



## IT250 - BAZE PODATAKA

Konceptualno modeliranje  
podataka primenom naprednih  
E/R dijagrama

Lekcija 04

PRIRUČNIK ZA STUDENTE

# IT250 - BAZE PODATAKA

## Lekcija 04

### *KONCEPTUALNO MODELIRANJE PODATAKA PRIMENOM NAPREDNIH E/R DIJAGRAMA*

- ✓ Konceptualno modeliranje podataka primenom naprednih E/R dijagrama
- ✓ Poglavlje 1: Ternarne relacije
- ✓ Poglavlje 2: Višestruke relacije i asocijativni entiteti
- ✓ Poglavlje 3: Specijalizacija i generalizacija
- ✓ Poglavlje 4: ID zavisni i slabi tipovi entiteta
- ✓ Poglavlje 5: Primeri naprednih ER dijagrama
- ✓ Poglavlje 6: Primer konceptualnog modela baze podataka COMPANY
- ✓ Poglavlje 7: Primer konceptualnog modela baze Tiny College
- ✓ Poglavlje 8: Pokazna vežba
- ✓ Poglavlje 9: 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?*

U ovoj lekciji se govori o naprednim ER dijagramima kod kojih se podrazumeva korišćenje složenih relacija kako što su ternarna, kvaternarna, rekurzivna, asocijativna relacija, specijalizacija, generalizacija itd. U lekciji su kroz veći broj primera:

1. Objašnjeni razlozi za uvođenje ternarnih relacija kod kojih je stepen relacije tri i način na koje se one mogu prikazivati. Zbog ograničenja CASE alata koji se na ovom predmetu koristi za crtanje ER modela, za predstavljanje svih n-arnih relacija se umesto simbola za relaciju koristiti simbol za tip entiteta
2. Pokazni primer korišćenja višestrukih relacija koji se mogu uspostaviti između dva tipa entiteta
3. Prikazani primeri asocijativnih entiteta koji predstavljaju specijalan slučaj relacije sa maksimalnom kardinalnošću "više prema više" koja zahteva da neki novokreirani tip entiteta pamti attribute te relacije.
4. Objašnjeni razlozi za uključivanje specijalizacije i generalizacije u model podataka kao i različite mogućnosti ovih specifičnih veza koje se uspostavljaju između tipova entiteta (npr. Hijerarhija specijalizacije, rešetka specijalizacije, Unija ili kategorija)

Svi navedeni napredni koncepti su dokumentovani sa većim brojem primera, a način njihove primene je takođe objašnjen kroz rešavanje primera koji su dati u lekciji.

## ▼ Poglavlje 1

# Ternarne relacije

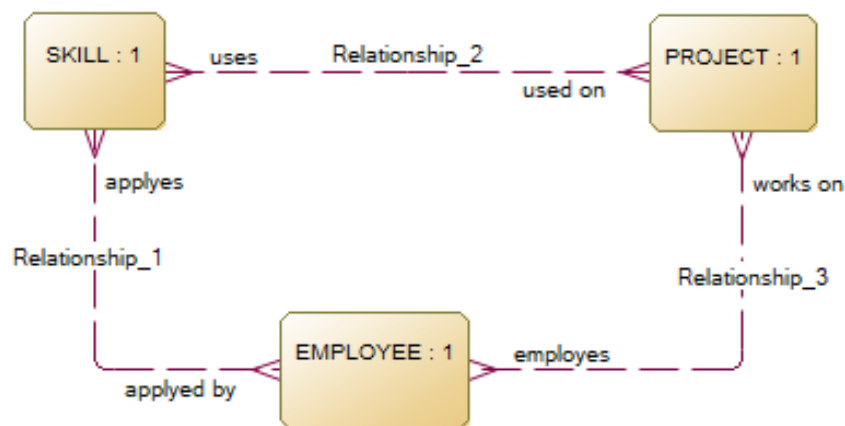
## ŠTA SU TERNARNE RELACIJE?

### Relacija čiji je stepen relacije tri

**Kardinalnost** izražava minimalan i maksimalan broj pojavljivanja entiteta povezanog sa jednim pojavljivanjem povezanog entiteta.

Napredni ER dijagrami podrazumevaju korišćenje složenih relacija kako što su ternarna, rekurzivna, asocijativna relacija, specijalizacija.

Ternarna relacija je relacija čiji je stepen relacije tri. Na slici 1.1 je dat primer na koji se može primeniti ternarna relacija: potrebno je da znamo koje veštine (SKILL) zaposleni (EMPLOYEE) koriste na kojim projektima (PROJECT). U ovom slučaju se koriste tri binarne relacije (SKILL-PROJECT, PROJECT - EMPLOYEE, EMPLOYEE- SKILL):



Slika 1.1 Primer na kojem se može primeniti ternarna relacija [Izvor: NM IT350 -2020/2021.]

1. Uloga **primenjuje** (*applies*) ukazuje na to koji zaposleni ima koje veštine.
2. Uloga **korišćena na** (*used on*) ukazuje koja se veština koristi na kom projektu.
3. Uloga **radi na** (*works on*) ukazuje koji zaposleni radi na kom projektu.

Ali to još uvek nije dovoljno da se prikaže koji zaposleni koristi koje veštine na kom projektu. Na primer:

- *works\_on* (*radi na*) – Petar i Ana rade na projektima A i B.
- *applies* (*primenjuje*) – Petar primenjuje veštine *interface design* i *database design* a Ana samo veštinu *database design*.
- *used on* (*korišćena na*)– Obe veštine se koriste na oba projekta

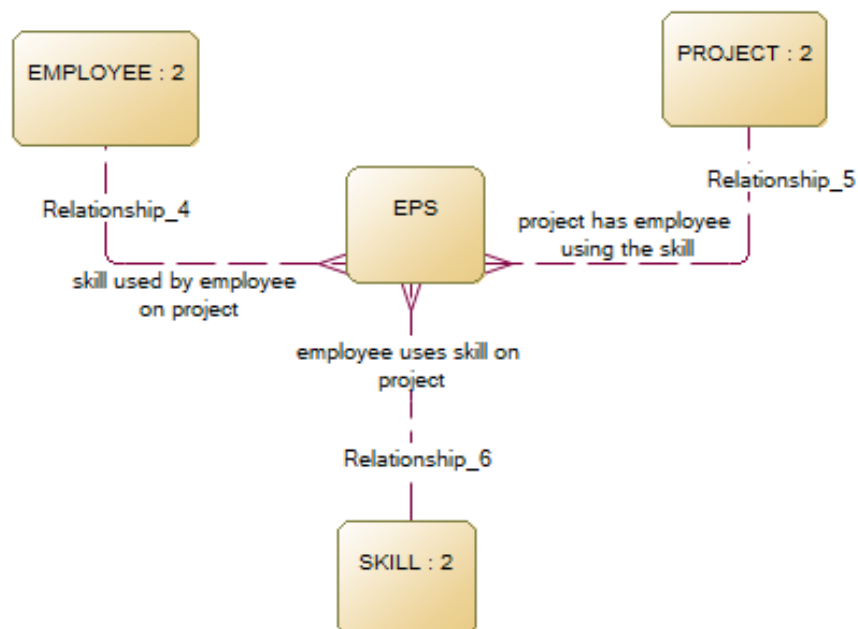
Na kojim projektima Petar koristi koje veštine? Možda interface design na projektu B a database design na projektu A — ili obratno? Ili obe na oba?

Nema dovoljno informacija u bazi.

## POTREBA ZA UVOĐENJEM TERNARNE RELACIJE

*Potrebno je napraviti ternarnu relaciju, ESP (employee, skill, project), između EMPLOYEE, SKILL i PROJECT*

Potrebno je napraviti ternarnu relaciju, ESP (employee, skill, project), između EMPLOYEE, SKILL i PROJECT, prikazanu na slici 1.2. Ona je na dijagramu predstavljena kao tip entiteta ESP zbog nemogućnosti da se u PowerDesigner-u koriste n-arne relacije.



Slika 1.2 Primer ternarne relacije ESP [Izvor: NM IT350 -2020/2021.]

Relacija ESP odražava činjenice kao što su:

1. Petar koristi interface design veštinu na projektu A.
2. Ana koristi database design veštinu na projektu A.
3. Petar koristi interface design veštinu na projektu B.
4. Petar koristi database design veštinu na projektu B.
5. Ana koristi database design veštinu na projektu B.

Ternarna relacija ESP sadrži informacije predstavljene binarnim relacijama :

1. Petar radi na projektu A (iz 1.).
2. Petar radi na projektu B (iz 3. i 4.).
3. Ana radi na projektu A (iz 2.).
4. Ana radi na projektu B (iz 5.).

5. Petar koristi veštinu interface design (iz 1. i 3.).
6. Petar koristi veštinu database design (iz 4.).
7. Ana koristi veštinu database design (iz 2. i 5.).
8. Obe veštine koriste se na oba projekta (iz svih).

## DA LI TERNARNA RELACIJA ISKLJUČUJE UPOTREBU BINARNIH RELACIJA?

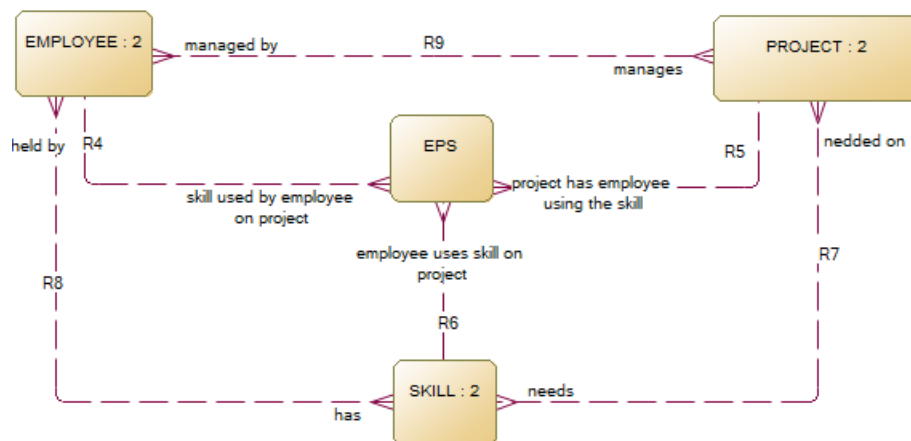
*U slučaju korišćenja ternarne relacije se mogu koristiti i binarne relacije*

Ternarni odnos ne isključuje potrebu za binarnim. **Binarni odnosi su suvišni samo kada odražavaju informaciju koja je podskup informacije koju sadrži ternarni odnos.**

Ako binarni odnos sadrži informaciju koja se razlikuje od one sadržane u ternarnom, binarni odnosi se zadržavaju. Na primer, na slici 1.3. relacija **ESP** ostaje ista kao i ranije. Binarne relacije su drugačije.

Relacija **has / held by** označava da zaposleni poseduje neke veštine. Ova relacija se razlikuje od **ESP** relacije zato što mogu postojati veštine koje zaposleni poseduje ali koje nije koristio u okviru nekog projekta.

Relacija **manages / managed by** označava da zaposleni upravlja projektom. To je potpuno drugačija dimenzija koja se ne može obuhvatiti relacijom **ESP**.



Slika 1.3 Ternarna relacija dopunjena binarnim relacijama [Izvor: NM IT350 -2020/2021.]

## TERNARNA RELACIJA-ODREĐIVANJE KARDINALNOSTI

*Metod određivanja kardinalnosti relacije višeg stepena se razlikuje od određivanja kardinalnosti binarne relacije*

Imamo tri tipa entiteta: AUTOMOBIL (A), KUPAC (K), PRODAVAC (P). Razmotrimo sledeću situaciju.

Postoji ternarna relacija PRODAJA koji povezuje sva tri tipa entiteta: Prodaja: "Kupac K kupio je automobil A od prodavca P."

Pitanje: Ko je prodao automobil Cobra kupcu Sam? (Don)

Između tipova entiteta A, K i P imamo tri binarne relacije:

kupuje		kupuje od		prodaje	
K	A	K	P	P	A
Sam	Mustang	Sam	Don	Don	Cobra
Sam	Cobra	Sam	Sharon	Don	Mustang
Jenn	Mustang	Jenn	Sharon	Sharon	Cobra
Jenn	Cobra			Sharon	Mustang

Slika 1.4 Tri binarne relacije Između tipova entiteta A, K i P [Izvor: NM IT350 -2020/2021.]

1. *kupuje* – Kupac K kupuje automobil A.
2. *kupuje od* – Kupac K kupuje od prodavca P.
3. *prodaje* – Prodavac P prodaje automobil A.

Pitanje: Ko je prodao automobil Cobra kupcu Sam? (Don ili Sharon) ? **Metod određivanja kardinalnosti relacije višeg stepena se razlikuje od određivanja kardinalnosti binarne relacije:**

1. Prekriti sve relacije (linije) koje idu od ternarne relacije (npr. ESP) prema nekom tipu entitetu, osim jedne. (Na primer, prekriti linije iz ESP prema EMPLOYEE i PROJECT.)
2. Za preostali tip entiteta, zapitati se: da li može da postoji najviše jedan entitet tog tipa entiteta, za svaku pojedinačnu kombinaciju preostalih entiteta, ili ih može biti više? (U našem primeru, zapitati se: da li neki zaposleni na nekom projektu može imati najviše jednu veštinu koju primenjuje na projektu, ili može primenjivati veći broj veština?)
  - Ako je odgovor "više", staviti simbol rakljice na liniju koja ulazi u taj tip entiteta;
  - ako je odgovor "1", staviti crticu preko te linije.
3. Ponoviti proces dok se ne ispita svaki tip entiteta u relaciji.

## ▼ Poglavlje 2

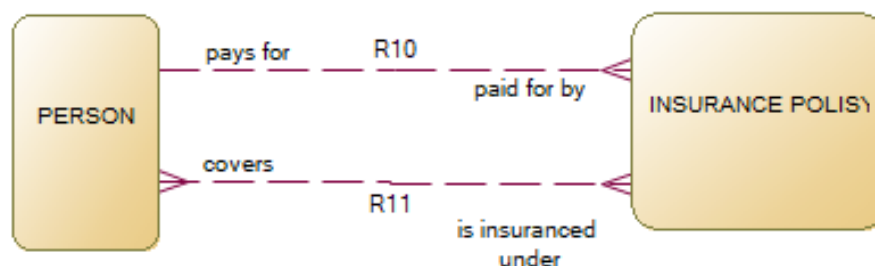
# Višestruke relacije i asocijativni entiteti

## VIŠESTRUKKE RELACIJE

*Dva entiteta mogu da grade više od jedne relacije.*

Višestruka relacije se javljaju u slučaju kada dva entiteta grade više od jednog tipa relacije. To se lako predstavlja ER dijagramom.

Na primer: Posmatrajmo tipove entiteta *PERSON* (osoba) i *INSURANCE POLISY* (polisa osiguranja) i relacije između njih *pays for* (uplaćuje) i *is insured under* (osiguran je po) (slika 2.1)



Slika 2.1 Primer všestrukkih relacija između tipova entiteta [Izvor: NM IT350 -2020/2021.]

Posmatrajmo ove relacije jednu po jednu.

- Jedna osoba uplaćuje nula ili više polisa osiguranja. Jednu polisu osiguranja uplaćuje tačno jedna osoba.
- Jedna osoba je osigurana po osnovu nula ili više polisa osiguranja. Jedna polisa osiguranja pokriva jednu ili više osoba.

Ovo su dve različite relacije. One označavaju dve različite stvari — zato su predstavljeni kao dve različite relacije na ER dijagramu.

## ASOCIJATIVNI TIP ENTITETA

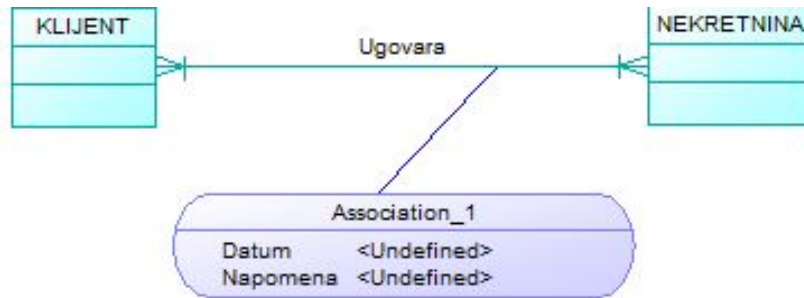
*Relacija sa atributima se predstavlja kao asocijativni tip entiteta.*

Atributi su ranije pridruživani samo tipovima entiteta, ali je moguće da i relacije imaju svoje atribute. Asocijativni entitet je specijalan slučaj relacije sa



maksimalnom kardinalnošću "više prema više" koji zahteva da neki novokreirani tip entiteta pamti attribute te relacije.

*Primer 1: na slici 2.2, prikazana je relacija između tipova entiteta KLIJENT i NEKRETNINA. Atribut Datum će se koristiti da bi se zabeležio datum kada je klijent razmatrao nekretninu, a atribut Komentar će se koristiti da bi se zapamtio svaki komentar koji je klijent imao u vezi sa nekretninom.*



Slika 2.2 Asocijativni entitet -primer [Izvor: NM IT350 -2020/2021.]

*Primer 2: Asocijativni ENROLL entitet na slici 2.3, zavisi od postojanja druga dva entiteta; sastav ENROLL entiteta zasniva se na primarnim ključevima entiteta koji su povezani preko composite entiteta. Asocijativni entitet može sadržavati dodatne attribute koji ne igraju ulogu u povezivanju procesa. Npr., iako entitet mora biti sastavljen barem od primarnih ključeva STUDENT i CLASS, može uključiti i dodatne attribute kao što su ocene, izostanci i druge podatke koji su jedinstveno identifikovani prema performansama studenta u određenom predmetu.*

Table name: STUDENT

STU_NUM	STU_LNAME
321452	Bowser
324257	Smithson

Database name: Ch04\_CollegeTry

Table name: ENROLL

CLASS_CODE	STU_NUM	ENROLL_GRADE
10014	321452	C
10014	324257	B
10018	321452	A
10018	324257	B
10021	321452	C
10021	324257	C

Table name: CLASS

CLASS_CODE	CRS_CODE	CLASS_SECTION	CLASS_TIME	CLASS_ROOM	PROF_NUM
10014	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10018	CIS-220	2	MMF 9:00-9:50 a.m.	KLR211	114
10021	QM-261	1	MMF 8:00-8:50 a.m.	KLR200	114

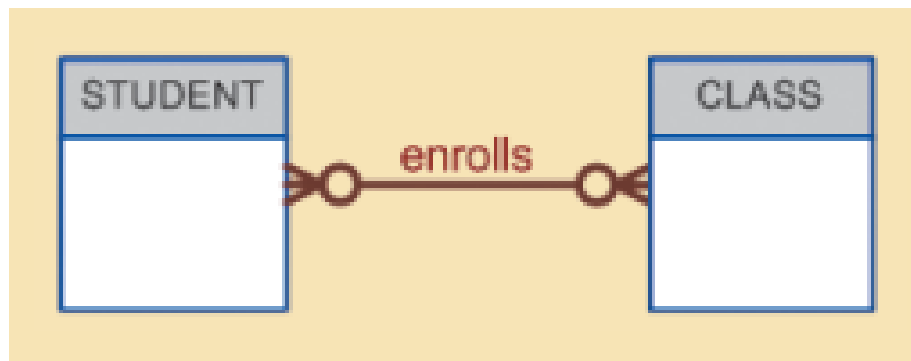
Slika 2.3 Pretvaranje M:N veze u dve 1:M veze. Izvor: [Autor]

## PRETVARANJE M:N VEZE U DVE 1:M VEZE

*Morate znati "1" i "M" strane svake veze, kao i da li su veze obavezne ili opcionalne.*

Napomenimo da je ključ tabele ENROLL (CLASS\_CODE i STU\_NUM) potpuno sastavljen od primarnih ključeva tabela CLASS i STUDENT. Zbog toga u ključnim atributima tabele ENROLL nije moguće imati prazne unose.

Implementacija male baze podataka prikazane na slici 2.4 zahteva jasno definisanje veza. Konkretno, morate znati "1" i "M" strane svake veze, kao i da li su veze obavezne ili opcionalne.



Slika 2.4 Veza M:N između STUDENT i CLASS. Izvor: [1]

Na STUDENT strani M:N veze između STUDENT i CLASS trebalo bi se pojaviti opcioni simbol.

Možemo tvrditi da bi osoba trebala biti klasifikovana kao STUDENT samo ako je upisana na bar jedan predmet. Iz tog razloga, iz čisto konceptualnog gledišta, CLASS je obavezan za STUDENT.

Međutim, kada student bude primljen na fakultet, taj student se još nije (do sada) upisao ni na jedan predmet. Zbog toga, bar na početku, CLASS je opcioni za STUDENT. Ako CLASS nije opcioni za STUDENT - iz perspektive baze podataka - dodela predmeta mora se izvršiti kada je student primljen. Međutim, to nije kako proces zapravo funkcioniše i dizajn baze podataka mora to odražavati. Ukratko, opcionalnost odražava praksu.

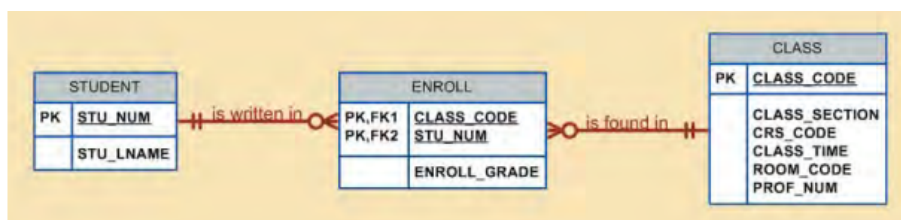
Zbog toga što je M:N veza između STUDENT-a i CLASS-a dekomponovana u dve 1:M veze preko ENROLL-a, opcionalnosti se prenose na ENROLL, slika 2.5. Odnosno, sada je moguće da se predmet ne pojavi u ENROLL-u ako nijedan student nije upisan na taj predmet. Pošto predmet ne mora biti prisutan u ENROLL-u, ENROLL entitet postaje opcioni za CLASS. I pošto se ENROLL entitet kreira pre nego što se bilo koji student upiše na predmet, ENROLL entitet takođe postaje opcioni za STUDENT-a, barem na početku.

## KAKO DOĆI DO ASOCIJATIVNIH TIPOVA ENTITETA?

*Prvo uraditi jednostavniji dijagrami (sa više-prema-više odnosima) a zatim profinjeniji, sa asocijativnim entitetima*

U originalnom ERM-u (ER-Model) opisanom od strane Chena, odnosi ne sadrže attribute. Ako naiđete na M:N odnose, morate stvoriti most između entiteta koji pokazuju takve odnose. Asocijativni entitet se koristi za implementaciju M:N odnosa između dva ili više entiteta. Ovaj asocijativni entitet (poznat i kao kompozitni ili most entitet) sastoji se od primarnih ključeva svakog od entiteta koji treba da budu povezani.

Za asocijativni entitet važi i više: on može da obezbedi bazi podataka podatke o samoj relaciji.



Slika 2.5 Asocijativni entitet u ER dijagramu. Izvor: [Autor]

Kako studenti počinju da se upisuju na svoje predmete, biće uneti u ENROLL entitet. Prirodno je da ako jedan student pohađa više predmeta, taj student će se pojaviti više puta u ENROLL-u. Na primer, obratite pažnju da se u ENROLL tabeli na Slici 2.3, STU\_NUM = 321452 pojavljuje tri puta. S druge strane, svaki student se pojavljuje samo jednom u STUDENT entitetu. (Primitite da STUDENT tabela na Slici 2.3 ima samo jedan unos sa STU\_NUM = 321452.) Zbog toga, na Slici 5, veza između STUDENT-a i ENROLL-a je prikazana kao 1:M, pri čemu je M na strani ENROLL-a.

Kao što možete videti na Slici 2.3, jedan predmet može se pojaviti više puta u ENROLL tabeli. Na primer, CLASS\_CODE = 10014 se pojavljuje dva puta. Međutim, CLASS\_CODE = 10014 se pojavljuje samo jednom u CLASS tabeli kako bi se odrazila veza između CLASS i ENROLL koja je 1:M. Primitite da na Slici 2.5, M se nalazi na strani ENROLL-a, dok se 1 nalazi na strani CLASS-a.

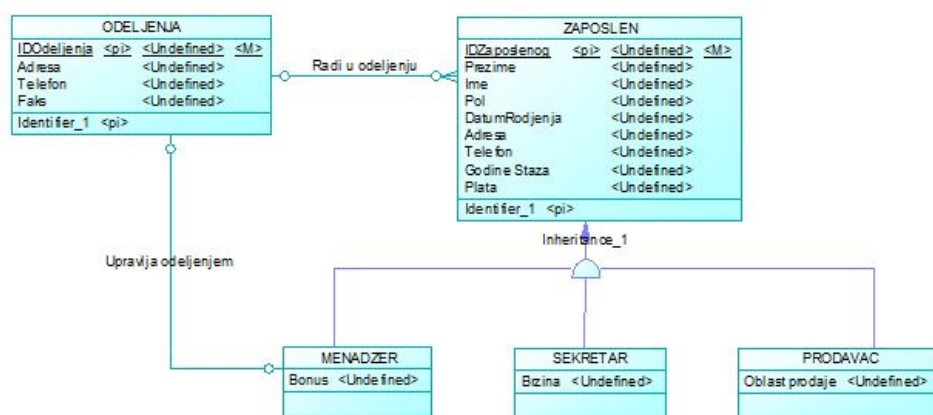
## ▼ Poglavlje 3

# Specijalizacija i generalizacija

## SPECIJALIZACIJA

*Predstavlja prilaz odozgo-nadole definisanjem nadentiteta i podentiteta.*

Specijalizacija je proces maksimiziranja razlika između entiteta nekog tipa entiteta identifikovanjem karakteristika koje ih razlikuju. Ovaj proces predstavlja prilaz odozgo-nadole, definisanjem superklase i njenih podklasa. Podklase se definišu na osnovu njihovih specifičnih karakteristika (atributa) u odnosu na nadklase. Svako podklasi se dodaju atributi koji predstavljaju tu karakteristiku, a zatim se definiše veza između podklasa i nadklase. Razmotrimo specijalizaciju prikazanu na slici 3.1. Nadtip entiteta ZAPOSLEN poseduje podtipove entiteta MENADŽER, SEKRETAR i PRODAVAC.



Slika 3.1 Primer specijalizacije [Izvor: NM IT350 -2020/2021.]

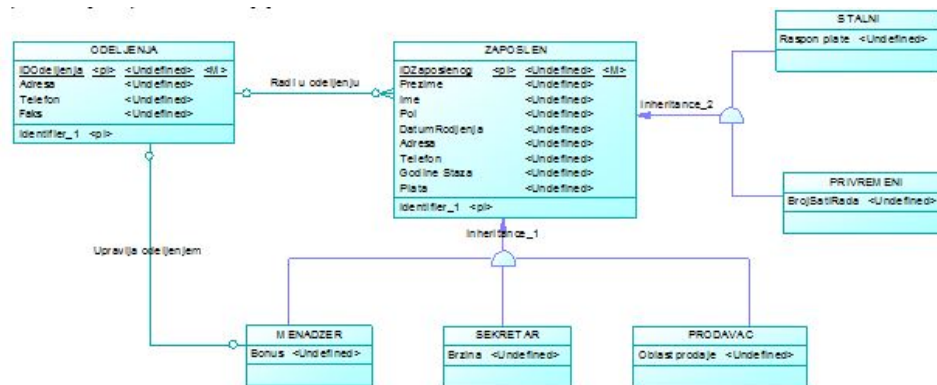
## NASTAVAK O SPECIJALIZACIJI

*Specijalizacija jednog tipa entiteta na osnovu različitih karakteristika*

I nadtip entiteta i podtip entiteta se predstavljaju simbolima za tipove entitete (pravougaonikom). Podtipovi entiteta (specijalizacije) su povezani linijama na kružić koji je povezan sa nadtipom entitetom.

U modelu podataka može postojati situacija u kojoj se vrši specijalizacija jednog tipa entiteta na osnovu različitih karakteristika. Na primer, na slici 3.2. prikazana je jedna takva situacija.

U njoj se vrši specijalizacija tipa entiteta ZAPOSLEN na osnovu dve karakteristike: načina angažovanja i vrste posla koje zaposleni obavljaju



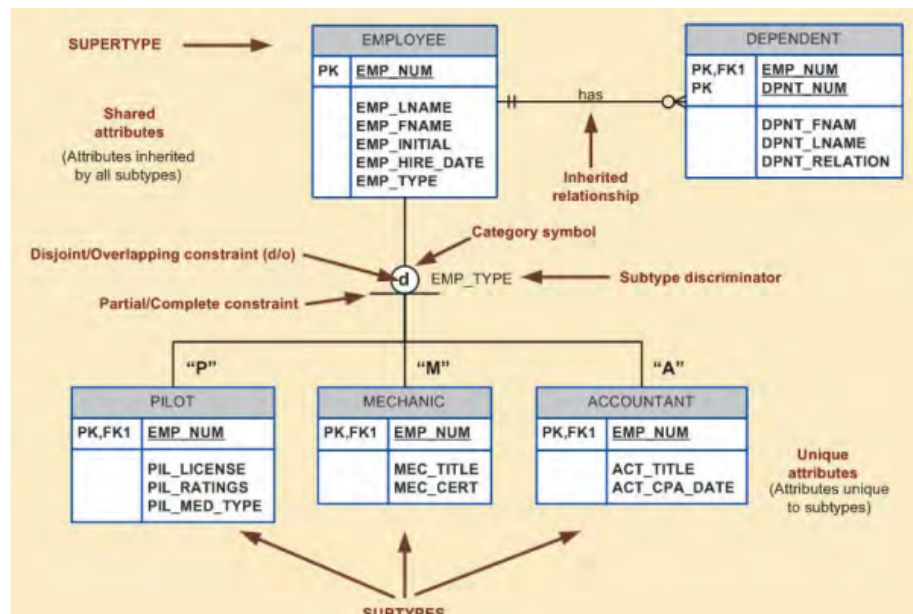
Slika 3.2 Primer višostruke specijalizacije [Izvor: NM IT350 -2020/2021.]

## OSOBINE SPECIJALIZACIJE

*Entitet koji pripada podklasi predstavlja entitet iz realnog okruženja kao i entitet povezan sa njim u superklasi, iako je isti entitet prikazan dva puta*

Entiteti nadtipova i podtipova organizovani su u hijerarhiju specijalizacije koja prikazuje raspored nadređenih entiteta-nadtipova (roditeljskih entiteta) i nižih entiteta-podtipova (dečijih entiteta). Slika 3.3 prikazuje hijerarhiju specijalizacije formiranu od nadtipa ZAPOSLENI i tri podtipa entiteta - PILOT, MEHANIČAR i RAČUNOVOĐA. Hijerarhija specijalizacije odražava odnos 1:1 između ZAPOSLENI i njegovih podtipova. Na primer, pojavljivanje podtipa PILOT je povezano sa jednim primerkom nadtipa ZAPOSLENI, a pojavljivanje podtipa MEHANIČAR je povezano sa jednim primerkom nadtipa ZAPOSLENI.

Odnosi prikazani unutar hijerarhije specijalizacije ponekad se opisuju kao odnosi "je-tipa" (is-a relationships). Na primer, pilot je zaposleni, mehaničar je zaposleni, računovođa je zaposleni. Važno je razumeti da unutar hijerarhije specijalizacije podtip može postojati samo u kontekstu nadtipa i svaki podtip može imati samo jedan nadtip sa kojim je direktno povezan. Međutim, hijerarhija specijalizacije može imati mnogo nivoa odnosa nadtip/podtip, možete imati hijerarhiju specijalizacije u kojoj nadtip ima mnogo podtipova; s druge strane, jedan od podtipova može biti nadtip za druge podtipove nižeg nivoa.



Slika 3.3 Hijerarhija specijalizacije. Izvor: [1]

## RAZLOZI ZA UKLJUČIVANJE SPECIJALIZACIJE U MODEL PODATAKA

*Prvi je da se određeni atributi mogu odnositi na neke a ne i na sve entitete nadtipa; drugi je da u nekim tipovima relacija mogu da učestvuju samo entiteti koji su članovi podtipa.*

Hijerarhija specijalizacije omogućava:

- Podršku nasleđivanju atributa.
- Definisanje posebnog nadtipa atributa poznatog kao diskriminator podtipa.
- Definisanje ograničenja koja se odnose na razdvojene/preklapajuće podtipove i potpune/delimične podtipove.

Svojstvo nasleđivanja omogućava podtipu entiteta da nasleđuje attribute i odnose od nadtipa. Kao što je ranije objašnjeno, nadtip sadrži one attribute koji su zajednički svim njegovim podtipovima. Nasuprot tome, podtipovi sadrže samo one attribute koji su jedinstveni za taj podtip. Na primer, Slika 3.3 ilustruje da piloti, mehaničari i računovođe nasleđuju broj zaposlenog, prezime, ime, srednje slovo, datum zaposlenja itd. od entiteta ZAPOSLENI. Međutim, Slika 3.3 takođe pokazuje da piloti imaju attribute koji su jedinstveni; isto važi i za mehaničare i računovođe. Jedna važna karakteristika nasleđivanja je da svi podtipovi entiteta nasleđuju atribut primarnog ključa od svog nadtipa. Napomenimo da je atribut EMP\_NUM primarni ključ za svaki od podtipova.

Podtipovi entiteta nasleđuju sve odnose u kojima učestvuje nadtip entitet. Na primer, Slika 3.3 pokazuje da nadtip entitet ZAPOSLENI učestvuje u odnosu 1:M sa entitetom ZAVISNI. Kroz nasleđivanje, svi podtipovi takođe učestvuju u tom odnosu. U hijerarhijama specijalizacije sa

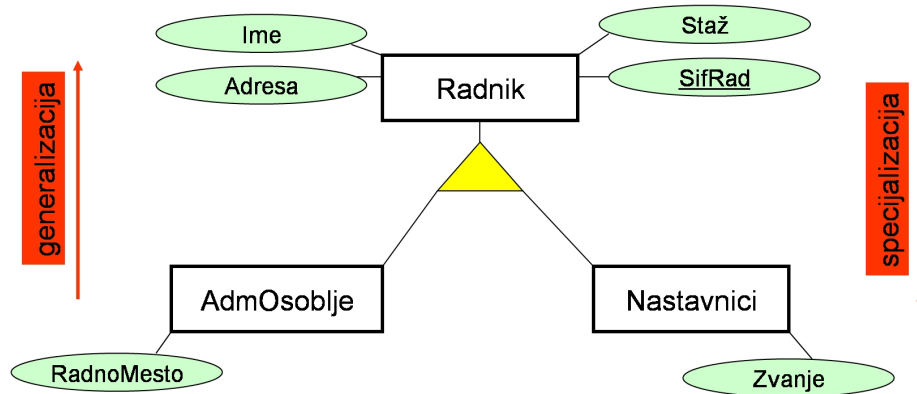
više nivoa nadtipova/podtipova, podtip nižeg nivoa nasleđuje sve atribute i odnose od svih njegovih nadređenih nadtipova.

## GENERALIZACIJA

*Proces minimizacije razlika između entiteta identifikovanjem zajedničkih karakteristika.*

Možemo razmišljati i o obrnutom procesu apstrakcije u kojem potiskujemo razlike između nekoliko tipova entiteta, identifikujemo njihove zajedničke karakteristike i generalizujemo ih u jednu superklasu čiji su originalni tipovi entiteta posebne podklase.

Na primer, možda biste identifikovali više tipova muzičkih instrumenata: klavir, violina i gitara. Koristeći pristup generalizacije, mogli biste identifikovati nadtip entitet "instrument sa žicama" koji će sadržati zajedničke karakteristike ovih različitih podtipova.



Slika 3.4 Generalizacija ili specijalizacija. Izvor: [Autor]

## HIJERARHIJA SPECIJALIZACIJE I REŠETKA SPECIJALIZACIJE

*U hijerarhiji specijalizacije svaka podklasa učestvuje kao podklasa u samo jednoj klasa/podklasa relaciji; U rešetki specijalizacije, podklasa može biti podklasa u više klasa/podklasa relacije*

Nadklasa predstavlja entitet koji uključuje potpuno odvojene podentitete koje treba predstaviti u modelu. Podklasa je entitet koji ima jedinstvenu ulogu i deo je nadklase. **Veza između ovih entiteta naziva se nadklasa/podklasa.**

Svaki član podklase je ujedno i član nadklase. Drugim rečima, podklasa je član nadklase osim što ima drugačiju ulogu. Veza između na klase i podklase je jedan-prema-jedan. **Nadklase i podklase mogu koristiti da bi se izbeglo opisivanje sličnih tipova entiteta korišćenjem različitih atributa u jednom istom tipu entiteta.**

Ukoliko se atributi koje imaju svi zaposleni kao i atributi koje imaju samo neke kategorije zaposlenih postave u jedan tip entiteta, u bazi podataka bi postojao veliki broj Null vrednosti.

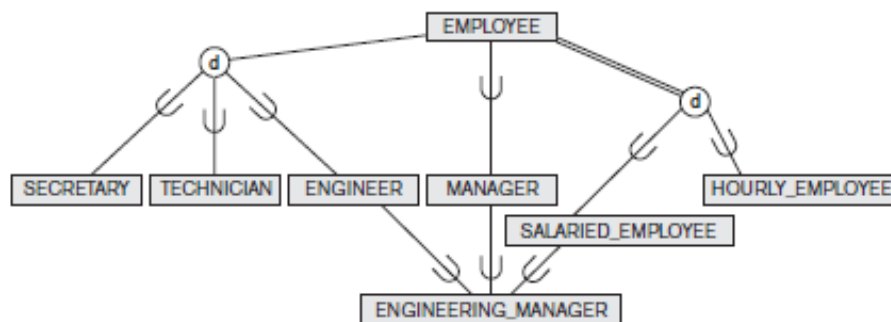


Zajednički atributi smeštaju se u nadklasu (na primer Prezime i Ime), a atributi za specifične kategorije zaposlenih se smeštaju u podklasu.

Podklasa je takođe entitet i on sam može imati svoje podklase. Ovako definisane nadklase i podklase formiraju hijerarhiju. Ovakve hijerarhije se u literaturi često nazivaju i hijerarhija specijalizacije.

Na primer, na slici 3.5. ENGINEER je podklasa EMPLOYEE i takođe je superklasa ENGINEERING\_MANAGER; ovo predstavlja realno ograničenje da svaki ENGINEERING\_MANAGER mora biti inženjer.

Hijerarhija specijalizacije ima ograničenje da svaka podklasa učestvuje kao podklasa u samo jednoj klasa / podklasa relaciji; to jest, svaka podklasa ima samo jednog roditelja, što rezultira strukturom stabla ili strogom hijerarhijom. Nasuprot tome, za rešetku specijalizacije, podklasa može biti podklasa u više od jedne klasa / podklasa relacije. Dakle, slika 3.5. je rešetka.



Slika 3.5 Rešetka specijalizacije pod klase ENGINEERING\_MANAGER. [Izvor: NM IT350 -2020/2021.]

## UNIJA ILI KATEGORIJA

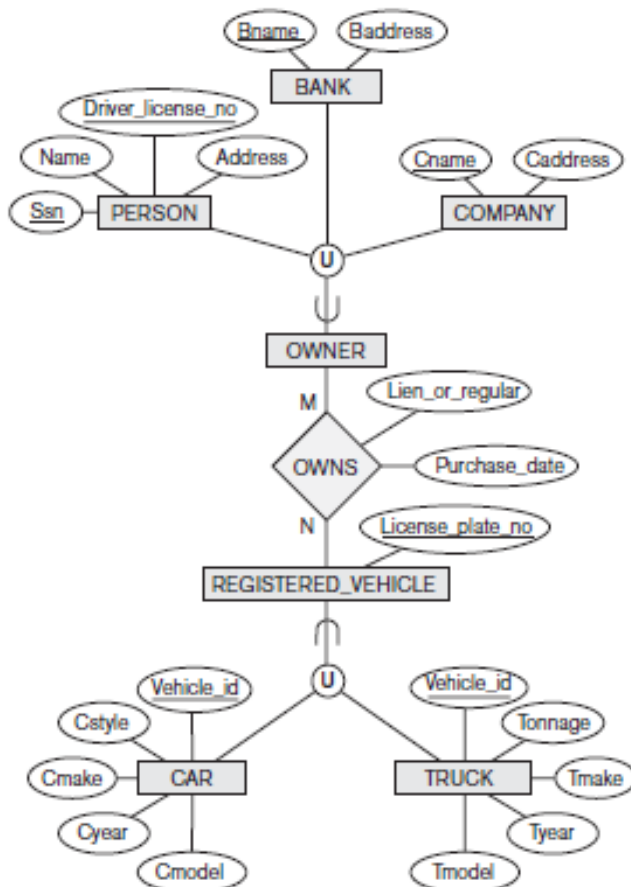
*Nastaje u slučajevima kada u jednoj relaciji superklase / podklase postoji više od jedne superklase. Podklasa u tom slučaju sadrži podskup UNIJE-e različitih superklasa.*

U svim vezama superklase / podklase koje smo do sada videli, postoji samo jedna superklasa. Zajednička podklasa kao što je ENGINEERING\_MANAGER u rešetki specijalizacije na slici 3.5 je podklasa u tri različite relacije superklase / podklase, gde svaka od tri relacije ima jednu superklasu. Međutim, ponekad je neophodno predstaviti jednu relaciju superklase / podklase sa više od jedne superklase, gde superklase predstavljaju različite tipove entiteta. U tom slučaju, podklasa će predstavljati skup objekata koji je podskup UNIJE-e različitih tipova entiteta; takvu podklasu nazivamo unija ili kategorija.

Na primer, pretpostavimo da imamo tri tipa entiteta: PERSON, BANK i COMPANY. U bazi podataka za registraciju motornih vozila, vlasnik vozila može biti osoba, banka (sa založnim pravom na vozilu) ili kompanija. Potrebno je da kreiramo klasu (kolekciju entiteta) koja uključuje entitete sve tri vrste da bi igrali ulogu vlasnik vozila. Za ovu svrhu može se kreirati kategorija (unija) OWENER koji je unija pod klasa od tri skupa entiteta COMPANY, BANC i PERSON. (slika 3.6)



Kategoriju u ER dijagramu prikazujemo kao što je prikazano na slici 3.6 Superklase COMPANY, BANK i PERSON su povezane krugom sa simbolom U koji označava operaciju unije. Na slici 3.6. imamo dve kategorije: OWNER, koji je podklasa unije COMPANY, BANK i PERSON; i REGISTERED\_VEHICLE, koji je podklasa unije CAR i TRUCK.



Slika 3.6 Dve kategorije (unije): OWNER i REGISTERED\_VEHICLE. [Izvor: NM IT350 -2020/2021.]

## ▼ Poglavlje 4

# ID zavisni i slabi tipovi entiteta

## ID ZAVISNI TIPOVI ENTITETA

*ID zavisni tipovi entiteta su entiteti čiji identifikator uključuje identifikator drugog entiteta.*

Osnovni koncepti ER modela podataka rešavaju probleme za većinu šema baza podataka. Međutim, obzirom na sve veći broj sistema za upravljanje bazama podataka, kao i sve veći broj aplikacija koje rade sa bazama podataka i CASE alata za modeliranje podataka, pojavili su se mnogo veći zahtevi od onih koji su postojali u tradicionalnim administrativnim aplikacijama. Osnovni koncepti ER modela ne pružaju mogućnost za modelovanje tih zahteva. Zato se pojavila potreba da se osnovni ER model proširi i unapredi tako da omogući modelovanje i novih zahteva.

Među ostalim konceptima koji se koriste u ER modeliranju biće objašnjeni:

1. **ID zavisni tipovi entiteta i**
2. **Slabi tipovi entiteti.**

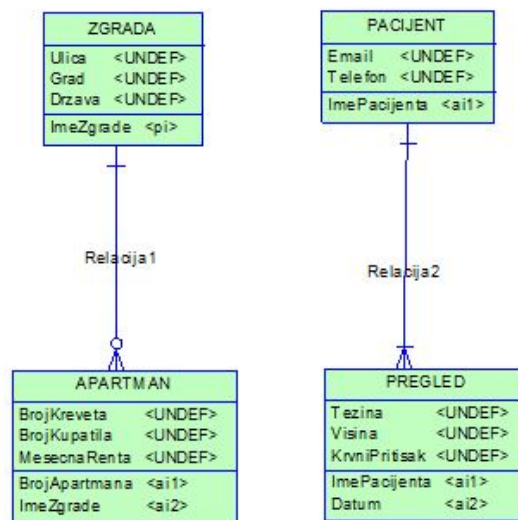
ID zavisni tipovi entiteta su entiteti čiji identifikator uključuje identifikator drugog entiteta.

*Na primer: razmotrimo slučaj apartmana (sobe) koji koriste studenti prilikom studiranja. Identifikator takvog entiteta je složen (BrojApartmana, ImeZgrade), gde je ImeZgrade identifikator tipa entiteta ZGRADA a BrojApartmana sam po sebi nije dovoljan da nekome kažete gde živite.*

Na slici 4.1 su prikazana dva primera ID zavisnih entiteta.

U oba ova slučaja ID zavistan tip entiteta ne može da postoji ukoliko roditelj (tip entitet od koga zavisi) takođe ne postoji. Tako je minimalna kardinalnost ID zavisnog entiteta uvek jedan. S druge strane, da li roditelj mora da ima ID zavistan entitet zavisi od potreba aplikacije.

Na slici 4.1 tip entitet APARTMAN je neobavezan dok je tip entiteta PREGLED obavezan.



Slika 4.1 Primeri ID zavisnih tipova entiteta [Izvor: NM IT350 -2020/2021.]

## SLABI TIPOVI ENTITETA

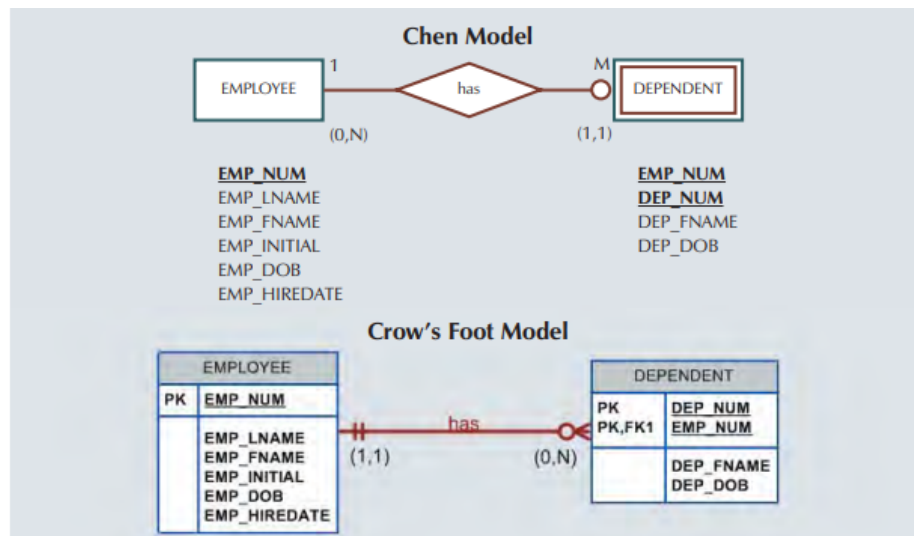
*Slabi tipovi entiteti su oni čije postojanje zavisi od prisustva drugih entiteta.*

Slabi tipovi entiteta su entiteti čije postojanje zavisi od prisustva drugih entiteta. Slab tip entiteta normalno ima parcijalni ključ, koji je atribut koji može jedinstveno identifikovati slab entitet koji je povezan sa entitetom roditelja.

Na primer, bar deo ključa entiteta *DEPENDENT* prikazanog na slici 4.2, je nasleđen od entiteta *EMPLOYEE*:

*EMPLOYEE* (*EMP\_NUM*, *EMP\_LNAME*, *EMP\_FNAME*, *EMP\_INITIAL*, *EMP\_DOB*, *EMP\_HIREDATE*)  
*DEPENDENT* (*EMP\_NUM*, *DEP\_NUM*, *DEP\_FNAME*, *DEP\_DOB*)

Primarni ključ entiteta *DEPENDENT* (*EMP\_NUM*, *DEP\_NUM*) sastoji se od delova ključa entiteta *EMPLOYEE* (*EMP\_NUM*) i dodatnog atributa *DEP\_NUM*. Ovo osigurava vezu između *DEPENDENT* entiteta i odgovarajućeg *EMPLOYEE* entiteta i obezbeđuje jedinstveno identifikovanje zavisnih entiteta za svakog zaposlenog.



Slika 4.2 Slabi entitet u ER dijagramu. Izvor: [1]

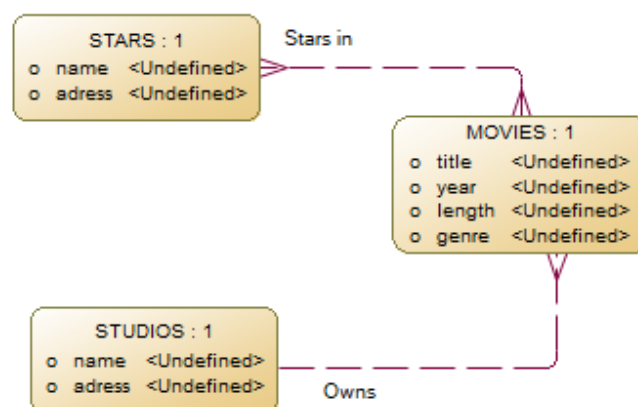
## ▼ Poglavlje 5

# Primeri naprednih ER dijagrama

## PRIMER ER DIJAGRAMA ZA BAZU FILMOVA

*E/R dijagram predstavlja konceptualni model baze podataka o filmovima*

Na slici 5.1 je prikazan ER dijagram koji predstavlja jednostavnu bazu podataka o filmovima. Tipovi entiteta su: MOVIES, STARS i STUDIOS.



Slika 5.1 ER dijagram koji predstavlja jednostavnu bazu podataka o filmovima [Izvor: NM IT350 -2020/2021.]

Tip entiteta MOVIES ima četiri atributa: title, year, length i genre. Druga dva tipa entiteta STARS i STUDIOS imaju iste attribute: name i address.

Takođe vidimo da je:

1. Stars-in relacija koja povezuje svaki film sa zvezdom (STARS) tog filma. Relacija takođe povezuje zvezdu filma sa filmom u kojem se ona pojavljuje.
2. Owns relacija povezuje svaki film sa studijom koji je vlasnik filma.

Instanca Stars-in relacije može biti vizualizovana kao tabela sa sledećim parovima:

<i>Movies</i>	<i>Stars</i>
Basic Instinct	Sharon Stone
Total Recall	Arnold Schwarzenegger
Total Recall	Sharon Stone

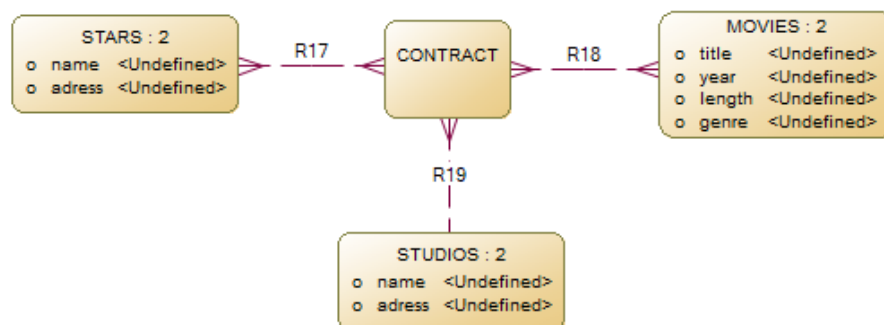
Slika 5.2 Podaci baze podataka [Izvor: NM IT350 -2020/2021.]

Članovi skupa relacija su redovi tabela. Npr, (Basic Instinct, Sharon Stone) je torka u pojavljivanjima relacija relacije Stars-in.

## TERNARNE RELACIJE

### Primer ternarne relacija sa tri tipa entiteta

Kao što je već rečeno, u CASE alatu PowerDesigner se ne mogu predstaviti n-arne relacije što je opravdano zbog činjenice da se u nekim modelima kao što je UML ili ODL mogu koristiti samo binarne relacije. Zbog toga se svaka veza u modelu koja povezuje više od dva tipa entiteta konvertuje u skup binarnih relacija.



Slika 5.3 N-arna relacija (asocijativni entitet) CONTRACT koji predstavlja vezu tipova entiteta STUDIOS, STARS i MOVIES [Izvor: NM IT350 -2020/2021.]

Tako je na slici 5.3, n-arna relacija predstavljena kao tip entiteta CONTRACT koja povezuje tipove entiteta STUDIOS, STARS i MOVIES. Ovaj tip entitet pokazuje da studio ima ugovor sa određenim zvezdama da igraju u određenim filmovima. Generalno, ovaj tip entiteta se može shvatiti kao skup pojavljivanja torki čije su komponente entiteti tipova entiteta koji učestvuju u tipu entiteta CONTRACT. Tako relacija Contract može biti opisana torkama sa strukturom (studio, star, movie).

Sa slike se zaključuje da za pojedinačnu zvezdu i film postoji samo jedan studio sa kojim zvezda ima ugovor da snimi film. Studio može imati ugovor sa više zvezda za dati film a zvezde mogu sklopiti ugovor sa jednim studijom za više od jednog filma.

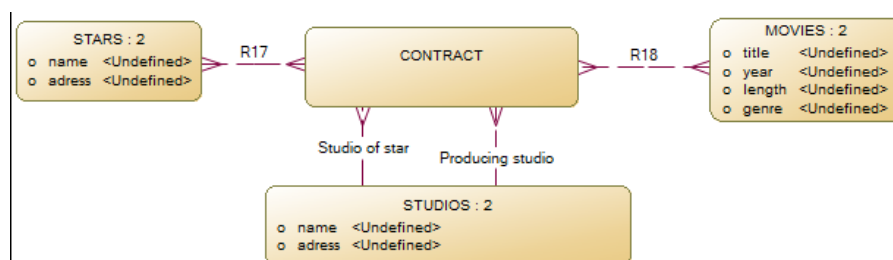
## KVATERNARNE RELACIJE

### Primer kvaternarne relacija sa četiri tipa entiteta

Na slici 5.4 je složenija verzija relacije CONTRACT iz primera 3. Sada relacija CONTRACT obuhvata dva studija, zvezdu i film. Namera je da se pokaže da se jedan studio u ugovoru generalno odnosi na studio date zvezde ali da takođe u ugovoru može biti i studiju koji producira film. Tako je relacija opisana sa četiri torke u formi: (studio1, studio2, star, movie).

Relacija CONTRACT je i u ovom slučaju zamenjena tipom entiteta koji se takođe može nazvati CONTRACT. Kao što se sa slike 5.4 može videti, tip entiteta takođe učestvuje u četiri relacije.

Relacija se može tumačiti na sledeći način: Za datu zvezdu i studio koji proizvodi filmove, može biti samo jedan studio koji je „vlasnik“ te zvezde (pretpostavka je da je zvezda u ugovoru sa tačno jednim studijom). Slično, jedan i samo jedan studio proizvodi dati film pa tako za datu zvezdu i studio te zvezde možemo odrediti jedinstven studio u kojem se film producira.



Slika 5.4 Relacija Contracts - složenija verzija [Izvor: NM IT350 -2020/2021.]

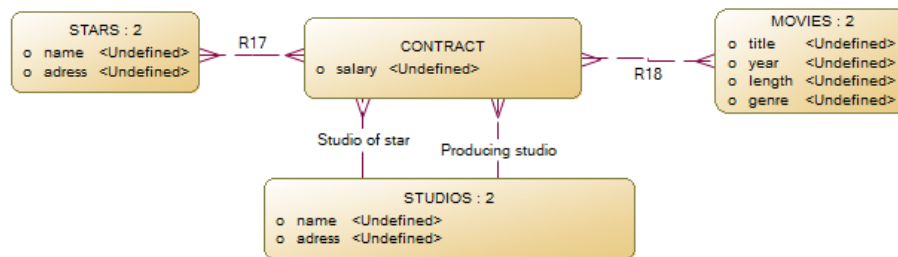
Napomenimo, da nam je u ovom slučaju, potreban samo jedan od entiteta kako bismo odredili jedinstvenu torku relacije Contract - npr. potrebno nam je samo da znamo film kako bismo odredili u kom je studiju on proizveden.

Takođe treba napomenuti da tip entiteta CONTRACT može imati sopstvene atribute.

## RELACIJA SA ATRIBUTIMA (ASOCIJATIVNI ENTITETI)

### Primer relacija sa atributima

Uzmimo kao primer prethodnu n-arnu relaciju koja predstavlja ugovor (CONTRACT) između zvezde, studija i filma (ona je kao što je objašnjeno predstavljena tipom entiteta CONTRACT na slici 5.4). U tom slučaju možemo poželeći da zapamtimo i zaradu (salary) koja ide uz ugovor. Međutim, zaradu ne možemo pridružiti tipu entiteta STARS, zvezda može imati više zarada za različite filmove. Takođe, zarada se ne može pridružiti ni tipu entiteta STUDIOS (oni mogu plaćati različite zarade različitim zvezdama na filmu - različite zvezde u filmu mogu primati različite zarade). Međutim, zaradu možemo pridružiti tipu entiteta CONTRACT što je prikazano na slici 5.5.



Slika 5.5 Primer relacije sa atributima koja je predstavljena kao tip entiteta [Izvor: NM IT350 -2020/2021.]

## INSTANCE ER DIJAGRAMA

*Kako se baza ne implementira u ER modelu, već samo dizajnira, instanca nikada ne postoji u smislu da instanca relacije postoji u DBMS.*

ER dijagrami su notacije za opis konceptualne šeme baze podataka. Možemo zamisliti da baza podataka koja je opisana E/R dijagramom sadrži podatke-instance baze. Kako se baza ne implementira u ER modelu, već samo dizajnira, instance relacija nikada ne postoji u DBMS. Međutim, često je korisno vizualizovati bazu koja se dizajnira onako kako bi ona trebalo da postaji. Za svaki tip entiteta, instanca baze podataka će imati određeni skup pojavljivanja entiteta. Svako od tih pojavljivanja entiteta ima određene vrednosti za svoje atribute. Za relaciju R koja povezuje N entiteta  $E_i$  se takođe može zamisliti da ima instance koje se sastoje od određenog skupa torki  $(e_1, e_2, \dots, e_N)$  gde je svako  $e_i$  izabrano od pojavljivanja entiteta koji se nalaze u postojećim instancama skupa entiteta  $E_i$ .

Svaku takvu torku posmatramo kao „povezan sa relacijom,, R. Taj skup torki se naziva skup relacija R. Često je korisno vizualizovati skup relacija kao tabelu ili relaciju. Međutim, torke skupa relacija nisu zaista torke u relaciji baze podataka obzirom da su njihove komponente entiteti a ne primitivni tipovi kao što su stringovi ili integeri. Kolona tabele je označena imenom skupa pojavljivanja entiteta obuhvaćenih relacijom, i svaka lista povezanih entiteta predstavlja jedan red tabele. Kao što ćemo videti, kada konvertujemo relaciju ER modela u relaciju baze podataka, rezultujuća relacija ER modela nije ista kao skup relacija u bazi podataka.



## ▼ Poglavlje 6

# Primer konceptualnog modela baze podataka COMPANY

## ZAHTEVI ZA PROJEKTOVANJEM KONCEPTUALNOG MODELA BAZE PODATAKA COMPANY

*Projektovana baza podataka treba da zadovolji definisane zahteve*

U ovom odeljku opisujemo aplikaciju baze podataka, nazvanu COMPANY, koja služi za ilustraciju osnovnih koncepata ER modela i njihove upotrebe u dizajnu šeme. Najpre će biti specificirani zahtevi za podatke baze podataka, a zatim ćemo kreirati njenu konceptualnu šemu korak po korak uvodeći koncepte modelovanja ER modela. Baza podataka COMPANY vodi evidenciju o zaposlenima, odeljenjima i projektima kompanije.

1. **Kompanija je organizovana u odeljenja.** Svaki odeljenje ima jedinstveno ime, jedinstveni broj i određenog zaposlenog koji upravlja odeljenjem. Pratimo datum početka kada je taj zaposleni počeo upravljati odeljenjem. Odeljenje može imati nekoliko lokacija.
2. **Odeljenje kontroliše brojne projekte,** od kojih svaki ima jedinstveno ime, jedinstveni broj i jednu lokaciju.
3. **Čuvamo ime svakog zaposlenog, broj socijalnog osiguranja, adresu, platu, pol i datum rođenja.** Zaposleni je raspoređen u jedno odeljenje, ali može raditi na nekoliko projekata, koji nisu nužno pod kontrolom istog odeljenja. Mi pratimo trenutni broj sati nedeljno koliko zaposleni radi na svakom projektu. Pratimo i direktnog supervizora svakog zaposlenog (koji je drugi zaposleni).
4. Želimo da **pratimo zavisne članove svakog zaposlenog za potrebe osiguranja.** Imamo ime, pol, datum rođenja i vezu sa zaposlenim.

## POČETNI IDEJNI PROJEKAT BAZE PODATAKA O KOMPANIJI

*Sastoji se od četiri tipa entiteta*

Na osnovu zahteva, možemo definisati tipove entiteta za bazu podataka COMPANY. Nakon što definišemo tipove entiteta i njihove atribute, usavršićemo naš projekat uvođenjem koncepta relacije. Prema navedenim zahtevima, možemo identifikovati četiri tipa entiteta: 1. Tip entiteta DEPARTMENT sa atributima:

- Name, (atributi ključa)
- Number, (atributi ključa)

- Locations ( višestruki atribut)
- Manager i
- Manager\_start\_date.

2. Tip entiteta PROJECT sa atributima:

- Name, (atributi ključa)
- Number, (atributi ključa)
- Location i
- Controlling\_department.

3. Tip entiteta DEPENDANT sa atributima:

- Employee,
- Dependent\_name,
- Sex,
- Birth\_date i
- Relationship (prema EMPLOYEE).

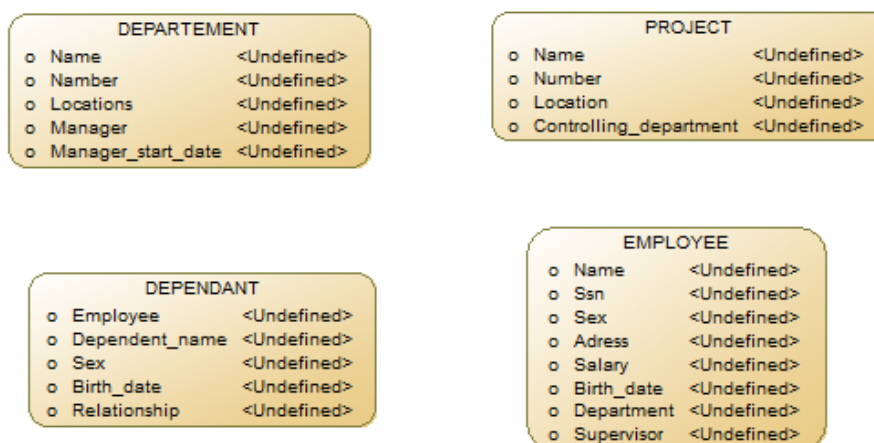
4. Tip entiteta EMPLOYEE sa atributima:

- Name, složeni atribut
- Ssn,
- Sex,
- Address, složeni atribut
- Salary,
- Birth\_date,
- Department
- Supervisor.

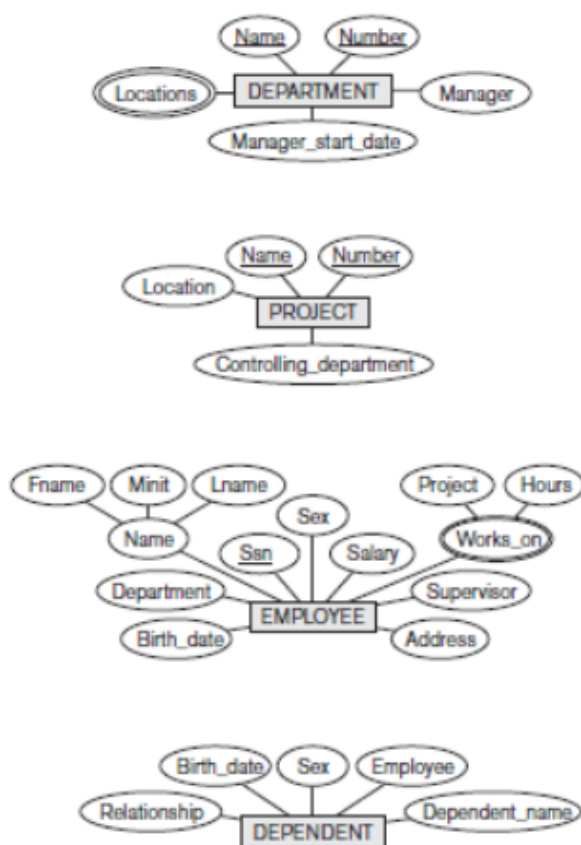
Do sada nismo predstavili činjenicu da zaposleni može da radi na više projekata, niti smo predstavili broj sati nedeljno zaposlenog na svakom projektu. Ova karakteristika je navedena kao deo trećeg zahteva, i može biti predstavljena višestrukim kompozitnim atributom nazvanim WORKS\_ON sa jednostavnim komponentama (Project, Hours) u tipu entiteta EMPLOYEE . Alternativno, može se predstaviti kao višestruki kompozitni atribut pod nazivom WORKERS sa jednostavnim komponentama (Employee, Hours) u tipu entiteta PROJECT. Izabrali smo prvu alternativu. Atribut Name tipa entiteta EMPLOYEE je prikazano kao složeni atribut, verovatno nakon konsultacija sa korisnicima.

## TIPOVI ENTITETA ZA BAZU PODATAKA COMPANY

*Tipovi entiteta predstavljeni korišćenjem notacija u PowerDesigner-u i Chen-ove notacije*



Slika 6.1 Tipovi entiteta predstavljeni korišćenjem notacija u PowerDesigner [Izvor: NM IT350 -2020/2021.]



Slika 6.2 Tipovi entiteta predstavljeni korišćenjem Chen-ove notacije [Izvor: NM IT350 -2020/2021.]

## UNAPREĐENJE ER MODELA ZA BAZU PODATAKA COMPANY

*Uvedene su sledeće vrste relacija:*

Sada možemo poboljšati dizajn baze podataka uvođenjem relacija. Kardinalnosti relacija su određene na osnovu navedenih zahteva. Ako se neka kardinalnost relacije ne može odrediti iz zahteva, korisnici moraju biti dodatno ispitani kako bi se utvrdila ova strukturna ograničenja. U našem primeru postoje sledeće relacije:

1. MANAGES, tip relacije 1: 1 između EMPLOYEE i DEPARTMENT. Učešće zaposlenih je opciono. Učešće odeljenja nije jasno iz zahteva. Ispitujemo korisnike koji kažu da odeljenje mora imati menadžera u svakom trenutku, što podrazumeva obavezno učešće.
2. WORKS\_FOR, tip relacije 1: N između DEPARTMENT and EMPLOYEE. Oba učešća su obavezna.
3. CONTROLES, tip relacije 1: N između DEPARTMENT i PROJECT. Učešće projekta je obavezno, dok je za odeljenje određeno kao opciono, nakon konsultacija sa korisnicima ukazuje da neka odeljenja ne mogu da kontrolišu projekte.
4. SUPERVISION, tip relacije 1: N između EMPLOYEE (u ulozi supervizora) i EMPLOYEE (u ulozi nadgledanog). Oba učešća su utvrđena kao opcionalna, nakon što su korisnici naveli da nije svaki zaposleni supervizor, niti da svaki zaposleni ima supervizora.
5. WORKS\_ON, određen kao tip relacije M: N sa atributom Hours, nakon što su korisnici naveli da projekat može imati više zaposlenih na njemu. Oba učešća su određena kao obavezna.
6. DEPENDENTS\_OF, tip relacije 1: N između EMPLOYEE i DEPENDENT, čime je ujedno i identifikovan odnos za slab tip entiteta DEPENDENT. Učestvovanje EMPLOYEE je opciono, dok je učešće DEPENDENT obavezno.

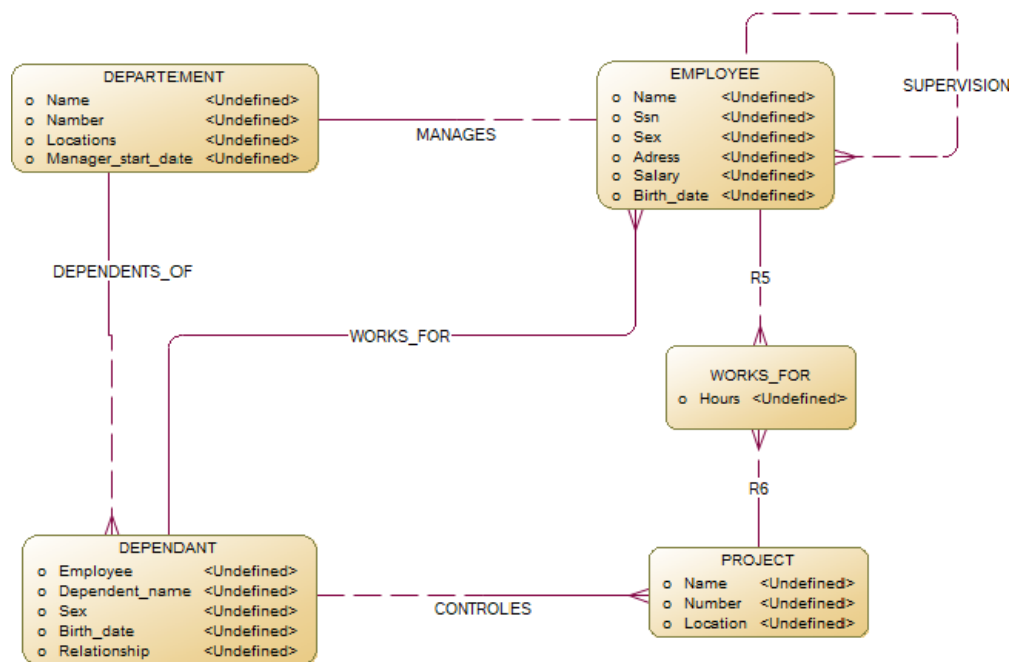
Nakon specificiranja gornjih šest tipova relacija, uklanjamo iz tipova entiteta odgovarajuće attribute. Ovo uključuje:

1. Manager iz DEPARTMENT-a;
2. Controlling\_department iz PROJECT;
3. Department, Supervisor, and Works\_on iz EMPLOYEE;

Važno je imati najmanju moguću redundantnost kada dizajniramo konceptualnu šemu baze podataka. Ako se želi neka redundantnost na nivou skladištenja ili na nivou prikaza korisnika, može se uvesti kasnije.

## KONCEPTUALNI MODEL BAZE PODATAKA COMPANY

*Urađen je korišćenjem notacija iz PowerDesignera i prikazan na slici 6.3.*



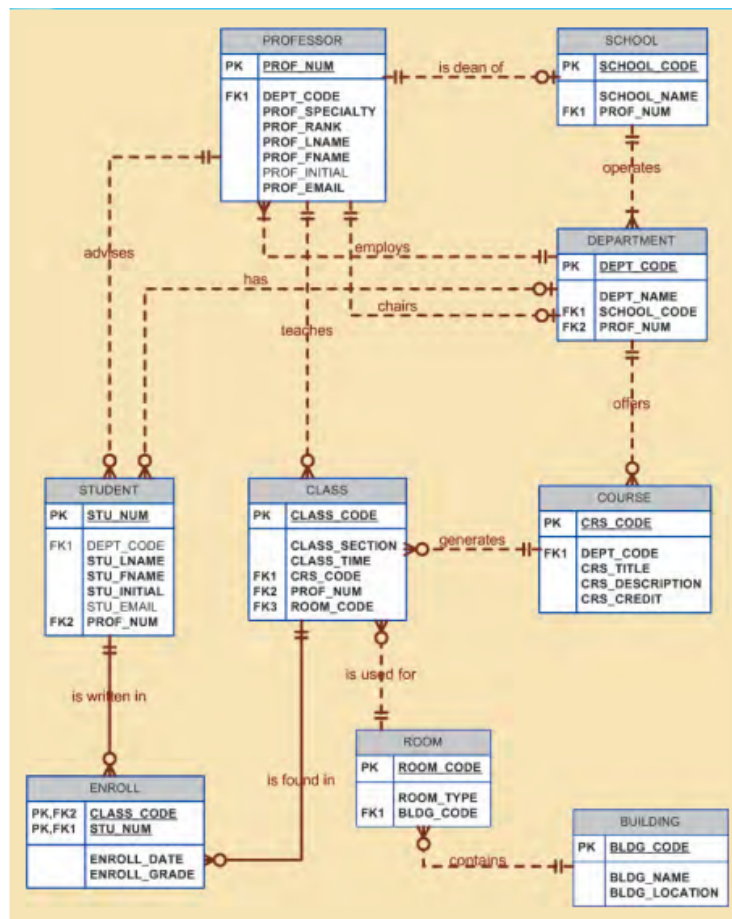
Slika 6.3 Konceptualni model baze podataka COMPANY [Izvor: NM IT350 -2020/2021.]

## ▼ Poglavlje 7

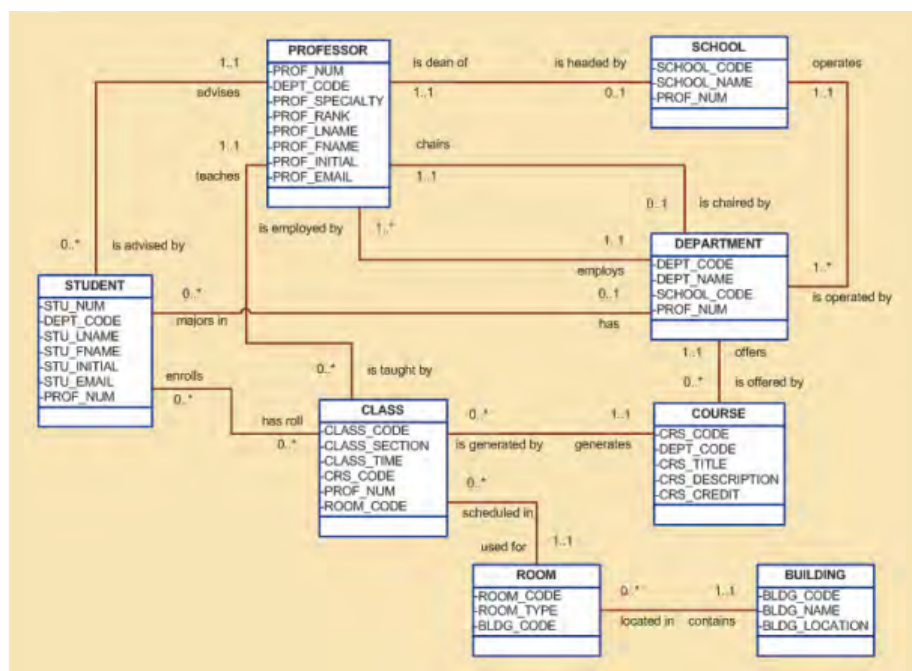
# Primer konceptualnog modela baze Tiny College

## ŠEMA BAZE PODATAKA TINY COLLEGE

*Na osnovu date šeme baze podataka Tiny College, mogu se specificirati zahtevi.*



Slika 7.1 Završeni ERD (Entitetski dijagram sa vezama) za "Tiny College". Izvor: [1]



Slika 7.2 Konceptualni UML dijagram klasa za "Tiny College". Izvor: [1]

## SAŽETAK KOMPONENTI ERM-A, IMENOVANJE ENTITETE I NJIHOVIH VEZA

*Na osnovu date šeme baze podataka Tiny College daje se sažetak komponenti ERM-a*

Entitet	Relacija	Kardinalnost	Entitet
SCHOOL	operates	1:M	DEPARTMENT
DEPARTMENT	has	1:M	STUDENT
DEPARTMENT	employs	1:M	PROFESSOR
DEPARTMENT	offers	1:M	COURSE
COURSE	generates	1:M	CLASS
PROFESSOR	is dean of	1:1	SCHOOL
PROFESSOR	chairs	1:1	DEPARTMENT
PROFESSOR	teaches	1:M	CLASS
PROFESSOR	advises	1:M	STUDENT
STUDENT	enrolls in	M:N	CLASS
BUILDING	contains	1:M	ROOM
ROOM	is used for	1:M	CLASS

Slika 7.3 Sažetak komponenti ERM-a, imenovanje entitete i njihovih veza. Izvor: [1]

### Diskusija:

Tiny College (TC) je podeljen na nekoliko škola: školu biznisa, školu umetnosti i nauka, školu obrazovanja i školu primenjenih nauka, slika 7.3. Svaka škola je upravljana dekanom koji je profesor. Svaki profesor može biti dekan samo jedne škole, i profesor nije obavezan da bude dekan bilo koje škole.

Svaka škola se sastoji od nekoliko departmana. Na primer, škola biznisa ima odeljenje za računovodstvo, odeljenje za menadžment/marketing, odeljenje za ekonomiju/finansije i odeljenje za informacione sisteme računara.

Jasno je da je atribut ROOM\_CODE u entitetu CLASS strani ključ (FK) za drugi entitet koji se naziva ROOM (učionica). Opet, svaka učionica se nalazi u jednoj zgradi.

Svaki departman može nuditi kurseve. Na primer, odeljenje za menadžment/marketing nudi kurseve kao što su Uvod u Menadžment, Osnove Marketinga i Upravljanje Proizvodnjom. Tiny College ima neka odeljenja koja su klasifikovana kao "samo za istraživanje", ta odeljenja ne bi nudila kurseve; zbog toga će entitet KURS biti opcioni za entitet DEPARTMAN.

Departman može ponuditi nekoliko sekcija (časova) istog kursa baze podataka. Svaki od tih časova predaje profesor u određeno vreme i na određenom mestu. Ukratko, postoji odnos 1:M između KURSA (COURSE) i ČASA (CLASS). Međutim, pošto kurs može postojati u katalogu kurseva Tiny College-a čak i kada se ne nudi kao čas u trenutnom rasporedu časova, ČAS je opcioni za KURS.

Svaki departman treba da ima jednog ili više profesora dodeljenih njemu. Samo jedan od tih profesora je predsednik (chair) departmana, i nije obavezno da profesor prihvati poziciju predsednika. Zbog toga je DEPARTMAN opcioni za PROFESORA u odnosu "predsednik". Svaki profesor može predavati do četiri časa; svaki čas je sekcija kursa. Profesor takođe može biti angažovan na istraživačkom ugovoru i uopšte ne predavati.

Student može upisati nekoliko časova, ali svaki čas može pohađati samo jednom tokom bilo kojeg datog perioda upisa. Pošto ČAS može postojati (na početku perioda upisa) čak i kada nijedan student nije upisan, STUDENT je opcioni za ČAS u M:N vezi. Ova M:N veza mora biti podeljena u dve 1:M veze pomoću upotrebe ENROLL entiteta.



## ▼ Poglavlje 8

### 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.

## ▼ 8.1 Pokazni primer

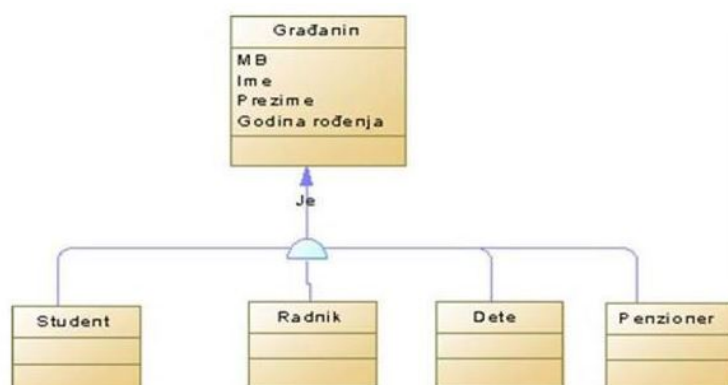
### KREIRANJE VEZE GENERALIZACIJA U POWERDESIGNER-U - (5 MIN.)

*Prikaz generalizacije u Power Designer-u.*

Generalizacija je apstrakcija u kojoj se skup sličnih tipova entiteta tretira kao generički entitet (nad entitet).

“Slični” tipovi entiteta su oni tipovi koji imaju neke zajedničke atribute i veze.

Specijalizacija je inverzni postupak u kome se za neki tip entiteta, definišu njegovi pod entiteti, koji imaju neke njima specifične atribute ili veze



Slika 8.1.1 Primer generalizacije [Izvor: NM IT350 -2020/2021.]

Na slici 9.1 je ilustrovan skup tipova entiteta: Radnik, Student, Penzioner, Dete koji se mogu predstaviti generičkim tipom Građanin.

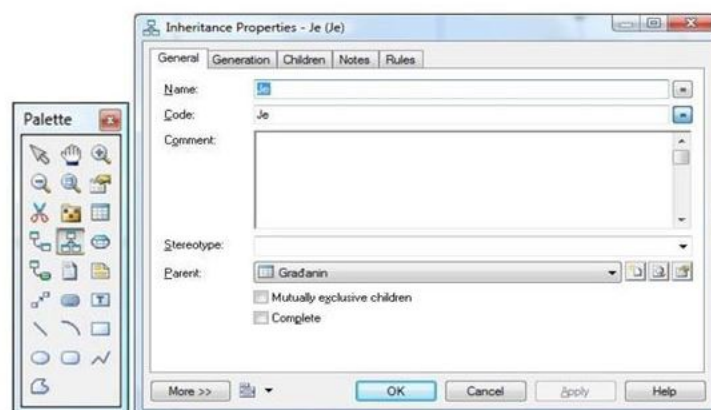
Isto tako se tip entiteta Radnik može specijalizovati u pod tipove Vozač, Inženjer, Lekar i druge. Nad tipa sadrži sva pojavljivanja "sličnih" tipova entiteta, sa onim atributima i preslikavanjima koji su im zajednički, a pod entiteti su podskupovi pojavljivanja nad entiteta sa dodatnim, njima specifičnim preslikavanjima.

## KAKO KREIRAMO VEZE GENERALIZACIJA? - (5 MIN.)

*Kreiranje generalizacije u PowerDesigner je podržano kroz opciju **Inheritance**.*

Kreiranje generalizacije u alatu PowerDesigner je podržano kroz opciju *Inheritance* koju možete odabrati iz komandne palete kao što je prikazano na slici 9.2.

Relacijom treba spojiti nad entitet sa pod entitetima čime ćemo dobiti željenu generalizaciju. Po uspostavljanju generalizacije možemo definisati dodatne osobine relacije.



Slika 8.1.2 Kreiranje nasleđivanja [Izvor: NM IT350 -2020/2021.]

Dodatne opcije koje možemo definisati su:

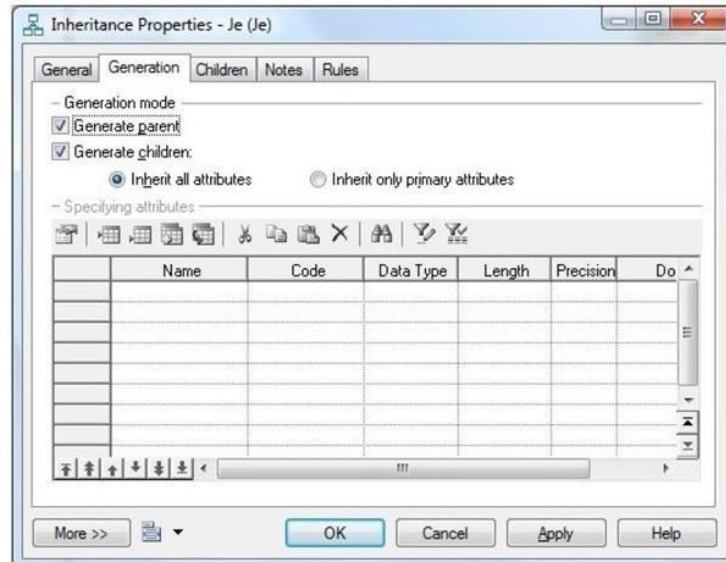
*Mutually exclusive children* : Za jedna od nadentiteta može postojati samo jedan podentitet.

*Complete*: U slučaju da je ova opcija selektovana alatu je data instrukcija da je generalizacija takva da za svaku pojavu nad entiteta mora postojati bar jedna pojava pod entiteta.

## NASTAVAK O KREIRANJU VEZE GENERALIZACIJA - (5 MIN.)

*Prikaz opcija prilikom kreiranja nasleđivanja.*

Veza generalizacije se može koristiti samo na nivou konceptualnog modela ali se prilikom transformacije konceptualnog modela u relacioni model ne može primeniti pravilo nasleđivanja osobina koje kaže da pod entiteti nasleđuju sve atribute i veze svoga nad entiteta.



Slika 8.1.3 Prikaz opcija prilikom kreiranja nasleđivanja u PowerDesigner-u [Izvor: NM IT350 -2020/2021.]

Na slici 9.3 su prikazane opcije za transformaciju veze generalizacije u logički a zatim i fizički model:

*Generate parent:* Pri kreiranju fizičkog modela podataka kreiraće se tabela koja predstavlja nadentitet.

*Generate children:* Pri kreiranju fizičkog modela podataka biće kreirana tabela za svaki podentitet.

*Inherit all attributes:* Svaka tabela pod entiteta će uz svoje atribute imati sve atribute koje ima i nadentitet

*Inherit only primary attributes:* Tabele pod entiteta će naslediti samo identifikator nadentiteta

## ▼ 8.2 Pokazna vežba - samostalni rad

### 1. ZADATAK ZA SAMOSTALNI RAD

*Opis zadatka 1- Student bira jedan od zadataka 1. i 2. (vreme izrade 25 min.) i zadatke 3. i 4. (vreme izrade 50 min.)*

Projektovati konceptualnu šemu baze podataka na osnovu sledećih podataka i zahteva (koristeći Power Designer). Analizirati zahteve i podatke i na osnovu njih odrediti entitete, attribute, veze:

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

## 2. ZADATAK ZA SAMOSTALNI RAD

*Opis zadatka 2- Student bira jedan od zadataka 1. i 2. (vreme izrade 25 min.) i zadatke 3. i 4. (vreme izrade 50 min.)*

Projektovati konceptualnu šemu baze podataka jednog univerziteta na osnovu sledećih podataka i zahteva (koristeći Power Designer). Analizirati zahteve i podatke i na osnovu njih odrediti entitete, attribute, veze:

1. U bazi podataka pored podataka o studentima treba evidentirati podatke i o profesorima i ne nastavnom osoblju na univerzitetu.
2. Prilikom upisa u svaku godinu studija studentima se dodeljuju predmeti koje on sluša i polaže u toj godini ali studenti pri tom biraju neke od ponuđenih izbornih predmeta koji imaju neke svoje specifičnosti kao što su stil učenja, dodatne obaveze, način polaganja itd.
3. 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.
4. 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 25 minuta.**

### 3. I 4. ZADATAK ZA SAMOSTALNI RAD

*Izradom navedenih primera u korišćenju relacija treba proveriti stečeno znanje - Student bira jedan od zadataka 1. i 2. (vreme izrade 25 min.) i zadatke 3. i 4. (vreme izrade 50 min.)*

**Zadatak 3:** Dizajnirajte bazu podataka za banku uključujući i informacije o korisnicima i njihovim računima. Informacije o korisnicima uključuju: ime, adresu telefon i socijalni broj. Račun ima attribute broj, tip računa, i stanje. Treba imati informaciju da li je korisnik vlasnik računa ili ne. Generalno, račun može imati samo jednog vlasnika i više ovlašćenih lica. Nacrtajte E/R dijagram za ovu bazu sa naznačenim kardinalnostima relacija.

1. Izmenite svoj originalni dijagram tako da kupac može imati više adresa koje sadrže ulicu i grad i skup telefona. Podsetimo se da u E/R modelu atributi ne mogu imati više vrednosti.
2. Zatim modifikujte svoj E/R dijagram tako korisnik može imati više adresa a na svakoj adresi po više telefona. **(25 min.)**

**Zadatak 4:** Nacrtajte dijagram za bazu podataka u kojoj se pamte informacije o timovima, igračima i njihovim fanovima uključujući sledeće

1. Za svaki tim treba sačuvati njegovo ime, njegove igrače, kapitena tima (jedan od igrača) i boju njihovih uniformi.
2. Za svakog igrača treba upamtiti njegovo ime.
3. Za svakog fana treba znati ime, tim za koji navija, omiljenog igrača i omiljenu boju uniformi.

U dijagramu koristite n-arne relacije.

1. Pretpostavite da prethodnoj šemi želite da dodate vezu Upravljan od (Led-by) između dva igrača u timu. Namera je da se pojavljivanja te relacije sastoje od torki (player1, player2, team) tako da player1 igra u timu u trenutku kada je neki drugi igrač (player2) kapiten tima.
2. Zamenite n-arnu relaciju novim tipom entiteta i skupom binarnih relacija.
3. Dijagram modifikujte tako da se za svakog igrača mogu istorijski pamtit svi timovi u kojima je on igrao, uključujući i vremenski period od kada do kada je igrao u svakom timu. **(25 min.)**

**Potrebno vreme za rešavanje zadataka 50 minuta.**

### VIDEO

*Modeliranje baze podataka "Prijem pacijenata"*

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

## ▼ Poglavlje 9

### Domaći zadatak

#### DOMAĆI ZADATAK 4 - VREME IZRADE 90 MIN.

##### *Uputstvo za rešavanje domaćeg zadatka 4*

Na modelu baze podataka koji si kreirao u prethodnom domaćem zadatku, izvrši odgovarajuće korekcije tako da, ako za tim postoji potreba, dodaš:

1. rekurzivne relacije
2. ternarne i kvaternarne relacije
3. asocijativne tipove entiteta
4. specijalizacije i generalizacije

pod pretpostavkom da opise baza podataka treba proširiti na način kako je to dato u daljem tekstu. Kako svako od proširenja sadrži po dve mogućnosti, studenti čiji se indeks završava brojevima od 0 do 4 treba da izaberu prvu mogućnost a oni sa indeksima koji se završavaju brojevima od 5 do 9 drugu mogućnost.

##### **Baza podataka za podršku proizvodnje automobila:**

- kupci automobila mogu biti veliko prodavci i individualni kupci za koje treba evidentirati za njih specifične attribute
- za svaki proizvedeni automobil postoji sastavnica koja sadrži listu svih delova koje treba ugraditi u svaki od modela automobila sa odgovarajućim količinama. Postoje dve mogućnosti:
  - Svaki automobil se može sastojati od više delova a svaki deo se može ugraditi u više automobila.
  - Svaki automobil se može sastojati od više delova a svaki deo se može ugraditi u samo jedan automobil.

##### **Baza podataka hotela:**

- za svaku hotelsku sobu je potrebno znati datum od kada do kada je soba rezervisana kao i da li će i koje dodatne usluge hotela (spa centar, teretana, medicinski tretmani itd.) koristiti u periodu rezervacije.
- za svaku vrstu hrane iz menija postoji recept koji sadrži listu svih namirnica koje treba koristiti prilikom pripreme sa odgovarajućom količinom. Postoje dve mogućnosti:
  - Svaka vrsta hrane se može sastojati od više namirnica a svaki namirnica se može koristiti u više vrsta hrane.
  - Svaka vrsta hrane se može sastojati od više namirnica a svaki namirnica se može koristiti u samo jednoj vrsti hrane.

## DOMAĆI ZADATAK 4 - NASTAVAK

### *Uputstvo za rešavanje domaćeg zadatka 4 - nastavak*

#### **Baza podataka doma zdravlja:**

- za svaku dijagnozu koja se može postaviti pacijentu je potrebno znati listu preporučenih lekova i prilikom prepisivanja lekova lekar može da pacijentu prepíše lek samo iz preporučene liste. Liste preporučenih lekova se menjaju svake kalendarske godine.
- za svakog lekara i medicinsku sestru je potrebno znati kom odeljenju pripada (hirurgija, kardiologija, pedijatrija, opšta medicina itd.) a takođe obezbediti informaciju ko je kome šef u svakom odeljenju. Postoje dve mogućnosti:
  - Lekar ili medicinska sestra pripadaju isključivo jednom odeljenju
  - Lekar ili medicinska sestra mogu pripadati većem broju odeljenja

#### **Baza podataka univerziteta:**

- Angažovanje nastavnika je potrebno znati po školskim godinama i semestrima pri čemu na jednom predmetu može biti angažovano više nastavnika a jedan nastavnik može biti angažovan na više predmeta. Za svakog nastavnika je takođe potrebno znati ulogu koju ima na predmetu za datu školsku godinu i semestar (profesor, asistent, demonstrator).
- Nastavnike treba podeliti po katedrama i za svaku katedru definisati ko je šef katedre. Postoje dve mogućnosti:
  - Nastavnik može pripadati isključivo jednoj katedri
  - Nastavnik može pripadati većem broju katedri

#### **Baza podataka za podršku rada neke picerije:**

- kupci pica mogu biti restorani i individualni kupci za koje treba evidentirati njima specifične atribute
- za svaku picu postoji recept koji sadrži listu svih namirnica koje treba dodati prilikom pripreme različitih vrsti pica sa odgovarajućim količinama. Postoje dve mogućnosti:
  - Svaka pica se može sastojati od više namirnica a svaki namirnica se može koristiti u više vrsta pica.
  - Svaka pica se može sastojati od više namirnica a svaki namirnica se može koristiti u jednoj vrsti pica.

## DOMAĆI ZADATAK 4 - UPUTSTVO ZA SLANJE DOMAĆEG ZADATKA

### *Prilikom slanja domaćih zadatka, neophodno je da ispunite sledeće:*

Prilikom slanja domaćih zadatka, neophodno je da ispunite sledeće:

Subject mail-a mora biti IT250-DZbr (u slučaju kada šalžete domaći za ovu nedelju to je IT250-DZ04)

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-DZ04-1234-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 poslati 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?

E/R dijagrami su najzastupljenija tehnika dizajniranja baza podataka koja se najčešće koristi kada se projektuje potpuno nova baza podataka na osnovu specificiranih zahteva korisnika.

U ovoj lekciji se detaljno objašnjavaju napredni koncepti E/R dijagrama jer je primena i korišćenje ovakvog projektovanja baza podataka jedan od najvažnijih ciljeva ovog kursa.

Tako su objašnjeni koncepti: n-arnih relacija, asocijativnih entiteta, specijalizacije i generalizacije, slabih i jakih entiteta 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.1.2.0j0i22i30k1l3j0i22i10i30k1.1916.4268.0.9223.6.6.0.0.0.105.573.4j2.6.0...0...1c.1.64.psyab..0.6.571...0i67k1.0.1gPVVogyWEA>