

**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET ORGANIZACIJE I INFORMATIKE**  
**VARAŽDIN**

**Tina Mehanović**

**ALATI ZA ERA MODELIRANJE**  
**ZAVRŠNI RAD**

**Varaždin, 2014.**

**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET ORGANIZACIJE I INFORMATIKE**  
**VARAŽDIN**

**Tina Mehanović**

**Redoviti student**

**Broj indeksa: 39072/10-R**

**Smjer: Informacijski sustavi**

**Preddiplomski studij**

**ALATI ZA ERA MODELIRANJE**

**ZAVRŠNI RAD**

**Mentor:**

Izv. prof. dr. sc. Kornelije Rabuzin

**Varaždin, rujan 2014.**

# Sadržaj

<b>1. UVOD</b> .....	<b>1</b>
<b>2. KONCEPTUALNO MODELIRANJE</b> .....	<b>2</b>
2.1 DEFINICIJA I TEMELJNI POJMOVI .....	2
2.2 VAŽNOST I ZNAČENJE KONCEPTUALNOG MODELA .....	2
<b>3. ERA MODEL</b> .....	<b>4</b>
3.1 OSNOVNE KOMPONENTE .....	4
3.1.1 Entiteti .....	4
3.1.2 Atributi .....	5
3.1.3 Veze .....	5
3.2 NOTACIJE .....	7
3.2.1 Chen notacija .....	7
3.2.2 Crow's foot (IE notacija) .....	8
3.2.3 Barkerova notacija .....	9
3.2.4 IDEF1X .....	9
3.2.5 UML notacija .....	11
3.3 PRIMJER BAZE PODATAKA .....	11
<b>4. ALATI</b> .....	<b>15</b>
4.1 NAVICAT .....	15
4.1.1 Općenito o alatu .....	15
4.1.2 Spajanje na MySQL server .....	16
4.1.3 Konceptualni model .....	17
4.1.3.1 Entiteti i atributi .....	18
4.1.3.2 Veze, kardinalnost i notacije .....	20
4.2. DBFORGE STUDIO .....	22
4.2.1 Općenito o alatu .....	22
4.2.2 Spajanje na MySQL server .....	23
4.2.3 Konceptualni model .....	24
4.2.3.1 Entiteti i atributi .....	24
4.2.3.2 Veze, kardinalnost i notacije .....	26
4.3 DBSCHEMA .....	27
4.3.1 Općenito o alatu .....	27
4.3.2 Spajanje na MySQL server .....	28
4.3.3. Konceptualni model .....	29
4.3.3.1 Entiteti i atributi .....	30
4.3.3.2 Veze, kardinalnost i notacije .....	31
4.4. TOAD .....	32
4.4.1 Općenito o alatu .....	32
4.4.2 Spajanje na MySQL server .....	33
4.4.3. Konceptualni model .....	33
4.4.3.1 Entiteti i atributi .....	34
4.4.3.2 Veze, kardinalnost i notacije .....	35
4.5 USPOREDBA ALATA .....	36
<b>5. ZAKLJUČAK</b> .....	<b>38</b>
<b>6. LITERATURA</b> .....	<b>39</b>

# 1. Uvod

Svjedoci smo vrlo brzog razvoja informacijske tehnologije. Tim razvojem sustavi koji se izrađuju sve su složeniji a zahtjevi su sve veći. Svaki informacijski sustav koji se implementira iza sebe ima bazu podataka koja mu pruža potrebne podatke za rad. Izrada baze podataka započinje analizom i prikupljanjem potrebnih podataka. Sljedeći korak je izrada konceptualnog modela. Nakon toga se konceptualni model pretvara u relacijski, što je faza logičkog modeliranja. Nakon izrade modela dolazi do fizičke implementacije baze podataka. Tema ovog završnog rada odnosi se na drugi korak dizajniranja baze podataka a to je konceptualno modeliranje.

Od 1960-ih, kada je datotečni sustav zamjenjen bazama podataka počinje razvoj konceptualnog modeliranja. Kako je značaj i potreba za dobrim konceptualnim modelima u stalnom porastu, počeli su se razvijati alati sa grafičkim korisničkim sučeljem koji uvelike olakšavaju i ubrzavaju proces izrade modela i baze podataka.

Rad je podijeljen na tri glavne cjeline koje se nadovezuju jedna na drugu. U prvom dijelu rada opisat ću osnovne pojmove te ulogu, značenje i važnost konceptualnog modeliranja. Navesti ću definiciju koncepta i modela te iz tih definicija definirati konceptualno modeliranje.

U drugom dijelu rada opisat ću ERA modeliranje kao najpoznatiju tehniku za izradu konceptualnih modela. Navesti ću komponente od kojih se sastoji svaki ERA model te ih opisati. Nakon toga opisat ću primjer koji ću implementirati dalje u radu uz pomoć alata za konceptualno modeliranje baze podataka.

Treći dio rada obuhvatit će 4 alata s grafičkim korisničkim sučeljem koji olakšavaju i ubrzavaju proces konceptualnog modeliranja. Alati koje ću detaljnije opisati su Navicat, dbForge, DbSchema i Toad. Za svaki alat posebno ću prikazati način rada u njemu te ih na kraju usporediti.

## 2. Konceptualno modeliranje

### 2.1 Definicija i temeljni pojmovi

Prije same definicije konceptualnog modeliranja, objasniti ću pojmove od kojih je sastavljen naslov ove teme, a to su koncept i model. Značenje riječi koncept preuzela sam iz online riječnika<sup>1</sup> gdje je koncept definiran kao ideja o nečemu, formirana mentalnim kombiniranjem svih njegovih karakteristika. To je značenje u našim mislima koje dajemo objektima iz stvarnog svijeta. Model je približni opis nekih pojava ili objekata u stvarnom svijetu uz pomoć simbolike [Anić i Goldstein, 2000, str. 873.]. Rezultat konceptualnog modeliranja je model koji prikazuje promatrani manji podskup realnog svijeta. Simboli koji se koriste za prikaz konceptualnog modela su najčešće kvadrati koji označavaju entitete i crte kao simbol veze između entiteta.

Konceptualno ili semantičko modeliranje je modeliranje načina na koji ljudi razumijevaju i procesiraju realnost. [Maleković Mirko, prezentacija sa predavanja: Konceptualno oblikovanje baze podataka, 2013, 2. slajd]. Ova definicija je na jednostavan način opisala konceptualno modeliranje. Može se reći da je to preslikavanje objekata i veza među objektima na model koristeći dogovorene simbole. U fazi konceptualnog modeliranja bitno je usredotočiti se na to koji su podaci potrebni za buduću bazu podataka i kako su oni međusobno povezani. Konceptualni model je prikaz objekata iz stvarnog svijeta na način na koji ih vidi dizajner tog modela.

### 2.2 Važnost i značenje konceptualnog modela

Konceptualno modeliranje je temelj za izgradnju dobre baze podataka. Kako bi naglasili važnost konceptualnog modeliranja autori knjige *Data Modeling Essentials* [2] usporedili su modeliranje baze podataka sa izradom nacrtu za izgradnju kuće i naveli nekoliko razloga zbog kojih bi konceptualnom modeliranju trebalo posvetiti najviše vremena. Isto kao što gradnja kuće bez nacrtu i planiranja dovodi do problema, postoji mogućnost da se zbog odabira lošeg materijala nešto sruši i slično, tako i baza podataka bez prijašnje izrade modela kad tad dođe do situacije da zbog nepotpunosti podataka ili loše povezanosti istih izradu cjelog sustava vraća na početak. To

---

<sup>1</sup> Rječnik: <http://www.thefreedictionary.com/concept>

<sup>2</sup> Simsion, G., Witt, G. (2005). *Data Modeling Essentials*. Dostupno na <http://uk-corp.org/Books/Other-Technical/Data%20Modeling%20Essentials%202005.pdf>, str. 8-10

nikako nije poželjno te zato mnogi autori naglašavaju važnost konceptualnog modeliranja. Kao ključni razlog zbog kojeg je konceptualno modeliranje najvažniji korak u dizajniranju baze podataka navedena je 'moć' koju ima mala promjena modela na sustav u cjelini. Iako je dizajniranje baze podataka mali dio ukupnog informacijskog sustava on puno utječe na složenost aplikacijskog rješenja. Dobro osmišljeni konceptualni model pojednostavljuje proces programiranja što rezultira značajnom uštedom u ukupnim troškovima. Dobra strana konceptualnog modela je njegova 'sažetost', u smislu da je sa jednim grafičkim modelom prikazana slika podataka cjelog budućeg ili postojećeg informacijskog sustava. Taj grafički prikaz se lakše i brže proučava nego kad bi iste te specifikacije bile opisane na nekoliko stranica teksta.

U knjizi *Data Modeling Essentials* [3] navodi se 9 osobina dobrog konceptualnog modela. Ja sam kao najbitnije izdvojila to da model treba biti kompletan, neredundantan, treba dobro provoditi poslovna pravila i treba biti jasan i razumljiv dizajnerima kao i programerima i krajnjim korisnicima. Kompletnost modela znači da su u modelu svi podaci koji će biti potrebni za implementaciju sustava. Model je nerundantan kada nesprega podatke o istom objektu više puta. Za model je bitno da dobro provodi poslovna pravila. Na primjer, u bazi podataka za prodaju kino ulaznica, poslovno pravilo glasi da jedna ulaznica vrijedi samo za jednu projekciju. Model dobro provodi to poslovno pravilo tako da se u tablici *ulaznice* nalazi atribut *projekcija*. Time je osigurano da se nepažnjom korisnika ili programera ne može dogoditi da jedna ulaznica sadržava više od jedne projekcije.

Prvi korak u dizajniranju svake baze je prikupljanje podataka i određivanje veza među podacima koji su potrebni. Prikupljeni podaci preslikavaju se u konceptualne modele korištenjem tehnika kao što su ERA ili UML modeliranje. U nastavku rada reći ću nešto o ERA modeliranju, tehnici koja se najčešće koristi za modeliranje baze podataka.

---

3 Simsion, G., Witt, G. (2005). *Data Modeling Essentials*. Dostupno na <http://uk-corp.org/Books/Other-Technical/Data%20Modeling%20Essentials%202005.pdf>, str. 10-15

## 3. ERA model

### 3.1 Osnovne komponente

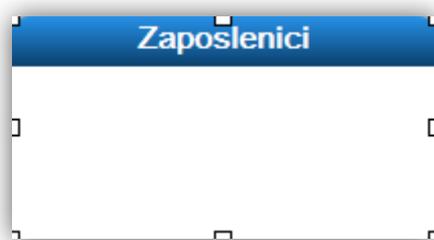
ERA je skraćena od eng. Entity Relationship Attribute, što u prijevodu znači entiteti, veze i atributi. Upravo to su osnovne komponente koje sačinjavaju svaki ERA model. Veze povezuju entitete, a entiteti su opisani atributima.

#### 3.1.1 Entiteti

Za entitet se može reći da je skup objekata iz stvarnog svijeta o kojemu se spremaju podaci u bazu. Entitet sadržava objekte kao što su živa bića (npr. osobe, studenti, životinje...), zemljopisna odredišta (npr. gradovi, države...) ili bilo koje druge objekte s kojima se susrećemo i potrebni su za bazu podataka (npr. vozila, zgrade, knjige...).

Razlikujemo slabe i jake entitete. Za razliku od jakih, slabi entiteti ovise o jakom entitetu kako bi se mogli identificirati. Slabi entiteti nastaju zbog toga što nije moguće na drugačiji način implementirati vezu više-više. Primarni ključ slabih entiteta sastavljen je od vanjskih ključeva koji sudjeluju u vezi više-više. Primjer slabog entiteta je žanr filmova, jednoznačno je određen dvokomponentnim primarnim ključem koji se sastoji od žanra i filma. Žanr filmova ovisi o jakim entitetima žanr i film.

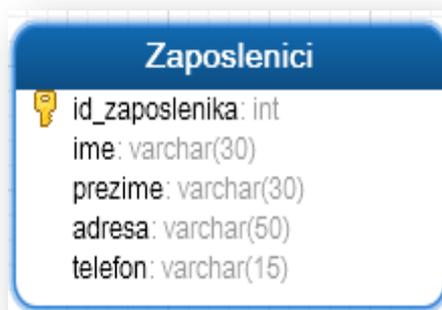
Svaka instanca entiteta mora se moći jednoznačno identificirati. U ERA modelu, entiteti se prikazuju simbolom kvadrata kao što se može vidjeti na slici **Slika 3.1**.



**Slika 3.1 Simbol entiteta u ERA modelu**

### 3.1.2 Atributi

Atributi su podaci kojima želimo opisati entitete. Svaki entitet ima pripadajući skup atributa kojima želimo opisati isti. Razlikujemo identifikacijske i opisne attribute. Identifikacijski atributi jednoznačno određuju svaku instancu entiteta, to su primarni ključevi. Opisni atributi služe za detaljniji opis instance entiteta. Opisni i identifikacijski atributi mogu biti sastavljeni od jednog ili više atributa. Na slici **Slika 3.2** prikazan je način na koji se atributi prikazuju na ERA modelu. Primjer identifikacijskog atributa je id\_zaposlenika a svi ostali atributi su opisni.



**Slika 3.2 Entitet i atributi**

### 3.1.3 Veze

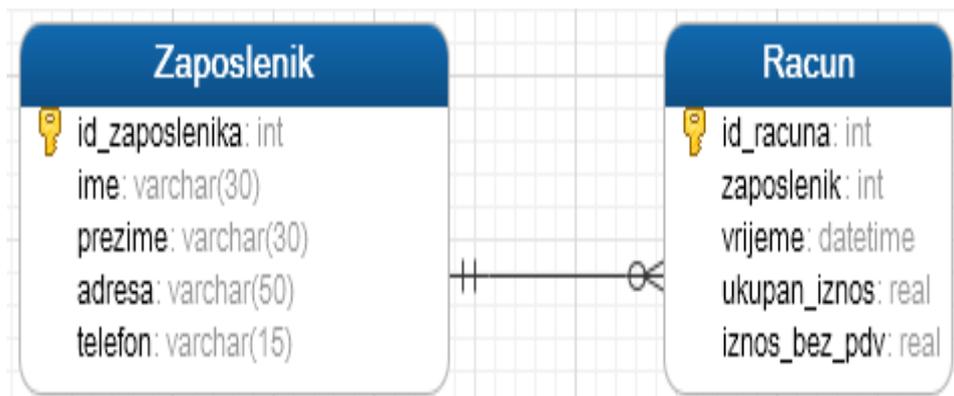
Veze povezuju entitete i obično su opisane glagolom. Dije se prema stupnju i kardinalnosti. Stupanj veze određuje se brojem entiteta koji sudjeluju u njoj, a kardinalitetom se određuje broj instanci jednog entiteta koji može biti povezan sa jednom instancom drugog entiteta. Prema stupnju se razlikuju unarne, binarne, ternarne i n-arne veze. Kardinalnost može biti jedan-jedan, jedan-više i više-više.

Primjer na slici **Slika 3.3** prikazuje unarnu vezu kardinalnosti jedan-jedan. Unarna veza koristi se kada je entitet povezan sa samim sobom. Na primjeru na slici ispod, osoba je u braku sa osobom, pa se entitet osoba referencira na samog sebe. Kardinalnost na slici ispod je jedan-jedan što znači da osoba može biti u braku sa najviše jednom osobom (ravna crtica na krajevima veze). Kružić na vezi označava opcionalnost, što znači da osoba može, a i ne mora biti u braku sa nekom osobom.



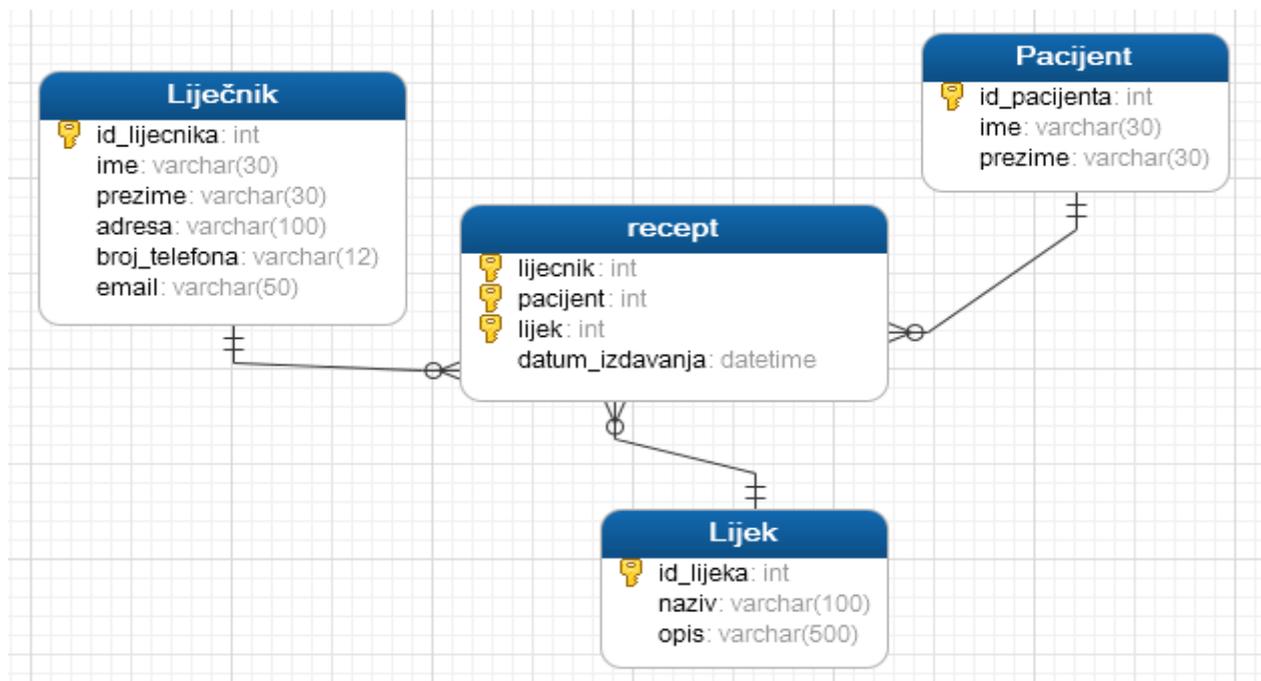
**Slika 3.3 Unarna veza, kardinalnost jedan-jedan**

**Slika 3.4** prikazuje binarnu vezu sa jedan-više kardinalitetom. To je binarna veza jer u njoj sudjeluju dva entiteta. Kardinalnost jedan više se označava ravnom crticom na strani 'jedan' i znakom više na strani 'više'. Primjer prikazuje binarnu vezu u kojoj zaposlenik može izdati nula ili više (opcionalnost) računa, a račun obavezno mora biti izdan od strane jednog zaposlenika.



**Slika 3.4 Binarna veza, kardinalnost jedan-više**

**Slika 3.5** prikazuje ternarnu vezu koja je implementirana dodavanjem nove tablice i tri binarne veze. U primjeru na slici u ternarnoj vezi sudjeluju entiteti liječnik, pacijent i lijek. Ternarna veza se (isto kao i veza više-više) prikazuje posebnom tablicom u kojoj se nalaze primarni ključevi svih entiteta koji sudjeluju u vezi. Liječnik može izdati nula ili više recepata, pacijentu može biti izdano nula ili više recepata i jedan lijek može biti prepisan na nula ili više recepata. Recept obavezno mora imati jednog liječnika, jednog pacijenta i jedan lijek.



**Slika 3.5 Ternarna veza realizirana kao 3 binarne veze**

Veze, odnosno kardinalnost na vezama se može prikazivati različitim notacijama. Na primjerima iznad korištena je 'Crow's foot' notacija. U nastavku rada opisati ću neke najčešće korištene notacije.

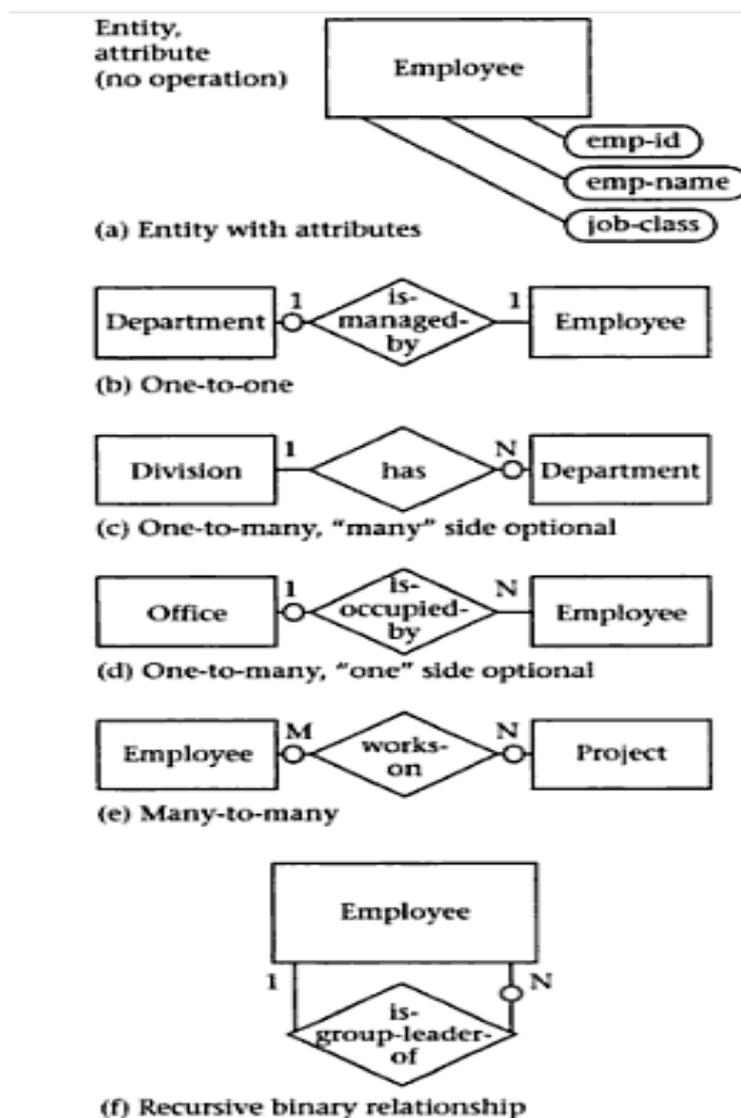
## 3.2 Notacije

### 3.2.1 Chen notacija

Chen notacija dobila je naziv po Peter Chenu koji je 1976. objavio rad u kojemu opisuje ERA model. U ovoj notaciji veze su prikazane simbolom romba koji se nalazi na liniji koja spaja dva entiteta. Atributi su zaokruženi i spojeni linijom sa entitetom kojemu pripadaju. Ova notacija se često koristi u akademskim radovima dok u alatima za konceptualno modeliranje nije baš često viđena.

**Slika 3.6** prikazuje grafički Chen notaciju. Na vrhu slike prikazan je entitet zaposlenik (eng. employee) sa svojim atributima emp\_id (id zaposlenika), emp\_name (ime zaposlenika) i job\_class (klasa posla). Slika (b) prikazuje vezu jedan-jedan. Naziv veze napisan je u rombu na liniji a kardinalnost na krajevima veze. Kružić označava opcionalnost. Na (c) dijelu slike prikazana je jedan-više veza između dva entiteta, strana 'više' označava se slovom N i u ovom primjeru je opcionalna. D dio slike prikazuje vezu jedan-više, isto kao i prethodni primjer samo što je ovdje opcionalnost na strani 'jedan'. E dio prikazuje vezu više-više, jedna i druga strana

veze su opcionalne a označavaju se sa slovima M i N. Na kraju, zadnji dio slike (f) prikazuje rekurzivnu vezu, gdje je entitet zaposlenik spojen sa samim sobom.



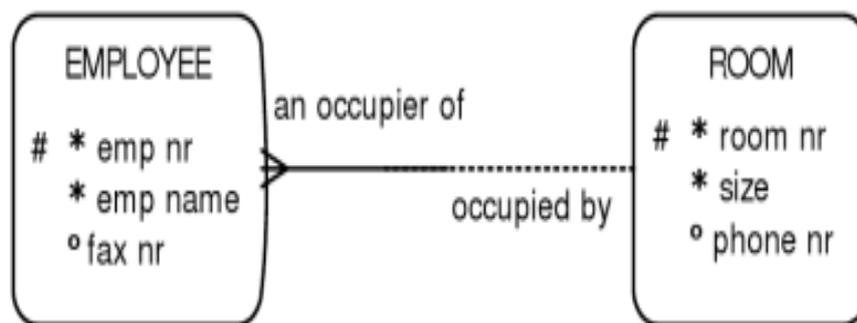
Slika 3.6 Chen notacija [Teorey, 1999, str. 23]

### 3.2.2 Crow's foot (IE notacija)

IE (Information Engineering) notaciju osmislio je australac Clive Finkelstein. To je najpoznatija notacija koja se koristi u većini grafičkih alata za konceptualno modeliranje. Još se zove i Crow's foot (u prijevodu – vranino stopalo) zbog prikaza kardinalnosti 'više' koja podsjeća na vranino stopalo. Ova notacija može se vidjeti na prethodnim slikama (Slika 3.3, Slika 3.4 i Slika 3.5). Za prikaz kardinalnosti 'jedan' koristi se ravna crtica | a za 'više' simbol < ili >. Opcionalnost se označava kružićem.

### 3.2.3 Barkerova notacija

Naziv ove metode dolazi od Richarda Barkera koji ju je objavio 90-ih godina prošlog stoljeća. U ovoj notaciji entiteti se prikazuju u kvadratima sa zaobljenim vrhovima. Atributi mogu biti navedeni ispod naziva entiteta, atribut sa ispunjenim kružićem ili zvjezdicom (\*) je obavezan, sa neispunjenim kružićem neobavezan, a oznaka # se koristi za primarne ključeve. **Slika 3.7** prikazuje Barkerovu notaciju na primjeru binarne veze između entiteta zaposlenik i entiteta soba. Veza između entiteta podijeljena je na dva dijela. Crtkana linija znači da je kardinalnost opcionalna i čita se kao 'nula ili više' tj. može, ali i ne mora biti. Puna linija znači obavezno i čita se kao 'jedan ili više' tj. najmanje jedan. Veza na kojoj se nalazi znak za više kao iz Crow's foot notacije čita se kao 'jedan ili više' ili 'nula ili više' ovisno o tome ako je linija puna ili iscrtkana, veza na kojoj se ne nalazi taj znak čita se kao 'nula ili jedan' ili kao 'jedan i samo jedan' također u ovisnosti o iscrtkanoj ili punoj liniji. Iz slike **Slika 3.7** možemo isčitati da zaposleniku pripada nula ili jedna soba, a soba pripada jednom ili više zaposlenika. To je veza 'jedan-više'.

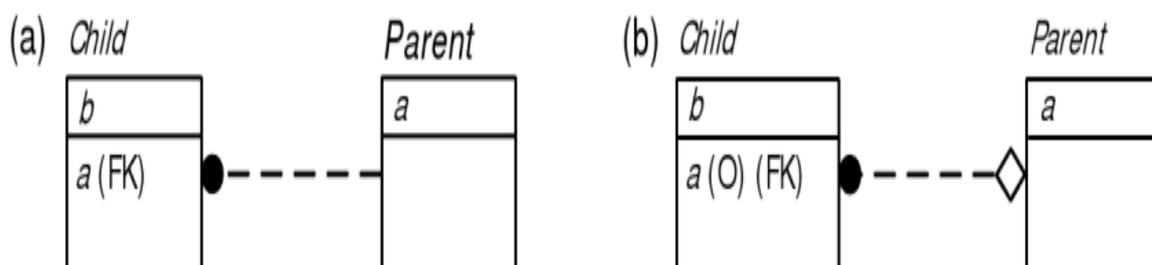


**Slika 3.7 Barkerova notacija [Halpin i Morgan, 2008, str. 309]**

### 3.2.4 IDEF1X

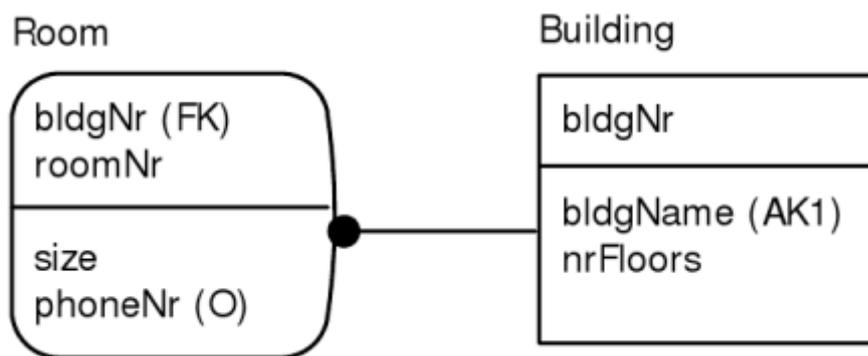
U IDEF1X notaciji entiteti se prikazuju kvadratima koji mogu imati zaobljene ili nezaobljene vrhove. Entiteti sa zaobljenim vrhovima ovise o nekom drugom entitetu dok entiteti koji nemaju zaobljene vrhove se mogu jednoznačno identificirati bez postojanja nekog drugog entiteta. Naziv entiteta napisan je izvan kvadrata. Veze između entiteta mogu biti zavisne i nezavisne. Zavisne veze prikazane su na slici **Slika 3.8** a nezavisne veze se označavaju na isti način jedina razlika je u tome što nezavisne veze imaju iscrtkane linije. Zavisne veze koriste se kada je primarni ključ

entiteta složen od primarnog ključa nekog drugog entiteta i označavaju se punim linijama. Nezavisne veze koriste se za spajanje entiteta čiji primarni ključ neovisi o primarnom ključu drugog entiteta. Nepostojanje oznake na vezi čita se kao 'jedan' (isključivo jedan, ni više ni manje), simbol romba se čita kao 'nula ili jedan', ispunjeni kružić se čita kao 'nula ili više' a ispunjeni kružić sa slovom 'p' se čita kao 'jedan ili više'.



**Slika 3.8 IDEF1X nezavisne veze [Halpin i Morgan, 2008, str. 329]**

Na slici **Slika 3.8** prikazane su nezavisne veze u IDEF1X notaciji. Na prvom djelu slike (a) je veza jedan-više između tablice 'roditelj' i tablice 'dijete', 'roditelj' može imati nula ili više 'djece' a 'dijete' pripada jednom i samo jednom 'roditelju'. Drugi dio slike isto prikazuje vezu jedan-više, razlika je u opcinalnosti na strani 'jedan'.



**Slika 3.9 IDEF1X zavisna veza [Halpin i Morgan, 2008, str. 327]**

Na slici **Slika 3.9** prikazan je primjer zavisne veze u IDEF1X notaciji. Primjer prikazuje vezu jedan-više, u zgradi se nalazi nula ili više soba i soba je u jednoj zgradi. Entitet soba prikazan je u kvadratu sa zaobljenim vrhovima jer ovisi o postojanju entiteta zgrade.

### 3.2.5 UML notacija

UML skraćenica je od eng. Unified Modeling Language (u prijevodu jedinstveni jezik modeliranja) a koristi se za modeliranje klasa objekata tj. entiteti su klase objekata. Entiteti se označavaju kvadratima u kojima piše naziv entiteta i njegovi atributi. Navicat alat, koji će biti opisan kasnije, nudi mogućnost izrade modela koristeći UML notaciju. UML notacija za prikaz ERA modela vrlo je slična IE notaciji. Razlika je u zapisu kardinalnosti a mogući zapisi su sljedeći:

1 – jedan i samo jedan

1...\* – jedan ili više

0...1 – nula ili jedan

0...\* – nula ili više.

### 3.3 Primjer baze podataka

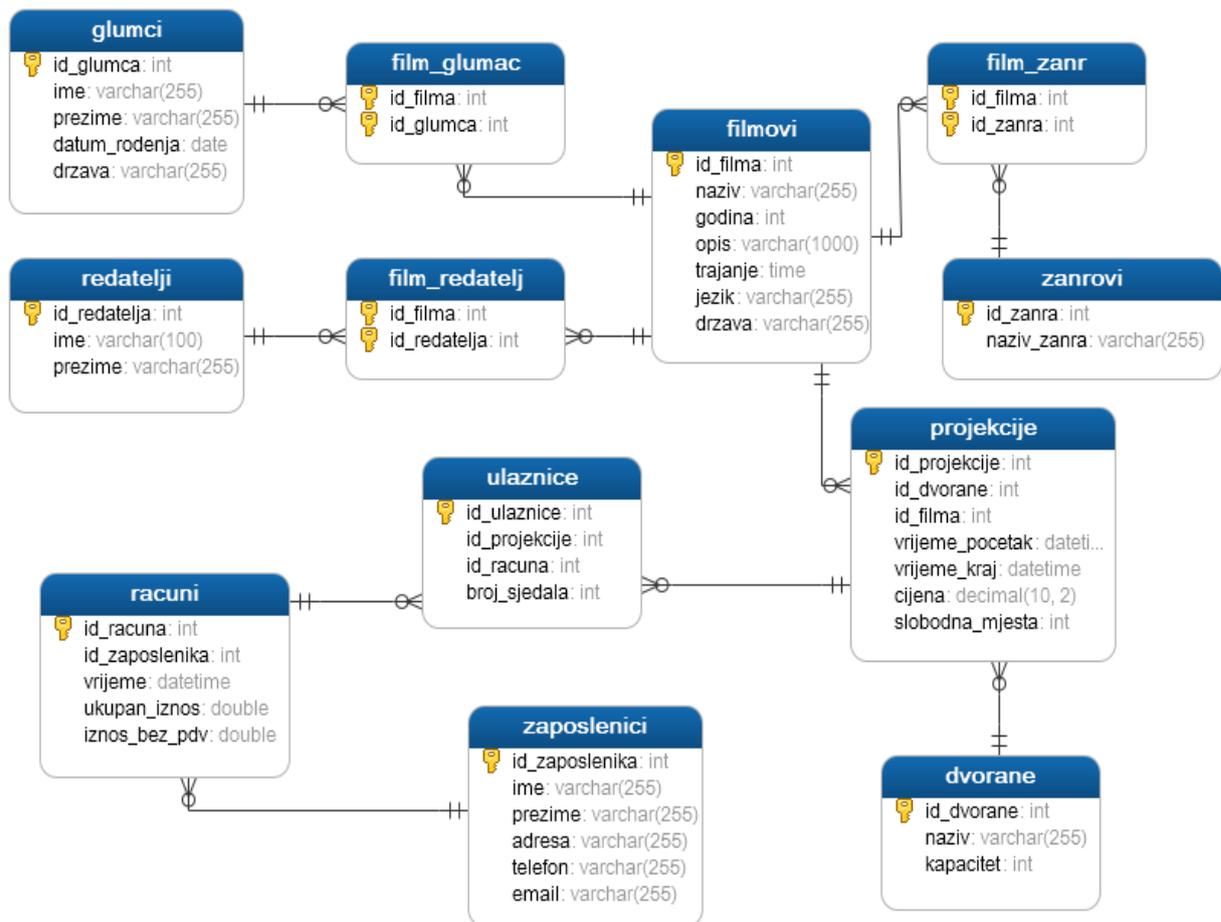
U ovom radu izradit ću model i implementirat ću bazu podataka koja bi služila aplikaciji za prodaju kino ulaznica. Za implementaciju spomenute baze bit će potrebno pohranjivati podatke o filmovima, glumcima koji u njima glume, redateljima i žanrovima. Također je potrebno pohranjivati podatke o dvoranama i projekcijama koje će se prikazivati. Entitet projekcija sadržavat će podatke o filmu, dvorani i vremenu u kojemu će se pojedina projekcija prikazivati. Pošto bi aplikacija trebala služiti za prodavanje kino ulaznica, još je potrebno izraditi entitet zaposlenik čija će se oznaka nalaziti na računu. Dakle, u modelu će se još nalaziti entitet račun i entitet ulaznica. U tablici **Tablica 3.1** nalazi se popis svih entiteta, njihovih atributa i opis istih. Primarni ključevi podcrtani su ravnom linijom a vanjski su podebljani žutom bojom.

**Tablica 3.1 Popis i opis svih entiteta i njihovih atributa**

Entiteti	Atributi	Opis
Filmovi	<u>Id_filma</u> Naziv Godina Trajanje Opis Jezik	Entitet film pohranjuje osnovne podatke o filmovima kao što su naziv, godina kada je snimljen film, vremensko trajanje, opis tj. kratak sadržaj, jezik i država.

	Drzava	
Glumci	<u>Id_glumca</u> Ime Prezime Datum_rodenja Drzava	Entitet glumac pohranjuje osnovne podatke o glumcima koji imaju glavne uloge u filmovima koji se prikazuju u kinu.
Film_glumac	<u>Id_filma</u> <u>Id_glumca</u>	Ovo je relacija koja je nastala zbog veze više-više između entiteta film i glumac. Sastoji se od dva atributa koji su vanjski ključevi na entitet film i glumac. Ta dva atributa tvore dvokomponentni primarni ključ.
Redatelji	<u>Id_redatelja</u> Ime Prezime	Entitet redatelj pohranjuje osnovne podatke o redateljima filmova koji se prikazuju u kinu.
Film_redatelj	<u>Id_filma</u> <u>Id_redatelja</u>	Ovo je relacija koja je nastala zbog veze više-više između entiteta film i redatelj. Sastoji se od dva atributa koji su vanjski ključevi na entitet film i redatelj. Ta dva atributa tvore dvokomponentni primarni ključ.
Zanrovi	<u>Id_zanra</u> Naziv_zanra	Entitet žanr pohranjuje nazive filmskih žanrova.
Film_zanr	<u>Id_filma</u> <u>Id_zanra</u>	Ovo je relacija koja je nastala zbog veze više-više između entiteta film i žanr. Sastoji se od dva atributa koji su vanjski

		ključevi na entitet film i žanr. Ta dva atributa tvore dvokomponentni primarni ključ.
Dvorane	<u>Id_dvorane</u> Naziv Kapacitet	U entitet dvorana sprema se naziv dvorane i kapacitet, kako bi se znalo koliko ulaznica se može prodati za pojedinu dvoranu.
Projekcije	<u>Id_projekcije</u> <u>Id_dvorane</u> <u>Id_filma</u> Vrijeme_pocetak Vrijeme_kraj Slobodna_mjesta Cijena	Ovaj entitet pohranjuje podatke o vremenu, mjestu, cijeni i preostalim slobodnim ulaznicama za pojedinu projekciju.
Zaposlenici	<u>Id_zaposlenika</u> Ime Prezime Adresa Telefon Email	U entitet zaposlenik pohranjuju se osnovni podaci o zaposlenicima koji su potrebni da bi se vidjelo tko je i u koje vrijeme izdao neki račun.
Racun	<u>Id_racuna</u> <u>Id_zaposlenika</u> Vrijeme Ukupan_iznos Iznos_bez_pdv	Entitet račun pohranjuje podatke o izdanim računima. Vrijeme kada je izdan račun, tko ga je izdao i koji je bio iznos na računu.
Ulaznice	<u>Id_ulaznice</u> <u>Id_projekcije</u> <u>Id_racuna</u> Broj_sjedala	Entitet sprema podatke o projekciji i broju sjedala. Povezan je i sa entitetom račun, kako bi se mogao izračunati ukupan iznos.



**Slika 3.10 ERA model baze za prodaju kino ulaznica**

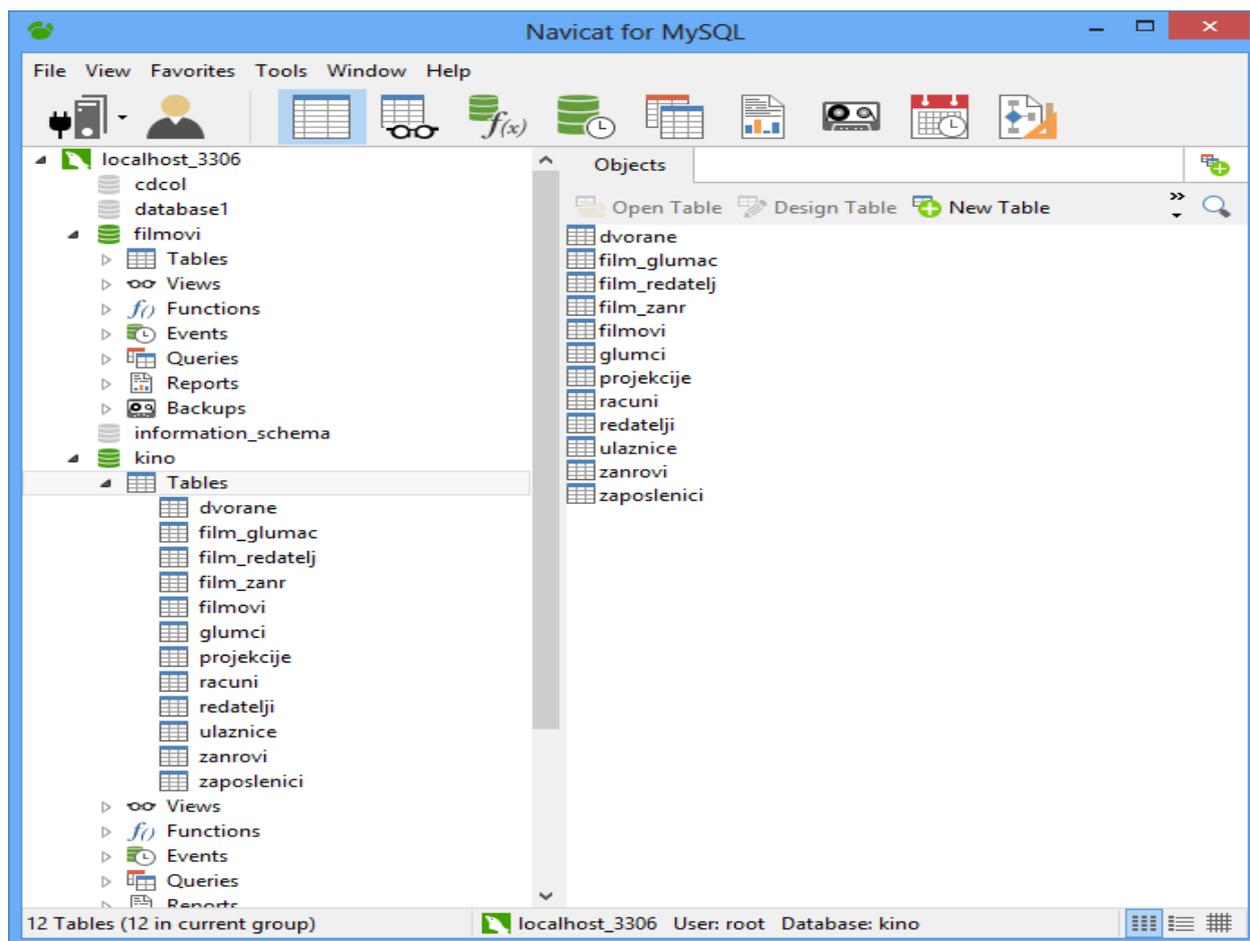
Na slici **Slika 3.10** prikazan je ERA model ranije opisanog primjera. Model je izrađen u Navicat alatu. Korištena notacija za prikaz entiteta i veza je Crow's foot notacija. U nastavku rada bit će opisan alat u kojemu je izrađen ovaj model.

## 4. Alati

### 4.1 Navicat

#### 4.1.1 Općenito o alatu

Navicat je alat s grafičkim korisničkim sučeljem koji služi za kreiranje i administriranje baza podataka. Može se spajati na MySQL, Oracle, PostgreSQL, SQLite, SQL Server ili MariaDB baze podataka. U ovom radu prikazat ću Navicat za MySQL bazu podataka. Alat je dostupan na Microsoft Windows, Mac OS X te Linux platformi. Za korištenje alata potrebno je instalirati MySQL server te Navicat for MySQL, koji se može preuzeti sa sljedeće stranice <http://www.navicat.com/download>. Na prethodno navedenoj stranici ponuđene su probne verzije koje omogućavaju besplatno korištenje alata 30 dana. Ukoliko želimo i dalje koristiti Navicat potrebno ga je kupiti. U ponudi su 3 izdanja: Enterprise (199 \$ za Windows platformu), Standard (129 \$ za Windows platformu) i nekomercijalno izdanje (99 \$ za Windows platformu) koje služi za edukativne svrhe.

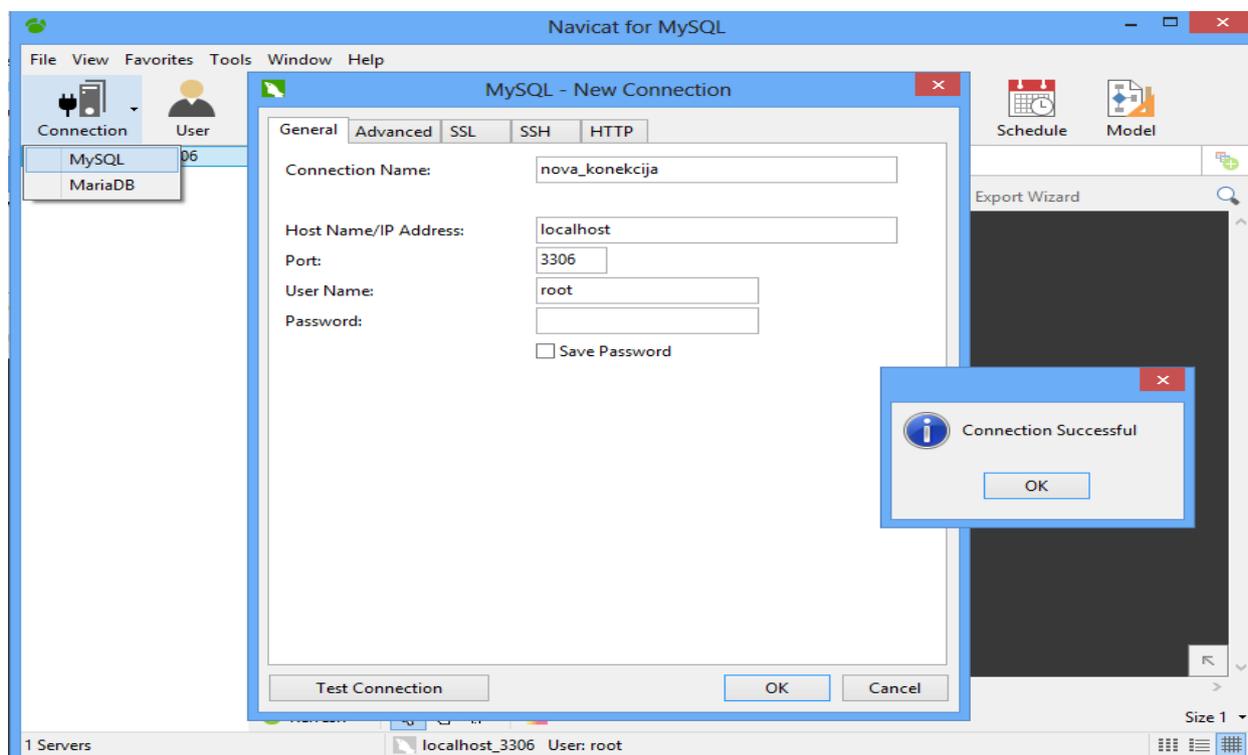


Slika 4.1 Početni zaslon Navicat alata

Na slici **Slika 4.1** prikazan je početni zaslon Navicat alata. U glavnoj alatnoj traci nalaze se osnovni objekti i funkcije za rad sa bazom podataka, a to su konekcija, tablice, upiti, funkcije, događaji, izvještaji, modeli i dr. Na lijevoj strani nalazi se popis svih konekcija, baza podataka i pripadajućih objekata. Otvorena kartica u kojoj se nalazi popis tablica je glavni radni dio u kojemu se otvaraju kartice s obzirom na odabranu funkcionalnost iz izbornika.

#### 4.1.2 Spajanje na MySQL server

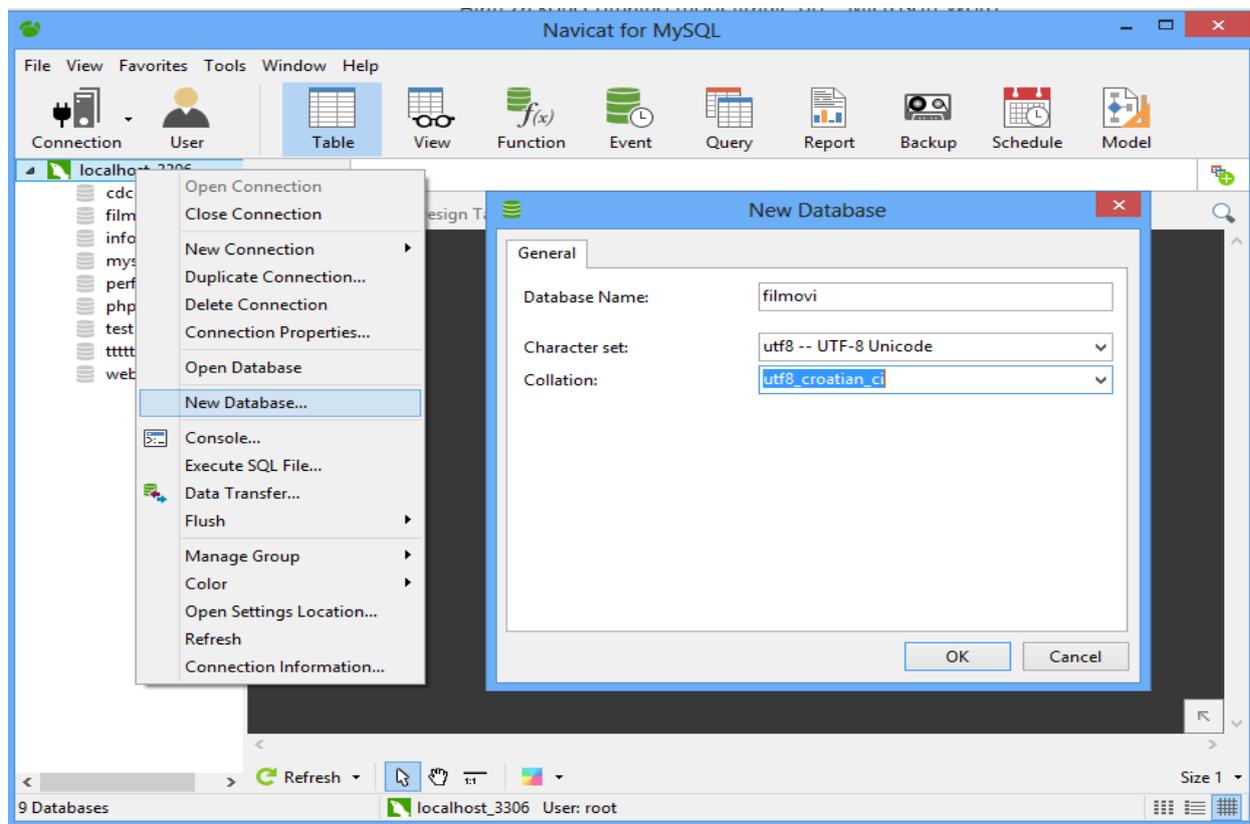
Za kreiranje nove baze ili administriranje postojeće, potrebno je povezati se sa MySQL serverom. Moguće je povezivanje sa udaljenim računalom ili lokalno. U primjeru koji ću prikazati spajanje je lokalno.



**Slika 4.2 Kreiranje nove konekcije - Navicat**

Na slici **Slika 4.2** prikazani su koraci za izradu nove konekcije. Iz alatne trake odaberemo *Connection*, zatim iz padajućeg izbornika MySQL. Otvori se novi prozor u kojemu se treba ispuniti naziv konekcije (*Connection Name*), ime računala na kojemu je instaliran MySQL (*Host Name*) u ovom primjeru je to *localhost*, port 3306 jer je to standardni port za MySQL te korisničko ime obično je *root* jer se spaja na lokalno računalo ali može biti i neko drugo ime. Kada su ispunjena sva potrebna polja klik na *Test Connection* za prikaz poruke da li je konekcija uspješno ili neuspješno kreirana. Sve uspješno kreirane konekcije nalaze se na popisu sa lijeve strane početnog zaslona.

Nakon uspješnog povezivanja sa MySQL serverom nastavljam sa kreiranjem baze podataka. Najprije se treba otvoriti konekcija. Odaberemo željenu konekciju sa lijeve strane početnog zaslona, desni klik na naziv konekcije i iz izbornika odabrati *Open Connection*.



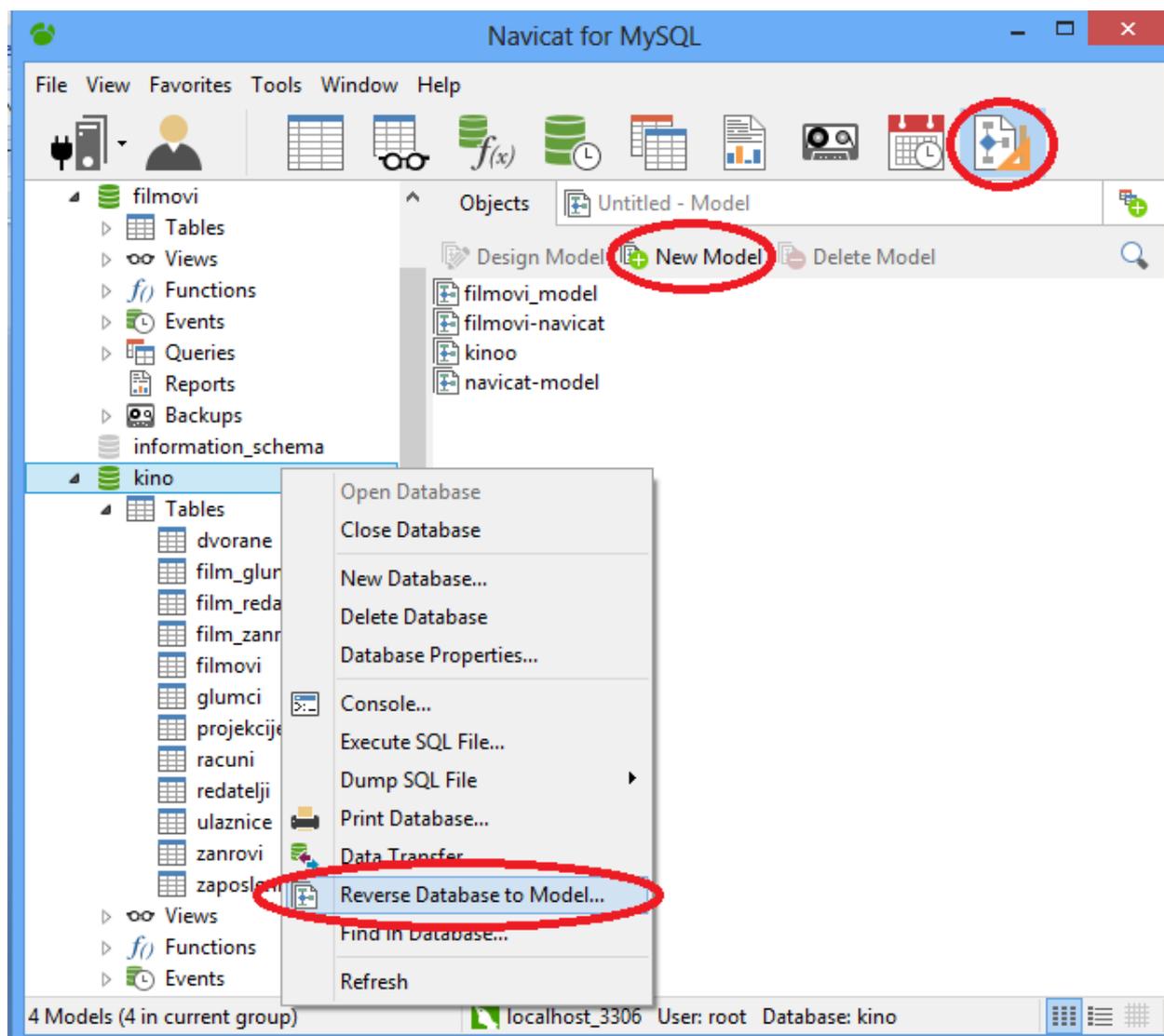
**Slika 4.3 Kreiranje nove baze – Navicat**

**Slika 4.3** prikazuje izradu nove baze podataka. Za izradu nove baze potrebno je desnim klikom na konekciju odabrati *New Database* te zatim ispuniti sva polja (ime baze, skup znakova i skup pravila za uspoređivanje znakova) u novootvorenom prozoru. Klikom na gumb *OK* nova baza je kreirana i možemo započeti sa kreiranjem tablica koje su opisane u prethodnom poglavlju.

#### 4.1.3 Konceptualni model

U Navicat alatu novi konceptualni model kreira se odabirom gumba *Model* iz alatne trake. Kada kliknemo na gumb *Model* otvori se kartica koja se može vidjeti na slici **Slika 4.4**. U otvorenoj kartici nalazi se popis već kreiranih modela, te ukoliko želimo uređivati neki od njih, odaberemo ih dvostrukim klikom na naziv. U suprotnom, ako želimo izradivati potpuno novi konceptualni model odaberemo opciju *New Model* koja je ponuđena u novootvorenoj kartici. Ovaj alat nudi mogućnost kreiranja modela iz već postojeće baze (eng. reverse engineering). Ukoliko želimo tako kreirati model, desnim klikom na naziv baze s lijeve strane otvara se izbornik iz kojega je

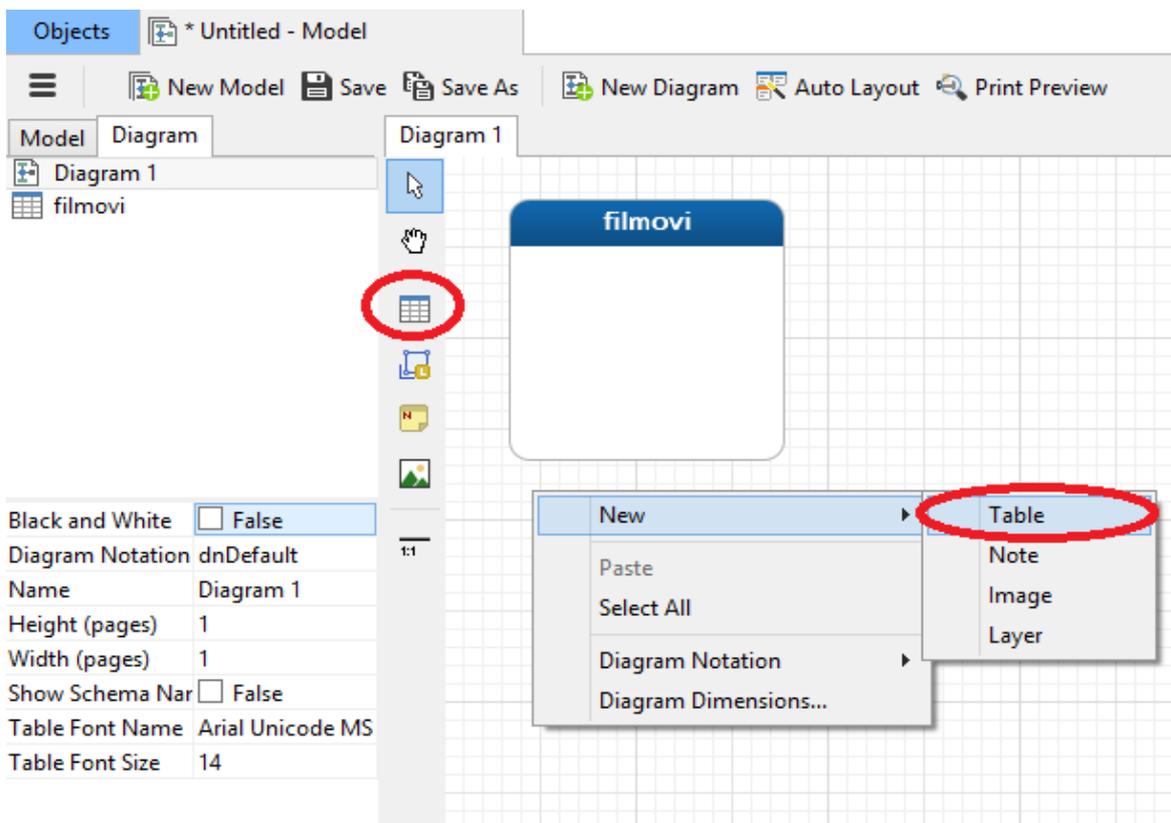
potrebno odabrati opciju *Reverse Database to Model*. U nastavku rada prikazat ću izradu novog konceptualnog modela (odabir *New Model*) i dodavanje objekata na isti.



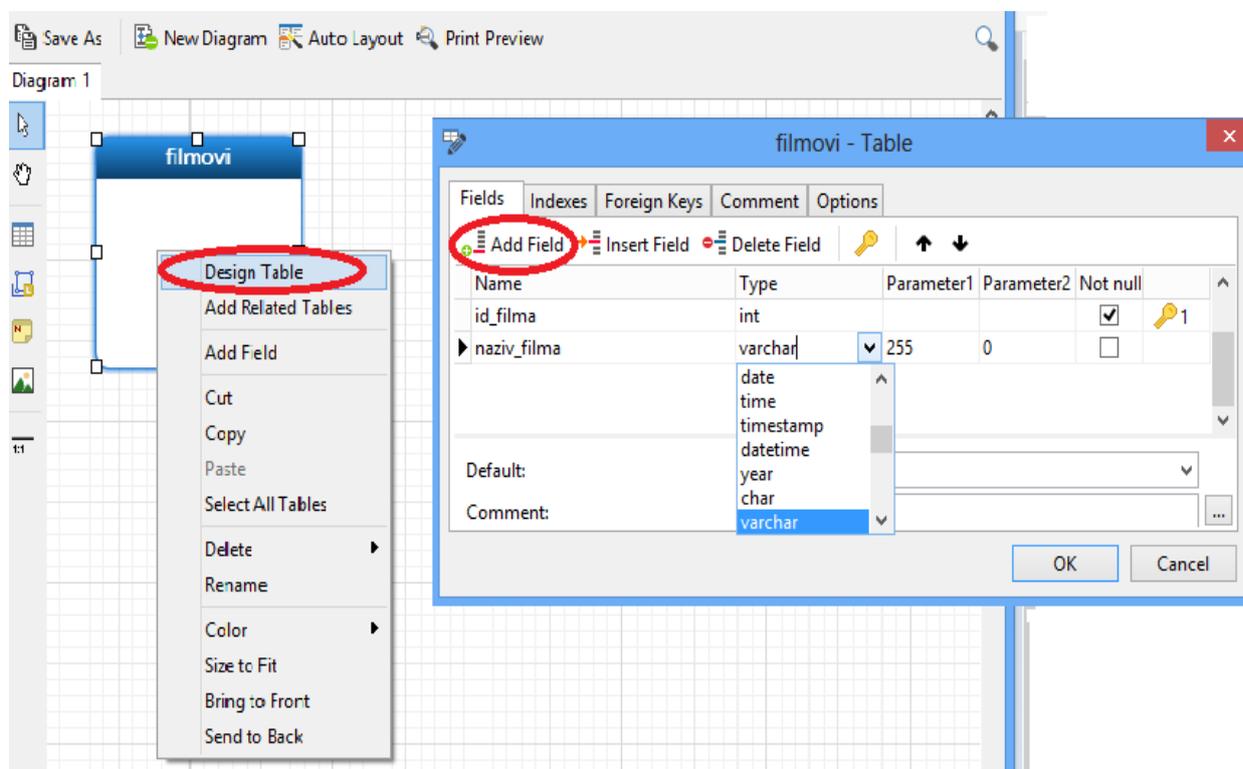
**Slika 4.4 Kreiranje modela u Navicat alatu**

#### **4.1.3.1 Entiteti i atributi**

Kada se otvori radna površina novog modela možemo započeti sa dodavanjem objekata na konceptualni model. Kreiranje novog entiteta prikazano je na slici **Slika 4.5**. Novi entitet može se dodati na dijagram odabirom simbola tablice iz okomitog izbornika pokraj radne površine ili desnim klikom na površinu i odabrati opciju *New Table*. Bez obzira koji način odabrali, na radnoj površini stvorit će se kvadratić u koji direktno upisujemo naziv entiteta koji želimo. Kada je entitet kreiran, možemo dodavati attribute kojima ga želimo identificirati i opisati. Atributi se entitetu dodaju desnim klikom na željeni entitet i odabrati opciju *Design Table*. Dodavanje atributa može se vidjeti na slici **Slika 4.6**.



Slika 4.5 Novi entitet u Navicat alatu

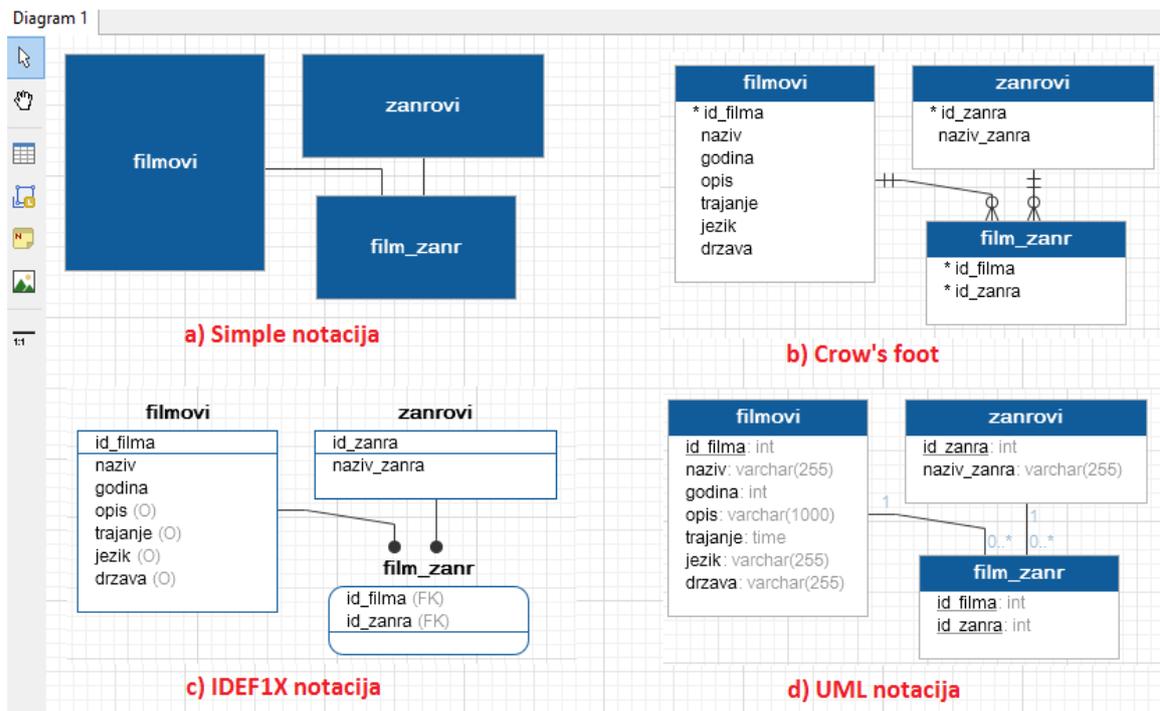


Slika 4.6 Dodavanje atributa u Navicat alatu

Kada se odabere *Design Table* za željeni entitet, otvori se poseban prozorčić za uređivanje. Novi atribut dodaje se odabirom kartice *Fields* te opcijom *Add Field*. U stupac *Name* potrebno je napisati naziv atributa, stupac *Type* služi za odabir tipa podataka koji će se moći unositi kao vrijednost tog atributa. Ostali parametri nisu obavezni za ispunjavanje. U stupac *Parameter* upisuje se broj dozvoljenih znakova za određeni atribut. *Not Null* možemo označiti ukoliko neki atribut obavezno mora biti ispunjen. U zadnjem stupcu nalazi se simbol ključa ukoliko je neki atribut primarni ključ. Odabirom kartice *Indexes* (pokraj *Fields*) možemo dodavati unique ograničenje tako da se ispuni naziv, odaberu se atributi koji trebaju biti jedinstveni a u stupcu *Index Type* odabere se *unique*. Odabirom kartice *Foreign Keys* mogu se dodavati vanjski ključevi ukoliko oni postoje za taj entitet. Kod kreiranja vanjskog ključa potrebno je ispuniti naziv, odabrati atribut koji će biti vanjski ključ te odabrati tablicu i primarni ključ te tablice na koji će se referencirati.

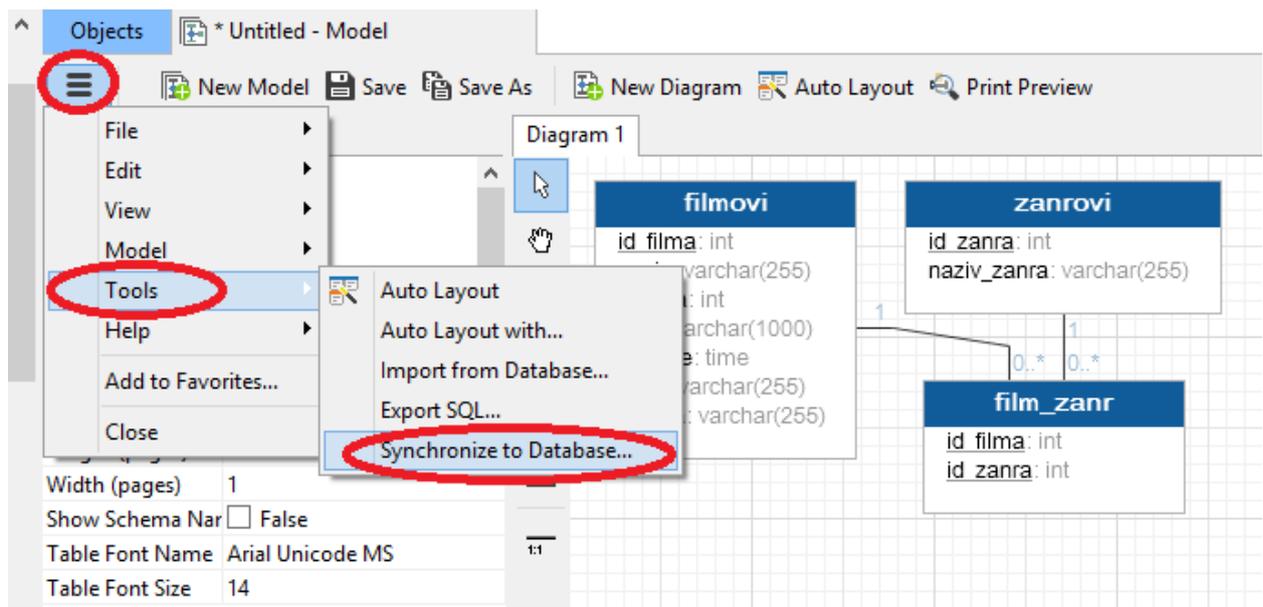
#### **4.1.3.2 Veze, kardinalnost i notacije**

Definiranjem primarnih i vanjskih ključeva kreiraju se veze između tablica na dijagramu. Preostaje jedino urediti veze kako se nebi presjecale radi preglednosti dijagrama. Veze se mogu pomicati klikom na njih, stvore se bijeli vrhovi u obliku kvadratića kojima se može promijeniti lokacija na bilo koje mjesto na rubovima kvadrata koji su simbol entiteta. Ako je potrebno presjeći vezu da nebude samo ravna crta, može se dodati još jedan 'vrh' desnim klikom na vezu i *Add Vertex*. Vanjski ključevi prema kojima su veze stvorene mogu se pregledavati i uređivati desnim klikom na vezu i odabir opcije *Design Relation*. Vezama koje su kreirane možemo mjenjati kardinalnost i notaciju. Kardinalnost veze određuje se desnim klikom na vezu, otvori se izbornik u kojemu su ponuđene opcije *Cardinality on 'naziv\_prvog\_entiteta'* i *Cardinality on 'naziv\_drugog\_entiteta'*. Odaberemo opciju s obzirom na to koju stranu veze želimo uređivati. Ponuđene kardinalnosti su: ništa, jedan i samo jedan, više, jedan ili više, nula ili jedan i nula ili više. Notacije koje su ponuđene u ovom alatu mogu se vidjeti i odabrati desnim klikom na radnu površinu i opcija *Diagram Notation*. **Slika 4.7** prikazuje sve notacije koje se mogu koristiti u Navicat alatu. Prva slika prikazuje simple notaciju sa kojom su prikazani samo nazivi entiteta spojeni linijama. Nisu prikazani ni atributi ni kardinalnost. Sljedeća slika b) prikazuje Crow's foot notaciju, primarni ključevi označeni su zvjezdicom pokraj. Na c) dijelu slike prikazana je IDEF1X notacija a na d) dijelu UML notacija. U primjeru sa slike sve veze imaju kardinalnost jedan i samo jedan naprema nula ili više.



**Slika 4.7 Notacije za prikaz konceptualnog modela u alatu Navicat**

Na kraju kada smo završili sa modelom, možemo spremiti promjene u model odabirom gumba *Save* iz trake ili možemo preslikati model u bazu podataka (eng. Forward engineering). **Slika 4.8** prikazuje postupak za preslikavanje modela u bazu. Nakon klika na *Synchronize to Database* pokreće se čarobnjak, najprije trebamo odabrati da li želimo izvršiti sinkronizaciju sa cjelom bazom ili samo sa tablicama koje su odabrane. Nakon toga odaberemo tablice i konekciju koje želimo usporediti i klikom na gumb *Compare* tablice sa modela uskladit će se sa onima u bazi.



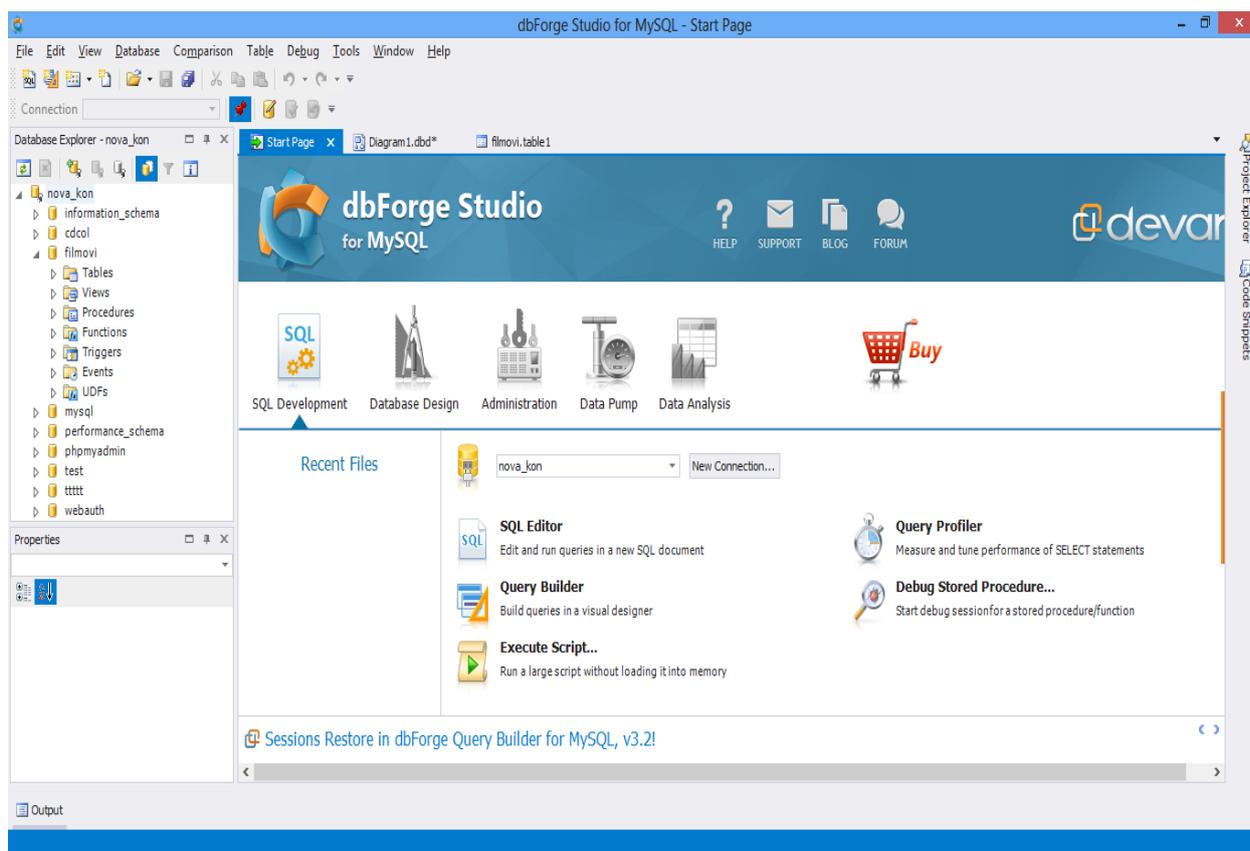
**Slika 4.8 Forward engineering u Navicat alatu**

## 4.2. dbForge Studio

### 4.2.1 Općenito o alatu

Alat dbForge Studio dostupan je za preuzimanje na stranici [www.devart.com](http://www.devart.com). U ponudi su 3 izdanja alata Express, Standard i Professional. Express edition je besplatno izdanje, sa osnovnim funkcionalnostima za rad sa objektima baze. Standard edition licenca može se kupiti za 95,95 \$ a Professional edition je izdanje sa najviše funkcionalnosti te se može kupiti za 199,95 \$. Prije nego se odlučimo na kupnju Professional izdanja imamo 30 dana besplatne verzije.

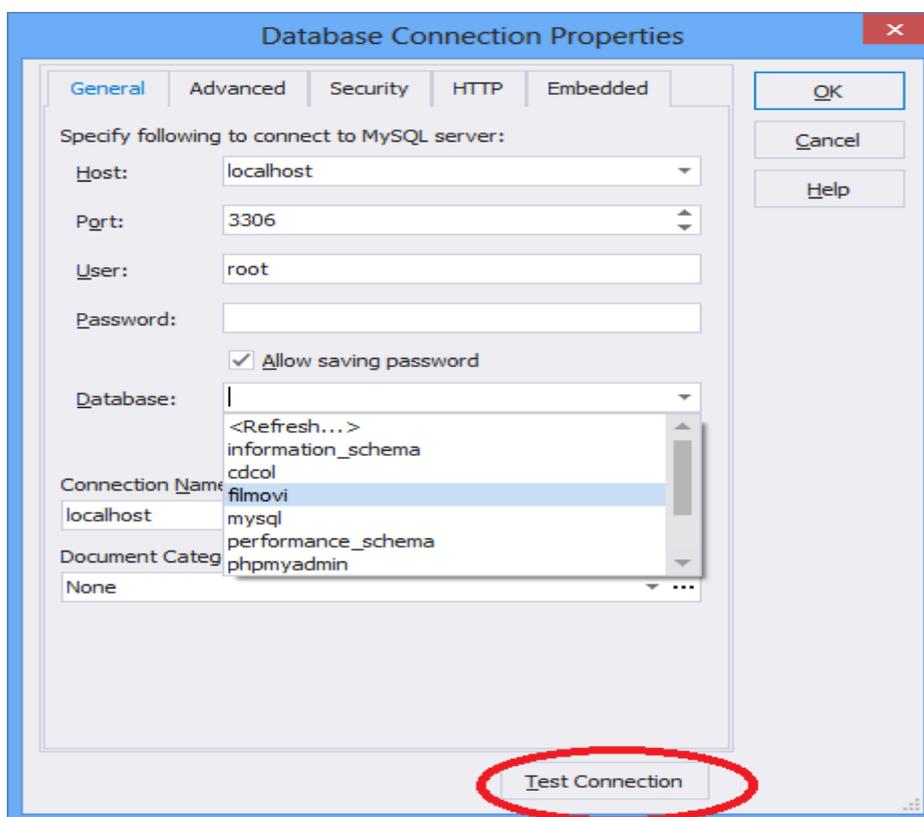
**Slika 4.9** prikazuje početni izgled alata, odmah pri pokretanju otvara se Start Page stranica na kojoj su funkcionalnosti grupirane u 5 kategorija. Prva grupa *SQL Development* sastoji se od *SQL editora* za pisanje SQL naredbi, *Query Builder* za izradu složenih upita i ostalih funkcionalnosti koje se mogu vidjeti na slici ispod. Sljedeća kategorija je *Database Design* koja služi za vizualni prikaz objekata iz baze sa kojom smo spojeni. *Administration* grupa koristi se za upravljanje korisničkim računima, backup baze i sl. U *Data Pump* kategoriji možemo napraviti izvoz i/ili uvoz podataka u/iz baze. Zadnja kategorija *Data Analysis* služi za kreiranje izvještaja.



**Slika 4.9** Početni zaslon alata dbForge Studio

## 4.2.2 Spajanje na MySQL server

Da bi započeli sa korištenjem alata, najprije se moramo povezati sa MySQL serverom. To ćemo napraviti odabirom *Database* iz alatne trake i *New Connection* iz pripadajućeg izbornika. Zatim će se otvoriti prozor kao što možemo vidjeti na slici **Slika 4.10**. U novootvorenom prozoru potrebno je ispuniti podatke kao što su računalo na koje se spajamo, u primjeru je to lokalno računalo, port za MySQL server i ime korisnika koji će se spajati. Ukoliko želimo možemo odabrati i već kreiranu bazu koja će se otvoriti svaki put kada otvorimo tu konekciju. Nakon ispunjavanja formulara potrebno je istestirati konekciju klikom na gumb *Test Connection*, i ako je sve u redu može se započeti sa daljnjim radom.

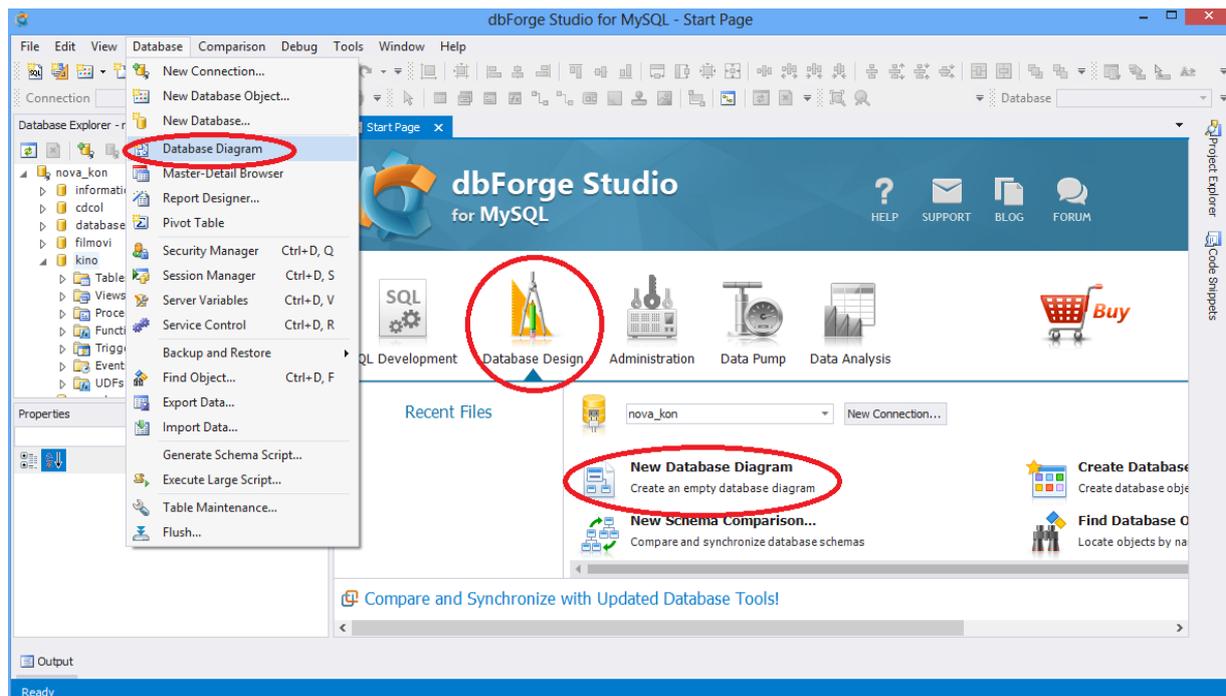


**Slika 4.10** Povezivanje sa MySQL serverom – alat dbForge Studio

Nakon uspješnog povezivanja vrijeme je za kreiranje nove baze podataka. U alatu dbForge najprije moramo otvoriti vezu desnim klikom na naziv veze i onda *Open*. Kada je veza otvorena, desnim klikom na nju odaberemo opciju *New Database*. U prozoru koji se otvori ispunimo ime baze, skup znakova i pravila za uspoređivanje, klik na *Update Database* i baze je kreirana. Za potrebe ovog rada, nisam kreirala novu bazu već sam iskoristila bazu koju sam kreirala u prethodno opisanom alatu Navicat.

## 4.2.3 Konceptualni model

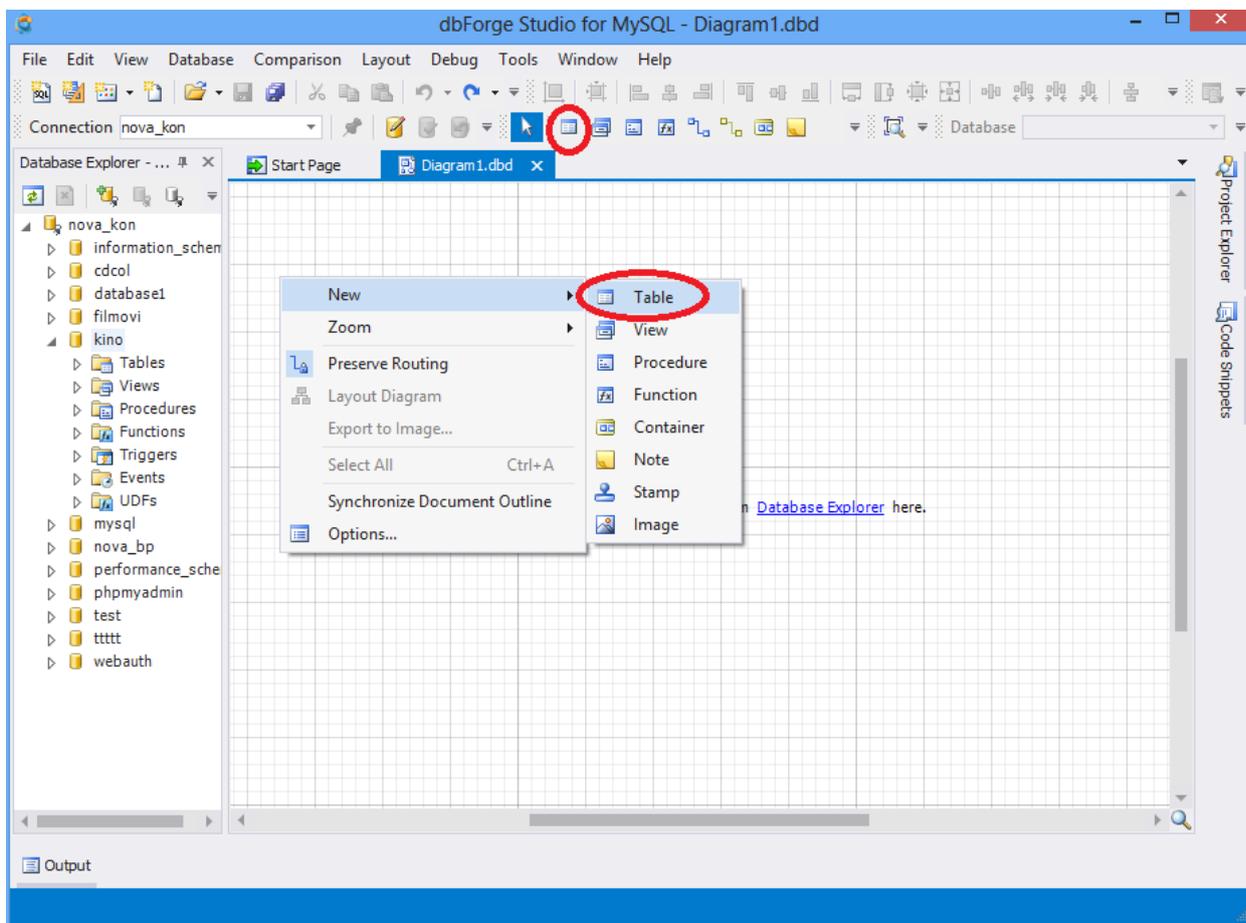
**Slika 4.11** prikazuje načine na koje možemo početi sa izradom novog konceptualnog modela. Novi model možemo izraditi odabirom *Database* iz alatne trake te *Database Diagram* iz ponuđenog izbornika. Drugi način za izradu modela je klikom na *Database Design* gumb na početnoj stranici i odabirom opcije *New Database Diagram*. Novi model može se kreirati i odabirom *File* iz alatne trake te opcije *New Database Diagram*. U svakom slučaju otvorit će se nova radna površina na koju možemo dodavati objekte.



**Slika 4.11 Konceptualni model – alat dbForge Studio**

### 4.2.3.1 Entiteti i atributi

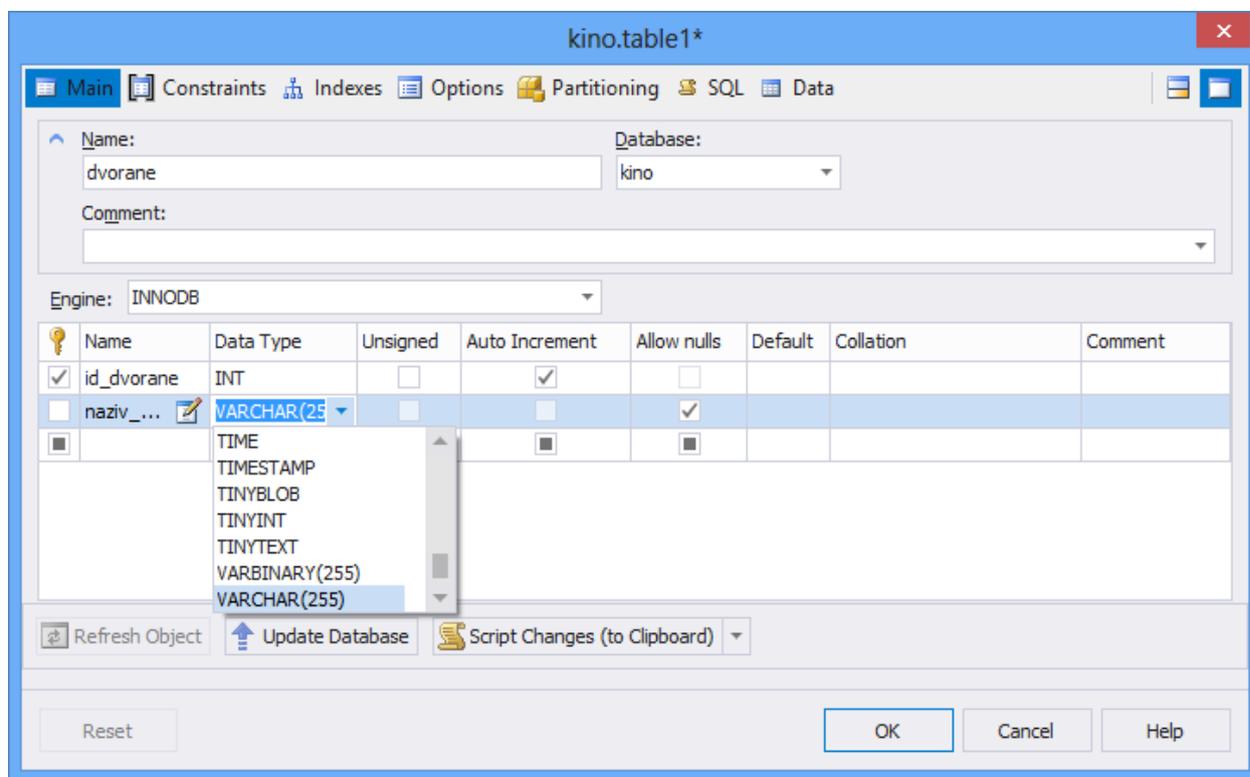
Novi entitet se može dodati na model na više načina. Neki od načina dodavanja novog entiteta prikazani su slikom **Slika 4.12**. Novi entitet možemo dodati odabirom oznake tablice iz trake iznad radne površine ili desnim klikom na radnu površinu i odabir opcije *New Table*. Alat nudi i mogućnost kreiranja modela iz već postojeće baze. Ukoliko želimo na model dodati entitete koji su već kreirani u nekoj od baza sa kojom smo spojeni možemo odabrati naziv baze ili samo nazive određenih tablica iz baze te ih povući i ispustiti na radnu površinu. Sve promjene na modelu alat automatski sprema u bazu koju smo definirali u trenutno otvorenoj konekciji.



**Slika 4.12 Dodavanje novog entiteta na model**

Odabirom opcije *New Table* otvara se novi prozor koji se može vidjeti na slici **Slika 4.13**. Za definiranje novog entiteta potrebno je upisati naziv kako ga želimo nazvati. Ispod se nalaze polja za unos atributa. Za attribute je potrebno definirati njihov naziv u stupac *Name* te tip podataka kojeg će biti taj atribut u stupac *Data Type*. Primarne ključeve označimo sa kvačicom u stupcu u kojemu se nalazi simbol ključa. Ostali stupci nisu obavezni za ispunjavanje ali pobliže opisuju svaki atribut. Možemo ih označiti u slučaju ako želimo odrediti da vrijednost nekog atributa može biti NULL (stupac *Allow nulls*), ako želimo da se neki atribut automatski povećava pri svakom novom unosu (stupac *Auto Increment*) i možemo odrediti default vrijednost koja će se unositi svaki put kada vrijednost atributa bude ostavljena prazna (stupac *Default*). Nakon unosa svih atributa odabirom kartice *Constraints* dodajemo vanjske ključeve ukoliko je potrebno. Novi vanjski ključ dodaje se desnim klikom na površinu unutar kartice *Constraints* i odabirom opcije *New Foreign Key*. Otvara se novi prozor u kojemu definiramo naziv vanjskog ključa, odaberemo atribut koji će biti vanjski ključ i tablicu i atribut na koji će se referencirati taj novi vanjski ključ. Klikom na OK novi vanjski ključ je kreiran i prema njemu će se stvoriti veza na dijagramu. Unique ograničenje se dodaje odabirom kartice *Indexes* desni klik na površinu i odabir opcije

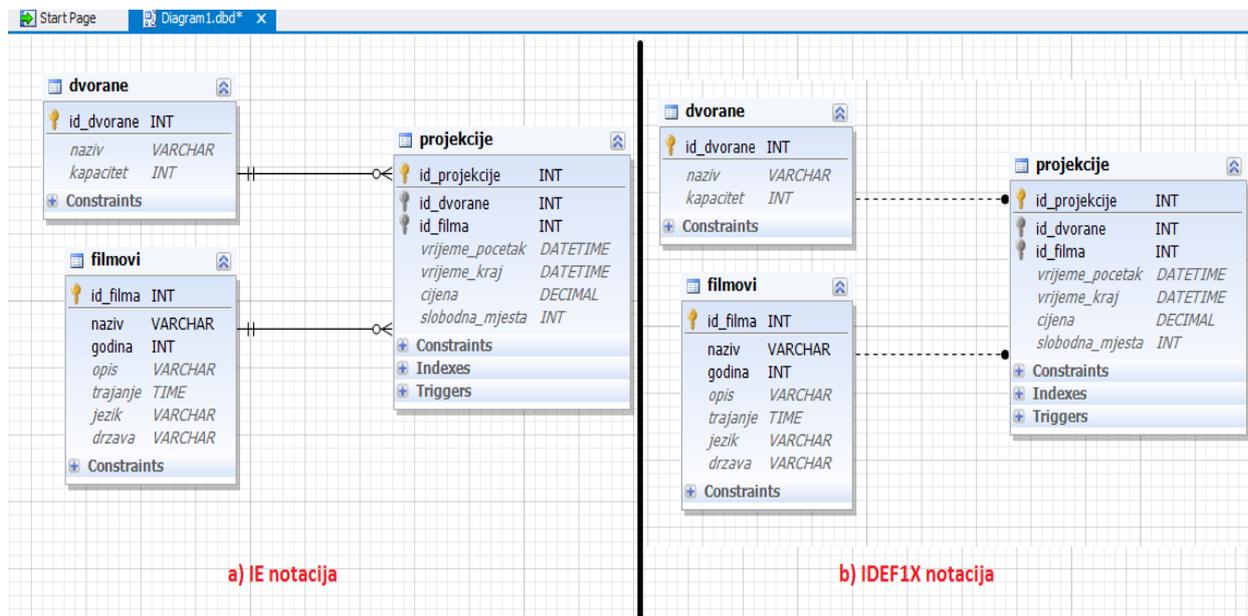
*New Index*. Otvara se novi prozor gdje definiramo naziv ograničenja i odaberemo atribut ili više njih na koje se treba odnositi ograničenje još je potrebno označiti sa kvačicom oznaku *unique* i kliknuti na gumb OK za završetak izrade. Nakon što dodamo sve atribute, ograničenja i vanjske ključeve klikom na OK u glavnoj (*Main*) kartici potvrđujemo promjene i novi entitet sa svim svojim obilježjima nalazi se na radnoj površini dijagrama.



**Slika 4.13 Naziv i atributi novog entiteta u alatu dbForge**

#### **4.2.3.2 Veze, kardinalnost i notacije**

Veze se kreiraju prema prethodno definiranim primarnim i vanjskim ključevima. Nedostatak ovog alata je taj što se nemože mjenjati kardinalnost na vezama. Desnim klikom na liniju koja predstavlja vezu između entiteta možemo uređivati, mjenjati ili brisati vanjske ključeve koji sudjeluju u toj vezi odabirom opcije *Edit Foreign Key*. Alat nudi dvije notacije za prikaz ERA modela IE i IDEF1X notaciju. Željenu notaciju možemo odabrati desnim klikom na radnu površinu, zatim *options*. U prozoru koji se otvorio odaberemo karticu *Notation* i unutar te kartice dio gdje piše *Relation notation*. Iz padajućeg izbornika odaberemo da li želimo IE ili IDEF1X notaciju. Na slici **Slika 4.14** prikazane su obadvije notacije. Na a) dijelu slike prikazan je model IE notacijom a na drugom dijelu slike prikazan je isti taj model samo IDEF1X notacijom.

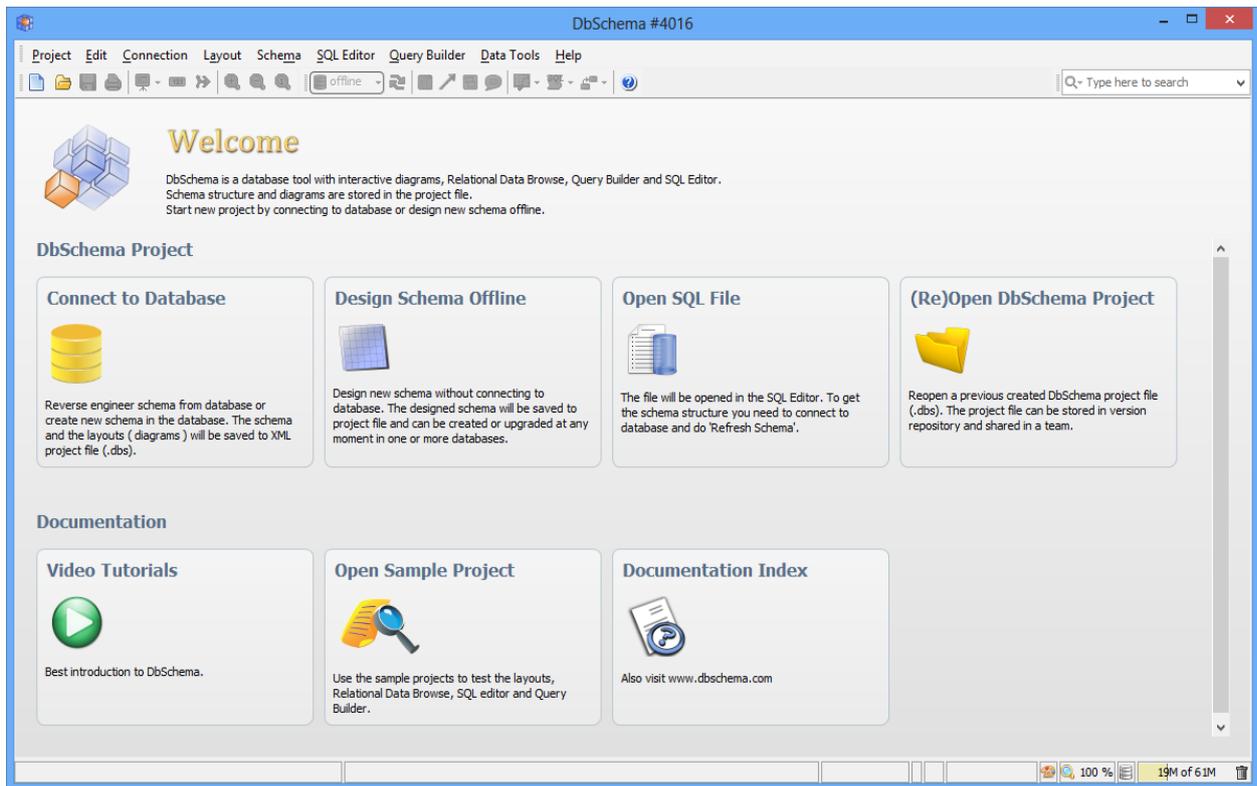


Slika 4.14 Notacije u alatu dbForge

## 4.3 DbSchema

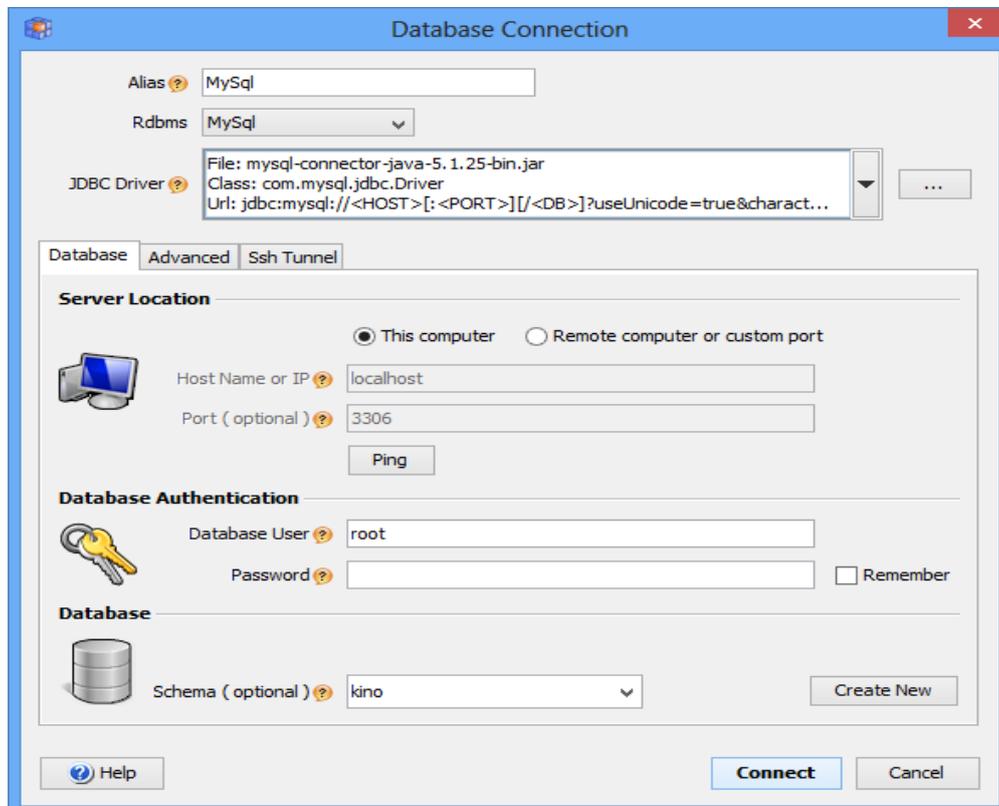
### 4.3.1 Općenito o alatu

Alat DbSchema može se preuzeti sa stranice <http://www.dbschema.com/>. Može se koristiti besplatno, ali sa smanjenim brojem funkcionalnosti i mogućnošću kreiranja samo 12 tablica. Besplatna verzija alata za mnoge korisnike nije dovoljna, te se u tom slučaju može preuzeti probna verzija Ultimate Edition-a koja traje 15 dana. Nakon toga se alat treba kupiti. U ponudi za kupnju su Personal, Commercial i Academic verzije. Personal verzija može se preuzeti za 127 \$ i može ju koristiti samo jedna osoba (neprenosiva je na više ljudi). Ako alat treba koristiti više ljudi, npr. u nekoj organizaciji, licenca je 198 \$ za Commercial Edition. Academic verzija najpovoljnija je i može se preuzeti za 63 \$, a namijenjena je za edukativne svrhe kao što joj i samo ime govori. Na slici **Slika 4.15** prikazan je izgled početnog zaslona DbSchema alata. Na početnom zaslonu istaknute su opcije sa kojima započinjemo rad sa alatom. Prva opcija je spajanje sa bazom, zatim je ponuđeno kreiranje konceptualnog modela bez spajanja, možemo otvoriti SQL datoteku ili već započeti projekt. Dalje u ovom radu prikazat ću prvu opciju, spajanje na bazu.



Slika 4.15 Početni zaslona alata DbSchema

#### 4.3.2 Spajanje na MySQL server

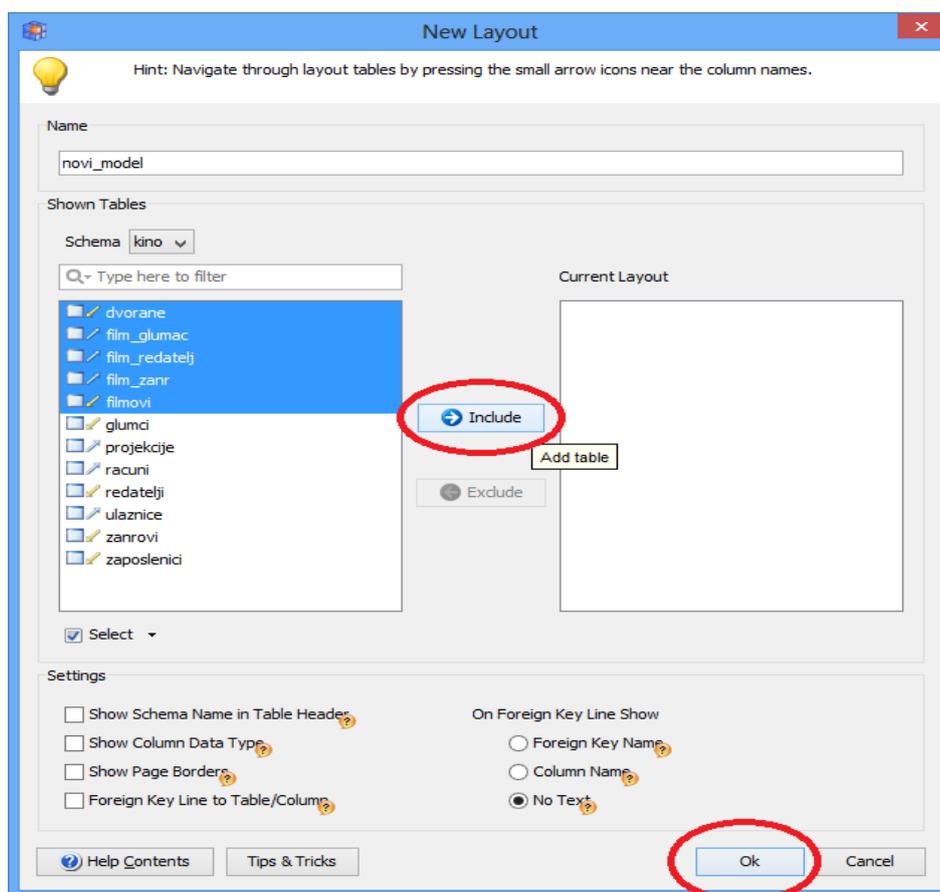


Slika 4.16 Spajanje na bazu – alat DbSchema

Nova veza sa bazom kreira se odabirom *Connect to Database* opcije sa početnog zaslona. Otvori se novi prozor sa postavkama za spajanje, kao što se može vidjeti na slici **Slika 4.16**. Potrebno je odabrati lokaciju servera, da li je to lokalno računalo (*This computer*) ili udaljeno računalo (*Remote computer*). Zatim se ispuni ime korisnika i možemo odabrati bazu na koju se želimo spojiti ili kreirati novu klikom na gumb *Create New*. U ovom primjeru odabrala sam već kreiranu bazu kino. U nastavku rada prikazat ću izradu konceptualnog modela baze podataka.

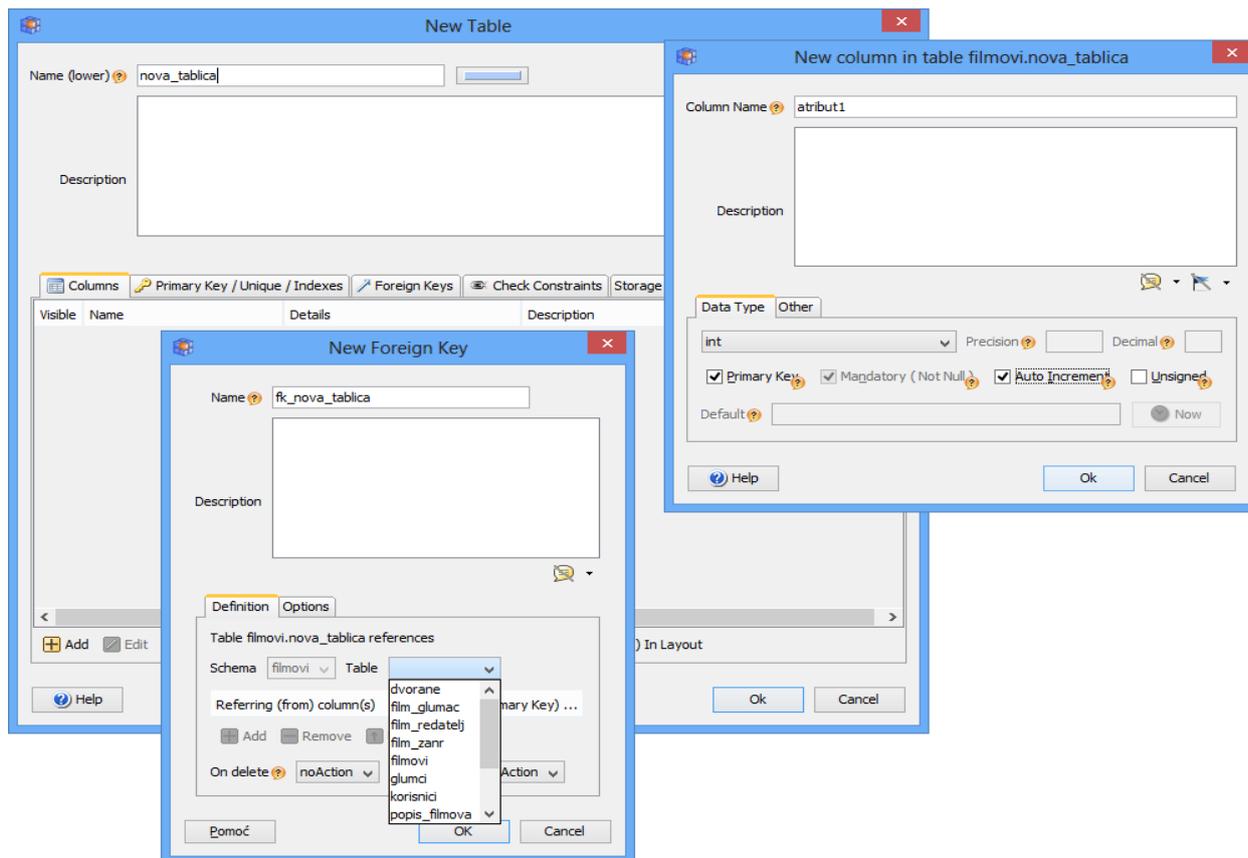
### 4.3.3. Konceptualni model

Konceptualni model u alatu DbSchema kreira se desnim klikom na *Layouts* i *New Layout*. Otvori se novi prozor, koji se može vidjeti na slici **Slika 4.17**. Napišemo naziv koji želimo za novi model, ispod se nalazi popis tablica iz baze koje možemo označiti i klikom na gumb *Include* smjestiti ih na model. Svaka promjena na radnoj površini, dodavanje, mjenjanje ili brisanje objekata automatski se bilježi u bazi sa kojom smo spojeni. Konceptualni model može se izraditi i klikom na *Project* iz trake na vrhu prozora, zatim iz ponuđenog izbornika odabrati *New project connected to database* ili *new project offline*, ovisno o tome da li želimo raditi bez spajanja ili sa spajanjem na bazu.



**Slika 4.17** Izrada konceptualnog modela u alatu DbSchema

### 4.3.3.1 Entiteti i atributi



Slika 4.18 Kreiranje novog entiteta – alat DbSchema

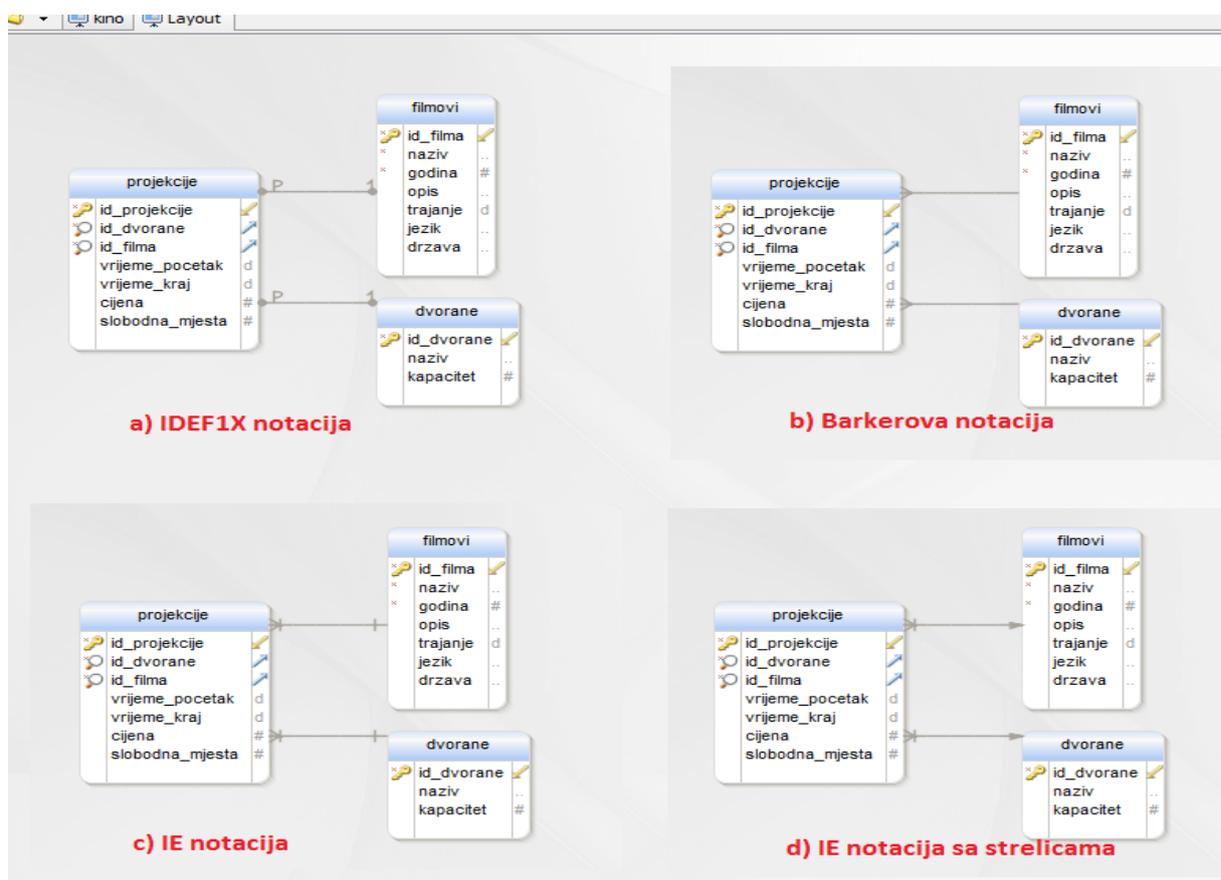
Entiteti se na model dodaju desnim klikom na radnu površinu i odabir opcije *Create Table* ili klikom na simbol tablice iz alatne trake na vrhu prozora. Bilo koji način odabrali, otvori se novi prozor, kao što se može vidjeti na slici **Slika 4.18**. U novom prozoru unosimo naziv entiteta u polje *Name*. Nakon toga možemo započeti sa dodavanjem atributa. Atributi se dodaju odabirom kartice *Columns* i zatim gumb *Add* na dnu prozora. Klikom na *Add* otvara se novi prozorčić koji se također može vidjeti na slici **Slika 4.18**. U njega upisujemo naziv atributa koji želimo dodati, određujemo tip podataka koji će imati taj atribut te označimo kvačicom *Primary Key* ako je taj atribut ujedno i primarni ključ i ako želimo da se pri svakom unosu automatski povećava označimo sa kvačicom *Auto Increment*. Kada smo završili sa osobinama atributa klik na *OK* vratit će nas na početni prozor u kojemu možemo dodavati sljedeći atribut na isti taj način. Za kreiranje unique ograničenja odabere se kartica pokraj *Columns* i opet kao i za novi atribut odaberemo gumb *Add* koji će otvoriti prozor za dodavanje novog ograničenja. Na isti način kreiraju se i vanjski ključevi, samo se najprije odabere kartica *Foreign Keys*. Prozor koji se otvori prikazan je na slici **Slika 4.18**. U njega se upisuje naziv vanjskog ključa, zatim se odabere

tablica koju želimo referencirati i ispod se odabere atribut koji će biti vanjski ključ i atribut iz referencirane tablice.

Alat nudi i mogućnost dodavanja već gotovih tablica iz baze sa kojom smo spojeni na model. Samo označimo naziv baze ili naziv pojedine tablice iz baze i povučemo ih te spustimo na radnu površinu.

Pokraj nekih atributa na modelu nalaze se simboli kao što su ključ, povećalo ili strelice. Simbol ključa nalazi se pokraj atributa koji je primarni ključ u nekoj tablici. Povećalo označava vanjske ključeve a strelice su oznake za attribute koji su referencirani ili koji se referenciraju na primarni ključ neke druge tablice. Tako žuta strelica okrenuta prema dolje znači da je taj atribut referenciran od neke druge tablice a plava strelica okrenuta prema gore znači da se taj atribut referencira na atribut iz neke druge tablice.

#### 4.3.3.2 Veze, kardinalnost i notacije



Slika 4.19 Notacije u alatu DbSchema

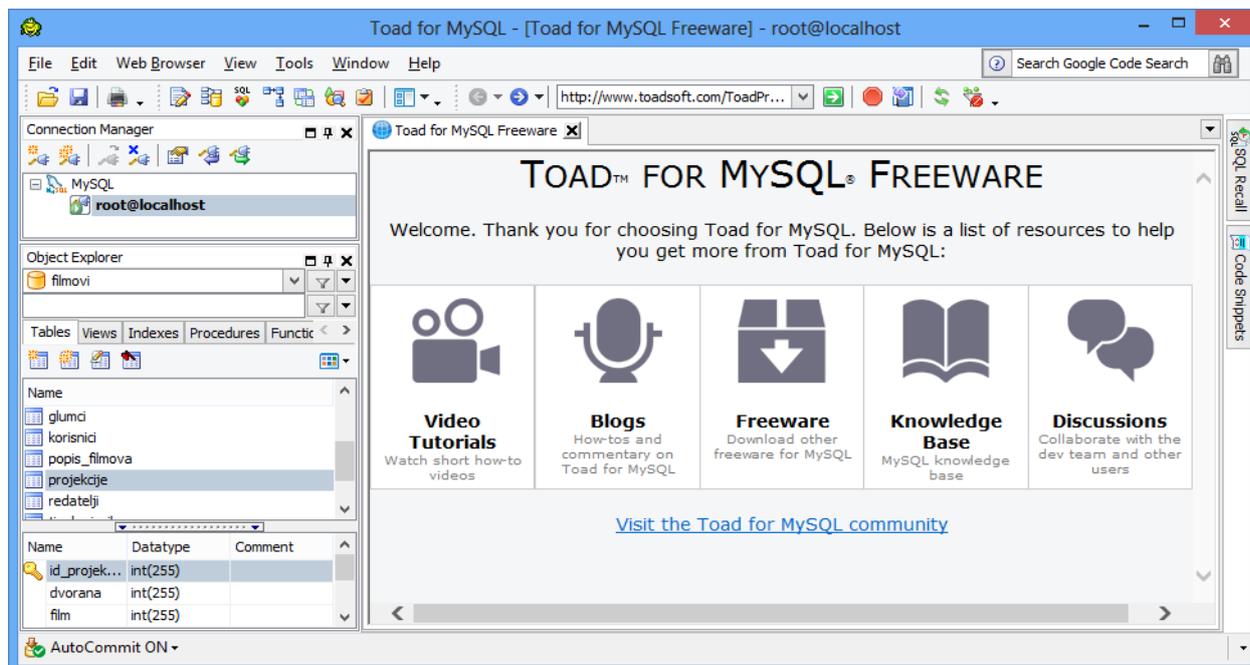
Kao i u ostalim, do sada opisanim alatima, veze između tablica kreiraju se prethodnim definiranjem primarnih i vanjskih ključeva. U ovom alatu tablice se mogu spajati i tako da odaberemo atribut iz jedne tablice i povučemo ga na atribut iz druge tablice sa kojom želimo izraditi vezu. Nedostatak ovog alata je što nemožemo mjenjati kardinalnost na modelu i ne mogu se pomicati veze kako bismo ih uredili radi bolje preglednosti. Veze koje su već kreirane možemo uređivati desnim klikom na vezu i odabir opcije *Edit*.

**Slika 4.19** prikazuje primjer gdje je spojena tablica projekcije sa tablicama filmovi i dvorane vezom jedan-više. Isti primjer prikazan je četiri puta, svaki put sa drugom notacijom koju podržava alat DbSchema. Tako, a) dio slike prikazuje IDEF1X notaciju, b) dio slike Barkerovu notaciju a c) i d) djelovi slike odnose se na IE odnosno Crow's foot notaciju s tim da je c) slika bez strelica, a d) slika sa strelicama koje se nalaze na strani entiteta koji je referenciran.

## 4.4. Toad

### 4.4.1 Općenito o alatu

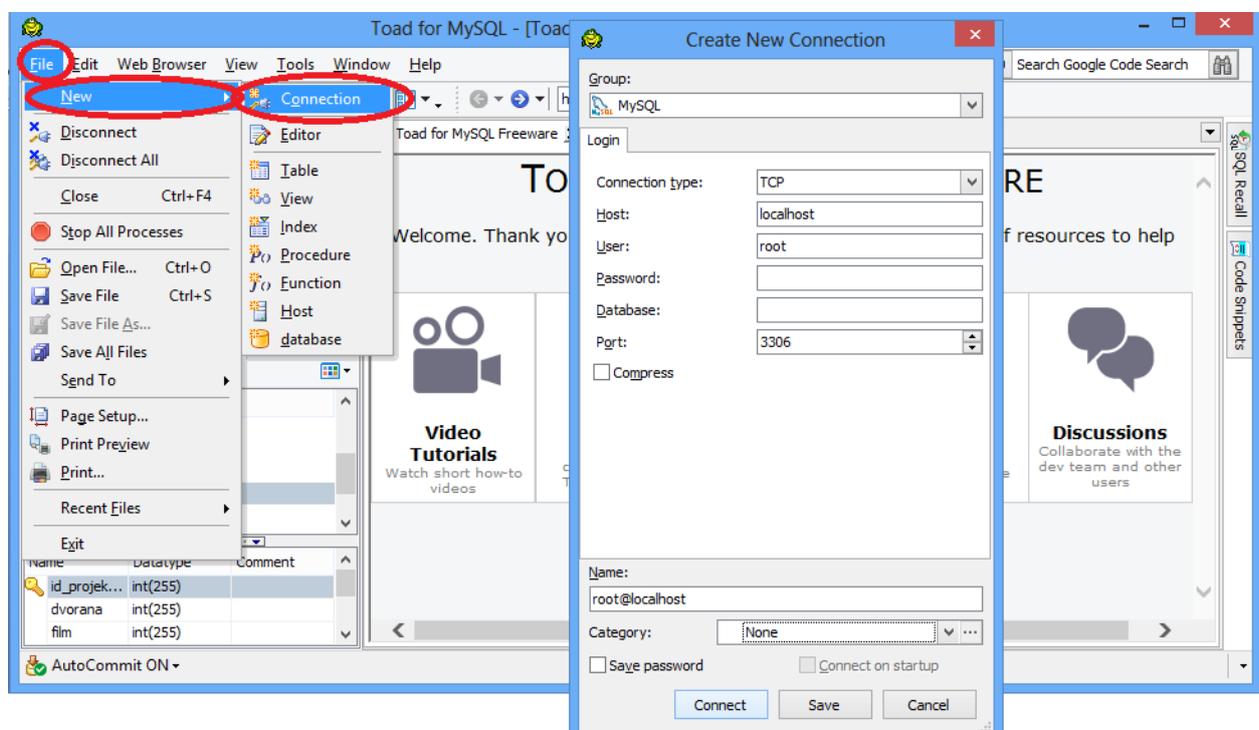
Toad alat za konceptualno modeliranje može se preuzeti sa sljedeće stranice <http://www.toadworld.com/m/freeware/552.aspx>. Alat koji sam preuzela je besplatan za korištenje, ali postoji i komercijalna verzija koja se može kupiti za 592 \$. Dostupan je samo za Microsoft Windows platformu. Nakon instalacije alata može se započeti sa korištenjem. Glavni prozor alata vidi se na slici **Slika 4.20**.



**Slika 4.20** Početni zaslon alata Toad

#### 4.4.2 Spajanje na MySQL server

Za početak, potrebno je povezati se sa MySQL serverom kako bismo mogli izraditi bazu podataka i njezine objekte, to se može vidjeti na slici **Slika 4.21**. U alatu Toad nova konekcija stvara se klikom na *File* iz alatne trake te odabirom *New Connection* opcije. Nakon toga ispuni se otvoreni formular sa potrebnim podacima kao što su ime računala, korisnik i port i klik na gumb *Connect*. Sada se nova konekcija nalazi na popisu sa lijeve strane glavnog prozora. Kada želimo koristiti tu konekciju, desnim klikom na naziv odaberemo opciju *Connect*. Nakon uspješnog povezivanja, ispod naziva konekcije bit će prikazane sve baze podataka koje su dostupne isto kao i objekti tih baza podataka. Nova baza podataka kreira se odabirom *File* iz alatne trake te zatim *New database* opcije.



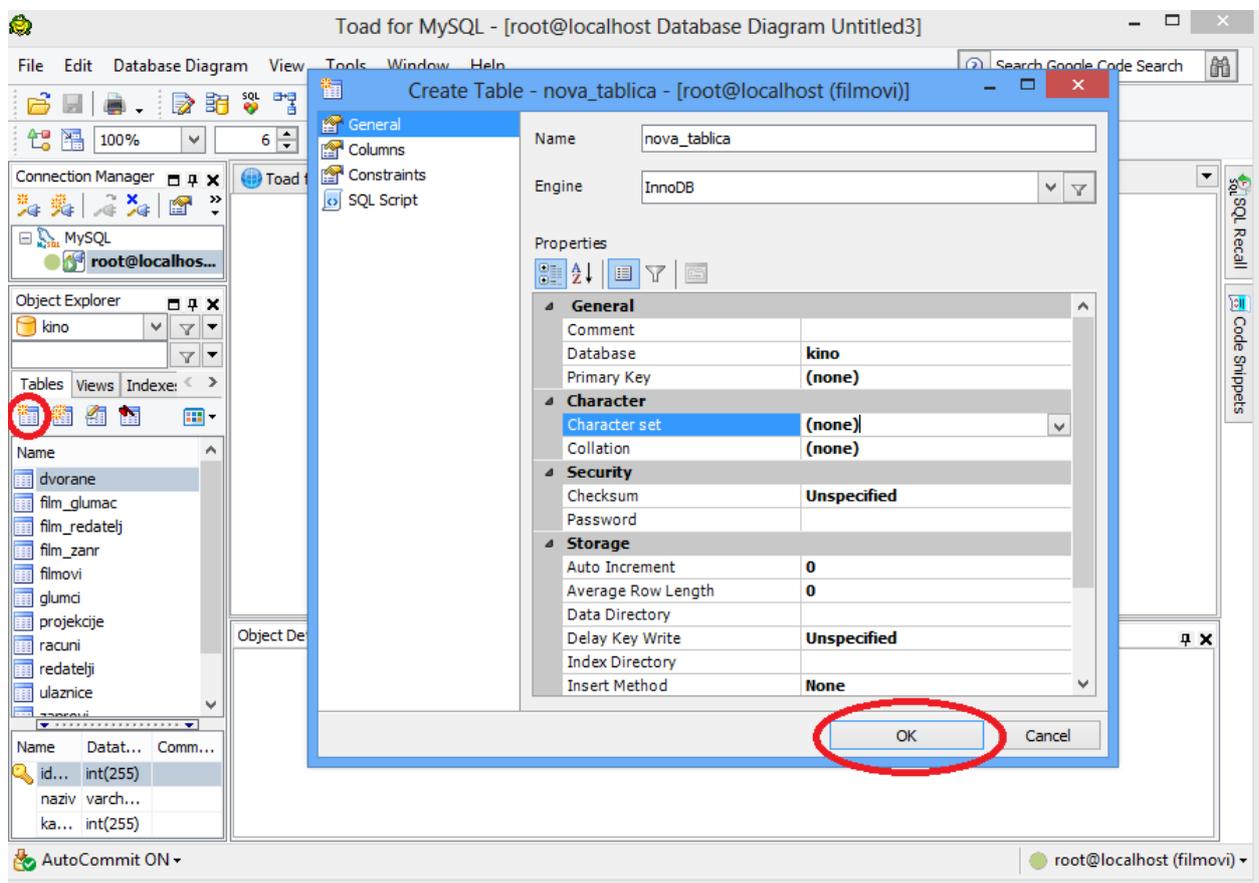
**Slika 4.21 Povezivanje sa MySQL serverom – alat Toad**

#### 4.4.3. Konceptualni model

Novi konceptualni model kreira se odabirom *Tools* iz alatne trake te zatim opcija *Database Diagram*. Otvori se kartica sa praznom radnom površinom na kojoj će se kasnije nalaziti model baze podataka. U ovom alatu obavezno je prvo spojiti se sa bazom kako bi mogli izrađivati modele i nepodržava offline izradu konceptualnih modela. Funkcionalnosti za izradu modela dosta su smanjene jer je ovo besplatna (eng. freeware) verzija alata.

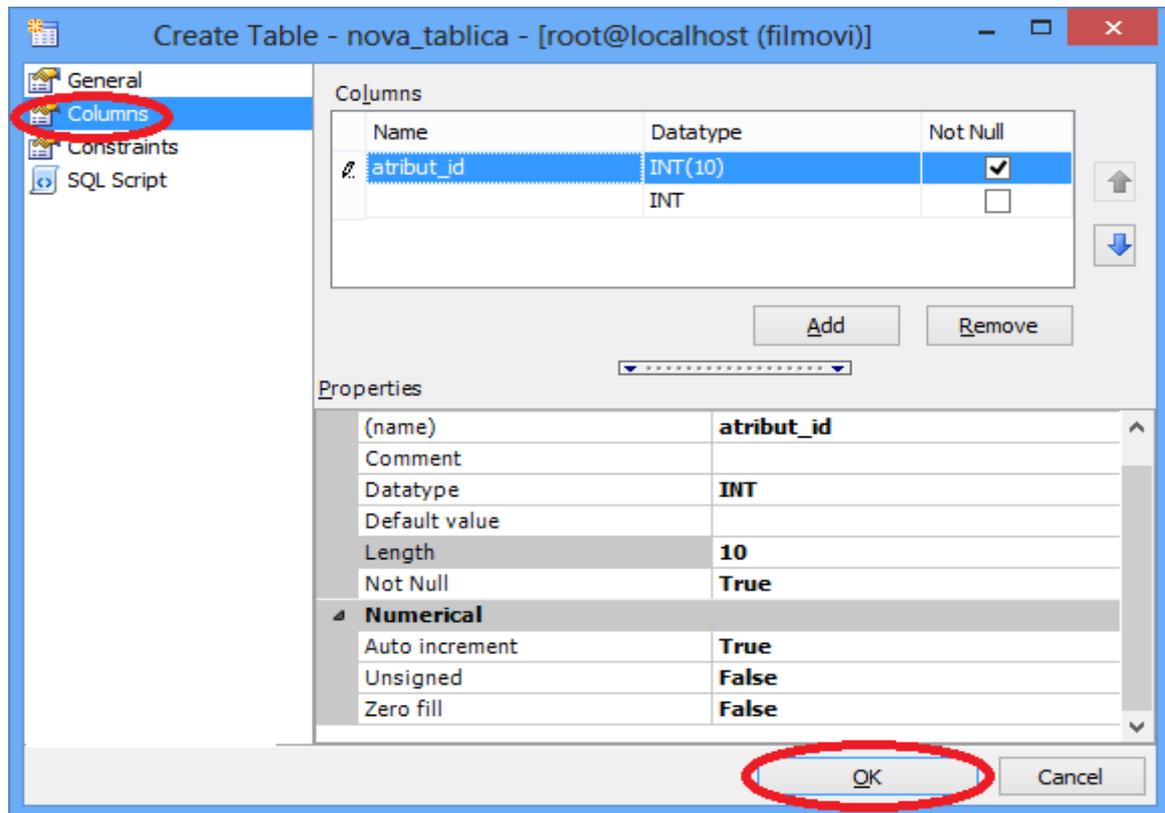
#### 4.4.3.1 Entiteti i atributi

Za dodavanje novih entiteta na model, najprije moramo kreirati tablicu u bazi te ju onda povući na površinu modela. Kako to izgleda u Toad alatu može se vidjeti na slici **Slika 4.22**. Da bismo dodali novu tablicu, najprije iz *Object Explorera* odaberemo bazu koju želimo. Zatim karticu *Tables* te simbol tablice koji je zaokružen na slici. Nakon toga otvara se novi prozor u kojemu ispunjavamo osnovne podatke o tablici kao što je naziv tablice, skup znakova koje će koristiti i ostale informacije koje se mogu vidjeti na slici ispod. U istom prozoru možemo odmah dodavati attribute i ograničenja ukoliko želimo. Kada se završi sa definiranjem tablice klik na gumb *OK* i nova tablica nalaziti će se na popisu sa lijeve strane. Zatim samo preostaje povući ju i spustiti na radnu površinu modela.



**Slika 4.22** Nova tablica – Toad alat

Na slici **Slika 4.23** prikazano je dodavanje atributa tablici. Atributi se dodaju odabirom opcije *Columns* pri kreiranju same tablice. Ispune se potrebne informacije kao što su ime atributa, tip, da li atribut može poprimiti null vrijednost ili ne te ostala svojstva navedena ispod. Primarni ključ se definira na prvoj opcije *General*. Na slici se vidi još jedna opcija pri kreiranju nove tablice, a to je *Constraints*, klikom na nju definiramo vanjske ključeve tablice.

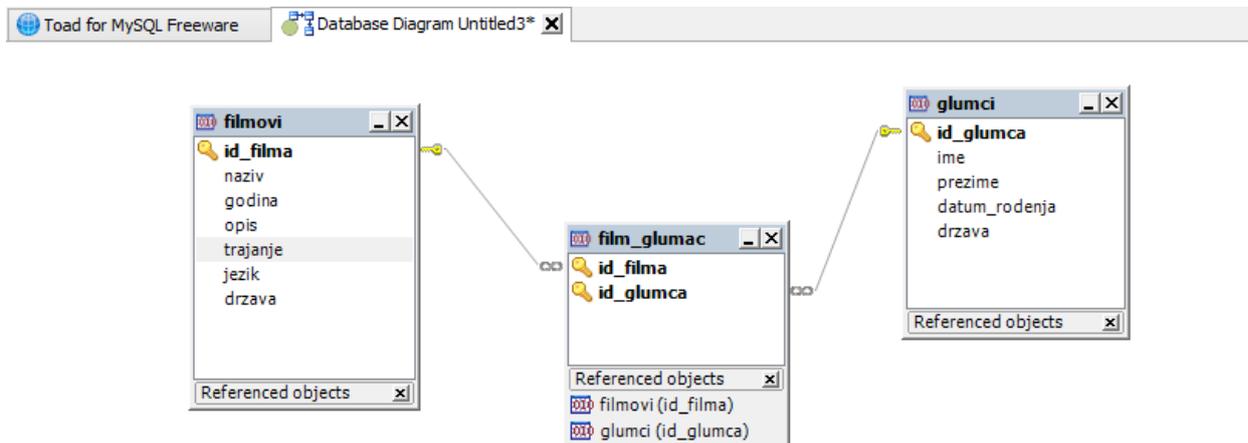


**Slika 4.23 Dodavanje atributa u alatu Toad**

U slučaju da nema potrebe za izradom novih tablica, ako sve potrebne tablice već imamo u bazi, alat nudi mogućnost preslikavanja svih objekata iz baze na model. To se može napraviti odabirom *Database Diagram* iz alatne trake te zatim klikom na opciju *All tables and Views From Current Database Only*. *Database Diagram* opcija iz alatne trake još nudi mogućnost *Arrange Tables* koja posloži tablice tako da nema presjecanja veza te *Autosize Tables* koja prilagodi veličinu tablice tako da se jasno vidi naziv i svi atributi svake tablice.

#### **4.4.3.2 Veze, kardinalnost i notacije**

Tablice se povezuju s obzirom na definirane primarne i vanjske ključeve. Možemo ih povezati i povlačenjem atributa iz jedne tablice na atribut iz druge tablice. Notacija kojom su prikazane veze među tablicama ne može se promjeniti. Oznaka ključa na vezi prikazuje tablicu u kojoj je primarni ključ a oznaka beskonačno je na strani gdje je vanjski ključ. Ovo je besplatna (eng. freeware) verzija alata te su stoga funkcionalnosti za uređivanje modela dosta smanjene. **Slika 4.24** prikazuje izgled konceptualnog modela u alatu Toad. Desnim klikom na radnu površinu odabirom *properties* možemo odrediti da li želimo da se prikazuju detalji atributa i primarni ključevi. Također desnim klikom na bilo koju tablicu možemo pregledavati ili mjenjati njezine postavke.



**Slika 4.24 Konceptualni model u alatu Toad**

## 4.5 Usporedba alata

U prethodnom dijelu opisala sam način rada sa 4 alata s grafičkim korisničkim sučeljem za konceptualno modeliranje baze podataka. Može se primjetiti da je rad u svim alatima vrlo sličan. U tablici **Tablica 4.1** prikazala sam neke osnovne značajke svakog alata prema kojima su alati uspoređeni. Značajke koje sam izdvojila su operacijski sustav na kojemu se alat može instalirati, podržani sustavi za upravljanje bazama podataka, cijena alata, notacije kojima se mogu prikazati veze na modelima i da li alat ima mogućnost kreiranja modela bez spajanja na bazu. Kreiranje modela iz baze i izrada baze iz modela je isto vrlo bitna značajka, nisam ju navodila u tablici jer ju svi alati podržavaju.

Iz tablice se može vidjeti da Navicat i DbSchema alat podržavaju Windows, Linux i MAC OS dok preostala dva alata, Toad i dbForge mogu biti instalirana samo na Windows platformi. Svi alati podržavaju poprilično dosta sustava za upravljanje bazama podataka. Prema cijeni licence alati su podjednaki. Izdvojila bi dbForge i Toad jer nude i potpuno besplatne verzije, iako sa smanjenim funkcionalnostima. Navicat i DbSchema imaju više notacija za prikaz konceptualnog modela nego preostala dva alata te podržavaju offline način rada. Notacije koje su navedene za Toad mogu se koristiti samo u komercijalnoj verziji alata. U besplatnoj verziji Toad alata koja je prikazana u ovom radu, za spajanje tablica koriste se linije sa simbolom ključa i beskonačno na krajevima linije.

**Tablica 4.1 Značajke alata za konceptualno modeliranje**

Naziv alata	OS	Podržani SUBP	Cijena licence	Podržane notacije	Offline način rada (bez konekcije)
Navicat	Windows, Linux, MAC OS	SQL Server, Postgre SQL, MySQL, Oracle, MariaDB, SQLite	Premium: 799\$ Enterprise: 199\$ Standard: 129\$ Non-commercial: 99\$	Simple, IE, IDEF1X i UML	DA
dbForge	Microsoft Windows	SQL Server, Postgre SQL, MySQL, Oracle	Express: besplatno KOMERCIJALNO Standard: 95,95\$ Professional: 199,95\$ NEKOMERCIJALNO Standard: 47,95\$ Professional: 99,95\$	IDEF1X i IE	NE
DbSchema	Windows, Linux, MAC OS	SQL Server, Postgre SQL, MySQL, Oracle, SQLite, Derby, Db2, Ingres, Sybase Ase ...	Commercial: 198\$ Personal: 127\$ Academic: 63\$	IE sa strelicama, IE bez strelica, Barker i IDEF1X	DA
Toad	Microsoft Windows	SQL Server, Postgre SQL, MySQL, Oracle, SQLite, Sybase Ase ...	Freeware: besplatno Komerrijalno: 592 \$	IDEF1X i IE	NE

## 5. Zaključak

Izradom ovog završnog rada proučila sam rad u četiri alata za modeliranje baze podataka. Sučelje tih alata vrlo je intuitivno i jednostavno za naučiti i za osobe koje nisu stručnjaci u tom području. Današnji alati osim izrade konceptualnih modela nude široki spektar ostalih funkcionalnosti. Uz pomoć njih može se izraditi model, pretvoriti model u logički model i izraditi bazu podataka. Preko grafičkog sučelja mogu se unositi podaci u bazu ili postavljati upiti. Alati obično nude i mogućnost nasumičnog generiranja podataka što je korisno za testiranje baze a skraćuje vrijeme koje bi potrošili na unos testnih podataka. Razvoj alata sa grafičkim korisničkim sučeljem doprinosi bržoj izradi konceptualnih modela.

Na kraju ovog rada mogu zaključiti i još jednom ponoviti da je izrada konceptualnog modela vrlo bitan korak pri dizajniranju baze podataka. Svaka izrada novog informacijskog sustava započet će sa modeliranjem baze podataka. Iako je modeliranje vrlo mali dio u odnosu na cjeli informacijski sustav, on je itekako bitan i utječe na kvalitetu budućeg sustava. Zbog raznih pritisaka i kratkih vremenskih rokova, ponekad se neposveti dovoljno vremena na analizu i dizajn. To može dovesti do kasnijih problema kao što su nedostatak ključnih podataka za izradu bitnih izvještaja ili netočni podaci. Dobro je utrošiti više vremena na dizajn baze podataka jer će se isplatiti u budućnosti.

## 6. Literatura

1. Anić, V., Goldstein, I. (2000). *Rječnik stranih riječi*. Novi liber.
2. Halpin, T., Morgan, T. (2008). *Information Modeling and Relational Databases*. Burlington: Morgan Kaufmann Publishers
3. Hay, D. (1999). A Comparison of data modeling techniques. Dostupno 5.9.2014. na [http://csis.bits-pilani.ac.in/faculty/goel/course\\_material/Data%20Warehousing/II%20sem%202004-05/Assignmet%201/comparison%20of%20DM%20techniquis.pdf](http://csis.bits-pilani.ac.in/faculty/goel/course_material/Data%20Warehousing/II%20sem%202004-05/Assignmet%201/comparison%20of%20DM%20techniquis.pdf)
4. Maleković, M. (2013). *Prezentacija sa predavanja: Konceptualno oblikovanje baze podataka*
5. Rabuzin, K. (2011). *Uvod u SQL*. Čakovec: Zrinski d.d.
6. Rabuzin, K. (2014). *SQL - napredne teme*. Čakovec
7. Simsion, G. (2007). *Data Modeling: Theory and Practise*. Bradley Beach: Technics Publications
8. Simsion, G., Witt, G. (2005). *Data Modeling Essentials*. Dostupno 5.9.2014. na <http://uk-corp.org/Books/Other-Technical/Data%20Modeling%20Essentials%202005.pdf>
9. Teorey, T. (1999). *Database Modeling & Design*. San Francisco: Morgan Kaufmann Publishers