



IT250 - BAZE PODATAKA

Konceptualno modeliranje
podataka primenom E/R
dijagrama

Lekcija 03

PRIRUČNIK ZA STUDENTE

IT250 - BAZE PODATAKA

Lekcija 03

KONCEPTUALNO MODELIRANJE PODATAKA PRIMENOM E/R DIJAGRAMA

- ✓ Konceptualno modeliranje podataka primenom E/R dijagrama
- ✓ Poglavlje 1: Konceptualno modeliranje baza podataka
- ✓ Poglavlje 2: Tip entiteta, entit, atribut i identifikator
- ✓ Poglavlje 3: Relacije
- ✓ Poglavlje 4: Verzije ER dijagrama
- ✓ Poglavlje 5: Poslovna pravila
- ✓ Poglavlje 6: Primeri kreiranja konceptualnog modela
- ✓ Poglavlje 7: Pokazna vežba
- ✓ Poglavlje 8: Domaći zadatak
- ✓ Zaključak

Copyright © 2017 – UNIVERZITET METROPOLITAN, Beograd. Sva prava zadržana. Bez prethodne pismene dozvole od strane Univerziteta METROPOLITAN zabranjena je reprodukcija, transfer, distribucija ili memorisanje nekog dela ili čitavih sadržaja ovog dokumenta., kopiranjem, snimanjem, elektronskim putem, skeniranjem ili na bilo koji drugi način.

Copyright © 2017 BELGRADE METROPOLITAN UNIVERSITY. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, without the prior written permission of Belgrade Metropolitan University.

▼ Uvod

UVOD

Šta ćemo naučiti u ovoj lekciji?

Konceptualno modeliranje podataka se radi s ciljem da se definiše struktura podataka relacione baze podataka. To je pojednostavljeno predstavljanje realnog sistema preko skupa entiteta, veza između entiteta i atributa entiteta. Konceptualni model podataka se najčešće predstavlja korišćenjem grafičkog jezika za predstavljanje strukture podataka - ER dijagrama. **Glavni koraci u kreiranju konceptualnog modela koji predstavljamo ER dijagramom se odnose na određivanje tipova entiteta, atributa tipova entiteta i relacije između tipova entiteta.**

U ovoj lekciji ćete naučiti:

- Glavne karakteristike komponenti entitet-odnos (Entity-Relationship)
- Kako se definišu, preciziraju i uključuju odnosi između entiteta u procesu dizajna baze podataka
- Kako komponente ER-a utiču na dizajn i implementaciju baze podataka
- Da stvarni dizajn baze podataka često zahteva usklađivanje suprotstavljenih ciljeva.

▼ Poglavlje 1

Konceptualno modeliranje baza podataka

ČEMU SLUŽE MODELI?

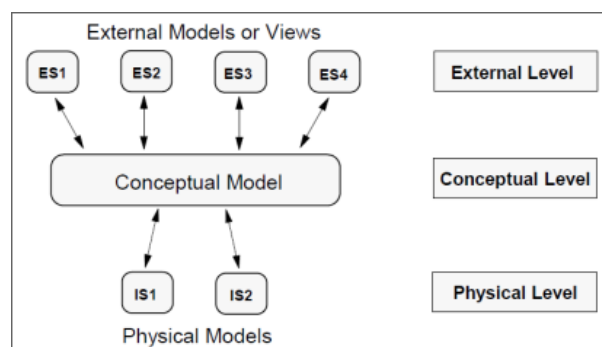
Svaki model je fokusiran na jedan aspekt realnosti koji je bitan za dati način njenog sagledavanja.

Svi modeli predstavljaju ograničenu prezentaciju realnosti.

Svaki model je fokusiran na jedan aspekt realnosti koji je bitan za dati način njenog sagledavanja.

Model podataka definiše strukturu i značenje podataka. Ova lekcija se prvenstveno bavi modelima podataka koji omogućavaju ponovnu upotrebu podataka u različitim aplikacijama, bilo integriranjem i razmenom podataka unutar jedne baze podataka, ili razmenom podataka na neke druge načine kao što je prenos datoteka.

Slika 1.1 pokazuje da se modeli podataka mogu javiti kao **spoljni modeli (ili pogledi)**, **konceptualni modeli** ili **fizički modeli**. Ovo nije jedini način da se modeli sagledaju, ali je koristan, posebno kada se javi potreba za upoređivanje modela.



Slika 1.1 ANSI / SPARC tronivovska arhitektura Izvor: [Matthew West and Julian Fowler (1999). Developing High Quality Data Models. The European Process Industries STEP Technical Liaison Executive (EPISTLE).]

Model podataka mora prikazivati određeni stepen konceptualne jednostavnosti, ne narušavajući semantičku kompletnost baze podataka. Nema smisla imati model podataka koji je teže konceptualizirati od stvarnog sveta.

Model podataka mora što vernije predstavljati stvarni svet. Ovaj cilj se lakše ostvaruje dodavanjem više semantike u reprezentaciju podataka modela. (Semantika se odnosi na dinamičko ponašanje podataka, dok reprezentacija podataka predstavlja statički aspekt

stvarnog scenarija.) Prikaz transformacija (ponašanja) stvarnog sveta mora biti u skladu sa karakteristikama konzistentnosti i integriteta svakog modela podataka.

EKSTERNI, KONCEPTUALNI I FIZIČKI MODEL PODATAKA

Eksterni model gleda na svet iz određene perspektive, za određenu svrhu; Konceptualni model podržava bilo koji eksterni model, a fizički predstavlja način za fizički smeštaj podataka

Eksterni model (ili pogled) gleda na svet iz određene perspektive, za određenu svrhu. Postoji mnogo mogućih spoljašnjih pogleda na svet: oni se mogu preklapati i ne moraju biti kompatibilni. Konkretno, spoljašnji pogledi su mesto za prikupljanje zahteva za podacima za određeni poslovni kontekst i pravila koja se odnose na njega. Modeli podataka većine aplikacija danas gledaju na svet iz perspektive te aplikacije, i na taj način predstavljaju spoljašnje modele.

Konceptualni model je osnovni ili neutralni model koji može podržati bilo koji valjan (a možda i promenljiv) spoljašnji pogled koji spada u njegov opseg. Kao rezultat, konceptualni model podataka nije odgovarajuće mesto za čuvanje pravila koja se mogu menjati. Korporativni model podataka bi trebalo da bude primer konceptualnog modela. Da bi podaci mogli da se integrišu iz različitih izvora, model podataka mora biti konceptualan u odnosu na izvorne modele podataka.

Fizički model predstavlja način na koji se podaci fizički čuvaju. Za konceptualni model može biti mnogo validnih fizičkih modela. Uslov je da fizički model mora da bude u stanju da podrži konceptualni model.

Najvažniji nivo modela za rešavanje problema razmene i deljenja podataka je konceptualni model. Konceptualni modeli su bolje prilagođeni modeliranju podataka na visokom nivou, dok su implementacioni modeli bolji za upravljanje sačuvanim podacima u svrhe implementacije. Model entitet-odnos (eng. **Entity-Relationship**) je primer konceptualnog modela, dok su hijerarhijski i mrežni modeli primeri implementacionih modela. Istovremeno, neki modeli, kao što su relacijski model i OODM (objektno orijentisani model podataka), mogu se koristiti i kao konceptualni i kao implementacioni modeli.

ZAŠTO SE RADI KONCEPTUALNO MODELIRANJE PODATAKA?

S ciljem da se definiše struktura podataka relacione baze podataka koja se naziva konceptualnim modelom podataka

Konceptualni model podataka je apstraktno viđenje stanja realnog sistema tj. definisane strukture podataka. To je pojednostavljeno predstavljanje realnog sistema preko skupa entiteta, veza između entiteta i atributa entiteta.

Konceptualni model podataka, preko skupa entiteta i njihovih međusobnih veza, predstavlja stanje sistema u jednom trenutku vremena i sadrži skup informacija o prošlosti i sadašnjosti sistema koje su potrebne da se pod dejstvom budućih poznatih ulaza mogu odrediti njegovi budući izlazi. Uobičajeno je da je to model koji nema dovoljno detalja za izgradnju stvarne baze podataka. Ovaj nivo opisuje strukturu celokupne baze podataka za grupu korisnika.

Konceptualni model je poznat i kao model podataka koji se može koristiti za opis idejne šeme kada se sistem baze podataka implementira. On sakriva interne detalje fizičkog skladištenja i cilj mu je da opiše entitete, tipove podataka, relacije i ograničenja.

Budući da konceptualni model predstavlja semantiku organizacije, a ne dizajn baze podataka, on može postojati na različitim nivoima apstrakcije. Originalna ANSI arhitektura sa četiri šeme počinje setom spoljnih šema od kojih svaka predstavlja viđenje jedne osobe o svetu oko sebe. One se konsoliduju u jednu konceptualnu šemu koja je podskup svih onih spoljašnjih pogleda.

KAKO SE PREDSTAVLJA KONCEPTUALNI MODEL PODATAKA?

Predstavlja se korišćenjem grafičkog jezika za predstavljanje strukture podataka (E/R dijagrama)

Konceptualni model pomaže u definisanju šta je bitno u samim podacima i kako oni međusobno povezani čine celinu. Konceptualni model je nezavisan i od softvera i od hardvera. Konceptualni model podataka se predstavlja korišćenjem grafičkog jezika za predstavljanje strukture podataka. Tokom prethodnih godina je razvijem veliki broj alata i tehnika za izradu modela podataka. Od njih se najduže u upotrebi zadržala tehnika Entity-Relationship dijagrami ili (ER) dijagrami koje je 1976. godine razvio Peter Chen.

ER digrami služe za predstavljanje konceptualnog modela podataka kojim se definiše strukturu podataka ili informacija koja se može implementirati u relacionoj bazi podataka. **ER dijagram je obično rezultat sistematske analize koja se vrši s ciljem da se definiše i opiše ono što je važno za procese u oblasti poslovanja.** Obično se crta korišćenjem pravougaonika koji predstavljaju tipove entiteta a koji su povezani linijama (relacijama) koje izražavaju asocijacije i zavisnosti između tipova entiteta.

Tipovi entiteta se mogu karakterisati ne samo relacijama, već i dodatnim osobinama (atributima), koji uključuju i identifikatore entiteta.

ER dijagrami se obično implementiraju kao relacione baze podataka. U jednostavnoj implementaciji relacione baze, svaka tabela predstavlja jedan tip entiteta, svaki red tabele predstavlja jednu instancu tipa entiteta, a svako polje u tabeli predstavlja atribut. U relacionoj bazi podataka odnos između entiteta se implementira čuvanjem primarnog ključa jednog entiteta kao pokazivača ili "stranog ključa" u tabeli drugog entiteta.

Kao softverska podrška za izradu konceptualnih modela korišćenjem ER dijagrama, koriste se CASE alati (eng. Computer Aided Software Engineering). Oni predstavljaju alate za definisanje konceptualnog modela podataka realnog sistema, odnosno za opisivanje skupa povezanih entiteta realnog sistema i događaja u njemu preko jednog složenog apstraktnog tipa podataka (strukture podataka, ograničenja i operacija).

Na ovom predmetu će se koristiti CASE alat PowerDesigner za koji Univerzitet Metropolitan ima licencu za korišćenje.

PRIMER KONCEPTUALNOG MODELA

Konceptualni E/R dijagram modeluje informacije prikupljene prikupljeni na osnovu poslovnih zahteva.

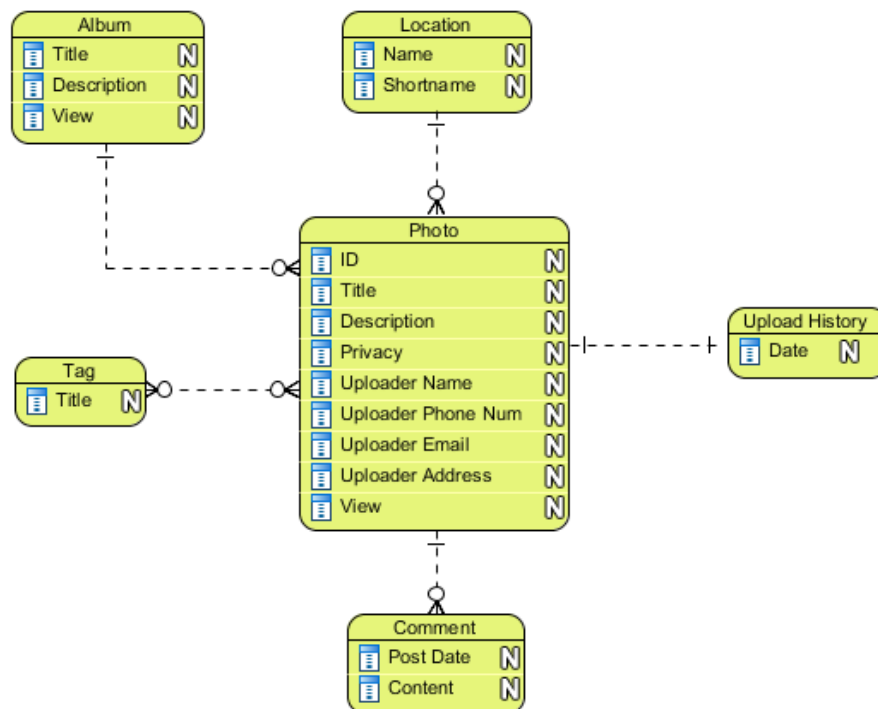
Kao što se vidi sa slike 1.2, konceptualni model podataka se sastoji od tipova entiteta, relacija između tipova entiteta sa definisanom kardinalnošću i atributa tipova entiteta.

Na dijagramu su predstavljeni sledeći tipovi entiteta sa odgovarajućim atributima:

1. **Photo** (slika) sa atributima ID, Title, Description, Privacy itd.
2. **Album** sa atributima Title, Description, View
3. **Tag** sa atributom Title
4. **Location** sa atributima Name i Shortname
5. **Upload History** sa atributom Date
6. **Comment** sa atributima Post Date i Content

Između tipova entiteta postoje relacije:

1. Da bi znali u kom se albumu nalazi slika, uspostavljena je relacija između tipova entiteta Album i Photo
2. Da bi znali lokaciju na kojoj je slika napravljena, uspostavljena je relacija između tipova entiteta Photo i Location
3. Da bismo mogli da ostavimo jedan ili više komentara vezanih za sliku, uspostavljena je relacija Photo i Comment itd.



Slika 1.2 Primer konceptualnog modela [Izvor: NM IT350 -2020/2021.]

KORACI U KREIRANJU KONCEPTUALNOG MODELA

Treba odrediti tipove entiteta i definisati relacije koje između njih postoje

Glavni koraci u kreiranje konceptualnog modela koji predstavljamo E/R dijagramom su:

1. Odrediti tipove entiteta
2. Odrediti relacije između tipova entiteta
3. Profiniti definiciju tipova entiteta

Najčešći scenario je:

1. Poći od tekstualnog opisa problema
2. Analizirati ga i uvesti neophodne pretpostavke
3. Kreirati ER dijagram koji odslikava situaciju i eksplicira relacije među tipovima entiteta
4. Korak od ER dijagrama ka implementaciji baze podataka je pravolinijski
5. Kreiranje baze podataka i jeste naš glavni cilj .

Dakle, navodimo korake u kreiranju konceptualnog modela:

Identifikacija zahteva: Razgovarajte sa relevantnim zainteresovanim stranama kako biste razumeli zahteve i potrebe koje model podataka treba da zadovolji. Ovo uključuje analizu poslovnih procesa, ciljeva sistema i očekivanja korisnika.

Identifikacija entiteta: Prepoznajte glavne entitete koji će biti predstavljeni u modelu podataka. Entiteti su osnovni elementi podataka koji predstavljaju ljude, objekte, događaje ili

koncepte koje želite pratiti u bazi podataka.

Definisanje odnosa: Identifikujte veze između entiteta kako biste razumeli kako su međusobno povezani. Odnosi mogu biti jedan-na-jedan, jedan-na-više ili više-na-više.

Postavljanje ograničenja: Definišite ograničenja koja regulišu validnost podataka i integritet baze podataka. Ova ograničenja mogu uključivati obavezna polja, jedinstvene vrednosti, referencijalni integritet itd.

Razmatranje performansi: Uzmite u obzir performanse sistema prilikom kreiranja konceptualnog modela. Razmislite o potrebama za brzim pretragama, efikasnim skladištenjem podataka i optimizacijom performansi.

Dizajn atributa: Definišite attribute (svojstva) koji opisuju svaki entitet. Ovi atributi će sadržati informacije koje želite čuvati o svakom entitetu.

Provera sa zainteresovanim stranama: Proverite i potvrdite konceptualni model sa zainteresovanim stranama kako biste osigurali da on odgovara njihovim zahtevima i očekivanjima.

Dokumentacija modela: Obezbedite jasnu i sveobuhvatnu dokumentaciju konceptualnog modela kako biste omogućili lako razumevanje i korišćenje modela od strane celokupnog tima.

PROBLEMI KOJI SE MOGU JAVITI U KONCEPTUALNOM MODELIRANJU PODATAKA

U modelima se kao rezultat različitih načina na koji su objekti stvarnog sveta predstavljeni u dijagramima entiteti veze, javljaju mnoge nedoslednosti

Mnogi sistemi sadrže iste osnovne komponente, koje su redefinisane za određenu svrhu. Na primer, za sledeće komponente se može koristiti isti osnovni model:

- Katalog materijala,
- Specifikacija proizvoda i brendova,
- Specifikacije opreme.

Međutim, modeli ovih komponenti se u praksi često razvijaju iz početka jer nemamo načina da kažemo da se radi o istim stvarima.

Mnoge nedoslednosti koja se pojavljuje među modelima podataka su rezultat različitih načina na koji su objekti stvarnog sveta predstavljeni u dijagramima entiteti veze. Ako se isti koncepti modeliraju na različite načine, onda ne postoji način na koji možete očekivati da će različiti modeli istih stvari izgledati isto.

Razlike između načina na koji se stvari modeliraju uzrokovane su izgradnjom modela u koje je ugrađeno određeno stanovište ili specifična pravila i ograničenja. Pošto drugi modeli mogu imati drugačija pravila ili drugačije stanovište, dolazimo do stvari za koje znamo da ne želimo u konceptualnim modelima podataka. **Zato je potrebno shvatiti kako predstaviti svet na neutralan način, tako da su rezultujući modeli fleksibilni i stabilni.**

Osnovni razlog za ove probleme koji se javljaju u modeliranju podataka je nedostatak standarda za modeliranje podataka.

▼ Poglavlje 2

Tip entiteta, entit, atribut i identifikator

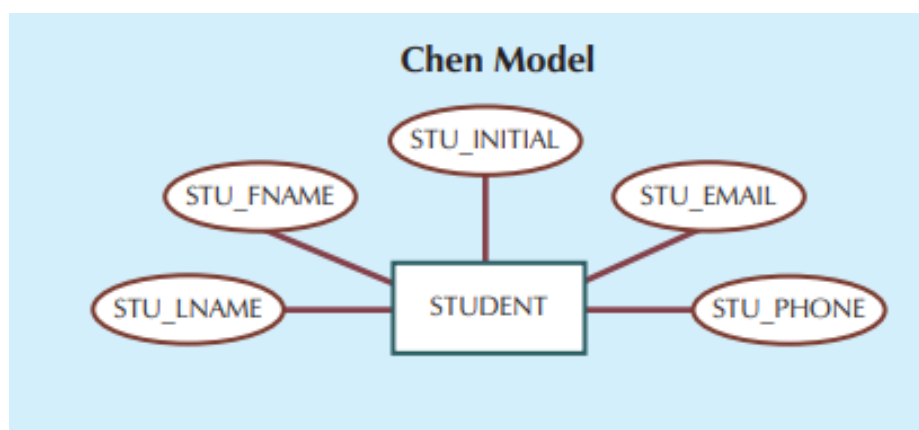
TIP ENTITETA I ENTITET

Tip entiteta je osnovni koncept ER modela i predstavlja skup objekata u stvarnom svetu koji imaju ista svojstva. Entitet je pojavljivanje (instanca) jednog tipa entiteta.

Sa gledišta obrade podataka realan svet se sastoji iz objekata posmatranja, koji mogu biti realni objekti (osoba, mašina, vozilo, kuća), predstavljati apstraktne koncepte (preduzeće, radno mesto), događaje (prekršaj, prevoz) ili odnose između objekata (student-predmet, muž-žena).

Svi takvi objekti se u konceptualnom modelu baze podataka predstavljaju tipovima entiteta. Ime entiteta, obično imenica, obično se piše velikim slovima (npr. 'OSOBA', 'TELEFON', 'ZAPOSLEN'). Može se reći da je osnovni koncept ER modela upravo tip entiteta. **Tip entiteta predstavlja skup entiteta iz stvarnog, realnog sveta koji imaju ista svojstva.** Svaki **entitet** u realnom svetu poseduje nezavisnu egzistenciju i može biti objekat sa fizičkim (stvarnim) postojanjem (osoba, automobil, kuća ili zaposleni) ili objekat sa konceptualnim (apstraktnim) postojanjem (posao, predmet koji se sluša na fakultetu, utakmica itd.) **Jedan entitet je jedan javni oblik datog tipa entiteta. Svaki entitet mora imati identitet različit od svih ostalih entiteta.**

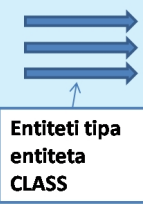
Entitet je objekat od interesa za krajnjeg korisnika. Na nivou ER modeliranja, entitet se zapravo odnosi na skup entiteta, a ne na pojedinačnu pojavu entiteta. Drugim rečima, reč "entitet" u ERM-u odgovara tabeli - ne redu - u relacijskom okruženju. ERM se odnosi na red tabele kao "instancu entiteta" ili "pojavu entiteta". Tipovi entiteta se u CASE alatima predstavljaju pravougaonikom koji sadrži ime entiteta.



Slika 2.1 Atributi entiteta STUDENT. Izvor: [1]

PRIMER: TIP ENTITETA, ENTITET

Na primeru tipa entiteta CLASS



CLASS_CODE	CRS_CODE	CLASS_SECTION	CLASS_TIME	ROOM_CODE	PROF_NUM
10012	ACCT-211	1	MMF 8:00-8:50 a.m.	BUS311	105
10013	ACCT-211	2	MMF 9:00-9:50 a.m.	BUS200	105
10014	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10015	ACCT-212	1	MMF 10:00-10:50 a.m.	BUS311	301
10016	ACCT-212	2	Th 6:00-8:40 p.m.	BUS252	301
10017	CIS-220	1	MMF 9:00-9:50 a.m.	KLR209	228
10018	CIS-220	2	MMF 9:00-9:50 a.m.	KLR211	114
10019	CIS-220	3	MMF 10:00-10:50 a.m.	KLR209	228
10020	CIS-420	1	W 6:00-8:40 p.m.	KLR209	162
10021	QM-261	1	MMF 8:00-8:50 a.m.	KLR200	114
10022	QM-261	2	TTh 1:00-2:15 p.m.	KLR200	114
10023	QM-362	1	MMF 11:00-11:50 a.m.	KLR200	162
10024	QM-362	2	TTh 2:30-3:45 p.m.	KLR200	162
10025	MATH-243	1	Th 6:00-8:40 p.m.	DRE155	325

Slika 2.2 Komponente i sadržaj tabele (entiteta) CLASS. Izvor: [Autor, 1]

Ako se CLASS_CODE sa slike 2.2 koristi kao primarni ključ, entitet CLASS može biti predstavljen u skraćenom obliku:

CLASS (**CLASS_CODE**, CRS_CODE, CLASS_SECTION, CLASS_TIME, ROOM_CODE, PROF_NUM)

ATRIBUT

Atribut predstavlja svojstvo tipa entiteta (entiteta) ili relacije.

Atribut predstavlja svojstvo tipa entiteta (entiteta) što smo videli iz prethodnih primera, a takođe može predstavljati svojstvo relacije što će se kasnije videti.

Na primer: tip entiteta Zaposleni se može opisati atributima Prezime, Ime, Adresa, Telefon itd. Atributi entiteta sadrže vrednosti koje opisuju svaki entitet.

Podaci koji se čuvaju u atributima predstavljaju najveći deo podataka smeštenih u bazi podataka.

Svaki atribut može imati jednu od skupa mogućih vrednosti koji se naziva domen atributa.

Domen atributa predstavlja skup vrednosti koje se mogu dodeliti atributu.

Na primer: domen atributa Prezime entiteta Zaposleni predstavlja skup svih prezimena koji zaposleni imaju.

Atributi se u modelima podataka predstavljaju zavisno od korišćenog CASA alata. Na primer, u PowerDesigner-u atributi se predstavljaju u okviru pravougaonika kojima su predstavljeni tipovi entiteta. Na slici 2.3 su prikazani atributi za tipove entiteta Zaposleni, Rođak i Odeljenje

ZAPOSLENI			
IDZaposlenog	<pi>	<Undefined>	<M>
Prezime		<Undefined>	
Ime		<Undefined>	
Pol		<Undefined>	
Datum Rodjenja		<Undefined>	
Adresa		<Undefined>	
Telefon		<Undefined>	
Godine Staza		<Undefined>	
Plata		<Undefined>	
Identifier_1	<pi>		

RODIAK	
Ime	<Undefined>
Adresa	<Undefined>
Telefon	<Undefined>
Rodbinska Veza	<Undefined>

ODELENJE			
IDodeljenja	<pi>	<Undefined>	<M>
Adresa		<Undefined>	
Telefon		<Undefined>	
Faks		<Undefined>	
Identifier_1	<pi>		

Slika 2.3 Način predstavljanja atributa u PowerDesigner-u [Izvor: NM IT350 -2020/2021.]

Atributi se mogu klasifikovati na:

1. **proste i složene,**
2. **sa jednom vrednošću i sa više vrednosti,**
3. **izvedene**

VRSTE ATRIBUTA

Prosti i složeni atributi, atributi sa jednom vrednošću i sa više vrednosti, izvedeni atributi.

Prost atribut se sastoji od jedne komponente koja ima nezavisnu egzistenciju. Ovakvi atributi se ne mogu dalje deliti. U literaturi se često ovakvi atributi označavaju kao atomski atributi.

Primeri prostih atributa: Pol i Plata.

Složen atribut se sastoji od više komponenti od kojih svaka ima nezavisnu egzistenciju. Ovakvi atributi se mogu dalje deliti na podkomponente.

Na primer: atribut Adresa sa vrednošću Cara Dušana 187, 11080 Zemun se može podeliti na attribute Ulica sa vrednošću Cara Dušana 187, Poštanski broj sa vrednošću 11080 i Opština sa vrednošću Zemun. Deljenje složenog atributa na proste komponente se vrši u zavisnosti od toga kako korisnik vidi atribut Adresa – kao jedinstvenu celinu ili kao pojedinačne delove.

Atribut sa jednom vrednošću sadrži jednu vrednost za jedan entitet. Većinu atributa u modelima predstavljaju atributi sa jednom vrednošću.

Napomenimo da jedno-vrednosni atribut nije nužno jednostavan atribut. Na primer, serijski broj dela, poput SE-08-02-189935, je jedno-vrednosni atribut, ali je složeni atribut jer se može podeliti na region u kojem je deo proizveden (SE), fabriku unutar tog regiona (08), smenu unutar fabrike (02) i broj dela (189935).

Atribut sa više vrednosti sadrži više vrednosti za jedan entitet.

Na primer: atribut Telefon može sadržati više vrednosti (broj kućnog telefona i broj mobilnog telefona).

Izvedeni atribut predstavlja vrednost koja je izvedena na osnovu vrednosti nekog povezanog atributa ili skupa njih, pri čemu oni ne moraju pripadati istom entitetu.

Na primer: atribut *Broj_godina* se izvodi na osnovu atributa *Datum_rođenja* te se može reći da su ova dva atributa u vezi. Atribut *Broj_godina* je izvedeni atribut jer se njegova vrednost izračunava na osnovu vrednosti atributa *Datum_rođenja*.

OPCIONOST ATRIBUTA

Opcionost atributa zavisi od poslovne situacije

Nemaju svi entiteti vrednost za svaki atribut; međutim, neki atributi moraju da imaju vrednost za sve entitete. Zato se atributi mogu klasifikovati u

- **opcione i**
- **obavezne**

Na primer, tip entiteta *Zaposleni* ima attribute *datum zaposlenja* i *datum prestanka*.

- *Datum zaposlenja* je svakako obavezni atribut jer ako nema datuma zaposlenja, zaposleni ne može biti zaposlen.
- *Datum prestanka* je opciono atribut. Za očekivanje je da mnogi zaposleni u bazi podataka nemaju datum prestanka dok drugi, bivši zaposleni, imaju vrednost tog atributa.

Opcionost atributa zavisi od poslovne situacije, od toga kako se informacija prikupljai kako poslovanje ažurira svoju bazu podataka. Isti atribut može da bude različito klasifikovan od strane raznih kompanija.

Primer: Tabela: PROIZVOD

ProizvodID (Primarni ključ)

Naziv (Obavezno polje)

Cena (Obavezno polje)

Opis (Opciono polje)

Kategorija (Opciono polje)

Boja (Opciono polje)

U ovom primeru, polja "Opis", "Kategorija" i "Boja" su opciono atributi. To znači da ne moramo uneti podatke za ova polja za svaki proizvod u tabeli. Na primer, možda neki proizvodi nemaju poseban opis, nisu kategorisani ili nemaju određenu boju koju treba navesti. Međutim, svi proizvodi moraju imati unet naziv i cenu, što su obavezni atributi.

Korišćenje opcionih atributa nam omogućava da se prilagodimo različitim situacijama i omogućava nam da unesemo samo relevantne informacije za svaki red u tabeli, bez potrebe da popunjavamo svako polje za svaki unos. Ovo čini bazu podataka fleksibilnijom i efikasnijom u upravljanju raznovrsnim podacima.

IDENTIFIKATOR ENTITETA

Predstavlja attribute koji identifikuju neki entitet (instancu tipa entiteta)

Identifikator entiteta, takođe poznat kao **ključ entiteta**, je atribut ili skup atributa koji jedinstveno identifikuje svaki entitet u modelu podataka. U kontekstu baze podataka, identifikator entiteta je ključ koji jednoznačno razlikuje jedan entitet od drugog.

Postoji nekoliko tipova identifikatora entiteta:

Primarni ključ (eng. Primary Key): Ovo je glavni identifikator entiteta u modelu podataka. Primarni ključ obezbeđuje jedinstvenost i jednoznačno razlikuje svaki entitet u tabeli.

Alternativni ključ (eng. Alternate Key): Ovo su dodatni ključevi koji takođe obezbeđuju jedinstvenost, ali se koriste kao alternativa primarnom ključu.

Kandidatski ključ (eng. Candidate Key): To su svi mogući ključevi koji jednoznačno identifikuju entitet, uključujući i primarni i alternativne ključeve.

Identifikator entiteta je od ključne važnosti za modeliranje podataka, jer omogućava precizno identifikovanje i upravljanje entitetima u bazi podataka. Pravilno definisan identifikator entiteta obezbeđuje doslednost, integritet i efikasnost u radu sa bazom podataka.

Zadatak:

Modelirajte entitet "Knjiga" sa sledećim atributima i odredite primarni ključ:

ISBN - Unikatan međunarodni broj knjige.

Naslov - Ime knjige.

Autor - Ime autora knjige.

Godina izdanja - Godina kada je knjiga izdata.

Izdavačka kuća - Naziv izdavačke kuće koja je objavila knjigu.

Broj stranica - Ukupan broj stranica u knjizi.

Cena - Cena knjige.

▼ Poglavlje 3

Relacije

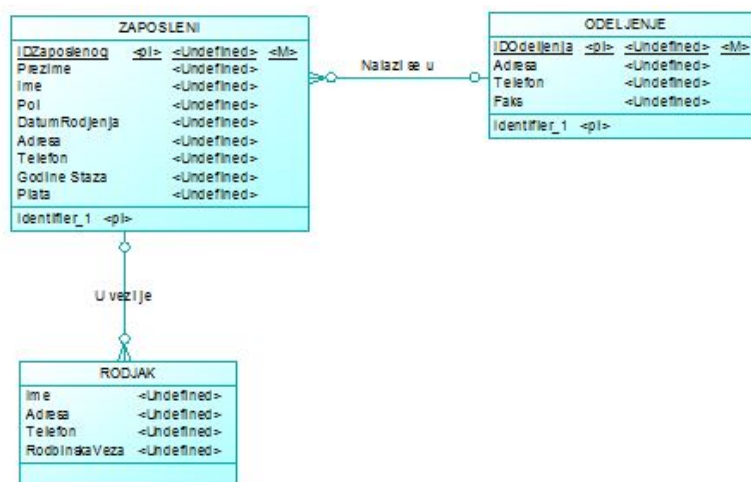
RELACIJA

Relacija (veza) predstavlja asocijacije koje se mogu uspostaviti između tipova entiteta.

Relacija (veza) predstavlja asocijacije koje se mogu uspostaviti između tipova entiteta. Svakoj relaciji se zadaje naziv koji opisuje njenu funkciju.

Na primer: tip entitet Zaposleni je u relaciji sa tipom entitetom Rodak preko relacije U_vezi_sa.

U notaciji koja se koristi u PowerDesigner-u, svaka relacija se grafički predstavlja linijom uz koju stoji naziv relacije. Na slici 3.1 prikazane su relacije između tipova entiteta Zaposleni, Rodak i Odeljenje.



Slika 3.1 Predstavljanje binarne relacija između entiteta [Izvor: NM IT350 -2020/2021.]

Pri definisanju relacija između tipova entiteta potrebno je razmotriti ograničenja koja mogu da postoje u asocijacijama uspostavljenim između njih. Ograničenja treba da budu u skladu sa ograničenjima koja postoje u realnom svetu.

Kardinalnost relacije (eng. *cardinality*) se definiše kao broj pojavljivanja jednog tipa entiteta koji se mogu povezati sa jednim pojavljivanjem drugog tipa entiteta. U svakoj relaciji se određuje maksimalna kardinalnost i minimalna kardinalnost. U ER modelu se relacije klasifikuju prema svojoj kardinalnosti.

KARDINALNOST RELACIJE

Kardinalnost se definiše kao broj entiteta jednog tipa entiteta koji se mogu povezati sa jednim entitetom drugog tipa entiteta.

Da bi definisali kardinalnost relacije treba razmotriti maksimalni broj entiteta jednog tipa koji se mogu pridružiti jednom entitetu drugog tipa.

Maksimalnu kardinalnost relacije treba definisati tako što za svaki par tipova entiteta treba odgovoriti na 2 pitanja:

1. Za svaki entitet tipa Y može da bude najviše ____ entiteta tipa X.
2. Za svaki entitet tipa X može da bude najviše ____ entiteta tipa Y.

(ako je odgovor na prvo pitanje, na primer, M a na drugo 1, onda je kardinalnost relacije između tipova entiteta X i Y M:1 (čita se: za svaki X može biti samo jedan Y i za svaki Y može biti više X.”)

Zadatak 1: odrediti maksimalnu kardinalnost sledećih relacija

1. Pacijent pod praćenjem lekara opšte prakse
2. Lekar vrši operaciju
3. Lekar ima specijalnost za bolest

Minimalnu kardinalnost relacije definišemo na osnovu postojanja tipa entiteta na drugoj strani relacije, i zadaje se kao opciona, obavezna ili nepoznata.

Veza opisuje povezanost između entiteta. Na primer, postoji veza između kupaca i agenata koja se može opisati na sledeći način: jedan agent može služiti mnogo kupaca, i svaki kupac može biti služen od strane jednog agenta. Modeli podataka koriste tri vrste veza: jedan prema više, više prema više i jedan prema jedan.

MAKSIMALNA KARDINALNOST JEDAN-PREMA-JEDAN

Maksimalna kardinalnost može biti: jedan-prema-jedan, jedan-prema-više i više-prema-više. U 1:1 relaciji, instanca entiteta jednog tipa se odnosi na bar jednu instancu entiteta drugog tipa.

Tri najčešće maksimalne kardinalnosti su tipa:

1. **Jedan-prema-jedan (1:1 ili 1..1) odnos.** Upravljačka struktura maloprodajne kompanije može zahtevati da svaka od njenih prodavnica bude upravljana od strane jednog zaposlenog. S druge strane, svaki menadžer prodavnice, koji je takođe zaposleni, upravlja samo jednom prodavnicom. Zbog toga je odnos "ZAPOSLENI upravlja PRODAVNICOM" označen kao 1:1.
2. **Jedan-prema-više (1:M ili 1..*) odnos.** Jedan slikar slika mnogo različitih slika, ali svaku od njih slika samo jedan slikar. Zbog toga je slikar (jedan) povezan sa

slikama (više). Zbog toga, dizajneri baza podataka označavaju odnos "SLIKAR slika SLIKU" kao 1:M. (Primetite da su nazivi entiteta često napisani velikim slovima kako bi ih lako prepoznali.) Slično, kupac (jedan) može generisati više faktura, ali svaka faktura (više) je generisana samo od strane jednog kupca. Odgovarajući odnos "KUPAC generiše FAKTURU" takođe bi bio označen kao 1:M.

3. Više-prema-više (M:N ili ..) odnos. Jedan zaposleni može naučiti više veština posla, a svaku veštinu posla može naučiti više zaposlenih. Zbog toga dizajneri baza podataka označavaju odnos "ZAPOSLENI uči VEŠTINU" kao M:N. Slično, jedan student može pohađati više predmeta, a svaki predmet može biti pohađan od strane više studenata, što rezultira odnosom M:N koji se izražava odnosom "STUDENT pohađa PREDMET".

Maksimalna kardinalnost tipa jedan-prema-jedan: U 1:1 relaciji, instanca entiteta jednog tipa se odnosi na bar jednu instancu entiteta drugog tipa.

MAKSIMALNA KARDINALNOST JEDAN-PREMA-VIŠE

Kardinalnosti 1:M - instanca jednog tipa entiteta je u vezi sa više instanci drugog tipa entiteta.

Maksimalna kardinalnost tipa jedan-prema-više: U ovom slučaju jedna instanca jednog tipa entiteta može biti u vezi sa više instanci drugog tipa entiteta.

Na primer: Na slici 3.2 prikazana je binarna relacija Nadgleda koja povezuje entitete Zaposleni i Nekretnina. Na strani entiteta Zaposleni maksimalna kardinalnost je 1 (označena vertikalnom linijom), a na strani entiteta Nekretnina maksimalna kardinalnost je više (označena simbolom u obliku rakljice). Ovo znači da, kada se gleda sa strane entiteta Zaposleni, jedan zaposleni može da nadgleda više nekretnina (kardinalnost je jedan-prema-više) a posmatrano sa strane entiteta Nekretnina, jednu nekretninu nadgleda najviše jedan zaposleni (kardinalnost je jedan-prema-jedan).



Slika 3.2 Maksimalna kardinalnost tipa jedan-prema-više [Izvor: NM IT350 -2020/2021.]

Kada se govori o relacijama jedan-prema-više, često se koristi termin roditelj i dete. **Tip entiteta roditelj** je entitet sa strane jedan relacije a **tip entiteta dete** je entitet sa strane relacije više.

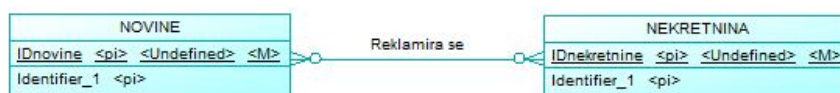
Na primer: U relaciji na slici 3.2 tip entiteta Zaposleni je **roditelj**, a tip entiteta Nekretnina **jedete**.

MAKSIMALNA KARDINALNOST VIŠE-PREMA-VIŠE

Kardinalnosti M:M- više instanci jednog tipa entiteta je u vezi sa više instanci drugog tipa entite

Maksimalna kardinalnost tipa više-prema-više: U ovom slučaju jedna instanca jednog tipa entiteta može biti u vezi sa više instanci drugog tipa entiteta ali i jedna instanca drugog tipa entiteta može biti u vezi sa više instanci prvog tipa entiteta.

Na primer: Razmotrimo binarnu relaciju Reklamira prikazanu na slici 3.3. Na strani entiteta Novine maksimalna kardinalnost je više (označena simbolom u obliku rakljice), a na strani tipa entiteta Nekretnina je takođe više (takođe označena simbolom u obliku rakljice). Ovo znači da, kada se gleda od strane entiteta Novine, jedne novine mogu da reklamiraju više nekretnina (kardinalnost je jedan-prema-više) a posmatrano sa strane entiteta Nekretnina, jednu nekretninu mogu takođe da reklamiraju više novina (kardinalnost je jedan-prema-više).



Slika 3.3 Relacija Reklamira se čija je maksimalna kardinalnost više-prema-više [Izvor: NM IT350 -2020/2021.]

MINIMALNA KARDINALNOST

Minimalna kardinalnost je broj instanci entiteta koji mora učestvovati u relaciji.

Minimalna kardinalnost je broj instanci tipa entiteta koji mora učestvovati u relaciji.

Postoje tri tipične minimalne kardinalnosti:

0 (nula) - Nula minimalna kardinalnost znači da nije obavezno da postoji povezanost između entiteta u odnosu. Entitet može biti povezan sa nula drugih entiteta.

1 (jedan) - Minimalna kardinalnost jedan znači da je veza obavezna za svaki entitet u odnosu. Svaki entitet mora biti povezan sa tačno jednim drugim entitetom.

N (više) - Minimalna kardinalnostviše znači da veza može biti uspostavljena sa bilo kojim brojem entiteta, uključujući nula. Ovo označava da veza nije obavezna za svaki entitet.

Na primer, ako imamo entitete "Student" i "Predmet" povezane relacijom "STUDENT pohađa PREDMET", minimalna kardinalnost može biti:

Minimalna kardinalnost 0: Studenti mogu ne pohađati nijedan predmet, ili može postojati student koji još uvek nije povezan ni sa jednim predmetom.

Minimalna kardinalnost 1: Svaki student mora pohađati tačno jedan predmet.

Minimalna kardinalnost N: Svaki student može pohađati bilo koji broj predmeta, uključujući nula.

STEPEN RELACIJE

Određuje broj učesnika u relaciji

Tipovi entiteti koji formiraju relacije se označavaju kao učesnici u relaciji. Broj učesnika u relaciji naziva se stepen relacije. Prema tome, stepen relacije **predstavlja broj tipova entiteta koji učestvuju u relaciji**. Relacija čiji je stepen dva naziva se binarna relacija.

Relacije kao atributi. Ponekad je zgodno o binarnom tipu relacije razmišljati u smislu atributa. Razmotrite relaciju WORKS_FOR.

U tip entiteta EMPLOYEE može se razmišljati o dodavanju atributa koji se zove Odeljenje (Department), gde je vrednost odeljenja za svaki entitet employee (referenca) na odeljenje za koje je zaposleni radi. Dakle, vrednost ovog atributa Odeljenje je skup svih entiteta DEPARTMENT.

Međutim, kada mislimo o binarnom odnosu kao atributu, uvek imamo dve opcije. U ovom primeru, alternativa je razmišljati o višestrukom atributu Employee u tipu entiteta DEPARTMENT čija vrednosti za svaki entitet DEPARTMENT je skup entiteta Employee koji rade u tom odeljenju ili o atributu Department u tipu entiteta EMPLOYEE. Bilo koji od ova dva atributa – Department u EMPLOYEE ili Employee u DEPARTMENT - može predstavljati tip odnosa WORKS_FOR.

Dakle, postoje tri uobičajena tipa stepena relacije:

Unarna relacija (stepen1): Ovde postoji samo jedan entitet koji je povezan sam sa sobom. Na primer, "Osoba" može biti u relaciji "rođeni su u istom gradu" gde je samo jedan entitet "Osoba" uključen.

Binarna relacija (stepen2): Ovde postoje dva entiteta koji su međusobno povezani. To je najčešći tip relacije u kojem učestvuju dva entiteta. Na primer, "Student" može biti u relaciji "pohađa" sa entitetom "Predmet".

Ternarna relacija (stepen3): Ovde postoje tri entiteta koji su međusobno povezani. Ternarne relacije su retke, ali se mogu pojaviti u određenim situacijama. Na primer, u bolnici, "Pacijent", "Lekar" i "Soba" mogu biti u relaciji "je smešten u" kako bi se označilo da je određeni pacijent smešten u određenu sobu pod nadzorom određenog lekara.

Opšti oblik stepena relacije može biti viši od 3, ali obično se koriste unarna, binarna i ternarna relacija jer pokrivaju većinu situacija u bazi podataka.

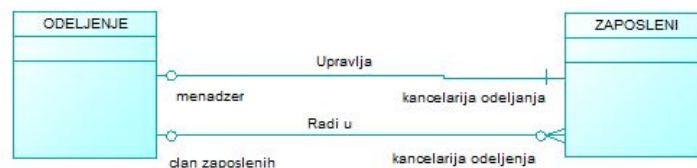
ULOGA

Relacijama se mogu zadati nazivi uloga (role names) da bi se prikazala uloga koji ima svaki učesnik relacije.

Svaki tip entiteta koji učestvuje u relaciji ima posebnu ulogu u toj relaciji. **Ime uloge** (engl. *role*) označava ulogu koju entitet iz datog tipa entiteta igra u relaciji i pomaže da se objasni šta znači relacija.

Na primer, u relaciji RAD_ZA, ZAPOSLENI igra ulogu zaposlenog ili radnika a ODELJENJE igra ulogu odeljenja ili poslodavca.

Imena uloga nisu tehnički neophodna u relacijama gde su svi tipovi entiteta koji u njoj učestvuju različiti, jer svako ime tipa entiteta može da se koristi kao ime uloge. Međutim, u nekim slučajevima isti tip entiteta učestvuje više puta u relaciji sa različitim ulogama. U takvim slučajevima **ime uloge postaje bitno za razlikovanje značenja uloge koju svaki entitet ima. To se najčešće dešava u slučaju rekurzivnih relacija.**



Slika 3.4 Zaposleni i odeljenje povezani sa dve relacije [Izvor: NM IT350 -2020/2021.]

Uloge se mogu koristiti i kada su dva entiteta povezana sa više relacija.

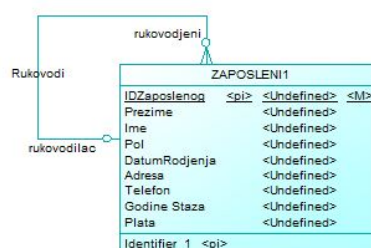
Na primer: na slici 3.4 prikazana je situacija u kojoj su entiteti Zaposleni i Odeljenje povezani sa dve relacije nazvane Upravlja i Radi_u. Namena uloga je da pojasni svaku relaciju. Na primer, u slučaju relacije Upravlja, jedan od zaposlenih kojem je data uloga Menadžer upravlja odeljenjem kojem je data uloga Kancelarija odeljenja.

REKURZIVNA RELACIJA

Relacija u kojoj jedan isti entitet učestvuje više puta u relaciji sa različitim ulogama.

Rekurzivna relacija je relacija u kojoj jedan isti entitet učestvuje više puta sa različitim ulogama. Ovakve relacije se nekada označavaju i ako unarne relacije.

Primer: Razmotrimo relaciju Rukovodi prikazanu na slici 3.5 na kojoj je prikazana veza zaposlenog sa rukovodiocem koji je i sam zaposleni. Drugim rečima, entitet tipa entiteta ZAPOSLENI dva puta učestvuje u relaciji – prvi put kao rukovodilac, a drugi put kao zaposleni.



Slika 3.5 Primer rekurzivne relacije [Izvor: NM IT350 -2020/2021.]

U ovom slučaju treba čitati dijagram:

- 1. u smeru kazaljke na satu: jedan zaposleni možda ne upravlja nijednim zaposlenim ali može da upravlja i većim brojem zaposlenih.**
- 2. u smeru suprotnom smeru kazaljke na satu: svakim zaposlenim ne mora da upravlja drugi zaposleni (nema svako u kompaniji direktora: predsednik kompanije nema svog direktora).**

Dajemo još jedan primer:

Entitet "Kategorija" može biti u rekurzivnoj relaciji "pripada" sa drugim entitetima "Kategorija", gde jedna kategorija može pripadati drugoj kategoriji kao nadređenoj. Ova struktura se često koristi za modeliranje hijerarhija kategorija, kao što su organizacija proizvoda, klasifikacija dokumenata ili organizacija poslovnih podataka.

Na primer, "Elektronika" može biti nadređena kategoriji "Računari i Laptopovi", koja može biti nadređena kategoriji "Laptopovi", koja dalje može biti nadređena kategoriji "15-inčni Laptopovi", i tako dalje.

Kroz ovakvu rekurzivnu relaciju, možemo lako organizovati i upravljati sa složenim strukturama i odnosima između kategorija koje mogu imati više nivoa hijerarhije.

Evo još jednog primera rekurzivne relacije:

Entitet "Organizacija" može biti u rekurzivnoj relaciji "sadrži" sa drugim entitetima "Organizacija". Ovo može predstavljati strukturu organizacija sa nadređenim i podređenim organizacijama. Na primer, "Organizacija A" može sadržati "Organizaciju B", dok "Organizacija B" može sadržati "Organizaciju C", i tako dalje.

▼ Poglavlje 4

Verzije ER dijagrama

RAZLIČITE VERZIJA ER DIJAGRAMA

Originalni Chen-ov model, modeli koji se koriste u CASE alatu ERWIN i VISIO.

U originalnom, Chen-ovom ER dijagramu je definisano da se za predstavljanje relacija koristi romb, tipova entiteta - pravougaonik a atributa - elipse koje su povezane sa odgovarajućim pravougaonikom. Na slici 4.1 je dat jedan primer Chen-ovog ER modela na kojem su predstavljeni tipovi entiteta EMPLOYEE , DEPARTMENT i DEPENDENT sa odgovarajućim atributima kao i relacije WORKS_ON, MANAGES, WORKS_FOR, CONTROLS, DEPENDENS_OF i SUPERVISION.

Međutim, ovaj originalni model se danas retko koristi iz dva razloga. Prvo, postoji mnogo novijih verzija ER modela i te verzije koriste drugačije simbole. Drugo, softverski proizvodi za modeliranje podataka (CASE alati) koriste različite tehnike.

U svim verzijama ER dijagrama se za predstavljanje tipova entiteta koristi pravougaonik. Međutim, koriste se različite notacije za predstavljanje relacija.

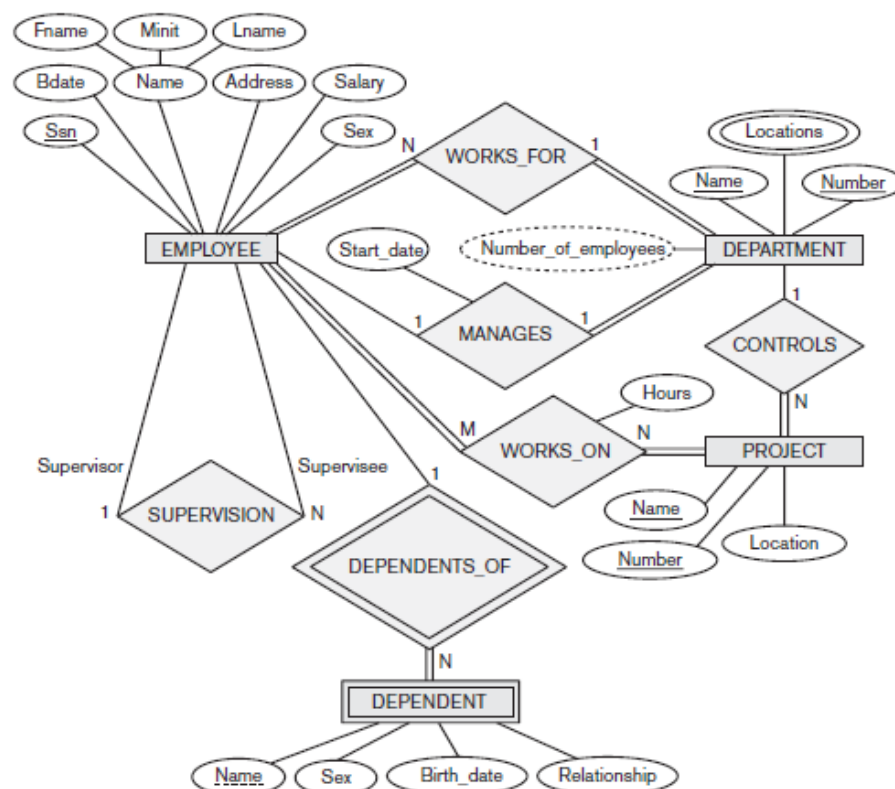
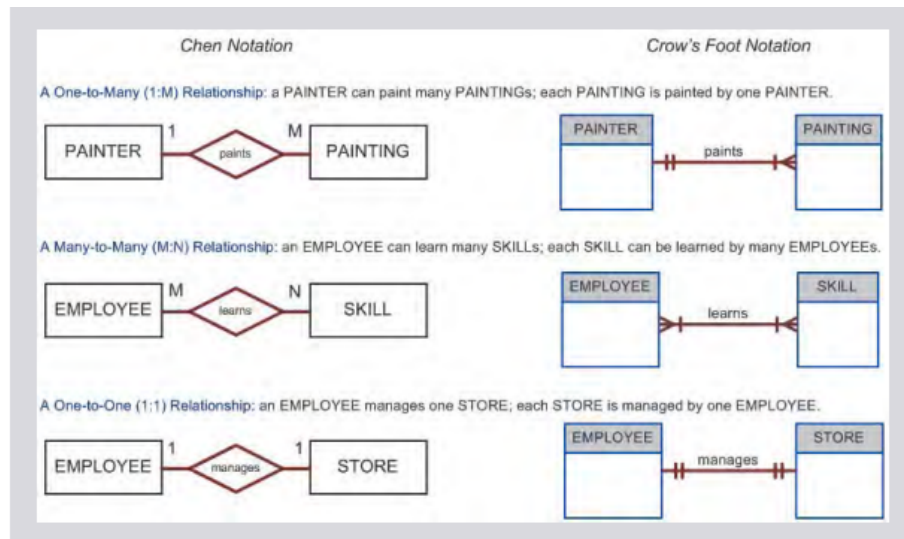


Figure 3.1

Slika 4.1 Originalni Chen-ov ER dijagram [Izvor: NM IT350 -2020/2021.]

NOTACIJE ZA PREDSTAVLJANJE RELACIJA 1:M I M:N

Prezentacija zavisi od CASE alata u kojem se veza uspostavlja.



Slika 4.2 Chen i Crow's Foot notacije. Izvor: [1]

Slika 4.2 prikazuje različite vrste veza koristeći dve ER notacije: originalnu Chen notaciju i savremeniju Crow's Foot notaciju.

Leva strana ER dijagrama prikazuje Chen notaciju. U ovoj notaciji, povezanosti se pišu pored svake oznake entiteta. Veze su predstavljene romboidom povezanim sa povezanim entitetima preko linije veze. Ime veze je napisano unutar romboida.

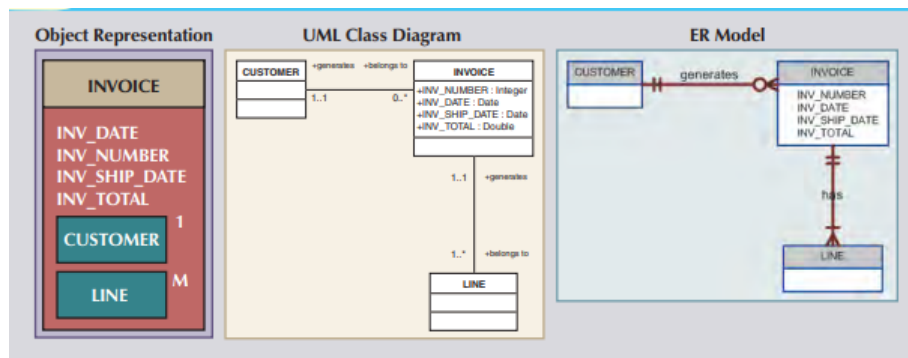
Desna strana slike 4.2 prikazuje Crow's Foot notaciju. Naziv "Crow's Foot" potiče od simbola sa tri prstena koji se koristi za predstavljanje "više" strane veze. Kada proučavate osnovni Crow's Foot ERD na slici 4.2, obratite pažnju da su povezanosti predstavljene simbolima. Na primer, "1" je predstavljeno kratkim segmentom linije, a "M" je predstavljeno sa tri prstena "crow's foot". U ovom primeru, ime veze je napisano iznad linije veze.

Na slici 4.2, entiteti i veze su prikazani u horizontalnom formatu, ali takođe mogu biti orijentisani vertikalno. Lokacija entiteta i redosled u kojem su predstavljeni nisu bitni; samo zapamtite da se 1:M veza čita od "1" strane ka "M" strani.

Chen notacija se koristi za ilustraciju nekih koncepata ER modeliranja kad god je to potrebno. Većina alata za modeliranje podataka vam omogućava da izaberete Crow's Foot notaciju.

POREĐENJE OO, UML I ER MODELA

Dato je upoređivanje tri modela



Slika 4.3 Uspoređivanje OO, UML i ER modela. Izvor: [1]

Razmotrimo sliku 4.3:

Objektno predstavljanje RAČUNA (INVOICE) uključuje sve povezane objekte unutar istog okvira objekta. Obratite pažnju da veze (1 i M) označavaju odnos povezanih objekata sa RAČUNOM. Na primer, broj 1 pored objekta KUPAC (CUSTOMER) ukazuje da je svaki RAČUN povezan samo sa jednim KUPCEM. Broj M pored objekta STAVKA (LINE) ukazuje da svaki RAČUN sadrži mnogo STAVKI.

UML dijagram klase koristi tri odvojene klase objekata (KUPAC, RAČUN i STAVKA) i dve veze da predstavi ovaj jednostavni problem fakturisanja. Obratite pažnju da se veze označavaju simbolima 1..1, 0..* i 1..*, i da su veze nazvane na oba kraja kako bi se predstavile različite "uloge" koje objekti igraju u odnosu.

ER model takođe koristi tri odvojena entiteta i dve veze da predstavi ovaj jednostavni problem fakturisanja.

Napredak u OODM (Objektno-Orijentisanih Modela Podataka) osećao se u mnogim oblastima, od modeliranja sistema do programiranja. Dodata semantika OODM omogućila je bogatiju reprezentaciju složenih objekata, što je zauzvrat omogućilo aplikacijama da podrže sve kompleksnije objekte na inovativne načine.

▼ Poglavlje 5

Poslovna pravila

ŠTA PRETSTAVLJAJU POSLOVNA PRAVILA?

Specifikacije koje čuvaju integritet lokalnog modela podataka.

Poslovna pravila su specifikacije koje čuvaju integritet lokalnog modela podataka. Četiri osnovna poslovna pravila se odnose na:

1. **Integritet entiteta**: Svaka instanca jednog tipa entiteta mora da ima jedinstveni identifikator koji nije null.
2. **Ograničenja referencijalnog integriteta**: Pravila se odnose na veze između tipova entiteta.
3. **Domene**: Ograničenja na validne vrednosti atributa
4. **Poslovna pravila koja štite validnost vrednosti atributa**

Generalno, poslovna pravila se skupljaju za vreme utvrđivanja zahteva i pamte se u CASE repozitorijumu gde se dokumentuju.

Integritetom entiteta se definiše nemogućnost postajanja dva primerka entiteta u istom tipu entiteta koja imaju istu vrednost atributa koji čine identifikator.

Referencijalni integritet definiše da vrednost sekundarnog ključa u jednoj tabeli mora odgovarati vrednosti primarnog ključa u drugoj tabeli čijim je spuštanjem nastao taj sekundarni ključ.

Poslovna pravila postavljaju osnovu za pravilno identifikovanje entiteta, atributa, veza i ograničenja. U stvarnom svetu, imena se koriste za identifikaciju objekata. Ako poslovno okruženje želi pratiti te objekte, postojaće posebna poslovna pravila za njih.

Pravila koja se odnose na domene i trigere su ilustrovana jednostavnim primerom iz oblasti bankarstva, koji je prikazan na slici 5. 1 .

U ovom primeru RACUN je entitet koji ima vezu (Je_za) sa entitom ODJAVA.

Na slici 5.1(a) su naznačeni atributi tipa entiteta RACUN i za svaki od njih specificirana ograničenja na njihove validne vrednosti (domeni). Na slici 5.1(b) je definisao poslovno pravilo koje štiti validnost vrednosti atributa u slučaju dodavanja novog entiteta tipu entiteta ODJAVA. Ovo poslovno pravilo je definisano u obliku trigera koji se vezuje za naredbu insert nad tabelom ODJAVA.

(a) Tipične definicije domena:	
Ime: Racun_Broj	Ime: Iznos
Značenje: Broj kupčevog računa u banci	Značenje: Dolarski iznos transakcije
Tip podatka: Karakter	Tip podatka: Numerik
Format: nnn-nnnn	Format: 2 decimalna mesta
Jedinstvenost: mora da bude jedinstven	Opseg: 0-10,000
Podrška null vrednostima: Not-null	Jedinstvenost: nije jedinstven
	Podrška null vrednostima: Not-null
(b) Tipična operacija za izvršenje trigera:	
Korisničko pravilo: Iznos iz ODJAVA ne sme da prekorači Stanje iz RACUN	
Događaj: Insert	
Ime entiteta: ODJAVA	
Uslov: ODJAVA Iznos > RACUN Stanje	
Akcija: Odbaci transakciju insertovanja	

Slika 5.1 Primer poslovnih pravila [Izvor: NM IT350 -2020/2021.]

DOMENI I TRIGERI

Domen je skup svih tipova podataka i opsega vrednosti koje mogu da imaju atributi.

Definicije domena obično specificiraju neke (ili sve) karakteristike atributa: tip podataka, dužina, format, opseg, dozvoljene vrednosti, značenje, jedinstvenost i null vrednosti.

Slika 1. iz prethodne sekcije (pod stavkom a.) prikazuje dve definicije domena za primer iz bankarstva.

1. **Racun_Broj:** Pošto je broj računa atribut identifikator, definicijom je specificirano da *Racun_broj* mora da bude jedinstven i da ne može da ima null vrednost. Definicija specificira da je tip atributa karakter i da je format nnn-nnnn. Svaki pokušaj da se unese vrednost ovog atributa koja ne odgovara tipu karakter ili formatu, se odbacuje i pojavljuje se greška.
2. **Iznos** (dolarski iznos tražene odjave): može biti null, ali njegova vrednost nije jedinstvena. Format dozvoljava dva decimalna mesta za smeštaj. Opseg vrednosti ima donju granicu nula (vrednost ne može biti negativna) i gornju granicu 10,000.

Domeni se mogu unapred pre definisati a zatim se nekim atributima iz modela podataka mogu pridružiti pre definisani domeni.

Na primer: Zajednički domeni kao što su *Datum*, *Socijalni broj*, *Broj telefona* treba da budu samo jedanput definisani u modelu.

Trigeri su pravila pod kojim se izvršavaju operacije za manipulaciju podacima kao što su: insert, update i delete. Operacije za izvršenje trigera mogu biti ograničene na attribute jednog entiteta ili attribute dva ili više entiteta. Operacije za izvršenje trigera obično imaju sledeće komponente:

1. Poslovno pravilo: koncizna rečenica kojom se opisuje poslovno pravilo koje treba da bude izvršeno operacijom;
2. Događaj: operacija koja se inicira (insert, delete ili update);
3. Ime tipa entiteta kojem se pristupa ili se modifikuje;
4. Uslov: pod kojim se operacija izvršava;

5. Akcija: koja se izvršava kada se operacija trigeruje.

Primer: Slika 5.1 (b) prikazuje jedan primer operacije za izvršenje trigera za bankarski sistem.

1. Poslovno pravilo: iznos koji se odjavljuje ne sme da bude veći od postojećeg stanja na računu.
2. Događaj je insertovanje instance u tip entiteta ODJAVA.
3. Ime tipa entiteta: ODJAVA
4. Uslov je: Iznos (iz ODJAVA) \leq Iznos (Iz RACUN).
5. Akcija: ako uslov nije zadovoljen odbacuje se transakcija insertovanja.

▼ Poglavlje 6

Primeri kreiranja konceptualnog modela

KORACI U KREIRANJU KONCEPTUALNOG MODELA PODATAKA

Osnovni koraci u procesu kreiranja konceptualnog modela podataka

Kreiranje konceptualnog modela podataka uključuje nekoliko koraka kako bi se jasno definisali entiteti, veze i atributi u okviru modela podataka. Evo osnovnih koraka u procesu kreiranja konceptualnog modela podataka:

Razumevanje poslovnih zahteva: Prvi korak je duboko razumevanje poslovnih zahteva organizacije ili sistema za koji se model kreira. Razgovarajte sa relevantnim zainteresovanim stranama kako biste jasno identifikovali šta je suština sistema i koje informacije se moraju zabeležiti.

Identifikacija entiteta: Na osnovu razgovora sa zainteresovanim stranama identifikujte glavne entitete (objekte) koji čine osnovu modela podataka. Entiteti bi trebali predstavljati ključne stvari o kojima se čuvaju podaci u sistemu.

Definisanje atributa: Za svaki identifikovani entitet, definišite njegove karakteristike koje se nazivaju atributi. Atributi opisuju svojstva entiteta i predstavljaju informacije koje će biti čuvane o svakom pojedinačnom entitetu.

Identifikacija veza: Razmislite o vezama između identifikovanih entiteta. Veze predstavljaju odnose i interakcije između različitih entiteta. Definišite kako entiteti međusobno komuniciraju i kako se međusobno povezuju.

Normalizacija : Ovo je proces organizovanja atributa entiteta u smislene i logične grupe kako bi se uklonile redundancije i održala konzistentnost podataka. Normalizacija pomaže da se poboljša struktura podatkovnog modela.

Revizija i validacija: Nakon što ste definisali konceptualni model podataka, pregledajte ga i proverite da li odgovara poslovnim zahtevima i da li je logički ispravan. Povratne informacije od relevantnih strana mogu biti od suštinskog značaja u ovom koraku.

Dokumentacija: Važno je detaljno dokumentovati konceptualni model podataka kako bi svi uključeni u projekt razumeli strukturu podataka i njegove karakteristike.

Napomena: *Konceptualni model podataka predstavlja apstraktnu i nezavisnu reprezentaciju podataka, fokusira se na šta podaci predstavljaju, a ne na kako će biti implementirani u bazu*

podataka ili aplikaciju. Kasnije, na osnovu konceptualnog modela, može se kreirati logički i fizički model podataka koji će se koristiti u implementaciji sistema.

PRIMER KONCEPTUALNOG MODELA ZA SISTEM ZA UPRAVLJANJE BIBLIOTEKOM

Jednostavan primer konceptualnog modela

Razmotrimo konkretan primer kreiranja konceptualnog modela podataka za sistem za upravljanje bibliotekom, slika 6.1.

Razumevanje poslovnih zahteva: Cilj je da se razvije sistem za upravljanje bibliotekom gde će biti moguće pratiti knjige, članove biblioteke, iznajmljivanje knjiga i slično.

Identifikacija entiteta i definisanje atributa: Identifikovani entiteti mogu biti:

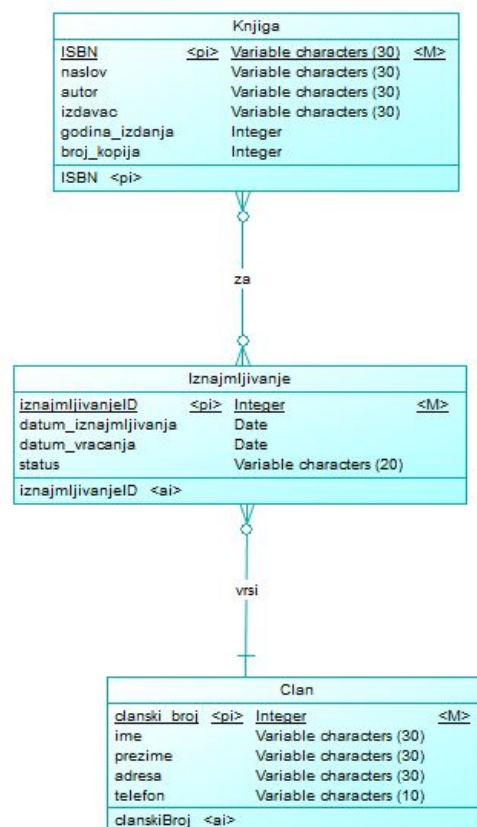
Knjiga (s atributima: ISBN, naslov, autor, izdavač, itd.)

Član biblioteke (s atributima: članski broj, ime, prezime, adresa, itd.)

Iznajmljivanje (s atributima: datum iznajmljivanja, datum vraćanja, status, itd.)

Identifikacija veza: Veza "Iznajmljivanje" može povezivati entitete "Knjiga" i "Član biblioteke" i imati attribute kao što su datum iznajmljivanja i datum vraćanja.

Normalizacija: Podaci mogu biti normalizovani kako bi se eliminisali višestruki unosi i redundancija. Na primer, adresa člana biblioteke može biti izdvojena u poseban entitet "Adresa" kako bi se izbeglo ponavljanje podataka za više članova sa istom adresom.



Slika 6.1 Konceptualni model - upravljanje bibliotekom. Izvor: [Autor]

PRIMER KREIRANJA KONCEPTUALNOG MODELA PODATAKA ZA SISTEM ZA UPRAVLJANJE ZAPOSLENIMA U KOMPANIJU

Dat je još jedan primer konceptualnog modela podataka

Razmotrimo još jedan primer kreiranja konceptualnog modela podataka za sistem za upravljanje zaposlenima u kompaniji, slika 6.2.

Razumevanje poslovnih zahteva: Cilj je da se razvije sistem koji će omogućiti praćenje informacija o zaposlenima, njihovim odeljenjima, pozicijama i platama.

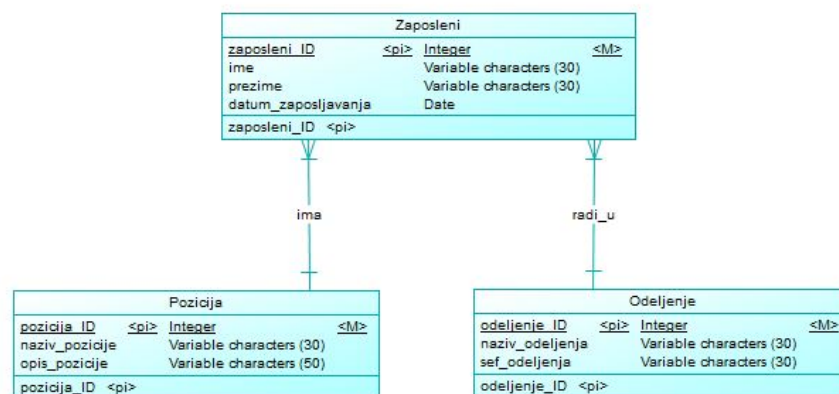
Identifikacija entiteta i definisanje atributa: Identifikovani entiteti mogu biti:

Zaposleni (s atributima: ID zaposlenog, ime, prezime, datum zapošljavanja, itd.)

Odeljenje (s atributima: ID odeljenja, naziv odeljenja, šef odeljenja, itd.)

Pozicija (s atributima: ID pozicije, naziv pozicije, opis pozicije, itd.)

Identifikacija veza: Veze će povezivati ove entitete. Na primer, veza između "Zaposleni" i "Odeljenje" će označavati da zaposleni može biti deo određenog odeljenja, dok će veza između "Zaposleni" i "Pozicija" označavati koju poziciju zaposleni zauzima.



Slika 6.2 Konceptualni model - upravljanje zaposlenima. Izvor: [Autor]

PRIMER KREIRANJA KONCEPTUALNOG MODELA PODATAKA ZA SISTEM ZA UPRAVLJANJE OCENAMA STUDENATA

Osnovni konceptualni model podataka za sistem upravljanja ocenama studenata

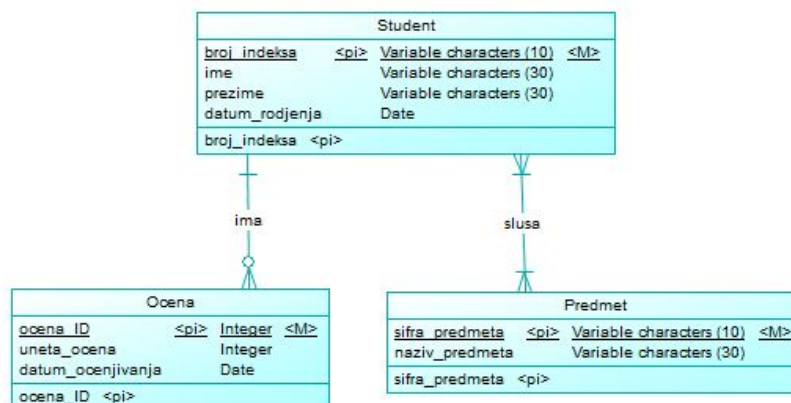
Razumevanje poslovnih zahteva: Cilj je da se razvije sistem za praćenje ocena studenata u različitim predmetima i godinama školovanja.

Identifikacija entiteta i definisanje atributa: Identifikovani entiteti mogu biti:

- **Student** (s atributima: broj indeksa, ime, prezime, datum rođenja, itd.)
- **Predmet** (s atributima: naziv predmeta, šifra predmeta, itd.)
- **Ocena** (s atributima: ocena, datum ocenjivanja, itd.)

Identifikacija veza: Veze će povezivati ove entitete. Na primer, veza između "Student" i "Predmet" će označavati da student može da sluša više predmeta, dok će veza između "Student" i "Ocena" označavati da student može imati više ocena u različitim predmetima.

Normalizacija: Podaci mogu biti normalizovani kako bi se eliminali višestruki unosi i redundancija. Na primer, informacije o studentu (ime, prezime, datum rođenja) mogu biti izdvojene u poseban entitet "Lični podaci" kako bi se izbeglo ponavljanje tih podataka za svaku ocenu koju student dobije.



Slika 6.3 Konceptualni model - upravljanje ocenama. Izvor: [Autor]

MOGUĆE POBOLJŠANJE KONCEPTUALNOG MODELA ZA SISTEM ZA UPRAVLJANJE ZAPOSLENIMA

Poboljšanja konceptualnog modela zavise od specifičnih zahteva i potreba organizacije.

Poboljšanja konceptualnog modela zavise od specifičnih zahteva i potreba organizacije. Uvek je važno sarađivati sa zainteresovanim stranama kako bi se model prilagodio njihovim stvarnim potrebama.

Dodavanje atributa u entitete: U entitet "Zaposleni", možemo dodati nove attribute kao što su kontakt telefon, e-mail adresa, broj radnih sati nedeljno, plata, i sl.

U entitet "Odeljenje", možemo dodati atribut "Opis odeljenja" koji će sadržati dodatne informacije o svakom odeljenju.

U entitet "Pozicija", možemo dodati atribut "Nivo pozicije" koji će opisati stepen odgovornosti i hijerarhije različitih pozicija.

Razmatranje veza: Možemo dodati novu vezu "Nadređeni zaposleni" koja će povezivati "Zaposlenog" sa drugim zaposlenim koji je njegov nadređeni, ako postoji.

Razdvajanje entiteta: Ako je potrebno pratiti više adresa za svakog zaposlenog (npr. privatna adresa, poslovna adresa), možemo razdvojiti entitet "Adresa" od entiteta "Zaposleni" kako bi se to podržalo.

Takođe, možemo razdvojiti entitet "Plata" od entiteta "Zaposleni" kako bi se omogućilo

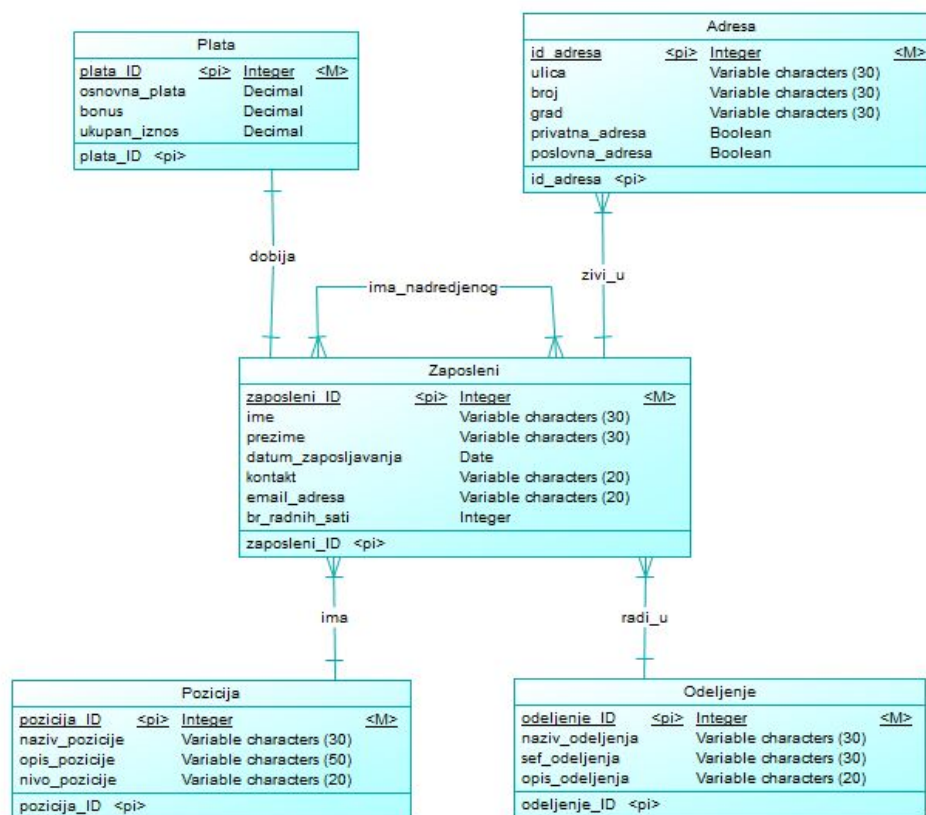
praćenje različitih informacija o plati za svakog zaposlenog (npr. osnovna plata, bonusi, itd.).

Razmatranje ograničenja: Možemo postaviti ograničenja na dužinu atributa, na primer, ograničiti broj karaktera za ime i prezime.

Unapređenje ER dijagrama: U ER dijagramu možemo dodati nove veze i attribute prema poboljšanim entitetima i ograničenjima.

POBOLJŠANI ER DIJAGRAM

Poboljšanje modela za sistem za upravljanje zaposlenima



Slika 6.4 Poboljšani ER dijagram - upravljanje zaposlenima. Izvor: [Autor]

▼ Poglavlje 7

Pokazna vežba

NAČIN ORGANIZACIJE POKAZNIH VEŽBI

Vežba je organizovana kroz uvod deo i deo za samostalni rad studenata

Vežba je organizovana kroz uvod deo i deo za samostalni rad studenata.

1. U uvodnom delu pokaznih vežbi se daje pokazni primer koji studentima treba da pomogne u samostalnom rešavanju zadataka.
2. Zadatke koji su zadati za samostalni rad student samostalno rešava uz pomoć asistenta.

▼ 7.1 Uvod u pokaznu vežbu

IZRADA KONCEPTUALNOG MODELA PODATAKA KORIŠĆENJEM POWER DESIGNER-A - (2 MIN.)

PowerDesigner je Sybase-ov alat za modelovanje procesa sistema i podataka

PowerDesigner je Sybase-ov alat za modelovanje procesa sistema i podataka. PowerDesigner je projektovan za Windows okruženje. Ona sadrži i dodatke kako bi koristila Java platformu i programsko okruženje Eclipse. PowerDesigner je 2002. godine zauzeo 40% tržišta. Posедуje mnogo mogućnosti u oblasti modelovanja:

- Modelovanje poslovnih procesa
- Generisanje programskog koda (Java, C#, VB.NET, JSF, PowerBuilder itd),
- Modelovanje podataka korišćenjem E/R dijagrama
- Modelovanje DataWarehouse-a,
- Eclipse programiranje,
- Objektno modeliranje
- Generisanje izveštaja,
- Rad sa repozitorijumima,
- Generisanje XML šema u DTD standardu itd.

U okviru ovog kursa, Power Designer će se koristiti za izradu konceptualnog modela baza podataka i njegovu transformaciju u logički a zatim i fizički model relacione baze podataka

ŠTA JE KONCEPTUALNI MODEL PODATAKA? - (2 MIN.)

Konceptualni model podataka je apstraktno viđenje stanja realnog sistema. Može se predstaviti različitim tehnikama a jedna od njih je tehnika E/R dijagrama

Model podataka se može predstaviti različitim tehnikama a jedna od njih je tehnika E/R dijagrama. Interpretacija podataka se u modelu podataka predstavljenim ER dijagramom ostvaruje kroz dve njegove osnovne komponente:

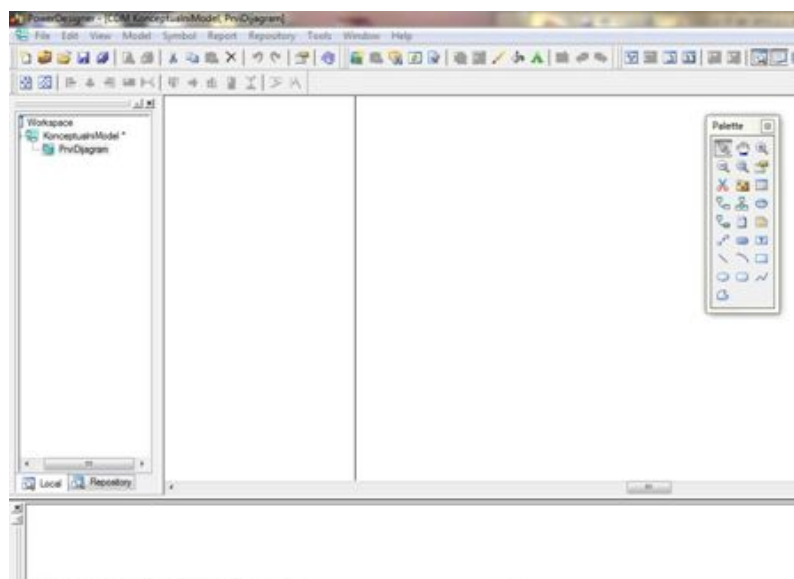
- Strukturu podataka, preko koje se predstavljaju statičke karakteristike sistema;
- Ograničenja na vrednosti podatka koja u svakom trenutku posmatranja (stacionarnom stanju) treba da budu zadovoljena.

Postoji mnogo verzija ER dijagrama i one se razlikuju po tome što za predstavljanje različitih koncepata modela kao što su entiteti, veze i atributi koriste različite simbole.

KREIRANJE KONCEPTUALNOG MODELA - (3 MIN.)

Crta se korišćenjem palete koja sadrži relevantne simbole

Kada se selektuje dijagram koji želi da se kreira (u ovom slučaju konceptualni dijagram) potrebno je dati ime modelu i kliknuti na OK dugme kako bi se otvorila radna površina konceptualnog modela.(slika 8.1.)



Slika 7.1.1 Površina za kreiranje konceptualnog modela [Izvor: NM IT350 -2020/2021.]

Kako bi kreirali neki konceptualni model prvenstveno selektujemo nekih od ponuđenih simbola iz palete.

Svaki simbol u paleti prikazanoj na slici 8.2. ima određeno značenje. Neki služe za izbor, neki za kreiranje entiteta, relacija, nasleđivanja i sl. Nakon odabira simbola za kreiranje entiteta (prelaskom miša preko tabele, gde dolazi do ispisivanja "Entity") mišem kliknemo na mesto gde želimo da postavimo entitet. Svakom entitetu treba dodeliti ime i za njega definisati odgovarajuće atribute. Nakon kreiranja više entiteta povezujemo ih relacijama.

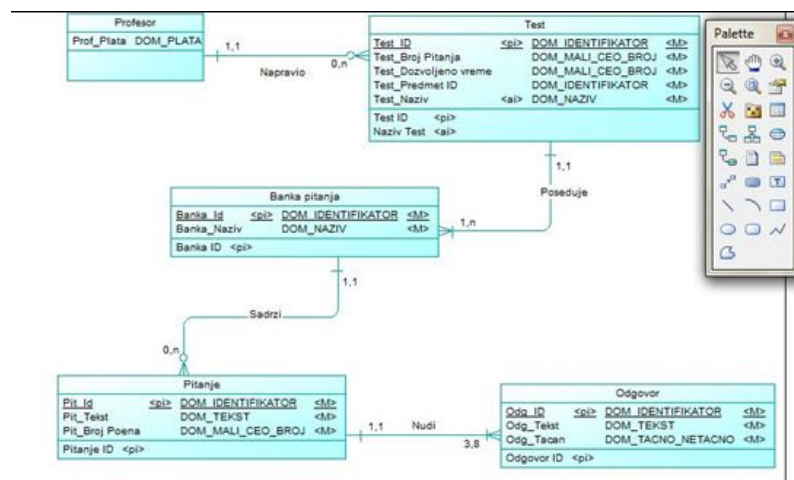


Slika 7.1.2 Simboli za kreiranje konceptualnog modela [Izvor: NM IT350 -2020/2021.]

ČUVANJE KONCEPTUALNOG DIJAGRAMA - (2 MIN.)

Objašnjenje postupka čuvanja

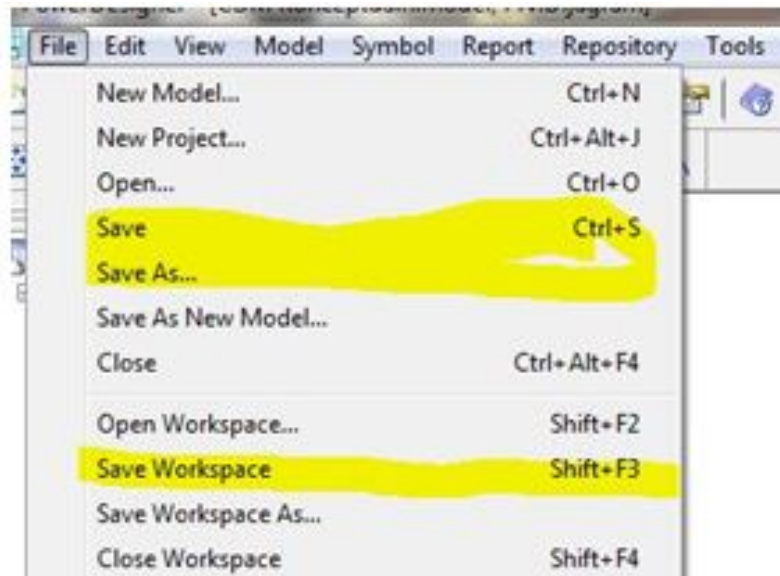
Ukoliko želimo da sačuvamo kreirani dijagram (na primer sa slike 8.3.) to se izvršava pritiskom na "File" meni koji je prikazan na slici 8.4. Bira se opcija save as ili samo save ukoliko želimo da snimimo samo kreirani dijagram. Ukoliko imamo više različitih dijagrama, različitih modela, možemo snimiti i celu radnu površinu klikom na



Slika 7.1.3 Konceptualni model za čuvanje [Izvor: NM IT350 -2020/2021.]

Primer jednog kreiranog konceptualnog modela u PowerDesigner-u prikazan je na slici 8.3.

"Save Workspace". Ukoliko se snimi cela radna površina nakon njenog ponovnog pokretanja, pokreću se i svi sačuvani modeli.



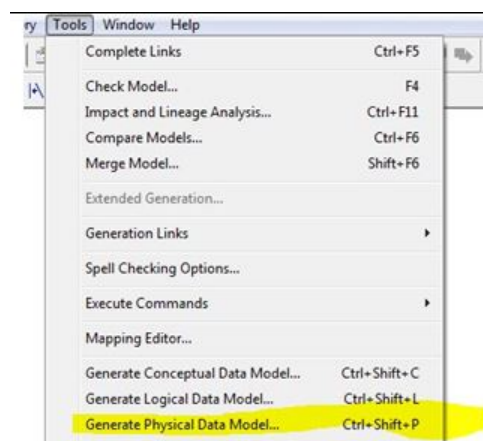
Slika 7.1.4 Čuvanje konceptualnog modela [Izvor: NM IT350 -2020/2021.]

KREIRANJE LOGIČKOG (RELACIONOG) DIJAGRAMA - (4 MIN.)

Moguće je na dva načina: generisanjem iz konceptualnog modela ili pravljenjem novog logičkog modela

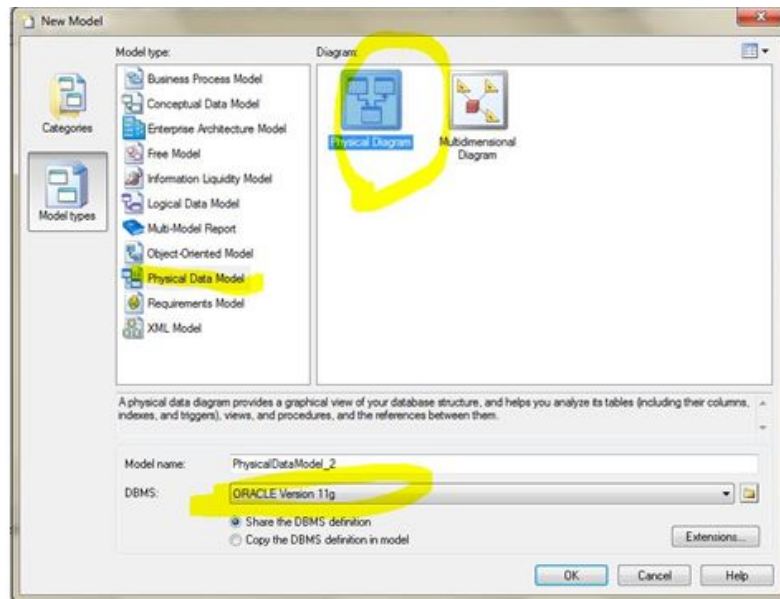
Ukoliko želimo da kreiramo logički dijagram, to se može izvršiti na dva načina.

1. Ukoliko imamo konceptualni model, logički model možemo generisati ulaskom u opciju "Tools" pa zatim "Generate Physical Data Model" (iako se radi o logičkom modelu). Potrebno je naglasiti prilikom kreiranja ili generisanja logičkog modela da se odabere vrsta RDBMS-a (sistema za upravljanje baza podataka) koji će se na dalje koristiti kako bi se nekog generisanja fizičkog modela iz logičkog modela, fizički model bio u određenom RDBMS (MySQL, oracle, MS Access itd) (slika 8.5).



Slika 7.1.5 Prikaz generisanja logičkog modela iz konceptualnog modela u PD-u [Izvor: NM IT350 -2020/2021.]

Ukoliko želimo da kreiramo logički model bez prethodno kreiranog konceptualnog modela, potrebno je prilikom odabira novog modela izabrati "Physical Data Model" a zatim izabrati "Physical diagram". Nakon toga otvara se prozor gde se nude opcije za naziv modela kao i odabir DBMS-a. U ovom slučaju je izabran Oracle 11g verzija. Nakon odabira otvara se okruženje za kreiranje logičkog modela. (slika 8.6.)

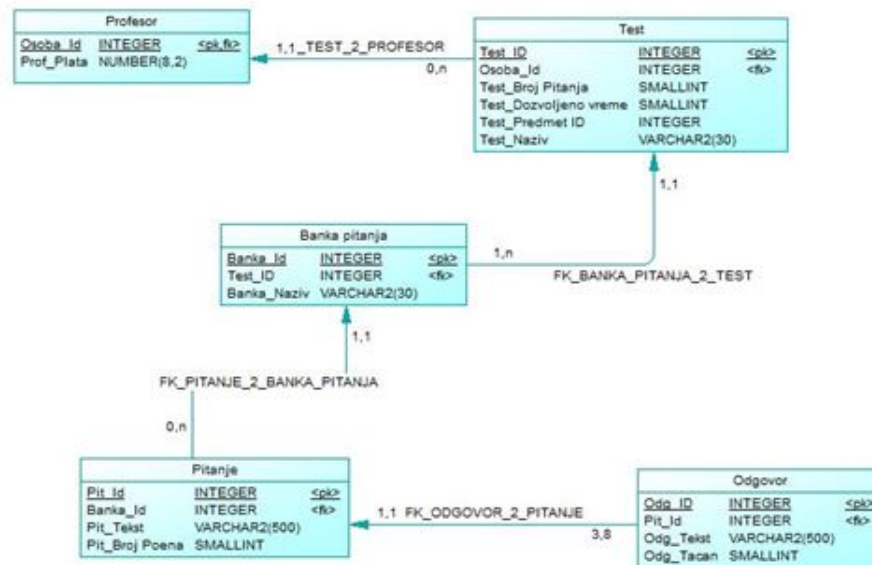


Slika 7.1.6 okruženje za kreiranje logičkog modela [Izvor: NM IT350 -2020/2021.]

GENERISANJE FIZIČKOG MODELA (SQL SKRIPTI) ZA KREIRANJE TABELA U BAZI NA SNOVU LOGIČKOG MODELA - (5 MIN.)

Tabele se generišu za izabrani RDBMS

Na slici 8.7 je prikazan primer logičkog modela koji odgovara prethodno kreiranom konceptualnom modelu.



Slika 7.1.7 Primer logičkog modela koji odgovara prethodno kreiranom konceptualnom modelu [Izvor: NM IT350 -2020/2021.]

Nakon kreiranog logičkog modela, model možemo sačuvati i na osnovu njega generisati skriptu za kreiranje tabela i ostalih elemenata u bazi podataka. To se izvršava tako što se nakon otvaranja taba "Database" koji se pojavljuje kada se na određenom logičkom modelu klikne na opciju "generate database". S obzirom da smo za DBMS odabrali Oracle 11g on će automatski biti selektovan i naredbe će biti prilagođene njemu. Nakon generisanja, dobiće se .sql file sa naredbama.

PRIKAZ GENERISANOG .SQL FILE-A - (3 MIN.)

.SQL file za kreiranje tabela u bazi

Na slici 8.8 je prikazan .sql file sa naredbama za kreiranje tabela u izabranom RDBMS.


```

crebas.sql - Notepad
File Edit Format View Help

/* Table: PITANJE */
/* ===== */
create table PITANJE (
  PIT_ID          INTEGER          not null,
  BANKA_ID        INTEGER          not null,
  PIT TEKST       VARCHAR2(500)    not null,
  PIT_BROJ_POENA  SMALLINT         not null,
  constraint PK_PITANJE primary key (PIT_ID)
);

comment on table PITANJE is
'Pitanje je gradivna jedinica svakog testa. Test može imati veliki broj pitanja,
raspoređenih u više banki pitanja, ali test takode definiše broj pitanja koje je
potrebno da uradi. Na primer, ukupan broj pitanja može biti 30 a student treba da
uradi samo 8, slučajnim izborom odabranih pitanja.

Pitanje sadrži 3-8 odgovora, od kojih SAMO JEDAN može, i mora, biti tačan.';

/* Index: SADRZI_FK */
/* ===== */
create index SADRZI_FK on PITANJE (
  BANKA_ID ASC
);

/* Table: PROFESOR */
/* ===== */
create table PROFESOR (
  OSOBA_ID        INTEGER          not null,
  PROF_PLATA      NUMBER(8,2)      not null,
  constraint CKC_PROF_PLATA_PROFESOR check (PROF_PLATA >= 1),
  constraint PK_PROFESOR primary key (OSOBA_ID)
);

comment on table PROFESOR is
'Profesor fakulteta. Podatke o predmetima na kojima profesor predaje ovaj podsistem
dobija od sistema univerziteta.';

```

Slika 7.1.8 Skripta za kreiranje tabela u BP [Izvor: NM IT350 -2020/2021.]

IZRADA ER DIJAGRAM NA PRIMERU KONCEPTUALNOG MODELA BAZE PODATAKA UNIVERZITETA METROPOLITAN - (3 MIN.)

*Konceptualni model baze podataka prikazuje tipove entiteta, njihove
atribute, identifikatore i relacije.*

Zadatak: za opisani problem kreirati E/R dijagram u CASE alatu PowerDesigner.

Opis problema: Nastavni proces na osnovnim studijama na Univerzitetu Metropolitan se
odvija na sledeći način:

Univerzitet organizuje osnovne akademske studije u okviru tri fakulteta:

*FIT, Fakultet za menadžment i Fakultet digitalnih umetnosti. Osnovne akademske studije traju
4 godine i nose 240 ESPB.*

*Na FIT-u izučavaju studijski programi: Softversko inženjerstvo ili Informacioni sistemi ili Razvoj
softvera za računarske igre . Na Fakultetu za menadžment studijski programi Operacioni
menadžment, Marketing menadžment i Menadžment u sportu a na Grafičkom dizajnu Dizajn
interaktivnih medija i Dizajn štampanih medija.*

*Student ima obavezu da u okviru studijskog programa položi obavezne zajedničke predmete.
Po završetku studijskog programa, studenti stiču odgovarajuća zvanja npr. Inženjer
Informacionih Tehnologija - "naziv studijske grupe", (pri čemu "naziv studijske grupe" može
biti Softversko inženjerstvo ili Informacioni sistemi ili Razvoj softvera za računarske igre).
Svaka studijska godina organizovana je u dva semestra sa po 15 nedelja aktivne nastave,
a svi predmeti su jednosemestralni. Obim svakog predmeta je iskazan brojem ESPB, a*

obim studija zbirom ESPB. U sve četiri godine zbir ESPB iznosi 60 što odgovara prosečnom ukupnom angažovanju studenata u obimu od 40 časova nedeljno tokom školske godine.

Zadatak: Na osnovu prethodnog opisa, identifikovati potencijalne tipove entiteta i njihove i atribut i kreirajte ih u CASE alatu PowerDesigner.

Rešenje: Potencijalni tipovi entiteta za opisani problem su:
Predmet, Student, Studijski Program, Studijska Godina, Semestar, Mentor itd.

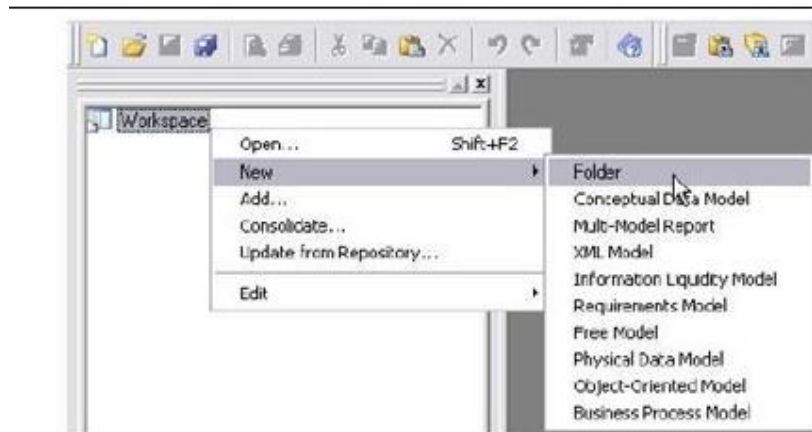
KORAK 1. U KREIRANJU ER DIJAGRAMA - (5 MIN.)

Kreiranje novog foldera u kojem će se smestiti konceptualni model

Da bi konceptualni model predstavili u PowerDesigneru treba uraditi sledeće:

1. Kreirati novi folder u kojem će se smestiti konceptualni model
2. Izabrati opciju za kreiranje konceptualnog modela
3. Kreirati entitete i njima dodeliti odgovarajuće atribut
4. Kreirati identifikatora za entitete
5. Za neke atribut kreirati odgovarajuće domene
6. Kreirati veze između entiteta (sa definisanom kardinalnošću i opcionalnošću)

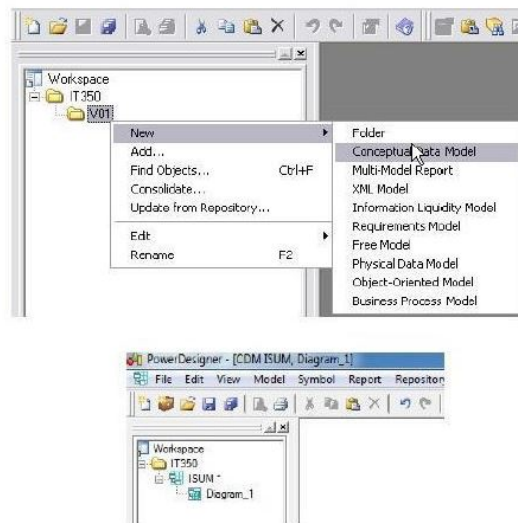
Da bi se kreirao novu folder nakon pokretanja PowerDesigner-a treba izvršiti akcije prikazane na slici 8.9.



Slika 7.1.9 Kreiranje novog foldera [Izvor: NM IT350 -2020/2021.]

KORACI 2. I 3. U KREIRANJU ER DIJAGRAMA - (3 MIN.)

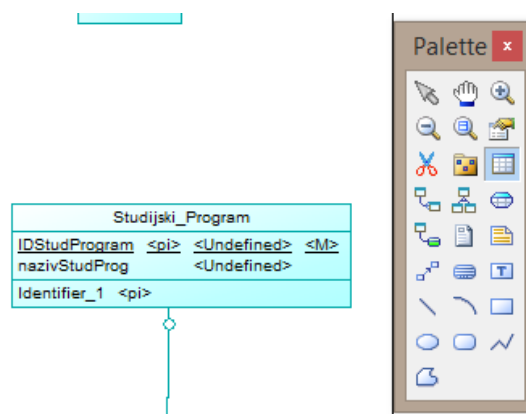
Korak 2: Izabrati opciju za kreiranje konceptualnog modela; Korak 3: Kreirati entitete i njima dodeliti odgovarajuće atribut



Slika 7.1.10 Kreiranje konceptualnog modela [Izvor: NM IT350 -2020/2021.]

Korak 2: Akcije koje treba izvršiti prilikom kreiranja konceptualnog modela prikazane se na slici 8.10.

Korak 3: Kreiranje entiteta se vrši izborom odgovarajućeg simbola za entitete što je prikazano na slici 8.11. Zatim se dvostrukim klikom na kreirani entitet dobija ekran iz kojeg treba izabrati opciju *attribute* uz pomoć koje treba kreirati sve atribute za entitet, što je prikazano na slici 8.11.



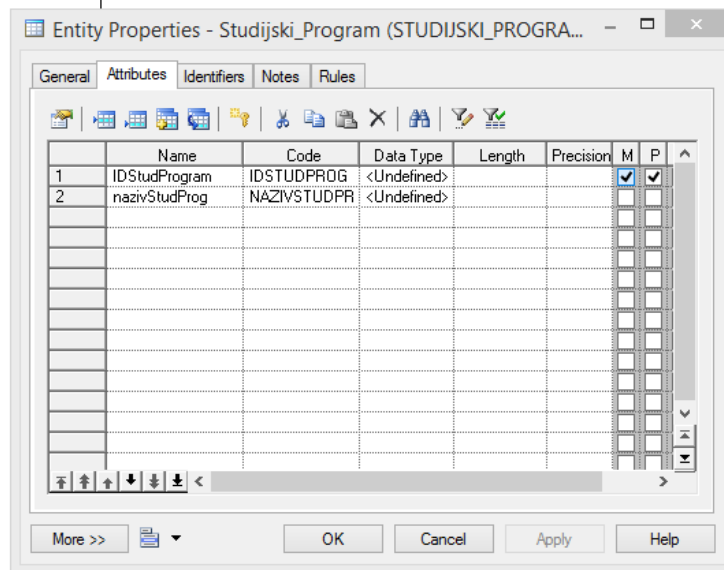
Slika 7.1.11 Izbor simbola za kreiranje entiteta [Izvor: NM IT350 -2020/2021.]

KORAK 4. U KREIRANJU ER DIJAGRAMA - (3 MIN.)

Korak 4: Kreiranje identifikatora za entitete

Identifikator predstavlja atribut ili skup atributa koji jednoznačno određuje jednu pojavu entiteta.

Atribut koji predstavlja identifikator mora biti obavezan (mandatory-M) i označen kao P (primary identifier). Kreiranje identifikatora za tip entiteta STUDIJSKI_PROGRAM je prikazan na slici 8.12.



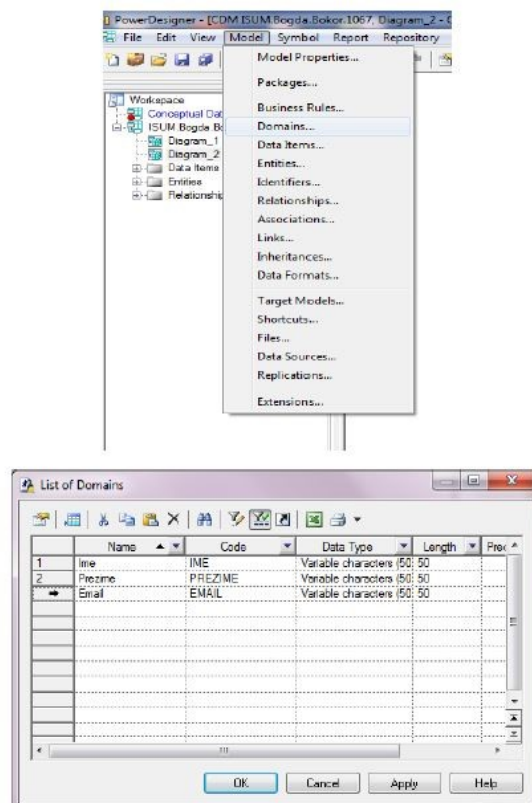
Slika 7.1.12 Kreiranje identifikatora entiteta [Izvor: NM IT350 -2020/2021.]

Isti postupak je potrebno ponoviti za sve entitete konceptualnog modela podataka.

KORAK 5. U KREIRANJU ER DIJAGRAMA - (2 MIN.)

Korak 5: Kreiranje domena

U PowerDesigner-u domeni se kreiraju na način prikazan na slici 8.13.



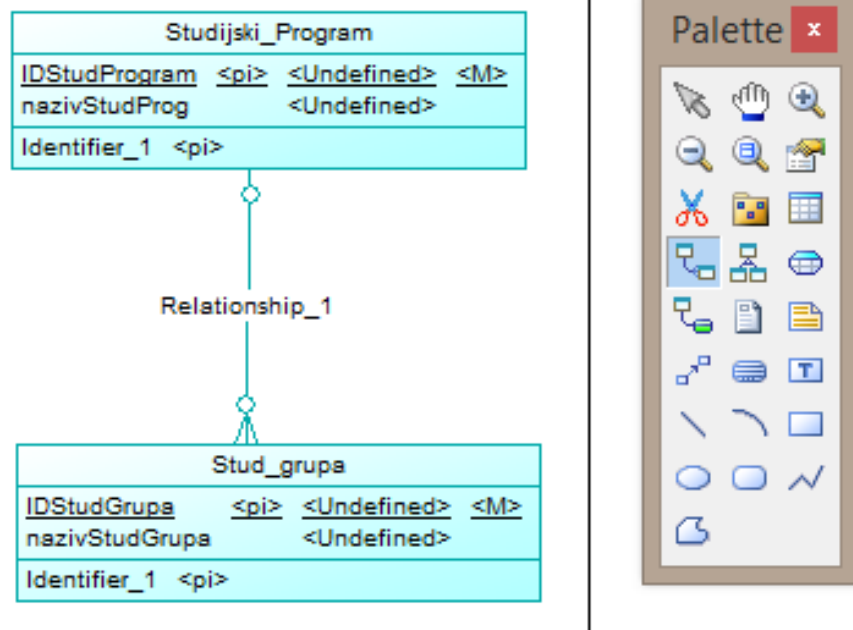
Slika 7.1.13 Prikaz kreiranja domena [Izvor: NM IT250 -2020/2021.]

Nakon kreiranja domena, možemo promeniti tip entiteta Kandidat kako bi njegovim atributima dodelili vrednost iz kreiranog domena.

KORAK 6. U KREIRANJU ER DIJAGRAMA - (3 MIN.)

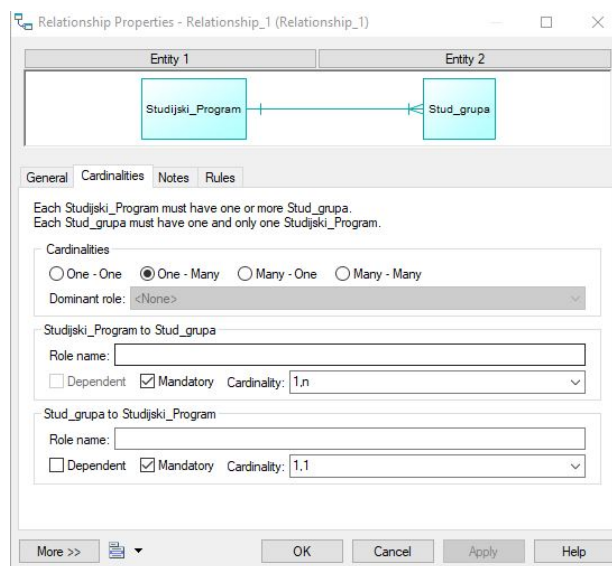
Korak 6: Kreiranje veza između entiteta

Veza se uspostavlja izborom simbola veze što je prikazano na slici 8.14. Podrazumevajuća kardinalnost veze je 1:M i veza je opciona što je prikazano kružićima na krajevima veze.



Slika 7.1.14 Veza između dva tipa entiteta koja se dobija izborom simbola veze iz palete. Podrazumeva se da je njena kardinalnost 1:M i da je ona opciona [Izvor: NM IT350 -2020/2021.]

Kardinalnost i opcionalnost veze se može promeniti dvostrukim klikom na samu vezu kada se prikazuje ekran iz kojeg treba izabrati opciju Cardinality kao na slici 8.15. Izborom opcije Cardinality može se izmeniti kardinalnost veze a opcije Mandatory opcionalnost veze. Na ovaj način treba kreirati sve veze između entiteta.



Slika 7.1.15 Podašavanje kardinalnosti i opcionosti veze [Izvor: Autor]

POKAZNI PRIMER BR. 1 - (2 MIN.)

U nastavku sledi opis problema za koji se treba napraviti konceptualni model i predstaviti ga E/R dijagramom

Jedna zlatara koja želi da se bavi on-line prodajom skupocenog nakita planira da napravi svoj informacioni sistem. Neki od zahteva koje su vlasnici buduće on-line zlatare postavili glase:

Za svakog korisnika ove on-line zlatare treba sačuvati osnovne podatke (ime, prezime, adresu kao i account koji se sastoji od username i password).

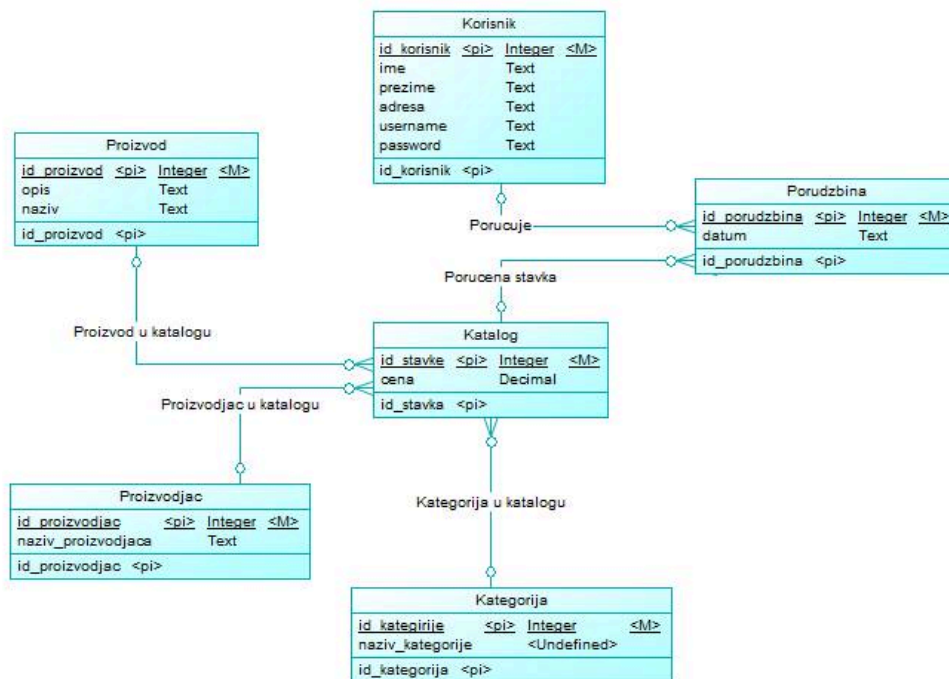
Svi proizvodi u on-line zlatari imaju svoj identifikacioni broj i opis koji se daje u tekstualnom obliku. Proizvodi formiraju katalog proizvoda iz kojeg se pored osnovnih podataka o proizvodu može videti kategorija kojoj proizvod pripada (ogrlica, minđuše, prsten, narukvica), proizvođač koji ga je proizveo kao i cena proizvoda. Cena proizvoda zavisi od kategorije proizvoda i proizvođača (kategorije proizvoda mogu imati različite cene zavisno od proizvođača, npr, minđuše jedan proizvođač može proizvesti po jednoj a drugi po drugoj ceni).

Kupac iz kataloga proizvoda može da izabere jedan ili više proizvoda stavljanjem u korpu (Shopping cart). Sadržaj Shopping cart se ne čuva trajno u bazi podataka već se njen sadržaj privremeno pamti u operativnoj memoriji računara dok traje kupovina (sesija). Po završetku kupovine, na osnovu kupljenog nakita iz Shopping cart se formira porudžbenica.

Zadatak: Pokušajte da nacrtate E/R dijagram u kojem će biti predstavljeni tipovi entiteta, njihovi atributi i veze za opisani problem. Na kraju proverite svoje rešenje sa priloženim rešenjem.

POKAZNI PRIMER BR. 1 - REŠENJE - (3 MIN.)

U nastavku sledi rešenje datog primera.



Slika 7.1.16 ER dijagram za bazu podataka online prodaje skupocenog nakita [Izvor: NM IT350 -2020/2021.]

7.2 Pokazna vežba - samostalni rad

ZADATAK 1. ZA ZA SAMOSTALNI RAD

Opis zadatka 1 - Student bira jedan od dva zadatka. Preostali zadatak se preporučuje za samostalni rad. Vreme izrade svakog zadatka je 45 min.

Student bira jedan od dva zadatka. Preostali zadatak se preporučuje za samostalni rad.

Projektovati konceptualnu šemu baze podataka i prikazati je korišćenjem E/R dijagrama (koristeći Power Designer). Analizirati zahteve i na osnovu njih odrediti entitete, atribute, veze:

1. Kablovski operater uvodi novu mogućnost u okviru svoje ponude - Gledanje na zahtev. Na zahtev se mogu gledati sve emisije, filmovi, serije, ... koji su se prikazivani u prethodnom periodu.
2. Takođe, postoje i filmovi i serije koji uvek dostupni na serveru operatera
3. Svaki korisnik se mora registrovati za ovu opciju jer se ona naplaćuje dodatno.
4. Registracija se vrši za gledanje jedne emisije, a registracija važi 24 sata. Emisija se može gledati nekoliko puta u roku od 24 časa

5. Za filmove i serije registracija važi za jedno gledanje. Nakon gledanja određenog filma ili serije, isti se može pogledati tek za 48 časova.

Potrebno vreme za rešavanje zadatka 45 minuta.

ZADATAK 2. ZA ZA SAMOSTALNI RAD

Opis zadatka 2 - Student bira jedan od dva zadatka. Preostali zadatak se preporučuje za samostalni rad. Vreme izrade svakog zadatka je 45 min.

Student bira jedan od dva zadatka. Preostali zadatak se preporučuje za samostalni rad.

Projektovati konceptualnu šemu baze podataka i prikazati je korišćenjem E/R dijagrama (koristeći Power Designer). Analizirati zahteve i na osnovu njih odrediti entitete, attribute, veze:

1. Studenti svake akademske godine upisuju godinu studija. Pri tom biraju neke od ponuđenih izbornih predmeta, tako da zbir njihovih bodova u svakom semestru bude barem 30. Predmeti moraju biti izabrani najkasnije do 15. oktobra.
2. Fakultet je organizovan u departmane. Svaki profesor/asistent/nastavnik je član tačno jednog departmana. Nastavnici iz istog departmana između sebe biraju šefa departmana. Da bi bolje upravljao ljudskim resursima, šef departmana može menjati plate svojih nastavnika. Šef departmana može biti samo profesor.
3. Jedna od zadataka departmana je da se brine o nastavi. Svake akademske godine departmana nudi nekoliko izbornih predmeta za studente i osigurava nastavnike koji će predavati te predmete. Pri tom svaki predmet ima samo jednog nastavnika. Takođe, departman brine i o ispitima. U toku akademske godine organizuje se nekoliko ispitnih rokova, sa određenom satnicom. Departman zakazuje ispite, određuje mesto održavanja ispita i o tome obaveštava profesore i studente. Svaki profesor je u obavezi da dežura na ispitu. Na kraju akademske godine fakultet pohvaljuje studente koji su postigli najbolje ocene iz upisanih predmeta.

Potrebno vreme za rešavanje zadatka 45 minuta.

VIDEO

Modeliranje baze podataka "Prijem pacijenata"

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

✓ Poglavlje 8

Domaći zadatak

DOMAĆI ZADATAK 3 - VREME IZRADE 90 MIN.

Uputstvo za izradu domaćeg zadatka

Svaki od studenata dobija, na način kako je to kasnije definisano, jednu od pet navedenih baza podataka za koju korišćenjem PowerDesigner-a treba da kreira konceptualni model baze podataka i predstavi ga korišćenjem E/R dijagrama. U okviru konceptualnog modela treba:

1. identifikovati tipove entiteta i njihove atributa
2. proizvoljno uspostaviti odgovarajuće relacije između tipova entiteta onako kako to po mišljenju svakog studenta izgleda najlogičnije (treba definisati minimalnu i maksimalnu kardinalnost svake relacija).

Predložene baze podataka su:

Baza podataka za podršku proizvodnje automobila koja treba da obezbedi evidenciju o:

- proizvedenim automobilima i njihovim modelima
- delovima i ostalih materijalima koji se ugrađuju u svaki model automobila
- dobavljačima i kupljenim delovima za automobile od dobavljača
- prodaji automobila kupcima

Baza podataka hotela koji pruža usluge izdavanja hotelskih soba i pripremu hrane za goste hotela a koja treba da obezbedi evidenciju o:

- hotelskim sobama za izdavanje,
- menijima sa pripremljenom hranom koja se nudi gostima hotela,
- potrebnim namirnicama za pripremu hrane
- dobavljačima i kupljenim namirnicama od dobavljača
- rezervaciji soba u hotelu i raspoloživosti soba

Baza podataka doma zdravlja koji treba da obezbedi evidenciju o:

- medicinskom osoblju doma zdravlja (lekarima i medicinskim sestrama)
- pacijentima i njihovim izabranim lekarima
- zakazanim pregledima pacijenata kod lekara,
- dijagnozama postavljenim pacijentima prilikom pregleda od strane lekara
- lekova i uputima za druge preglede izdatim pacijentima nakon obavljenog pregleda od strane lekara

DOMAĆI ZADATAK 3 - NASTAVAK

Uputstvo za izradu domaćeg zadatka - nastavak

Baza podataka univerziteta koji treba da obezbedi evidenciju o:

- studentima i nastavnicima na fakultetu
- angažovanju nastavnika na predmetima
- predmetima koji studenti stižu pravo da polažu upisom u odgovarajuću godinu studija,
- prijavama studenata za polaganje ispita i ocenama dobijenim na ispitu

Baza podataka za podršku rada neke picerije koji treba da obezbedi evidenciju o:

- napravljenim picama,
- potrebnim namirnicama za pravljenje pica
- dobavljačima i kupljenim namirnicama od dobavljača
- prodaji pica kupcima

Raspodelu zadataka vrši asistent tako što:

1. bazu podataka br. 1 dodeljuje studentima čiji se broj indeksa završava na 0 i 5.
2. bazu podataka br. 2 dodeljuje studentima čiji se broj indeksa završava na 1 i 6.
3. bazu podataka br. 3 dodeljuje studentima čiji se broj indeksa završava na 2 i 7.
4. bazu podataka br. 4 dodeljuje studentima čiji se broj indeksa završava na 3 i 8.
5. bazu podataka br. 5 dodeljuje studentima čiji se broj indeksa završava na 4 i 9.

Asistent ili profesor su u obavezi da studentima daju sve potrebne dodatne informacije vezano za baze podataka koje analiziraju pri rešavanju domaćih zadataka.

DOMAĆI ZADATAK 3 - UPUTSTVO ZA SLANJE DOMAĆEG ZADATKA

Kreiranje ER modela

Prilikom slanja domaćih zadataka, neophodno je da ispunite sledeće:

- Subject mail-a mora biti IT250-DZbr (u slučaju kada šaljete domaći za ovu nedelju to je IT250-DZ03)
- U prilogu mail-a treba da se nalazi arhiviran projekat koji se ocenjuje imenovan na sledeći način: IT250-DZbr-BrojIndeksa-Ime Prezime. Na primer, IT250-DZ03-1234-VeljkoGrkovic
- Telo mail-a treba da ima pozdravnu poruku
- Arhivu sa zadatkom poslati na adresu predmetnog asistenta:
milica.vlajkovic@metropolitan.ac.rs (studenti u Beogradu i online studenti) ili
tamara.vukadinovic@metropolitan.ac.rs (studenti u Nišu).

Svi poslani mail-ovi koji ne ispunjavaju navedene uslove **NEĆE** biti pregledavani. Za sva pitanja ili nedoumice u vezi zadatka, možete se obratiti asistentu

▼ Zaključak

ZAKLJUČAK

Šta smo naučili u ovoj lekciji?

Relacioni model je trenutni standard implementacije baza podataka. U relacionom modelu, krajnji korisnik percipira podatke kao da su smešteni u tabelama. Tabele su povezane međusobno putem zajedničkih vrednosti u zajedničkim atributima. Model entitetskih veza (ER model) je popularno grafičko sredstvo za modeliranje podataka koje dopunjuje relacioni model.

ER model omogućava dizajnerima baza podataka da vizualno predstavljaju različite poglede na podatke - kao što ih vide dizajneri baza podataka, programeri i krajnji korisnici - i da integrišu podatke u zajednički okvir.

U ovoj lekciji se detaljno objašnjavaju osnovni koncepti ER dijagrama jer je primena i korišćenje ovakvog projektovanja baza podataka jedan od najvažnijih ciljeva ovog kursa. Tako se objašnjavaju koncepti tipova entiteta, entiteta, atributa, relacija, kardinalnosti relacija itd. Svi ovi koncepti su objašnjeni korišćenjem karakterističnih primera što studentima treba da pomogne njihovo razumevanje.

LITERATURA

Za pisanje ove lekcije je korišćena sledeća literatura

1. [http://corpgov.crew.ee/Materjalid/Database%20Systems%20-%20Design,%20Implementation,%20and%20Management%20\(9th%20](http://corpgov.crew.ee/Materjalid/Database%20Systems%20-%20Design,%20Implementation,%20and%20Management%20(9th%20)
2. Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom, DATABASE SYSTEMS –The Department of Computer Science, Stanford University, 2009 by Pearson Education Inc.
3. David M. Kroenke, Database Processing – fundamentals, design and implementation, Prentice Hall, 2004.
4. C. J. Date, An introduction to Database Systems, Addison-Wesley Publishing Company, 1990
5. <https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/>
6. <https://www.google.com/search?ei=bESaW8KsNqTSgAazvaiYBw&q=entity+relationship+diagram+powerdesigner&oq=entab.1.2.0j0i22i30k1i3j0i22i10i30k1.1916.4268.0.9223.6.6.0.0.0.105.573.4j2.6.0....0...1c.1.64.psyab..0.6.571...0i67k1.0.1qPVVoqyWEA>