

Napredne baze podataka – Microsoft Access 2010

Priručnik

Autor: Ivan Ljubičić

Autor:

Ivan Ljubičić

Glavni naslov:

ITdesk.info – projekt računalne e-edukacije sa slobodnim pristupom - Priručnik za digitalnu pismenost

Podnaslov:

Napredne baze podataka – Microsoft Access 2010, priručnik

Stručna recenzija:

Frane Šesnić, mag.ing.elektrotehnike, [EdukaCentar, ustanova za obrazovanje](#), R. Hrvatska

Naslovnica:

Silvija Bunić

Nakladnik:

Otvoreno društvo za razmjenu ideja (ODRAZI), Zagreb

Mjesto i godina izdavanja:

Zagreb, 2014.

Agencija za odgoj i obrazovanje Republike Hrvatske odobrila je uporabu ovog pomoćnog nastavnog sredstva u osnovnim školama rješenjem

klasa: 602-09/14-01/0419

urbroj: 561-03-03/10-15-4

Zagreb, travanj 2015.

Autorsko pravo:

Slobodno kopirajte, tiskajte i dalje distribuirajte cijelu ili dio ove publikacije, uključujući i u svrhu organiziranih školovanja, bilo u javnim ili u privatnim edukacijskim organizacijama, ali isključivo u nekomercijalne svrhe (dakle bez naplate krajnjim korisnicima/cama za korištenje publikacije) te uz navođenje izvora (izvor: www.ITdesk.info - projekt računalne e-edukacije sa slobodnim pristupom). Izvedeni (derivirani) radovi nisu dopušteni bez prethodnog odobrenja nositelja autorskih prava (udruga *Otvoreno društvo za razmjenu ideja* - ODRAZI). Kontakt za traženje dozvole: info@ITdesk.info

Današnje društvo obilježeno je naglim rastom i razvojem informacijske tehnologije (IT), što je rezultiralo velikom ovisnošću društva, u širem smislu, o znanju i kompetencijama osobe u IT području. I premda ta ovisnost raste iz dana u dan, **ljusko pravo na obrazovanje i informacije** nije prošireno i na IT područje. Pojavili su se problemi koji utječu na društvo u cjelini, koji stvaraju prepreke i udaljavaju ljude od glavnog razloga i motivacije za napredak, od **prilike**. Biti računalo nepismena osoba, danas znači biti osoba koja nije u mogućnosti sudjelovati u modernom društvu, biti osoba bez prilike. Unatoč priznanju Europske komisije, UNESCO-a, OECD-a i ostalih relevantnih institucija o neophodnosti i korisnosti informatičke pismenosti, još uvijek postoje grupe ljudi s otežanim pristupom osnovnoj računaloj naobrazbi (npr. osobe s invaliditetom, osobe s poteškoćama u učenju, radnice/i migranti, nezaposlene osobe, osobe koje žive na udaljenim mjestima gdje nemaju pristup računaloj edukaciji).

Ovaj priručnik, zajedno s ostalim materijalom objavljenim na stranici ITdesk.info, predstavlja naš doprinos realizaciji i promociji ljudskog prava na edukaciju i informacije u IT području. Nadamo se da će Vam ova edukacija pomoći u savladavanju osnovnih računalojnih vještina i s tom nadom želimo Vam da naučite što više i tako postanete aktivni/a član/ica modernog IT društva.

Iskreno Vaši,

ITdesk.info team



SADRŽAJ:

Predgovor	i
1. Osnovni pojmovi	1
1.1 Razvoj i korištenje baza podataka	1
Modeli baza podataka.....	1
Životni ciklus baze podataka	2
Poslovne aplikacije baza podataka	3
SQL – strukturirani jezik za upite	3
2. Tablice i relacije	5
2.1 Polja/stupci	5
Kreiranje, mijenjanje i brisanje pretraživanja u polju/stupcu	5
Kreiranje, mijenjanje i brisanje maske za unos podataka u polje/stupac	6
Postavljanje obveznih unosa vrijednosti podataka na polje/stupac: potrebno, nije potrebno	8
2.2 Relacije i spajanje	9
Kreiranje, mijenjanje i brisanje veza između tablica: jedan-na-jedan, jedan-na-više	9
2.3 Kreiranje i mijenjanje veza između tablica: više-na-više koristeći spojne tablice	11
Primjena referencijalnog integriteta između tablica	11
Primjena automatskog ažuriranja povezanih polja	13
Primjena automatskog brisanja povezanih zapisa	13
Primjena i promjena unutarnjeg i vanjskog spajanja	14
Kreiranje i promjena vrste spajanja	16
3. Upiti	17
3.1 Vrste upita.....	17
Kreiranje i pokretanje upita za ažuriranje podataka u tablici	17
Kreiranje i pokretanje upita za dodavanje zapisa u tablici	19
Kreiranje i pokretanje upita za brisanje u tablici	21
3.2 Kreiranje i pokretanje upita za spremanje odabranih podataka kao nove tablice	22
Kreiranje i pokretanje unakrsnog upita	22
Kreiranje i pokretanje upita za prikaz zapisa koji se ponavljaju u tablici	25
Kreiranje i pokretanje upita za prikaz neuparivih zapisa u povezanoj tablici	28
3.3 Dotjerivanje upita.....	32
Kreiranje, mijenjanje i pokretanje parametarskog upita s jednom, dvije varijable	32
Korištenje zamjenskih znakova u upitu: [], !, -, #.....	35
4. Opći uvjeti korištenja	38



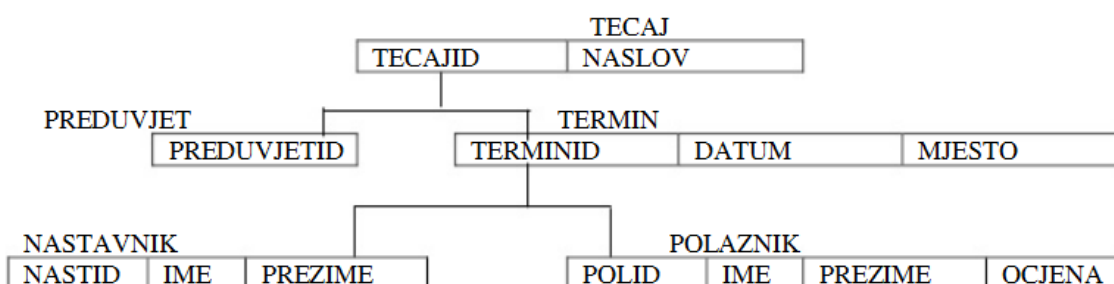
OSNOVNI POJMOVI

1.1 Razvoj i korištenje baza podataka

Modeli baza podataka

Postoje tri osnovna modela baza podataka:

- Hijerarhijski – prevladava od šezdesetih do osamdesetih godina, a zasniva se na hijerarhijskim strukturama podataka koji imaju oblik stabla. Svako stablo ima jedan osnovni čvor/slog (engl. *root*) i nula ili više čvorova/slogova podstabla. Slog se sastoji od polja koji odgovaraju atributima. Odnos između tih slogova se naziva odnosom roditelj-dijete (engl. *parent-child relationship*)



- Mrežni – skupa s hijerarhijskim prevladava od šezdesetih do osamdesetih godina, zasniva se na mreži podataka u kojoj ne postoje ni podređeni ni osnovni segmenti. Svako dijete slog može imati više roditelja slogova
- Relacijski model – od osamdesetih godina prevladava ovaj model kod kojeg se baza podataka sastoji od skupa pravokutnih tablica ili relacija. Svi podaci su spremljeni u tabličnom obliku. Svaka relacija ima ime po kojem se razlikuje od ostalih relacija u bazi podataka. Svaki stupac relacije sadrži vrijednost jednog atributa entiteta ili veze koji/koja se prikazuju relacijom (tablicom) (polja u hijerarhijskom modelu odgovaraju stupcima u relacijskom modelu). Atribut ima ime kojim se razlikuje od ostalih atributa. Vrijednost jednog atributa su podaci iste vrste ili tipa. Skup vrijednosti jednog atributa nazivamo domenom atributa. Broj atributa je stupanj relacije. Jedan redak relacije predstavlja primjerak entiteta ili bilježi vezu između dva ili više primjeraka entiteta. Primjerak entiteta zovemo n-torka. U jednoj relaciji ne mogu biti dvije jednake n-torke. Broj n-torki je kardinalnost relacije. Temelj relacijskih baza podataka je povezivanje entiteta različitog tipa.

Grad_ID	Ime	Koordinate
ZAG	Zagreb	(x1, y1)...
BER	Berlin	(x2, y2)...
GRZ	Graz	(x3, y3)...
...

Država_ID	Ime	Koordinate
GER	Njemačka
AUT	Austrija
CRO	Hrvatska	...

Grad_ID	Država_ID
ZAG	CRO
BER	GER
GRZ	AUT

- Objektno orijentirane baze podataka se pojavljuju sredinom 80-ih godina kao rješenje određenih problema relacijskih baza. Primjerice, objektne baze dozvoljavaju definiciju raznih tipova podataka varijabilne složenosti što relacijske baze u pravilu ne dozvoljavaju korisniku.

Životni ciklus baze podataka

Uvođenje baze podataka u neko poduzeće ili ustanovu je kompleksan posao koji se može podijeliti u pet faza:

1. Analiza potreba – proučavaju se tokovi informacija u poduzeću, uočavaju se podaci koje treba pohranjivati i veze među njima. Unutar poduzeća često postoje razne grupe korisnika i pojavljuju se različiti „pogledi” na podatke. Ova faza obuhvaća i analizu transakcija (operacija) koje će se obavljati nad bazom podataka. Rezultat analize je dokument koji se zove specifikacija potreba. Postoje formalne metode analize zahtjeva (analize potreba) koje se koriste uglavnom na velikim projektima na kojima radi više ljudi organiziranih u projektne timove.

2. Modeliranje podataka – različiti „pogledi” na podatke iz prve faze se sintetiziraju u jednu cjelinu – globalnu shemu, precizno se utvrđuju tipovi podataka. Shema se doraduje tako da zadovolji određene zahtjeve kvalitete. Shema se prilagođava ograničenjima koje postavlja zadani model podataka, te se dodatno modificira da bi mogla udovoljiti zahtjevima na performanse. Rezultat ove faze su pogledi (pod-scheme) koje se izvode za pojedine grupe korisnika.
3. Implementacija – u ovoj fazi se na osnovu rezultata prethodnih faza, fizički realizira baza podataka na računalu. U sklopu DBMS-a (engl. *Database Management System*), podešavaju se parametri da bi se osigurao efikasan rad najvažnijih transakcija. Razvijaju se skup programa koji realiziraju pojedine transakcije te pokrivaju potrebe raznih aplikacija. Baza se inicijalno puni podacima
4. Testiranje – korisnici testiraju bazu i provjeravaju da li ona zadovoljava svim zahtjevima. Nastoje se otkloniti moguće greške iz ranijih faza razvoja. Greške u ranijim fazama razvoja imaju teže posljedice, primjerice, ukoliko dođe do greške u analizi potreba u vidu da je korisnicima potrebno nešto drugo što nije implementirano. U posljednje vrijeme se prije prave implementacije razvijaju prototipovi baze podataka koji se demonstriraju korisnicima.
5. Održavanje – ova faza se odvija u vremenu kad je baza već ušla u redovnu upotrebu. Održavanje se sastoji od sljedećeg: popravak grešaka koje nisu otkrivene u fazi testiranja, uvođenje promjena zbog novih zahtjeva korisnika i podešavanje parametara u DBMS-u u svrhu poboljšavanja performansi. U fazi održavanja, stalno se prati rad s bazom i to na način da to praćenje ne ometa korisnike.

Poslovne aplikacije baza podataka

Aplikacije, odnosno programi i programski sustavi u kojem se koriste neki oblici baza podataka su široko rasprostranjeni. Može se reći da gotovo svaka aplikacija koristi manju ili veću bazu podataka različitih vrsta.

Kod dinamičkih web stranica, često se koristi MySQL, besplatni sustav za upravljanje bazama podataka.

Kod većih poslovnih informacijskih sustava, kao što su npr. Informacijski sustavi financijskih institucija, često se koriste Oracle i IBM rješenja. Uz Oracle-ovu bazu podataka, često se koristi i Oracle-ov programski jezik namijenjen usko za svrhu upravljanja bazama podataka – PL/SQL. PL/SQL u osnovi ima SQL jezik koji je pobliže objašnjen u narednom dijelu.

SQL – strukturirani jezik za upite

Povijest SQL-a (engl. *Structured Query Language*) kao strukturiranog jezika za upite počinje 1970. godine u IBM Research Laboratory-u u San Joseu. Komercijalna primjena počinje 80-ih godina. SQL omogućava stvaranje i izmjene strukture baze podataka, dodavanje prava

korisniku za pristup bazama podataka i tablicama, traženje informacija od baze podataka i mijenjanje sadržaja baze podataka.

Postoje dvije grupe funkcija, prva grupa su DDL (engl. *Data Definition Language*) funkcije za definiciju podataka. Primjer te grupe funkcija je naredba *CREATE TABLE imeTablice ()*.

Druga grupa su DML (engl. *Data Manipulation Language*) funkcije za upravljanje podacima. Primjer je osnovna SQL naredba:

```
SELECT * FROM imeTablice;
```

Korištenjem ovakvog upita i ključne riječi *SELECT* dohvaćaju se svi retci iz tablice *imeTablice*.

Ažuriranje tablice se vrši korištenjem ključne riječi *UPDATE*, primjerice, upit:

```
UPDATE imeTablice SET ime_atributa=vrijednost 1 WHERE ime_atributa2=vrijednost2;
```

U ovom primjeru u tablici *imeTablice* u retku/retcima koji ima/imaju vrijednost atributa *ime_atributa2* jednaku vrijednosti *vrijednost2* postavlja vrijednost atributa *ime_atributa* na *vrijednost1*.

Brisanje redaka iz tablice se vrši korištenjem ključne riječi *DELETE*. Primjerice:

```
DELETE FROM imeTablice WHERE ime_atributa = vrijednost
```

U prethodnom upitu u tablici *imeTablice* brišu se svi retci koji imaju vrijednost atributa *ime_atributa* jednaku vrijednosti *vrijednost*.

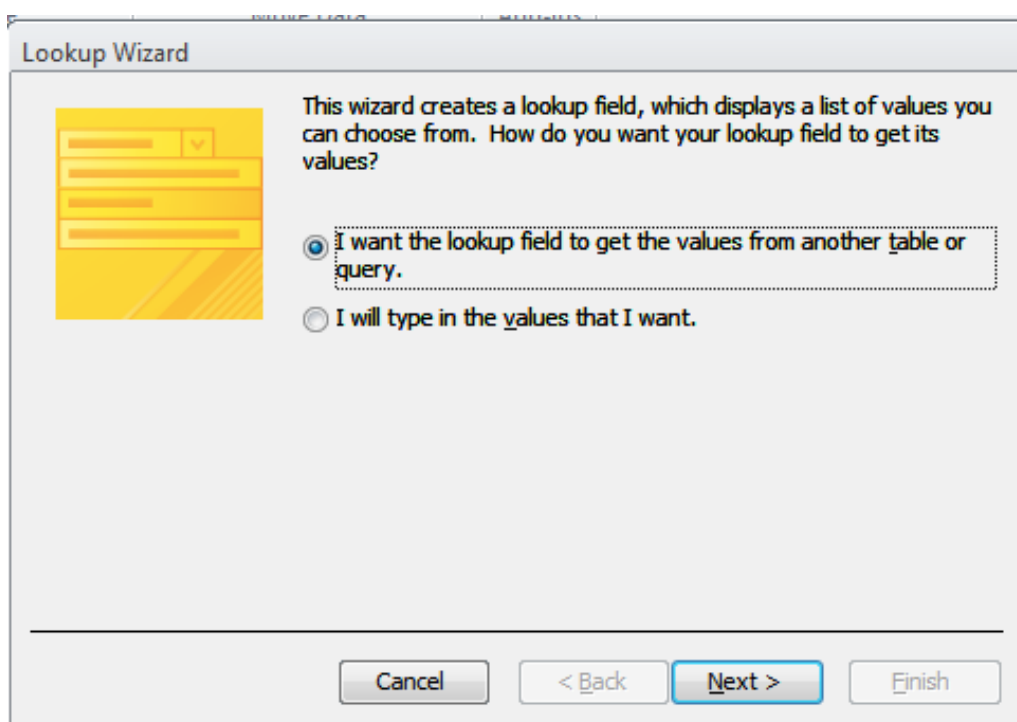
TABLICE I RELACIJE

1.2 Polja/stupci

Kreiranje, mijenjanje i brisanje pretraživanja u polju/stupcu

Stupac (polje) za pretraživanje je polje u tablici čija se vrijednost učitava iz druge tablice ili s popisa vrijednosti.

Otvorimo npr. tablicu Igrači. Desni klik na dodavanje novog polja („Click to add”) i odaberimo „Lookup and relationship”.



Na idućoj stranici se odabire da li se stupac za pretraživanje želi temeljiti na vrijednostima u tablici ili upitu ili na popisu vrijednosti koji se upiše, pri čemu je češća prva opcija. Zatim se odabire tablica, odnosno polja iz tablice na osnovu koje se želi kreirati stupac za pretraživanje. Odaberimo polje „Kod kluba” iz tablice „Klub”.

Stupac za pretraživanje možemo mijenjati desnim klikom na stupac i zatim „Modify Lookups”, odnosno obrisati s „Delete Field”.

ID	Ime	Prezime	Kod kluba	Visina	Težina	Datum rođenja	Kod kluba-1
2	Dario	Strawberry	CI	196	93	15.6.1985.	
3	Jozo	Brkić	ZD	211	101	16.12.1986.	
4	Romeo	Travis	ZD	201	100	10.12.1984.	
5	Predrag	Šuput	CD	200	111	1.6.1977.	
6	Bracey	Wright	CD	191	95	1.7.1984.	
7	Nikola	Mirotić	RM	209	107	11.2.1991.	
8	Dontaye	Draper	RM	180	82	10.8.1984.	
9	Kobe	Bryant	LL	198	93	23.8.1978.	
10	Steve	Nash	LL	191	81	7.2.1974.	
11	Chris	Paul	LC	183	79	6.5.1985.	
12	Blake	Griffin	LC	208	114	16.3.1989.	
13	Tyson	Chandler	NK	216	109	2.10.1982.	
14	Carmelo	Anthony	NK	203	104	29.5.1984.	
15	Deron	Williams	BN	191	95	26.6.1984.	
16	Mirza	Teletović	BN	206	116	17.9.1985.	
17	Nikola	Peković	MT	211	131	3.1.1986.	
18	Ricky	Rubio	MT	193	82	21.10.1990.	
*	(New)						

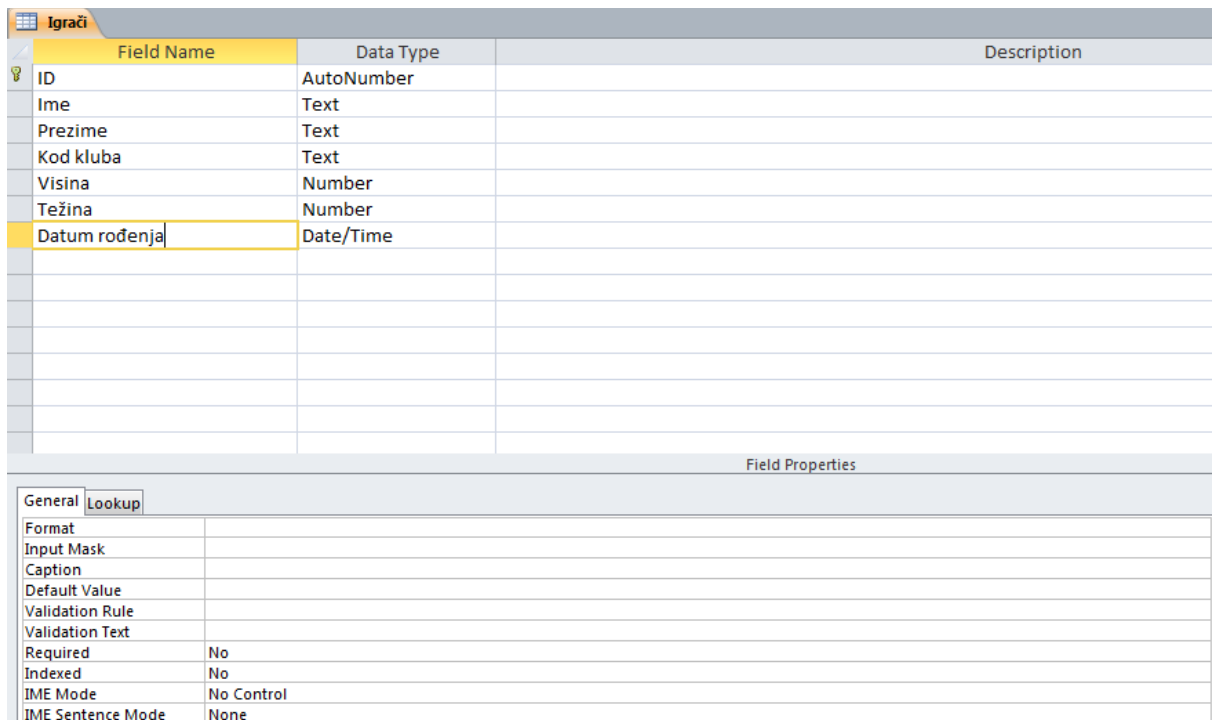
Kreiranje, mijenjanje i brisanje maske za unos podataka u polje/stupac

Maska za unos podataka služi za onemogućavanje unosa podataka neispravnog formata u bazu, odnosno osiguravanje unosa podataka ispravnog raspona.

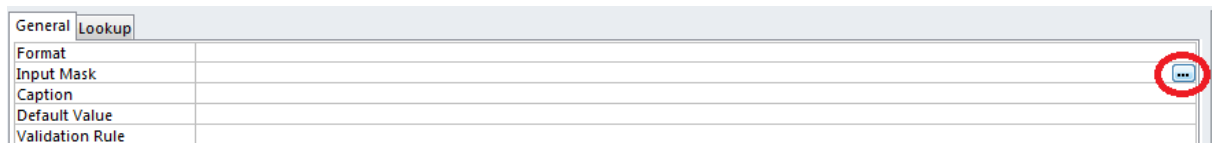
Na primjeru baze koja je kreirana, u tablici „Igrači” kreiramo masku za unos podataka na atributu „Datum rođenja”.

Field Name	Data Type
ID	AutoNumber
Ime	Text
Prezime	Text
Kod kluba	Text
	Number
	Number
	Date/Time

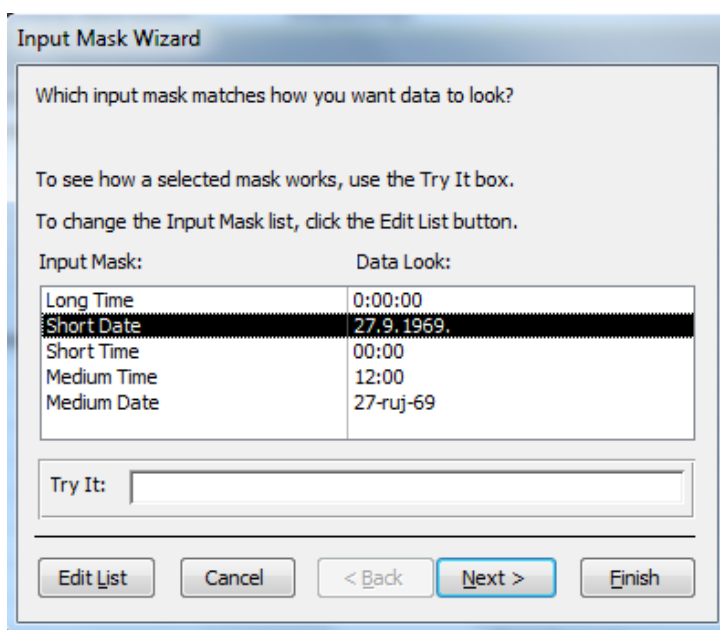
Desnim klikom na tablicu „Igrači” unutar izbornika „Tables”, otvara se padajući izbornik iz kojeg se izabire opcija „Design View”. Klikom na atribut/polje „Datum rođenja”, otvaraju se „Field Properties” s tabovima „General” i „Lookup”.



Unutar taba „General”, nalazi se opcija „Input Mask”. Klikom na tu opciju, otvara se mogućnost odabira gumba označenog s „...”.



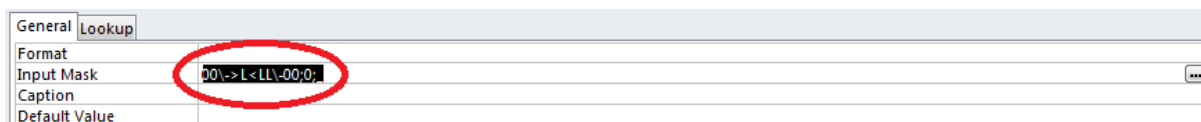
Klikom na gumb, otvara se prozor koji nudi predefimirane maske za unos podataka.



Odabirom opcije „Short Date”, i klikom na gumb „Finish”, kreiramo masku za unos podataka tako da će nadalje za polje „Datum rođenja” biti moguće unijeti jedino vrijednosti koje su u formatu oblika čiji je primjer prikazan u stupcu „Data Look” (u ovom slučaju 27.9.1969.).

Za mijenjanje maske za unos podataka ponovimo iste korake, te u zadnjem koraku odaberemo željenu masku za unos podataka. Primjerice, možemo odabrati opciju „Medium Date”, nakon čega je za polje „Datum rođenja” moguće unijeti samo podatke u formatu oblika prikazanog u stupcu „Data Look” (primjerice, 27.ruj.69).

Za brisanje maske za unos podataka, unutar prikaza „Field Properties”, taba „General”, mišom označimo masku (na slici označenu elipsom), te pobrišemo pritiskom tipke „Delete” ili „Backspace”.



Postavljanje obveznih unosa vrijednosti podataka na polje/stupac: potrebno, nije potrebno

Postavljanjem obveznog unosa vrijednosti podataka na polje/stupac, osigurava se da svi reci koji se unose u tablicu moraju imati definiranu vrijednost podatka u tom polju/stupcu.

Primjerice, možemo postaviti obvezan unos vrijednosti podatka na polje/stupac „Populacija” u tablici „Grad”. Kao i u prethodnom dijelu, desnim klikom na tablicu „Grad” unutar izbornika „Tables”, otvara se padajući izbornik iz kojeg se izabire opcija „Design View”. Klikom na atribut/polje „Populacija”, otvaraju se „Field Properties” s tabovima „General” i „Lookup”. Unutar taba „General”, na opciji „Required” definiranjem vrijednosti opcije na „Yes” postavlja se obvezu unosa vrijednosti podatka na polje/stupac „Populacija” u tablici „Grad”.

Field Name	Data Type	Description
Kod grada	Text	
Grad	Text	
Država	Text	
Populacija	Number	

Field Properties

General Lookup

Field Size: Long Integer

Format:

Decimal Places: Auto

Input Mask:

Caption:

Default Value:

Validation Rule:

Validation Text:

Required: No

Indexed: Yes

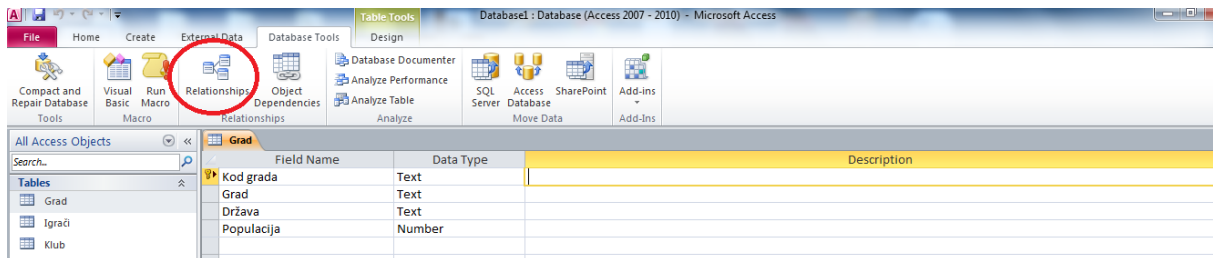
Smart Tags: No

Text Align: General

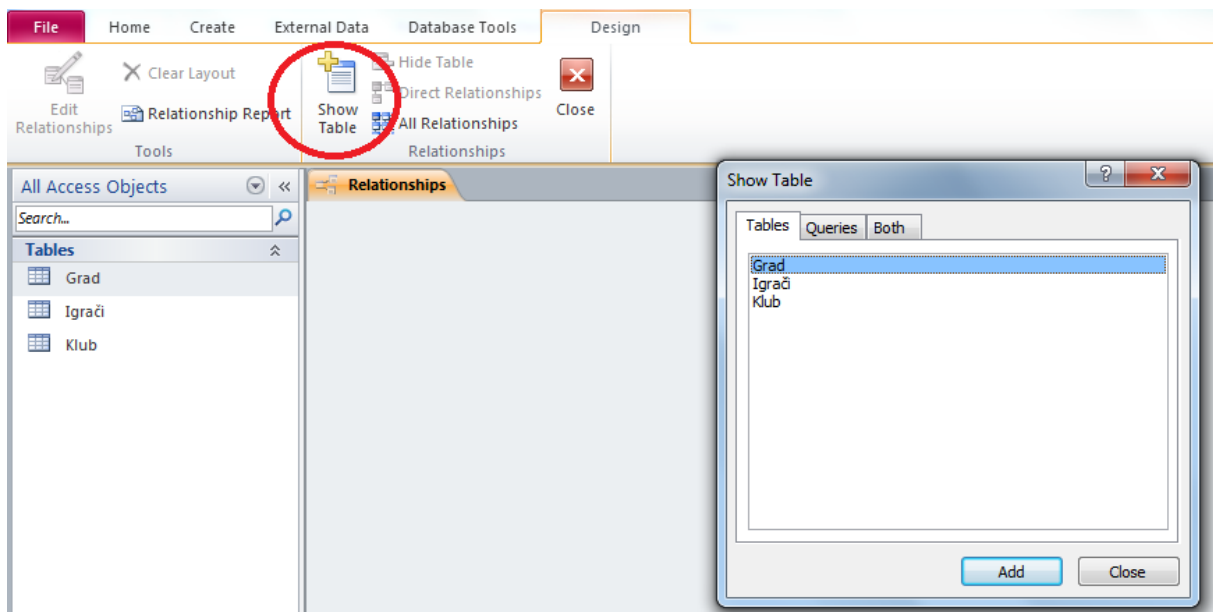
1.3 Relacije i spajanje

Kreiranje, mijenjanje i brisanje veza između tablica: jedan-na-jedan, jedan-na-više

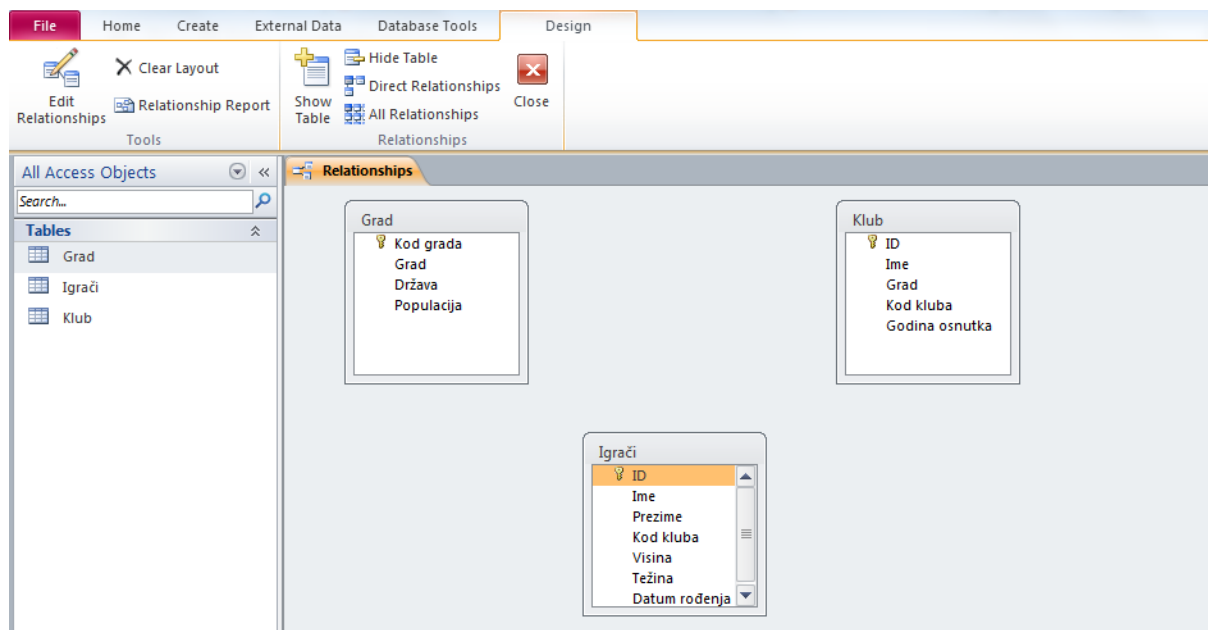
Za kreiranje veza između tablica, unutar taba „Database Tools“, u grupi „Relationship“, odaberemo „Relationships“.



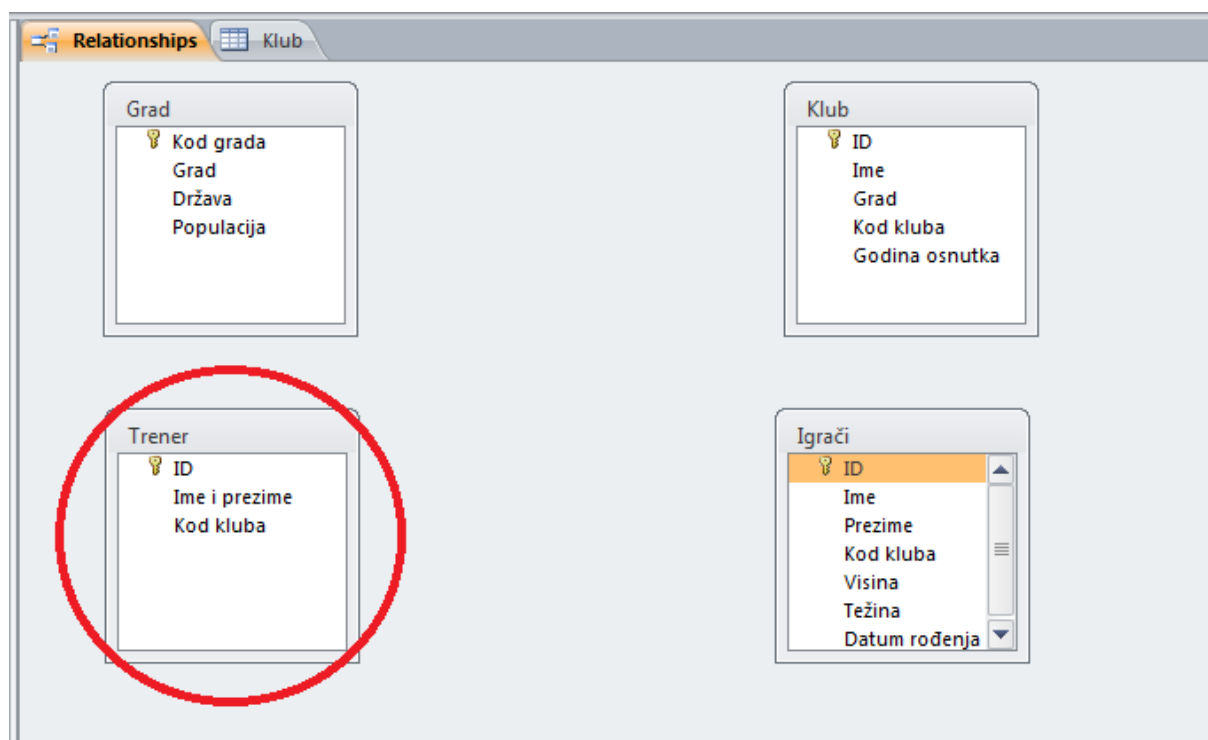
Zatim, unutar „Design“ taba, potrebno je kliknuti „Show Table“



Odabirom sve tri tablice i klikom na tipku „Add“, a zatim „Close“, možemo vidjeti tablice prikazane na prostoru označenom s „Relationships“.



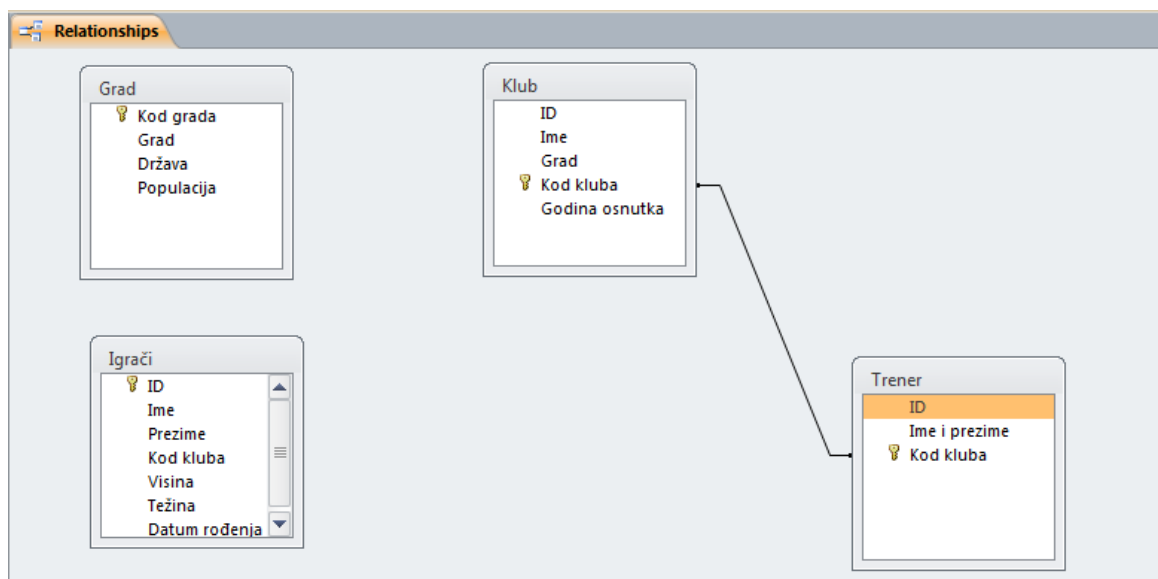
Za potrebe ovog primjera, kreirat ćemo i tablicu „Trener” čija su polja/stupci prikazani na slici u sklopu prikaza „Relationships”.



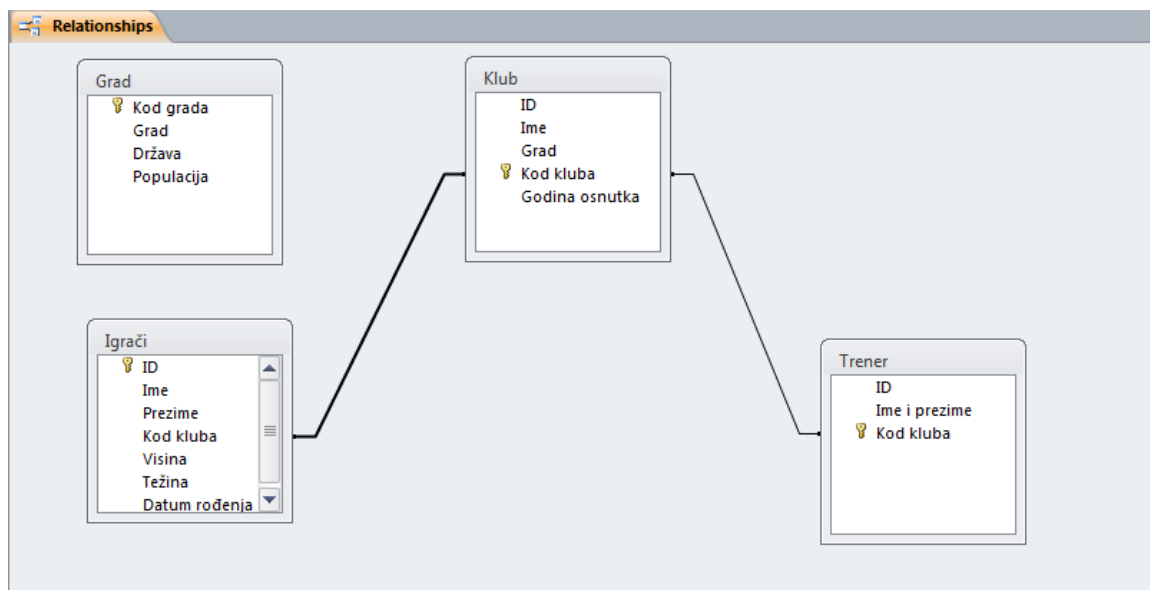
Vežu „jedan-na-jedan” možemo definirati između tablica „Trener” i „Klub”, budući da svaki trener (u pravilu) može voditi samo jedan klub, odnosno svaki klub (u pravilu) ima jednog trenera.

Klikom na polje (obično primarni ključ), držanjem tipke miša te otpuštanjem tipke na polju druge tablice (obično strani ključ) kreira se veza. U ovom slučaju, vezu kreiramo preko polja „Kod kluba”. Za potrebe ovog primjera, definiramo da je primarni ključ u obje tablice upravo

polje „Kod kluba”. Na taj način osiguravamo da svaki redak u tablicama „Klub” i „Trener” ima jedinstvenu vrijednost polja „Kod kluba”. Na ovaj način, kreiramo vezu jedan-na-jedan.



Kao primjer veze jedan-na-više, kreirat ćemo vezu između tablica „Klub” i „Igrači”. Jedan klub može imati više igrača, dok jedan igrač može imati samo jedan klub. Veza se kreira preko polja „Kod kluba” što je primarni ključ tablice „Klub” i strani ključ u tablici „Igrači”.



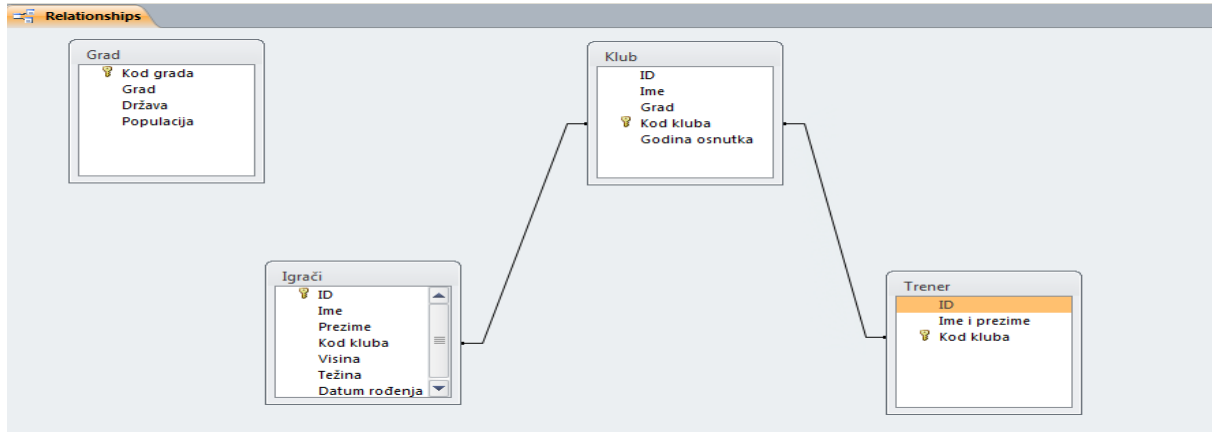
1.4 Kreiranje i mijenjanje veza između tablica: više-na-više koristeći spojne tablice

Primjena referencijalnog integriteta između tablica

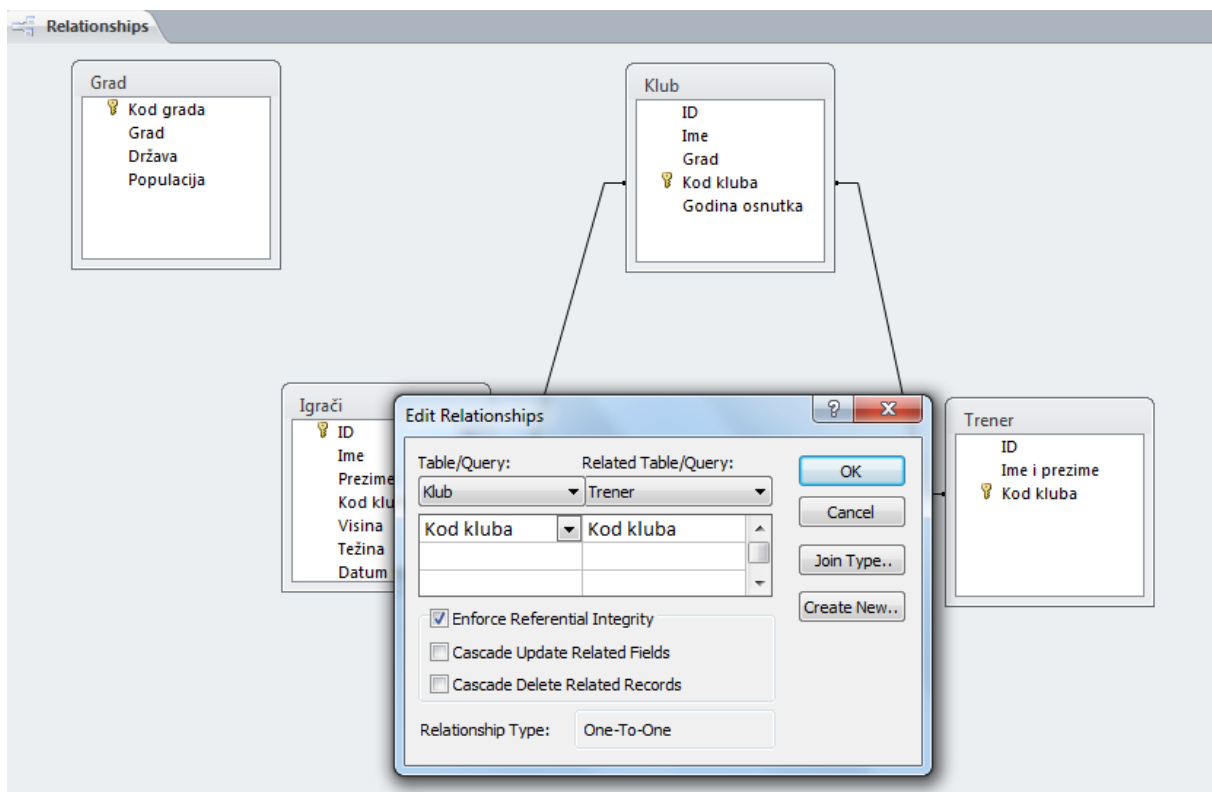
Uloga referencijalnog integriteta je sprječavanje pojavljivanja zapisa koji nikamo ne pripadaju, te održavanje referenci sinkroniziranima kako ne bi došlo do zapisa koji referiraju druge zapise koji više ne postoje. Referencijalni integritet se definira na vezama, nakon čega MS Access odbacuje sve operacije koje bi povrijedile referencijalni integritet za odnos te

tablice (odbijaju se ažuriranja koja mijenjaju cilj reference i brisanja koja uklanjaju cilj reference).

Prema prethodnim primjerima, otvorimo prikaz „Relationships” u kojem otvorimo sve postojeće tablice.



Ukoliko odaberemo (kliknemo) vezu između tablica „Klub” i „Trener”, i odaberemo opciju „Enforce Referential Integrity”, definirali smo referencijalni integritet.



Nakon definiranja referencijalnog integriteta, primjenjuju se sljedeća pravila:

- Ne može se unijeti vrijednost u polje vanjskog ključa povezane tablice (polje „Kod kluba” u tablici „Trener”) ako ta vrijednost ne postoji u polju primarnog ključa primarne tablice (polje „Kod kluba” u tablici „Klub”)

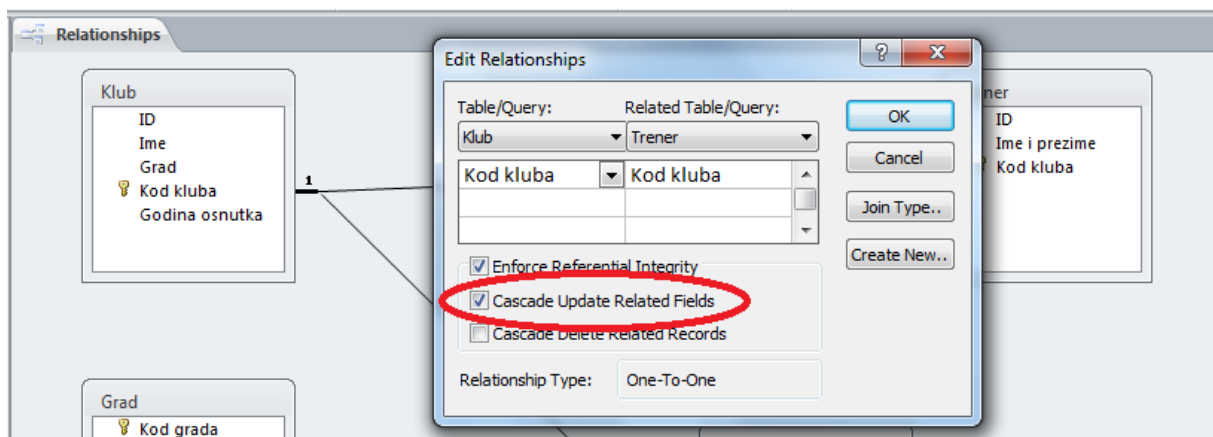
- Ne može se izbrisati zapis iz primarne tablice („Klub”) ako u povezanoj tablici („Trener”) postoji zapis koji se podudara
- U primarnoj tablici („Klub”) ne može se promijeniti vrijednost primarnog ključa ako bi time nastali zapisi koji nikome ne pripadaju. Primjerice, ne mogu se promijeniti vrijednosti atributa u nekom retku tablice „Klub” ako postoji redak dodijeljen tom retku u tablici „Trener”

Primjena automatskog ažuriranja povezanih polja

Postoje situacije u kojoj postoji potreba za promjenom vrijednosti s jedne strane veze između tablica. U tom slučaju je potrebno da se automatski ažuriraju svi reci koji su pod utjecajem te operacije čime se osigurava da se baza potpuno ažurira kako podaci ne bi bili nedosljedni.

Kako bi ostvarili automatsko ažuriranje, potrebno je najprije definirati referencijalni integritet nad vezom, a zatim automatsko ažuriranje.

Primjerice, klikom na vezu između tablica „Klub” i „Trener”, automatsko ažuriranje definiramo označavanjem opcije „Cascade Update Related Fields”.



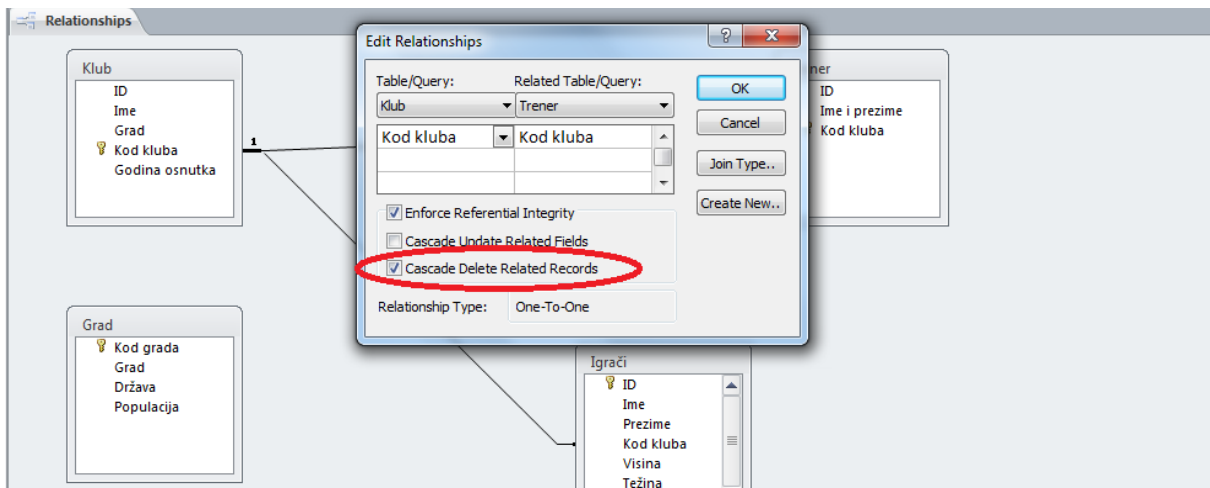
Nakon definiranja automatskog ažuriranja, bilo koja promjena u tablici „Klub” na polju „Kod kluba”, automatski se ažurira u tablici „Trener”.

Primjena automatskog brisanja povezanih zapisa

Ponekad postoji potreba za brisanjem retka i automatskim brisanjem svih povezanih zapisa.

Kako bi ostvarili automatsko brisanje, potrebno je najprije definirati referencijalni integritet nad vezom, a zatim automatsko brisanje.

Primjerice, klikom na vezu između tablica „Klub” i „Trener”, automatsko brisanje definiramo označavanjem opcije „Cascade Delete Related Fields”.



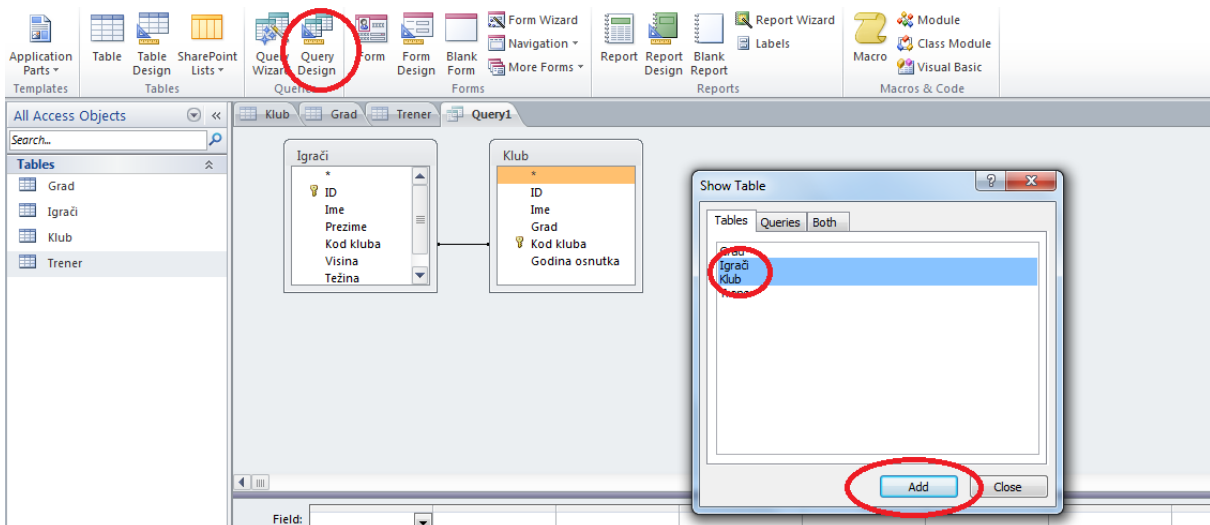
Nakon definiranja automatskog brisanja, brisanje zapisa u tablici „Klub” automatski se brišu zapisi u tablici „Trener” koji referiraju primarni ključ obrisanog zapisa u tablici „Klub”.

Primjena i promjena unutarnjeg i vanjskog spajanja

Spojevi služe da bi se kombinirali zapisi iz oba podatkovna izvora tako da svaki par zapisa iz izvora u rezultatima upita postaje jedan zapis.

Unutarnji spoj je onaj u kojem se uvrštavaju podaci iz tablice samo ako postoje odgovarajući podaci u povezanoj tablici i obratno. Ukoliko su tablice već u nekom odnosu, onda Access stvara unutarnji spoj kojim održava taj spoj. Unutarnji spojevi se najčešće koriste. Kad stvaramo spoj i ne navedemo njegov tip, Access pretpostavlja da želimo unutarnji spoj.

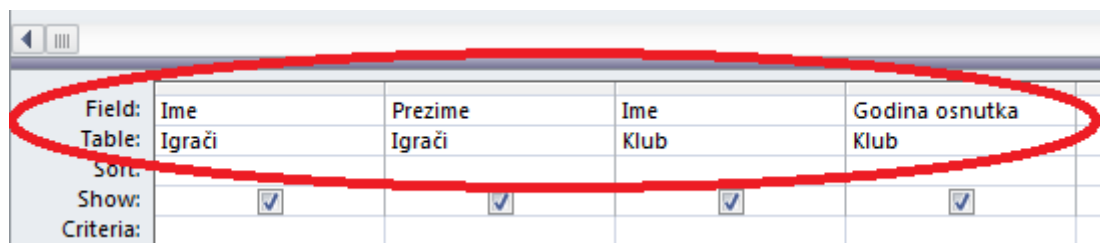
Primjerice, da bi definirali unutarnji spoj između tablica „Klub” i „Igrači”, unutar taba „Create”, u grupi „Queries”, odaberemo „Query Design”. U „Show Table” dijalogu, odaberimo tablice „Klub” i „Igrači” i kliknemo „Add”.



Možemo vidjeti da već postoji spoj između ove dvije tablice. To je spoj koji je kreiran automatski pošto smo već ranije definirali vezu između ove dvije tablice.

Ukoliko dvaput kliknemo na vezu između ovih tablica, u prikazanom dijalogu „Join Properties” možemo vidjeti da je označena prva opcija: „Only include rows where the joined fields from both tables are equal.” što predstavlja unutarnji spoj.

U donjem dijelu ekrana možemo odabrati što želimo prikazati kao rezultat upita. Odaberimo polja „Ime” i „Prezime” iz tablice „Igrači”, te polja „Ime” i „Godina osnutka” iz tablice „Klub”.



Kliknemo na opciju „View” u grupi „Results” da bi prikazali rezultat.

Igrači.Ime	Prezime	Klub.Ime	Godina osnu
Deron	Williams	Nets	1967
Mirza	Teletović	Nets	1967
Dario	Šarić	Cibona	1946
Darryl Eugene	Strawberry	Cibona	1946
Jozo	Brkić	Zadar	1945
Romeo	Travis	Zadar	1945
Predrag	Šuput	Cedevita	1991
Bracey	Wright	Cedevita	1991
Nikola	Mirotić	Real	1931
Dontaye	Draper	Real	1931
Kobe	Bryant	Lakers	1947
Steve	Nash	Lakers	1947
Chris	Paul	Clippers	1970
Blake	Griffin	Clippers	1970
Tyson	Chandler	Knicks	1946
Carmelo	Anthony	Knicks	1946
Nikola	Peković	Wolves	1989
Ricky	Rubio	Wolves	1989
*			

Vanjski spojevi je isto što i unutarnji, samo što uvrštava i preostale retke iz jedne od tablica. Vanjski spojevi sadrže i podatke o smjeru. Lijevi vanjski zapis sadrži sve zapise iz lijeve tablice (prve tablice u spoju), a desni vanjski zapis sadrži sve zapise iz desne tablice (druge tablice u spoju).

Primjerice, ukoliko ponovno odaberemo spoj između tablica „Klub” i „Igrači” i ovaj put označimo opciju 2: „Include all records from „Klub” and only those records from „Igrači” where the joined fields are equal., i prikažemo rezultate, vidimo da su prikazani svi zapisi iz

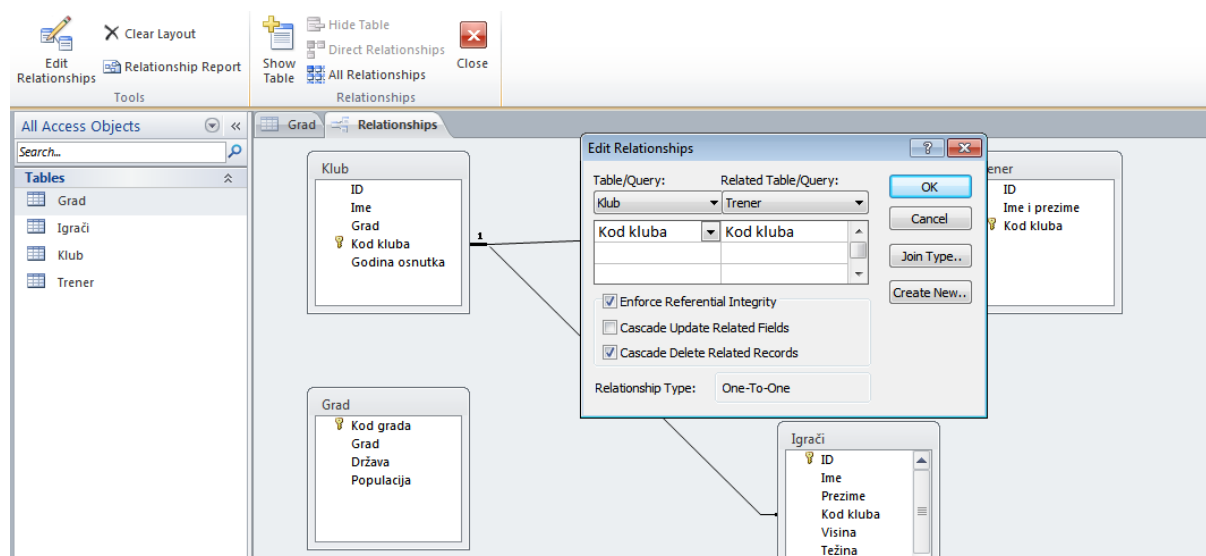
tablice „Klub”, pa čak i oni koji nemaju odgovarajući par u tablici „Igrači” (redak koji ima vrijednost polja „Klub.Ime” jednaku „Split”).

Igrači.Ime	Prezime	Klub.Ime	Godina osnu
Deron	Williams	Nets	1967
Mirza	Teletović	Nets	1967
Dario	Šarić	Cibona	1946
Darryl Eugene	Strawberry	Cibona	1946
		Split	1945
Jozo	Brkić	Zadar	1945
Romeo	Travis	Zadar	1945
Predrag	Šuput	Cedevita	1991
Bracey	Wright	Cedevita	1991
Nikola	Mirotić	Real	1931
Dontaye	Draper	Real	1931
Kobe	Bryant	Lakers	1947
Steve	Nash	Lakers	1947
Chris	Paul	Clippers	1970
Blake	Griffin	Clippers	1970
Tyson	Chandler	Knicks	1946
Carmelo	Anthony	Knicks	1946
Nikola	Peković	Wolves	1989
Ricky	Rubio	Wolves	1989
*			

Kreiranje i promjena vrste spajanja

Odaberimo tab „Database Tools” i zatim „Relationships”. Promjenu veze možemo napraviti dvoklikom na crtu odnosa ili na kartici „Dizajn” odaberimo „Edit Relationships”.

Pojavljuje se dijaloški okvir u kojem možemo editirati postojeće spajanje.



UPITI

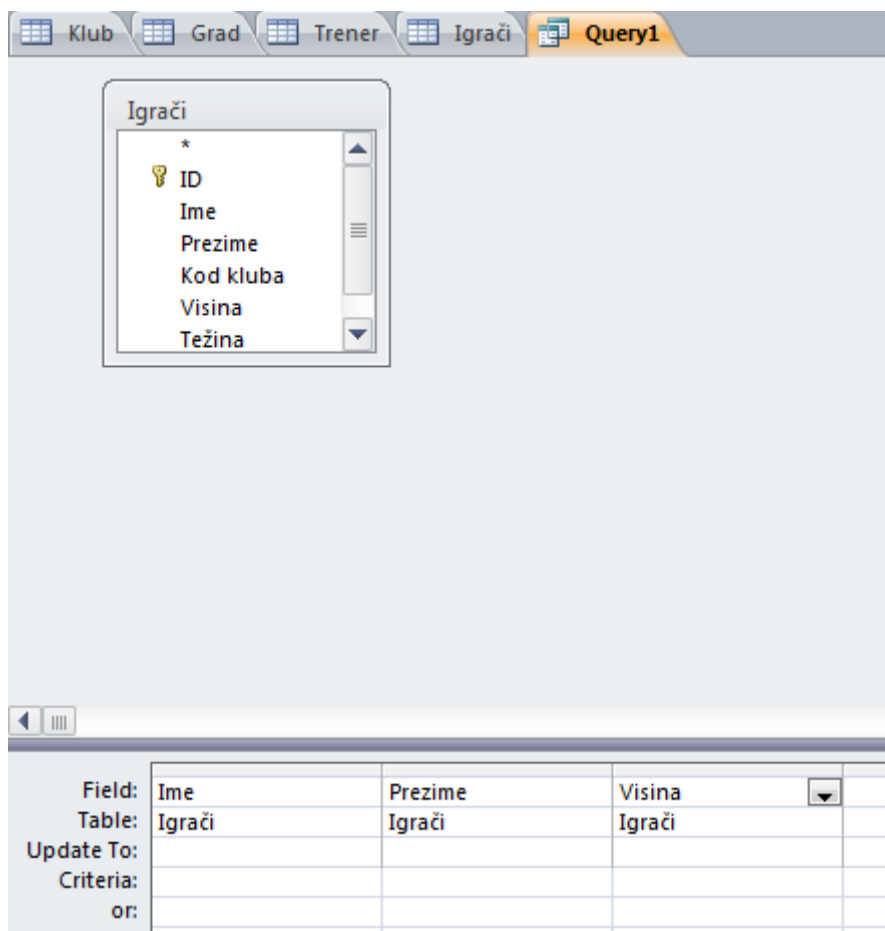
1.5 Vrste upita

Kreiranje i pokretanje upita za ažuriranje podataka u tablici

Upiti za ažuriranje se koriste za dodavanje, mijenjanje (da bi se postojeća polja bez vrijednosti promijenila u vrijednosti različite od „null”) ili brisanje podataka iz postojećih zapisa. Prilikom vježbanja kreiranja upita za ažuriranje podataka u tablici, korisno je napraviti kopiju baze da bi sačuvali originalne podatke.

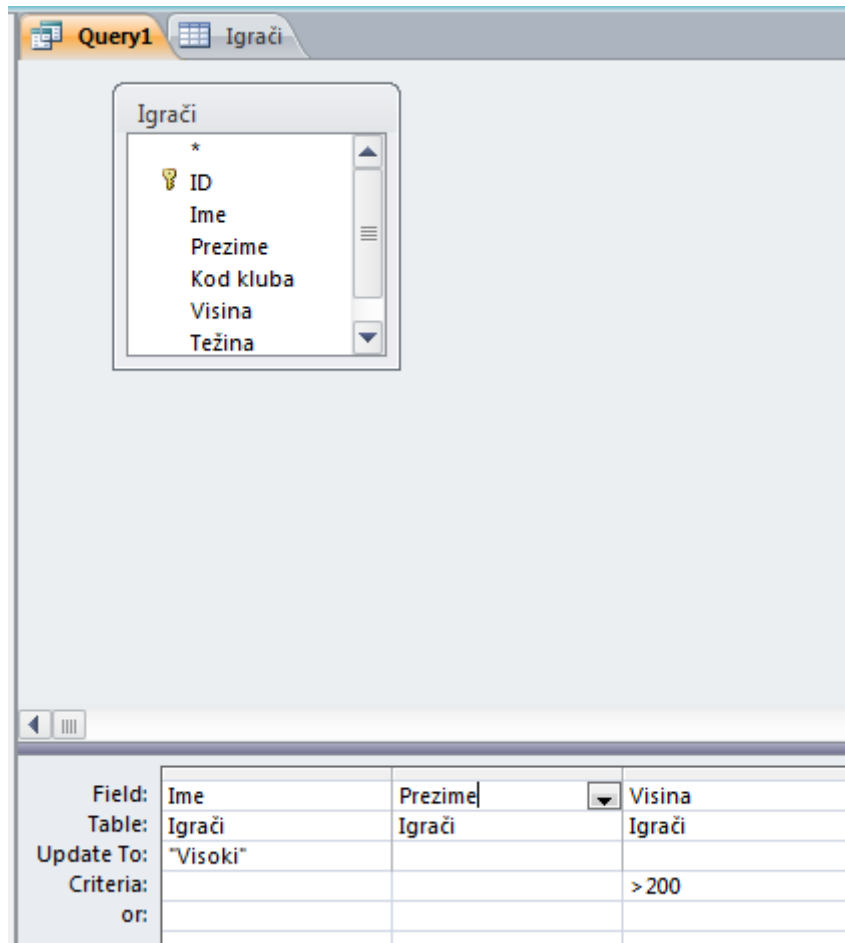
Kao u ranijim primjerima, otvorimo „Query Design” unutar taba „Create”, te odaberimo „Update” unutar „Query Type”. U dijalogu „Show Table” odaberimo, primjerice tablicu „Igrači”.

Dvoklikom na polja koja želimo ažurirati, polja se pojavljuju u prikazu, koordinatnoj mreži za dizajn upita.



Primjerice, želimo prikazati imena i prezimena svih igrača koji su visoki više od 200cm, te ćemo za takve igrače ažurirati vrijednost polja „Visina” u „DA”.

U dijelu „Criteria” koordinatne mreže za dizajn upita, definiramo kriterij za polje visina kao „>200”, čime ćemo kao rezultat upita dobiti samo one retke koji gdje je polje visina veće od 200. U dijelu „Update to” za polje „Ime”, napisat ćemo „Visoki”.



Klikom na „Run” u dijelu „Results” pokrećemo upit. Nakon pokretanja, možemo vidjeti da su se u tablici „Igrači” ažurirale vrijednosti polja „Ime” gdje je visina veća od 200.

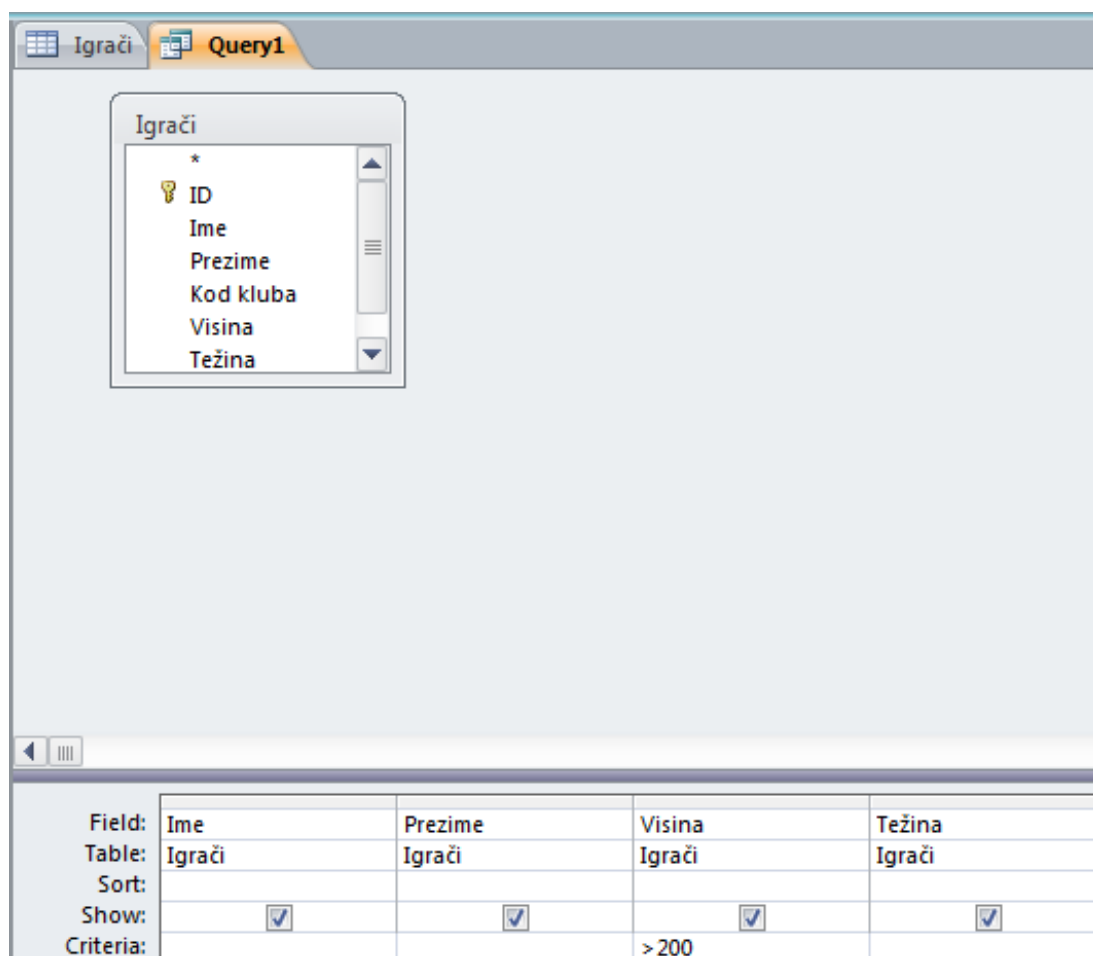
ID	Ime	Prezime	Kod kluba	Visina	Težina	Datum rođe	Click to Add
1	Visoki	Šarić	CI	208	104	8.4.1994.	
2	Darryl Eugene	Strawberry	CI	196	93	15.6.1985.	
3	Visoki	Brkić	ZD	211	101	16.12.1986.	
4	Visoki	Travis	ZD	201	100	10.12.1984.	
5	Predrag	Šuput	CD	200	111	1.6.1977.	
6	Bracey	Wright	CD	191	95	1.7.1984.	
7	Visoki	Mirotić	RM	209	107	11.2.1991.	
8	Dontaye	Draper	RM	180	82	10.8.1984.	
9	Kobe	Bryant	LL	198	93	23.8.1978.	
10	Steve	Nash	LL	191	81	7.2.1974.	
11	Chris	Paul	LC	183	79	6.5.1985.	
12	Visoki	Griffin	LC	208	114	16.3.1989.	
13	Visoki	Chandler	NK	216	109	2.10.1982.	
14	Visoki	Anthony	NK	203	104	29.5.1984.	
15	Deron	Williams	BN	191	95	26.6.1984.	
16	Visoki	Teletović	BN	206	116	17.9.1985.	
17	Visoki	Peković	MT	211	131	3.1.1986.	
18	Ricky	Rubio	MT	193	82	21.10.1990.	

Kreiranje i pokretanje upita za dodavanje zapisa u tablici

Za primjer kreiranja i pokretanja upita za dodavanje zapisa u tablicu, kreirat ćemo tablicu „Igrači_2” te ćemo je napuniti zapisima iz tablice „Igrači”. U tablicu „Igrači_2”, dodat ćemo sve one zapise iz tablice „Igrači” kod kojih je vrijednost polja „Visina” veća od 200.

Prvi korak je stvaranje upita za odabir zapisa za kopiranje. Kao u prethodnom primjeru, otvorimo „Query Design” i tablicu „Igrači”, a unutar „Query Type”, odaberimo „Select”.

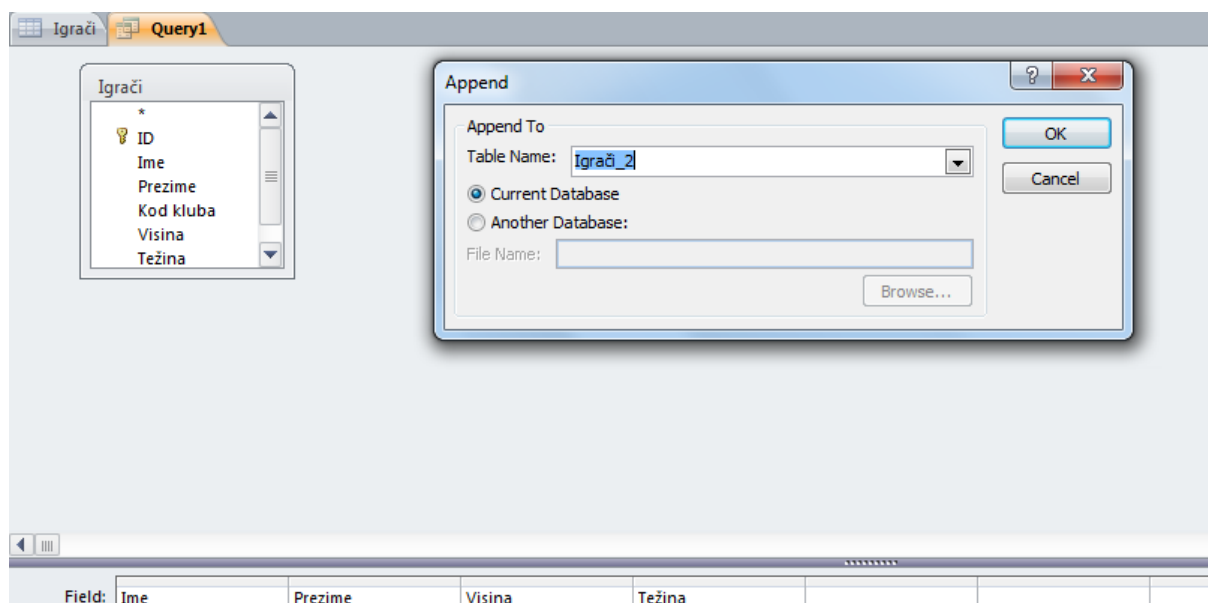
Dvoklikom na polja, odaberemo sva polja koja želimo dodati. Odaberimo „Ime”, „Prezime”, „Visina” i „Težina”. Unutar opcije „Criteria”, na polju „Visina” definiramo kriterij „> 200”.



Klikom na „Run”, pokrenimo upit koji vraća retke s vrijednošću polja „Visina” većom od 200, i to polja „Ime”, „Prezime”, „Visina” i „Težina”.

Ime	Prezime	Visina	Težina
Dario	Šarić	208	104
Jozo	Brkić	211	101
Romeo	Travis	201	100
Nikola	Mirotić	209	107
Blake	Griffin	208	114
Tyson	Chandler	216	109
Carmelo	Anthony	203	104
Mirza	Teletović	206	116
Nikola	Peković	211	131
*			

Drugi korak je pretvorba upita s odabiranjem u upit s dodavanjem.. U tabu „Home”, u grupi „View”, odaberimo „Design View”. U grupi „Query Type”, odaberimo „Append” kao vrstu upita. U dijalogu koji se otvori, odaberimo tablicu „Igrači_2” i kliknimo „OK”.



U koordinatnoj mreži za dizajn upita, imena atributa „Append To” trebaju odgovarati željenim imenima polja u koje se žele postaviti zapisi, ukoliko to nije slučaj, upit za dodavanje će pogrešno dodati nove zapise. Primjerice, ukoliko postavimo za polje „Ime” vrijednost „Append To” na „Prezime”, upit će dodati zapise s vrijednosti polja „Prezime” unutar vrijednosti polja „Ime”.

Field:	Ime	Prezime	Visina	Težina
Table:	Igrači	Igrači	Igrači	Igrači
Sort:				
Append To:	Ime	Prezime	Visina	Težina
Criteria:				
or:				

Pokrenimo upit klikom na „Run” i dobit ćemo željene zapise u tablici „Igrači_2”, odnosno one zapise iz tablice „Igrači” gdje je vrijednost polja „Visina veća od 200.

ID	Ime	Prezime	Kod kluba	Visina	Težina	Datum rođe	Click to Add
1	Dario	Šarić		208	104		
2	Jozo	Brkić		211	101		
3	Romeo	Travis		201	100		
4	Nikola	Mirotić		209	107		
5	Blake	Griffin		208	114		
6	Tyson	Chandler		216	109		
7	Carmelo	Anthony		203	104		
8	Mirza	Teletović		206	116		
9	Nikola	Peković		211	131		
*	(New)						

Kreiranje i pokretanje upita za brisanje u tablici

Kao primjer kreiranja upita za brisanje u tablici, kreirat ćemo upit koji će obrisati sve retke s vrijednošću polja Visina većom od 210 u tablici „Igrači_2” koju smo kreirali i napunili u prethodnom dijelu.

Otvorimo „Query Design” unutar taba „Create”. U dijalogu „Show Table” odaberimo tablicu „Igrači_2”. Odaberimo „Delete” unutar „Query Type”.

Dvoklikom na polja u tablici, dodajmo sva polja iz tablice u koordinatnu mrežu, i definirajmo kriterij za polje „Visina” veće od 210.

Field:	Ime	Prezime	Kod kluba	Visina	Težina
Table:	Igrači_2	Igrači_2	Igrači_2	Igrači_2	Igrači_2
Delete:	Where	Where	Where	Where	Where
Criteria:				>210	
or:					

Pokrenimo upit klikom na „Run”.

Nakon osvježavanja, u tablici su ostali samo zapisi s vrijednošću polja „Visina” manjom od 210.

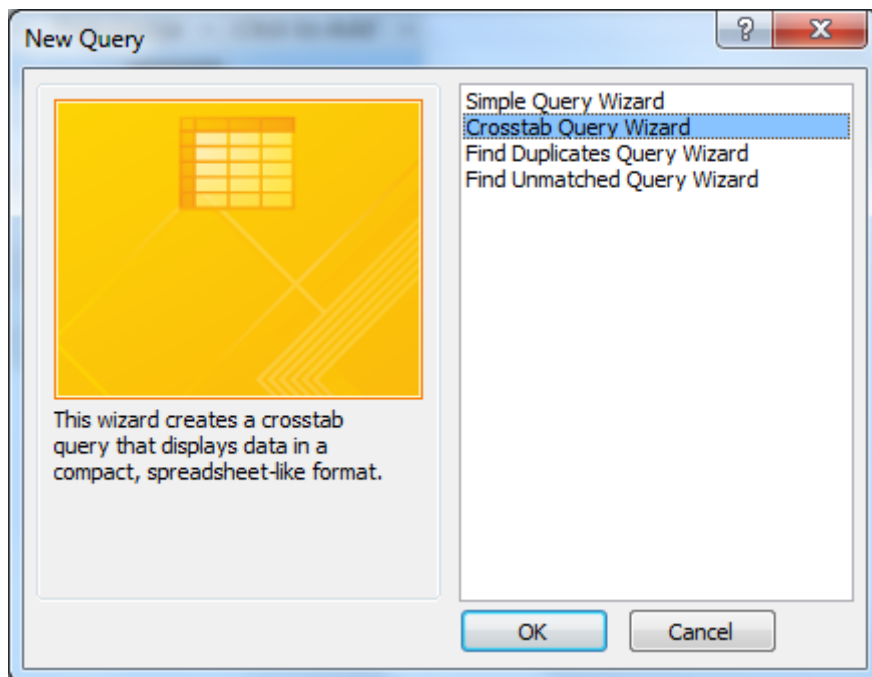
ID	Ime	Prezime	Kod kluba	Visina	Težina	Datum rođe	Click to Add
1	Dario	Šarić		208	104		
3	Romeo	Travis		201	100		
4	Nikola	Mirotić		209	107		
5	Blake	Griffin		208	114		
7	Carmelo	Anthony		203	104		
8	Mirza	Teletović		206	116		
*	(New)						

1.6 Kreiranje i pokretanje upita za spremanje odabranih podataka kao nove tablice

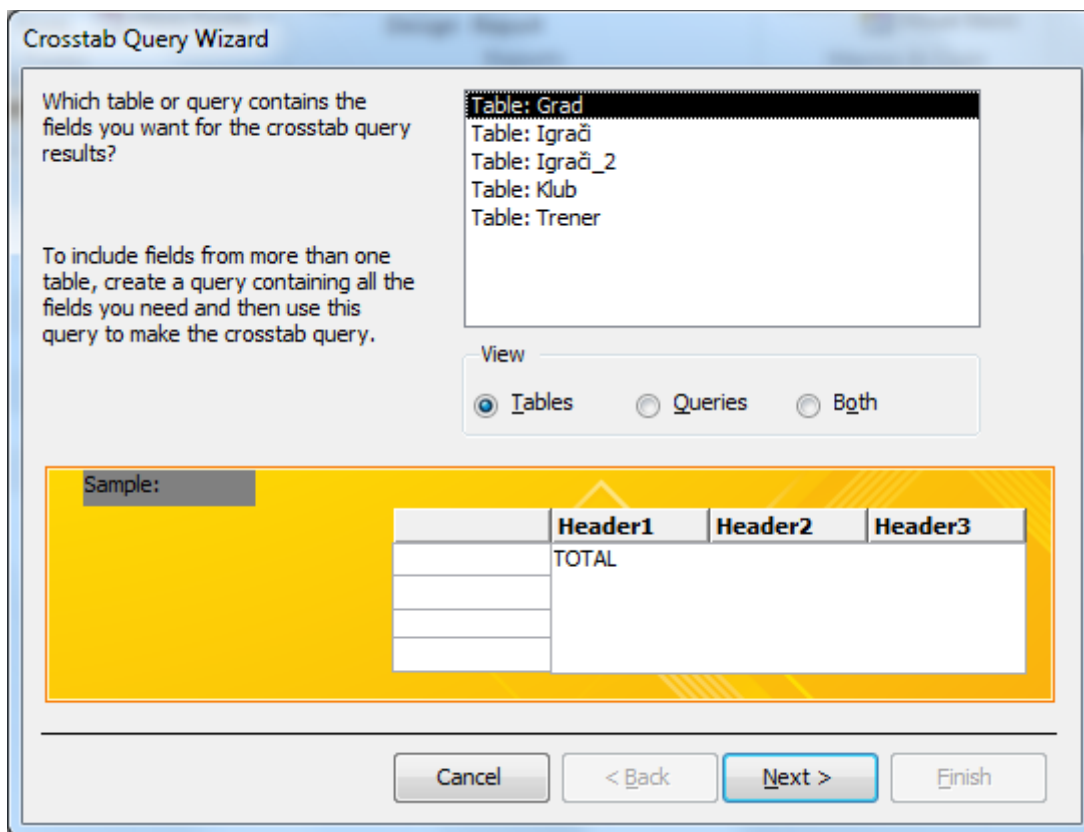
Kreiranje i pokretanje unakrsnog upita

Unakrsni upit je vrsta upita s odabirom. Njegova prednost je ta što ponekad omogućuje lakše čitanje od jednostavnog upita s odabirom koji prikazuje iste podatke. Unakrsni upit izračunava zbroj, prosjek i druge funkcije zbrajanja, a zatim grupira rezultate po dva skupa vrijednosti - jedan sa strane podatkovnog lista i drugi po gornjem retku.

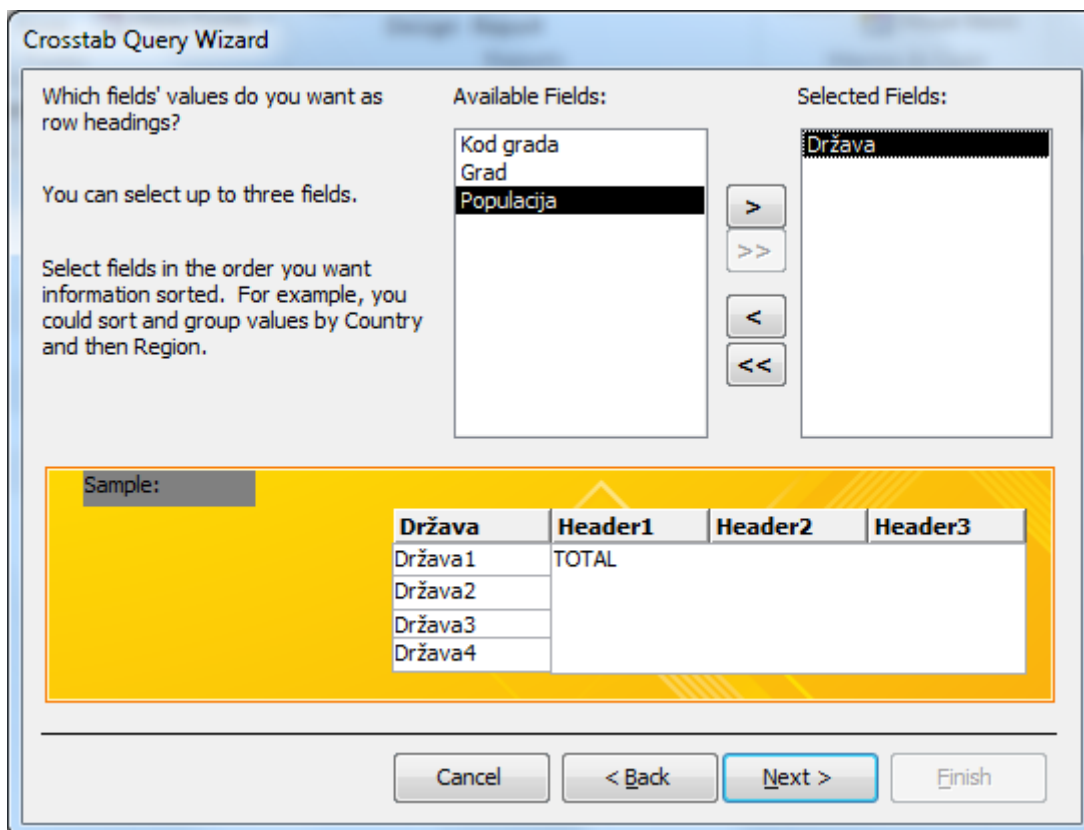
Za stvaranje unakrsnog upita, najbrži i najjednostavniji način je korištenje čarobnjaka za unakrsne upite. Na tabu „Create” u grupi „Queries”, odaberimo „Query Wizard”. U dijalogu „New Query”, odaberimo „Crosstab Query Wizard”.



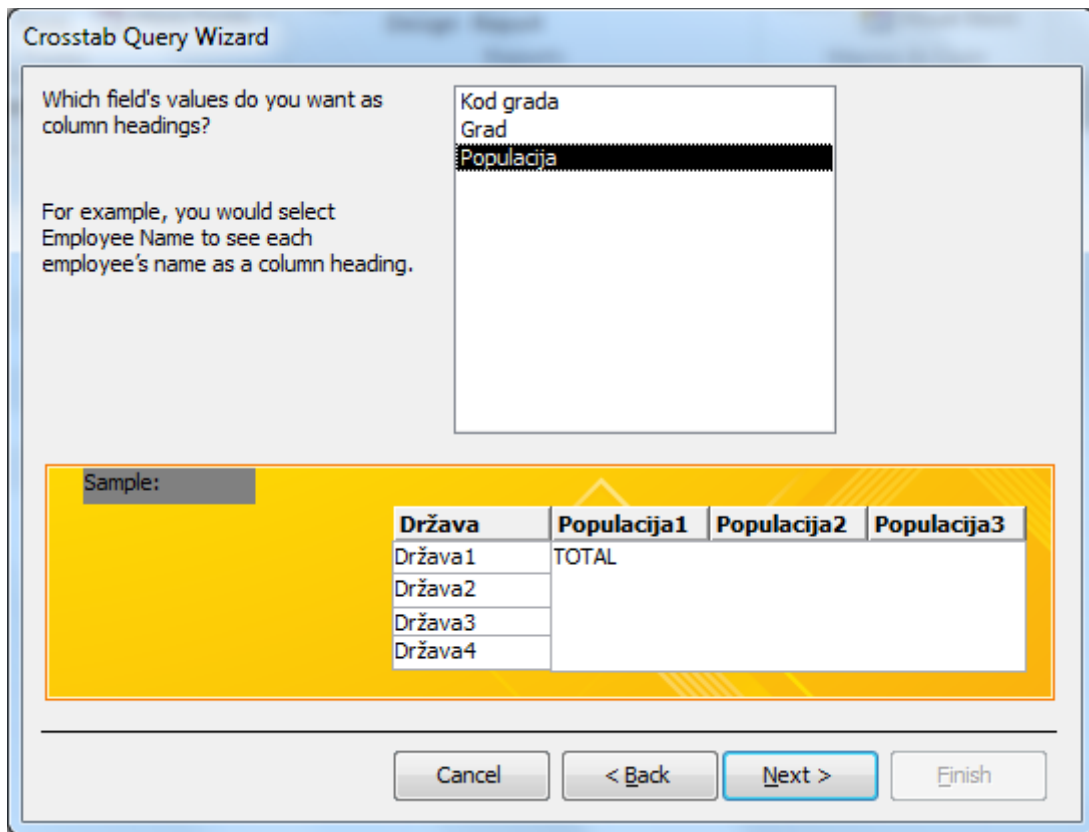
Zatim odaberimo tablicu ili upit koji želimo koristiti pri stvaranju unakrsnog upita. Odaberimo tablicu „Grad”.



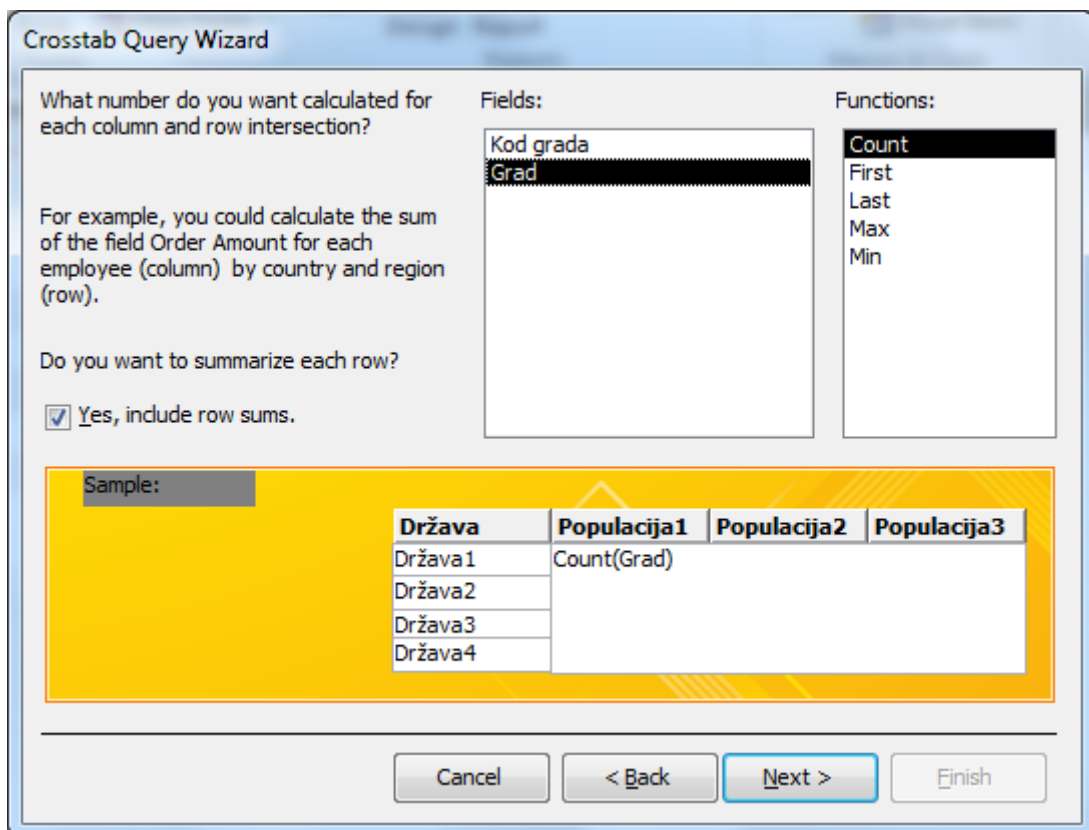
U idućem dijaloškom okviru, odaberemo polja koja želimo koristiti za zaglavlja redaka. Odaberimo polje „Država“.



U idućem dijaloškom okviru odabiremo polje koje želimo koristiti za zaglavlje stupaca. Odaberimo polje „Populacija“.

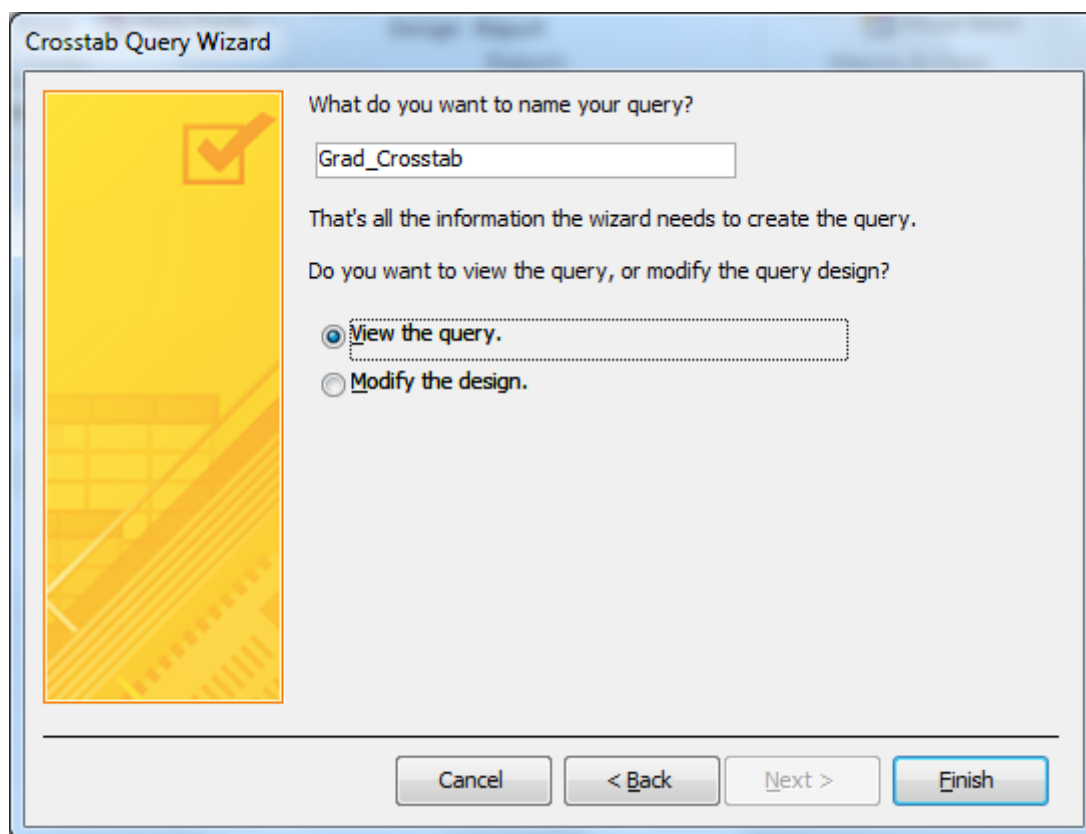


U idućem dijalogu se odabire polje i funkcija za izračun sažetih vrijednosti. Odaberimo polje „Grad“ i funkciju „Count“.



Na istom dijalogu možemo potvrditi ili poništiti opciju „Yes, include row sums.“ Čime se uvrštava ili izostavlja zbroj redaka.

Na idućem dijalogu, biramo ime upita, kao i opciju hoćemo li prikazati upit ili se vratiti na modificiranje upita. Prikažimo upit označavanjem opcije „View the query“.



Pogledajmo dobiveni upit.

Država	Total Of Gra	70000	200000	400000	800000	3200000	3800000	8300000
Hrvatska	3	1	1		1			
SAD	3			1			1	1
Španjolska	1					1		

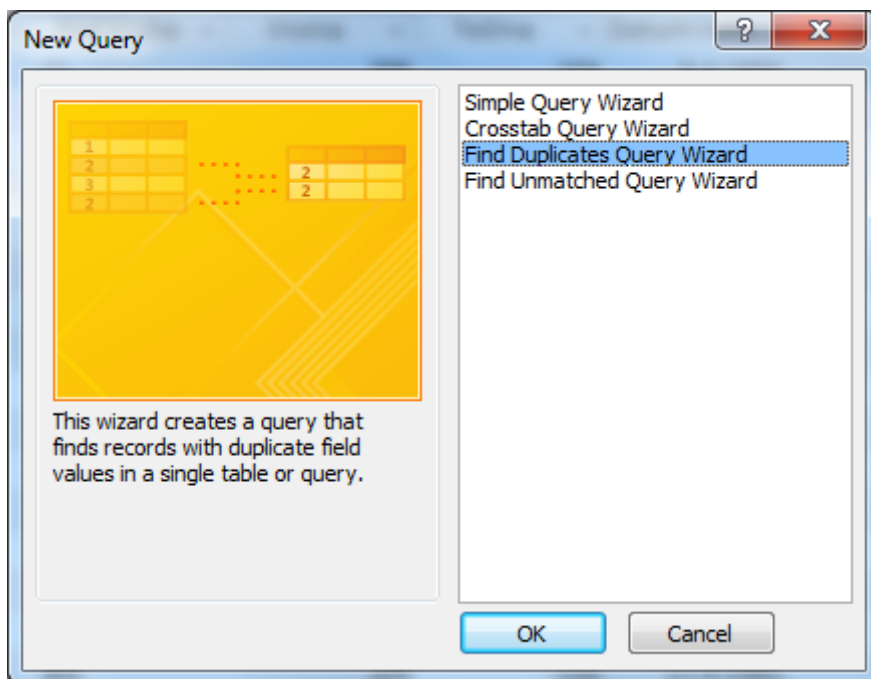
Unakrsni upit koji smo načinili je sumirao broj gradova iz tablice „Grad“ koji imaju polje „Država“, tako da smo u polju „Total Of Grad“ dobili 3 za Hrvatsku, 3 za SAD i 1 za Španjolsku. Također, u daljnjim poljima označen je broj gradova s određenim populacijama koji postoje u pojedinim državama (gledajući tablicu „Grad“).

Kreiranje i pokretanje upita za prikaz zapisa koji se ponavljaju u tablici

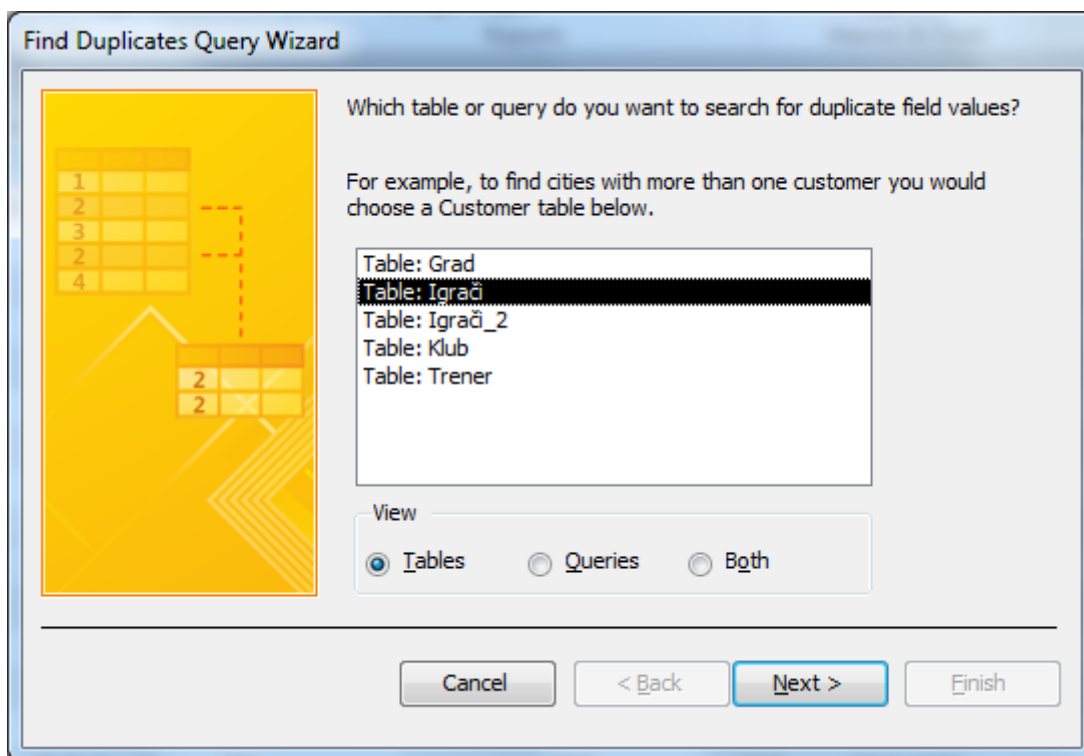
Za pronalaženje zapisa u kojima se neka polja djelomično ili u cijelosti podudaraju, najprikladnije je koristiti čarobnjak za kreiranje upita.

U ovom primjeru, koristit ćemo ovaj tip upita da bi našli djelomične duplikate u tablici „Igrači“, u ovom primjeru tražit ćemo one zapise koji imaju iste vrijednosti polja „Kod kluba“. Potpuni duplikati su oni koji imaju vrijednosti svih polja iste, dok djelomični imaju jedno ili više polja s istom vrijednošću, ali ne sve.

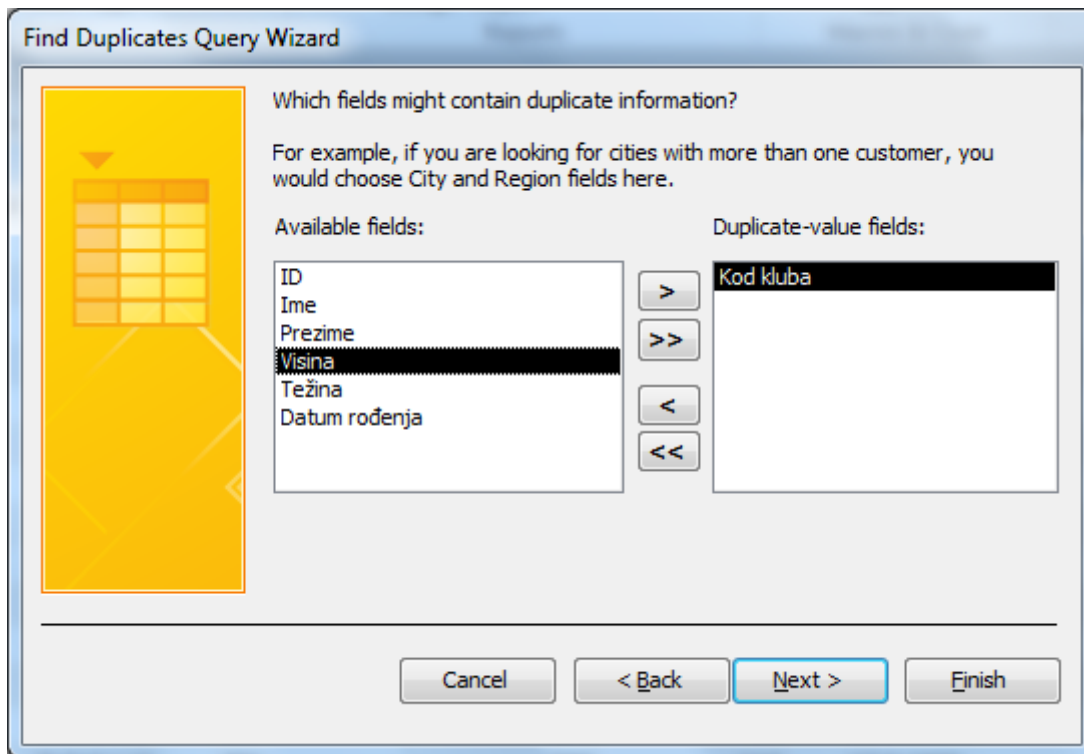
Kao u prošlom dijelu, koristimo „Query Wizard“ s razlikom što ćemo ovaj put u dijalogu „New Query“ odabrati „Find Duplicates Query Wizard“ umjesto „Crosstab Query Wizard“.



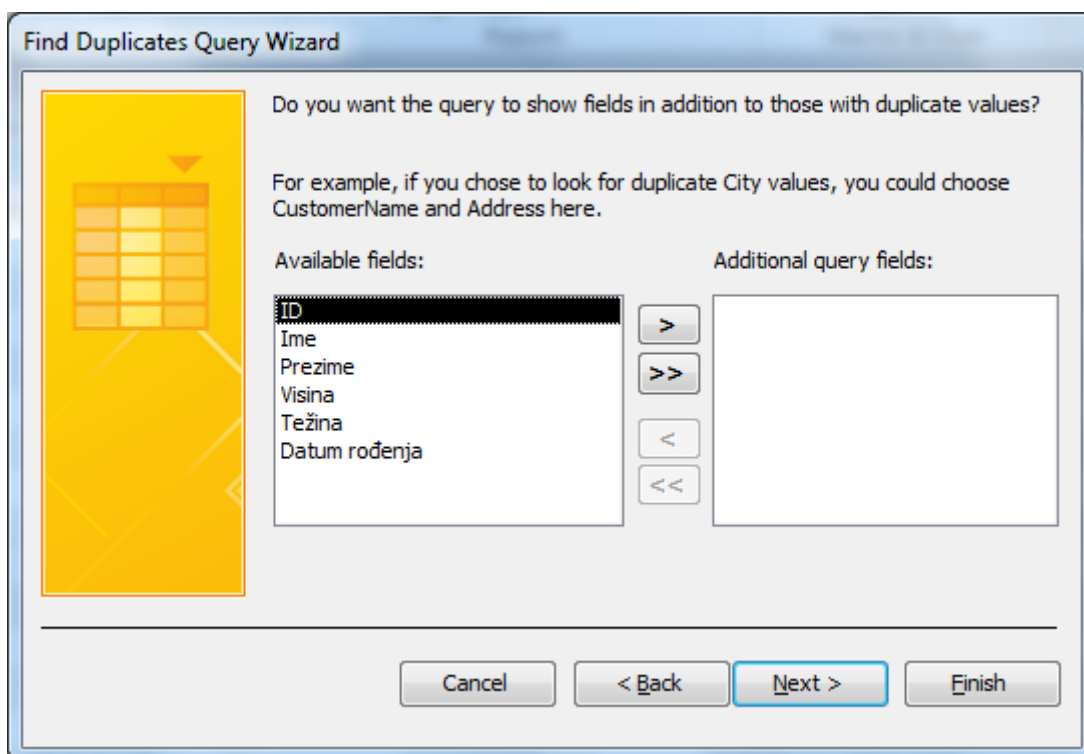
U idućem dijaloškom okviru, odaberimo tablicu „Igrači“.



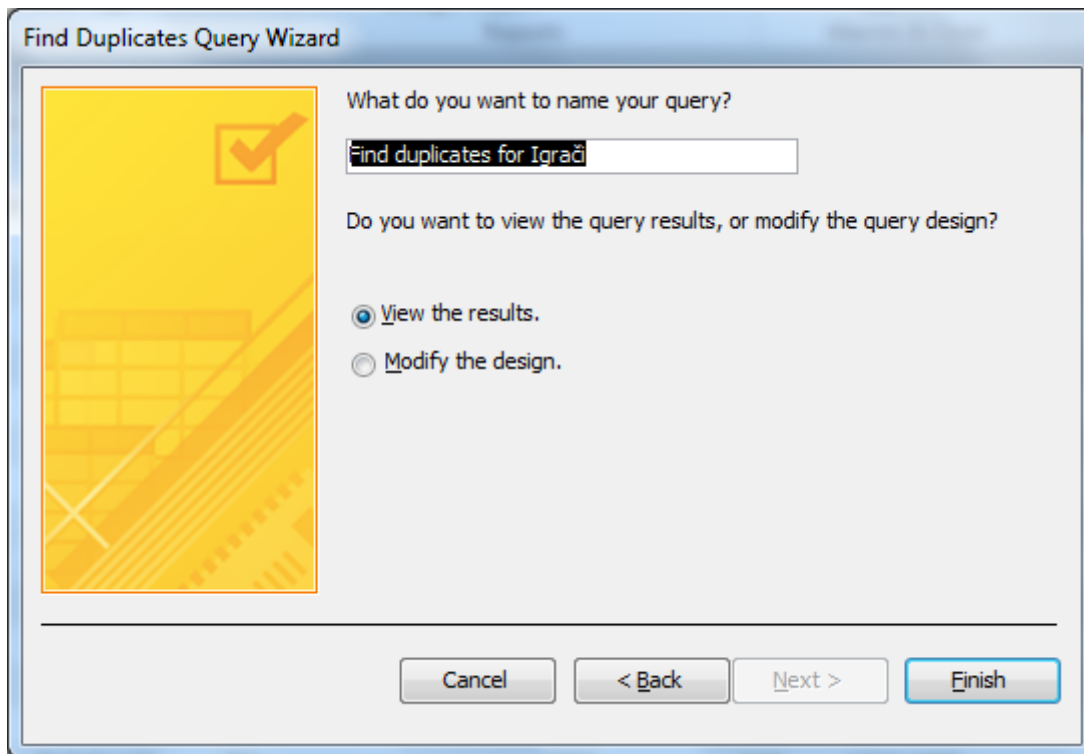
Zatim odabiremo polja koja bi mogla duplikati u tablici „Igrači“. U ovom slučaju, to je polje „Kod kluba“.



Zatim odabiremo dodatna polja koja ćemo prikazati u upitu. Ostavimo ovu opciju praznom.



Odaberimo ime za upit, i prikažimo ga.



Dobili smo broj djelomičnih duplikata, odnosno broj zapisa koji imaju jednaku vrijednost polja „Kod kluba“ u tablici „Igrači“.

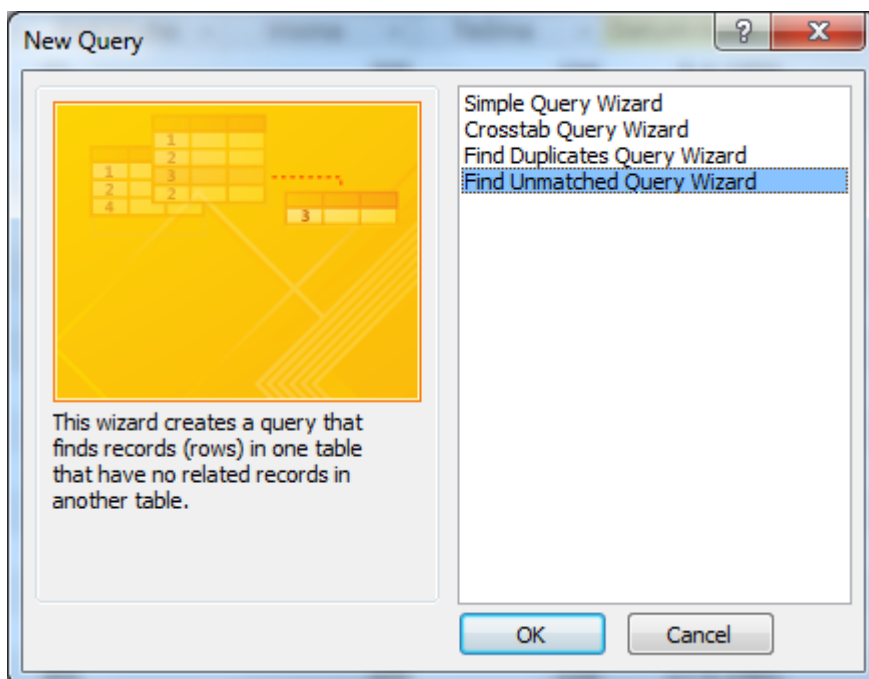
Kod kluba Fi	NumberOfDups
BN	2
CD	2
CI	2
LC	2
LL	2
MT	2
NK	2
RM	2
ZD	2

Kreiranje i pokretanje upita za prikaz neuparivih zapisa u povezanoj tablici

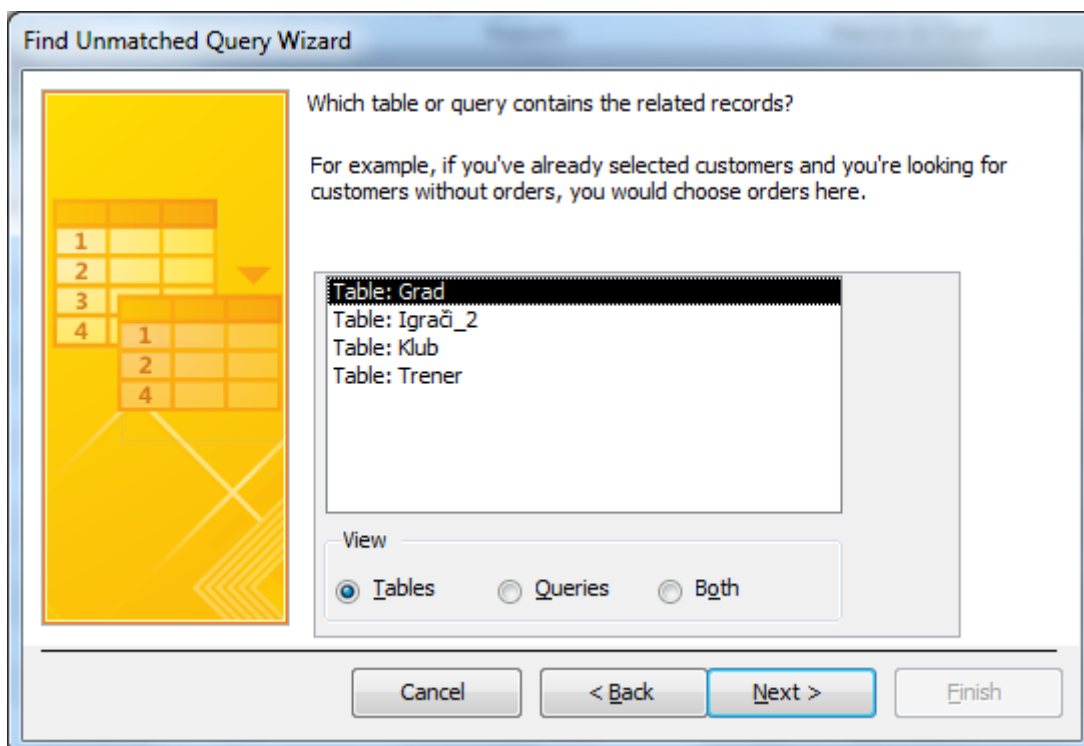
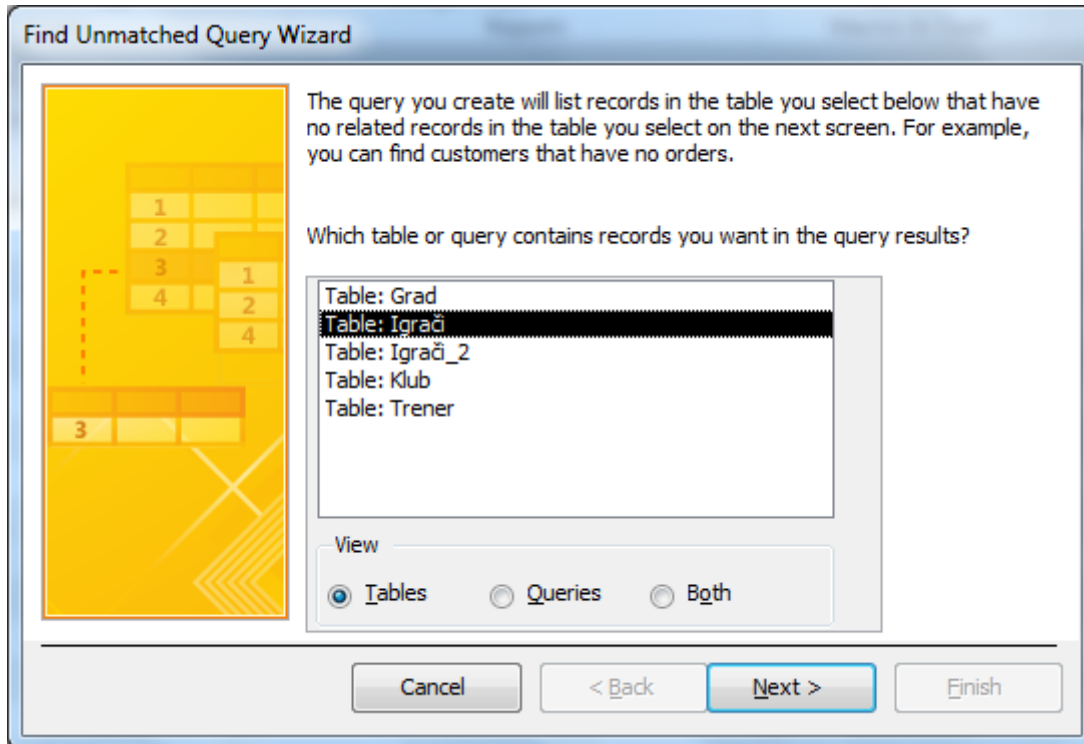
Za potrebe ovog dijela, dodat ćemo dodatni zapis u tablicu „Igrači“.

ID	Ime	Prezime	Kod kluba	Visina	Težina	Datum roђе	Click to Add
1	Dario	Šarić	CI	208	104	8.4.1994.	
2	Darryl Eugene	Strawberry	CI	196	93	15.6.1985.	
3	Jozo	Brkić	ZD	211	101	16.12.1986.	
4	Romeo	Travis	ZD	201	100	10.12.1984.	
5	Predrag	Šuput	CD	200	111	1.6.1977.	
6	Bracey	Wright	CD	191	95	1.7.1984.	
7	Nikola	Mirotić	RM	209	107	11.2.1991.	
8	Dontaye	Draper	RM	180	82	10.8.1984.	
9	Kobe	Bryant	LL	198	93	23.8.1978.	
10	Steve	Nash	LL	191	81	7.2.1974.	
11	Chris	Paul	LC	183	79	6.5.1985.	
12	Blake	Griffin	LC	208	114	16.3.1989.	
13	Tyson	Chandler	NK	216	109	2.10.1982.	
14	Carmelo	Anthony	NK	203	104	29.5.1984.	
15	Deron	Williams	BN	191	95	26.6.1984.	
16	Mirza	Teletović	BN	206	116	17.9.1985.	
17	Nikola	Peković	MT	211	131	3.1.1986.	
18	Ricky	Rubio	MT	193	82	21.10.1990.	
19	Michael	Jordan	CB	198	98	17.2.1963.	
*	(New)						

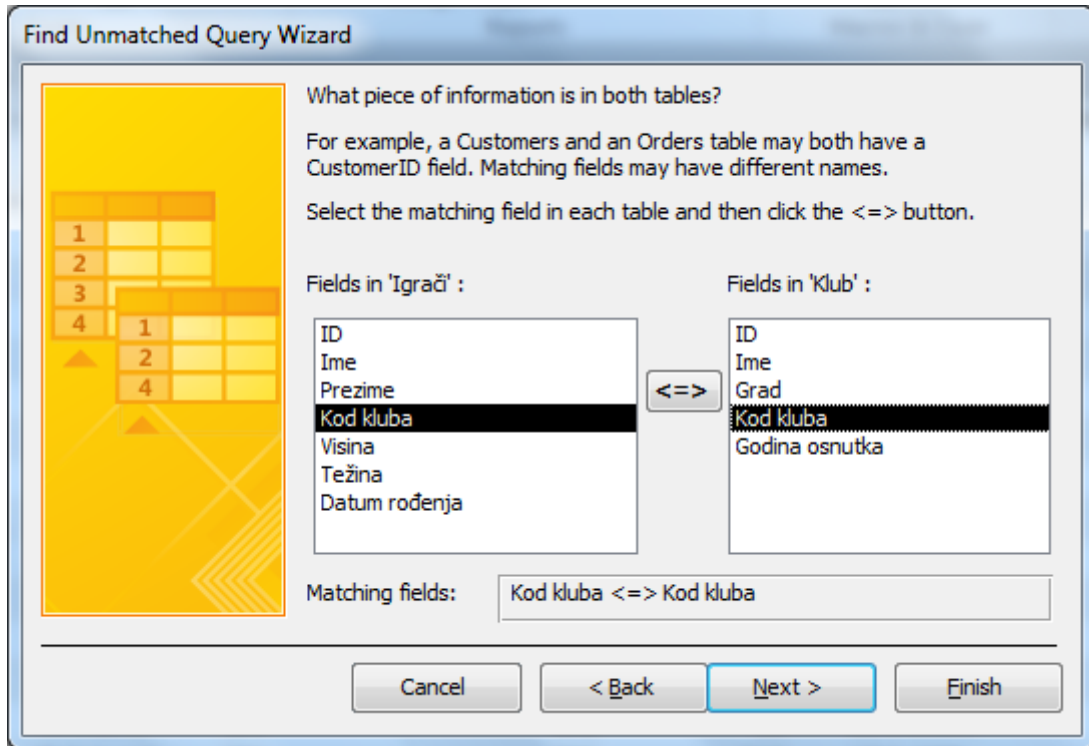
Za kreiranje upita za prikaz neuparivih zapisa u povezanoj tablici, koristit ćemo čarobnjak za kreiranje upita. Za prikaz neuparivih zapisa u povezanoj tablici, odabiremo „Find Unmatched Query Wizard” u dijalogu „New Query”.



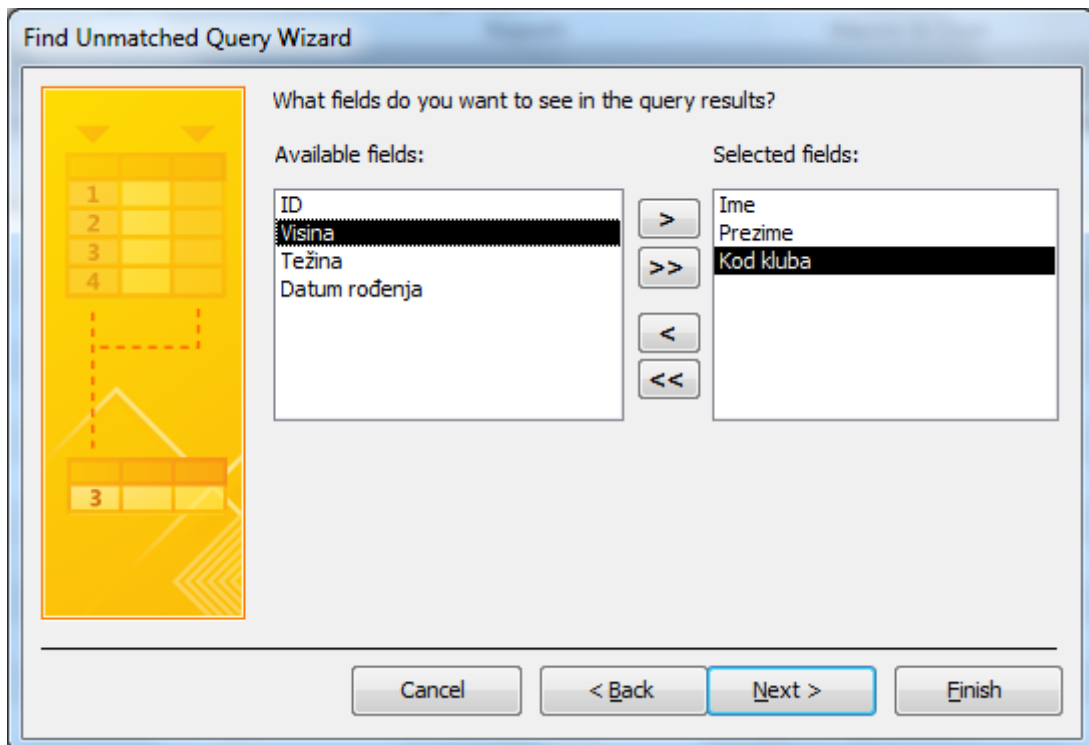
U narednom dijalogu odaberimo tablicu „Igrači”, a zatim povezanu tablicu „Klub”.



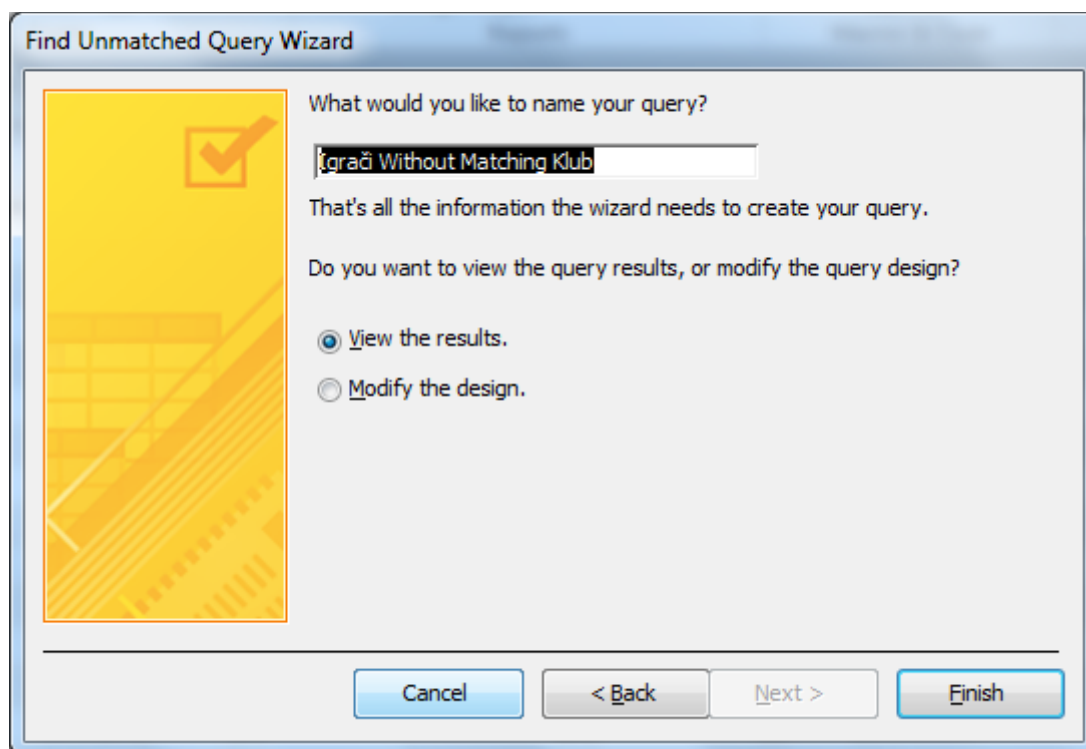
Naredni dijaloški okvir definira koji dio traženi informacije se nalazi u obje tablice. U obje tablice se nalazi polje „Kod kluba“ i to je polje čijim korištenjem želimo pronaći neuparive zapise.



U narednom dijalogu, odabiremo koja ćemo polja dodatno prikazati u upitu. Odaberimo „Ime“, „Prezime“ i „Kod kluba“.



Definirajmo ime upitu i prikazimo ga.



Ime	Prezime	Kod kluba
Michael	Jordan	CB
*		

Možemo vidjeti da je nepovezani zapis onaj koji ima vrijednost polja „Kod kluba” jednaku „CB”, budući da takva vrijednost polja „Kod kluba” u tablici „Klub” ne postoji.

1.7 Dotjerivanje upita

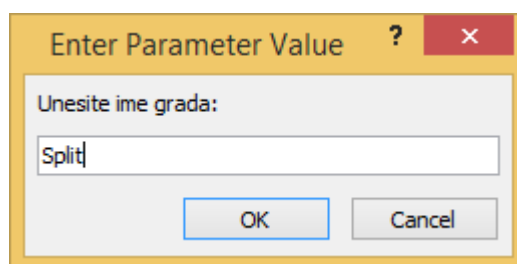
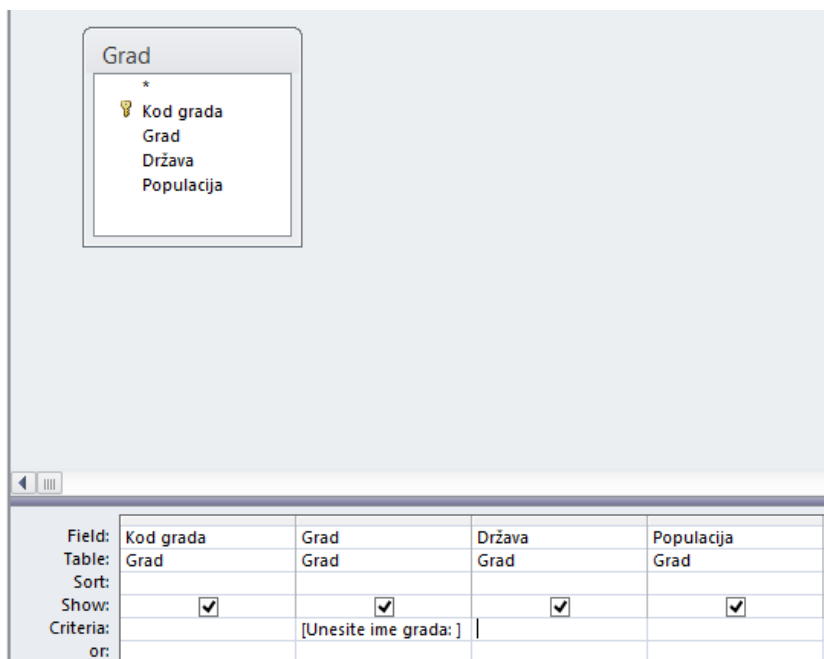
Kreiranje, mijenjanje i pokretanje parametarskog upita s jednom, dvije varijable

Parametar je informacija koja se upisuje u upit prije njegovog samog izvođenja. Mogu se koristiti pojedinačno i u sklopu većeg upita s više parametara. Koriste se kod svih tipova upita.

Za kreiranje parametarskog upita, otvorimo „Query Design” i tablicu „Grad”. Odaberimo polja Kod grada, Grad, Država i Populacija. Unutar opcije Criteria, za polje grad upišimo sljedeći niz znakova „[Unesite ime grada:]”.

Pokrenimo upit klikom na „Run”. Otvara se dijaloški okvir unutar kojeg možemo unijeti ime grada, primjerice „Split”.

Primijetimo da se u dijaloškom okviru rečenica koju smo upisali pojavljuje bez uglatih zagrada.



Dobivamo rezultat upita, a to su podaci o gradu Splitu iz tablice Grad.

Kod grada	Grad	Država	Populacija
ST	Split	Hrvatska	200000
*			

Za dodavanje još jednog kriterija, možemo na postojeći upit dodati kriterij nad poljem Država. Upišimo idući niz znakova [Unesite naziv države:] u opciju Criteria za polje Država.

Pokrenimo upit. U dijaloški okvir koji se pojavio i u prethodnom primjeru ponovno upišimo „Split”. Pored njega, pojavljuje se novi dijaloški okvir u koji upisujemo naziv države. Za provjeru, upišimo naziv države „SAD”.

Grad

- *
 - Kod grada
 - Grad
 - Država
 - Populacija

Field:	Kod grada	Grad	Država	Populacija
Table:	Grad	Grad	Grad	Grad
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:		[Unesite ime grada:]	[Unesite naziv države:]	
or:				

Enter Parameter Value ? x

Unesite naziv države:

OK Cancel

Kao što smo očekivali, nismo dobili niti jedan redak kao rezultat, budući da u tablici grad nema redaka koji imaju vrijednost polja „Grad” jednaku Split, te vrijednost polja „Država” jednaku SAD.

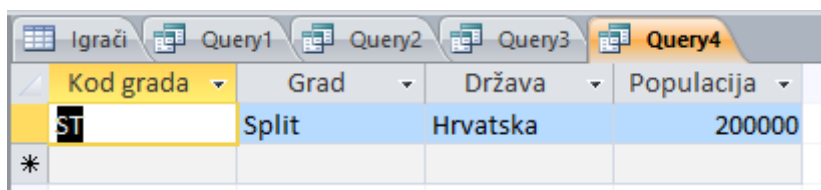
Ponovimo isti upit, ali ovaj put za naziv države upišimo Hrvatska.

Enter Parameter Value ? x

Unesite naziv države:

OK Cancel

Prema očekivanjima, ovaj put dobijamo jedan redak u kojem je vrijednost polja „Grad” jednaka Split, a vrijednost polja „Država” jednaka Hrvatska.



Kod grada	Grad	Država	Populacija
ST	Split	Hrvatska	200000
*			

Korištenje zamjenskih znakova u upitu: [], !, -,

Za lociranje specifičnog retka u slučaju kad se ne možemo sjetiti točne vrijednosti polja, mogu se koristiti zamjenski znakovi u upitu. Pored toga, zamjenski znakovi mogu stajati umjesto nepoznatih znakova u tekstualnoj vrijednosti u upitu i korisni su za lociranje više redaka sa sličnim, ali ne i istim podacima. Zamjenski znakovi mogu pomoći pri lociranju redaka koji se temelje na specifičnom uzorku.

Konkretno, u slučaju prethodnog dijela, imali smo korištenje znakova [] koji služe za zamjenu znakova unutar zagrada. Npr., u slučaju da smo imali 'Z[Unesite ime grada]', kao rezultat upita dobili bi one retke čije polje „Grad” počine sa znakom „Z” te se nastavlja i završava s nizom znakova koje smo unijeli u dijaloškom okviru.

Zamjenski znak ! služi za isključivanje znakova u upitu. Primjerice, Spl![u]t, eliminira niz „Splut”, a dopušta Split, Splet itd.

Zamjenski znak – služi za označavanje raspona znakova (abecedno gledano). Npr., S[p-z]lit će dopustiti sve retke koji imaju vrijednost polja koja ima 5 slova, počinje sa „S”, drugi znak ima u rasponu p – z, i nastavlja (završava se) sa „lit”.

Zamjenski znak # se koristi kod numeričkih vrijednosti polja. Zamjenjuje bilo koju znamenku u broju. Npr. 1#2 dopušta 102, 112, 122, 132 itd.

**Ovaj priručnik je namijenjen za učenje uz materijale
objavljene na sljedećim linkovima:**

Pomoću kvizeva objavljenih na linkovima ispod testirajte poznavanje naprednih baza podataka u programu Microsoft Access 2010 (*pripremila i izradila: Gorana Čelebić*):

<p>Dizajn tablica – Polja, stupci:</p> <p>Primijeniti i promijeniti tipove podataka kao što su text, memo, hyperlink, currency, date & time u polju (stupcu); Razumjeti posljedice promjene tipova podataka; Kreirati, urediti mogućnost odabira (lookup) za polje (stupac); Kreirati, urediti pravilo provjere podataka (validation rule) za polje (stupac); Kreirati, urediti masku za unos (input mask) u polju (stupcu); Primijeniti, izmijeniti zadane vrijednosti za polje (stupac); Postaviti polje (stupac) koje zahtijeva obavezan unos podataka</p> <p>Dizajn tablica – Relacije / Spojevi:</p> <p>Kreirati, izmijeniti veze među tablicama tipa jedan prema jedan (one-to-one) i jedan prema više (one-to-many) Razumjeti i izmijeniti relaciju među tablicama tipa više prema više (many-to-many); Primijeniti unutarnje (inner join), vanjske (outer join) i vlastite spojeve (self-join); Primijeniti referentnu cjelovitost i automatsko brisanje povezanih slogova; Povezati / spojiti podatke prilikom kreiranja upita</p>	<p>pokreni kviz</p>
<p>Dizajn upita (Queries) – Tipovi upita:</p> <p>Kreirati i upotrijebiti upit za ažuriranje podataka u tablici; Kreirati i upotrijebiti upit za brisanje slogova iz tablice; Kreirati i upotrijebiti upit za pospremanje odabranih podataka u obliku tablice; Dodati slogove u tablicu pomoću upita Grupirati informacije u upitu; Upotrijebiti funkcije u upitu: sum, count, average, max, min Upotrijebiti crosstab query; Upotrijebiti zamjenske znakove u upitu; Upotrijebiti aritmetička i logičke izraze u upitu</p> <p>Dizajn upita (Queries) – Dotjerivanje upita:</p> <p>Prikazati duplikate; Prikazati slogove koji nemaju povezane podatke (unmatched values);</p>	<p>pokreni kviz</p>

<p>Prikazati niz najvećih ili najmanjih vrijednosti u upitu; Dozvoliti unos proizvoljnih parametara u upit putem okvira za unos podataka (parameter query); Dotjerivanje upita korištenjem vrijednosti Null i NOT</p>	
<p>Dizajn obrazaca (Forms) – Kontrole: Kreirati vezane (bound) i slobodne (unbound) kontrole; Kreirati i urediti kontrole combo box, list box, check box i grupe mogućnosti (option group) Kreirati kontrole sa aritmetičkim ili logičkim izrazima na obrascu; Umetnuti podatkovno polje koje će se pojavljivati u zaglavlju obrasca, podnožju prve stranice ili svih stranica Dizajn obrazaca (Forms) – Podobrasci: Kreirati podobrazac i povezati ga sa glavnim obrascem; Izmijeniti podobrazac za promjenu prikazanih slogova</p> <p>Dizajn izvješća – Računanje: Kreirati kontrole za aritmetičko ili logičko računanje na izvješću; Kreirati kontrolu za izračun postotka na izvješću Upotrijebiti formule i izraze na izvješću, npr. sum, count, average, max, min, concatenate; Kreirati tekuće zbrojeve na izvješću</p> <p>Dizajn izvješća – Presentacija: Umetnuti podatkovno polje koje će se pojaviti u zaglavlju izvješća, podnožju prve stranice ili svih stranica; Umetnuti prijelome stranica za grupe podataka na izvješću</p> <p>Makronaredbe – Snimanje i dodjela: Snimiti jednostavnu makronaredbu, npr. zatvaranje obrasca; Pokrenuti makronaredbu Dodijeliti / priložiti makronaredbu obrascu, izvješću, kontroli</p> <p>Uvoz, izvoz i povezivanje podataka – Upravljanje podacima: U bazu podataka uvesti datoteku tipa tekst, tablična kalkulacija, csv, dBASE, Paradox Izvesti podatke u obliku tablične kalkulacije, txt, dBASE, Paradox; Povezati vanjske podatke sa bazom podataka</p>	<p>pokreni kviz</p>

OPĆI UVJETI KORIŠTENJA

Web–stranicu www.ITdesk.info je pokrenula udruga „Otvoreno društvo za razmjenu ideja - ODRAZI“ u sklopu aktivnog promoviranja ljudskog prava na slobodan pristup informacijama te ljudskog prava na edukaciju.

Slobodno kopirajte i distribuirajte ovaj dokument, uz uvjet da ne mijenjate ništa u njemu!

Nad svim programima i uslugama navedenim na web–stranici ITdesk Home na web adresi ITdesk.info isključivo pravo posjeduju njihovi autori/ce. Microsoft, Windows, i Windowsxx su registrirani zaštitni znakovi tvrtke Microsoft Corporation. Ostali zaštitni znaci korišteni na ITdesk Home web–stranicama su isključivo vlasništvo njihovih vlasnika/ca. Ako imate pitanja vezana uz uporabu ili redistribuciju bilo kojeg programa, molimo kontaktirajte autore/ice dotičnog programa. Sva dodatna pitanja pošaljite na info@itdesk.info.

Ove web–stranice sadržavaju linkove na ostale web–stranice ili izvore. ITdesk.info tim NIJE odgovoran za tekstualni i/ili reklamni sadržaj, odnosno za proizvode koji su na tim web–stranicama /izvorima ponuđeni, kao što NIJE odgovoran niti za sadržaj koji je putem njih dostupan; mogućnost korištenja ili točnost sadržaja. Linkove koristite na vlastitu odgovornost. Također, ITdesk.info tim ne garantira:

- da je sadržaj na ovim web–stranicama oslobođen od pogrešaka ili pogodan za svaku svrhu,
- da će ove web–stranice ili web usluge funkcionirati bez pogrešaka ili prekida,
- da će biti odgovarajući za vaše potrebe,
- da implementacija takvog sadržaja neće narušavati patente, autorska prava, zaštitni znak ili ostala prava neke treće strane.

Ako se ne slažete s ovim općim uvjetima korištenja ili ako niste zadovoljni web–stranicama koje pružamo, prekinite s korištenjem ovih web–stranica i web usluga. ITdesk.info tim nije odgovoran vama, niti trećim osobama za bilo koju nastalu štetu, bila ona direktna, indirektna, slučajna ili posljedična, povezana s ili proizlazeći iz vaše uporabe, pogrešne uporabe ovih web–stranica ili web usluga. Iako vaše potraživanje može biti bazirano na garanciji, ugovoru, prekršaju ili nekom drugom pravnom uporištu, neovisno o našoj obaviještenosti o mogućnosti nastanka takve štete, oslobađamo se svake odgovornosti. Prihvatanje ograničenja naše odgovornosti nužan je preduvjet korištenja ovih web–stranica i web usluga

Svi softveri navedeni u ovom ili drugim dokumentima objavljenim na stranici ITdesk.info su navedeni samo za edukativne svrhe ili kao primjer te mi, na bilo koji način, ne preferiramo navedeni softver u odnosu na neki drugi softver. Bilo koju izjavu da neki navedeni softver preferiramo više u odnosu na drugi, koji se spominje ili ne spominje u materijalima, smatrat će se kao lažni iskaz. Našu izravnu i bezrezervnu podršku imaju jedino softveri otvorenog koda (*open source*) koji omogućuju korisnicima/cama da bez prepreka postanu digitalno pismeni, koriste računalo i sudjeluju u modernom informatičkom društvu.

ITdesk.info

– projekt računalne e-edukacije sa
slobodnim pristupom

Nakladnik:
Otvoreno društvo za razmjenu ideja
(ODRAZI), Zagreb