Adresa	Telefon	e-mail	Grad	Add New Field
	1	Winwin	Bulevar Nemar 018546545		wiwwin@gmail.com	Nis	
	2	Puls	Kneza Milosa 4 fsd		puls@gmail.com	Pirot	
	3	Jakov sistem	Bulevar Nemar fds		jakovsistem@gmail.com	Nis	
*	(New)						

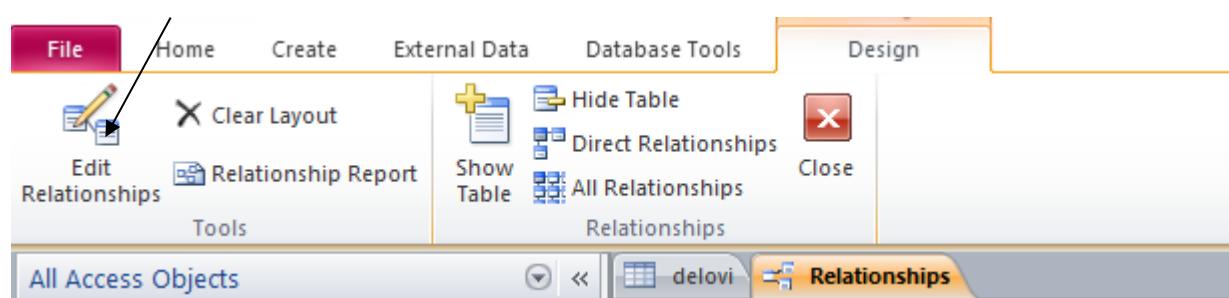
Tabela "delovi"

	ID_dela	Naziv dela	Opis dela	Cena dela	ID_uredjaja	Add New Field
	1	1/4W 51K	Otpornik, 510	100,00	2	
	2	TRIMER VL 33K	Trimer VL 33 K	80,00	2	
	3	PREKIDAC TS 50	Prekidac TS 50	150,00	1	
	5	ANTENA 2M 5/	Antena 2m 5/8	365,00		
	6	DUGME 11313	Dugme 11313	150,00		
	7	KABL LICNA	crveni, 1 X 0.14	500,00		
*	(New)					

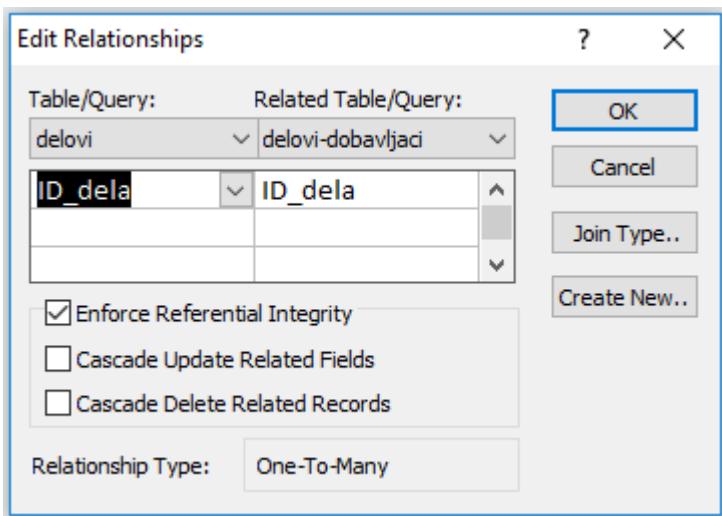
Tabela "delovi-dobavljac"

	ID_dela	ID_dobavlja
	1	1
	2	2
	3	3
*		

7.1 Uspostavljanje relacija vršimo već opisanim postupkom:



I iz padajućeg menija biramo jednu i drugu tabelu za uspostavljanje relacije. One mogu biti u odnosima: 1:1 1:n n:n povezujući ključne atribute u njima.



8. UPITI

Upiti u Accessu su veoma moćni i svestrani. Upiti odabiranja nam omogućavaju da prikažemo, sažmemo i obavimo izračunavanja sa podacima koji se nalaze u tabelama. Akcioni upiti nam omogućavaju da unesemo i ažuriramo podatke u tabelu i izbrišemo ih iz tabele.

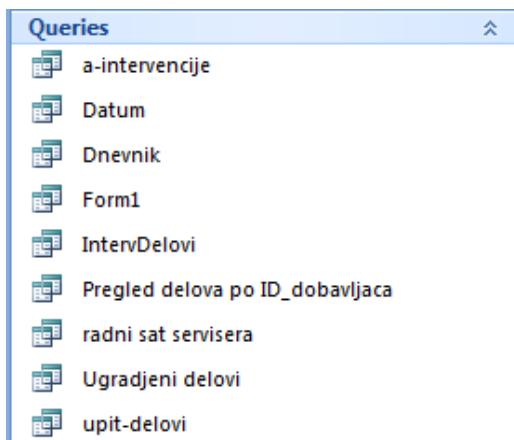
Da bismo izvršili upit, iz spiska Objects odaberimo Queries, dva puta kliknite upit koji želimo da izvršimo, ili označimo upit koji želimo da izvršimo, a zatim kliknimo Open. Pošto izvršimo odabrani upit, u tabelarnom obliku će se prikazati sva polja koja su zadata upitom I svi slogovi koji zadovoljavaju kriterijum upita. Pošto pokrenemo akcioni upit, Access obavlja zadatu akciju, recimo, pravljenje nove tabele ili dodavanje podataka u postojeću tabelu. U principu, podatke koje dobijamo kao rezultat upita možemo ažurirati jer je rezultat upita zapravo dinamički skup slogova, na osnovu podataka tabele. Znači, upiti su posebni pregledi kojima se postavlja neki upit nad tabelom i koji izdvajaju podatke za pregled na ekranu ili štampanje putem izveštaja.

Upiti mogu filtrirati podatke ne samo iz jedne tabele već se to može činiti iz više tabela ili čak iz jednog ili više drugih upita. Takođe mogu istovremeno filtrirati podatke po više kriterijuma.

Pošto pokrenemo upit Employee Sales by Country, pojaviće se tabelami prikaz podataka koji obuhvata sva polja zadata upitom i sve slogove koji zadovoljavaju kriterijum upita. Pošto zapišemo upit, samo njegovu definiciju, plan ili svojstva formatiranja, i tabelami prikaz su

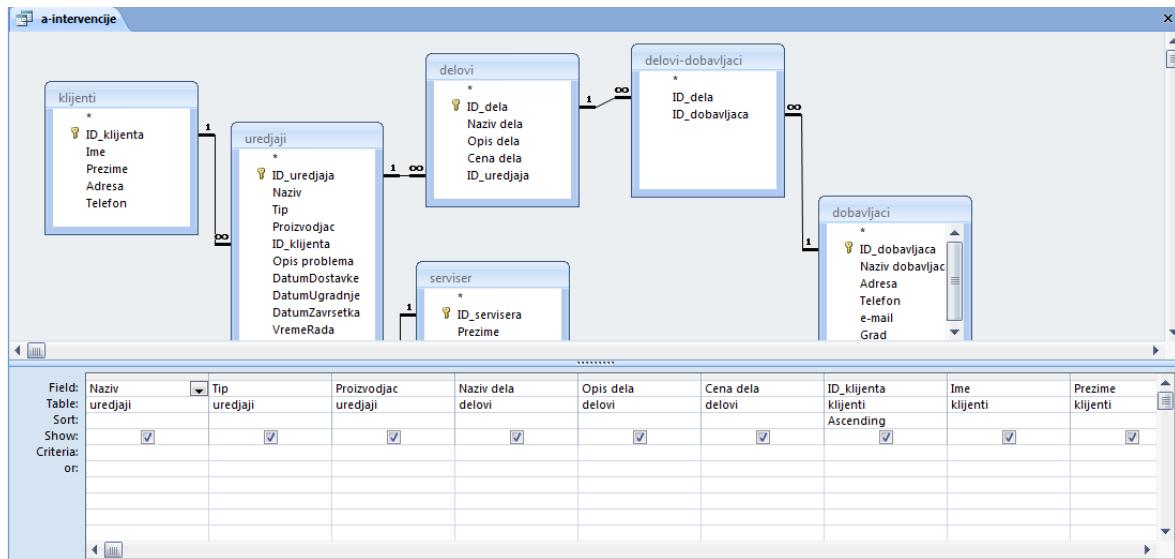
zapravo zapisani u bazi podataka. Access nam za strukturiranje uputa nudi intuitivni alat. Da bismo otvorili prozor Query Design, u prozoru Database iz spiska Objects odaberimo Queries, zatim odaberimo upit koji želimo da izmenimo, pa kliknimo Design. U toku izvršavanja upit traži kriterijum izvršavanja i pomoću njega određuje koje slogove će prikazati u rezultatu. Sem izdvajanja podataka postoje i posebni upiti koji mogu kreirati tabelu koristeći drugu tabelu ili više drugih tabela nad kojom je postavljen upit, odnosno mogu brisati podatke iz postojeće tabele takođe po nekom kriterijumu. Postoje i drugi: Update upit (proračunava podatke i rezultat upisuje u polje tabele, npr. ukupno=cena*količina - upisuje rezultat u polje ukupno), Union upit, Crosstab upit (proračun nad prividnom matricom) i Append upit koji dodaje slogove u postojeću tabelu.

Za potreb izrade ovog rada ja sam kreirao 9 upita koji su prikazani na slici 8.2.



Slika 8.2. Kreirani upiti

Kreiran upit "a-intervencije" prikaza je na slikama 8.3. i 8.4.



Slika 8.3. Upit “a-intervencije” prikazan u Design View-u

```

SELECT uredjaji.Naziv, uredjaji.Tip, uredjaji.Proizvodjac, uredjaji.[Opis problema], klijenti.Ime, klijenti.Prezime, klijenti.Adresa, klijenti.Telefon, uredjaji.[Opis problema], uredjaji.DatumUgradnje, uredjaji.DatumZavrsetka, uredjaji.DatumDostavke, uredjaji.Kolicina, [Cena dela]*[Kolicina] AS cena_delova, uredjaji.VremeRada, serviser.RadniSat, [VremeRada]*[RadniSat] AS cena_rada, [cena_delova]+[cena_rada] AS ukupna_cena, delovi.ID_dela, dobavljac.[Naziv dobavljaca], uredjaji.ID_uredjaja
FROM (serviser INNER JOIN (klijenti INNER JOIN uredjaji ON klijenti.ID_klijenta = uredjaji.ID_klijenta) ON serviser.ID_servisera = uredjaji.ID_servisera) INNER JOIN (dobavljac INNER JOIN (delovi INNER JOIN (dobavljac ON delovi.ID_dela = [delovi-dobavljac].ID_dela) ON dobavljac.ID_dobavljaca = [delovi-dobavljac].ID_dobavljaca) ON uredjaji.ID_uredjaja = delovi.ID_uredjaja)
ORDER BY klijenti.ID_klijenta;
    
```

Slika 8.4. Upit “a-intervencije” prikazan u SQL View-u

Upit “Datum”

```

SELECT uredjaji.DatumDostavke, uredjaji.DatumUgradnje, uredjaji.DatumZavrsetka
FROM uredjaji;
    
```

Upit “Dnevnik”

```

SELECT uredjaji.Naziv, uredjaji.Tip, uredjaji.Proizvodjac, uredjaji.[Opis problema], klijenti.Ime, klijenti.Prezime, klijenti.Adresa, klijenti.Telefon, uredjaji.[Opis problema], uredjaji.DatumDostavke, uredjaji.DatumUgradnje, uredjaji.DatumZavrsetka, uredjaji.Kolicina, delovi.[Cena dela], serviser.RadniSat, [VremeRada]*[RadniSat] AS cena_rada, [cena_delova]+[cena_rada] AS ukupna_cena, [Cena dela]*[Kolicina] AS cena_delova, klijenti.ID_klijenta, uredjaji.ID_uredjaja
FROM (serviser INNER JOIN (klijenti INNER JOIN uredjaji ON klijenti.ID_klijenta = uredjaji.ID_klijenta) ON serviser.ID_servisera = uredjaji.ID_servisera) INNER JOIN delovi ON uredjaji.ID_uredjaja = delovi.ID_uredjaja;
    
```

Upit “Form1”

```

SELECT delovi.ID_dela, dobavljac.[Naziv dobavljaca], delovi.[Naziv dela], delovi.[Opis dela], delovi.[Cena dela], delovi.ID_uredjaja, [delovi-dobavljac].ID_dobavljaca
FROM dobavljac INNER JOIN (delovi INNER JOIN [delovi-dobavljac] ON delovi.ID_dela = [delovi-dobavljac].ID_dela) ON dobavljac.ID_dobavljaca = [delovi-dobavljac].ID_dobavljaca
ORDER BY delovi.[Naziv dela], delovi.[Opis dela];
    
```

Upit "IntervDelovi"

```
IntervDelovi
SELECT delovi.ID_dela, delovi.[Naziv dela], uredjaji.Kolicina, delovi.[Cena dela]
FROM uredjaji INNER JOIN delovi ON uredjaji.ID_uredjaja = delovi.ID_uredjaja;
```

Upit "Pregled delova po ID_dobavljaca"

```
Pregled delova po ID_dobavljaca
SELECT dobavljaci.[Naziv dobavljaca], dobavljaci.Adresa, dobavljaci.Telefon, dobavljaci.Grad, dobavljaci.[e-mail], delovi.[Naziv dela], delovi.[Cena dela]
FROM dobavljaci INNER JOIN (delovi INNER JOIN [delovi-dobavljaci] ON delovi.ID_dela=[delovi-dobavljaci].ID_dela) ON dobavljaci.ID_dobavljaca=[delovi-dobavljaci].ID_dobavljaca
WHERE dobavljaci.[ID_dobavljaca]=[Inesite id dobavljaca];
```

Upit "radni sat servisera"

```
radni sat servisera
SELECT serviser.RadniSat, serviser.ID_servisera
FROM serviser;
```

Upit "Ugradjeni delovi"

```
Ugradjeni delovi
SELECT delovi.[Naziv dela], delovi.[Cena dela], uredjaji.ID_uredjaja
FROM uredjaji INNER JOIN delovi ON uredjaji.ID_uredjaja = delovi.ID_uredjaja;
```

Upit "upit-delovi"

```
upit-delovi
SELECT delovi.[Naziv dela], delovi.[Cena dela], dobavljaci.[Naziv dobavljaca], delovi.ID_dela, delovi.[Opis dela]
FROM dobavljaci INNER JOIN (delovi INNER JOIN [delovi-dobavljaci] ON delovi.ID_dela = [delovi-dobavljaci].ID_dela) ON dobavljaci.ID_dobavljaca = [delovi-dobavljaci].ID_dobavljaca
ORDER BY delovi.ID_dela;
```

9. FORME

Iako u pogledu Datasheet tabele možemo unositi i menjati podatke, ne možemo dovoljno dobro kontrolisati akcije korisnika, niti možemo učiniti mnogo toga kako bismo olakšali postupak unošenja podataka. Zbog toga se koriste formulari. Accessovi formulary imaju mnogo osobina i veoma su moćni i fleksibilni.

Forme, maske, elektronski obrasci ili formulari su neki od naziva koji su odomaćeni za englesku reč Forms. Elektronski obrasci predstavljaju ekvivalent papirnim obrascima, a koriste se za unos podataka u elektronske, računaru razumljive, baze podataka. Prednost elektronskih obrazaca nad papirnim je višestruka: unos podataka može biti delimično kontrolisan, u određenim poljima se mogu unapred ponuditi vrednosti dok se pojedina polja mogu popuniti automatski proračunom jedne ili više unetih vrednosti u druga polja. Pre osmišljavanja izgleda

elektronskih obrazaca valja uzeti na uvid papirne obrasce sa kojima su operateri do sada radili. U principu, nama je svejedno kako ćemo ih dizajnirati dok god poštujemo određena pravila, ali na taj način bismo olakšali operaterima prilagođavanja na nov način rada.

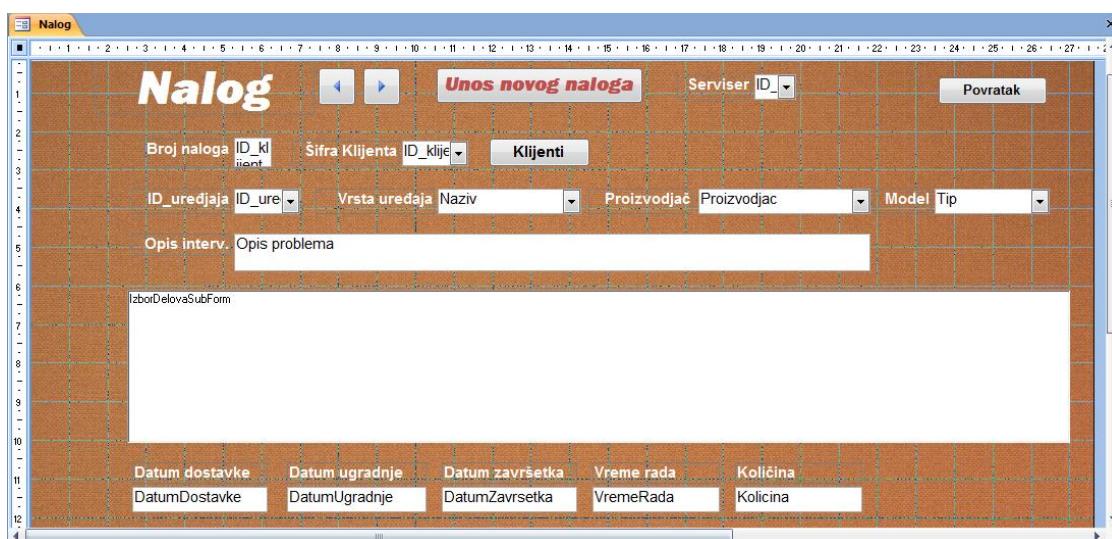
Da bismo prikazali bilo koji formular, iz spiska Objects odaberimo Forms. Zatim dva puta kliknimo formular koji želimo da prikažemo, ili obeležimo formular koji želimo da prikažemo tako što ćemo na njega kliknuti pa zatim kliknimo Open. Kao što to možemo da uradimo za tabele i upite, i formulare možemo prikazati u pogledu Design. Da bismo videli strukturu formulara, iz spiska Objects odaberimo ikonu Forms, odaberimo formular čiju strukturu želimo da promenimo, pa zatim kliknimo Design.

Forma je samo "pogled" na tabelu u čoveku prihvatljivijem izgledu. Nad formiranim tabelom koja sadrži podatke o nekom članu formiramo masku za unos koja ima izgled papirnog obrasca - kartice, te manipulant ne mora da poznaje rad sa Accessom, nego samo koristi elektronske obrasce i unosi podatke putem tastature, baš kao što je to ranije činio olovkom.

Forma za unos podataka može sadržati slike, komandne tastere, može sadržati logička polja predstavljena putem prekidača, može sadržati podformu itd.

Forma ne mora da sadrži niti jedno polje za unos ili prikaz podataka. Ona jednostavno može biti pozdravni ekran vaše aplikacije sa prigodnim tekstrom i slikom, koje se sama posle isteka nekog vremenskog intervala zatvara. Takve forme mogu uzeti oblik pozdravne slike koja se pojavljuje prilikom starta nekog većeg programa, kao što se to događa u programima paketa Microsoft Office.

Za potrebe ovog rada ja sam kreirao i dizajnirao osamnaest formi. Neke od kreiranih formi prikazane su na slikama 9.1, 9.2. i 9.3.



Slika 9.1 Forma "Nalog" prikazana u Design View-u

Slika 9.2. Forma "Delovi" prikazana u Design View-u

Slika 9.3. Forma "Dobavljači" prikazana u Design View-u

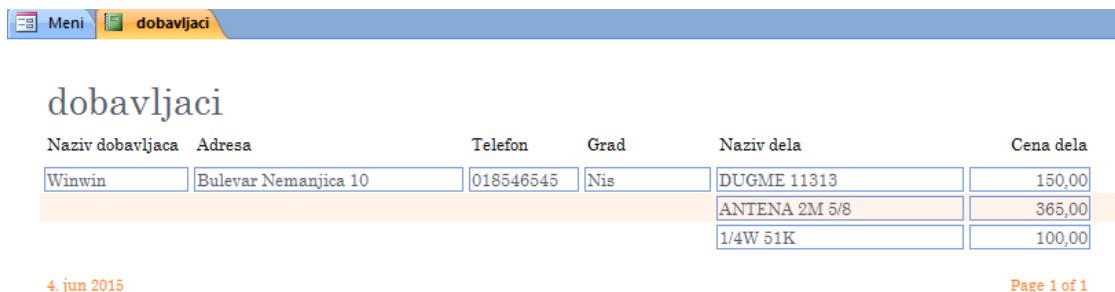
10. IZVEŠTAJI

Izveštaji su elektronski oblik klasičnih papirnih obrazaca koje možemo gledati na ekranu ili štampati na papir. Podaci prikazani u izveštaju mogu biti izvedeni iz tabele ili iz upita. Dizajniranje izveštaja je veoma slično dizajniranju formi. Kada kreiramo izveštaj za koji očekujemo da će imati više strana, u nekom od zaglavlja strane dodelimo i svojstvo [Pages] koje daje ukupan broj strana. Izveštaj u čijem dnu piše, recimo, "Strana 6 od 75" ne samo da daje korisnu informaciju, već svakako daje utisak ozbiljnog dokumenta.

Dok formiramo izgled lista izveštaja, često prelazimo iz Design prozora u Print Preview, kako bismo stekli prvi utisak o nekom izvedenom koraku. Ako smo definisali svojstvo [Pages], svaki put će se ova vrednost ponovo izračunavati prolaskom kroz celu bazu i nepotrebno nam oduzimati vreme.

Zato, ako imamo nameru da koristimo ovo svojstvo, nećemo ga definisati pre samog kraja posla. U međuvremenu u zagлавje stavimo neki običan tekst (recimo, "OvdeBrojStrana"), koji će nas podsetiti na taj detalj. U slučaju rada sa velikom bazom, razlika u vremenu dobijanja rezultata u prozoru Print Preview će biti drastična.

U ovom radu kreirana su četiri izveštaja: dobavlјaci (Slika 10.1.), dosadasnje intervencije (Slika 10.2.), Lista delova (Slika 10.3.), Racun (Slika 10.4.).

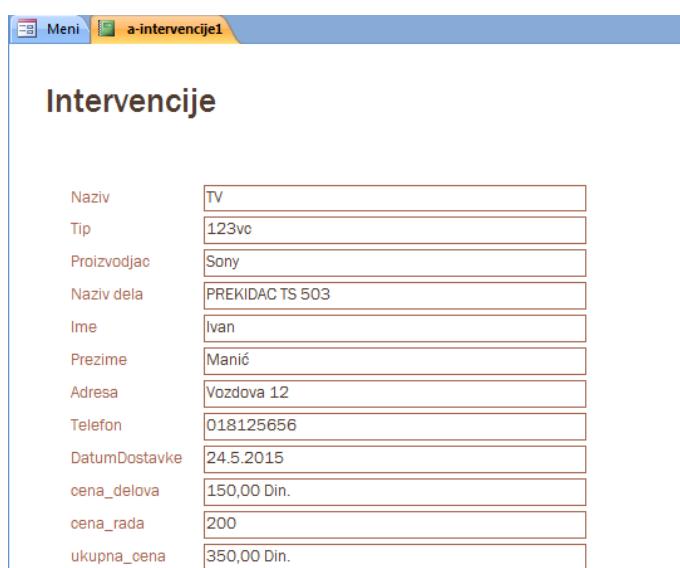


The screenshot shows a report titled "dobavlјaci". At the top, there are two tabs: "Meni" and "dobavlјaci", with "dobavlјaci" being the active tab. The report displays a table with the following columns: Naziv dobavlјaca, Adresa, Telefon, Grad, Naziv dela, and Cena dela. The data in the table is as follows:

Naziv dobavlјaca	Adresa	Telefon	Grad	Naziv dela	Cena dela
Winwin	Bulevar Nemanjica 10	018546545	Nis	DUGME 11313	150,00
				ANTENA 2M 5/8	365,00
				1/4W 51K	100,00

At the bottom left of the report area, it says "4. jun 2015". At the bottom right, it says "Page 1 of 1".

Slika 10.1. Izveštaj "dobavlјaci"



The screenshot shows a report titled "Intervencije". At the top, there are two tabs: "Meni" and "a-intervencije1", with "a-intervencije1" being the active tab. The report displays a form with the following fields and their values:

Naziv	TV
Tip	123vc
Proizvodjac	Sony
Naziv dela	PREKIDAC TS 503
Ime	Ivan
Prezime	Manić
Adresa	Vozdova 12
Telefon	018125656
DatumDostavke	24.5.2015
cena_delova	150,00 Din.
cena_rada	200
ukupna_cena	350,00 Din.

Slika 10.2. Izveštaj "dosadasnje intervencije"

Dobavljac	ID_dela	Naziv dela	Opis dela	Cena dela
Winwin	1	1/4W 51K	Otpornik, 51000.00 Ohm, 0.25W, Uglijenos	100,00
Puls	2	TRIMER VL 33K	Trimer VL 33 KO	80,00
Jakov sistem	3	PREKIDAC TS 503	Prekidac TS 503, 2X1, 250V/2A	150,00
Winwin	5	ANTENA 2M 5/8	Antena 2m 5/8 za kola	365,00
Winwin	6	DUGME 11313	Dugme 11313	150,00
Puls	7	KABL LICNA	crveni, 1 X 0.14mm	500,00

4. jun 2015 Page 1 of 1

Slika 10.3. Izveštaj "Lista delova"

Racun broj: 1	Datum predaje: 4.6.2015	
Klijent: Ivan Manić Vozdova 12 018125656	Uredjaj: 1 TV Sony 123vc	
	Cena delova: 150,00 Din. Cena rada: 200	
Uredjaj preuzeo:	Overava:	Ukupna cena: 350,00 Din.

Slika 10.4. Izveštaj "Racun"

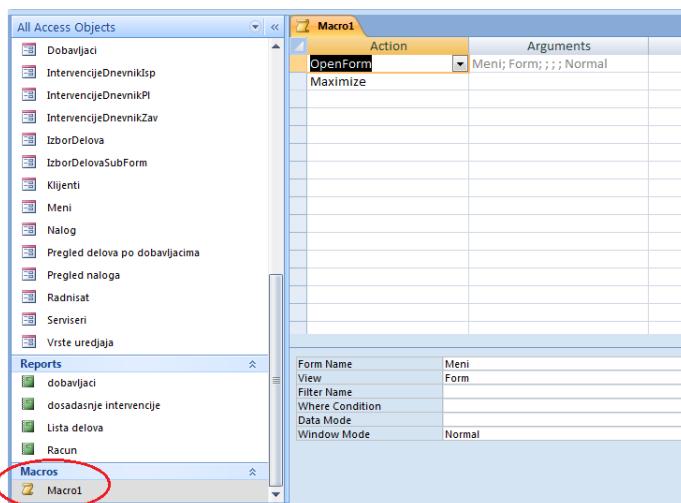
11. MAKRO PROGRAM

Makroi u Accessu nisu kao makroi u drugim članovima paketa Office. Ne možemo ih snimiti, kao što to možemo uraditi npr. u Excelu, a Access ih ne zapisuje kao kod u programskom jeziku Visual Basic for Applications (VBA). Pomoću Accessovih makroa možemo obaviti većinu poslova koje možemo obaviti pomoću tastature, menija i paleta alata. Makroi nam

omogućavaju da u aplikaciju ugradimo logiku. Uopšte, radije koristimo VBA kod koji se nalazi u modulima, a ne u makroima, za obavljanje poslova koje aplikacija mora da obavi. To je zato što su moduli VBA koda fleksibilniji i moćniji od makroa. Iako su se u ranijim verzijama Accessa neki poslovi mogli obaviti samo pomoću makroa, programeri ih danas veoma malo koriste. Stoga se u ovom radu neću baviti pravljenjem aplikacija pomoću makroa.

Da bismo pokrenuli makro, iz spiska Objects odaberite Macros, kliknimo makro koji želimo da pokrenemo, pa zatim kliknimo Run. Access će zatim izvršiti akcije koje su navedene u makrou. Da bismo videli strukturu makroa, iz spiska Objects odaberimo Macro, obeležimo makro koji želimo da izmenimo, a zatim kliknimo design kako bismo ga otvorili u prozoru Macro.

Na slici 11.1. prikazan je makro koji sam kreirao za potrebe mog rada. Na slici 11.2. prikazana je Meni strana koju pokreće ovaj Macro.



Slika 11.1. Macro1 prikazan u Design View-u



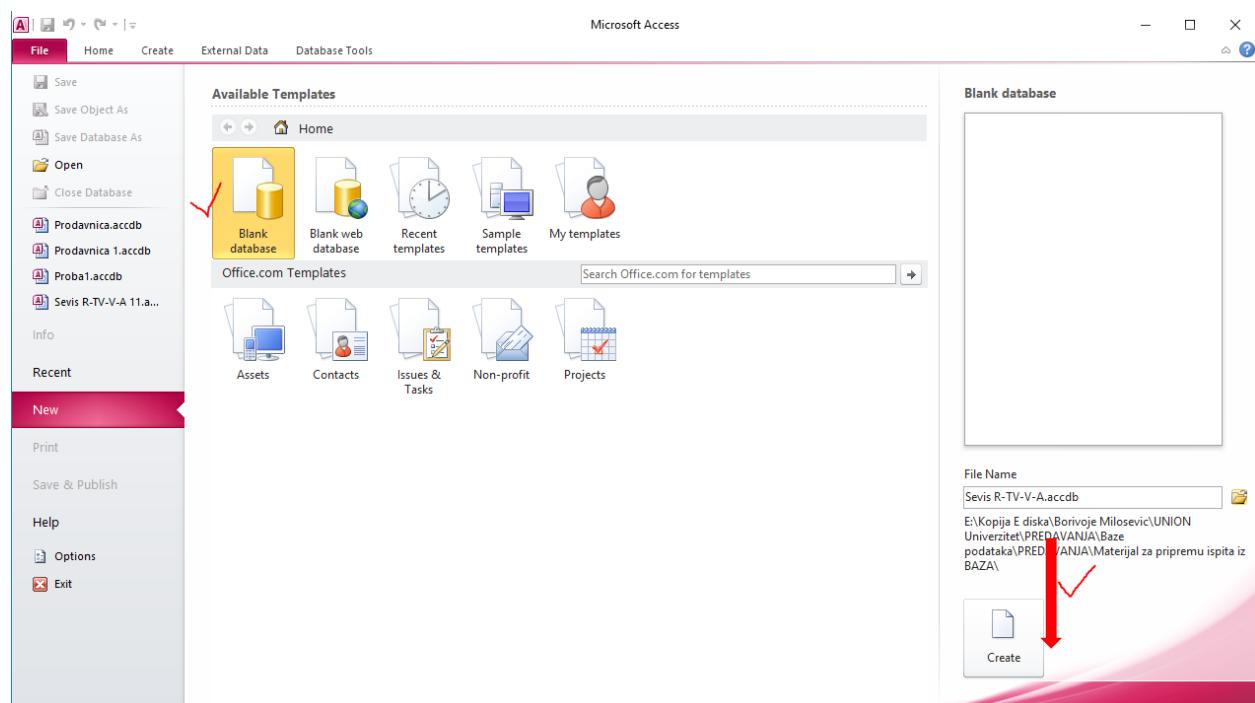
Slika 11.2. Izgled Meni strane koju pokreće Macro1

12. BAZA PODATAKA – realizacija preko SQL

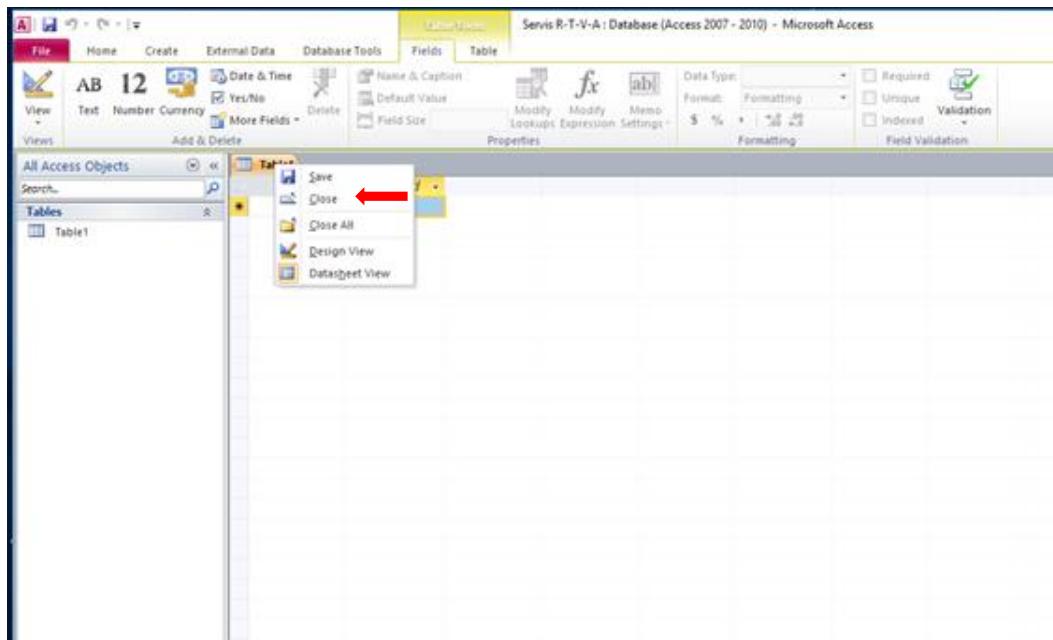
Drugi deo

Otvorićemo bazu podataka MS Access kao **Blank database** koja će ćemo zapamtiti na tekućem direktorijumu (ili na onom koga odaberemo) i u polje uneti naziv baze:

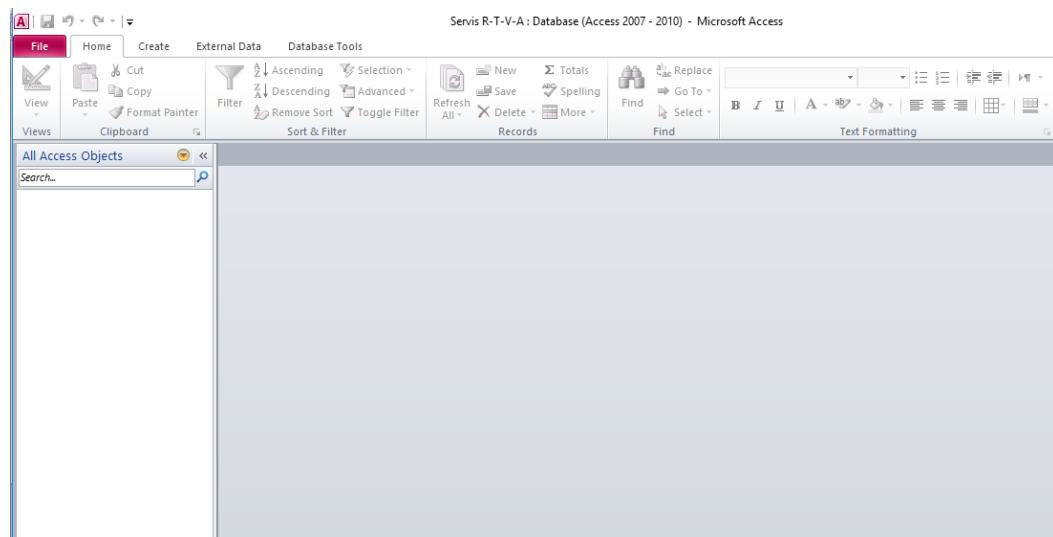
Servis R-T-V-A



Zatvorićemo ponudjenu tabelu Table1, desnom tasterom na njen jezičak i izborom iz padajućeg menija opcije Close.

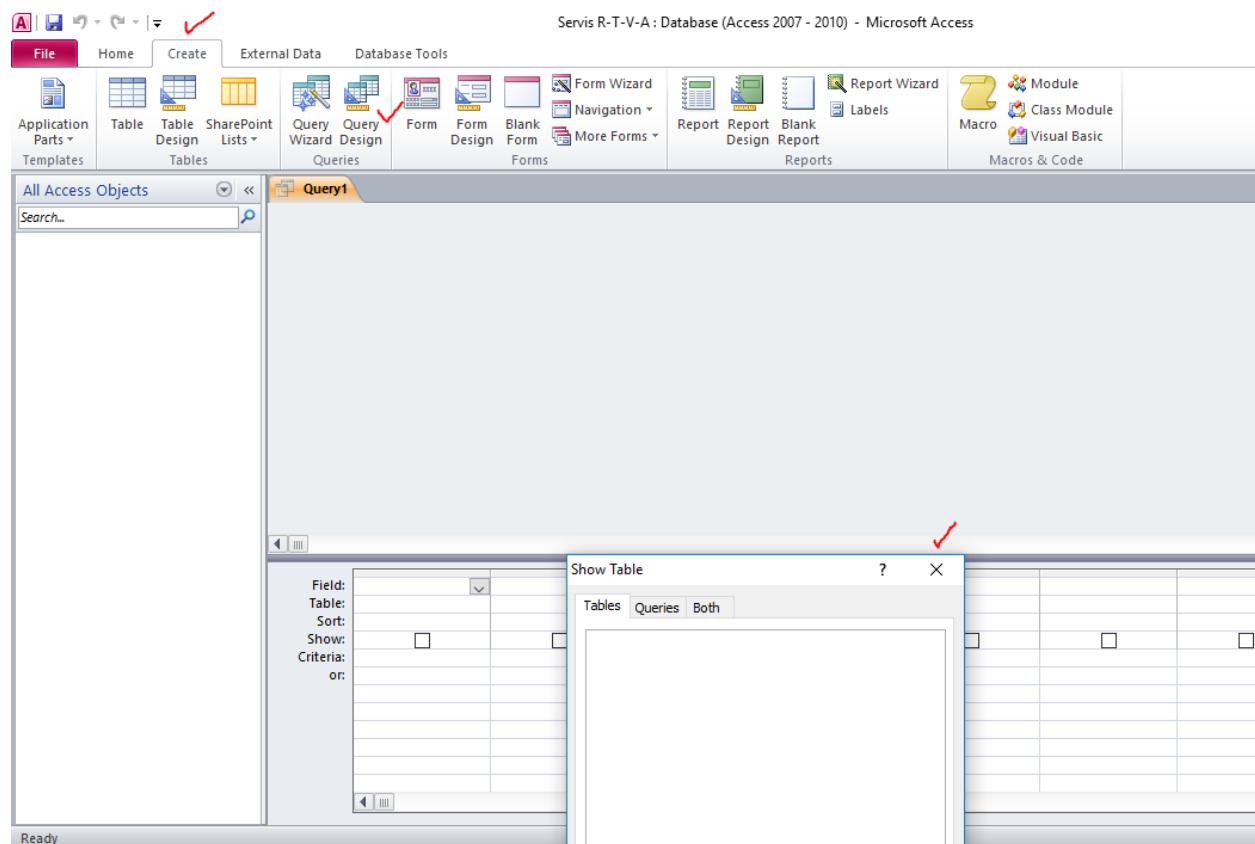


Sada nam se otvara prazan prostor za rad sa našom bazom i njenu organizaciju:

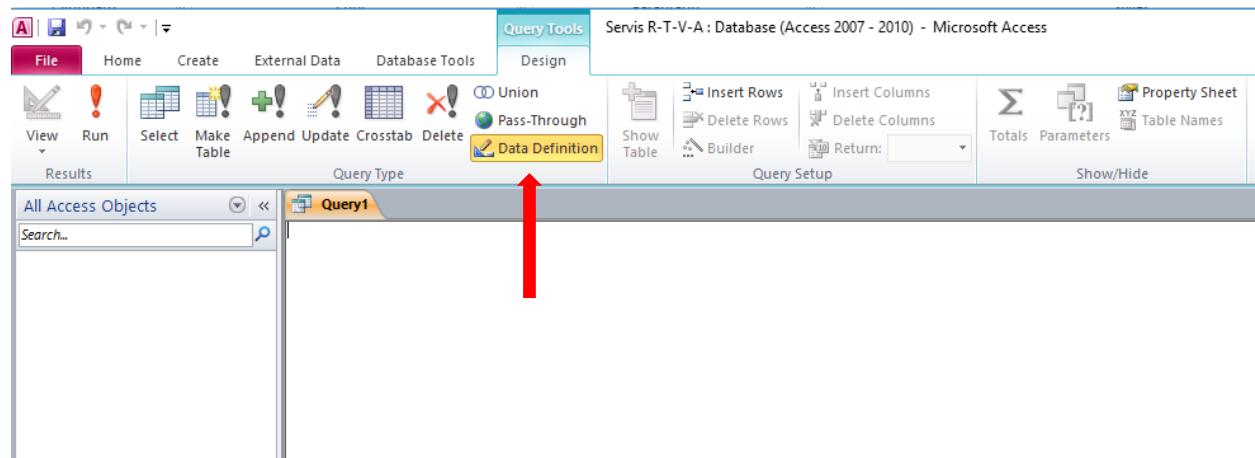


Dalje pristupamo primenom SQL jezika da formiramo sve tabele baze podataka:

1. Izborom opcije Create
2. Izborom opcije Query Design
3. Zatvaramo prozor Show Table opcijom Close (X)



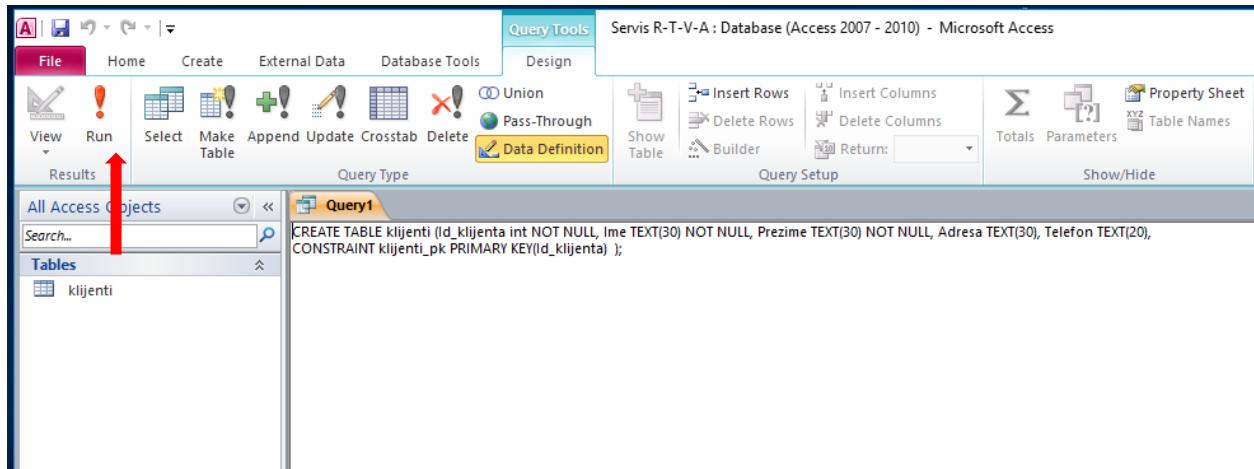
Da bi otvorili editorski prozor za unos našeg prvog SQL zapisa (prikazuje se jezičak prvog upita Query1) i organizovanje prve tabele pokrenućemo opciju:



Otvara se editorski prozor za ukucavanje (ili copy/peste) našeg prvog SQL skripta (Query1) za formiranje tabele "kljienti" sa definisanim atributima i njihovim tipom podataka kao i za definisanje primarnog ključa. Naš prvi script za organizovanje tabele "kljenti" počinje naredbom CREATE TABLE praćeno imenom nove tabele.

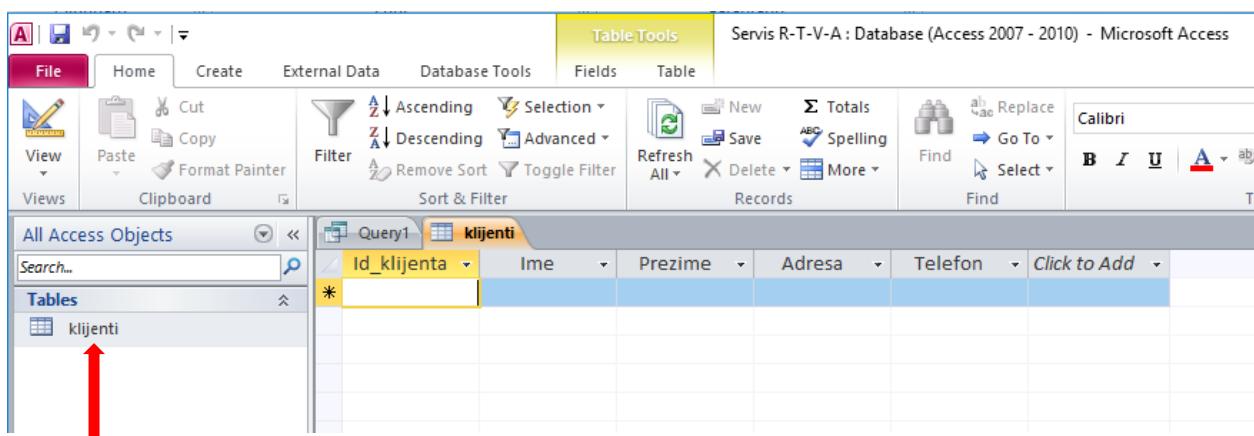
U naredbi CONSTRAINT definišemo da je primarni kluč ove tabele "Id_klijenta", i dajemo virtualni naziv primarnog ključa za tabelu "kljenti" nazvanog npr. "kljenti_pk".

```
CREATE TABLE klijenti (Id_klijenta int NOT NULL, Ime TEXT(30)
NOT NULL, Prezime TEXT(30) NOT NULL, Adresa TEXT(30), Telefon
TEXT(20),
CONSTRAINT klijenti_pk PRIMARY KEY(Id_klijenta) );
```

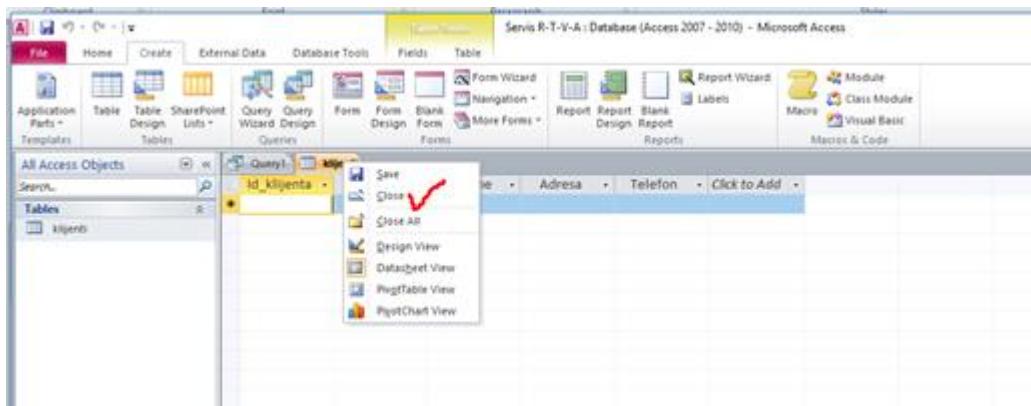


Skript se izvršava pritiskom tastera Run i odmah formira tabelu "klijenti" sa postavljenim parametrima napisanih u skriptu !!!

Možemo proveriti kako sa svim atributima izgleda naša nova tabela ako dva puta kliknemo na tabelu "klijenti" u prozoru All Access Objects.

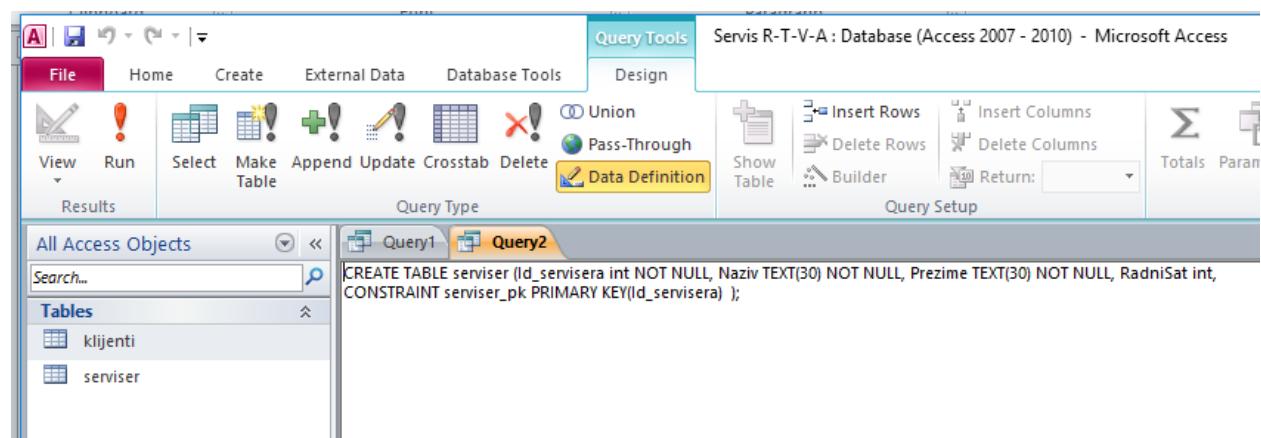


Moramo sada zatvoriti tabelu "klijenti" kako bi pristupili organizovanju sledeće:

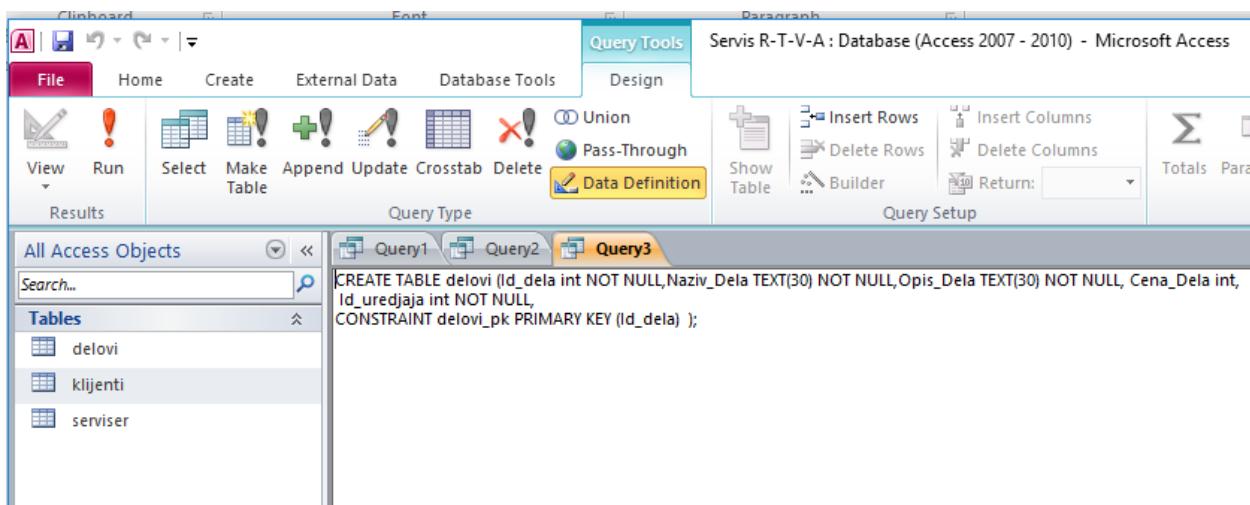


Zatim se postupak ponavlja za organizovanje svih ostalih tabela u bazi podataka.

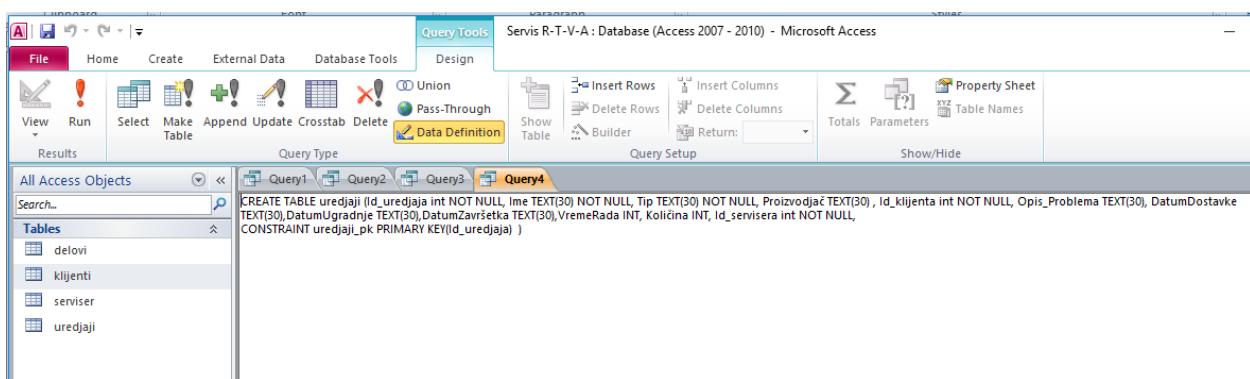
```
CREATE TABLE serviser (Id_servisera int NOT NULL, Naziv TEXT(30)
NOT NULL, Prezime TEXT(30) NOT NULL, RadniSat int,
CONSTRAINT serviser_pk PRIMARY KEY(Id_servisera) );
```



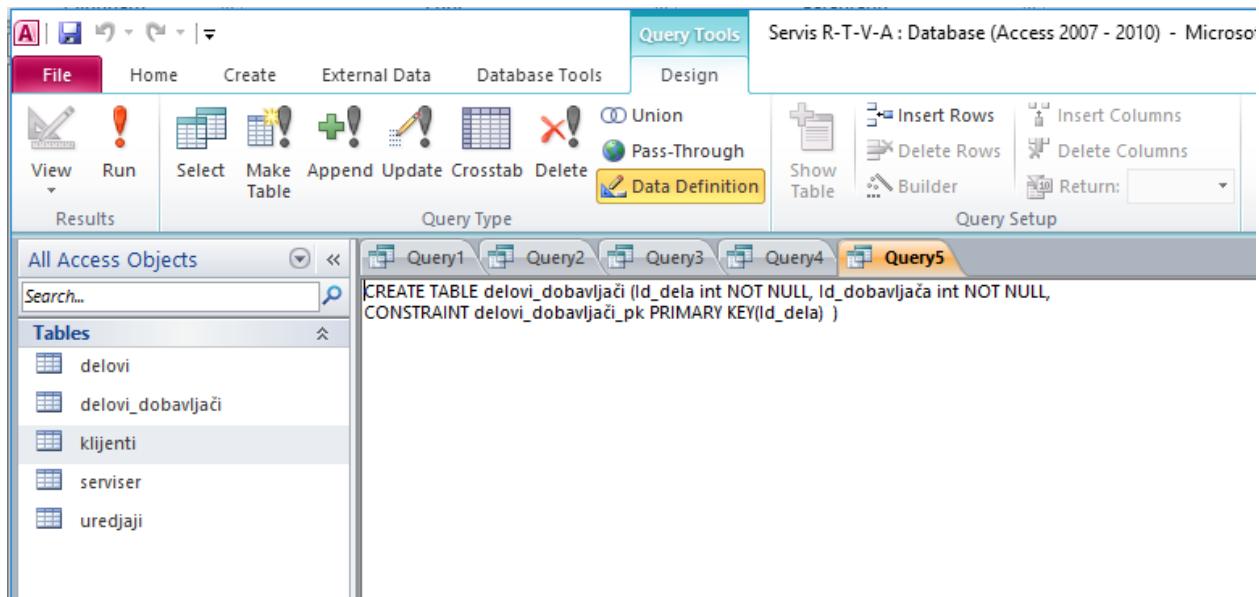
```
CREATE TABLE delovi (Id_dela int NOT NULL,Naziv_Dela TEXT(30)
NOT NULL,Opis_Dela TEXT(30) NOT NULL, Cena_Dela int,
Id_uredjaja int NOT NULL,
CONSTRAINT delovi_pk PRIMARY KEY (Id_dela) );
```



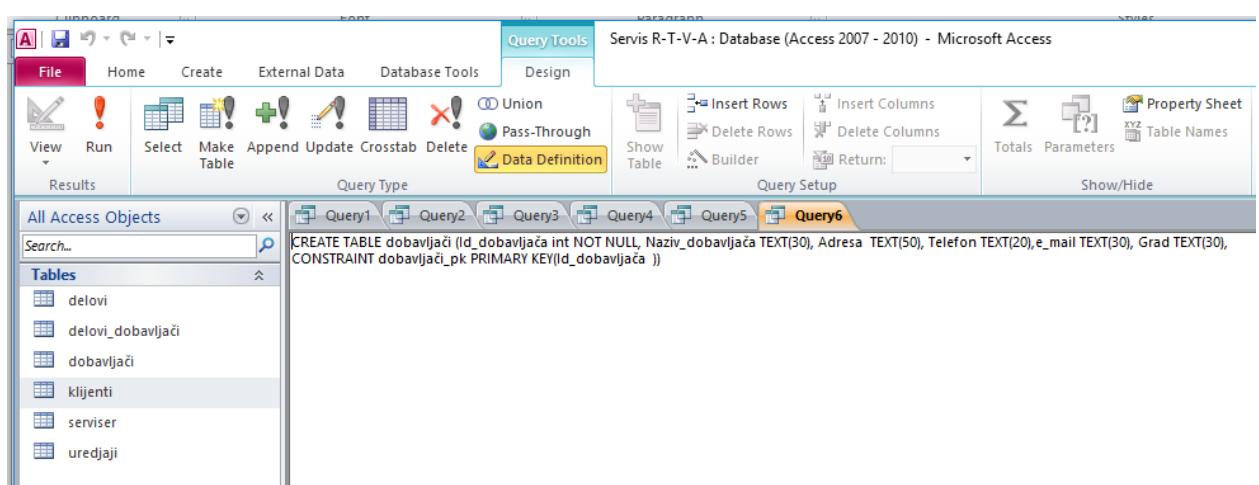
```
CREATE TABLE uredjaji (Id_uredjaja int NOT NULL, Ime TEXT(30)
NOT NULL, Tip TEXT(30) NOT NULL, Proizvodjač TEXT(30) ,
Id_klijenta int NOT NULL, Opis_Problema TEXT(30), DatumDostavke
TEXT(30), DatumUgradnje TEXT(30), DatumZavršetka
TEXT(30), VremeRada INT, Količina INT, Id_servisera int NOT NULL,
CONSTRAINT uredjaji_pk PRIMARY KEY(Id_uredjaja) )
```



```
CREATE TABLE delovi_dobavljači (Id_dela int NOT NULL,
Id_dobavljača int NOT NULL,
CONSTRAINT delovi_dobavljači_pk PRIMARY KEY(Id_dela) )
```



```
CREATE TABLE dobavljači (Id_dobavljača int NOT NULL,
Naziv_dobavljača TEXT(30), Adresa TEXT(50), Telefon
TEXT(20), e_mail TEXT(30), Grad TEXT(30),
CONSTRAINT dobavljači_pk PRIMARY KEY(Id_dobavljača ))
```



Sada možemo pregledati sadržaje svih tabela, nazine njihovih atributa i tipova podataka vezanih za njih, kao i definisani primarni ključ. To možemo izvršiti dvostrukim klikom miša na željenu tabelu i izabratи View/Datasheet View ili View/Design View:

Table Tools

Fields Servis R-T-V-A : Database (Access 2007 - 2010) - Microsoft Access

Records

Find

Design

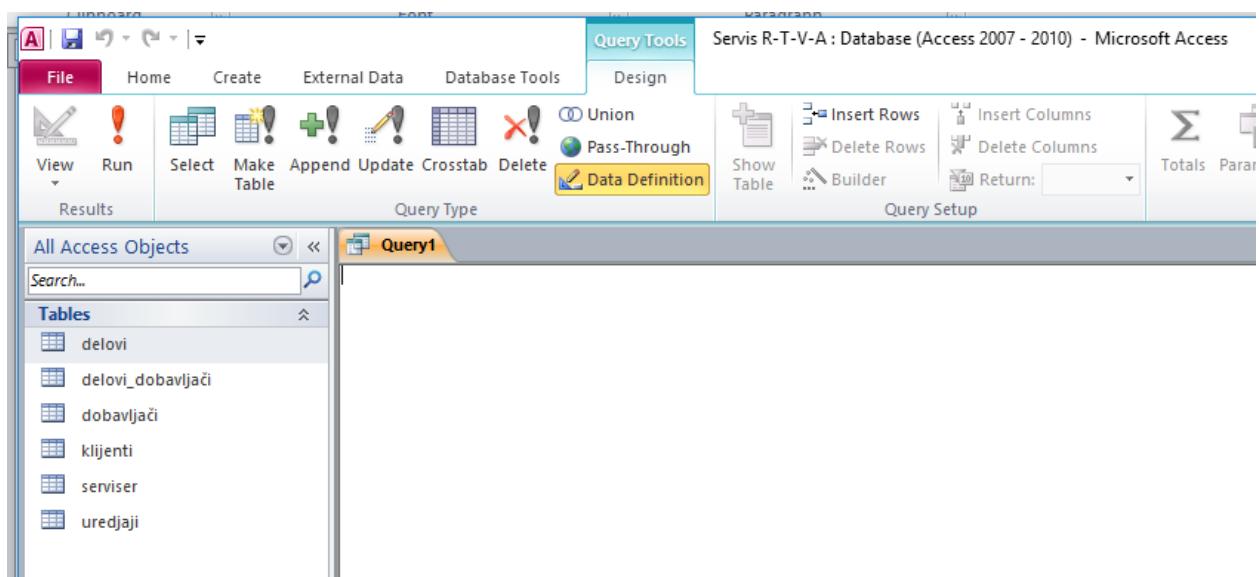
Field Properties

Field Name	Data Type
Id_dela	Number
Naziv_Dela	Text
Opis_Dela	Text
Cena_Dela	Number
Id_uredjaja	Number

13. POPUNJAVANJE TABELA

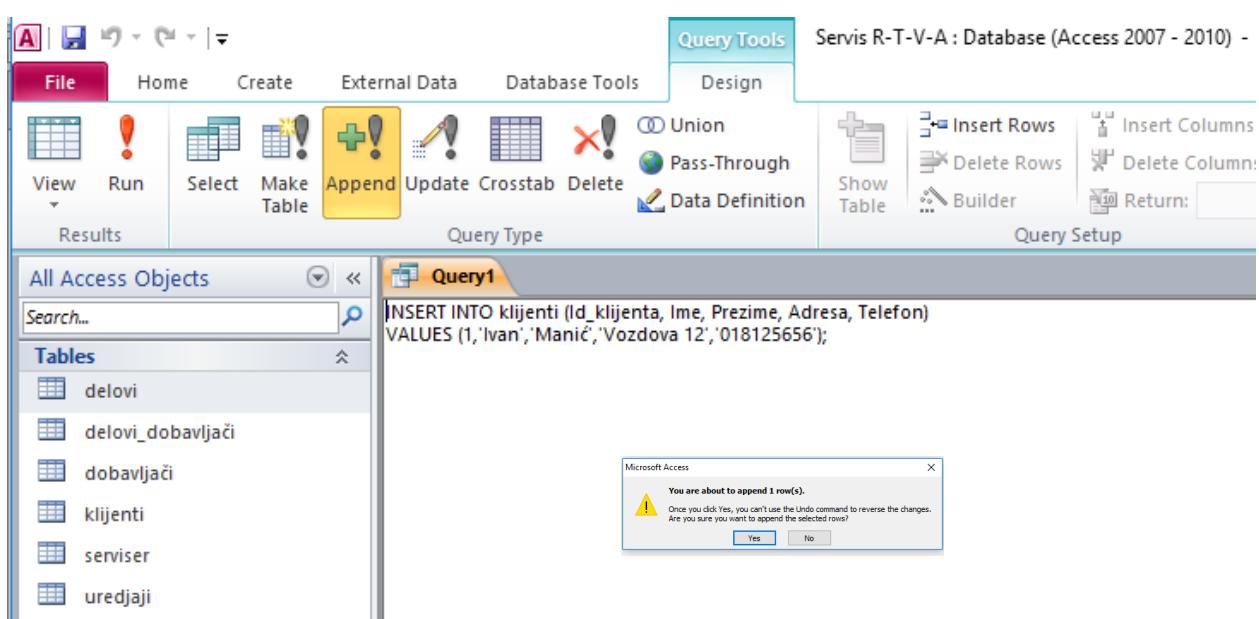
Potrebno je dakle sada zatvoriti pogledanu tabelu "delovi", desnim klikom miša na nju i izborom Close, u zatvoriti sve izvršene upite Query1.... QueryN istim postupkom ali sada izborom Close all.

Popunjavanje ili punjenje tabela vršimo naredbom INSERT INTO koju prati naziv prve tabele "klijenti" sa već definisanim atributima koji moraju biti isti kao u postupku kada smo je formirali. Zatim, unosimo u tabelu vrednosti ili pojave atributa naredbom VALUES u kojoj između zagrada stoje redom definisane njihove stvarne vrednosti - pojave. Sprovodimo isti postupak do otvaranja editorskog prozora SQL jezika.



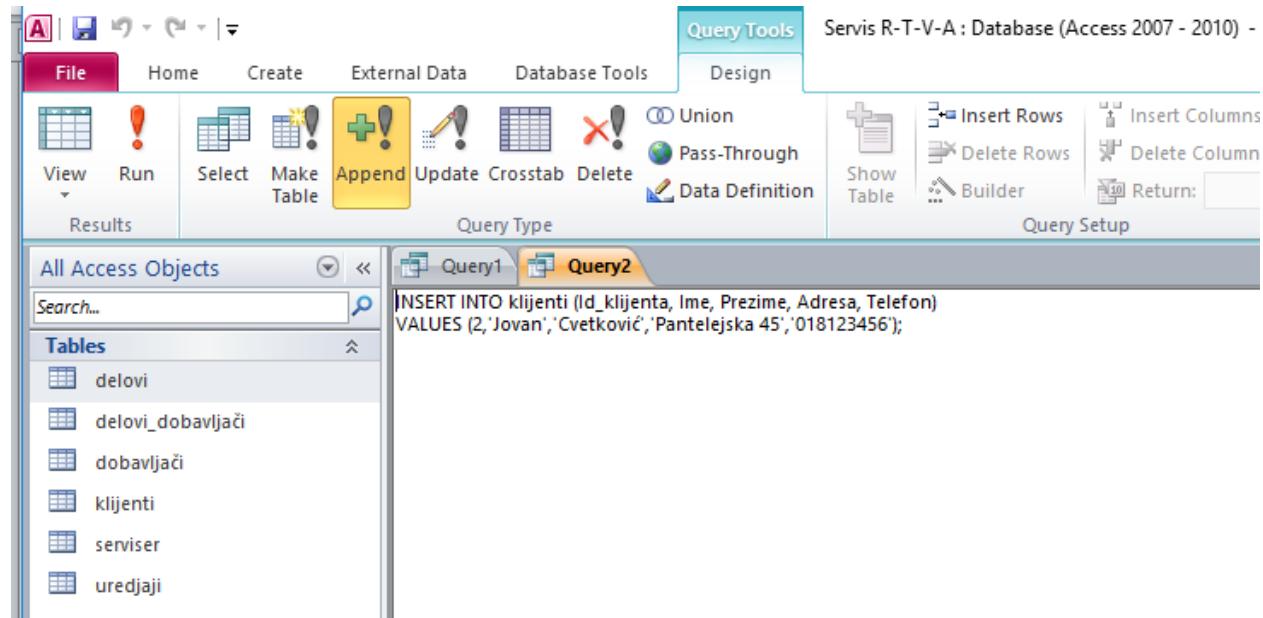
Izvršenjem Copy/Paste naredbe unosimo naš SQL script u editorski prozor i izvršavamo ga opcijom Run !!

```
INSERT INTO klijenti (Id_klijenta, Ime, Prezime, Adresa, Telefon)
VALUES (1, 'Ivan', 'Manić', 'Vozdova 12', '018125656');
```



Ponavljamo postupak za novu n-torku stvarnih vrednosti tabele "klijenti" :

```
INSERT INTO klijenti (Id_klijenta, Ime, Prezime, Adresa, Telefon)
VALUES (2, 'Jovan', 'Cvetković', 'Pantelejska 45', '018123456');
```



Kada završimo sa popunjavanjem tabele "klijenti", isti postupak važi i za popunjavanje svih ostalih tabela svojim definisanim skriptovima, postupno jedan po jedan.

```
INSERT INTO serviser (Id_servisera, Naziv, Prezime, RadniSat)
VALUES (1, 'Ćirić', 'Miljan', 200);

INSERT INTO serviser (Id_servisera, Naziv, Prezime, RadniSat)
VALUES (2, 'Pešić', 'Bratislav', 200);
```

GOTOVO za tabelu "serviser".

```
INSERT INTO delovi(Id_dela,Naziv_Dela,Opis_Dela,Cena_Dela,Id_uredjaja)
VALUES (1, 'Otpornik', '1/4W 51K', 100, 2);

INSERT INTO delovi(Id_dela,Naziv_Dela,Opis_Dela,Cena_Dela,Id_uredjaja)
VALUES (2, 'TRIMER VL 33K', 'Trimer VL 33 KO', 80, 2);

INSERT INTO delovi(Id_dela,Naziv_Dela,Opis_Dela,Cena_Dela,Id_uredjaja)
VALUES (3, 'PREKIDAC TS 503', 'Prekidac TS 503', 150, 1);

INSERT INTO delovi(Id_dela,Naziv_Dela,Opis_Dela,Cena_Dela,Id_uredjaja)
VALUES (4, 'ANTENA 2M 5/8', 'Antena 2m 5/8', 365, 1);

INSERT INTO delovi(Id_dela,Naziv_Dela,Opis_Dela,Cena_Dela,Id_uredjaja)
VALUES (5, 'DUGME 11313', 'DUGME 11313', 150, 1);

INSERT INTO delovi(Id_dela,Naziv_Dela,Opis_Dela,Cena_Dela,Id_uredjaja)
VALUES (6, 'KABL LICNA', 'crveni, 1 X 0.14', 150, 1);
```

GOTOVO za tabelu " delovi".

```
INSERT INTO uredjaji (Id_uredjaja, Ime, Tip, Proizvodjač, Id_klijenta,
Opis_Problema, DatumDostavke, DatumUgradnje, DatumZavršetka, VremeRada, Količina,
Id_servisera)
VALUES (1, 'TV', '123vc', 'Sony', 1, 'Nece da se upali', '5/24/2015', '5/24/2015',
'5/24/2015', 1, 1, 1);
```

```
INSERT INTO uredjaji (Id_uredjaja, Ime, Tip, Proizvodjač, Id_klijenta,
Opis_Problema, DatumDostavke, DatumUgradnje, DatumZavršetka, VremeRada, Količina,
Id_servisera)
VALUES (2, 'DVD', '010vc', 'Panasonic', 2, 'Ne cita disk', '5/24/2015', '5/24/2015',
'5/24/2015', 2, 1, 2);
```

GOTOVO za tabelu "uredjaji".

```
INSERT INTO delovi_dobavljači (Id_dela, Id_dobavljača)
VALUES (1, 1);

INSERT INTO delovi_dobavljači (Id_dela, Id_dobavljača)
VALUES (5, 1);

INSERT INTO delovi_dobavljači (Id_dela, Id_dobavljača)
VALUES (6, 1);

INSERT INTO delovi_dobavljači (Id_dela, Id_dobavljača)
VALUES (2, 2);

INSERT INTO delovi_dobavljači (Id_dela, Id_dobavljača)
VALUES (7, 2);

INSERT INTO delovi_dobavljači (Id_dela, Id_dobavljača)
VALUES (3, 3);
```

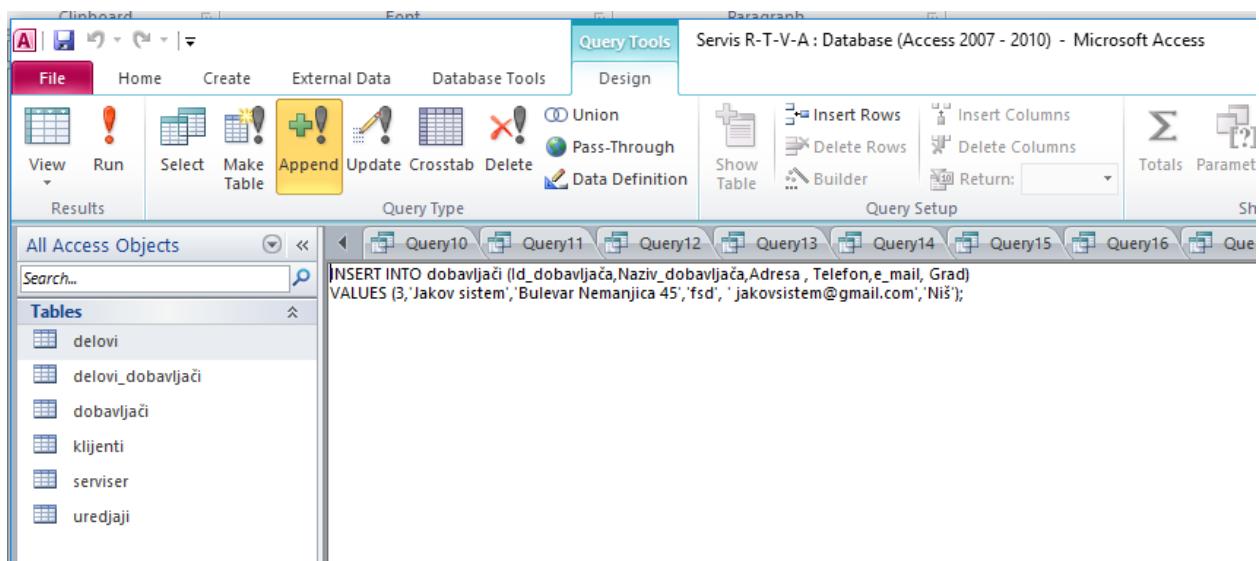
GOTOVO za tabelu "delovi_dobavljači".

```
INSERT INTO dobavljači (Id_dobavljača, Naziv_dobavljača, Adresa ,
Telefon, e_mail, Grad)
VALUES (1, 'Winwin', 'Bulevar Nemanjica 10', '018546545',
'winwin@gmail.com', 'Niš');

INSERT INTO dobavljači (Id_dobavljača, Naziv_dobavljača, Adresa ,
Telefon, e_mail, Grad)
VALUES (2, 'Puls', 'Kneza Milosa 41', 'fsd', 'puls@gmail.com', 'Pirot');

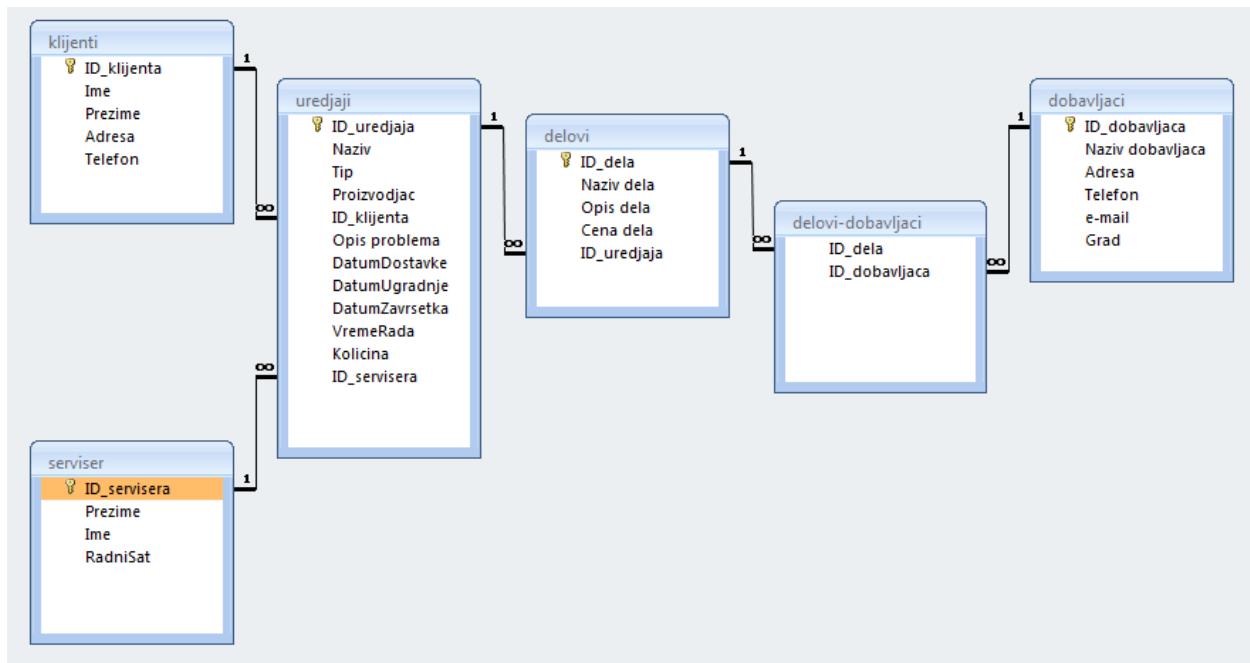
INSERT INTO dobavljači (Id_dobavljača, Naziv_dobavljača, Adresa ,
Telefon, e_mail, Grad)
VALUES (3, 'Jakov sistem', 'Bulevar Nemanjica 45', 'fsd', 'jakovsistem@gmail.com', 'Niš');
```

GOTOVO za tabelu "dobavljači".



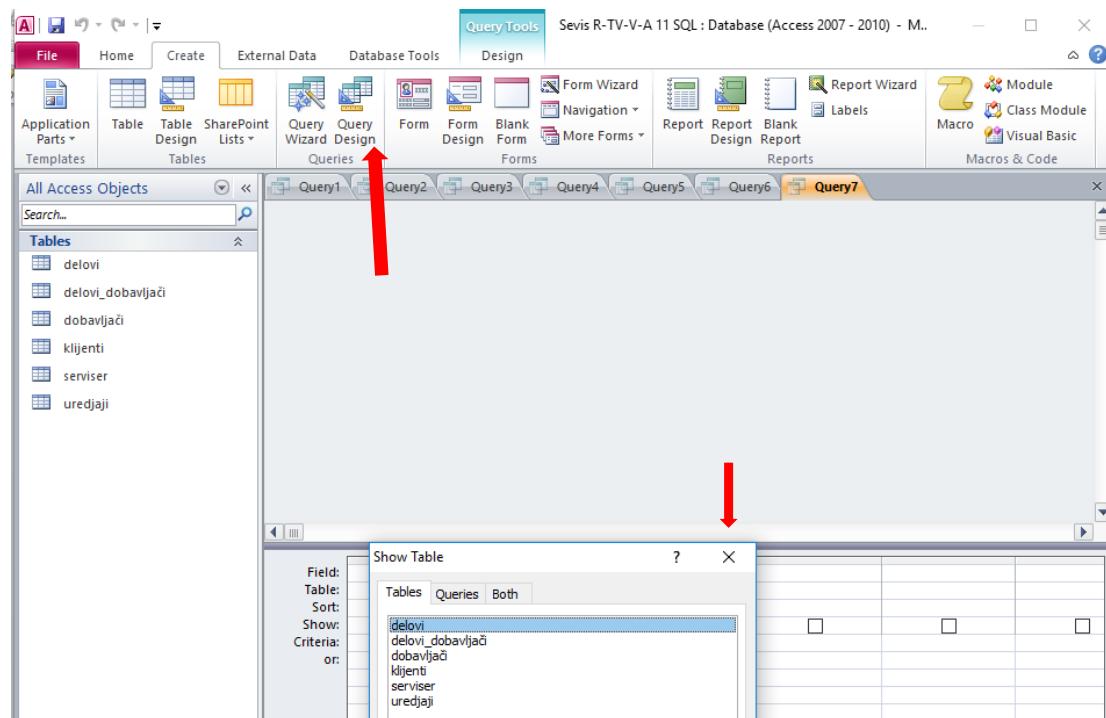
Na kraju smo korišćenjem SQL zapisa formirali sve tabele baze podataka I možemo obrisati sve Query(n) upite u bazu, desnim klikom na liniju Query1... QueryN opcijom Delete All.

14. Izvodjenje RELACIJA



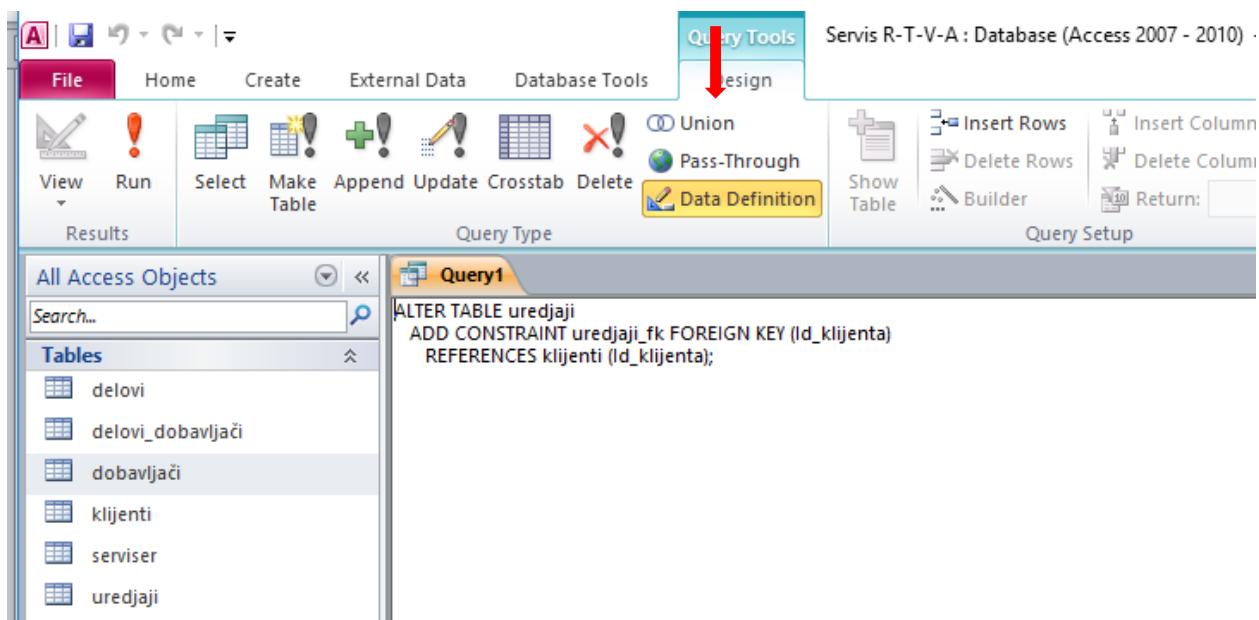
Po uradjenoj projektnoj dokumnetaciji relacije izmedju tabela su postavljene na način prikazan na slici.

Za izvodjenje relacija koristićemo sličan postupak kao u prethodnom delu. Primeničemo SQL komande za modifikaciju tabela i definisanje veza primarnih i sekundarnih ključeva:



Zatim unosimo SQL script za formiranje sekundarnih ključeva i relacija izmedju tabela, naprimer za tabelu "uredjaji". Izmeničemo tabelu "uredjaji" naredbom ALTER TABLE i u tabeli "uredjaji" naredbom ADD CONSTRAINT sad definišemo njen sekundarni ključ (Foreign Key) koji je u ovom slučaju "id_klijenta", (njegovo virtualno ime stavimo npr. "uredjaji_fk") koji treba da uspostavi relaciju naredbom REFERENCES sa tabelom "klijenti" preko njenog primarnog ključa "id_klijenta":

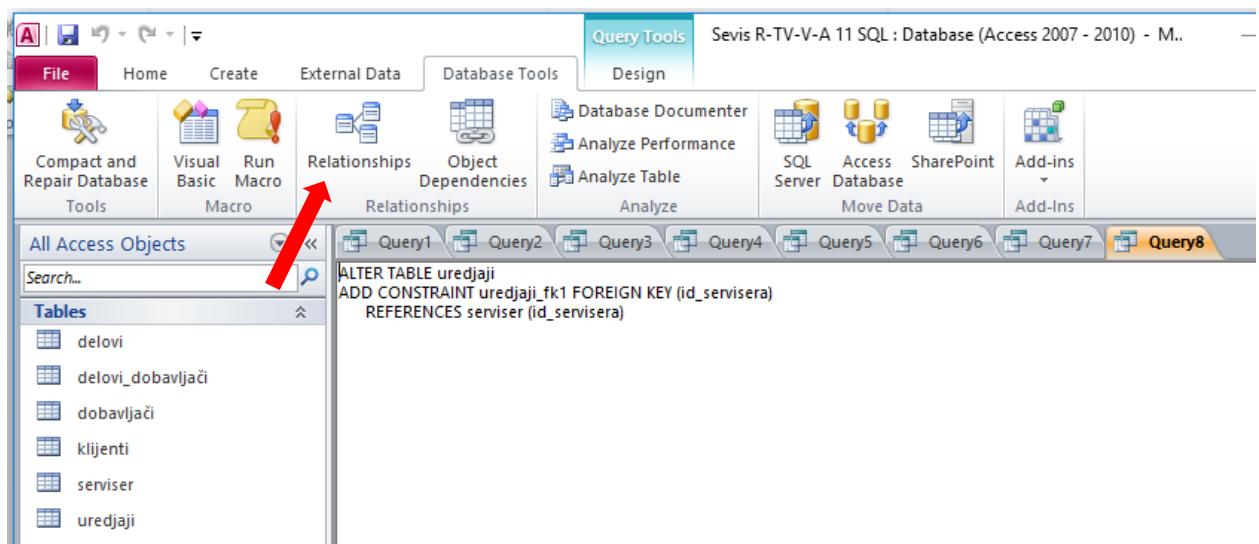
```
ALTER TABLE uredjaji
ADD CONSTRAINT uredjaji_fk FOREIGN KEY (Id_klijenta)
REFERENCES klijenti (Id_klijenta);
```



Skript se izvršava pritiskom ekranskog tastera RUN !!!

Ponavljamo postupak za sledeću sledeću novu relaciju:

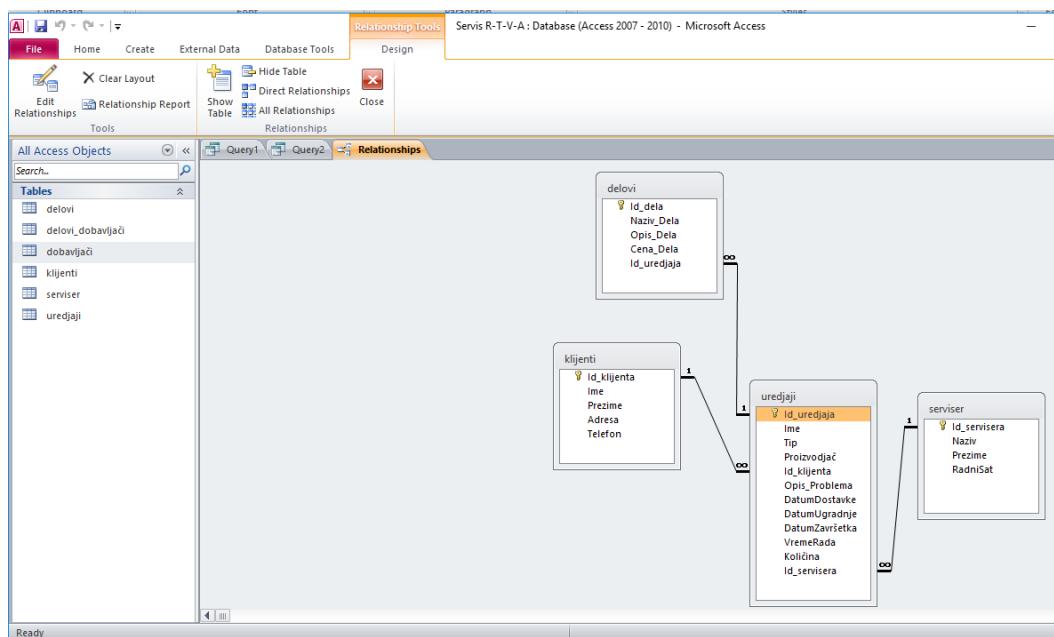
```
ALTER TABLE uredjaji
ADD CONSTRAINT uredjaji_fk1 FOREIGN KEY (Id_servisera)
REFERENCES serviser (Id_servisera);
```



```
ALTER TABLE delovi
ADD CONSTRAINT delovi_fk1 FOREIGN KEY (Id_uredjaja)
REFERENCES uredjaji (Id_uredjaja)
```

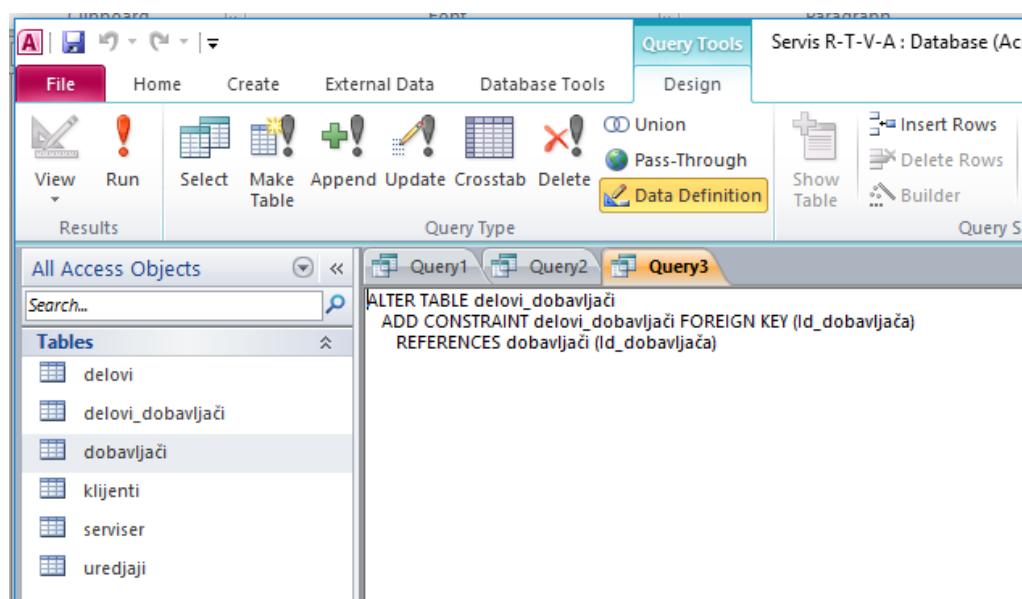
Da bi proverili šta smo do sada uradili sa relacijama i dobili grafički izgled tabela i relacija koristimo iz opcija glavnog menija opciju Database Tools i ekranski taster Relationships:

Dobijamo grafički prikaz do sada postavljenih relacija:



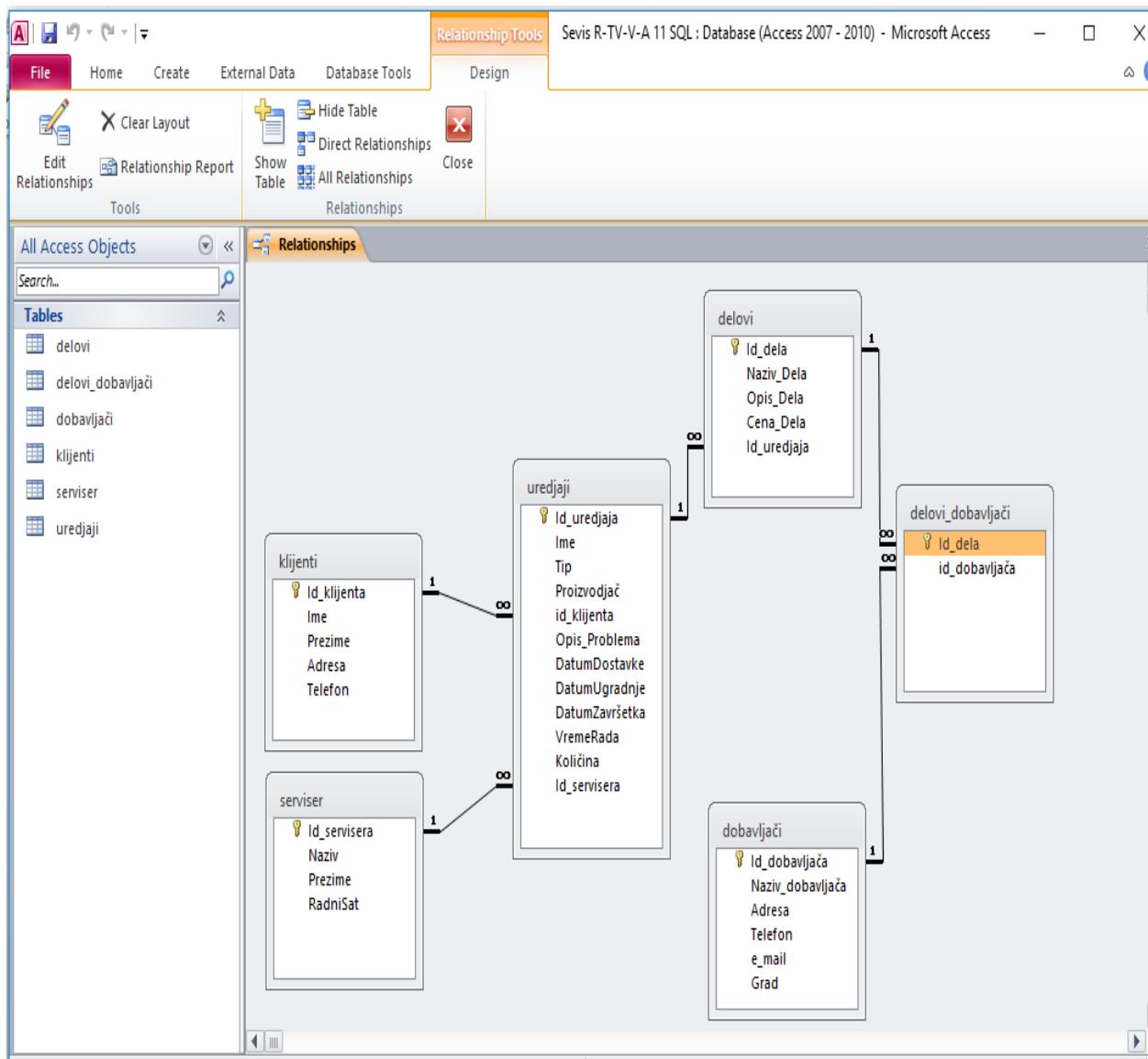
I tako dalje za sve relacije:

```
ALTER TABLE delovi_dobavljači
ADD CONSTRAINT delovi_dobavljači FOREIGN KEY (Id_dobavljača)
REFERENCES dobavljači (Id_dobavljača)
```



```
ALTER TABLE delovi_dobavljači
ADD CONSTRAINT delovi_dobavljači1 FOREIGN KEY (id_dela)
REFERENCES delovi (id_dela)
```

Na kraju dobijamo relacije uspostavljene nad bazom podataka koje izgledaju:



Na kraju možemo obrisati sve Query(n) i razmestiti tabele u prozoru Relationships jednostavim prevlačenjem na odgovarajuće mesto.

15. UPITI NAD BAZOM PODATAKA

UPIT1: Prikaži sve iz tabele "klijenti"

SELECT * FROM klijenti

	Id_klijenta	Ime	Prezime	Adresa	Telefon
1	Ivan	Manić	Vozdova 12	018125656	
2	Jovan	Cvetković	Pantelejska 45	018123456	

UPIT2: Prikaži sve iz tabele "dobavljači"

SELECT * FROM dobavljači

	Id_dobavljači	Naziv_doba	Adresa	Telefon	e_mail	Grad
1	Winwin	Bulevar Nemar	018546545		winwin@gmail.com	Niš
2	Puls	Kneza Milosa 4	fsd		puls@gmail.co	Pirot
3	Jakov sistem	Bulevar Nemar	fsd		jakovsistem@	Niš

Ovaj upit, kao i sve ostale možemo zapamtiti kada desnim tasterom miša kliknemo na upit (ovde je npr. Query1), Save As (damo ime npr. "Pokaži sve iz dobavljača") i overimo

tasterom OK. Tako opit staje stalno zabeležen u tabeli All Access Objects i može se uvek kada terba izvršiti.

The screenshot shows the Microsoft Access ribbon with the 'File' tab selected. In the main area, a query window titled 'Query1' is open, displaying the following SQL command:

```
Prikaži sve iz dobavljača
```

The results of the query are shown in a table with columns: Id_dobavljač, Naziv_dobav, Adresa, Telefon, e_mail, and Grad. The data is as follows:

Id_dobavljač	Naziv_dobav	Adresa	Telefon	e_mail	Grad
1	Winwin	Bulevar Nemar	018546545	winwin@gmail.com	Niš
2	Puls	Kneza Milosa 4	fsd	puls@gmail.co	Pirot
3	Jakov sistem	Bulevar Nemar	fsd	jakovsistem@...	Niš

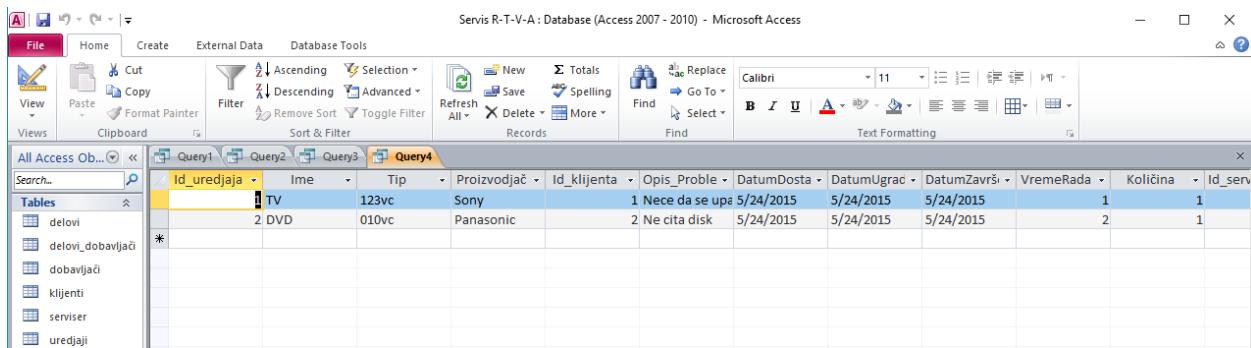
A 'Save As' dialog box is overlaid on the query window, containing the query name 'Prikaži sve iz dobavljača' and buttons for 'OK' and 'Cancel'.

The screenshot shows the Microsoft Access ribbon with the 'File' tab selected. In the main area, a query window titled 'Prikaži sve iz dobavljača' is open, displaying the same SQL command and results as the previous screenshot. The data is identical:

Id_dobavljač	Naziv_dobav	Adresa	Telefon	e_mail	Grad
1	Winwin	Bulevar Nemar	018546545	winwin@gmail.com	Niš
2	Puls	Kneza Milosa 4	fsd	puls@gmail.co	Pirot
3	Jakov sistem	Bulevar Nemar	fsd	jakovsistem@...	Niš

UPIT3: Prikaži sve iz tabele "uredjaji"

SELECT * FROM uredjaji

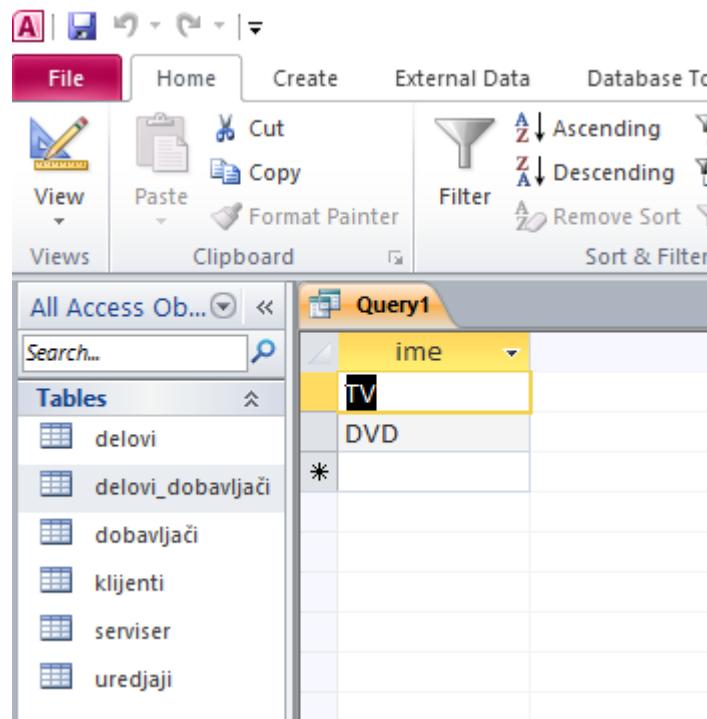


The screenshot shows the Microsoft Access ribbon with the 'Home' tab selected. Below the ribbon is a toolbar with various icons for operations like Cut, Copy, Paste, and Sort. The main area displays a grid of data from the 'uredjaji' table. The columns are labeled: Id_uredjaja, Ime, Tip, Proizvodjač, Id_klijenta, Opis_Proble, DatumDosta, DatumUgrad, DatumZavrš, VremeRada, Količina, and Id_serv. Two rows of data are visible: one for a TV and one for a DVD.

Id_uredjaja	Ime	Tip	Proizvodjač	Id_klijenta	Opis_Proble	DatumDosta	DatumUgrad	DatumZavrš	VremeRada	Količina	Id_serv
1	TV	123vc	Sony	1	Nece da se upa	5/24/2015	5/24/2015	5/24/2015	5/24/2015	1	1
2	DVD	010vc	Panasonic	2	Ne cita disk	5/24/2015	5/24/2015	5/24/2015	5/24/2015	2	1

UPIT4: Prikaži kolonu "ime" iz tabele "uredjaji"

SELECT ime FROM uredjaji



The screenshot shows the Microsoft Access ribbon with the 'Home' tab selected. Below the ribbon is a toolbar with various icons for operations like Cut, Copy, Paste, and Sort. The main area displays a grid of data from the 'uredjaji' table, specifically showing the 'ime' column. The data shows two entries: 'TV' and 'DVD'.

ime
TV
DVD

UPIT5: Prikaži kolone "Proizvodjač", "Opis_problema", "DatumZavršetka" iz tabele "uredjaji"

SELECT Proizvodjač, Opis_problema, DatumZavršetka FROM uredjaji

Proizvodjač	Opis_problema	DatumZavršetka
Sony	Nece da se upa	5/24/2015
Panasonic	Ne cita disk	5/24/2015
*		

UPIT6: Prikaži iz tabele "uredjaji" proizvodjača "Sony"

SELECT * FROM uredjaji
WHERE Proizvodjač='Sony';

Id_uredjaja	Ime	Tip	Proizvodjač	Id_klijenta	Opis_Proble	DatumDosta	DatumUgrad	DatumZavrši	VremeRada	Količina	Id_se
1	TV	123vc	Sony	1	Nece da se upa	5/24/2015	5/24/2015	5/24/2015	1	1	1
*											

UPIT7: Prikaži ime i prezime iz tabele "klijenti" gde je prezime "Cvetković"

```
SELECT ime,prezime FROM klijenti
WHERE Prezime='Cvetković' ;
```

ime	prezime
Jovan	Cvetković

UPIT8: Prikaži ime,Opis_problema iz tabele "uredjaji" gde je VremeRada manje ili jednako 2 i količina menje ili jednako 1.

```
SELECT ime,Opis_problema FROM uredjaji WHERE VremeRada<=2 AND kolicina <=1;
```

ime	Opis_problema
TV	Nece da se upali
DVD	Ne cita disk

UPIT9: Prikaži prezime iz tabele "klijenti" i Opis_Problema iz tabele "uredjaji" i poveži te dve tabele sa njihovim ključevima i atributima

```
SELECT prezime, Opis_Problema
FROM klijenti, uredjaji
WHERE klijenti.Id_klijenta = uredjaji.Id_klijenta
```

UPIT10: Prikaži sve objekte tabele "klijenti" spojenu sa tabelom "uredjaji". Vezu uspostaviti preko njihovih ključeva i atributa

```
SELECT * FROM klijenti INNER JOIN uredjaji
ON klijenti.Id_klijenta=uredjaji.Id_klijenta
```

UPIT11: Dodati u tabelu "klijenti" atribute i novu kolonu "Grad". Postojeća tabela izgleda:

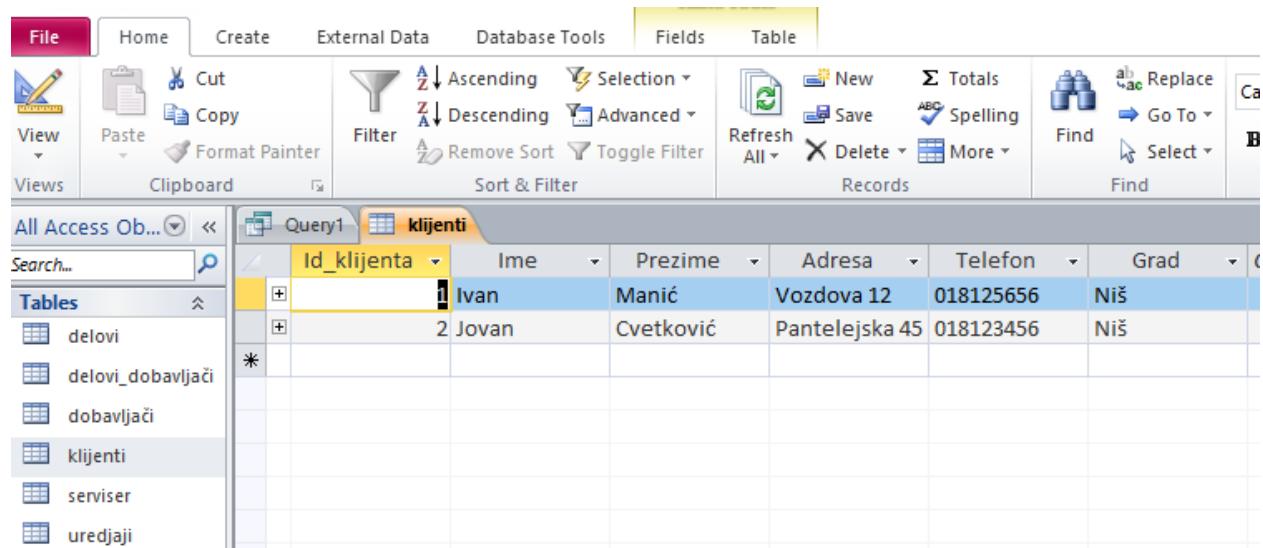
	Id_klijenta	Ime	Prezime	Adresa	Telefon	Click to Add
		Ivan	Manić	Vozdova 12	018125656	
*		2	Jovan	Cvetković	Pantelejska 45	018123456

ALTER TABLE klijenti
ADD Grad TEXT(20);

	Id_klijenta	Ime	Prezime	Adresa	Telefon	Grad	Click to Add
		Ivan	Manić	Vozdova 12	018125656		
*		2	Jovan	Cvetković	Pantelejska 45	018123456	

UPDATE klijenti SET Grad = 'Niš' WHERE Id_klijenta = 1;	UPDATE klijenti SET Grad = 'Niš' WHERE Id_klijenta = 2;
---	---

Izvršavamo ih postupno, jedan po jedan !!!



The screenshot shows the Microsoft Access ribbon with the 'File' tab selected. Below the ribbon is a search bar and a 'Tables' list containing 'delovi', 'delovi_dobavljači', 'dobavljači', 'klijenti', 'serviser', and 'uredjaji'. The main area displays a query results grid titled 'Query1' for the 'klijenti' table. The columns are 'Id_klijenta', 'Ime', 'Prezime', 'Adresa', 'Telefon', and 'Grad'. Two rows are visible: one for 'Ivan Manić' and another for 'Jovan Cvetković'. The 'Id_klijenta' column is sorted ascendingly.

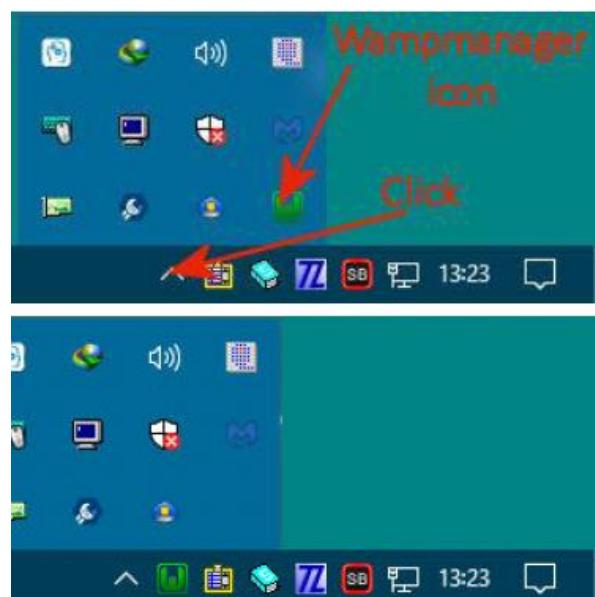
16. Neka objašnjenja o tome kako Vampserver funkcioniše:

Nakon što pokrenete Vampserver preko njegove prečice, videćete kratku pojavu prozora, a zatim morate da dobijete ikonu Vampmanager-a u polju za obaveštenja u donjem desnom uglu ekrana. Ova ikona mora biti **ZELENA**.

U nekim retkim slučajevima, ova ikona se ne pojavljuje, ali se nalazi u oblasti skrivenih ikona. Zatim je potrebno pogledati u prozor skrivenih ikona oblasti za obaveštavanje: Kliknite na vertikalni ševron levo od oblasti za obaveštenja: zatim prevucite i otpustite ikonu Vampmanagera u oblast za obaveštavanja: Ova ikona može biti u tri boje:

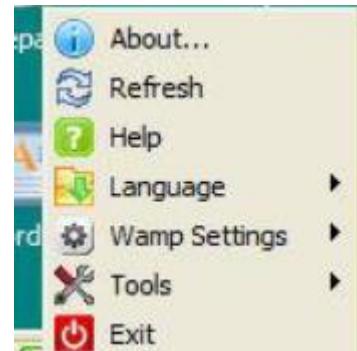


Crvena znači da niješta usluga nije pokrenuta, narandžasta znači da je pokrenuta najmanje jedna usluga, a zelena znači da su svi servisi pokrenuti. Da biste mogli da koristite Vampserver, neophodno je da ikona bude zelena. Nema smisla raditi Apache, PHP, MySQL ili MariaDB operacije dok ikona ne postane zelena. Ako vaša ikona nije zelena.



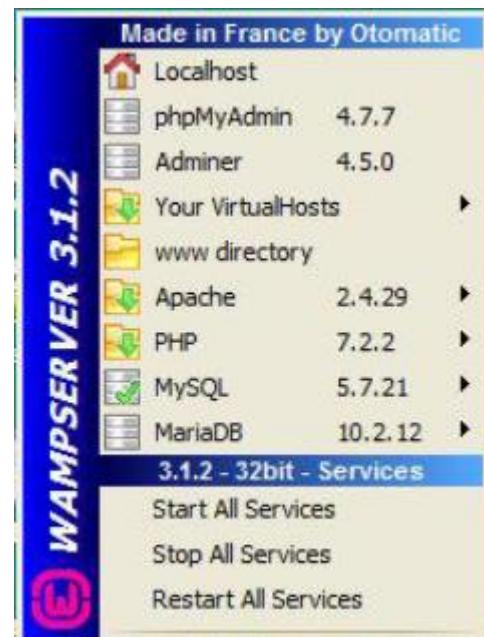
- Desnim klikom na ovu ikonu dobijate pristup:

- About: korišćene verzije i druge informacije n
- Refresh: Ponovo kreira vampmanager .ini datoteku
- Help: Pristup Vampserver forumima pomoći
- Language: Change the language used
- Wamp parameters: Omogućava vam da izmenite određeni Vampserver opcije
 - Tools: Uslužni programi za rešavanje problema
 - Close: Uslužni programi za rešavanje problema Neophodni za zatvaranje Vampserver-a. Mora se uraditi PRE zatvaranja Windovs-a da bi se omogućilo izbacivanje PHP, MySql i MariaDB keš memorije i izbegao gubitak podataka.



•Kliknite levim tasterom miša na ovu ikonu daje vam pristup:

- Prikazuje brojke korišćenih verzija
 - Direktan pristup početnoj stranici Vampserver-a (localhost)
 - Direktan pristup PhpMyAdmin (MySQL / MariaDB GUI)*
 - Direktan pristup Adminer (MySQL / MariaDB GUI)*
- * Može se omogućiti ili onemogućiti u Vamp podešavanjima.
- Podmeni Virtualhosts
 - Otvorite File Explorer u fascikli WWW
 - Različiti podmeniji za Apache, PHP, MySQL i MariaDB koji omogućavaju promenu verzije; dodati ili ukloniti učitavanje modula ili proširenja; modifikovati parametre; administriranje usluga za svaki entitet; otvorite neke datoteke u podrazumevanom uređivaču teksta (php.ini, httpd.conf, https-vhosts.conf, mi.ini, datoteke evidencije, itd.)
 - Upravljanje uslugama: Stop, Start, Restart



Pokrenite phpMyAdmin Kada pokrenete phpMyAdmin, od vas će se tražiti korisničko ime i lozinka. Nakon instaliranja Vampserver 3, podrazumevano korisničko ime je "root" (bez navodnika) i nema lozinke, što znači da morate da ostavite polje za lozinku praznom. Biće upozorenje: Povezani ste kao 'root' bez lozinke, što odgovara podrazumevanom MySQL privilegovanim nalogu.

Vaš MySQL server radi sa ovim podrazumevanim, otvoren je za rad, ali bi zaista trebalo da popravite ovu bezbednosnu rupu postavljanjem lozinke za korisnika 'root'. Ovo nije problem sve dok pristup MyAdmin-u ostaje lokalno. Međutim, neke veb aplikacije ili CMS traže da MySQL korisnik ima lozinku. U tom slučaju, kreirajte korisnika sa lozinkom preko kartice Korisnici naloga MyAdmin.

Podrška MariaDB-a

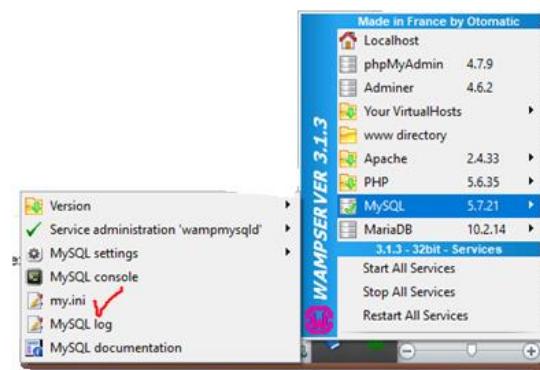
Pročitajte datoteku „mariadb_support.tkt“ u vamp instalacionim direktorijumu

Korišćenje menija i podmenija Vampmanager-a

Nemojte koristiti tastaturu za navigaciju kroz menije i podmenije ikone Vampmanager.

Ako vam se port WAMP servera "gadja" sa portom od nekih, kod vas, instaliranih programa uradite sledeće:

1. Desni klik na WampServer ikonu.
2. Izaberemo Mysql=>my.ini fajl.



3. Otvorimo my.ini fajl.
4. Promenimo mysql port. To radimo u liniji. # The MySQL server. [wampmysqld] port = 3306. # # [mysqld] ...

```
my.ini - Notepad
File Edit Format View Help
: Example MySQL config file for medium systems.

: This is for a large system with memory of 1G-2G where the system runs me
: MySQL.

: MySQL programs look for option files in a set of
: locations which depend on the deployment platform.
: You can copy this option file to one of those
: locations. For information about these locations, see:
: http://dev.mysql.com/doc/mysql/en/option-files.html

: In this file, you can use all long options that a program supports.
: If you want to know which options a program supports, run the program
: with the "--help" option.

: The following options will be passed to all MySQL clients
[client]
password = your_password
port = 3306
socket = /tmp/mysql.sock

: Here follows entries for some specific programs

: The MySQL server
[wampmysqld]
skip-grant-tables
port = 3306
socket = /tmp/mysql.sock
key_buffer_size = 256M
max_allowed_packet = 1M
bind-address = 0.0.0.0
:Added to reduce memory used (minimum is 400)
table_definition_cache = 600
```

5. Menjamo instance i unosimo drugi port npr. 3307
6. Pamtimmo my.ini fajl,
7. Restartujemo Wamp MYSQL.
8. Gotovo.

Sada pristupamo organizaciji datoteke naše baze podataka Servis_R_T_V_A, u okviru WAMP servera i opcije phpMyAdmin, koju na osnovu prethodnog rada pamtimmo u Word Pad_u kao Servis_R_T_V_A.sql. na odgovarajućem direktorijumu.

Ona sada izgleda ovako:

```
-- phpMyAdmin SQL Dump
-- version 4.7.9
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1:3306
-- Generation Time: Apr 25, 2022 at 11:10 AM
-- Server version: 5.7.21
-- PHP Version: 5.6.35

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

-- Ovaj prethodni deo za setovanje ostaviti onakav kakav je !!!
-- Database: `Servis_R_T_V_A`
--

-----
-- Struktura tabele `klijenti`
--

DROP TABLE IF EXISTS `klijenti`;
CREATE TABLE IF NOT EXISTS `klijenti`(
(`Id_klijenta` int(10) NOT NULL,
`Ime` char(30) NOT NULL DEFAULT '',
`Prezime` char(30) NOT NULL DEFAULT '',
`Adresa` char(30) NOT NULL DEFAULT '',
`Telefon` char(30) NOT NULL DEFAULT '',
PRIMARY KEY (`Id_klijenta`))
ENGINE=InnoDB AUTO_INCREMENT=4080 DEFAULT CHARSET=latin1;
-- 
-- Struktura tabele `serviser`
-- 

DROP TABLE IF EXISTS `serviser`;
```

```

CREATE TABLE IF NOT EXISTS `serviser`
(`Id_servisera` int(10) NOT NULL,
 `Naziv` char(30) NOT NULL DEFAULT '',
 `Prezime` char(30) NOT NULL DEFAULT '',
 `RadniSat` int(10) NOT NULL,
 PRIMARY KEY (`Id_servisera`))
ENGINE=InnoDB AUTO_INCREMENT=4080 DEFAULT CHARSET=latin1;
-- 
-- Struktura tabele `delovi`
-- 

DROP TABLE IF EXISTS `delovi`;
CREATE TABLE IF NOT EXISTS `delovi`
(`Id_dela` int(10) NOT NULL,
 `Naziv_Dela` char(30) NOT NULL DEFAULT '',
 `Opis_Dela` char(30) NOT NULL DEFAULT '',
 `Cena_Dela` int(10) NOT NULL,
 `Id_uredjaja` int(10) NOT NULL,
 PRIMARY KEY (`Id_dela`))
ENGINE=InnoDB AUTO_INCREMENT=4080 DEFAULT CHARSET=latin1;
-- 

-- Struktura tabele `uredjaji`
-- 

DROP TABLE IF EXISTS `uredjaji`;
CREATE TABLE IF NOT EXISTS `uredjaji`
(`Id_uredjaja` int(10) NOT NULL,
 `Ime` char(30) NOT NULL DEFAULT '',
 `Tip` char(30) NOT NULL DEFAULT '',
 `Proizvodjac` char(30) NOT NULL DEFAULT '',
 `Id_klijenta` int(10) NOT NULL,
 `Opis_Problema` char(30) NOT NULL DEFAULT '',
 `DatumDostavke` char(30) NOT NULL DEFAULT '',
 `DatumUgradnje` char(30) NOT NULL DEFAULT '',
 `DatumZavršetka` char(30) NOT NULL DEFAULT '',
 `VremeRada` int(10) NOT NULL,
 `Kolicina` int(10) NOT NULL,
 `Id_servisera` int(10) NOT NULL,
 PRIMARY KEY (`Id_uredjaja`))
ENGINE=InnoDB AUTO_INCREMENT=4080 DEFAULT CHARSET=latin1;
-- 

-- Struktura tabele `dobavljac`
-- 

DROP TABLE IF EXISTS `dobavljac`;
CREATE TABLE IF NOT EXISTS `dobavljac`
(`Id_dobavljaca` int(10) NOT NULL,
 `Naziv_dobavljaca` char(30) NOT NULL DEFAULT '',
 `Adresa` char(30) NOT NULL DEFAULT '',
 `Telefon` char(30) NOT NULL DEFAULT '',
 `e_mail` char(30) NOT NULL DEFAULT '',
 `Grad` char(30) NOT NULL DEFAULT '',
 PRIMARY KEY (`Id_dobavljaca`))
ENGINE=InnoDB AUTO_INCREMENT=4080 DEFAULT CHARSET=latin1;
-- 

-- Struktura tabele `delovi_dobavljac`
-- 

DROP TABLE IF EXISTS `delovi_dobavljac`;

```

```

CREATE TABLE IF NOT EXISTS `delovi_dobavljenici`
(`Id_dela` int(10) NOT NULL,
 `Id_dobavljenica` int(10) NOT NULL,
 PRIMARY KEY (`Id_dela`))
ENGINE=InnoDB AUTO_INCREMENT=4080 DEFAULT CHARSET=latin1;
--



-- Skladištenje podataka u tabelu `klijenti`
-- 
INSERT INTO `klijenti` ('Id_klijenta', `Ime`, `Prezime`, `Adresa`, `Telefon`) VALUES
(1,'Ivan','Manic','Vozdova 12','018125656'),
(2,'Jovan','Cvetkovic','Pantelejska 45','018123456');

-- 
-- Skladištenje podataka u tabelu `serviser`
-- 
INSERT INTO `serviser` ('Id_servisera', `Naziv`, `Prezime`, `RadniSat`) VALUES
(1,'Ciric','Miljan', 200),
(2,'Pešić','Bratislav', 200);

-- Skladištenje podataka u tabelu `delovi`
-- 
INSERT INTO `delovi` ('Id_dela', `Naziv_Dela`, `Opis_Dela`, `Cena_Dela`, `Id_uredjaja`)
VALUES
(1,'Otpornik','1/4W 51K',100,2),
(2,'TRIMER VL 33K', 'Trimer VL 33 KO',80, 2),
(3,'PREKIDAC TS 503', 'Prekidac TS 503',150, 1),
(4, 'ANTENA 2M 5/8', 'Antena 2m 5/8', 365, 1),
(5, 'DUGME 11313', 'DUGME 11313', 150, 1),
(6, 'KABL LICNA', 'crveni, 1 X 0.14', 150, 1);

-- Skladištenje podataka u tabelu `uredjaji`
-- 
INSERT INTO `uredjaji`(`Id_uredjaja`, `Ime`, `Tip`, `Proizvodjac`, `Id_klijenta`, `Opis_Problema`, `DatumDostavke`,
`DatumUgradnje`, `DatumZavršetka`, `VremeRada`, `Kolicina`, `Id_servisera` )
VALUES
(1,'TV','123vc','Sony',1,'Nece da se upali','5/24/2022','5/24/2022', '5/24/2022',1,1,1),
(2,'DVD','010vc','Panasonic',2,'Ne cita disk','5/24/2022','5/24/2022', '5/24/2022',2,1,2);

-- 
-- 

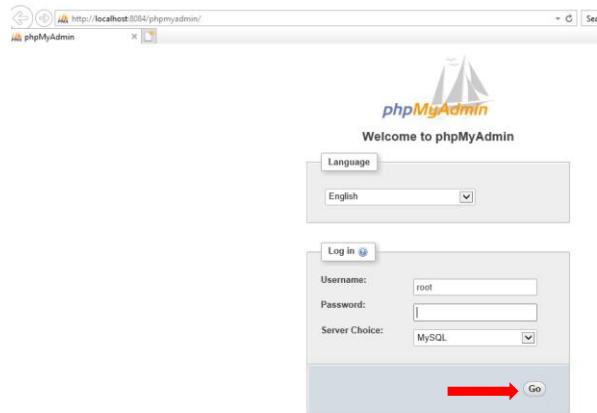
-- Skladištenje podataka u tabelu `dobavljenici`
-- 
INSERT INTO `dobavljenici`(`Id_dobavljenica`, `Naziv_dobavljenica`, `Adresa`, `Telefon`, `e_mail`, `Grad`)
VALUES
(1,'Winwin','Bulevar Nemanjica 10','018546545', 'winwin@gmail.com','Niš'),
(2,'Puls','Kneza Milosa 41','fsd', 'puls@gmail.com','Pirot'),
(3,'Jakov sistem','Bulevar Nemanjica 45','fsd', 'jakovsistem@gmail.com','Niš');

-- Skladištenje podataka u tabelu `delovi_dobavljenici`
-- 
INSERT INTO `delovi_dobavljenici`(`Id_dela`, `Id_dobavljenica`)
VALUES
(1,1),
(5,1),

```

(6,1),
(2,2),
(5,2),
(3,1),
(1,3);

Pokrećemo WAMP server i u okviru menija phpMyAdmin. Dobijamo osnovni prozor sa opcijama za logovanje. Po difoltu Username je root pa samo izvršavamo prijavu sa Go !!!:



Dobijamo glavni prozor phpMyAdmin i sada moramo definisati ime baze. Rekli smo da je to Servis_R_T_V_A. Moramo voditi računa da ovde nije moguće ostavljati prazna mesta za opis objekata I da ima razlike u apostrofu za definisanje tabela i podataka !!! Otvaramo našu bazu podataka opcijom New iz padajućeg menija:

Kreiramo bazu podataka ekranskim tasterum Create !!!

The screenshot shows the 'Databases' section of phpMyAdmin. On the left is a tree view of databases: New, information_schema, mysql, performance_schema, portal, servis_r_t_v_a, sys, videotekuhdd, videotekubssd, world, and zaposleni. In the main area, there is a table with columns 'Database', 'Collation', and 'Action'. A new database 'Servis_R_T_V_A' has been created and is listed. The 'Collation' for this database is set to 'latin1_swedish_ci'. A red arrow points to the 'Create' button at the top right of the table.

Inicijalna baza je kreirana kao što se vidi sa slike. Sada je potrebno importovati naš sql fajl koji samo zapamtili na odredjenom direktorijumu, na osnovu koga iniciramo sve tabele , punimo tabele i izvodimo relacije:

Iz glavnog menija izaberemo Import i kroz opciju Browse nalazimo naš fajl na odredjenom direktorijumu. Tada tasterom Go importujemo bazu u phpMyAdmin - localhost okruženje.

The screenshot shows the 'Import' page of phpMyAdmin. The 'File to import:' section is visible, showing the file path 'C:\Users\IBORA\Desktop\' and a 'Character set of the file:' dropdown set to 'utf-8'. Other sections like 'Partial import:', 'Other options:', 'Format:', and 'Format-specific options:' are also present.

Table	Action	Rows	Type	Collation	Size	Overhead
delovi	Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	32 KiB	-
delovi_dobavljac	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	32 KiB	-
dobavljaci	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 KiB	-
klijenti	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16 KiB	-
serviser	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16 KiB	-
uredjaji	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	48 KiB	-
6 tables	Sum		15	MyISAM	latin1_swedish_ci	160 KiB

Sadržaje datoteka možemo videti dvostrukim klikom na odgovarajuću tabelu u padajućem meniju:

Id_dela	Naziv_Dela	Opis_Dela	Cena_Dela	Id_uredjaja
1	Otpornik	1/4W 51K	100	2
2	TRIMER VL 33K	Trimer VL 33 KO	80	2
3	PREKIDAC TS 503	Prekidac TS 503	150	1
4	ANTENA 2M 5/8	Antena 2m 5/8	365	1
5	DUGME 11313	DUGME 11313	150	1
6	KABL LICNA	crveni, 1 X 0.14	150	1

U okviru ovog grafičkog okruženja sada možemo dopunjavati tabele, menjati ih, uspostavljati nove relacije i što je najvažnije izvršavati upite:

17. ZAKLJUČAK

Baza podataka predstavlja integrisani skup podataka o nekom sistemu i skup postupaka za njihovo održavanje i korišćenje, organizovan prema potrebama korisnika. Baza podataka je dobro struktuirana kolekcija podataka, koja postoji jedno određeno vreme, koja se održava i koju koristi više korisnika ili programa.

U ovom radu prikazana je baza podataka "Servis računara, televizora, video i audio uređaja". Osnovna namena ove baze je da čuva podatke o klijentima koji su doneli uređaje na servis, podatke o servisiranim udeđajima, cenu servisa, kao i cenu i katalog delova koji su dostavljeni od dobavljača servisu. Na početnu rada smo videli šta je Microsoft Access i koji su osnovni elementi za kreiranje baze. Zatim smo videli kako se planira osnova za izradu baze, kako se kreiraju tabele i kako se povezuju. Posle toga prikazan je i izgled prvih zapisa u tabelama, kako se kreiraju upiti, forme, izveštaji i makro programi.

Na kraju možemo zaključiti da je za savladavanje Access-a potrebno više vremena nego za sve ostale programe Office paketa. Kreiranje baze, kakva je ova za servis računara, televizora, video i audio uređaja ne zahteva veliko poznavanje programa, jer većinu poslova mogu pokriti čarobnjaci. Rezultate rada čarobnjaka ćemo izmeniti po potrebi i program je tu. Ostaje da se malo programira, šminkaju forme, prilagođavaju izveštaji. Taj posao, kao sada mnogo lakši i u grafičkom okruženju, uradili smo preko WAMP servera – jedne od najboljih aplikacija za organizovanje distribuirane baze podataka.