

Prof. Dr. Ermir Rogova

# Të dhënat e korporatave dhe menaxhimi i databazave

Pjesa 3



# UML (Unified Modeling Language)

- ▶ Është gjuhë grafike për komunikimin e specifikacioneve të dizajnit të softverit
- ▶ Është krijuar nga komuniteti i zhvillimit të softverit O-O për të përshkruar softverin OO në dizajn
- ▶ UML ka evoluar në një standard për dizajnin e sistemeve digjitale në përgjithësi

# Tipet e diagramit UML

- ▶ Janë disa tipe të ndryshme të diagrameve UML që përdoren për qëllime të ndryshme
- ▶ Tipi i klasave dhe ai i aktiviteteve janë jashtëzakonisht të dobishëm për çështjet në lidhje me dizajnimin e databazave
- ▶ Diagrami UML i klasave paraqet aspektet strukturale të skemave të databazave
- ▶ Diagrami UML i aktiviteteve paraqet proceset dinamike që përfshihen në dizajnin e databazave
- ▶ Këto koncepte janë të dobishme për planifikimin, dokumentimin, diskutimin, dhe implementimin e databazave

# Tipet e diagramit UML (2)

- ▶ Diagramet UML të klasave dhe ER modelet janë të ngjajshme në formë dhe kuptim
- ▶ Krijuesit e UMLsë indikojnë influencën e modelit ER në origjinën e diagramit të klasave
- ▶ Gjithashtu, UML ka pasur influencë në komunitetin e krijuesëve të databazave
- ▶ Diagramët e klasave tash paraqiten shpesh në literaturë kur përshkruhen skemat e databazave
- ▶ UML diagramët e aktiviteteve janë të ngjajshëm me hartat e rrjedhjes së informatave.

# UML diagrami i klasave (Class diagram)

- ▶ Klasa është përshkrues i objekteve që kanë të njëjtat karakteristika (attribute) dhe/ose operacione
- ▶ Konceptin e klasave e përdorim në jetën e përditshme, p.sh:
  - Vetura ka karakteristika si: nr. shasisë, ngjyra, km e kaluara, etj.
  - Vetura ka operacione si: përshpejtimi, ngadalsimi, kthimi, etj.
- ▶ Të gjitha veturat i kanë këto karakteristika dhe operacione

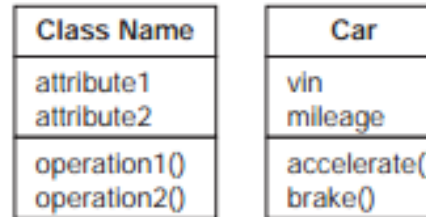
# UML dhe ER

- ▶ Ekziston një ngjajshmëri e madhe në mes të diagramit të klasave në UML dhe diagramit ER.
- ▶ Skemat e databazave mund të skicohen duke përdorur UMLKlasat janë analoge me entitetet
- ▶ Shemat e databazave mund të paraqiten duke përdorur UML-në. Tabela mund të konceptualizohet si klasë
- ▶ Kolonat e tabelës janë karakteristikat (atributet) dhe rreshtat janë objektet e asaj klase.
- ▶ Ndryshimi më i madh në mes të klasave dhe entiteteve është mungesa e operacioneve në entitete

# Diagram i thjeshtë i klasave

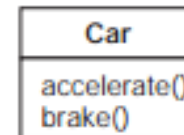
## Classes

Notation and Example

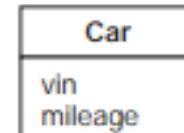


Notational Variations

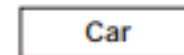
Emphasizing Operations



Emphasizing Attributes



Emphasizing Class



## Relationships

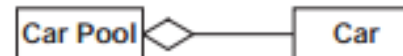
Association



Generalization



Aggregation



Composition

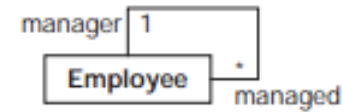


# ER VS UML

Concept	Representation & Example
Degree recursive binary	
binary	
ternary	
Connectivity one-to-one	
one-to-many	
many-to-many	
Existence optional	
mandatory	

Degree

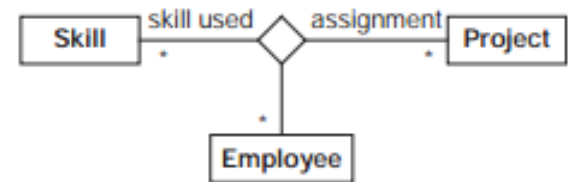
reflexive  
association



binary  
association

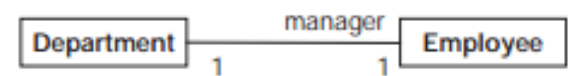


ternary  
association



Multiplicities

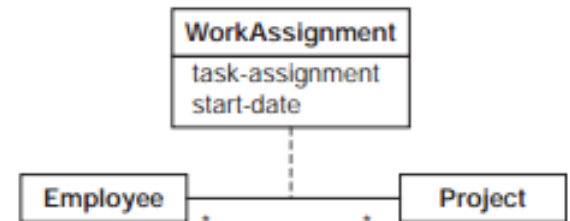
one-to-one



one-to-many

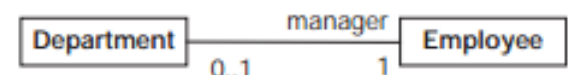


many-to-many

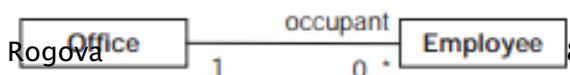


Existence

optional

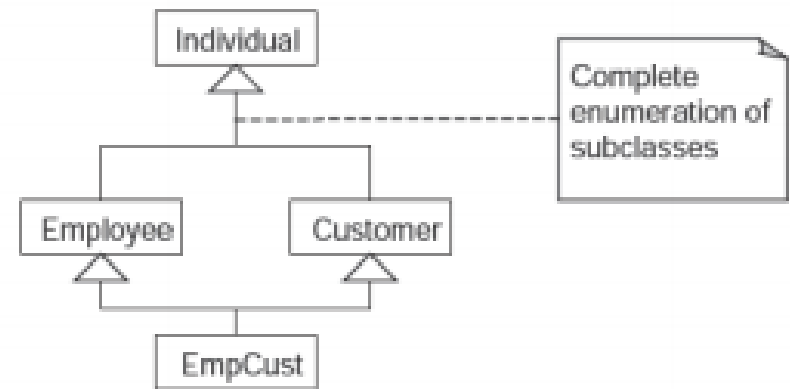
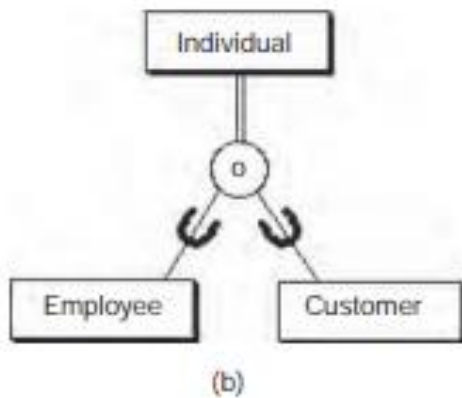
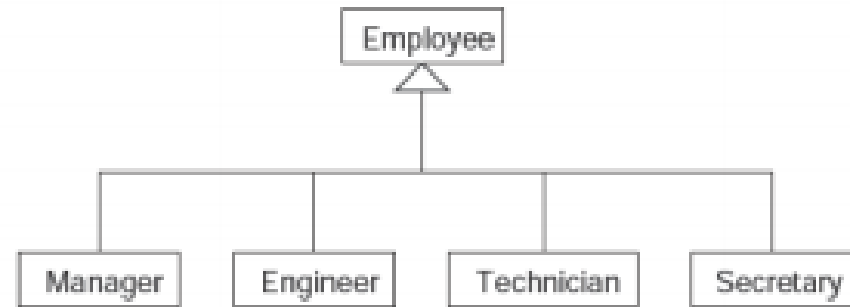
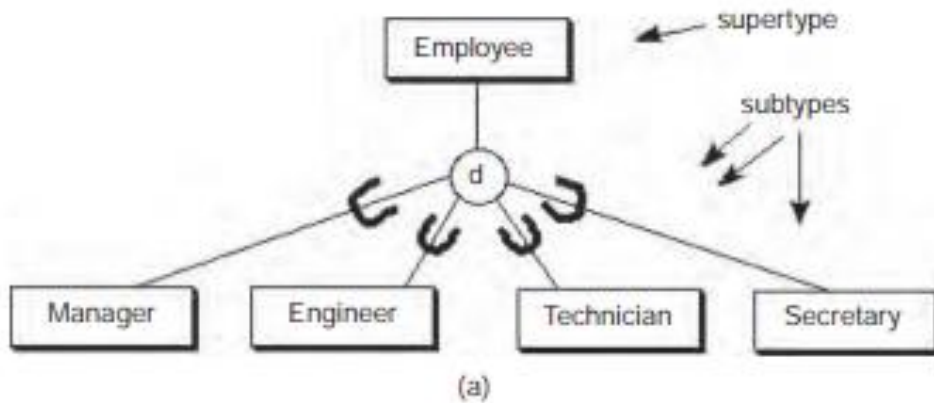


mandatory





# Gjeneralizimi

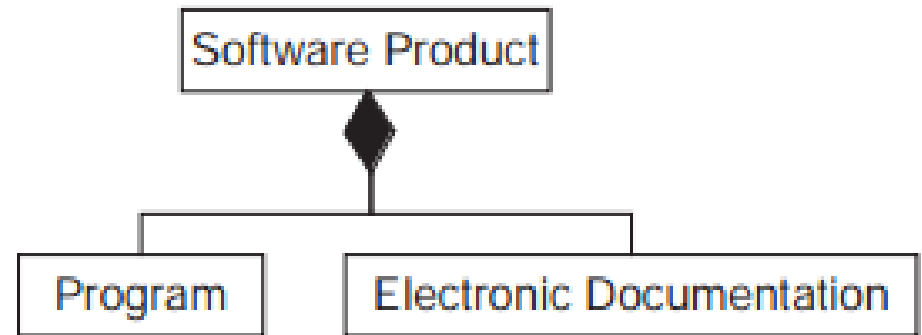


# përmbajtja dhe Grumbullimi (composition and aggregation)

## Përmbajtja shembull:

Programi dhe Dokumentacioni kontribuojnë në përmbajtjen e Produktit Softverik.

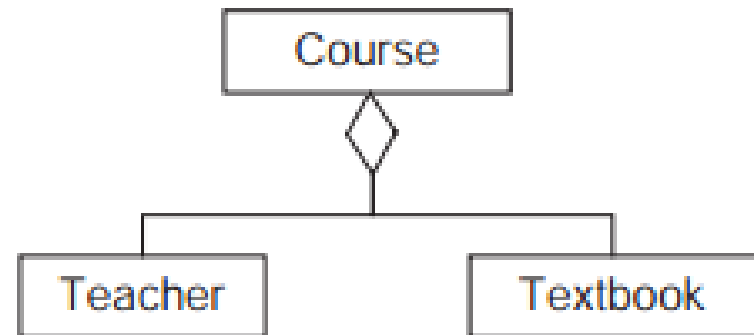
Pjesët nuk ekzistojnë pa Produktin Softverik (ska pirateri!!!)



## Grumbullimi shembull:

Mësuesi dhe libri i lëndës grumbullohen në lëndë, por ekzistojnë edhe veçmas

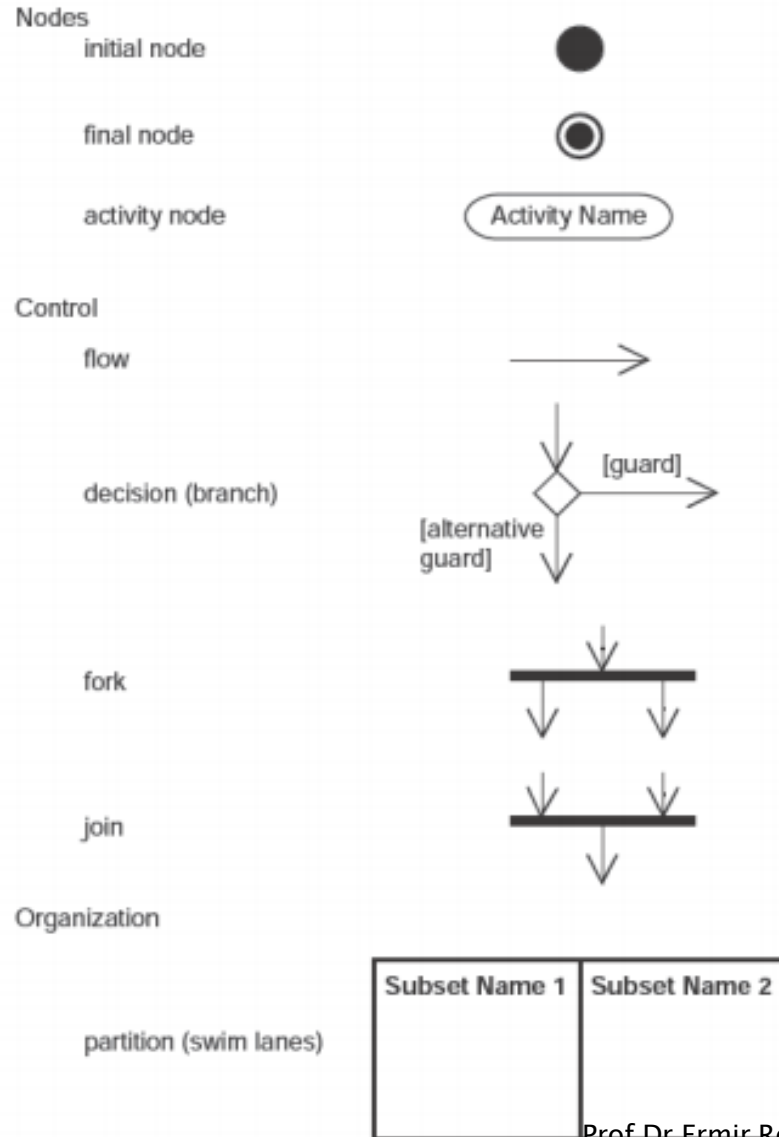
Nëse lënda anulohet, mësuesi dhe libri vazhdojnë të ekzistojnë



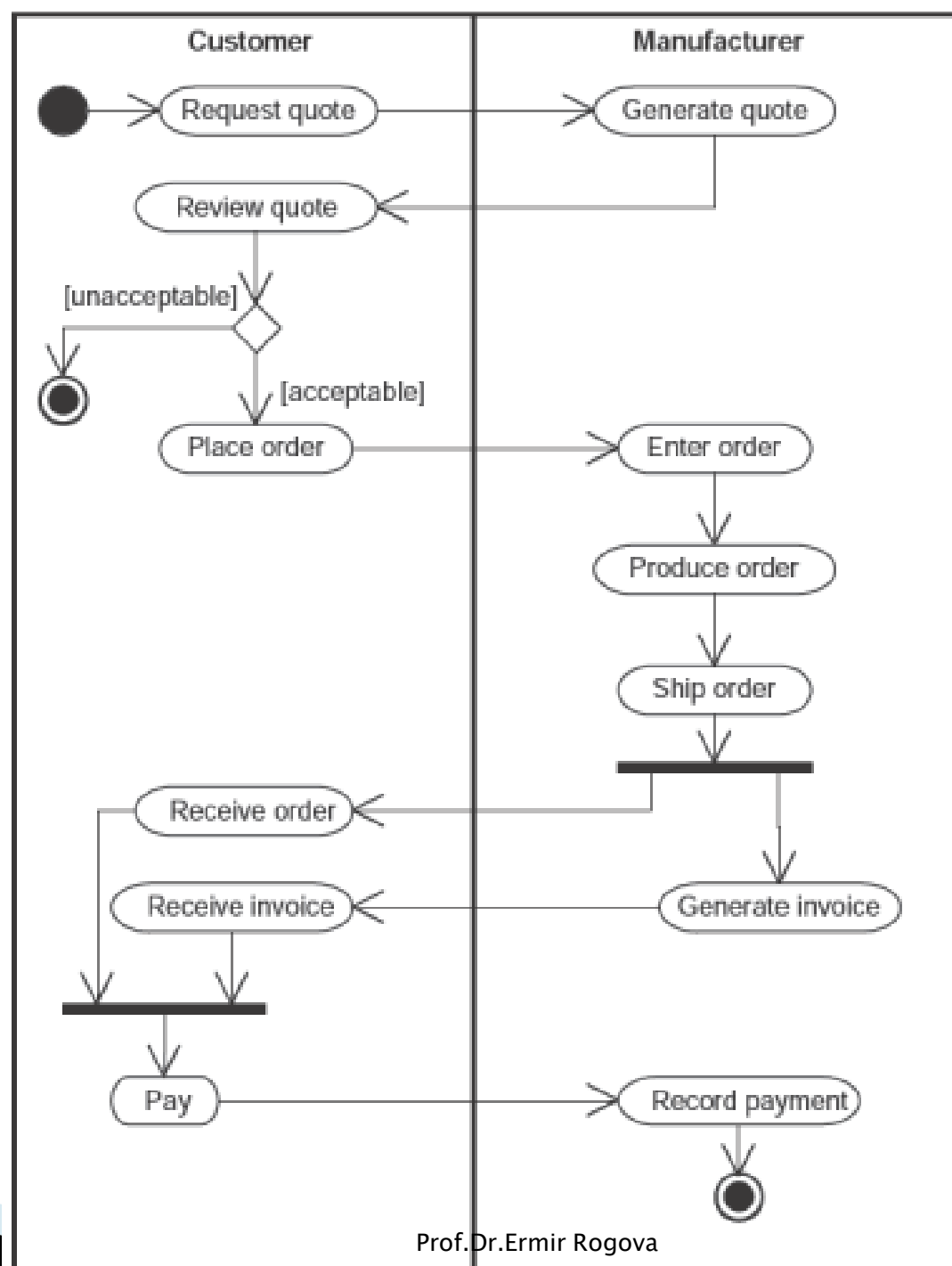
# Diagrami i aktiviteteve

- ▶ Diagrami i aktiviteteve përdoren që të specifikojnë aktivitetet dhe kontrollin e rrjedhjes në një proces
- ▶ Procesi mund të jetë rrjedhje e punës që ndiqet nga njerëzit, organizatat ose gjëra të tjera fizike (prodhimtaria)
- ▶ Procesi mund të jetë edhe algoritëm i implementuar në softver
- ▶ Sintaksa dhe semantika(kuptimi) i konstrukteve UML është e njëjtë, pa marrur parasysh procesin e përshkruar

# Konstruktet e diagramit te aktiviteteve



# Shembull



# ER dhe UML në ciklin e jetës

- ▶ Në këtë pjesë do të shikojmë se si diagramet ER dhe UML aplikohen në ciklin e jetës së databazës
- ▶ Kjo përfshin hapat prej analizës së kerkesave deri te integrimi i pamjeve
- ▶ Qasja në dizajnimin logjik që do të shtjellohet formon nje kornizë punuese për transformimin e konstrukteve konceptuale në tabela.

# Analiza e kërkesave

- ▶ Është hapi i parë në ciklin e jetës së databazës dhe zakonisht kërkon punë intensive
- ▶ Dizajnuesi duhet të intervistojë përdoruesët dhe të përcaktojë saktësisht se:
  - për çka do të përdoret databaza dhe
  - çka do të përmbajë ajo

# Objektivat e analizës së kërkesave

- ▶ Të paraqesë kërkesat për të dhëna në atë ambient në formë të elementeve themelore të të dhënave (data elements)
- ▶ Të përshkruaj informacionin mbi elementët e të dhënave dhe marrëdhëniet në mes tyre
- ▶ Të përcaktojë llojet e transaksioneve që planifikohet të ekzekutohen në databazë dhe interaksionin në mes të tyre dhe elementeve të të dhënave



# Objektivat e analizës së kërkesave (2)

- ▶ Të definojë kufizimet që duhen të imponohen në databazën rezultuese si:
  - Integritetit
  - Përformancës
  - Sigurisë
  - Administrativë
- ▶ Të specifikojë kufizimet dizajnuese apo implementuese si:
  - Teknologjitë specifike
  - Hardverin dhe softverin
  - Gjuhët programuese
  - Politikat
  - Standardet
  - Interfejsët e jashtëm

# Objektivat e analizës së kërkesave (3)

- ▶ Të dokumentojë rrënjësisht të gjitha pikat e lartëpërmendura në dokument të detalizuar të kërkesave
- ▶ Elementet e të dhënave (data elements) duhet të definohen në një fjalor të të dhënave i cili zakonisht jepet si pjesë integrale e sistemit të menagjimit të databazës

# Modelimi konceptual

- ▶ Modelimi konceptual realizohet duke kryer hapat e mëposhtëm:
  - Klasifikimi i entiteteve dhe attributeve (klasët dhe atributet -> UML)
  - Identifikimi i hierarkive gjeneralizuese
  - Definimi i marrëdhënieve (asocijimeve ->UML)

# Klasifikimi i entiteteve dhe attributeve

- ▶ Si vendoset se a është një data element entitet, atribut apo marrëdhënie?
  - P.sh. Vendndodhja e projekteve është nëpër qytete. A është qyteti entitet apo atribut?
- ▶ Udhëzimet për klasifikimin e entiteteve dhe attributeve janë:
  - Entitetet duhet të përmbajnë informata përshkruese
  - Atributet shumëvlerëshe duhet të klasifikohen si entitete
  - Atributet duhen ti bashkangjiten entiteteve të cilat i përshkruajnë në mënyrën më direkte

# Përmbajtja e entiteteve

- ▶ Nëse ekziston informacion përshkrues për data elementin, atëherë ai duhet të klasifikohet si entitet
- ▶ Nëse elementit i nevojitet vetem një identifikues dhe nuk ka marrëdhënie, atëhere duhet të klasifikohet si atribut.
- ▶ Qyteti:
  - Entitet, (popullata, sipërfaqja, etj)
  - Atribut (identifikues i vendndodhjes se projektit)
  - Si entitet nëse identiteti duhet të kufizohet

# Atributet shumëvlerëshe

- ▶ Kur një atribut mund të ketë shumë vlera, p.sh. Nr. Tel apo adresat e e-mailit të studentit
- ▶ Këto attribute duhet të riklasifikohen si entitete dhe të lidhen me entiteti prej të cilit rrjedhin
- ▶ Me këtë arrihet thjeshtësimi i dizajnit dhe implementimit

# Bashkangjitja e attributeve

- ▶ Atributet ti bashkangjiten entiteteve te cilat i përshkruajnë sa me direkt
  - P.sh. Salla\_e\_mesimit duhet ti bashkangjitet entitetit Objekti në vend të entitetit Studenti
- ▶ Procedura e identifikimit te entiteteve dhe attributeve është iterative (përsëritëse)
- ▶ Gjatë modelimit shpesh elementet ndyshojnë vendet prej attribute në entitete dhe anasjelltas, varësisht se a i shkelin apo jo udhëzimet e dhëna

# Identifikimi i gjeneralizimeve

- ▶ P.sh. Nëse identifikohen 5 entitete si:
  - Employee, with identifier empno and descriptors empname, address, and date-of-birth.
  - Manager, with identifier empno and descriptors empname and jobtitle.
  - Engineer, with identifier empno and descriptors empname, highest-degree, and jobtitle.
  - Technician, with identifier empno, and descriptors empname and specialty.
  - Secretary, with identifier empno, and descriptors empname and best-skill.



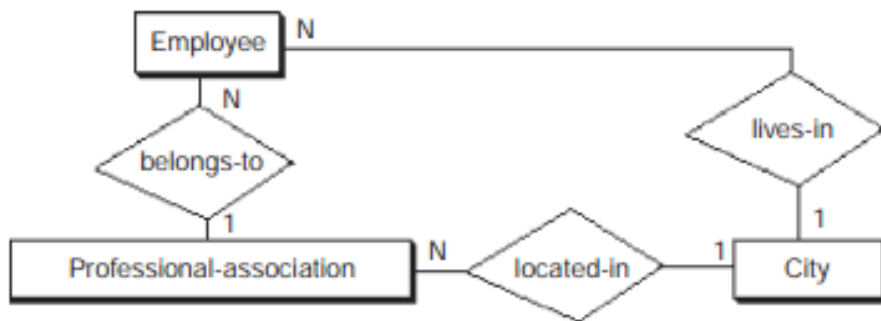
# Identifikimi i gjeneralizimeve (2)

- ▶ Me analizim përcaktojmë se entiteti Employee mund të krijohet si gjeneralizim për Manager, Engineer, Technician dhe Secretary
- ▶ Vendosim empno dhe përshkruesit e përgjithshëm si, empname, address, DOB në supertip, kurse empno dhe përshkruesit specifik në secilin nëntip (subtip)
- ▶ Nëse më vonë vendosim të eliminojmë entitetin employee, atëherë identifikuesi dhe përshkruesit e përgjithshëm rishpërndahen

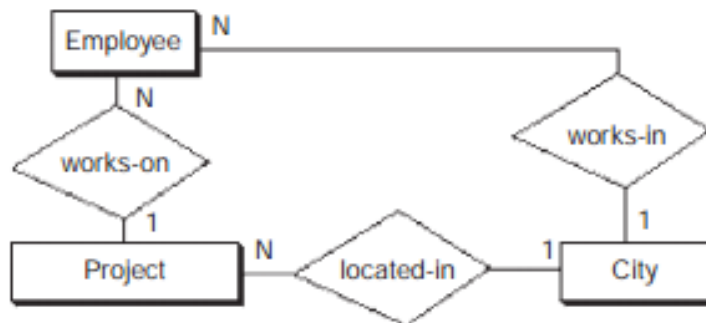
# Definimi i marrëdhënieve

- ▶ Marrëdhëniet janë data elementet që paraqesin lidhje (asocijim) në mes entiteteve
- ▶ Gramatikisht identifikohen me folje (punon–ne, menagjon, blen, ligjëron, ndëgjon, etj)
- ▶ Për secilën marrëdhënie duhet të specifikohen:
  - Shkalla (binare, trinare, etj)
  - Lidhja (1–1, 1–M, N–M)
  - Ekzistenca (opcionale apo e detyrueshme)
  - Çfardo atributi që është asociuar me marrëdhënien e jo entitetet

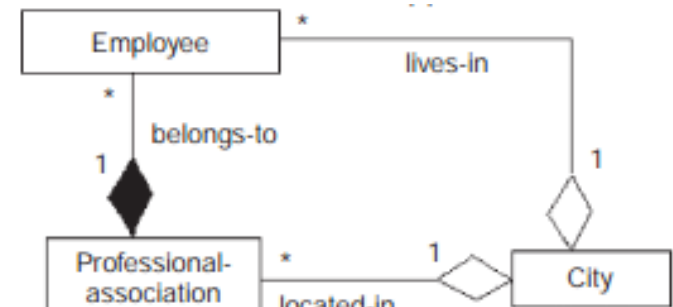
# Marrëdhëniet e tepërta (redundante)



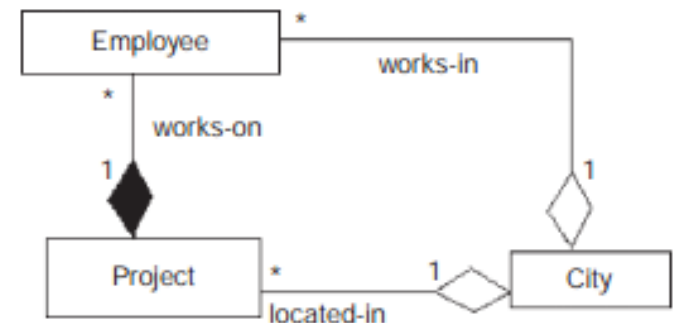
(a)



(b)

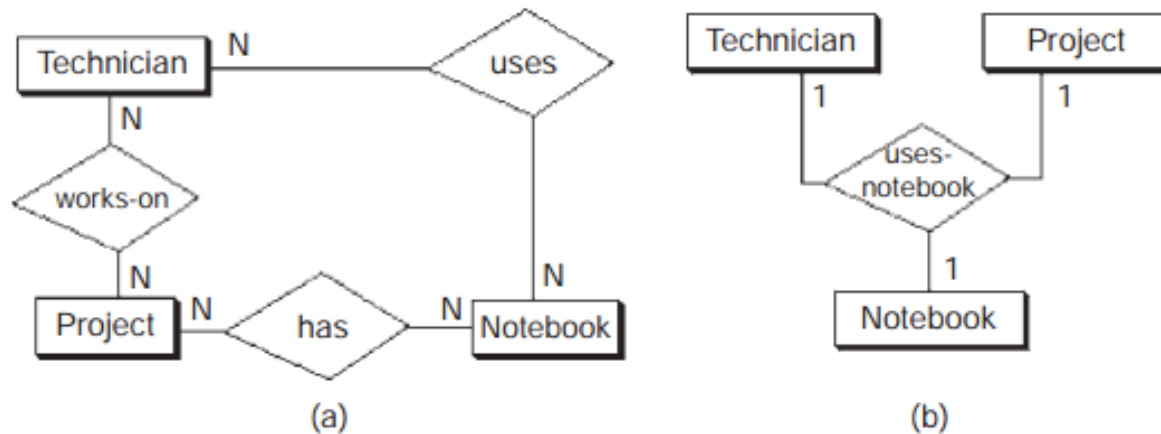


(c)



(d)

# Marrëdhëniet trinare (shembull)



- ▶ Nëse secili teknik mund të punojë në secilin prej disa projekteve duke përdorur të njëjtën fletore shënimesh në secilin projekt, atëherë mund të definohen tri m-m marrëdhënie.
- ▶ Nëse secili teknik është i kufizuar të përdorë saktësisht 1 fletore për secilin projekt dhe ajo fletore i takon vetëm një tekniku, atëherë duhet të definohet një marrëdhëni trinare 1-1-1

**Pyetje ???**