

# Entity Relationship model (ER) Model Entita i Odnosa (MEO)

prema

<http://db4u.wikidot.com/start>

# ER modeliranje

- ER: Peter Chen, 1976: cilj da objedini mrežni i relacioni pogled na bazu podataka
- Jednostavno: ER model je konceptualni model podataka koji realni svet “vidi” kroz entitete i njihove odnose
- Osnovno svojstvo modela: dijagramska tehnika kojom se objekti podataka predstavljaju vizuelno
- Nije model u pravom smislu
- Primena u semantičkoj metodi logičkog projektovanja
- ER shema se jednostavno transformiše u relacionu
- Tanki sloj nad relacionim modelom
- Proširenja - PME0 (EER)

# Koncepti ER modeliranja: tip entiteta - stavka od značaja

- Cilj ER modeliranja je da stvori verni odraz realnosti u bazi podataka
- ER modeliranje je konceptualno modeliranje
- ER model ne daje opis baze podataka
- Proizvod ER modeliranja je međukorak iz kojeg se lako definiše baza podataka
- **Primer**
- Potrebno je kreirati bazu podataka za sledeću situaciju:
- Svaki sektor u okviru kompanije pripada tačno jednom odeljenju. Svako odeljenje ima veše sektora. Nema gornje granice broja sektora u okviru jednog odeljenja. Na primer, *New Business Development* — sektor kojim upravlja Mackenzie — i *Higher Education* sektor su u odeljenju *Marketing*.
- Mnoge stvari se podrazumevaju u vezi sa ovim opisom. Na primer, svako odeljenje ima ime, ime je jedinstveno u okviru kompanije, i sl.
- Prvi korak u ER modeliranju je uočiti stavke od značaja u opisanoj situaciji.
- **Problem 1.1**
- **Koje su stavke od značaja u opisanoj situaciji?**

# Koncepti ER modeliranja: tip entiteta / entitet

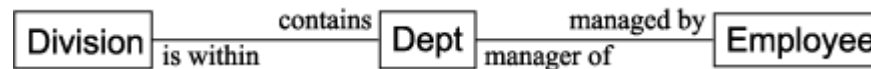
- Definicija: **tip entiteta** je kolekcija entiteta sa zajedničkom definicijom. *Entitet* je osoba, mesto, pojam ili stvar o kojoj je potrebno imati podatke
- Tako, *Sektor* je ime *tipa entiteta*. Jedna *instanca* ovog tipa entiteta (=entitet) je *New Business Development sektor*. *Marketing odeljenje* je instanca tipa entiteta *Odeljenje*. *Mackenzie* je instanca tipa entiteta *Zaposleni*.
- Tipovi entiteta omogućuju da se, umesto u formi iskaza o odnosima među entitetima, informacija može formulisati u formi iskaza o odnosima između **tipova entiteta** (npr. “svaki sektor je u tačno jednom odeljenju”)
- **Problem 1.2**
- **Šta su tipovi entiteta a šta entiteti ? Navesti primere entiteta za tipove entiteta.**
  - **Ford**
  - **Clothes**
  - **Employee**

# Koncepti ER modeliranja: veze (odnosi)

- U prethodnom opisu postoje određene veze između *Sektora* i *Odeljenja* i druga vrsta veze između *Sektora* i *Zaposlenog*. Prva veza je *veza sadržavanja* – svako odeljenje ima jedan ili više sektora, a jedan sektor pripada tačno jednom odeljenju.
- Drugi tip veze govori da je jedan zaposleni u izvesnom odnosu prema nekom sektoru, i to da taj zaposleni *upravlja* tim sektorom, a da se sektorom *upravlja od* (sektorom je *upravljano od*) strane tog zaposlenog.
- Drugi važan korak u ER modeliranju je određivanje odnosa između tipova entiteta
- *Odnos (Relationship)* je veza između tipova entiteta.
- **Problem 1.3**
- **Kako biste nazvali ove dve vrste veza?**

# Koncepti ER modeliranja: kardinalnost

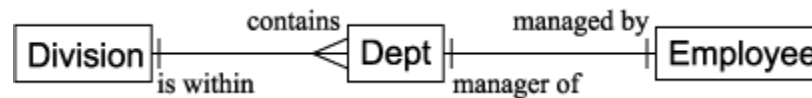
- Karakteristika odnosa je da uključuje nekoliko tipova entiteta. Zato *ime* ili *datum rođenja* **NISU** odnosi.
- ER modeli se obično prikazuju grafički, dijagramima. Jedna notacija predstavlja tipove entiteta pravougaoncima a odnose linijama među pravougaoncima. Za naš primer,



- Objašnjenje: Sektor (Department) / Odeljenje (Division), Sektor / Zaposleni
- Zaposleni nisu u direktnoj vezi sa odeljenjima
- Znamo i više: Znamo da u svakom (*jednom*) odeljenju može da bude *više* sektora. U jeziku ER modeliranja, to je **1:M** (“jedan prema više”) odnos.
- **Problem 1.4**
- **Kakav je odnos između sektora i direktora (jedan / više)?**
- **U svakom sektoru može da bude najviše \_\_\_\_\_ direktora.**
- **Za svakog direktora, može da bude najviše \_\_\_\_\_ sektora.**

# Koncepti ER modeliranja: kardinalnost

- Prethodna informacija može da se predstavi ER dijagramom:
- 



- **M** kraj odnosa se u ovoj notaciji predstavlja “viljuškom” pored odgovarajućeg tipa entiteta u odnosu, dok se **1** kraj predstavlja kratkom crticom preko linije odnosa. Sada ER dijagram predstavlja više informacija:
  - Svako odeljenje može da sadrži više sektora, a svaki sektor pripada najviše jednom odeljenju
  - Svaki sektor može da ima najviše jednog direktora. Svaki zaposleni može da upravlja (da bude direktor) najviše jednim sektorom
- Druga pitanja:
  - Koliki je najmanji broj sektora u odeljenju?
  - Da li sektor mora da bude pridružen odeljenju?
  - Da li sektor mora da ima direktora?
  - I ova pitanja moraju da dobiju odgovor u ER modelu.

# ER modeliranje: glavni koraci

- Odrediti tipove entiteta
- Odrediti koji tipovi entiteta grade odnos
- Profiniti definiciju odnosa
  
- Postoji više metoda / tehnika za grafičku reprezentaciju ER modela. Za predstavljanje odnosa neke koriste liniju sa nazivima na njoj, druge koriste romb.
- Najčešći scenario:
  - Poći od tekstualnog opisa problema
  - Analizirati ga i uvesti neophodne pretpostavke
  - Kreirati ER dijagram koji odslikava situaciju i eksplicira odnose među tipovima entiteta
  - Korak od ER dijagrama ka implementaciji baze podataka biće pravolinijski
  - Kreiranje baze podataka i jeste naš glavni cilj



# Odnosi (veze, engl. relationships)

- U ranijem primeru imali smo da su *odeljenja* direktno pridružena *sektorima* a *sektori* direktno pridruženi *zaposlenima*. Direktno pridruživanje *odeljenja* i *zaposlenih* nije dato. Ali ER dijagram utvrđuje da i takav odnos postoji i da se može izvesti iz direktnih odnosa. ER dijagram treba da minimizira broj odnosa iz kojih se ostali mogu izvesti.
- Sledeći korak je utvrđivanje *kardinalnosti* odnosa.
- U ER modelu, *kardinalnost* odnosa je maksimalni broj entiteta jednog tipa koji se mogu pridružiti jednom entitetu drugog tipa (u EER kardinalnost će se definisati na drugačiji način !!!)
- Postoje tri osnovne kardinalnosti binarnog odnosa (između tipova entiteta X i Y):
- 1:1
  - jedan prema jedan — Jedan entitet tipa X može biti pridružen najviše jednom entitetu tipa Y. Jedan entitet tipa Y može biti pridružen najviše jednom entitetu tipa X. Primer: *automobil* i *volan*.
- 1:M
  - jedan prema više — Jedan entitet tipa X može biti pridružen većem broju entiteta tipa Y. Jedan entitet tipa Y može biti pridružen najviše jednom entitetu tipa X. Primer: *zgrada* i *prostorije*.
- M:M
  - više prema više — Jedan entitet tipa X može biti pridružen većem broju entiteta tipa Y. Jedan entitet tipa Y može biti pridružen većem broju entiteta tipa X. Primer: *automobil* i *oprema*

# Odnosi

- **Problem 2.1**
- Za svaki par tipova entiteta prvo odgovoriti na 2 pitanja:
  - Za svaki entitet tipa Y može da bude najviše \_\_\_\_\_ entiteta tipa X.
  - 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 odnosa između tipova entiteta X i Y **M:1** i čita se: **za svaki X može biti samo jedan Y i za svaki Y može biti više X.**)

- Zatim, odrediti kardinalnost sledećih odnosa (**M:1, 1:M, M:M, 1:1**) :
  - Pacijent pod **praćenjem** lekara opšte prakse
  - Lekar **vrši** operaciju
  - Lekar **ima specijalnost za** bolest

# Egzistencija (u EER: donja granica kardinalnosti)

- Prethodno smo razmatrali maksimalni broj entiteta jednog tipa koji se mogu pridružiti jednom entitetu drugog tipa. Sada se bavimo minimalnim brojem entiteta u odnosu.
- Egzistencija odnosa definiše ono što znamo o postojanju entiteta na drugoj strani odnosa, i zadaje se kao **opcionalna**, **obavezna** ili **nepoznata**.
- **Primer:** Razmotrimo odnos *upravlja* između sektora i zaposlenog. Znamo da je kardinalnost odnosa **1:1**. To nam govori da je jedan zaposleni - direktor najviše jednog sektora i da jedan sektor ima najviše jednog zaposlenog kao svog direktora.
- Egzistencija ovog odnosa govori o **najmanjem** broju sektora kojim može da upravlja jedan zaposleni, i najmanjem broju zaposlenih koji mogu da upravljaju jednim sektorom. Mogućnosti su:
  - Za sektor:
    - **Opcionalna:** Sektor ne mora da ima direktora *||*
    - **Obavezna:** Sektor mora da ima bar jednog direktora *||*
    - **Nepoznata:** ne zna se da li sektor mora da ima direktora
  - Slično, za zaposlenog
    - **Opcionalna:** Radnik ne mora da upravlja nijednim sektorom
    - **Obavezna:** Radnik mora da upravlja bar jednim sektorom
    - **Nepoznata:** ne zna se da li radnik mora da upravlja sektorom

# Egzistencija

- *Problem 2.2.*
- *Šta biste izabrali iz svakog od prethodna dva skupa?*

# Egzistencija

- Objašnjenje problema 2.2
- Za proizvoljni sektor, mora da postoji zaposleni na drugoj strani odnosa **upravlja**, pa je odnos obavezan u tom smeru. Predstavlja se crticom upravnom na liniju odnosa, bližom sredini te linije.
- Iz sličnih razloga, odnos je opcioni u drugom smeru; predstavlja se kružićem.
- Možemo da pretpostavimo da je odnos **sadrži** obavezan u oba smera, pa ER dijagram može da se modifikuje tako da postane:



- Dijagram se čita sleva na desno i zdesna na levo. Unutrašnje oznake predstavljaju minimalni broj u odnosu, spoljašnje maksimalni broj u odnosu. Crtica označava obaveznost (najmanji broj 1), kružić opcionost (najmanji broj je 0); ako nema oznaka, minimalni broj je nepoznat
- Oznake bliže pravougaonicima predstavljaju maksimalni broj u odnosu. Viljuška označava kardinalnost "više", crtica "1".

# Nadtip / podtip

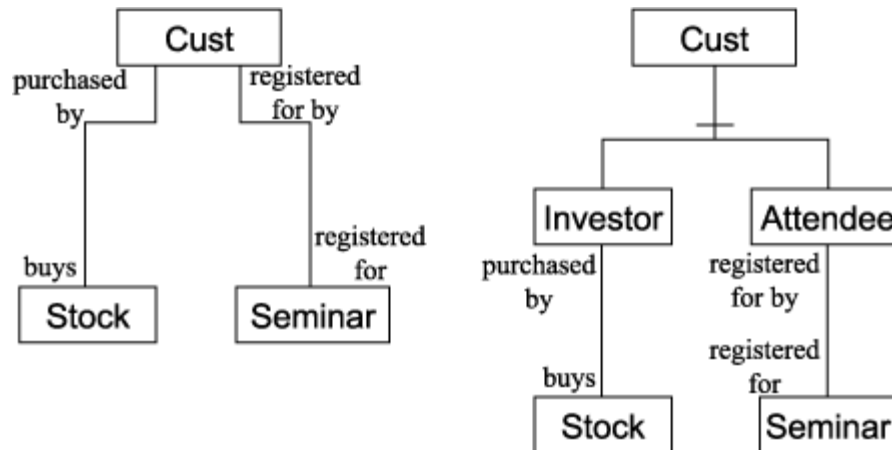
- Moguće je vršiti generalizaciju (uopštavanje) nad tipovima entiteta, ili nad podskupovima tipova entiteta.
- Primer:
- Računovodstvena baza podataka sa podacima o kompanijama kojima dugujemo i koje nam duguju, sa iznosima dugovanja.
- Kreiramo *podtipove* entiteta.
- Podtip entiteta je kolekcija entiteta istog tipa na koju se može primeniti uža definicija i dodatni atributi i/ili odnosi.
- ***Problem 2.4***
- ***Šta je tip entiteta a šta podtipovi u prethodnom primeru?***

# Podtipovi

- U mnogim situacijama moguće je kreirati podtipove, mada ne bi trebalo.
- Podtipove treba kreirati
  - kada jedan podtip gradi odnose koje drugi podtipovi ne grade *iii*
  - kada podtip treba da ima dodatne podatke koji se pamte
- U suprotnom *ne treba* kreirati podtipove.
  
- *Problem 2.5*
- *Koji odnos postoji između sledećih tipova entiteta? Nabrojati neke podatke zajedničke za sve elemente u svakom pitanju. Nabrojati neke podatke za svaki podtip koji nisu zajednički za ostale podtipove. Nacrtati ER dijagram za svaki.*
  
- *knjiga, publikacija, časopis*
- *laptop, kompjuter, desktop kompjuter*

# Podtipovi

- Neke situacije nisu tako jasne, na primer



- Odnos / podtip?
- Radi se o vezi koja samo definiše članove potencijalnog podtipa, bez novih svojstava / odnosa
- Bolje odnos nego podtip



# Atributi

- Prethodno smo pominjali “podatke” i “atribute” koji opisuju ili karakterišu neke entitete.
- *Atributi* su karakteristike tipa entiteta za koje smo zainteresovani.
- *Atribut* je deskriptor čije su vrednosti pridružene individualnim entitetima specifičnog tipa.
- **Vrednost atributa jednog entiteta je *jedinstvena* u svakom trenutku, a može se menjati u vremenu.**
- Na primer, *zarada* je atribut zaposlenog.
- Na primer, tip entiteta *sektor* ima atribut *ime*, kao i tipovi entiteta *odeljenje* i *zaposleni*. Mogući atributi tipa entiteta *zaposleni* (koji nisu pomenuti) jesu *datum zaposlenja*, *kućna adresa*, *telefon na poslu*, *adresa na poslu*.
- Svaki tip entiteta ima ***identifikator***. On na jedinstveni način identifikuje entitet. Vrednost identifikatora se ne menja u vremenu.
- JMBG je mogući identifikator osobe. Šta je mogući problem? Zašto se ne koriste imena kao identifikatori?

# Atributi

- *Problem 2.8*
- *Navesti mogući identifikator za svaki od sledećih tipova entiteta:*
- *Knjiga*
- *Automobil*
- *BU student*
- *Boja*
  
- *Problem 2.9*
- *Pretpostavimo da imamo bazu podataka sa tabelom automobil, i da ta tabela ima identifikator id i polje model (npr. "Ford Mustang GT", "Ford Mustang SHO", ili "Accord Integra GS-R"). Ako poželimo da dodamo informaciju o tipu modela (npr. "sportski", "sedan"), da li je bolje da dodamo polje (atribut) tip\_modela u tabelu automobil, ili da napravimo drugu tabelu sa ovim informacijama?*

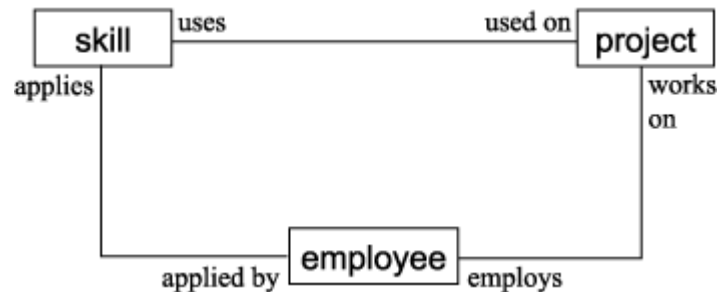
# Napredne teme u ER modeliranju

- **Stepen odnosa**
  - Odnosi mogu da se klasifikuju prema broju tipova entiteta koji u njima učestvuju. Taj broj se naziva *stepenom odnosa*. Do sada smo se bavili najčešćom vrstom odnosa – binarnim odnosom tj. odnosom između dva tipa entiteta. Uobičajeni stepeni odnosa su:
    - binarni
      - Odnos između dva tipa entiteta
    - ternarni
      - Odnos između tri tipa entiteta
    - rekurzivni
      - Odnos koji uključuje samo jedan tip entiteta

# Ternarni odnosi

- Na primer, koje veštine (skill) zaposleni (employee) koriste na kojim projektima (project). Mogu da se koriste tri binarna odnosa (veštine – projekti, projekti – zaposleni, zaposleni – veštine):

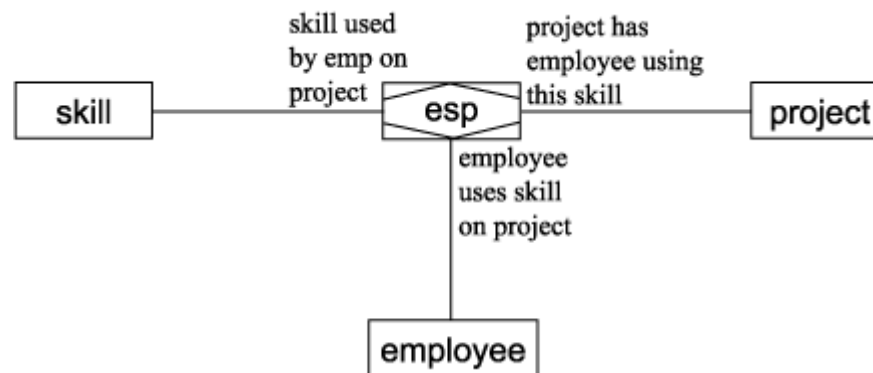
- 



- Odnos *primenjuje* (apply) ukazuje na to koji zaposleni koristi koje veštine. Odnos *korišćena na* (used on) ukazuje koja se veština koristi na kom projektu. Odnos *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:

# Ternarni odnosi – nast.

- *radi na*
  - Petar i Ana rade na projektima A i B.
- *primenjuje*
  - Petar primenjuje veštine *interface design* i *database design* a Ana samo veštinu *database design*.
- *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.
- Potreban je ternarni odnos, *esp* (*employee, skill, project*), između zaposlenih, veština i projekata.

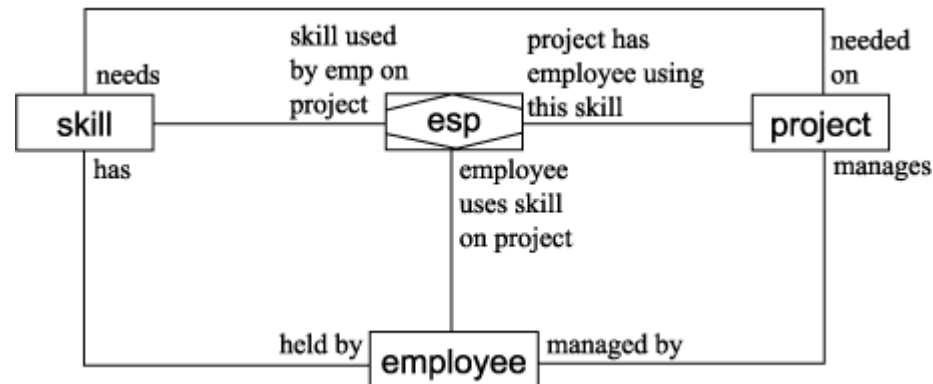


# Ternarni odnosi – nast.

- Odnos *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*.
- Ternarni odnos *esp* sadrži i informacije predstavljene binarnim odnosima:
  - Petar radi na projektu *A* (iz 1.).
  - Petar radi na projektu *B* (iz 3. i 4.).
  - Ana radi na projektu *A* (iz 2.).
  - Ana radi na projektu *B* (iz 5.).
  - Petar koristi veštinu interface design (iz 1. i 3.).
  - Petar koristi veštinu database design (iz 4.).
  - Ana koristi veštinu database design (iz 2. i 5.).
  - Obe veštine koriste se na oba projekta (iz svih).

# Ternarni odnosi – nast.

- 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,



# Ternarni odnosi – nast.

- Odnos *esp* ostaje isti kao i ranije. Binarni odnosi su drugačiji.
- odnos *poseduje / posedovan od (has / held by)*
  - Zaposleni *poseduje* neke veštine. Ovaj odnos se razlikuje od *esp* odnosa zato što mogu postojati veštine koje zaposleni poseduje ali koje nije koristio u okviru nekog projekta.
- odnos *zahteva / biti potreban (needs / needed on)*
  - Neki projekat *zahteva* specifične veštine. Ovaj odnos se razlikuje od *esp* odnosa zato što mogu da postoje veštine za koje nekom projektu još nisu pridruženi zaposleni koji ih poseduju.
- odnos *upravlja / upravljani od (manages / managed by)*
  - Zaposleni *upravlja* projektom. To je potpuno drugačija dimenzija od veštine i zato se ne može obuhvatiti *esp* odnosom.



# Problem 3.1

- Imamo tri entiteta: automobil, kupac, prodavac. Razmotrimo sledeću situaciju.
- #1
- Postoji ternarni odnos *prodaja* koji povezuje sva tri entiteta.
- Prodaja: “Kupac K kupio je automobil A od prodavca P.”

```
=====
                prodaja
=====
A           K       P
=====
Cobra         Sam      Don
Mustang       Sam      Don
Mustang       Sam      Sharon
Cobra         Jenn     Sharon
Mustang       Jenn     Sharon
```

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

# Problem 3.1

- #2
- 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

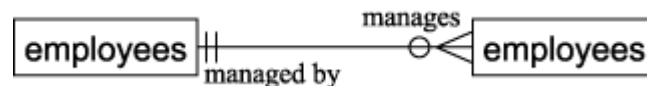
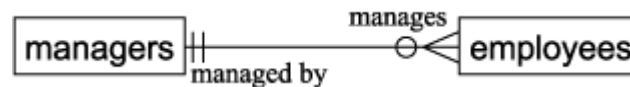
- kupuje
    - Kupac K kupuje automobil A.
  - kupuje od
    - Kupac K kupuje od prodavca P.
  - prodaje
    - Prodavac P prodaje automobil A.
- Pitanje: Ko je prodao automobil Cobra kupcu Sam? (Don ili Sharon)

# Kardinalnost ternarnog odnosa

- Metod određivanja kardinalnosti odnosa višeg reda razlikuje se od određivanja kardinalnosti binarnog odnosa:
  - Prekriti sve linije koje idu od ternarnog odnosa (npr. *esp*) prema nekom entitetu, osim jedne. (Na primer, prekriti linije iz *esp* prema *employee* i *project*.)
  - Za preostali entitet, zapitati se: da li može da postoji najviše jedan primerak tog entiteta, za svaku pojedinačnu kombinaciju preostalih entiteta, ili ih može biti više? (U našem primeru, zapitati se: da li može da postoji najviše jedna veština za nekog zaposlenog i neki projekat, tj. koju neki zaposleni primenjuje na nekom projektu, ili zaposleni može da primenjuje veći broj veština na jednom projektu?)
  - Ako je odgovor “više”, staviti viljušku na liniju koja ulazi u taj entitet; ako je odgovor “1”, staviti crticu preko te linije.
  - Ponoviti proces dok se ne ispita svaki entitet u odnosu.
- *Problem 3.2*
- *Za esp odnos, koje su kardinalnosti pojedinačnih tipova entiteta?*

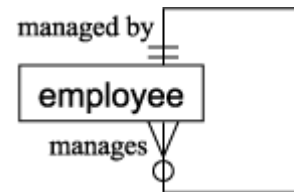
# Rekurzivni odnos

- Verovatno najkomplicovaniji je *rekurzivni* odnos.
- To je odnos koji entitet gradi sa samim sobom. Može se zamisliti i kao binarni odnos nad dva ista tipa entiteta.
- Primer: zaposleni koji je direktor zaposlenima.
- 
- Jedan direktor upravlja sa više zaposlenih, a jedan zaposleni ima tačno jednog neposrednog direktora. Ali direktor je samo još jedno ime za zaposlenog. Zato se *direktori* (managers) može zameniti sa *zaposleni* (employees) u ovom dijagramu:



# Rekurzivni odnos

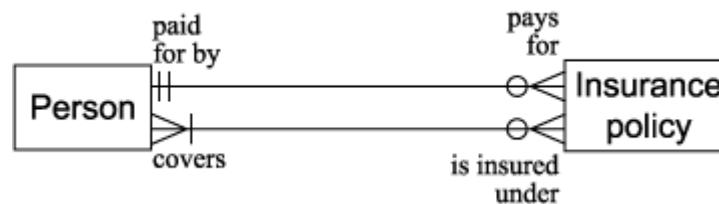
- Na prethodnom dijagramu, umesto da se tip *zaposleni* pojavi dva puta, linija odnosa može da se “vrati” do tipa entiteta zaposleni, pa dijagram ima sledeći izgled:



- Čitanje dijagrama u smeru kazaljke na satu, polazeći od tipa *zaposleni*: jedan zaposleni možda ne upravlja nijednim zaposlenim ali može da upravlja i većim brojem zaposlenih.
- Čitanje u smeru suprotnom smeru kazaljke na satu, polazeći od tipa *zaposleni*: svakim zaposlenim upravlja tačno jedan zaposleni.
- Nema *svako* u kompaniji direktora. Predsednik kompanije nema direktnog direktora. Ovo se postiže podacima u tabeli tako da je predsednikov direktor – sam predsednik.

# Višestruki odnosi

- Dva entiteta mogu da grade više od jednog tipa odnosa. To se lako predstavlja i bazom podataka i ER dijagramom.
- Posmatrajmo tipove entiteta *person* (osoba) i *insurance policy* (polisa osiguranja) i odnose među njima *pays for* (uplaćuje) i *is insured under* (osigurana je po).



- Posmatrajmo ove odnose jedan po jedan.
- Jedna osoba uplaćuje nula ili više polisa osiguranja. Jednu polisu osiguranja uplaćuje tačno jedna osoba.
- Jedna osoba je osigurana po nula ili više polisa osiguranja. Jedna polisa osiguranja pokriva jednu ili više osoba.
- Ovo su dva različita odnosa. Oni označavaju dve različite stvari — zato su predstavljeni kao dva različita odnosa na ER dijagramu.

# Drugi tipovi entiteta: asocijativni entitet

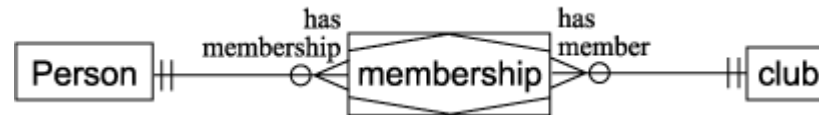
- Atributi su ranije pridruživani samo tipovima entiteta, ali je moguće da i odnosi imaju attribute. Specijalan slučaj odnosa je “više prema više” odnos koji zahteva da neki novokreirani entitet pamti informaciju o tom odnosu. Ta nova, specijalna vrsta entiteta naziva se *asocijativnim entitetom*.
- Na primer, osoba može da bude član u većem broju klubova i klub može da ima više članova, što se predstavlja sledećim ER dijagramom:



# Drugi tipovi entiteta: asocijativni entitet

- Prethodni ER dijagram možemo da preformulišemo tako da više-prema-više odnos zamenimo sa dva jedan-prema-više odnosa sa asocijativnim entitetom u sredini:

- 



- Ova reprezentacija je poželjnija jer se direktno preslikava na bazu podataka koja će biti kreirana kao rezultat ovog odnosa. Odnos više-prema-više mora biti predstavljen posebnom tabelom, baš kao što se i posebni tipovi entiteta predstavljaju posebnim tabelama. U tom slučaju imamo asocijativni entitet koji se koristi da predstavi odnos između dva tipa entiteta.

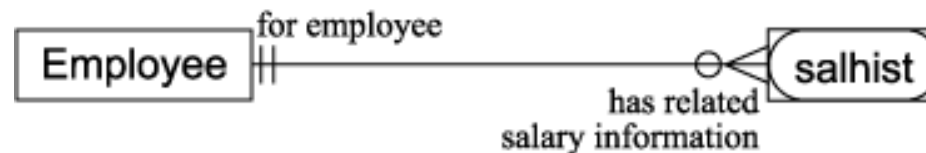


# Drugi tipovi entiteta: asocijativni entitet

- Za asocijativni entitet važi i više: on može da obezbedi bazi podataka podatke o samom odnosu
- Za očekivanje je da se odnosu osoba i klubova pridruži podatak o datumu ućlanjenja. Ovaj atribut se ne može pridružiti tipu entiteta *osoba* jer se ne bi znalo na ćlanstvo u kom klubu se odnosi. Ne bi mogao da se pridruži ni tipu entiteta *klub* jer se ne bi znalo na ućlanjenje kog njegovog ćlana se datum ućlanjenja odnosi. Rešenje je da se “datum ućlanjenja” dodeli kao atribut asocijativnom entitetu *ćlanstvo*.
- Postupak kojim smo došli do asocijativnog entiteta je uobićajeni postupak razvoja ER dijagrama: prvo jednostavniji dijagrami (sa više-prema-više odnosima) a zatim profinjeniji, sa asocijativnim entitetima koji ih zamenjuju.

# Drugi tipovi entiteta: Opisni (atributivni, zavisni) entiteti

- Opisni entiteti su entiteti sa jednom posebnošću: postoje samo zato što postoje i neki drugi entiteti koje ovi prvi opisuju.
- Egzistencija opisnog entiteta je potpuno zavisna od drugog entiteta.
- Na primer, za dva tipa entiteta, *employee* (zaposleni) i *salary-history* (istorija zarade), drugi tip entiteta je opisni entitet zato što zapis istorije zarada zaposlenog ne može da postoji ako ne postoji zapis samog zaposlenog. Istorija zarade Petra Petrovića nema mnogo smisla ako u bazi podataka ne postoji zaposleni Petar Petrović.
- Jedna reprezentacija opisnog entiteta je zaobljeni pravougaonik unutar drugog pravougaonika:

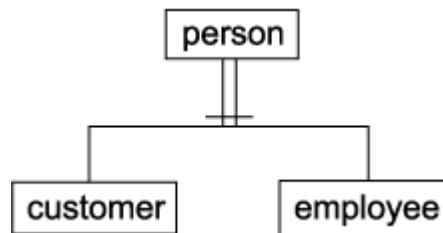


# Drugi tipovi entiteta: Podtipovi

- Dva načina klasifikacije podtipova nekog tipa entiteta, prema tome kako se odnose među sobom i prema nadtipu:
  - opcioni / obavezni podtipovi
  - disjunktни / preklapajući podtipovi
- Podtipove treba klasifikovati po obe dimenzije — moguće su sve četiri kombinacije
- Pretpostavimo tip entiteta *person* (osoba), i podtipove entiteta tog tipa *customer* (klijent) i *employee* (zaposleni). Dizajner baze podataka ima dve mogućnosti pri kreiranju osobe:
  - Obavezni podtip: osoba mora da se klasifikuje kao jedan od podtipova
  - Opcioni podtip: osoba može da se kreira bez klasifikovanja u bilo koji od podtipova
- Izbor zavisi od poslovne situacije.

# Podtipovi: **opcionni** / **obavezni**

- Obavezni podtip se predstavlja duplom linijom od nadtipa (*person* u našem primeru) do linije koja povezuje podtipove. Opcionni podtip se predstavlja jednom linijom od nadtipa do linije koja povezuje podtipove. U oba slučaja dodaje se crtica preko linije koja ide do nadtipa.



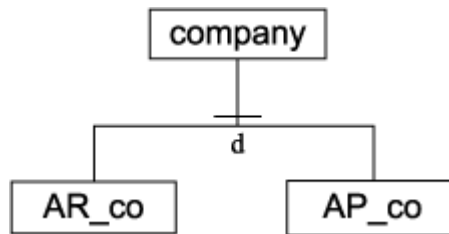
- Ova slika govori da (zbog duple linije), kadgod se u bazu podataka unesu podaci o novoj osobi, ona mora da bude klasifikovana bilo kao klijent bilo kao zaposleni. Da je podtip bio opcionni, korisnik bi mogao, pri unošenju podataka o novoj osobi, da je klasifikuje kao klijenta ili zaposlenog, ali ne bi morao.

# Podtipovi: **disjunktni / preklapajući**

- Posmatrajmo sada nadtip *company* i podtipove *AR\_co* i *AP\_co*.
- Dizajner treba da odluči da li entitet tipa *AR\_co* može istovremeno da bude i entitet tipa *AP\_co*.
- Dve su mogućnosti:
  - Ako entiteti mogu da budu najviše jednog podtipa, tada su podtipovi *disjunktni*.
  - Ako neki entitet može da se klasifikuje u više podtipova, tada su podtipovi *preklapajući*.

# Podtipovi: disjunktни / preklapajući

- Disjunktни podtipovi obeleženi su sa “d” ispod linija nadtipa, preklapajući sa “o” (“overlapping”).
- Tumačenje slike:



- *Problem 3.3*
  - *Pretpostavimo da postoji tip entiteta student sa podtipovima CIS majors, finance majors, i accounting majors. Da li je ovo obavezni ili opcioni podtip? Da li su podtipovi disjunktни ili preklapajući? Kako bi još mogla da se predstavi opisana situacija (bez potipova)?*

# Atributi: klasifikacija

- Atribute je moguće klasifikovati na sledeći način:
- osnovni
  - To su vrednosti koje se obezbeđuju poslovanju. Takve smo atribute razmatrali do sada – na primer ime, adresa, itd. Ove vrednosti ne mogu da se izvedu iz drugih atributa.
- definisani
  - To su izmišljeni atributi i postoje samo u bazi podataka, na primer jedinstveni identifikator sektora. Jednom dodeljena vrednost se ne menja.
- izvedeni
  - Ovo je vrednost koja može da se izračuna iz vrednosti drugih atributa u bazi podataka. Na primer, starost zaposlenog kada je datum rođenja u bazi. Ovi atributi u opštem slučaju ne bi trebalo da se pamte u bazi podataka već da se izračunavaju po potrebi.

# Opcionost atributa

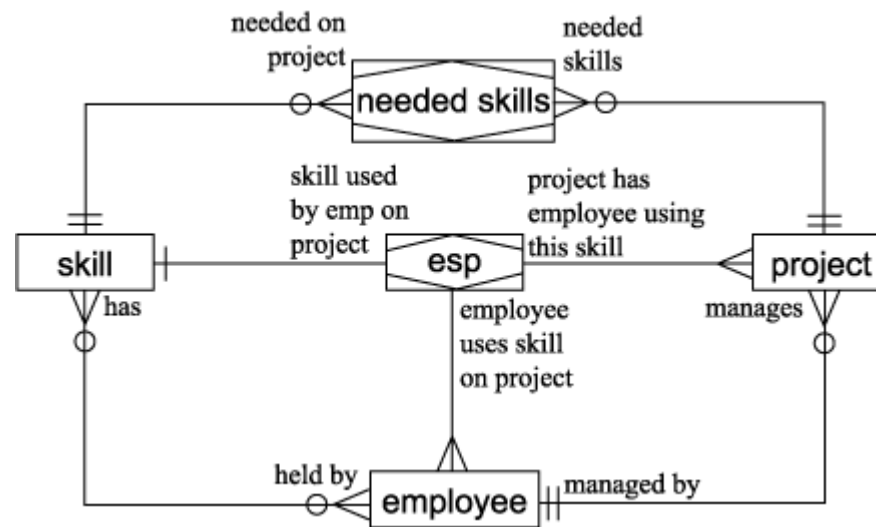
- Nemaju svi entiteti vrednost za svaki atribut; međutim, neki atributi moraju da imaju vrednost za sve entitete. Zato se atributi mogu klasifikovati u
  - opcione i
  - obavezne
- Na primer, tip entiteta *employee* (zaposleni) ima attribute *datum zaposlenja* i *datum prestanka*. *Datum zaposlenja* je svakako obavezni atribut jer ako nema datuma zaposlenja, nije zaposleni.
- *Datum prestanka* je opcioni 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 prikuplja i kako poslovanje ažurira svoju bazu podataka. Isti atribut može da bude različito klasifikovan od strane raznih kompanija.
- Razmotrimo sledeći primer:



# Opcionost atributa: primer

- Razmotrimo atribut *prodajna cena* tipa entiteta *kataloški proizvod* kompanije za naručivanje e-poštom. Kompanija *A* ima politiku da ne stavlja proizvod u katalog dok nema cenu; dakle, oni ne kreiraju entitet *kataloški proizvod* dok ne dodele vrednost njegovom atributu *prodajna cena*. Za ovu kompaniju, *prodajna cena* je obavezni atribut.
  - S druge strane, kompanija *B* ima politiku da stavlja proizvod u katalog čim odluči da ga skladišti. Na taj način prikazuju što je moguće širu lepezu proizvoda. Oni stave u katalog oznaku "Pozovite nas za najnovije informacije" (*Call us for latest quote*) umesto cene. Dakle, za ovu kompaniju *prodajna cena* je opcioni atribut.
- 
- *Problem 3.4*
  - *Za svaki od sledećih atributa, odrediti da li je opcioni ili obavezan:*
  - *Datum isporuke narudžbenice*
  - *Datum naručivanja narudžbenice*
  - *Naslov knjige*
  - *Serijski broj televizora*

# Interpretiranje ER dijagrama



# Interpretiranje ER dijagrama

- Interpretacija odnosa
- needed (potreban)
  - Jedna veština može da bude potrebna na većem broju projekata, a može da ne bude potrebna ni na jednom.
  - Jednom projektu može da bude potrebna nula ili više veština.
- manages (upravlja)
  - Jedan zaposleni može da upravlja većim brojem projekata a može da ne upravlja ni jednim.
  - Jednim projektom mora da upravlja (rukovodi) neki zaposleni.
- has-skill (ima-veštine)
  - Jedan zaposleni može da ima više veština, a može da nema ni jednu.
  - Jednu veštinu može da poseduje više zaposlenih. Ima i veština koje ne poseduje ni jedan zaposleni.
- used-on (korišćen):
  - Jedan zaposleni koristi jednu veštinu na *više* projekata.
  - *Više* zaposlenih može da koristi jednu veštinu na jednom projektu.
  - Jedan zaposleni može da koristi samo *jednu* veštinu na jednom projektu.
- Ovo čitanje se dobija “prekrivanjem” dve od tri grane ternarnog odnosa.