The Enhanced Entity-Relationship (EER) Model

406.426 Design & Analysis of Database Systems

Jonghun Park

jonghun@snu.ac.kr

Dept. of Industrial Engineering

Seoul National University

outline

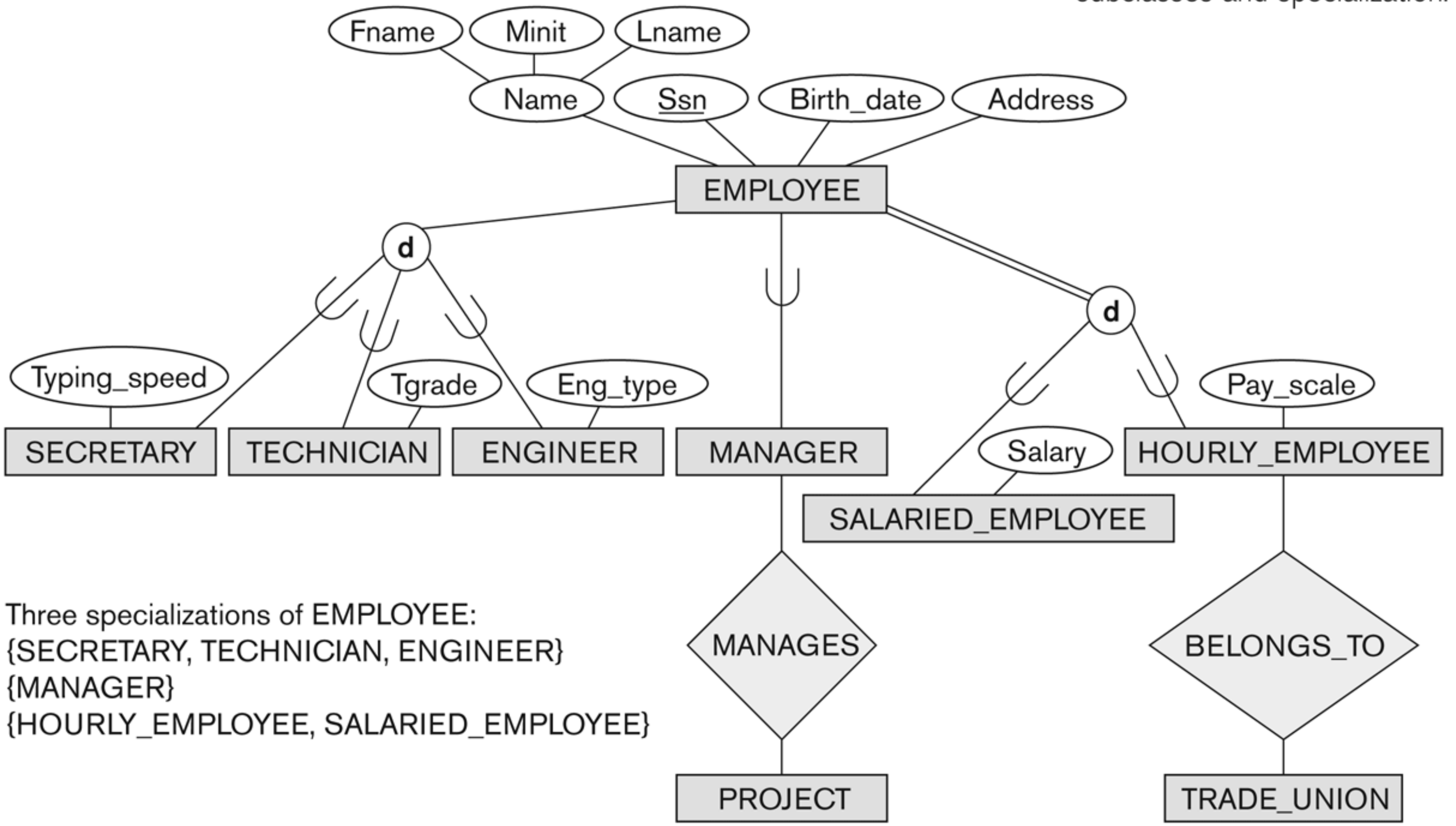
- EER model concepts
  - includes all modeling concepts of basic ER
  - additional concepts:
    - subclasses/superclasses
    - specialization/generalization
    - categories (UNION types)
    - attribute and relationship inheritance
  - these are fundamental to conceptual modeling
- the additional EER concepts are used to model applications more completely and more accurately
  - EER includes some object-oriented concepts, such as inheritance

subclasses, superclasses, and inheritance

- EER (Enhanced ER)
  - ER + subclass, superclass, specialization, generalization, category, attribute and relationship inheritance
- superclass / subclass relationship
  - example: SECRETARY, ENGINEER, MANAGER, … -> a **subset** of the entities that belong to the EMPLOYEE entity set
  - an entity cannot exist in the DB merely by being a member of a subclass -> it **must also be a member of the superclass**
  - it is not necessary that every entity in a superclass be a member of some subclass
  - an entity that is a member of a subclass **inherits all the attributes** of the entity as a member of the superclass
  - the subclass entity also **inherits all the relationships** in which the superclass participates

EER diagram for subclasses and specialization

Figure 4.1
EER diagram notation to represent subclasses and specialization.

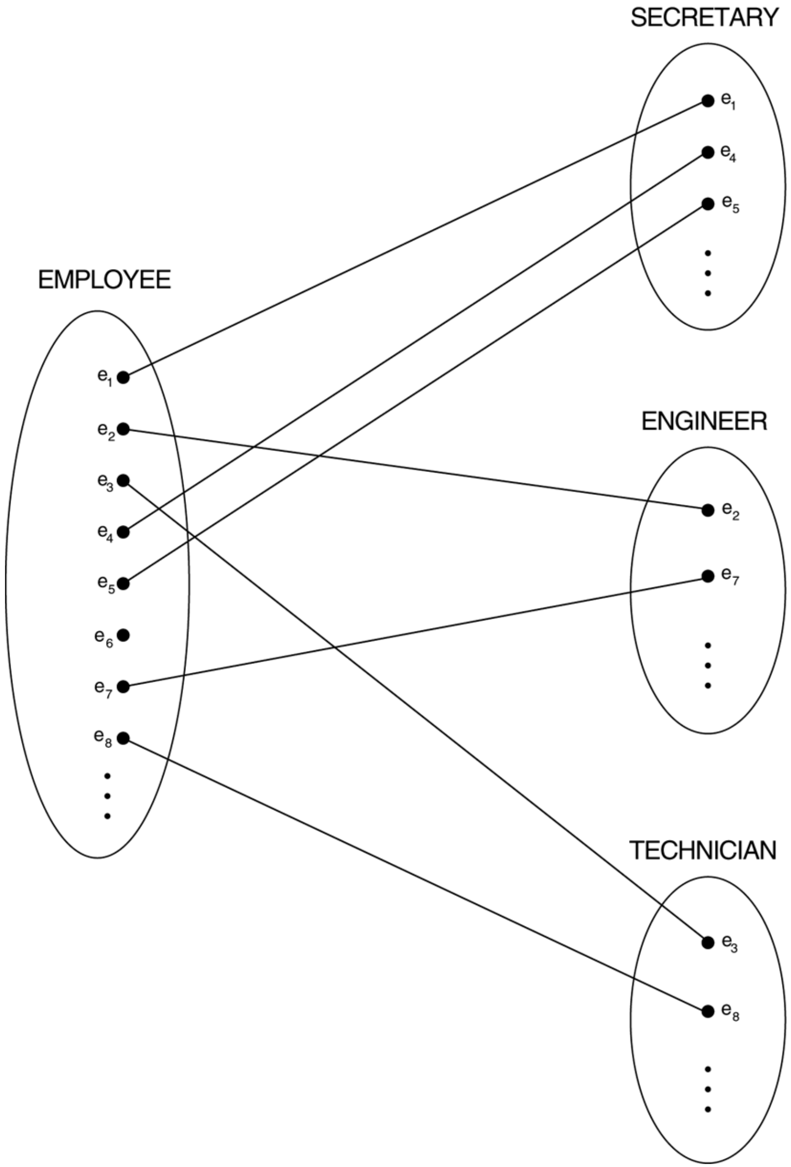


Three specializations of EMPLOYEE:
{SECRETARY, TECHNICIAN, ENGINEER}
{MANAGER}
{HOURLY_EMPLOYEE, SALARIED_EMPLOYEE}

specialization

- specialization: process of **defining a set of subclasses** of an entity type
  - defined on the basis of some **distinguishing characteristic** of the entities in the superclass
- specific attributes
  - attributes that apply **only to entities of a particular subclass**
- specific relationship types
  - e.g., BELONGS_TO relationship type
- allows us to
  - define a set of **subclasses** of an entity type
  - establish **additional specific attributes** with each subclass
  - establish **additional specific relationship types** between each subclass and other entity types or other subclasses

instances of a specialization



generalization

- generalization: the process of **generalizing** several entity types into a **single superclass** of which the original entity types are special subclasses
  - e.g., CAR, TRUCK -> VEHICLE
  - the inverse of the specialization process

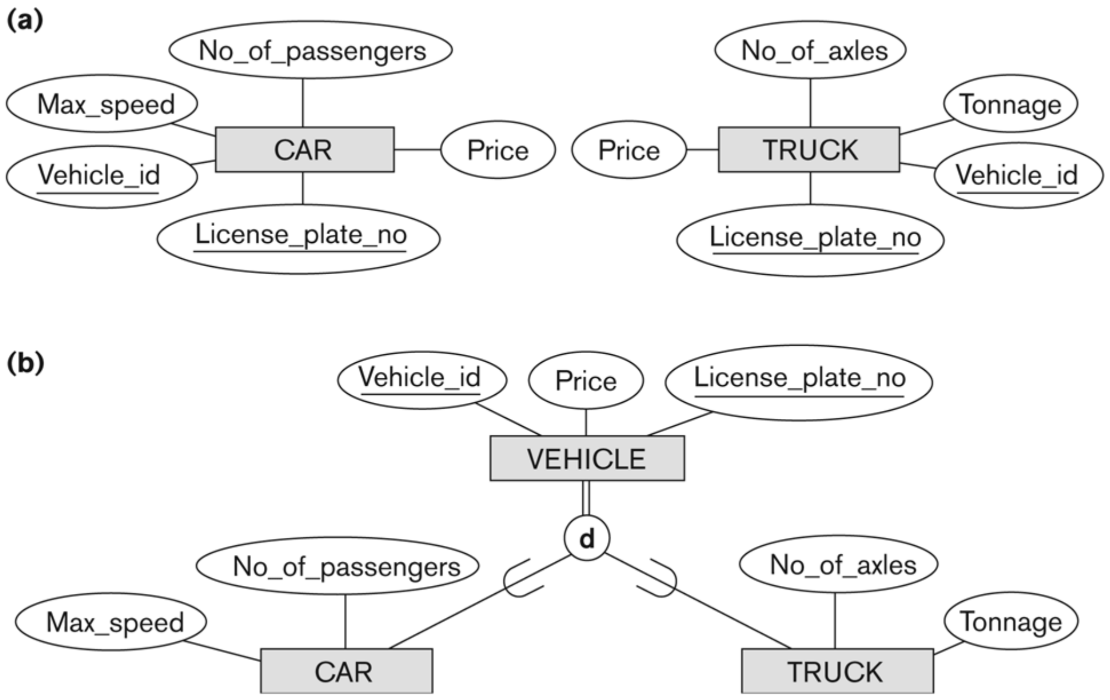
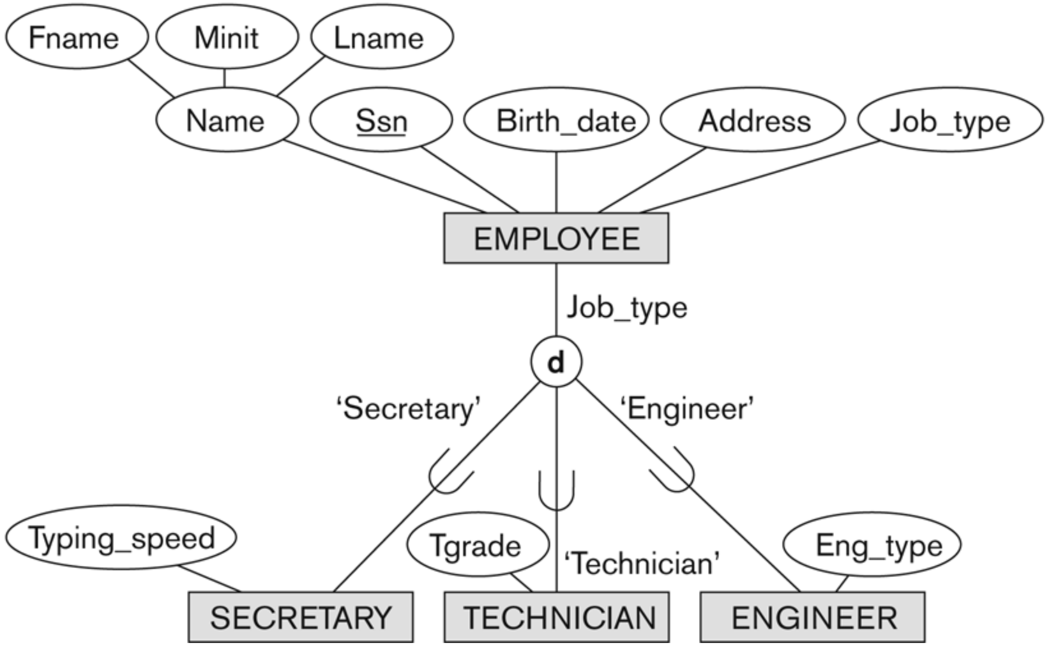


Figure 4.3
Generalization. (a) Two entity types, CAR and TRUCK.
(b) Generalizing CAR and TRUCK into the superclass VEHICLE.

constraints on specialization and generalization

- in general, we may have several specializations defined on the same entity type
- **predicate-defined** subclasses: the entities that become members of each subclass by placing a **condition** on the value of some attribute of the superclass
  - e.g., (JobType = ‘Secretary’) on SECRETARY
- if all subclasses in a specialization have their membership condition on the same attribute of the superclass, the specialization is called an **attribute-defined specialization**
- user-defined: specified individually **for each entity** by the user

Figure 4.4
EER diagram notation for an attribute-defined specialization on Job_type.



constraints on the specialization

- **disjointness** constraint
  - specifies that the subclasses of the specialization must be disjoint
  - an entity can be a member of **at most one** of the subclasses of the specialization
  - a specialization that is **attribute defined** implies the **disjointness** constraint if the attribute used to define the membership predicate is **single-valued**
  - notations: d, o

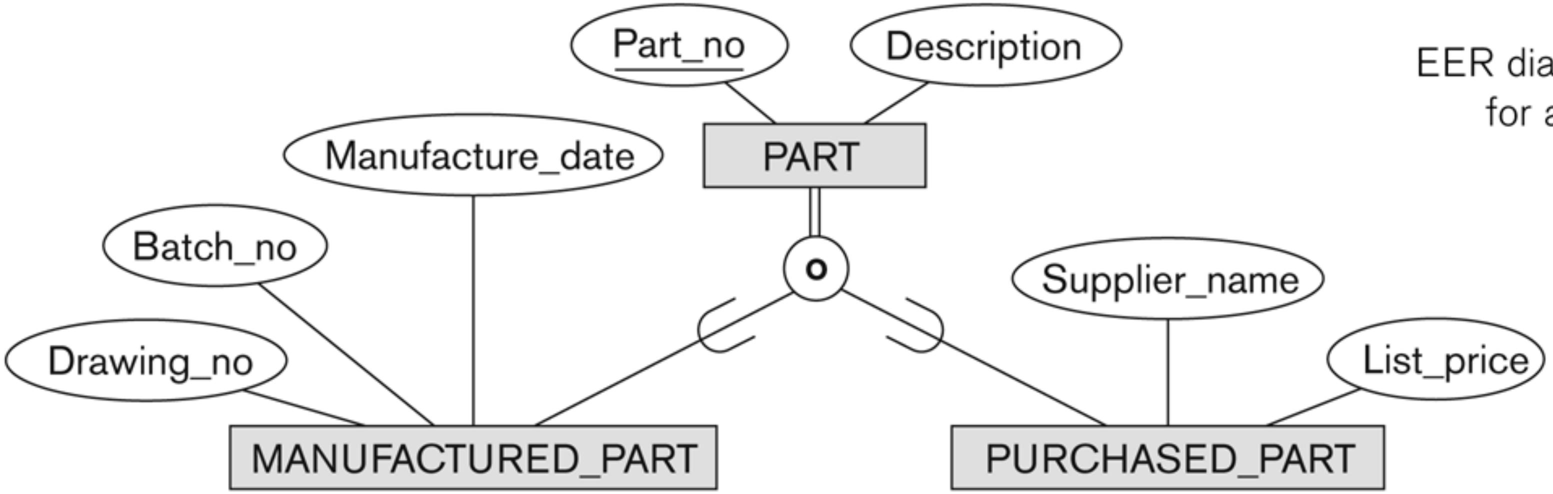


Figure 4.5
EER diagram notation for an overlapping (nondisjoint) specialization.

constraints on the specialization

- **completeness** constraint
  - **total specialization**: every entity in the superclass must be a member of at least one subclass in the specialization
  - **partial specialization**: **allows an entity not to belong to any of the subclasses**
- disjointness and completeness constraints are **independent**
  - i.e., disjoint, total / disjoint, partial / overlapping, total / overlapping, partial
- in general, a superclass that was identified through the **generalization** process usually is **total**
- some rules on the insertion and deletion
  - **deleting** an entity from a superclass implies that it is automatically deleted from all the subclasses to which it belongs
  - **inserting** an entity in a superclass implies that the entity is mandatorily inserted in all predicate-defined (or attribute-defined) subclasses for which the entity satisfies the defining predicate
  - **inserting** an entity in a superclass of a total specialization implies that the entity is mandatorily inserted **at least one** of the subclasses of the specialization

specialization hierarchies vs. lattices

- specialization **hierarchies**
  - every subclass must participate as a subclass in only **one** class/subclass relationship
- specialization **lattices**
  - a subclass can be a subclass in more than one class/subclass relationship

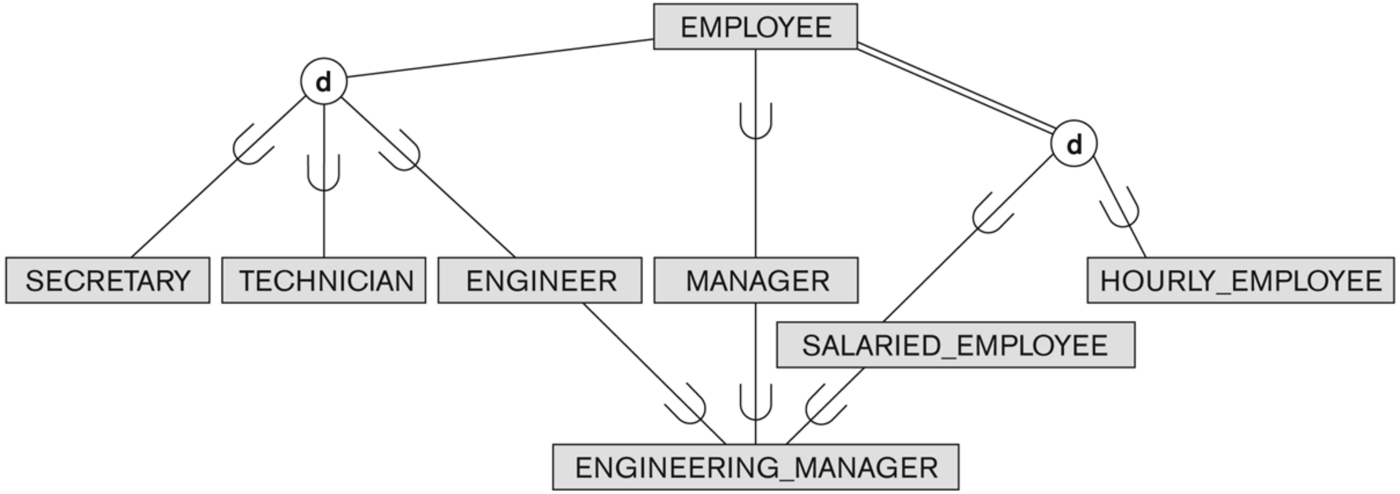


Figure 4.6
A specialization lattice with shared subclass ENGINEERING_MANAGER.

characteristics of specialization

- a subclass inherits the attributes not only of its direct superclass but also of all its **predecessor** superclasses all the way to the root of the hierarchy or lattice
- an entity may exist in several leaf nodes of the hierarchy
  - e.g., GRADUATE_STUDENT and RESEARCH_ASSISTANT
- a subclass with more than one superclass is called a **shared subclass**
- some models do not allow multiple inheritance
  - need to create additional subclasses to cover all possible combinations of classes that may have some entity belong to all these classes simultaneously
  - e.g., {EMPLOYEE, ALUMNUS, STUDENT} -> E, A, S, E_A, E_S, A_S, E_A_S

example

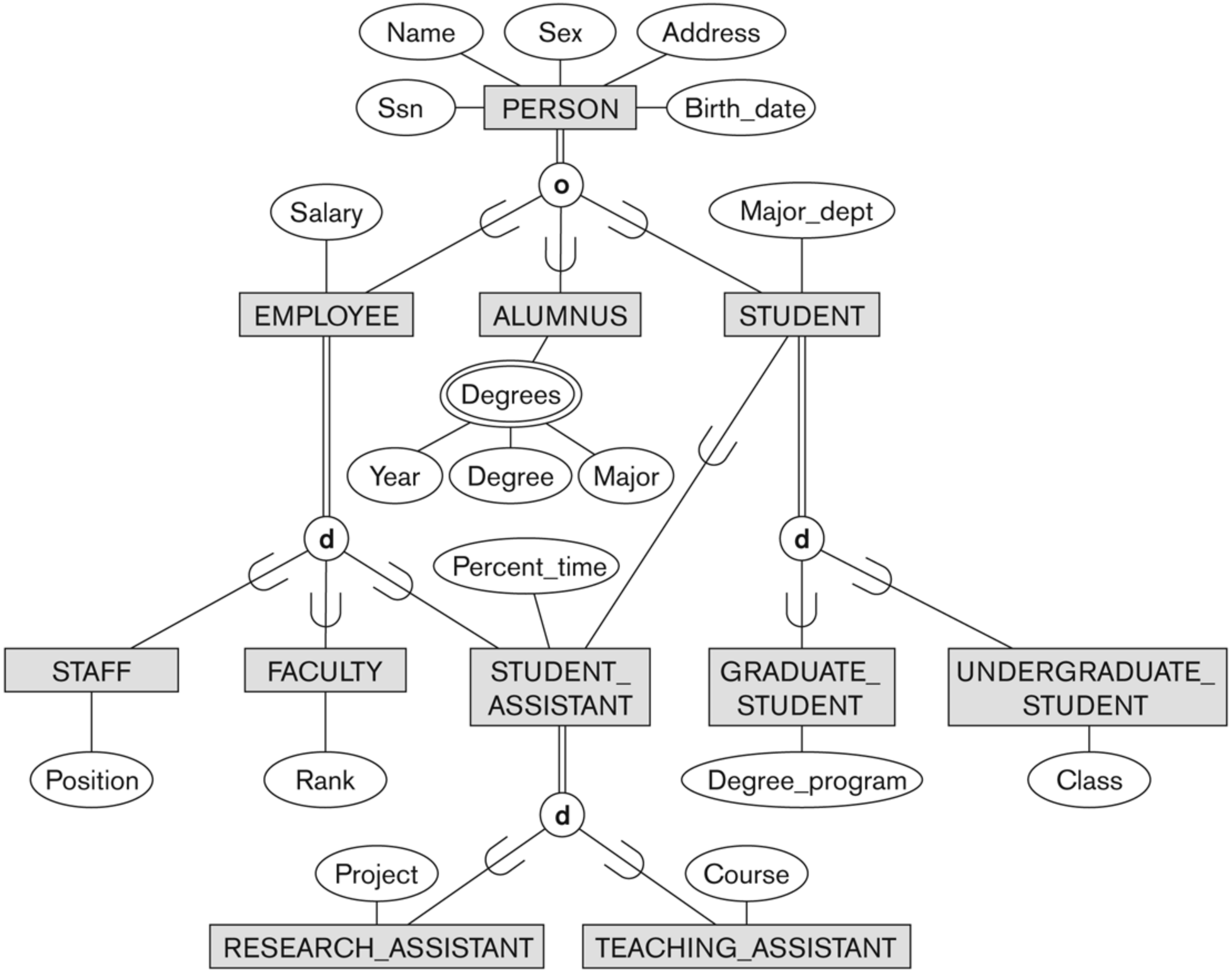


Figure 4.7
A specialization lattice with multiple inheritance for a UNIVERSITY database.

categories

- modeling a single superclass / subclass relationship with **more than one superclass**
- the subclass will represent a collection of objects that is a **subset** of the **UNION** of distinct entity types: called a **category**
  - example: OWNER

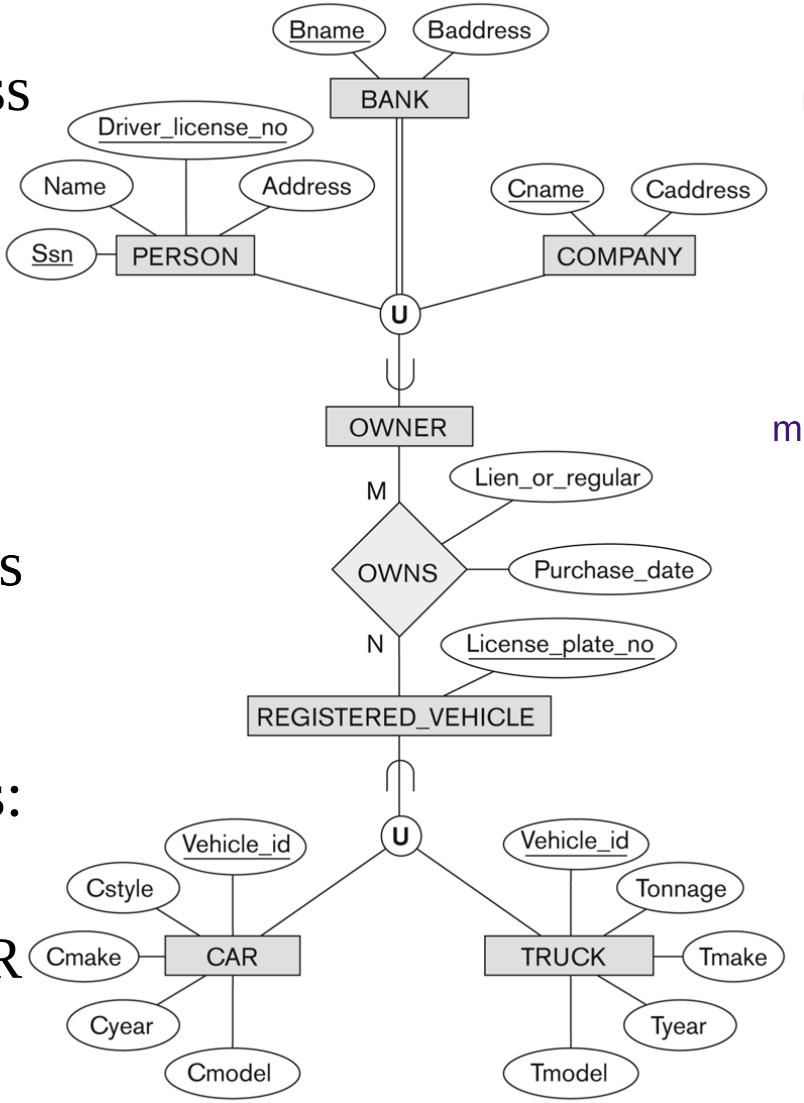


Figure 4.8
Two categories (union types): OWNER and REGISTERED_VEHICLE.

mushroom

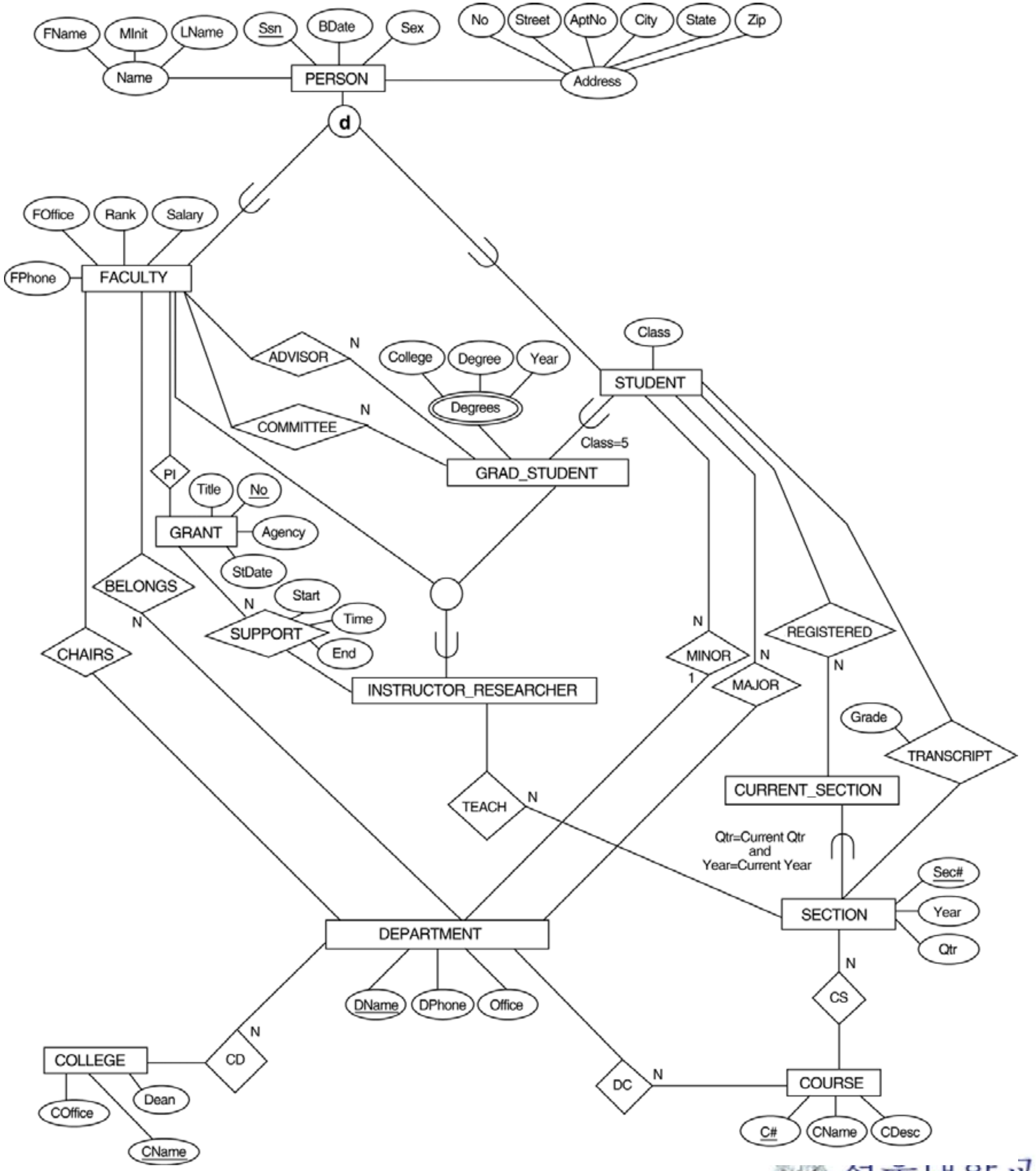
person

skull

category (cont.)

- comparison with the shared subclass
  - shared subclass
    - a subset of the **intersection** of the three superclasses
    - inherits all the attributes of its superclasses
  - category
    - a subset of the **union** of its superclasses
    - **inherits** the attributes **selectively** depending on the superclass to which the entity belongs
- category can be total or partial
  - a total category holds the union of all **entities** in its superclass
  - a partial category holds a subset of the union
  - if a category is total, it may be represented alternatively as a **total specialization (or a total generalization)**

summing up



FName
MInit
LName
Ssn
BDate
Sex
No
Street
AptNo
City
State
Zip
Name
PERSON
Address
d
FOffice
Rank
Salary
FPhone
FACULTY
Class
ADVISOR
N
College
Degree
Year
STUDENT
Degrees
COMMITTEE
N
Class=5
GRAD_STUDENT
PI
Title
No
GRANT
Agency
StDate
BELONGS
N
Start
N
Time
SUPPORT
End
CHAIRS
INSTRUCTOR_RESEARCHER
N
MINOR
1
N
MAJOR
REGISTERED
N
Grade
TRANSCRIPT
TEACH
N
CURRENT_SECTION
Qtr=Current Qtr and Year=Current Year
Sec#
Year
Qtr
SECTION
DEPARTMENT
N
DName
DPhone
Office
CS
N
CD
COLLEGE
DC
N
COURSE
Dean
COffice
CName
C#
CName
CDesc

formal definitions for the EER model

- class: a set or collection of entities
  - e.g., entity types, subclasses, superclasses, categories
- subclass S is a class whose entities must always be a subset of the entities in another class, called the superclass C of the superclass/subclass (or IS-A) relationship (denoted by C/S)
  - i.e., $S \subseteq C$
- a specialization $Z = \{S_1, S_2, \ldots, S_n\}$ is a set of subclasses that have the same superclass G (denoted by G/S_i)
  - G is called a generalized entity type
  - Z is said to be total if $\cup S_i = G$; partial, o.w.
  - Z is said to be disjoint if $S_i \cap S_j = \phi$ for $i \neq j$; overlapping, o.w.

formal definitions for the EER model (cont.)

- a subclass *S* of *C* is said to be **predicate-defined** if a predicate *p* on the attributes of *C* is used to specify which entities in *C* are members of *S*
  - $S = C[p]$
- a specialization *Z* (or generalization *G*) is said to be **attribute-defined** if a predicate ($A = c_i$), where *A* is an attribute of *G* and c_i is a constant value from the domain of *A*, is used to specify membership in each subclass S_i in *Z*
- a category *T* is a class that is a subset of the union of *n* defining superclasses, D_1, D_2, …, D_n, $n > 1$
  - $T \subseteq (D_1 \cup D_2 \ldots \cup D_n)$
- a predicate p_i on the attributes of D_i can be used to specify the members of each D_i that are members of *T*
  - $T = (D_1[p_1] \cup D_2[p_2] \ldots \cup D_n[p_n]\ )$

specialization/generalization in UML

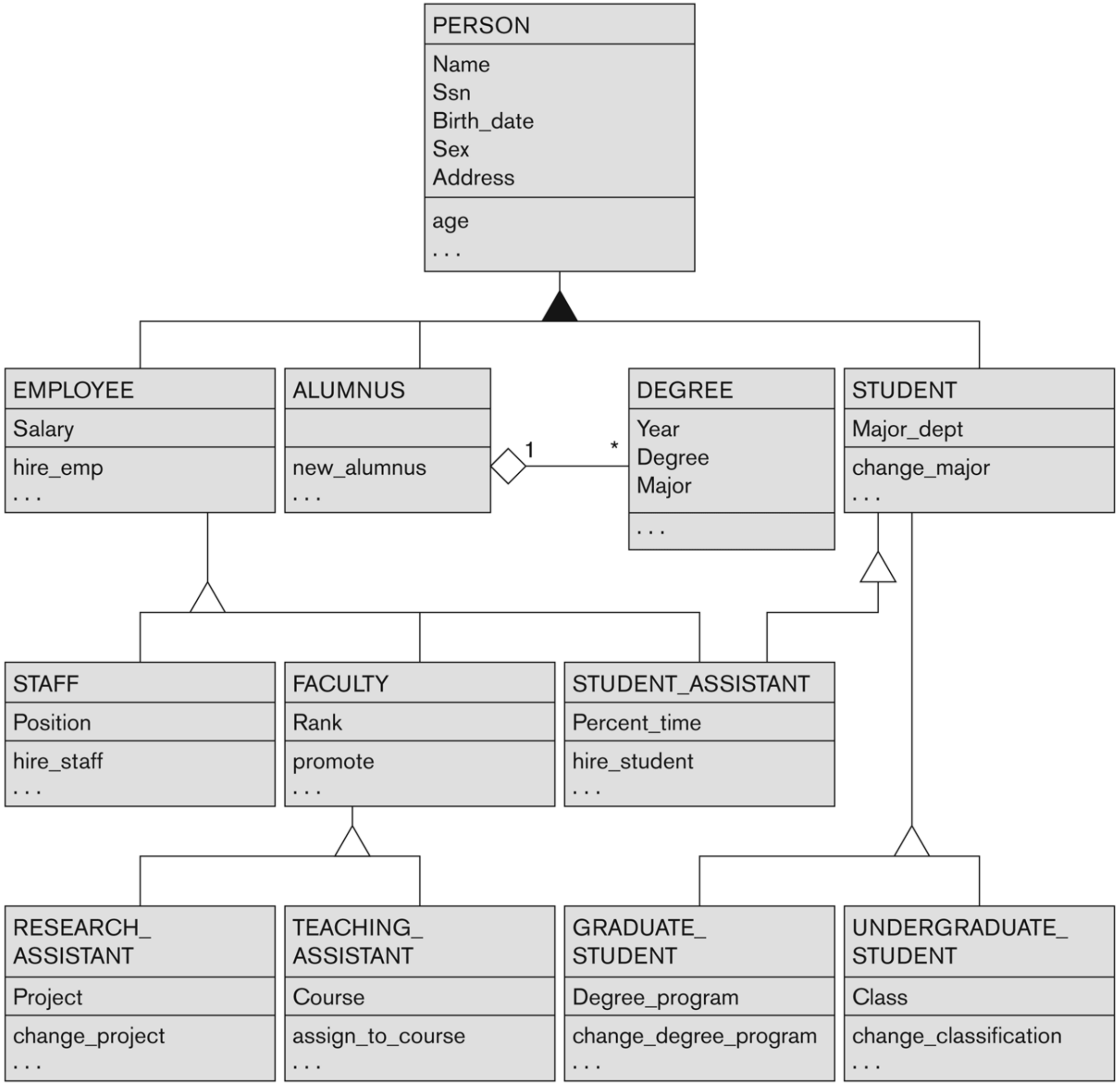
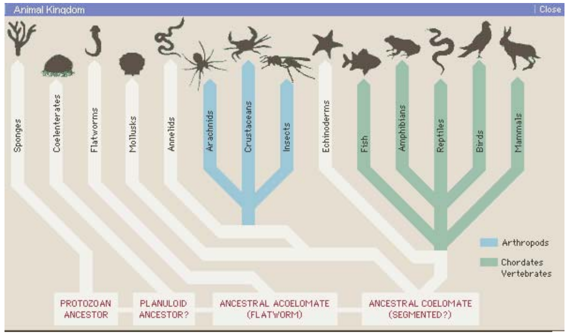
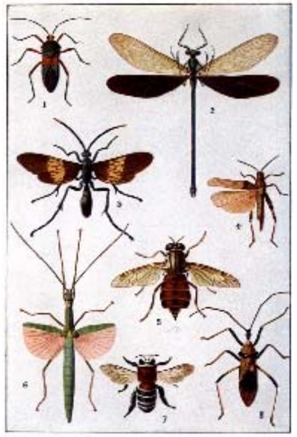


Figure 4.10
A UML class diagram corresponding to the EER diagram in Figure 4.7, illustrating UML notation for specialization/generalization.

abstraction concepts

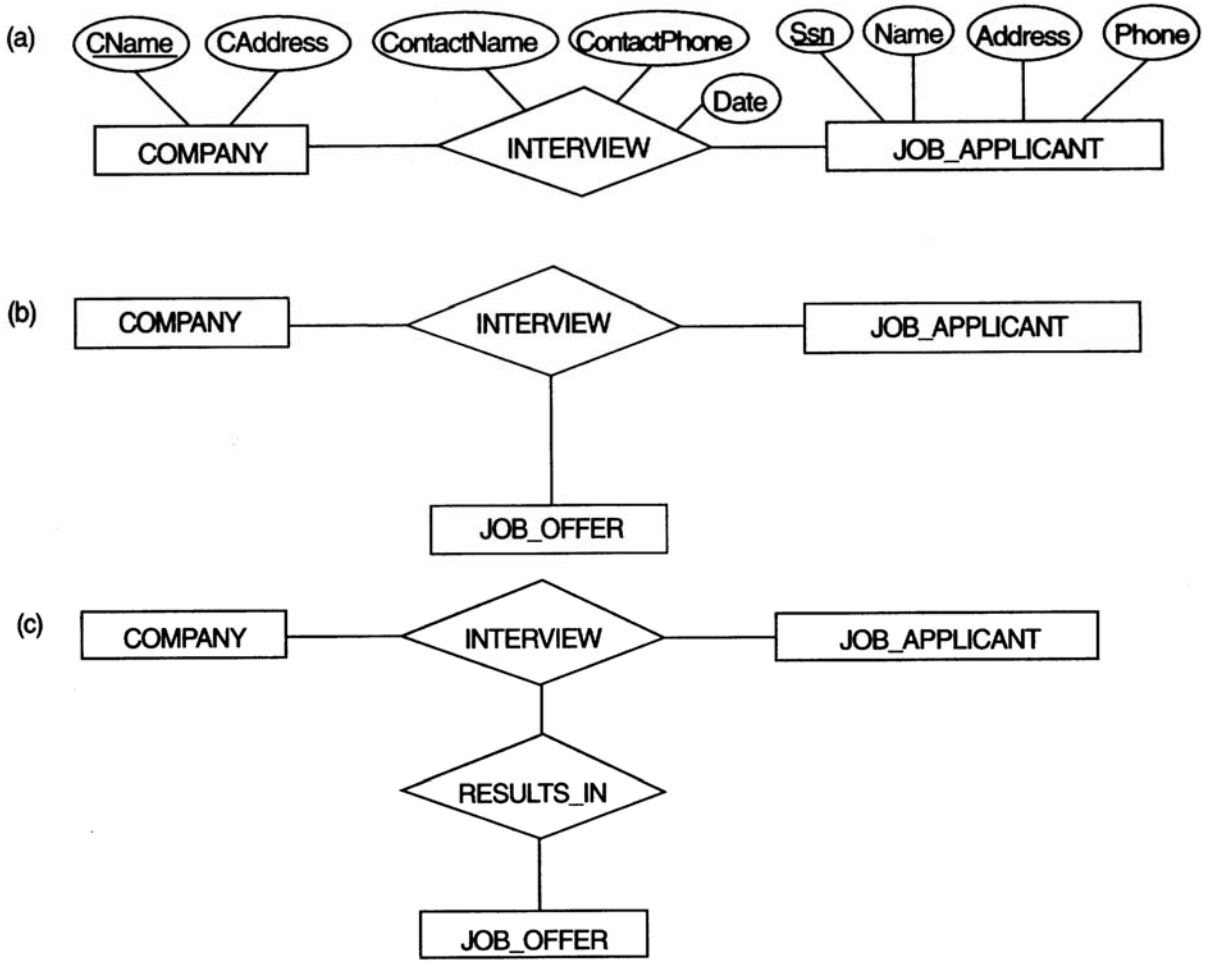
- classification and instantiation
  - classification: systematically assigning similar objects to object classes
  - instantiation: generation of distinct objects of a class (IS-AN-INSTANCE-OF relationship)
- identification
  - abstraction process whereby classes and objects are made uniquely identifiable by means of some **identifier**
- specialization and generalization
  - specialization: the process of classifying a class of objects into more specialized subclasses
  - generalization: generalizing several classes into a higher level abstract class that includes the objects in all these classes (IS-A relationship)
- aggregation and association
  - **aggregation**: building composite objects from their component objects (IS-A-PART-OF)
  - **association**: association of objects from several independent classes (IS-ASSOCIATED-WITH)





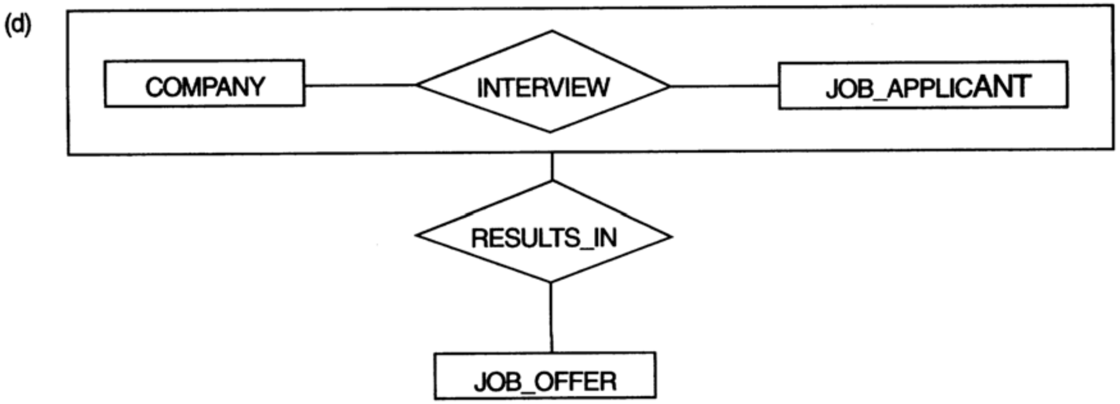
concept of aggregation and association

- suppose that some interviews result in job offers, whereas others do not

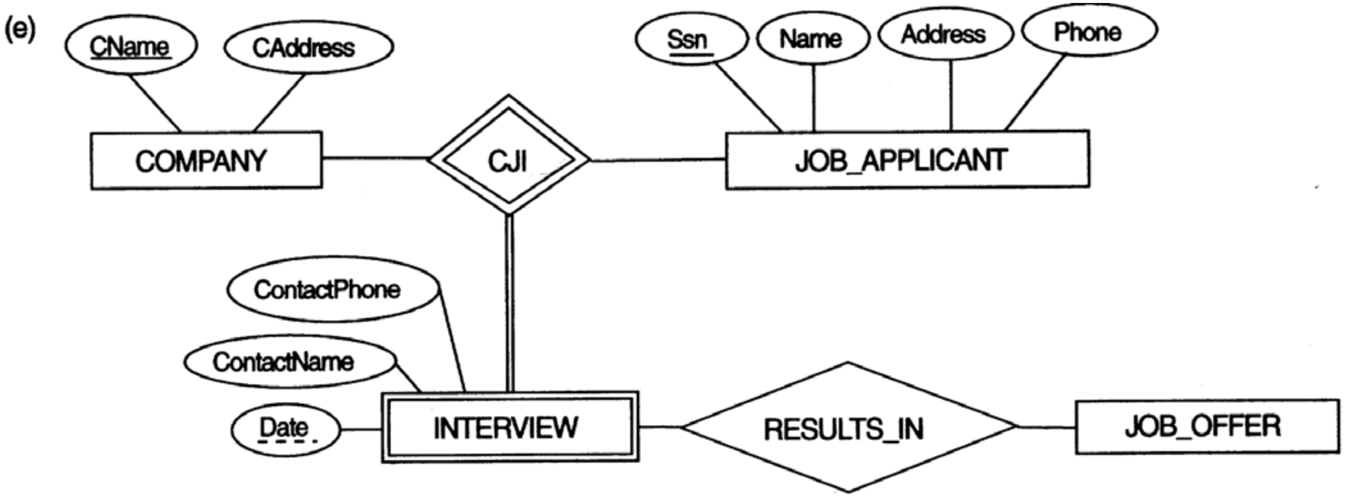


concept of aggregation and association (cont.)

- composite object (not allowed in ER)



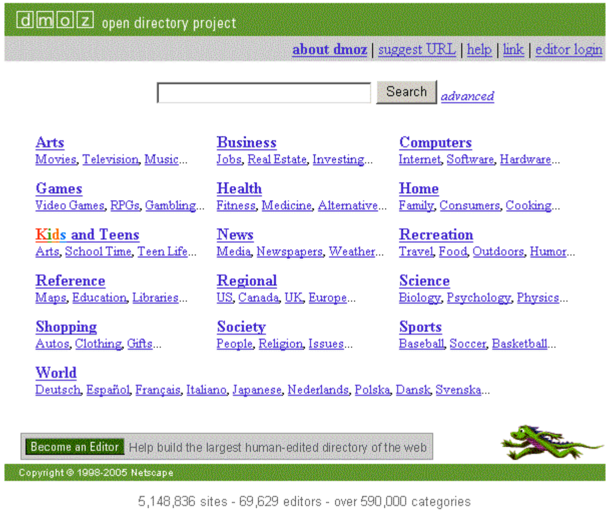
- creation of a new **weak entity type** INTERVIEW



semantic web and ontology

- semantic web
  - to allow **information exchange among computers** on the web
  - attempts to create knowledge representation models that are quite general in order to allow meaningful information exchange and **search among machines**
- ontology
  - a specification of a **conceptualization**
  - attempts to describe the **structures** and **relationships** that are possible in reality through some common vocabulary, and so it can be considered as a way to **describe the knowledge** of a certain community about reality
  - possible techniques: thesaurus, dictionary, taxonomy, database schema, logical theory, …

W3C® World Wide Web consortium



dmoz open directory project

about dmoz | suggest URL | help | link | editor login

Search advanced

Arts
Movies, Television, Music...

Business
Jobs, Real Estate, Investing...

Computers
Internet, Software, Hardware...

Games
Video Games, RPGs, Gambling...

Health
Fitness, Medicine, Alternative...

Home
Family, Consumers, Cooking...

Kids and Teens
Arts, School Time, Teen Life...

News
Media, Newspapers, Weather...

Recreation
Travel, Food, Outdoors, Humor...

Reference
Maps, Education, Libraries...

Regional
US, Canada, UK, Europe...

Science
Biology, Psychology, Physics...

Shopping
Autos, Clothing, Gifts...

Society
People, Religion, Issues...

Sports
Baseball, Soccer, Basketball...

World
Deutsch, Español, Français, Italiano, Japanese, Nederlands, Polska, Dansk, Svenska...

Become an Editor Help build the largest human-edited directory of the web

Copyright © 1998-2005 Netscape

5,148,836 sites - 69,629 editors - over 590,000 categories