

Access

1. OSNOVNI POJMOVI O BAZAMA PODATAKA

Baza podataka služi za smeštanje informacija. Informacije su vezane za neku namenu i smeštaju se u bazu podataka u tabelama. Informacije u tabeli su organizovane u poljima (polje se nalazi u preseku neke kolone i nekog reda u tabeli).

Jedana od primera baze podataka može biti adresar. Različite kategorije informacija se mogu smestiti u adresar. Ako adresar posmatramo kao tabelu, tada se u poljima te tabele mogu naći informacije o imenu i prezimenu određene osobe, informacije o njegovoj adresi, telefonskom broju itd. Svaki unos podataka u adresar (odnosno u tabelu) se naziva zapis. Primer izgleda jedne ovakve tabele je dat na slici 1.

rb	Prezime	Ime	Ulica	Mesto	Post. br.	Telefon
1	Petrović	Petar	Maksima Gorkog	Novi Sad	21000	0211234567
...
10	Marković	Marko	Futoška	Novi Sad	21000	0219876543

Slika 1. Primer baze podataka – adresar

U tabelu se može vršiti upis i snimanje novih zapisa, pretraživanje svih postojećih zapisa radi izmene podataka u poljima ili radi brisanja zapisa.

Ovako isprojektovana baza podataka spada u grupu onih baza kod kojih se svi podaci čuvaju u jednoj tabeli.

Za razliku od ovakvih baza podataka postoje i relacione baze podataka. One su kompleksnije za projektovanje, međutim zauzvrat nude mnogo veću funkcionalnost. Microsoft Access služi upravo za izgradnju relacionih baza podataka.

Razliku između relacionih baza podataka i baza kod kojih se podaci čuvaju u jednoj tabeli možemo objasniti i na primeru baze podataka o studentima.

1.1. OBIČNA BAZA PODATAKA SA JEDNOM TABELOM

Student	Fakultet	Odsek	Smer	Ocene po predmetima
Petrović Petar	Fakultet tehničkih nauka	IIM	I1	APP=10, ORTP=9, ENGL_2=7, OSN_T_SISTEMA=8
Marković Marko	Fakultet tehničkih nauka	IIM	I2	ORTP=8, ENGL. 2=9
.....

Problem koji se javlja ovde je da se u različitim zapisima u tabeli, polje **Ocene po predmetima** ima različitu strukturu. Kod studenta Petrović Petra se mogu upisati ocene za četiri predmeta dok kod studenta Marković Marka samo dva.

Pravilo koje je potrebno poštovati prilikom izgradnje baze podataka je da svaki zapis nudi mogućnost da se u njegovim poljima može upisati ista količina podataka, odnosno da svako polje ima istu strukturu.

Ovaj problem se može rešiti na način prikazan u sledećoj tabeli.

Student	Fakultet	Odsek	Smer	Predmet	Ocena
Petrović Petar	Fakultet tehničkih nauka	IIM	I1	APP	10
Petrović Petar	Fakultet tehničkih nauka	IIM	I1	ORTP	9
Petrović Petar	Fakultet tehničkih nauka	IIM	I1	ENGL_2	7
.....
Marković Marko	Fakultet tehničkih nauka	IIM	I2	ORTP	8
Marković Marko	Fakultet tehničkih nauka	IIM	I2	ENGL_2	9

Sada svako polje ima istu strukturu i baza je korektno formirana.

Međutim, javlja se pojava višestrukog unosa istih podataka u bazi, što zauzima memorijski prostor računara na kojem se nalazi baza podataka.

Primer 1 : Ukoliko student Petrović Petar promeni smer i pređe iz I1 u I2 tada je na svim mestima u bazi podataka na kojima je upisan Petrović Petar neophodno promeniti podatke u polju smer. Prepostavimo da se u bazi nalazi više hiljada studenata i da za svakog postoji više zapisa. Ako svake godine stotinu studenata promeni smer i pređe na drugi, tada je potrebno da sistem za upravljanje bazom podataka pretraži celu bazu i izvrši izmene u svakom pojedinacnom polju u koloni **Smer**.

Primer 2 : Ukoliko se na promeni naziv predmeta potrebno je tada izmeniti isti u poljima kolone **Predmet**.

Mogućnost pojave nekozistetnosti podataka u ovim slučajevima je velika.

1.2. RELACIONA BAZA PODATAKA

Rešenje prethodno navedenih problema se ostvaruje putem relacionih baza podataka:

- svaki student se može predstaviti kao entitet,
- svaki predmet se može predstaviti kao entitet,
- svaka ocena je **veza** između studenta i predmeta,

stoga potrebne su i tri tabele :

- **tabela Student**

ID_Studenta	Student	G_Studija	Odsek	Smer
1020	Petrović Petar	Druga	IIM	I1
1067	Marković Marko	Druga	IIM	I2
.....

- **tabela Predmet**

ID_Predmet	Naziv_Predmeta	Godina	Br._Casova
001	Automatizacija procesa poslovanja	Druga	2+4
002	Engleski jezik 2	Druga	2+2
003	Osnovi računarskih tehnologija	Prva	2+2
.....

- **tabela Ocena**

ID_Student	ID_Predmet	Ocena
1020	001	10
1020	002	7
1067	002	9
.....

Svaki student u tabeli **Student** ima svoje **ključno polje po kojem se može identifikovati**, odnosno svoj **ID_Studenta** (broj indeksa) koje garantuje da će se studenti u tabeli razlikovati. Za identifikaciju nekog studenta u tabeli nije uzeto njegovo ime i prezime iz razloga što na primer na istoj godini na istom smeru mogu postojati dva studenta koja mogu imati isto ime i prezime.

Svaki predmet u tabeli **Predmet** ima svoje **ključno polje po kojem se može identifikovati**, odnosno svoj **ID_Predmet** koje garantuje da će se predmeti u tabeli razlikovati.

U tabeli **Ocena** veoma lako povezujemo studente i predmete preko vrednosti iz njihovih ključnih polja.

Prednosti relacionih baza podataka su objašnjene u primeru 3.

Primer 3 : Ukoliko neki student odluči da promeni smer i pređe sa I2 na I1, tada je tu promenu potrebno izvršiti **samo na jednom mestu u tabeli Student**. Određenog studenta je **jednostavno pronaći** preko njegovog identifikatora i tada se može izmeniti podatak u polju Smer za tog studenta.

Uvođenjem relacionih baza podataka **fleksibilniji je rad sa podacima** (unos novih podataka, izmena ili brisanje postojećih podataka) i **smanjuje se mogućnost pojave potencijalnih grešaka u podacima**.

Pre nego što se pređe na izgradnju baze podataka u Microsoft Accessu, neophodno je uraditi sledeće :

- odrediti svrhu baze podataka koju želimo da napravimo,
- odrediti tabele koje će činiti bazu podataka,
- odrediti polja koja želimo da se nalaze u tabelama,
- identifikovati koja će to polja biti ključna polja u tabelama (nazivaju se **primarnim ključem tabele**),
- odrediti veze između tabela (veze se nazivaju još i **relacijama**),
- pregledati još jednom ono što je prethodno osmišljeno i po potrebi preraditi određene delove buduće baze podataka (tabele, polja ili relacije),
- osmisliti forme, upite i izveštaje.

2. PRIMER BAZE PODATAKA – MENTORSKO PRAĆENJE STUDENATA

Pre prelaska na Microsoft Access potrebno je proći sve korake koji se odnose na definisanje onoga šta želimo da uradimo i na koji način to možemo da realizujemo.

Svrha baze podataka je da omogući mentorsko praćenje studenata. Da bi to bilo moguće potrebno je napraviti sledeće tabele:

- Osnovna tabela je ona koja se odnosi na studente, jer se želi omogućiti njihovo mentorsko praćenje u roku njihovog studiranja. Tu tabelu možemo nazvati **Student**.
- Svaki student je upisan na određenom smeru i zato pravimo još jednu tabelu koja će se zvati **Smer**.
- Svaki student se nalazi u određenom semestru i zato je potrebno napraviti tabelu **Semestar**.
- Studenti slušaju predmete i pravimo još jednu tabelu koja se zove **Predmet**.
- Za svaki predmet su predviđene ocene od 6 do 10 i pravimo tabelu **Ocena**.
- Određeni student iz nekog određenog predmeta ima svoju ocenu sa ispita i zato se pravi tabela **Student_Predmet**.

Kada su određene sve potrebne tabele neophodno je da budu definisana polja koja treba da se nalaze u tabelama. Svako polje mora imati svoj tip i podtip podatka iz razloga što će se u polja unositi različite vrste podataka. Na primer podatak u nekom polju brojčana vrednost (Number) onda u podtipu moramo reći o kom se tipu brojeva radi(npr. Integer). Može se takođe definisati i željena veličina polja (npr. u neko polje je moguće uneti broj od 5 cifara itd.).

Tabela Student

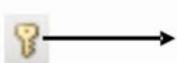


Naziv polja	Tip podatka	Podtip	Veličina
Index	Text	-	5
Prezime	Text	-	50
Ime	Text	-	50
Skola	Text	-	50
IDSmer	Number	Long Integer	-
EnglCita	Yes/No	Yes/No	-
EnglPise	Yes/No	Yes/No	-
EnglGovori	Yes/No	Yes/No	-
NemCita	Yes/No	Yes/No	-
NemPise	Yes/No	Yes/No	-
NemGovori	Yes/No	Yes/No	-
RusCita	Yes/No	Yes/No	-
RusPise	Yes/No	Yes/No	-
RusGovori	Yes/No	Yes/No	-
OstJezici	Text	-	50
OstCita	Yes/No	Yes/No	-
OstPise	Yes/No	Yes/No	-
OstGovori	Yes/No	Yes/No	-
Interes	Text	-	100
Adresa	Text	-	50
Email	Text	-	50
Telefon	Text	-	50

Za svakog studenta je potrebno da postoji polje Index koje se odnosi na njegov broj indeksa. To polje je ovde sa namerom podebljano i naznačeno, jer je ono ustvari primarni ključ, odnosno polje po kojem će se svaki student moći jedinstveno identifikovati u tabeli. Ovaj podatak je određeno kao obavezno za unos pri popunjavanju tabele. Za Index je tip podatka stavljen Text jer broj

indeksa studenta industrijskog inženjerstva i menadžmenta sadrži i slova i brojeve (npr. i5000). Pored ovog polja postoji za svakog studenta mogućnost unosa njegovog prezimena i imena, telefona, adrese, email adrese itd. Svaki student može govoriti neki jezik i zato je u tim poljima stavljen tip podatka Yes/No.

Tabela Smer



Naziv polja	Tip podatka	Podtip	Veličina
ID_Smer	Number	Long Integer	-
NazSmer	Text	-	50

Svaki smer se jednoznačno identificuje u tabeli pomoću svog identifikatora.

Tabela Semestar



Naziv polja	Tip podatka	Podtip	Veličina
Sortiranje	Number	Integer	-
Semestar	Text	-	50

Sortiranje je primarni ključ u tabeli. Predstavlja redni broj semestra (1,2,3....).

Tabela Predmet



Naziv polja	Tip podatka	Podtip	Veličina
ID_Predmet	Text	-	5
Naz_Predmet	Text	-	70
Fond	Text	-	50
Sem_Pol	Text	-	50
Objasnenje	Memo	-	-
Sortiranje	Number	Integer	

Tabela Ocena



Naziv polja	Tip podatka	Podtip	Veličina
Ocena	Number	Integer	-
Sortiranje	Number	Integer	-

Primarni ključ je ocena od 6 do 10.

Tabela Student_Predmet



Naziv polja	Tip podatka	Podtip	Veličina
Index	Text	-	5
ID_Predmet	Text	-	5
Ocena	Number	Integer	-
Datum	Date	-	-

Ovde će primarni ključ predstavljati i **Index** i **ID_Predmet**. Pošto se vrednosti za Index i ID_Predmet moraju preuzeti iz tabela Student i Predmet, tada moramo u tabeli Student_Predmet staviti da su polja Index i ID_Predmet istog tipa i podtipa podatka kao i u njihovim baznim tabelama.

Veze među tabelama su određene upravo u prethodnoj tabeli Student_Predmet. Takođe bi ovde pored veza sa tabelama Student i Predmet sa tabelom Student_Predmet, mogli dodati i vezu između tabele Ocena i tabele Student_Predmet.

Veze se uvek grade preko primarnih ključeva u tabelama.

Ukoliko se pogleda šta je do sada urađeno vidi se da je definisana svrha baze podataka, tabele koje je čine, polja koja čine svaku tabelu i veze između samih tabela. Nije potrebno redefinisati ovako koncipiranu bazu podataka, jer je obuhvaćeno sve što je potrebno za izgradnju baze podataka.

Upiti, forme i izveštaji bi trebali da olakšaju rad sa podacima koji se nalaze u bazi podataka. Oni će biti objašnjeni u nastavku, a prethodno je potrebno kreirati samu bazu podataka(napraviti novu bazu, kreirati tabele, polja, primarne ključeve i veze među tabelama).

3. IZGRADNJA BAZE PODATAKA U MICROSOFT ACCESS-u

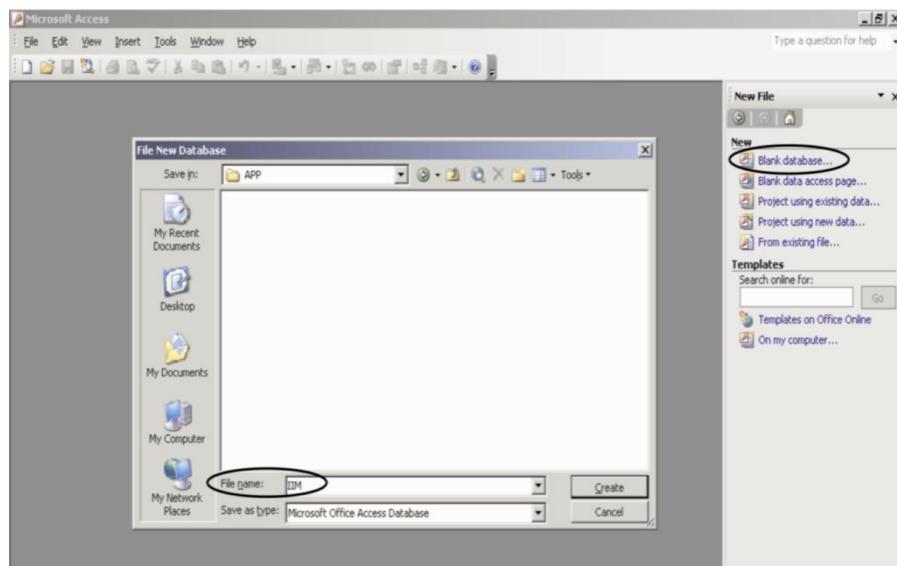
3.1. TABELE

Pokretanje Access-a se vrši iz menija Start na sledeći način:

Start > All Programs > Microsoft Office > Microsoft Office Access 2003

Kada se Access pokrene potrebno je kreirati (imenovati) novu bazu podataka:

- u meniju **File** opcija **New..**
- nakon toga sa desne strane prozora potrebno je odabrat **Blank database..**
- potrebno je dati ime fajla u kojem se čuva baza podataka. U ovom slučaju zvaće se **IHM**. (slika 2)

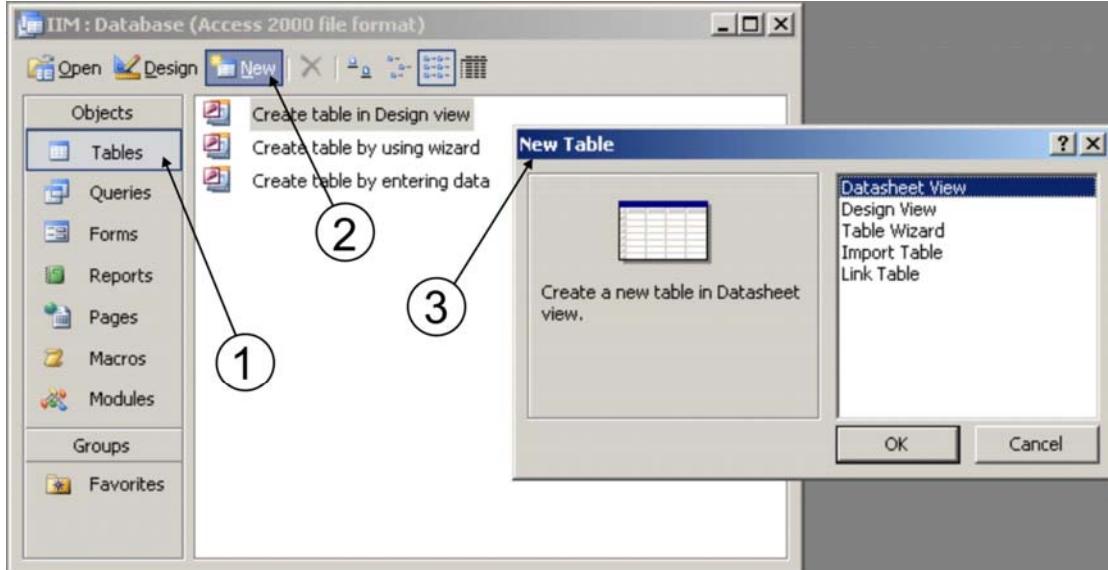


Slika 2. Kreiranje nove baze podataka

Baza podataka u Ms Access-u se čuva u fajlu sa ekstenzijom **.mdb**. Nakon kreiranja baza podataka potrebno je napraviti tabele. U prozoru koji je prikazan na **slici 3** potrebno je kliknuti na karticu **Tables** (označeno brojem 1) a zatim na karticu **New** (označeno brojem 2), jer će se tako kreirati nova tabela. Nakon toga otvara se novi prozor u kojem je potrebno odabrati kako će se kreirati nova tabela (označeno brojem 3). Postoji nekoliko različitih načina za kreiranje tabele :

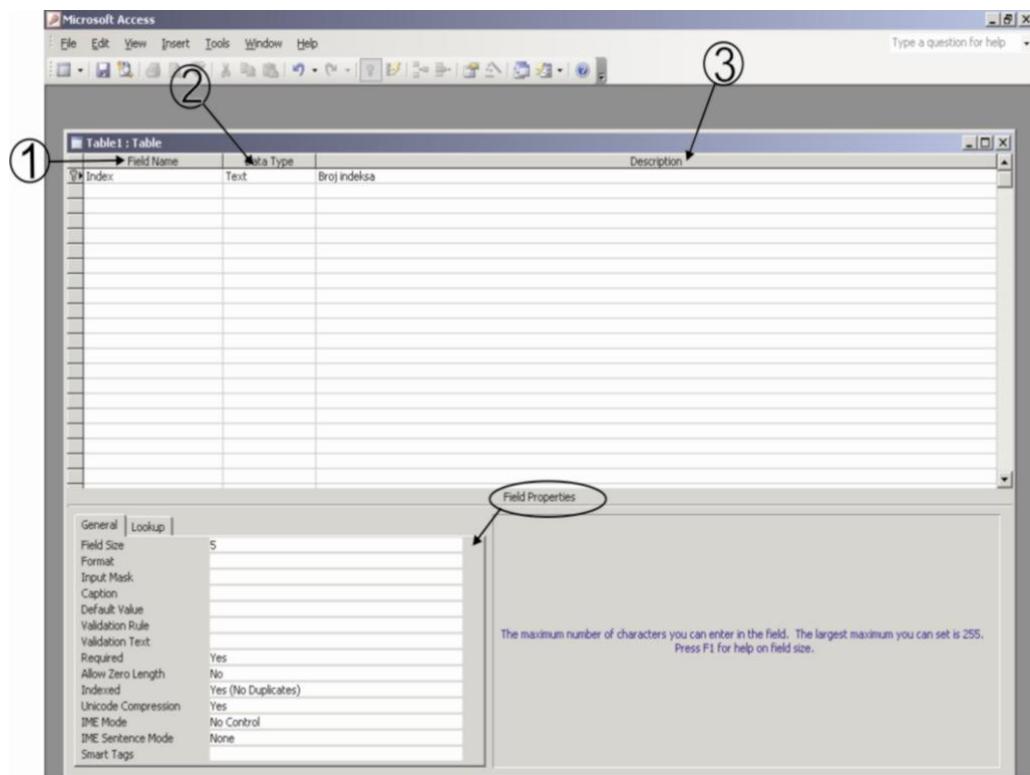
- Datasheet View – pogled na tabelu u kojoj ne postoje definisana polja, tipovi i podtipovi podataka. Kada se unesu podaci u tabelu, tada Access određuje kojem tipu podatak pripada.
- Design View – dizajn pogled na tabelu. Moguće je definisati nazine polja, tipove i podtipove podataka.
- Table Wizard – služi za pokretanje čarobnjaka (Wizard) koji omogućuje kreiranje tabele kroz sistem odgovora koje dobija od onoga ko je kreira.

- Import table – ukoliko već postoji negde ranije definisana tabela neke baze podataka moguće je ponovo iskoristiti tu tabelu tako što ćemo je importovati (vesti) iz te baze podataka
- Link Table – ovime se omogućuje da se postavi link ka tabeli(veza ka nekoj tabeli koja postoji u nekoj drugoj bazi podataka i da se podaci iz te tabele upotrebljavaju).



Slika 3. Početak kreiranja nove tabele

Potrebno je izabrati **Design View** i nakon što kliknemo na dugme **OK** pojavljuje se prozor prikazan na **slici 4**. Polja za tebele koja su ranije osmišljene sada mogu biti kreirana . Počnimo sa tabelom **Student** i poljem **Index** koje je primarni ključ tabele Student. Podatke o tipu, podtipu i veličini polja se uzimaju iz ranije isplanirane tabele Student.



Slika 4. Prozor za kreiranje polja u tabeli

Na **slici 4.** je sa brojem 1 označen deo u kojem se definiše naziv polja i u ovom slučaju polju je dano ime Index. U delu označenim sa brojem 2 potrebno je da se definiše tip podatka i ovde je tip podatka Text. Izbor tipa podatka se vrši jednostavnim izborom sa liste (**slika 5**).

	Field Name	Data Type	
1	Index	Text	Broj indeksa
2		Text	
		Memo	
		Number	
		Date/Time	
		Currency	
		AutoNumber	
		Yes/No	
		OLE Object	
		Hyperlink	
		Lookup Wizard...	

Kao što se sa slike 5 vidi, za svako polje je moguće odabrati neki od sledećih tipova podataka :

- Text
- Memo
- Number
- Date/Time
- Yes/No

itd.

Slika 5. Izbor tipa podatka

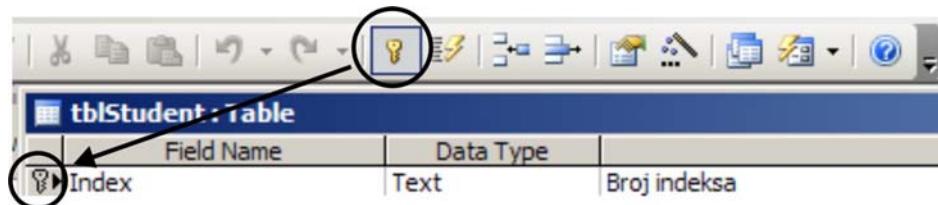
U delu označenim sa brojem 3 moguće je upisati opis (Description) za neko polje. U delu za opis je stavljeno da polje Index predstavlja Broj indeksa.

Za polje **Index** koje je **tipa Text** treba definisati i dodatne osobine koje se nalaze u delu **Field Properties** :

- **Field Size** – osobina u kojoj se definiše broj mesta za unos podatka u polje (Field Size = 5)
- **Required** – osobina u kojoj se definiše da li je obavezan unos podatka u polje (postavi se na Yes, jer se broj indeksa mora uneti za svakog studenta)
- **Indexed** – osobina koja definiše da li je polje indeksirano ili ne. Ako jeste, tada se ubrzava pretraživanje podataka.

Na ovaj način se podešavaju i sva ostala polja koja su tipa **Text** u tabeli **Student** (Index, Prezime, Ime, Skola, OstaliJez, Interes, Telefon, Adresa, Email).

Za polje **Index** potrebno je nekako **naznačiti da je ono ključno polje u tabeli** odnosno da je ono primarni ključ (**slika 6**).



Slika 6. Podešavanje primarnog ključa tabele

Prvo je potrebno označiti polje naziva **Index**, a zatim se klikne na ikonicu na kojoj se nalazi ključ (Slika 6.). Nakon toga će se kod polja **Index** pojaviti ključ koji će označiti to polje za **primarni ključ tabele**.

U tabeli Student pored polja koja su tipa Text postoje polja koja su **tipa Number (IDSmer)** i **tipa Yes/No (EnglCita, EnglPise, EnglGovori, NemCita, NemPise, NemGovori, RusCita, RusPise, RusGovori, OstCita, OstPise, OstGovori)**. Za svaki od ovih polja takođe se moraju definisati **Field properties**. Podatke o tipu, podtipu i veličini polja se uzimaju iz ranije isplanirane tabele Student.

Za polje **IDSmer** :

Field size – ovde se definiše podtip podatka za polje koje je tipa Number. Kao podtip je stavljenog **Long Integer**.

Decimal Places – Osobina koja definiše koliko je potrebno ostaviti decimalnih mesta za broj koji se upisatuju u polja (postavljeno Decimal Places na 0).

Caption – Osobina koja definiše naziv koji će stajati pored polja kada se ono nađe na formi. Ukoliko se ova osobina ne podesi tada se na formi prikazuje naziv polja koji je definisan u delu Field Name. Upisano je naziv polja **Smer** u osobinu Caption.

Default Value – podrazumevana vrednost koja стоји у полju. Postavljeno je **0**.

Indexed – Yes

Za polje **EnglCita** :

Format - osobina koja omogućuje izbor načina na koji će polje izgledati. Odabранo je **Yes/No**.

Required – No

Indexed – No

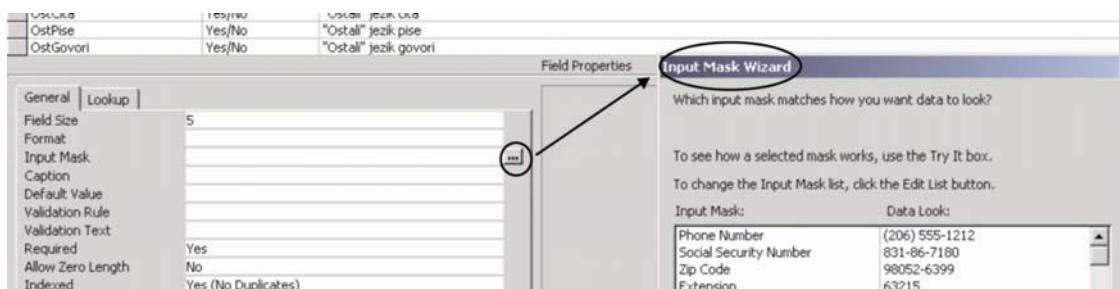
Na ovaj način potrebno je podesiti i sva ostala polja tipa **Yes/No**.

Dodatne osobine koje se mogu podešavati za polja neke tabele u **Field Properties** su :

Validation Rule – definiše pravilo koje se mora poštovati pri upisu vrednosti u polje (npr.ako je definisano za neko polje tipa Integer ili tipa Long pravilo **>0** to znači da u polje ne mogu biti upisane vrednosti manje ili jednake nuli.)

Validation Text – tekst koji se ispisuje u vidu poruke ukoliko se Validation Rule ne poštuje.

Input Mask – predstavlja masku za unos podataka, odnosno koristi se da bi omogućila unos podataka u nekom određenom formatu. Primarno se upotrebljava za polja tipa Text i za polja tipa Date/Time. Ukoliko je zamisao da se datum u polju prikazuje u određenom formatu to ovde možemo da se podesi. Takođe se može koristiti i za polja tipa Number i Currency. Može se uneti ručno ili putem Wizard-a (čarobnjaka) koji se poziva na način prikazan na **slici 7**. U Wizardu jednostavno izabere kako je zamišljeno da naši podaci izgledaju, odnosno bira se njihov format.



Slika 7. Pozivanje Wizard-a za podešavanje maske za unos podataka

Prikaz tabele **tblStudent** u Design View je dat na **slici 8**.

tblStudent : Table

	Field Name	Data Type	Description
Index	Text	Broj indeksa	
Prezime	Text	Prezime	
Ime	Text	Ime	
Skola	Text	Završena skola	
IDSmer	Number	Upisan na smer	
EnglCita	Yes/No	Engleski jezik cita	
EnglPise	Yes/No	Engleski jezik pise	
EnglGovori	Yes/No	Engleski jezik govori	
NemCita	Yes/No	Nemacki jezik cita	
NemPise	Yes/No	Nemacki jezik pise	
NemGovori	Yes/No	Nemacki jezik govori	
RusCita	Yes/No	Ruski jezik cita	
RusPise	Yes/No	Ruski jezik pise	
RusGovori	Yes/No	Ruski jezik govori	
OstaliJez	Text	Ostali jezici	
OstCita	Yes/No	"Ostali" jezik cita	
OstPise	Yes/No	"Ostali" jezik pise	
OstGovori	Yes/No	"Ostali" jezik govori	
Interes	Text	Područje interesovanja	
Adresa	Text	Adresa	
Email	Text	E-mail adresa	
Telefon	Text	Telefonski broj(evi)	

Field Properties

General | Lookup |

Field Size: 5

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Format
Input Mask
Caption
Default Value
Validation Rule
Validation Text
Required
Allow Zero Length
Indexed
Unicode Compression
IME Mode
IME Sentence Mode
Smart Tags

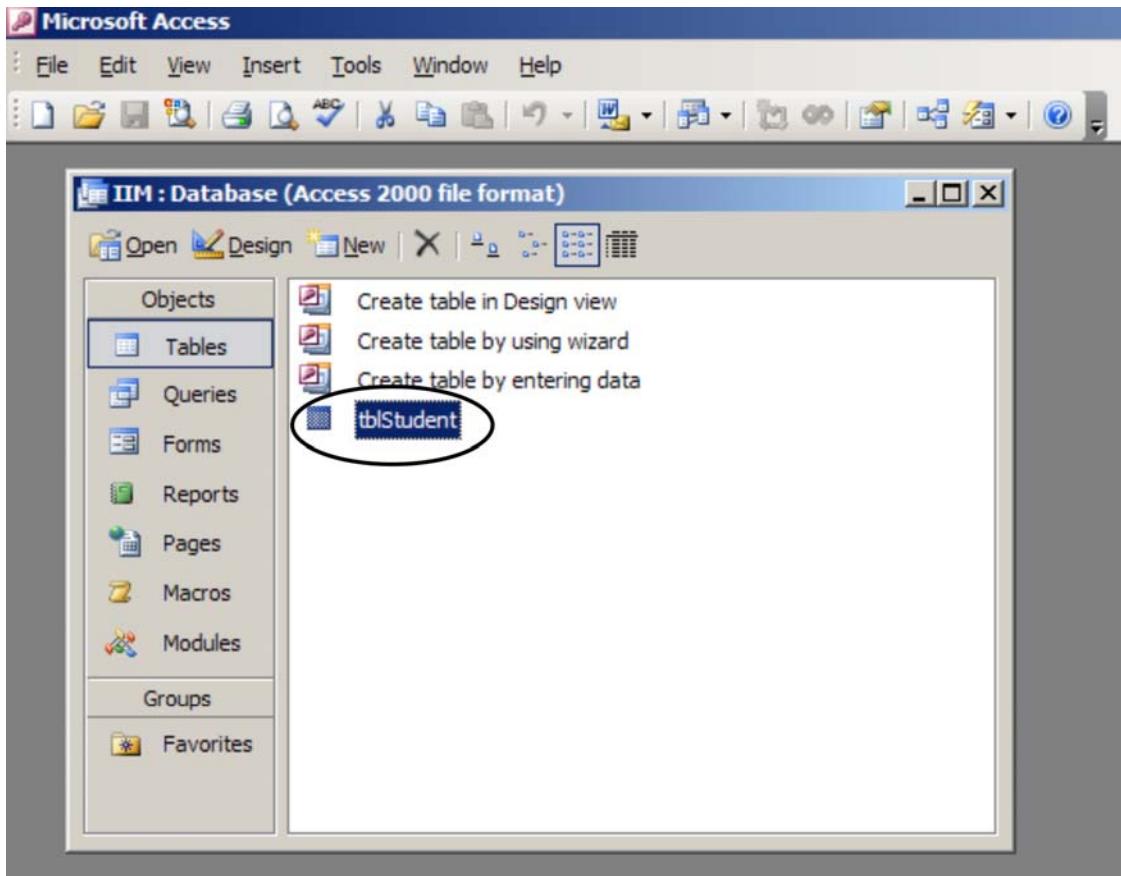
Slika 8. Tabela Student

Potrebno je snimiti tabelu. Pošto se ovde radi o tabeli ona će imati ime **tblStudent** (prefiks **tbl** je od **Table**). Snimanje tabele je prikazano na slici 9.



Slika 9. Snimanje tabele

Nakon što je tabela snimljena, zatvara se prozor sa tabelom i vraća se prikaz na osnovni prozor gde može povideti da je kreirana tabela **tblStudent** (slika 10.).



Slika 10. Tabela tblStudent u osnovnom prozoru

Dvostruki klik na tabelu **tblStudent** omogućuje pokretanje tabele u **Datasheet View**. Sada se javlja mogućnost za unos podataka o studentima, pri čemu je potrebno voditi računa da se primarni ključ tabele uvek mora biti unešen.

Izgled tabele u **Datasheet View** je dat na slici 11.

Microsoft Access

File Edit View Insert Format Records Tools Window Help

Mentorsko pranje studenata

Type a question for help

tblStudent : Table

Index	Prezime	Ime	Skola	Smer	EnglCita	EnglPise	EnglGovori	NemCita
+ I046	Jokovic	Dragana		Smer za inzenjerski manadzment	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
+ I047	Krstanovic	Sasa	Elektrotehnicka	Smer za industrijske sisteme	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
+ I048	Sijacic	Ljubomir	Gimnazija "Sve"	Smer za inzenjerski manadzment	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
+ I049	Klinac	Anita	Ekonomска sk.	Smer za industrijske sisteme	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
+ I050	Dakic	Dragana	Gimnazija "Sve"	Smer za inzenjerski manadzment	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
+ I051	Kesic	Igor	Elektrotehnicka	Smer za industrijske sisteme	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
+ I052	Vindzanovic	Dejan	Gimnazija "Sve"	Smer za inzenjerski manadzment	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
+ I053	Grahovac	Branka	Gimnazija "Sve"	Smer za inzenjerski manadzment	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
+ I054	Sogorovic	Djordje	Masinsko-tehni	Smer za inzenjerski manadzment	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
+ I055	Aleksic	Olivera		Smer za inzenjerski manadzment	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
+ I056	Panic	Mirko	Elektrotehnicka	Smer za inzenjerski manadzment	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
+ I057	Manojlovic	Marko	Ekonomска sk.	Smer za inzenjerski manadzment	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
+ I058	Kresovic	Natasa	Gimnazija "Sve"	Smer za inzenjerski manadzment	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
+ I059	Cvetkovic	Jovica		Smer za inzenjerski manadzment	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Slika 11. Izgled tabele u **Datasheet View**

Kada se zatvori prozor tabele **tblStudent** prikaz se vraća na osnovni prozor.

Na isti način je neophodno da budu kreirane i ostale tabele koje čine bazu podataka. U nastavku će biti prikazane tabele samo u Design View (na slikama tabela u Design View biće prikazani Field Properties za primarne ključeve u tabelama).

Tabela tblSmer (Design View)

Field Name	Data Type	Description
IDsmer	Number	ID broj smera
NazSmer	Text	Naziv smera

Field Properties

General | Lookup |

Field Size: Long Integer
Format: 0
Decimal Places: 0
Input Mask:
Caption:
Default Value:
Validation Rule:
Validation Text:
Required: Yes
Indexed: Yes (No Duplicates)
Smart Tags:

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Tabela tblSemestar (Design View)

Field Name	Data Type	Description
Semestar	Text	Semestar
Sortiranje	Number	Sortiranje

Field Properties

General | Lookup |

Field Size: Integer
Format: 0
Decimal Places: 0
Input Mask:
Caption:
Default Value:
Validation Rule:
Validation Text:
Required: No
Indexed: Yes (No Duplicates)
Smart Tags:

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Tabela tblPredmet (Design View)

Tabela tblOcena (Design View)

Tabela tblStud_Pred (Design View)

Primarni ključ u tabeli **tblStud_Pred** se kreira tako što se označi **Index** i **IDPredmet** i klikne se na ikonicu sa ključem.

Sada kada postoje sve tabele potrebno je da budu povezane u jednu celinu, što se radi definisanjem relacija između tabela.

3.2. VEZE IZMEĐU TABELA – RELACIJE

Uspešnim povezivanjem tabela u bazi podataka stvara se osnova za kreiranje upita, izveštaja i programskih formi. Za povezivanje tabela potrebno je koristiti primarne ključeve iz tabela. Za polja preko kojih se vrši povezivanje tabele poželjno je da budu istog tipa i podtipa podatka.

Primer 4 : Povezivanje tabele `tblStudent` sa tabelom `tblStud_Pred` se može izvršiti preko polja `Index` koje je primarni ključ u tabeli `tbl Student` i polja `Index` koje se nalazi u tabeli `tblStu_Pred`. Ovo i jeste logičan način za povezivanje tabela jer će brojevi indeksa u tabeli `tblStud_Pred` moći da se preuzimaju iz tabele `tblStudent`. Preko broja indeksa se mogu veoma lako saznati i svi ostali podaci o nekom studentu.

Postoje nekoliko tipova veza koje mogu da postoje između tabela:

Veza jedan prema više – kreira se ukoliko je samo jedan od povezanih polja primarni ključ u jednoj od tabela (veoma čest tip veze među tabelama),

Veza jedan prema jedan - kreira se ukoliko su oba od povezanih polja primarni ključevi u svojim tabelama.

Veza više prema više – predstavlja ustvari dve veze **jedan prema više**. Javlja se kada se povezuju dve tabele sa nekom trećom tabelom. U toj trećoj tabeli primarni ključ čine oba primarna ključa iz prve dve tabele (ovo se može videti na tabeli `tblStud_Pred`, gde postoji prelaz ključeva `Index` i `ID_Predmet` iz tabela `tblStudent` i `tblPredmet`).

Kreiranje veza između tabela u Access-u će biti prikazano na primeru povezivanja tabela **tblStudent** i **tblPredmet** sa tabelom **tblStud_Pred**. Povezuju se polja **Index** i **IDPredmet** (**slika 12**).

tblStudent : Table		
Field Name	Data Type	Description
Index	Text	Broj indeksa
Prezime	Text	Prezime
Ime	Text	Ime
Skola	Text	Završena skola
IDSmer	Number	Ilinjan na smer

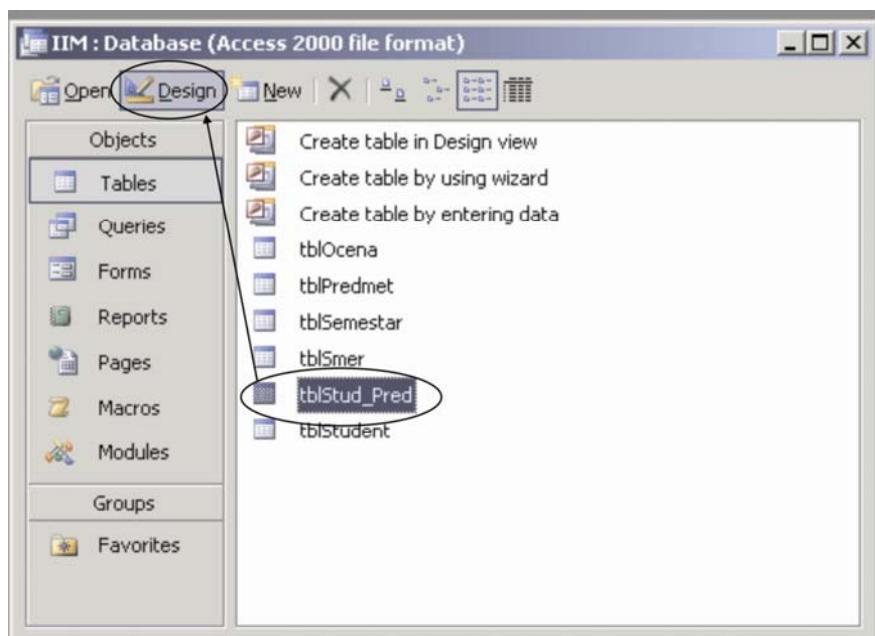
tblStud_Pred : Table		
Field Name	Data Type	Description
Index	Text	Broj indexa
ID_Predmet	Text	Identifikacioni broj predmeta
Ocena	Number	Ocena
Datum	Date/Time	Datum

tblPredmet : Table		
Field Name	Data Type	Description
ID_Predmet	Text	Identifikacioni broj predmeta
Naz_Pred	Text	Naziv predmeta
Fond	Text	Fond casova, predavanja i ve
Sem_Pol	Text	Semestar u kome se polaze pr

Slika 12. Polja koja je potrebno povezati

Kreiranje veze između tabela **tblStudent** i **tblStud_Pred** preko polja **Index**:

- o otvoriti tabelu **tblStud_Pred** u **Design View** (označi se tabela **tblStud_Pred** u početnom prozoru i klikne na dugme **Design** - **slika 13**)



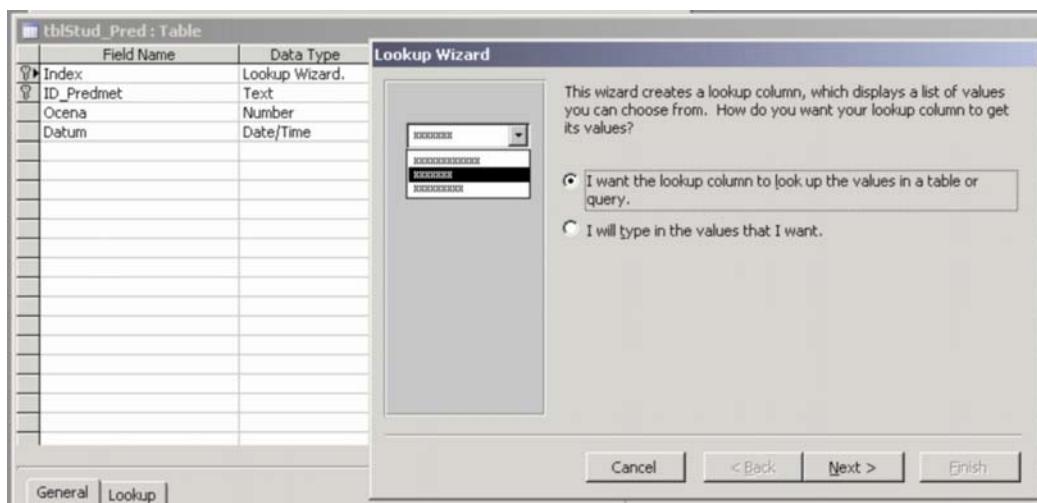
Slika 13. Otvaranje postojeće tabele u **Design View**

- u tabeli **tblStud_Pred** kod polja **Index** otvorimo se padajuća lista u delu **Data Type** (slika 14) i odabereme se opcija **Lookup Wizard** (čarobnjak za kreiranje veza među tabelama).

	Field Name	Data Type	
Index	Text	Broj indexa	
ID_Predmet	Text	Identifikacioni broj predmeta	
Ocena	Memo	Ocena	
Datum	Number	Datum	
	Date/Time		
	Currency		
	AutoNumber		
	Yes/No		
	OLE Object		
	Hyperlink		
	Lookup Wizard...		

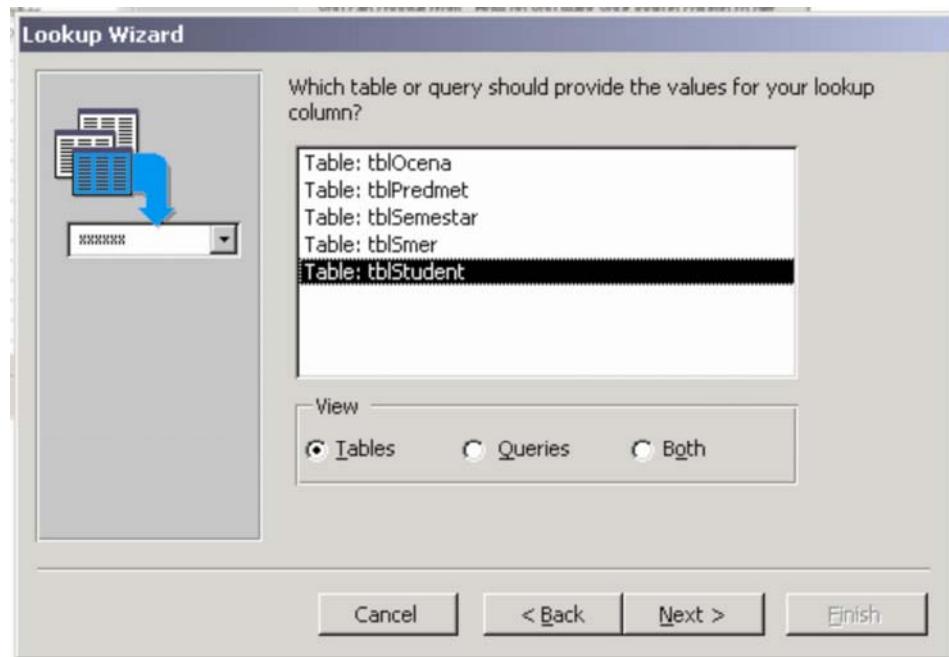
Slika 14. Odabir Lookup Wizard-a u delu Data Type

- u prvom koraku u Lookup Wizard-u odabere se prva opcija kojom se kaže da se želi da polje Index iz tabele **tblStud_Pred** preuzima postojeće vrednosti iz neke tabele (u ovom slučaju to će biti tabela **tblStudent**), što je prikazano na slići 15. Odabir druge opcije u ovom prvom koraku omogućuje unos bilo kojih brojeva indeksa i stoga je ipak potrebno se odlučiti za prvu opciju. Klik na dugme Next za prebacivanje na sledeći korak.



Slika 15. Prvi korak

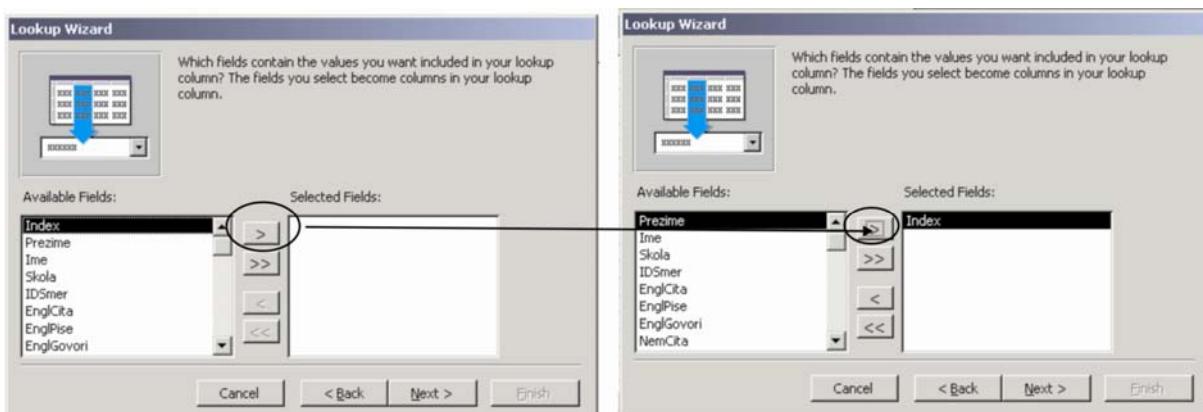
- U drugom koraku odabira se tabela iz koje se želi preuzeti vrednosti. Odabrana je tabelu **tblStudent** – slika 16. Klik na Next za sledeći korak.



Slika 16. Drugi korak – odabir tabele sa kojom se želi povezati

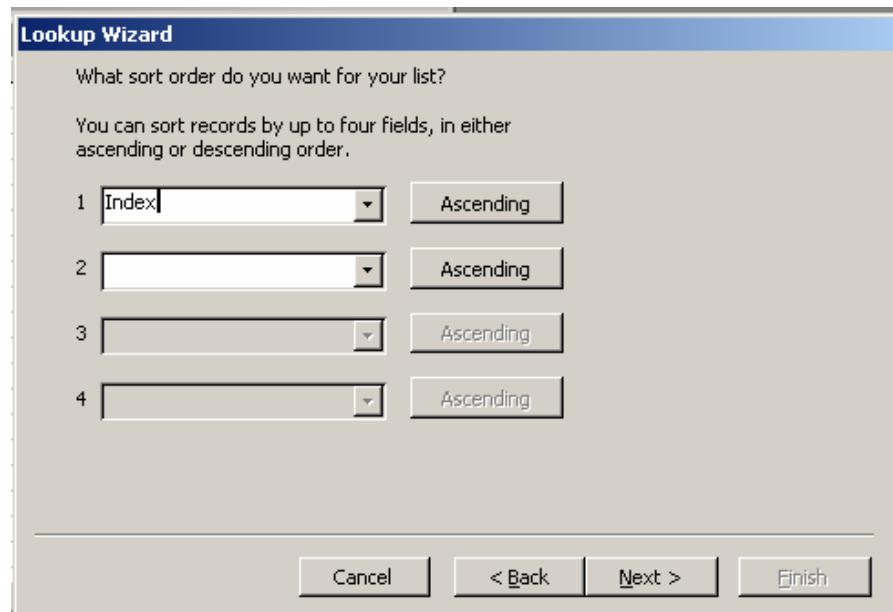
U trećem koraku odabira se Index iz tabele tblStudent, jer se žele preuzeti brojevi indeksa iz tabele tblStudent – **slika 17**.

Da su kojim slučajem prebačena i još recimo polja Prezime i Ime, mogla biti i ona viđena u tabeli tblStud_Pred.



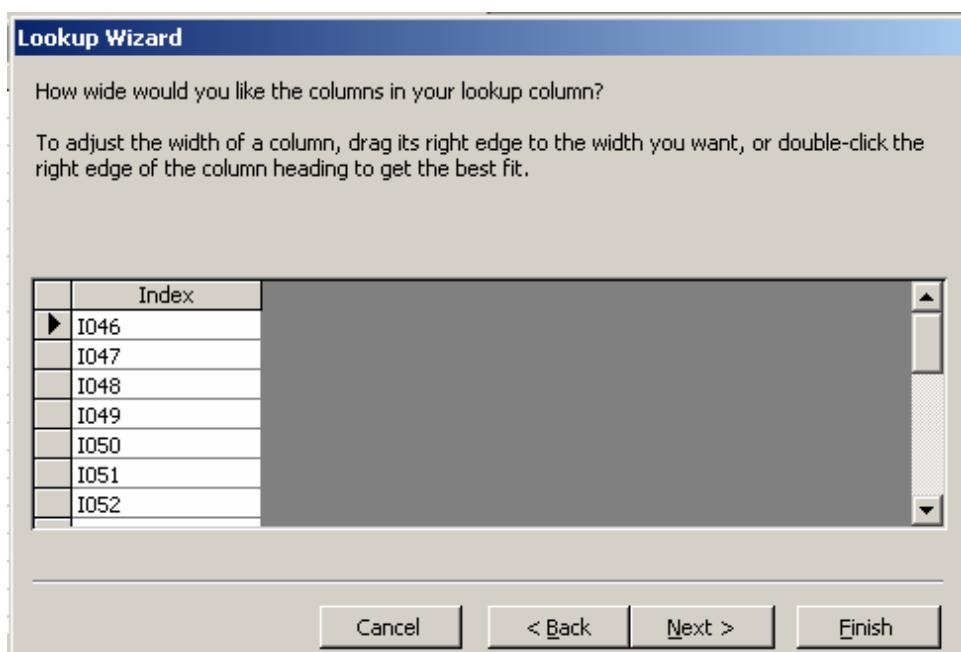
Slika 17. Treći korak - Odabir polja sa kojim se želi povezati

U četvrtom koraku se bira redosled sortiranja zapisa po opadajućem ili rastućem redosledu – **Slika 18.**



Slika 18. Četvrti korak – biranje redosleda sortiranja

Peti korak služi za podešavanje izgleda polja i služi da omogući promenu širine – Slika 19.



Slika 19. Peti korak – podešavanje izgleda polja

U poslednjem koraku odabira se ime kolone u kojoj će se nalaziti polja – Slika 20.



Slika 20. Poslednji korak – Biranje naziva za kolonu

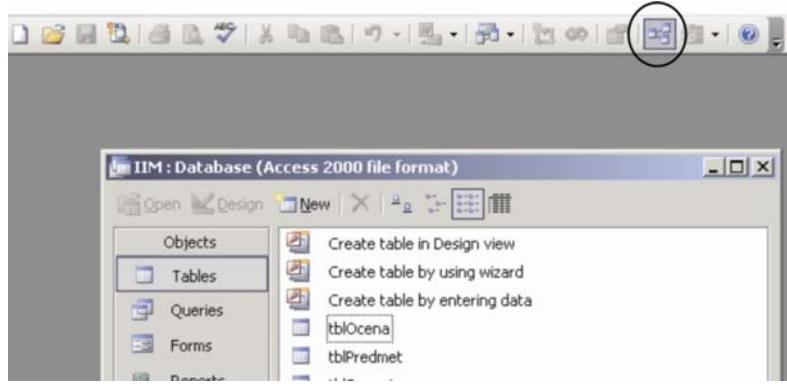
Klikom na dugme Finish, kreirana je veza odnosno relacija između tabele tblStud_Pred i tabele tblStudent putem indeksa studenta. Na isti način se može kreirati i veza između tabela tblStud_Pred i tblPredmet (preko IDPredmet). Postupak je isti i neće se ovde ponavljati.

Potrebno je još ograničiti unos podataka za broj indeksa i oznaku predmeta u tabeli tblStud_Pred na one koji postoje u tabelama tblStudent i tblPredmet. Ako se tabela tblStud_Pred otvorи u Design View i u delu Field Properties klikom na karticu Lookup videće da se pri dnu nalazi osobina koja se zove **Limit to List**. Potrebno je da se podesi vrednost na Yes i za Index i za IDPredmet. Kada je ovo urađeno zatvoranjem prozora tabele se prikazuje osnovni prozor.

Za pravilno povezivanje tabela neophodno je još da bude definisan i tip veze između njih : **jedan prema više, jedan prema jedan ili više prema više**. Ukoliko se pogledaju ranije navedene napomene o tipovima veza možemo se zaključiti:

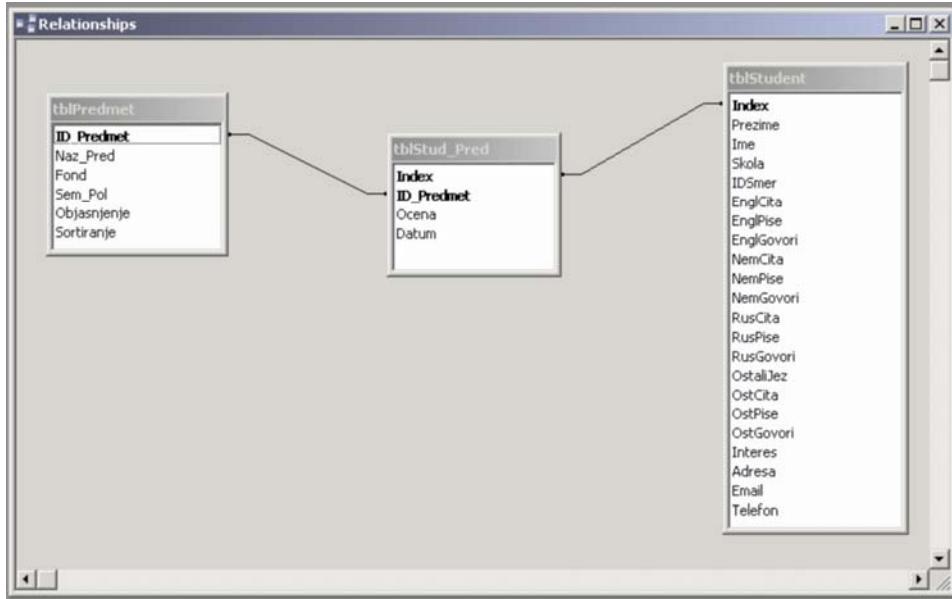
- pošto se ovde radi o povezivanju dve tabele sa trećom, odnosno o povezivanju tabele tblStudent i tblPredmet sa tabelom tblStud_Pred, a ključevi iz tabela tblStudent i tblPredmet čine zajedno ključ tabele tblStud_Pred tada se radi o tipu veze **više prema više**. Tip veze **više prema više** činiće **dve veze tipa jedan prema više** i to:
 - **jedan prema više** između tabela **tblStudent i tblStud_Pred** – ova veza kaže da jedan student može imati **jedan ili više** položenih predmeta,
 - **jedan prema više** između tabela **tblPredmet i tblStud_Pred** - ova veza kaže da jedan predmet može biti položen od strane **jednog ili više** studenata.

Da bi bio otvoren prozor za definisanje tipa veze potreban je klik na ikonicu Relationships kao na slici 21.

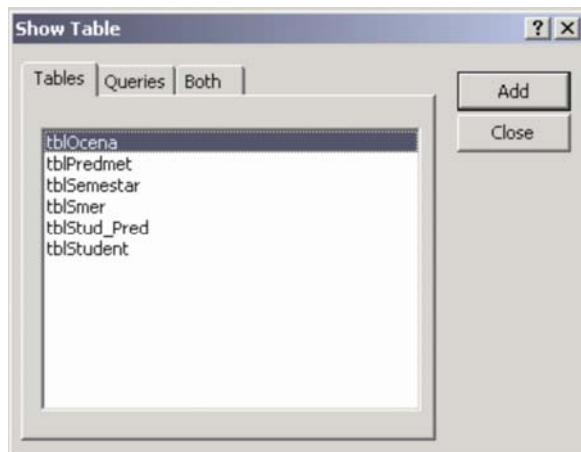


Slika 21. Prikaz komandnog dugmeta Relationships

Otvoren je prozor koji pokazuje **tabele tblPredmet, tblStud_Pred i tblStudent** i veze između njih (**slika 22**). Ako se ne vide table i veze tj. ako se prikaže samo prazan prozor potrebno je desnim dugmetom miša kliknuti unutar prozora i odabira se opciju **Show Table...** i tada se otvara prozor i mogućnost da izaberemo tabele za prikaz (**slika 23**) u prozoru **Relationships**.

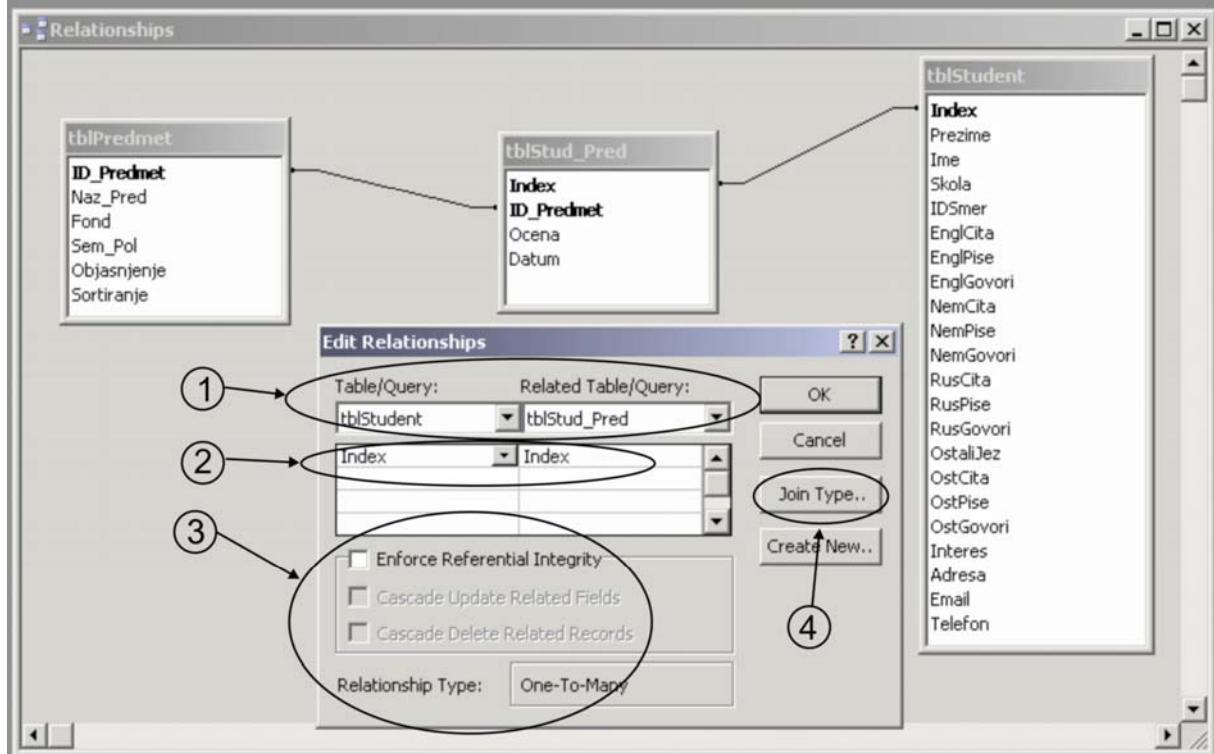


Slika 22. Prikaz tabela i veza



Slika 23. Prozor Show Table

Da bi bio definisan tip veze između tabela potrebno je da jednostavno kliknuti dva puta na vezu između tabela i otvara se prozor **Edit Relationships** - slika 24. Definisanje tipa veze će biti prikazano na primeru veze između tabela **tblStud_Pred** i **tblStudent**. **Definisanje tipa veze između tabela **tblStud_Pred** i **tblPredmet** se ostavlja studentima za vežbu.**

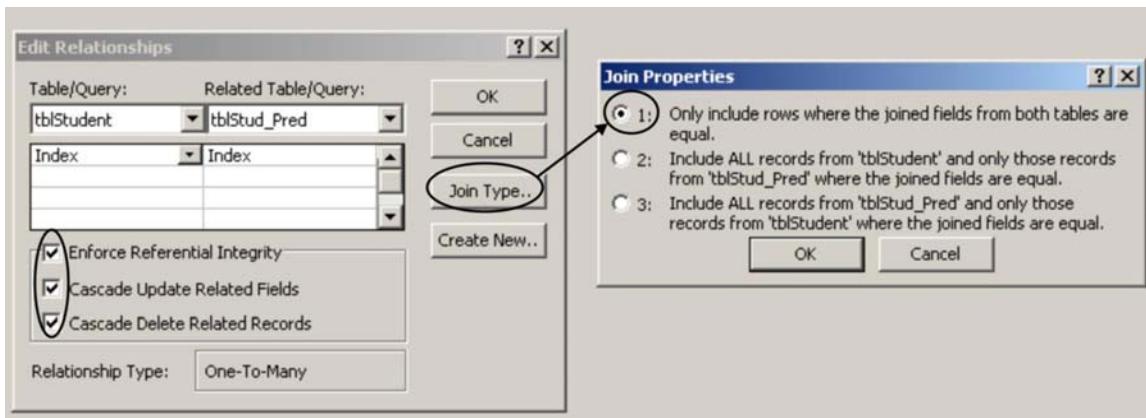


Slika 24. Prozor Edit Relationships

Na slici se vidi nekoliko celina u prozoru **Edit Relationships** :

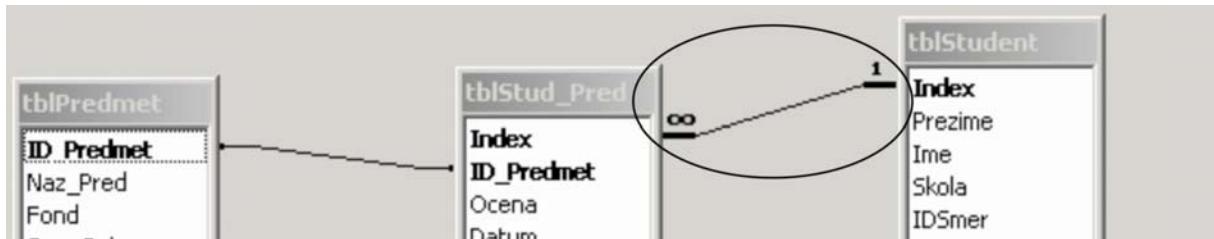
1. – prikazane su tabele koje se povezuju (**tblStudent** i **tblStud_Pred**)
2. – prikazana su polja koja su povezana (**Index** iz **tblStudent** i **Index** iz **tblStud_Pred**)
3. – prikazane su opcije za primenu **referencijalnog integriteta**. Referencijalni integritet predstavlja sistem pravila koje Access koristi da bi se omogućilo da su veze između tabela validne i koje onemogućuju slučajno brisanje povezanih podataka. Ispod opcija se vidi **Relationship Type : One-To-Many** što znači veza **jedan prema više**.
4. – komandno dugme **Join Type** - klik na njega omogućuje definisanje koje će se vrednosti prikazati u tabeli **tblStud_Pred** u zavisnosti od izabrane opcije 1, 2, ili 3. Ovo je bitno pri pravljenju upita nad tabelama i ne utiče direktno na vezu. Potrebno je da bude odabrana opciju 1, koja će dozvoliti da se u tabeli **tblStud_Pred** pojave podaci o studentima, ali samo oni čiji brojevi indeksa već postoje u tabeli **tblStud_Pred**.

Podešavanje opcija u prozoru **Edit Relationships** je prikazano na **slici 25**.



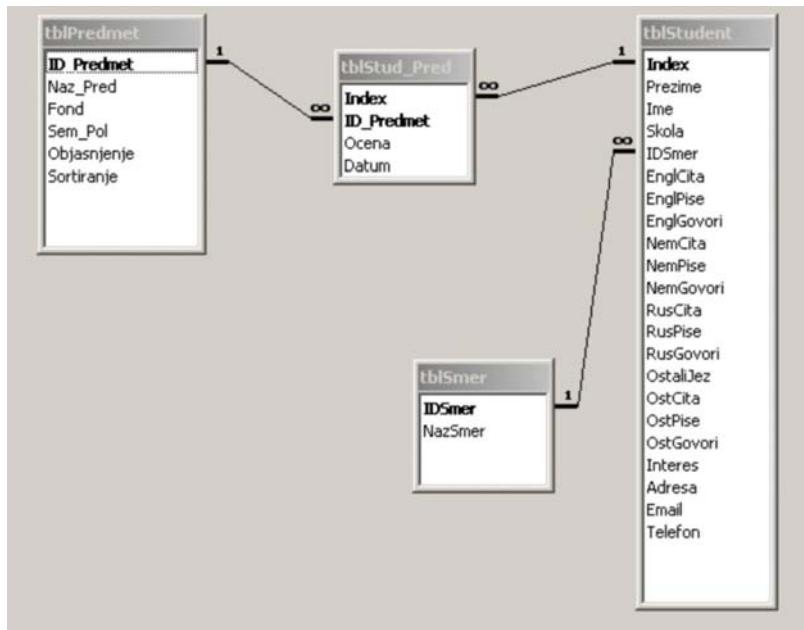
Slika 25. Uključivanje pravila **referencijalnog integriteta** i odabir odgovarajuće opcije u prozoru **Join Properties**

Kada su podešeni svi parametri potrebno je kliknuti na dugme **OK**. Veza između tabela **tblStudent** i **tblStud_Pred** se promenila (**slika 26**).



Slika 26. Prikaz redefinisane veze između tabela

Što se tiče ostalih tabela i veza između njih na **slici 27** je prikazan prozor Relationships sa svim vezama, te je stoga potrebno proći sve korake i definisati te veze.



Slika 27. Konačni izgled veza među tabelama

Kada su jednom definisane sve veze između tabela moguće je vršiti kreiranje upita nad tabelama.