Intro to DB

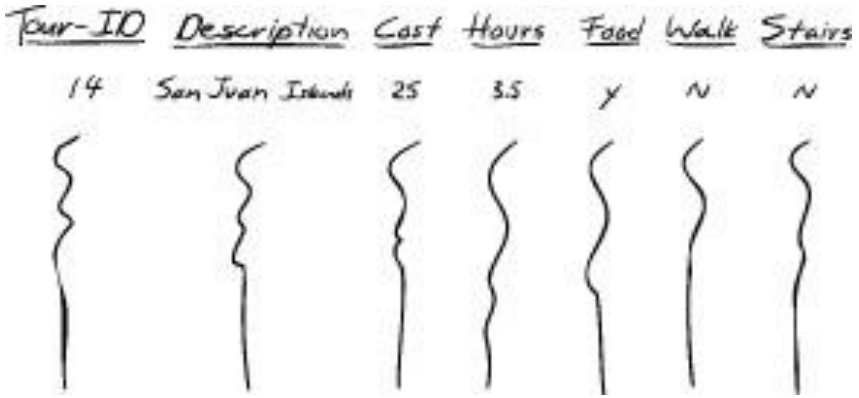
CHAPTER 7
DATABASE DESIGN
& THE E-R MODEL

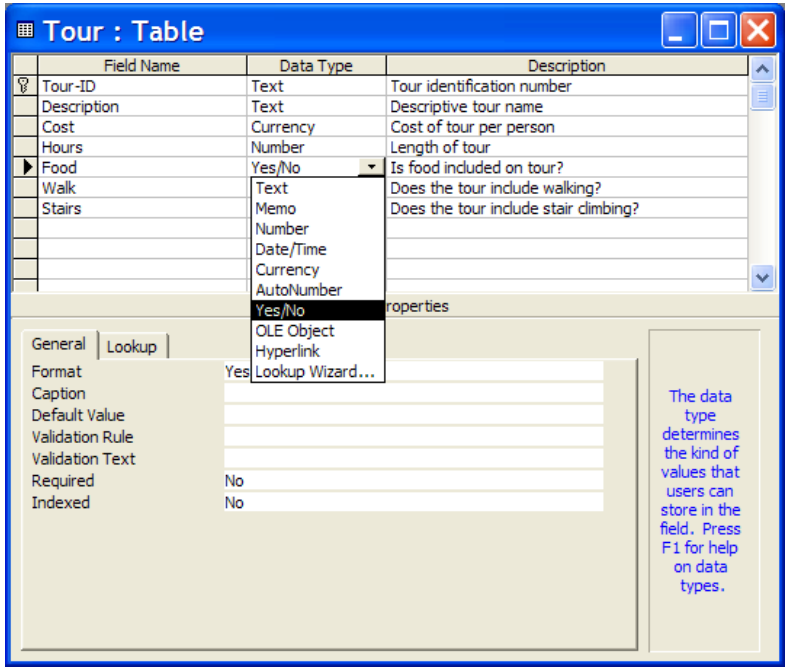
Chapter 7. DB Design & the E-R Model

- Design Process
- Entity-Relationship Model
- Constraints
- Removing Redundant Attributes
- E-R Diagram
- Extended E-R Features
- Reduction to Relation Schemas
- Design Issues
- Alternative Notations for Modeling Data
- Other Aspects of Database Design

Creating a Database

-
  - Reports you will need
  - Inquiries you will want to make
- Sketch the table structure – what kind of data is needed in each column
- Determine characteristics of field
  - Field name: Each field must have a unique field name
  - Field type & length
    - Character, numeric, date, …
-
  - Define each field in the table
  - Define primary key





Entering the Data

- Enter data into the tables in datasheet view
- Enter data into the tables by using a graphical form

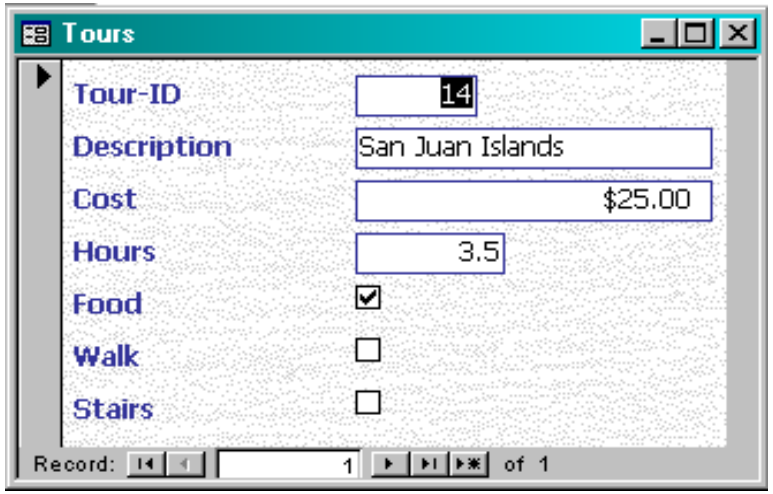
Microsoft Access - [Tour : Table]

File Edit View Insert Format Records Tools Window Help

Tour-ID	Description	Cost	Hours	Food	Walk	Stairs
14	San Juan Islands	$25.00	3.5	☑	☐	☐
16	Local ferry ride	$2.50	1	☐	☐	☐
23	Boeing plant	$0.00	2.4	☐	☑	☑
26	Museum tour	$15.50	0	☐	☑	☐
34	Cityscape bus tour	$24.00	2.5	☑	☐	☐
35	Chinatown at night	$30.00	3	☑	☑	☐
36	Namedroppers tour	$25.00	3	☐	☑	☐
47	Northwest Trek	$12.50	4	☑	☐	☑
58	Mount Ranier	$22.00	5	☑	☑	☐
79	Seattle Locks	$0.00	2	☐	☑	☐
81	Underground tour	$5.50	1.5	☐	☑	☐
84	Puget Sound boat	$42.00	4.5	☑	☐	☐
*		$0.00	0	▩	▩	▩

Record: 1 of 12

Tour identification number NUM



[Capron & Johnson, 2003]

Exercise: Part-Project-Supplier DB

"Parts are supplied to projects. There can be multiple suppliers for a part. Each part has part-id, part-name, and size. You must keep track of the project managers and contact information for each project."

Your Design

Design 1

Supplier	*Part-ID*	*Part-Name*	*Size*	*Proj-ID*	*Manager*	*Contact*	*Quantity*
Sammi	N125	Nut	1.25”	P002	K. Kang	222-1234	1320
Sammi	N100	Nut	1.00”	P003	S. Choi	333-2345	1000
Sammi	B332	Bolt	12”	P002	K. Kang	222-1234	1320
Sammi	B332	Bolt	12”	P003	S. Choi	333-2345	1000
ABC	N125	Nut	1.25”	P004	R. Smith	444-3456	500
ABC	B332	Bolt	12”	P004	R. Smith	444-3456	500
…							

Design 1.2

Supplier	*S-contact*	*Part-ID*	*Part-Name*	*Size*	*Proj-ID*	*Location*	*Manager*	*P-contact*	*Quantity*
Sammi	555-5555	N125	Nut	1.25”	P002	Seocho	K. Kang	222-1234	1320
Sammi	555-5555	N100	Nut	1.00”	P003	Mapo	S. Choi	333-2345	1000
Sammi	555-5555	B332	Bolt	12”	P002	Seocho	K. Kang	222-1234	1320
Sammi	555-5555	B332	Bolt	12”	P003	Mapo	S. Choi	333-2345	1000
ABC	777-7777	N125	Nut	1.25”	P004	Jamsil	R. Smith	444-3456	500
ABC	777-7777	B332	Bolt	12”	P004	Jamsil	R. Smith	444-3456	500
…									

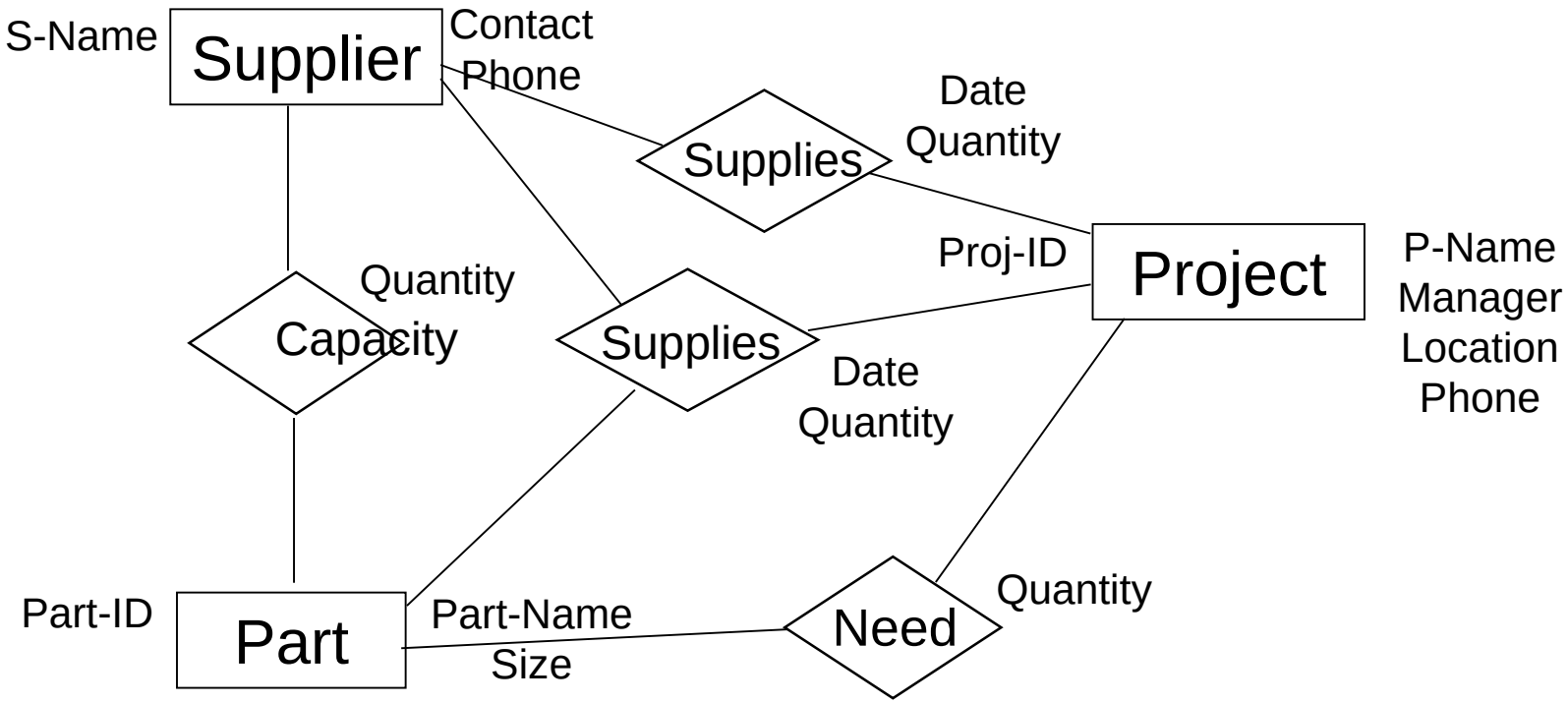
Design 2

Supplier	*S-contact*
Sammi	555-5555
ABC	777-7777
…	

Part-ID	*Part-Name*	*Size*
N125	Nut	1.25”
N100	Nut	1.00”
B332	Bolt	12”
…		

Proj-ID	*Location*	*Manager*	*P-contact*
P002	Seocho	K. Kang	222-1234
P003	Mapo	S. Choi	333-2345
P004	Jamsil	R. Smith	444-3456
…			

Design 3 <ER Diagram>



Design 3 <Relations>

Supplier

S-Name	Contact	Phone
...		

Capacity

S-Name	Part-ID	Quantity
...		

Part

Part-ID	Part-Name	Size
...		

Supplies

Proj-ID	Part-ID	S-Name	Date	Quantity
...				

Project

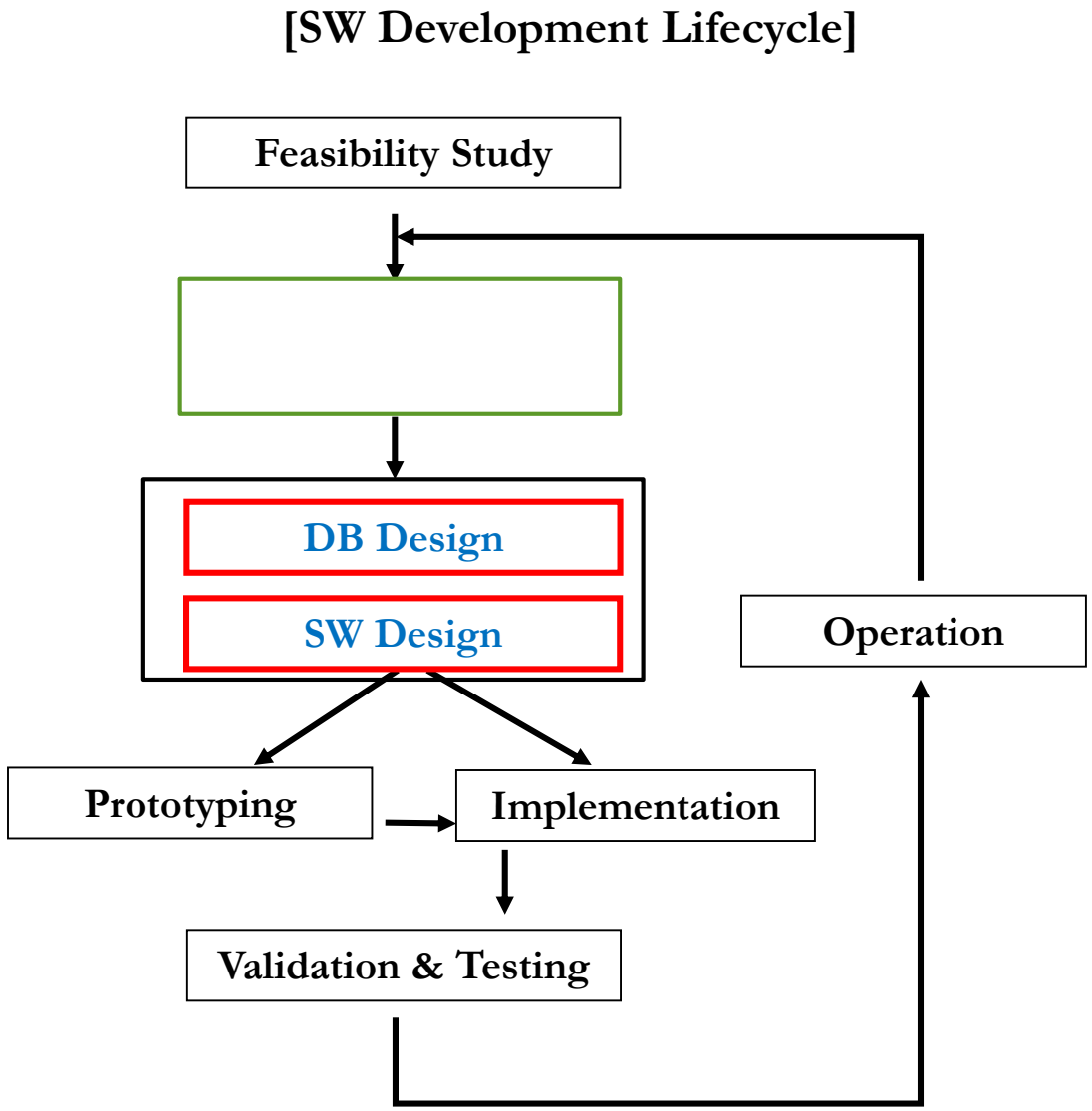
Proj-ID	Proj-Name	Location	Manager	P-contact
...				

Needs

Proj-ID	Part-ID	Quantity
...		

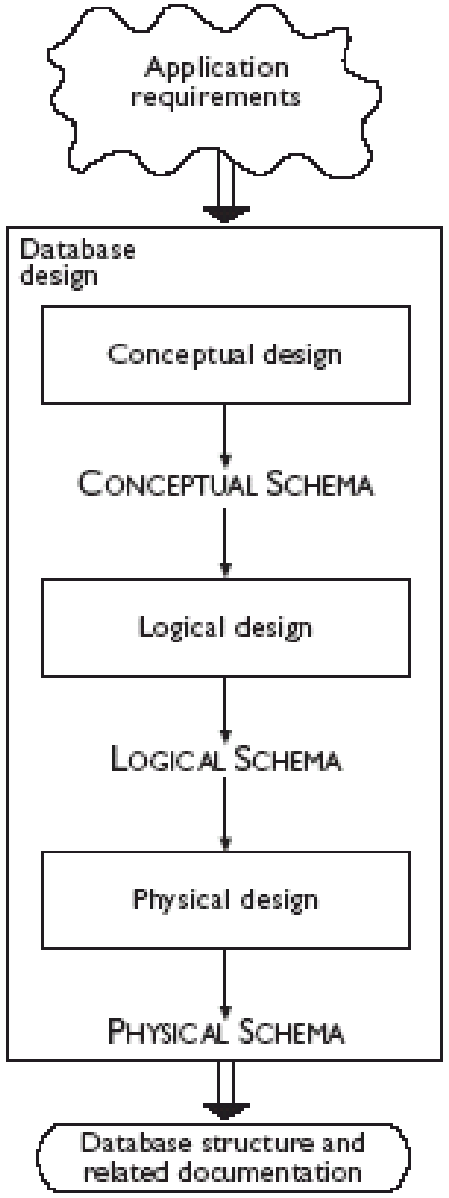
Database Design

- Decide on the DB schema that
  - is able to hold all information in consideration,
  - with minimal (or no) redundancy, and
  - allows for effective & efficient data operations
- Critical in reducing operations and maintenance costs of SW systems



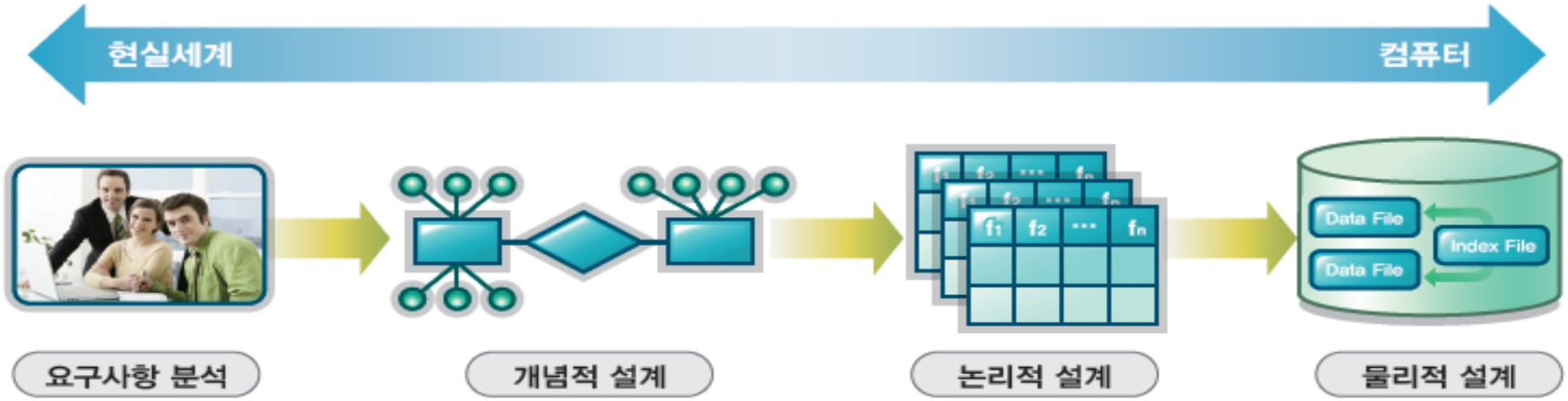
Phases of Database Design

- Three phases
- *Conceptual design*
  - Construction of an ER schema
  -
- *Logical design*
  - map onto the implementation data model of the DBMS (such as RDB)
- *Physical design*
  - specify physical features of the database (issues pertaining to performance rather than information contents; index, sequential order, etc.)



[Atzeni, et al, 2000]

Phases of Database Design



[이상구 외, 2012]

Entity-Relationship Model

- Proposed by P. Chen in 1976

  “The Entity-Relationship Model: Toward a Unified View of Data”, *ACM Transactions On Database Systems*, Jan.1976.

- A very powerful tool in the design of databases
  - Simple model
  - 
- E-R model is not an implementation model
  - i.e., there is no DBMS whose internal structures are based on the E-R model

Database Modeling

- A *database* can be modeled as:
  -
  -
- ***Entity***
  - an object that exists and is distinguishable from other objects
  - entity instance

    ex. specific person, company, event, plant
- ***Attributes***
  - Entities have attributes
  - ex. people have *names* and *addresses*
- ***Entity set***
  - a set of entities of the same type that share the same properties
  - ex. set of all persons, companies, trees, holidays

Entity & Entity Sets - examples

instructor_ID	instructor_name
76766	Crick
45565	Katz
10101	Srinivasan
98345	Kim
76543	Singh
22222	Einstein

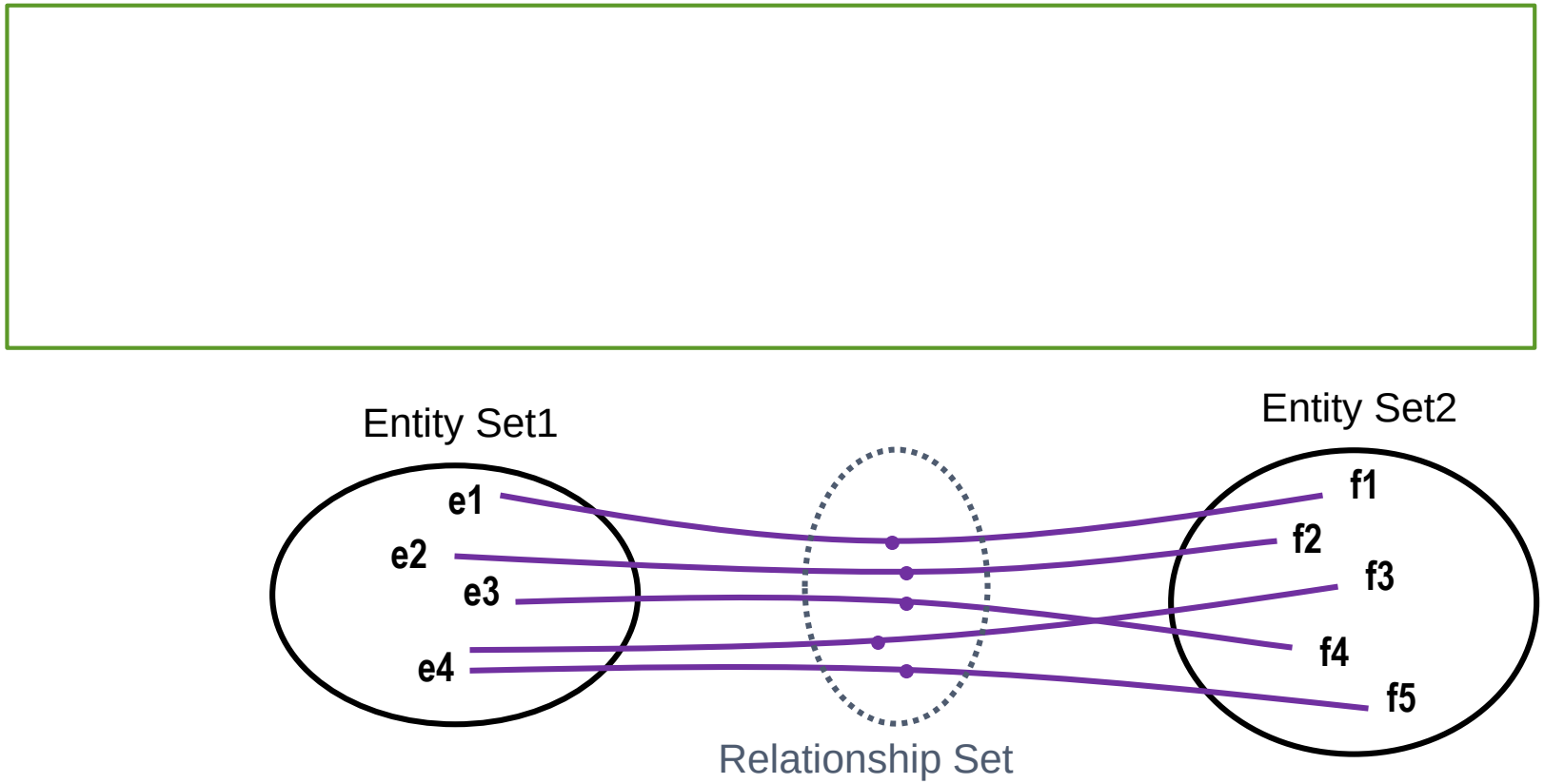
instructor

student-ID	student_name
98988	Tanaka
12345	Shankar
00128	Zhang
76543	Brown
76653	Aoi
23121	Chavez
44553	Peltier

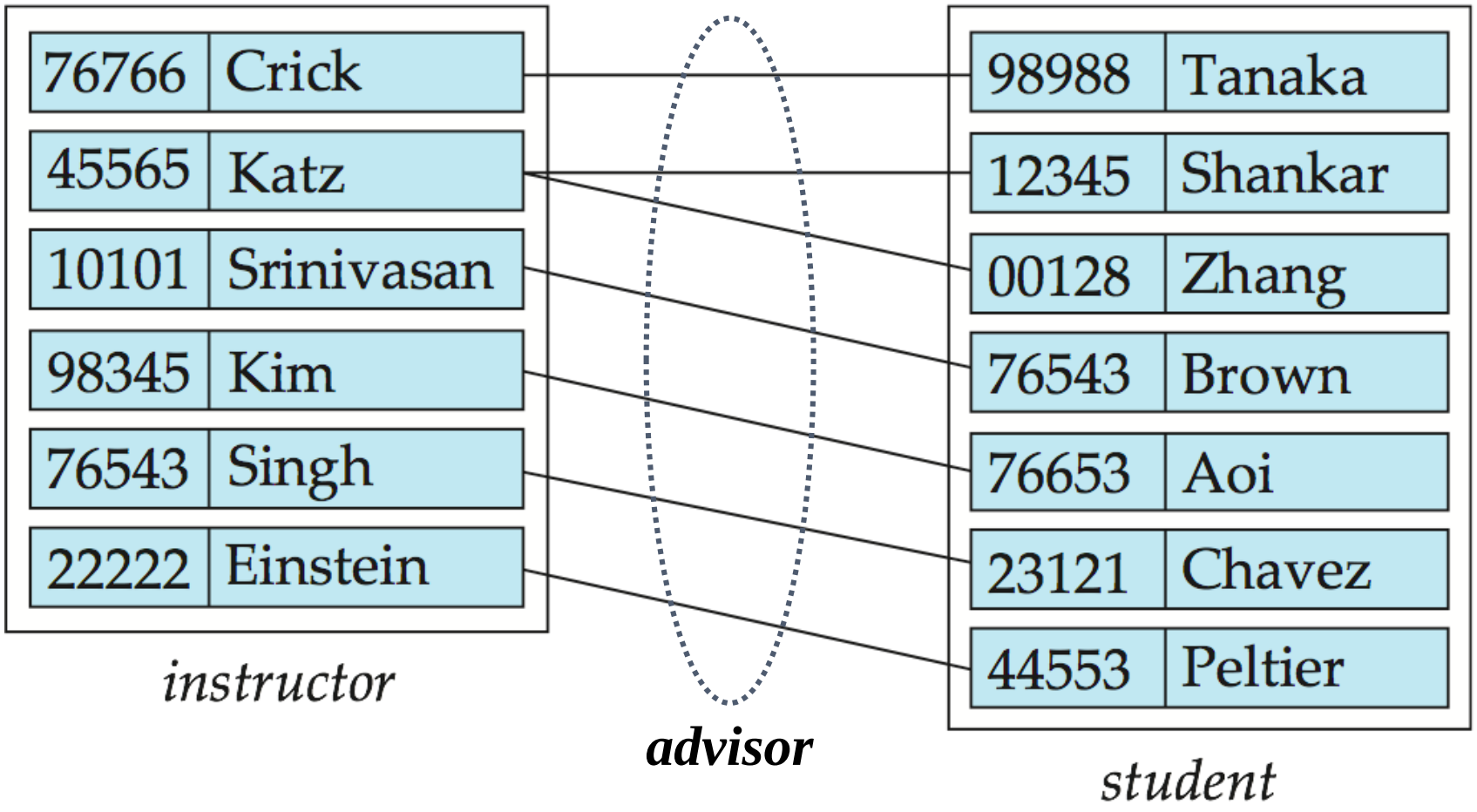
student

Relationships

- Relationships are defined between entities
- Relationship set:

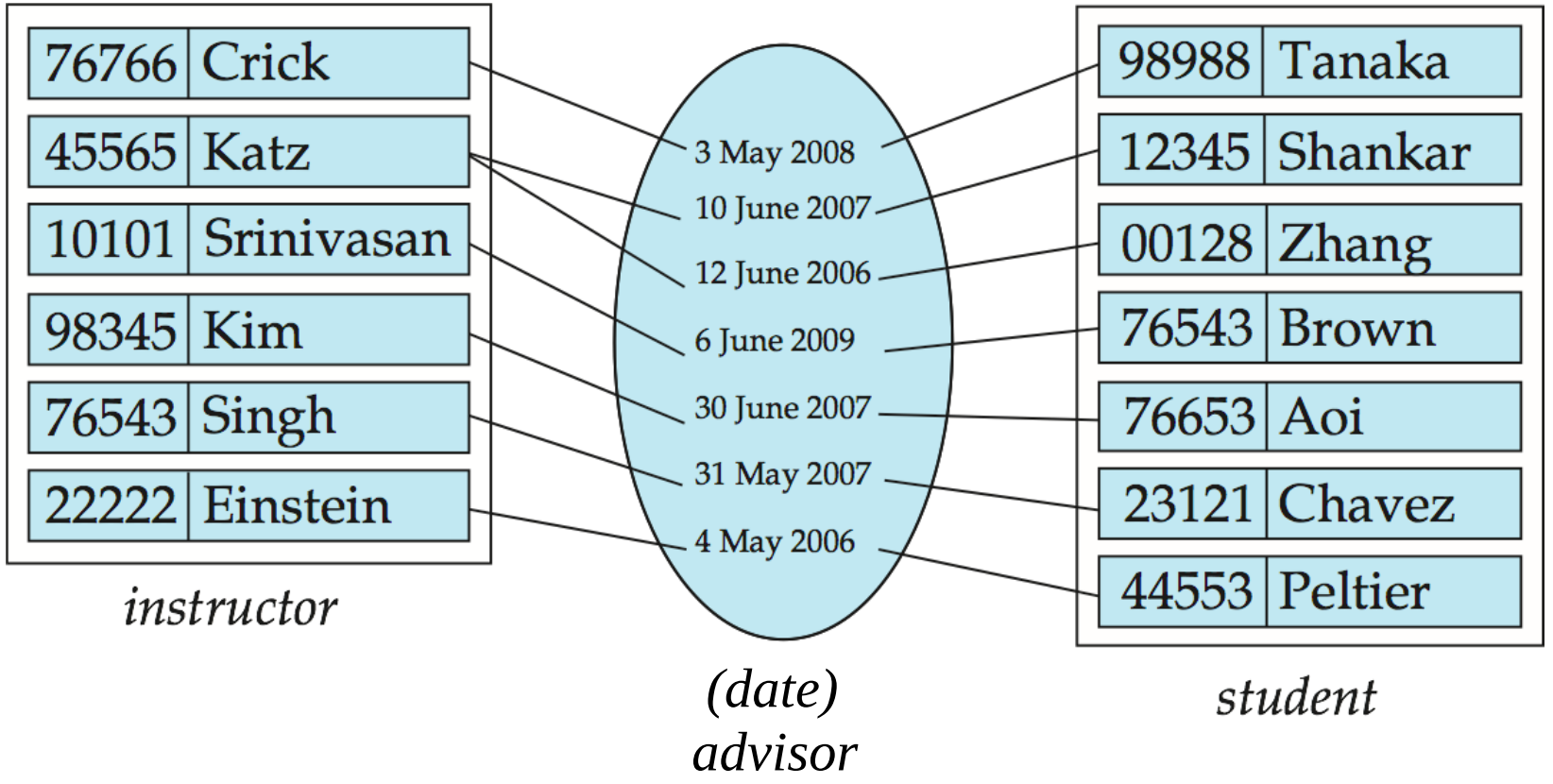


Relationship Set *advisor*

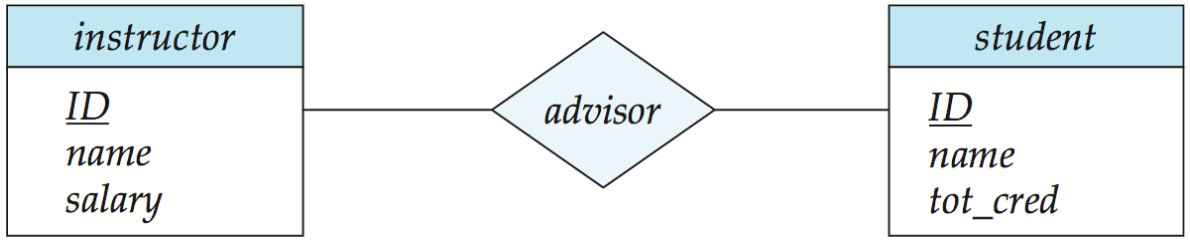


Attributes of Relationships

- Relationships can have attributes

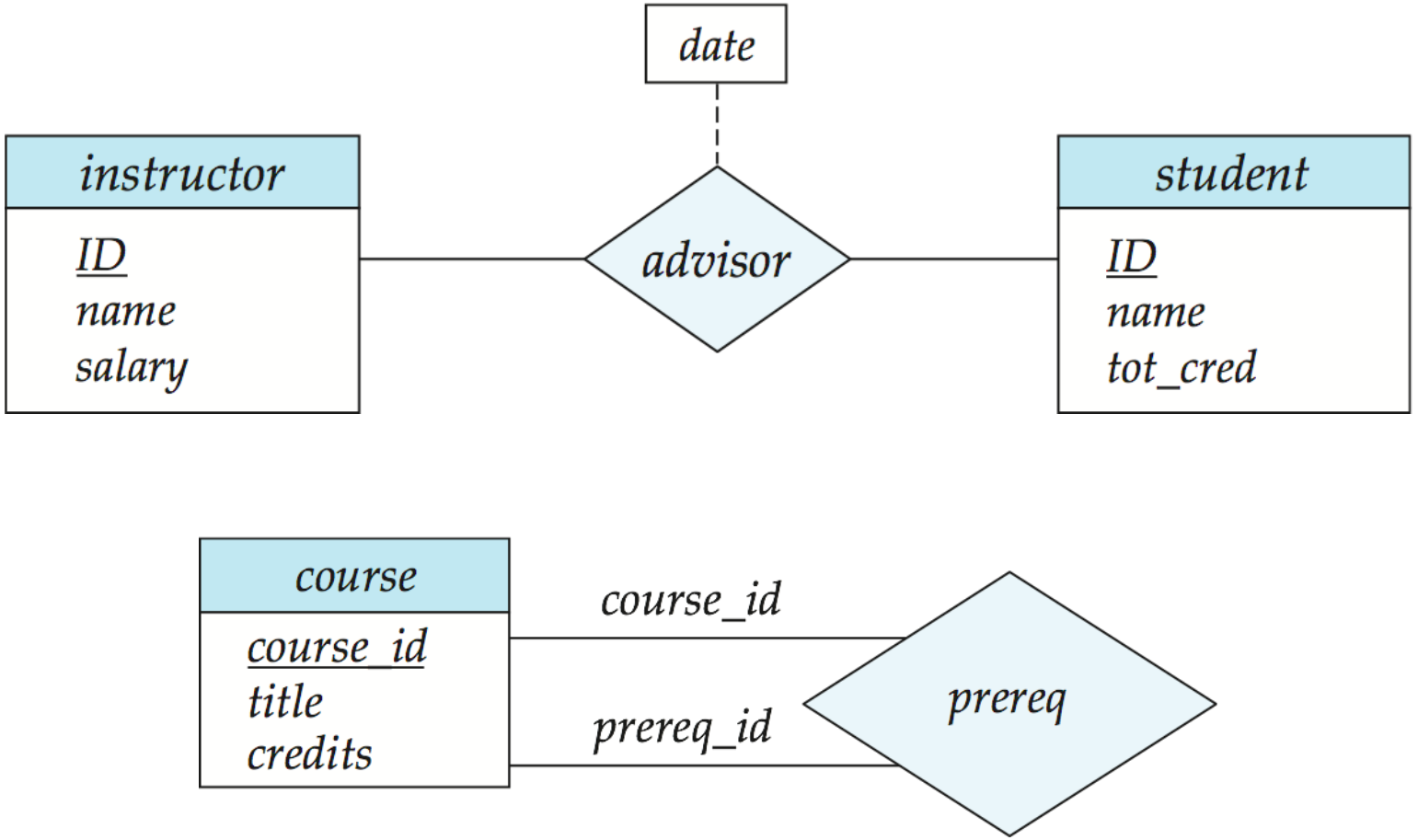


E-R Diagrams

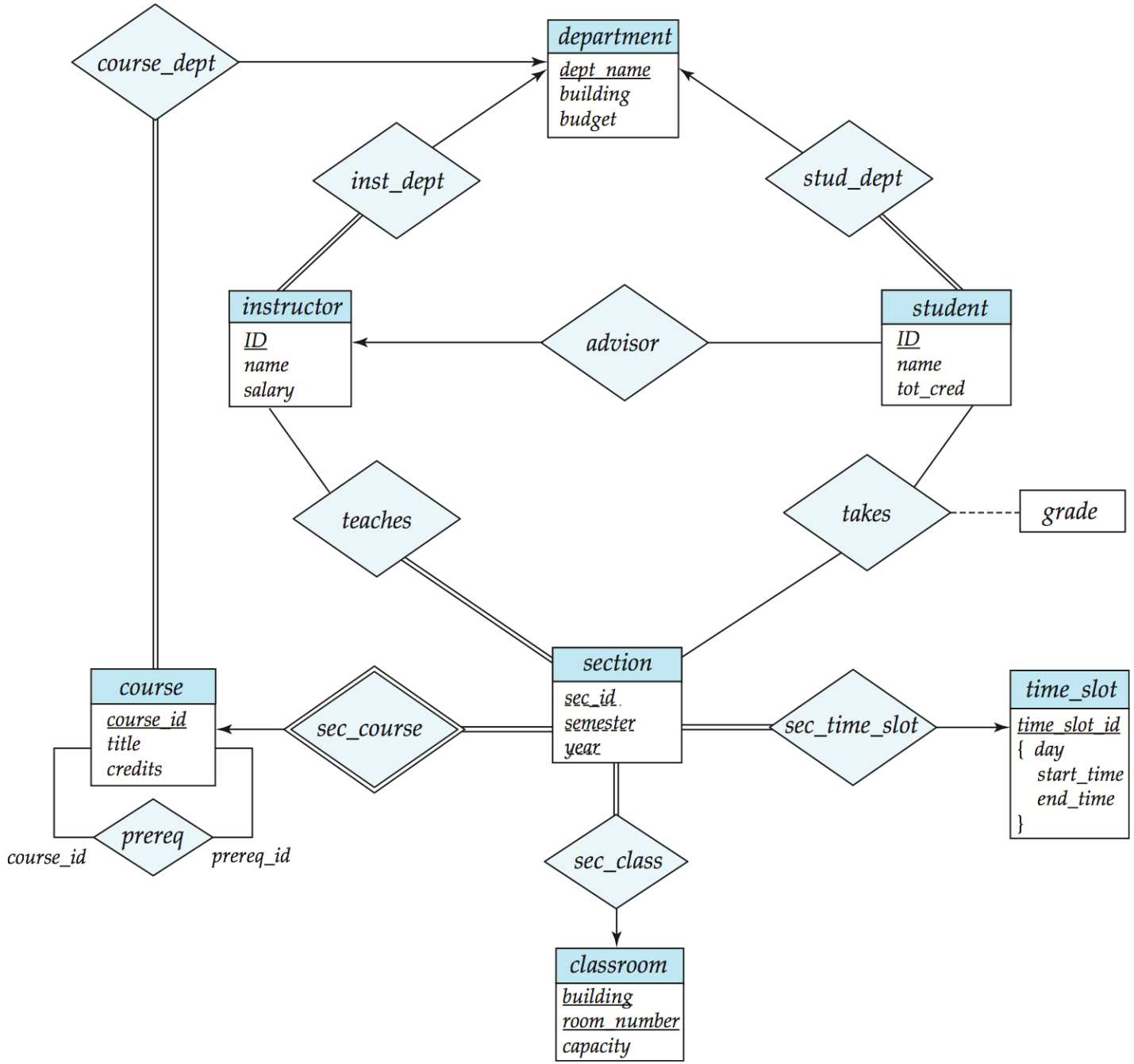


- **Rectangles** represent entity sets.
- **Diamonds** represent relationship sets.
- Attributes listed inside entity rectangle
- **Underline** indicates primary key attributes

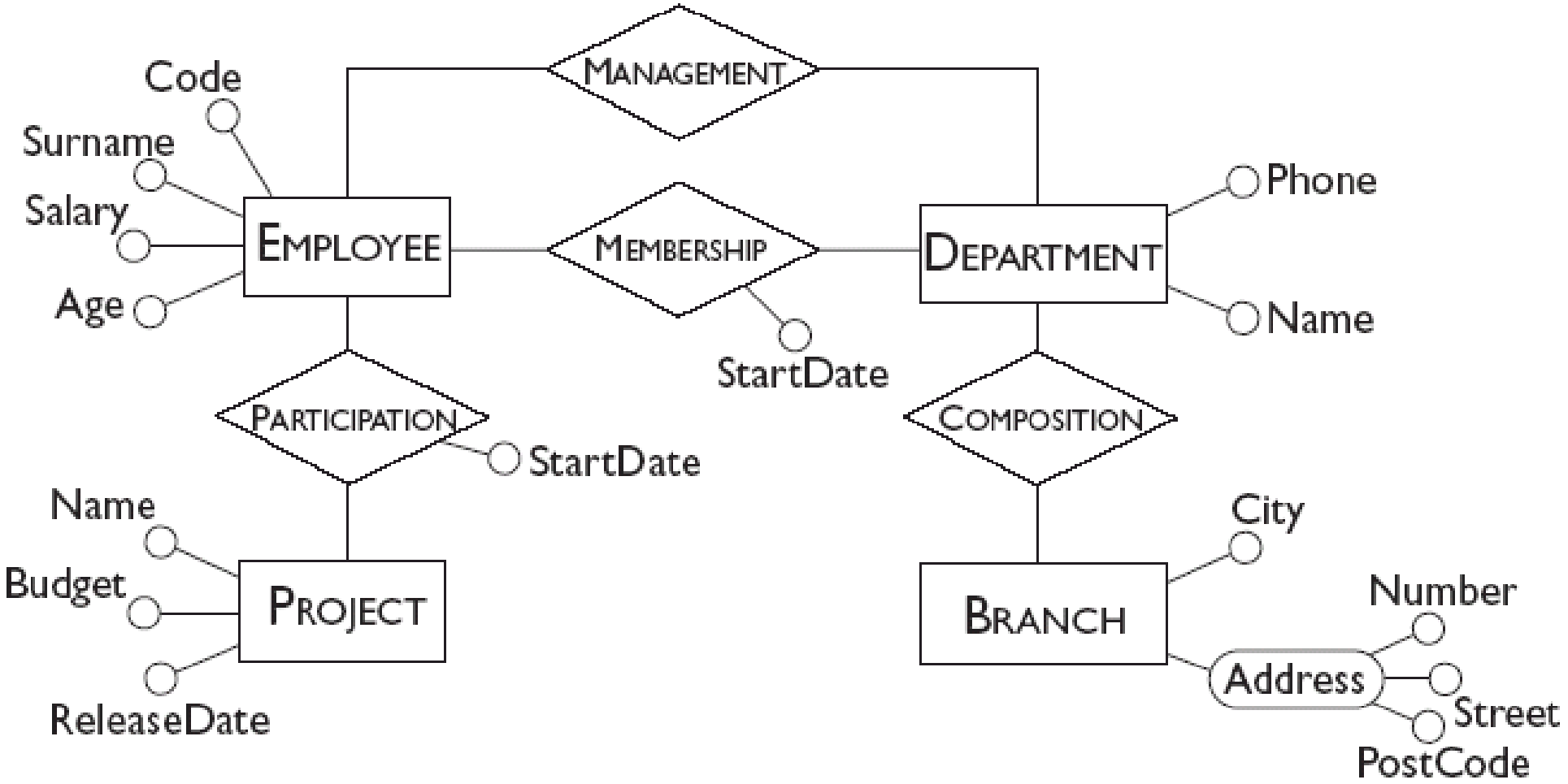
Relationship Sets: Attributes & Roles



An ER Schema


course_dept
department
dept_name
building
budget
inst_dept
stud_dept
instructor
ID
name
salary
advisor
student
ID
name
tot_cred
teaches
takes
grade
section
sec_id
semester
year
course
course_id
title
credits
sec_course
sec_time_slot
time_slot
time_slot_id
{ day
start_time
end_time
}
prereq
course_id
prereq_id
sec_class
classroom
building
room_number
capacity

An ER Schema



[From Atzeni, et al, *Database Systems: Concepts, Languages and Architectures*, 2000]

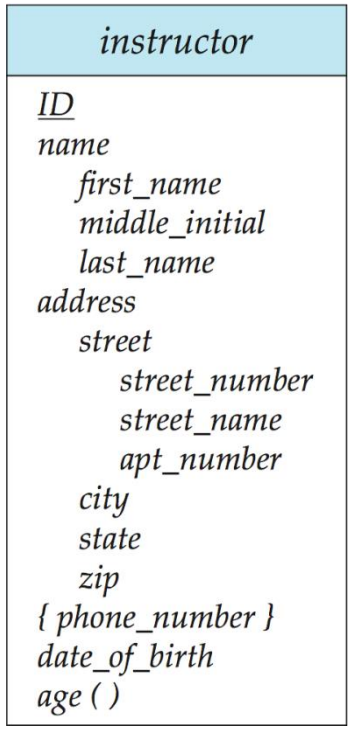
Types of Attributes

- Simple vs Composite attributes
  - Simple attribute
    -
    -
  - Composite attribute
    - composed of multiple parts
    - *name = (lastnm, firstnm)*
    - *phone# = (number, extension)*

- Null
  - null value: a special value meaning “missing” or “unknown”
  - some attributes are not allowed to have null values

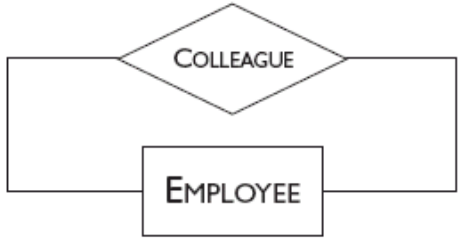
Types of Attributes (cont.)

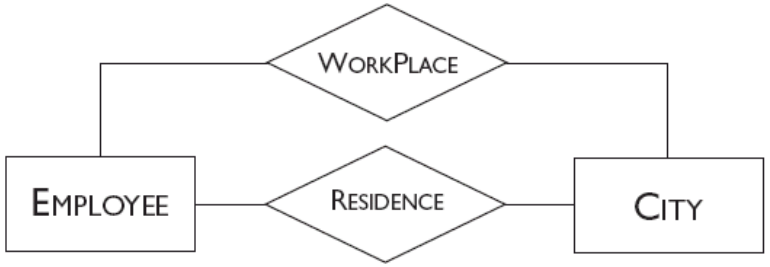
- Single-valued vs multivalued :
  - Single-valued attribute
    - each attribute has a single value for an entity
    - *id, name, dept*
  - Multivalued attribute
    - an attribute may have more than one value for an instance
    - *children = {john, tom}, phone#={5567, 5568}*
- Derived attributes
  -
  -

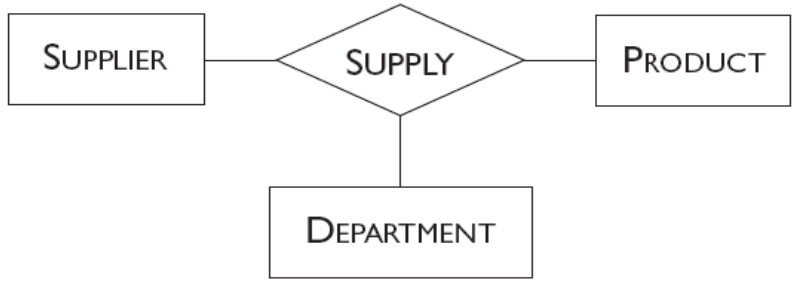


Degree of a Relationship Set

- Most relationships are binary
  - involve two entity sets (or degree two)
  - $R = \{ [e_1, e_2] \mid e_1 \in E_1, e_2 \in E_2 \}$
- You can define non-binary relationships
  - $R = \{ [e_1, e_2, e_3] \mid e_1 \in E_1, e_2 \in E_2, e_3 \in E_3 \}$: ternary



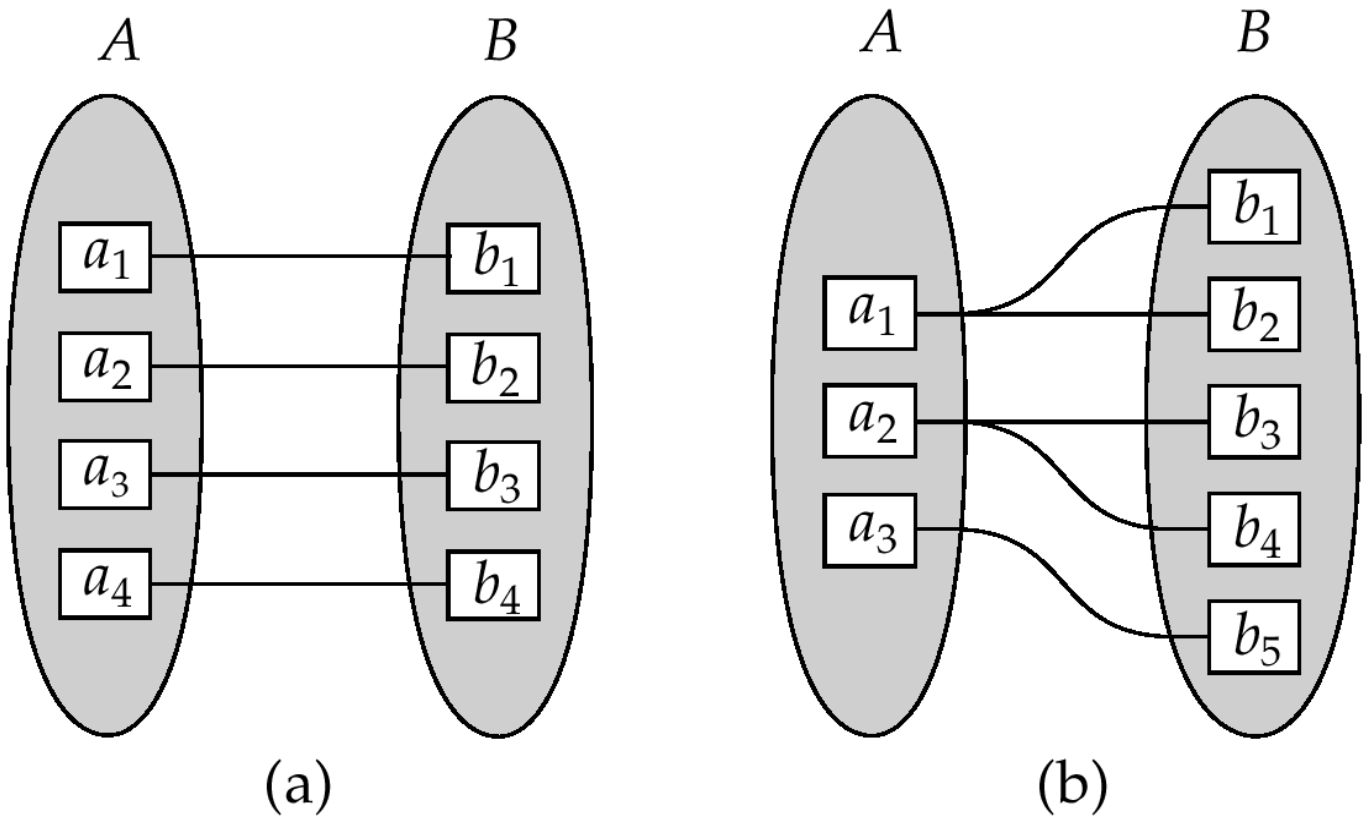




Mapping Constraints

- Relationship *cardinality*
  - 
- Generic types
  - 1 : 1
  - 1 : *m*
  - *m* : 1
  - *m* : *n*

Mapping Cardinalities

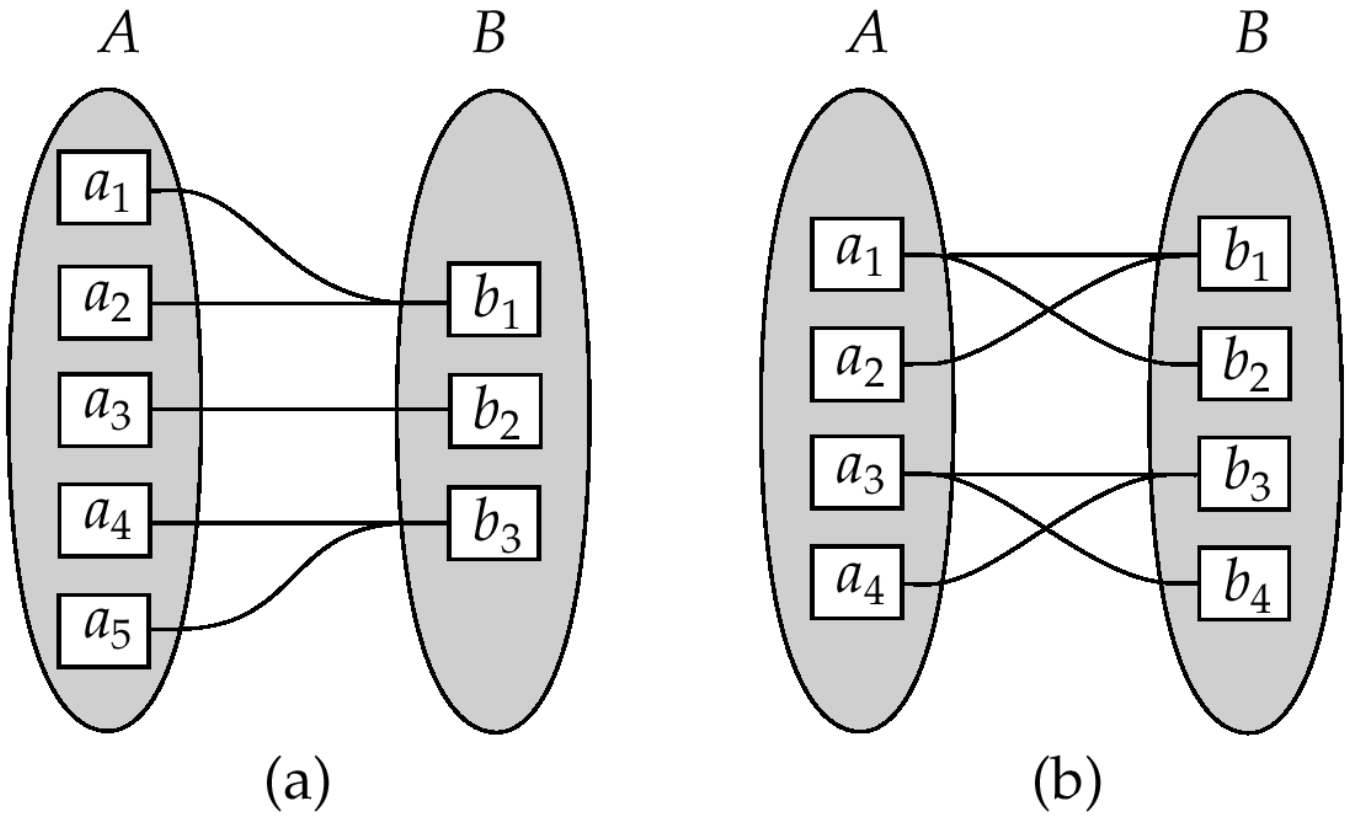


One to one

One to many

Note: Some elements in A and B may not be mapped to any elements in the other set

Mapping Cardinalities (cont.)

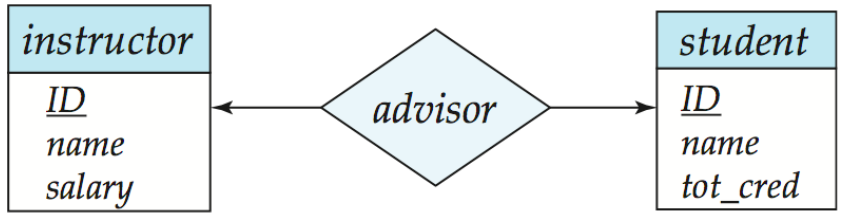


Many to one

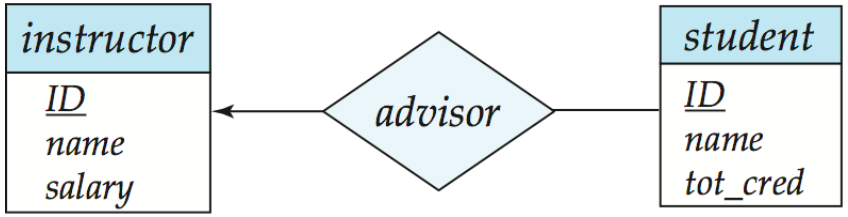
Many to many

Note: Some elements in A and B may not be mapped to any elements in the other set

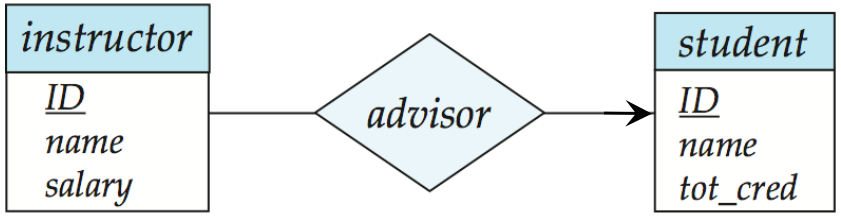
Mapping (Cardinality) Constraints



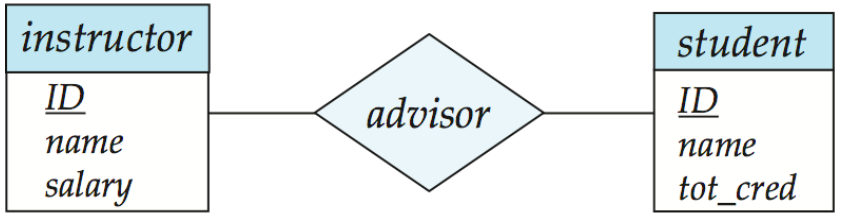
one-to-one



one-to-many



many-to-one

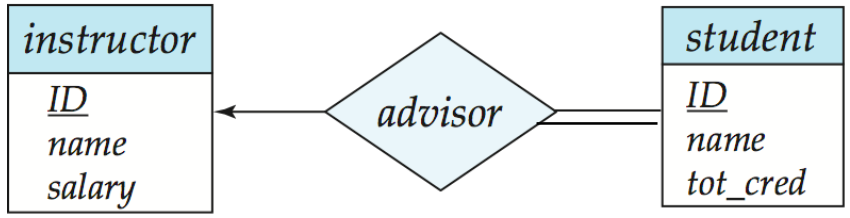


many-to-many

Participation of an Entity Set in a Relationship Set

- Total participation (double line)
  - every entity in the entity set participates in at least one relationship in the relationship set
  - eg.: every *section* must have an associated course

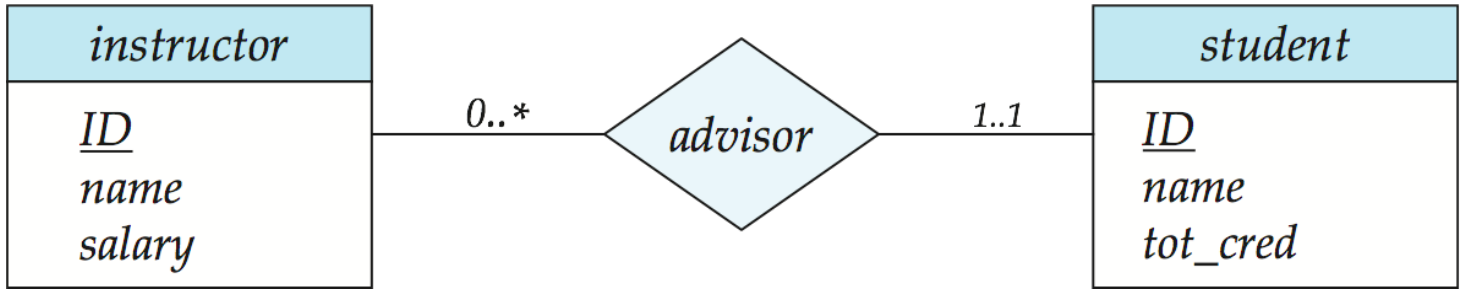
- Partial participation
  - some entities may not participate in any relationship in the relationship set
  - eg.: participation of *instructor* in *advisor* is partial



Alternative Notation for Cardinality & Participation

- Express cardinality by upper and lower limits
  - *min … max*

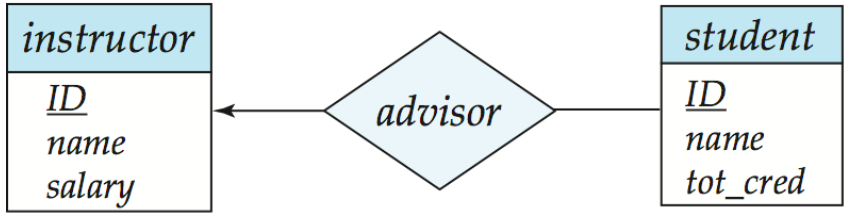
- Can also express participation constraints



Keys

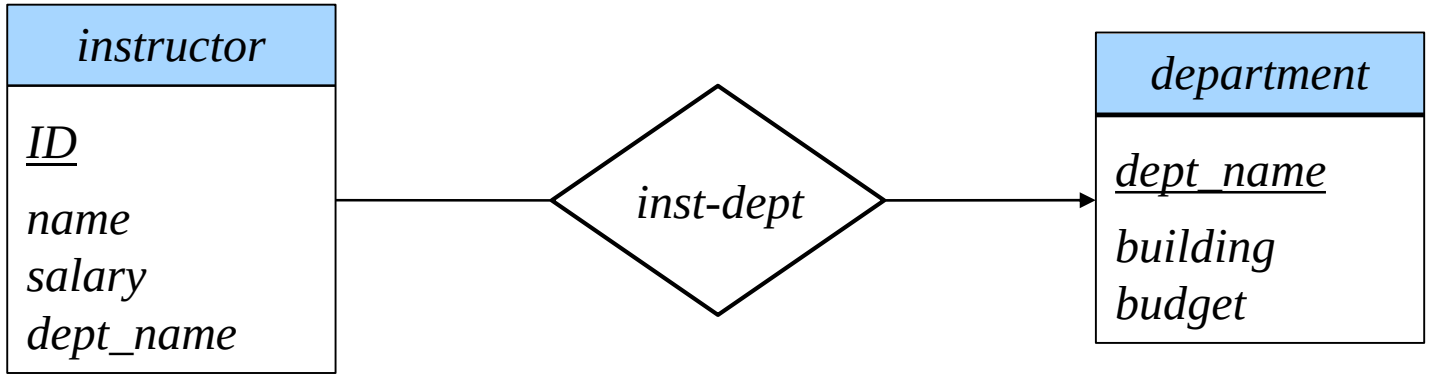
- Key for an Entity (Set)
  - Same as keys in relational models: super key, candidate key, primary key
  - Set of attributes whose values can distinguish entities from each other

- Key for a Relationship (Set)
  - combination of primary keys of the participating entity sets forms a super key
  - Must consider the mapping cardinality of the relationship set when deciding what are the candidate keys (and primary key)



Redundant Attributes

- Suppose we have entity sets

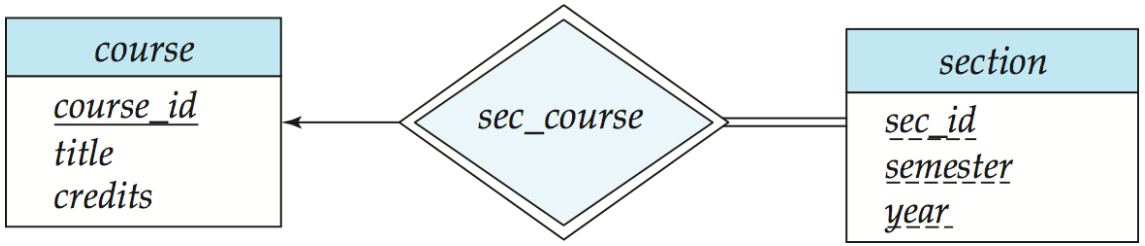


- *dept_name* in *instructor* is redundant

  since there is an explicit relationship which relates instructors to departments
  - The attribute replicates information present in the relationship, and should be removed from *instructor*
  -

Weak Entity Sets

- *Weak Entity Set*: An entity set that does not have sufficient attributes to form a primary key
  - ⇔ *strong entity set*

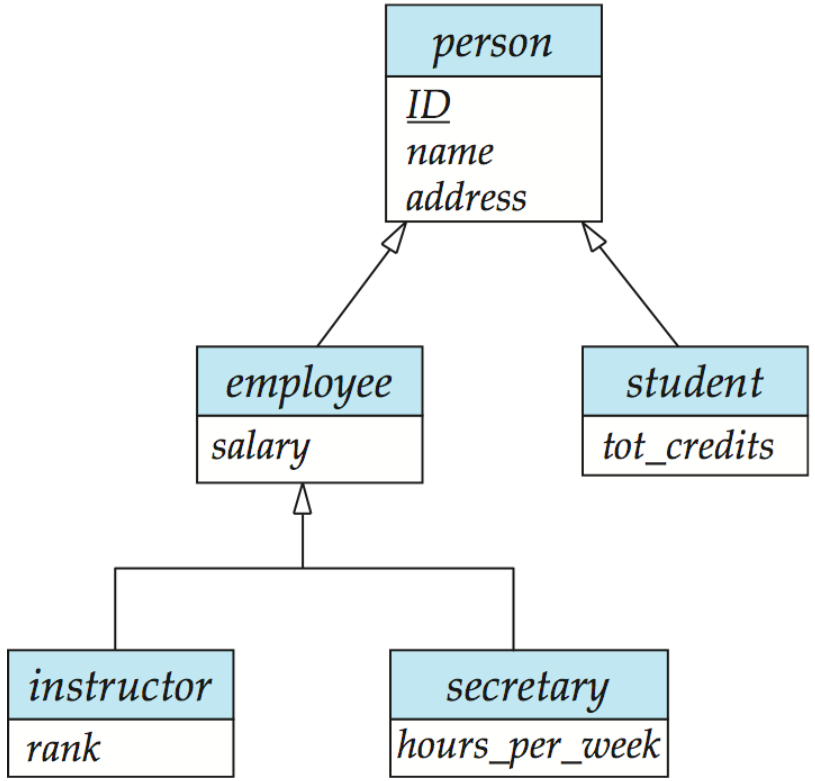


- The existence of a weak entity set depends on the existence of a **identifying entity set**
  - **Identifying relationship** depicted using a double diamond
- The **discriminator** (*or partial key)* of a weak entity set: set of attributes that distinguishes among all the weak entities related to the same strong entity
- *primary-key*(weak entity set)

  = *primary_key*(identifying strong entity) U *discriminator*(weak entity set)

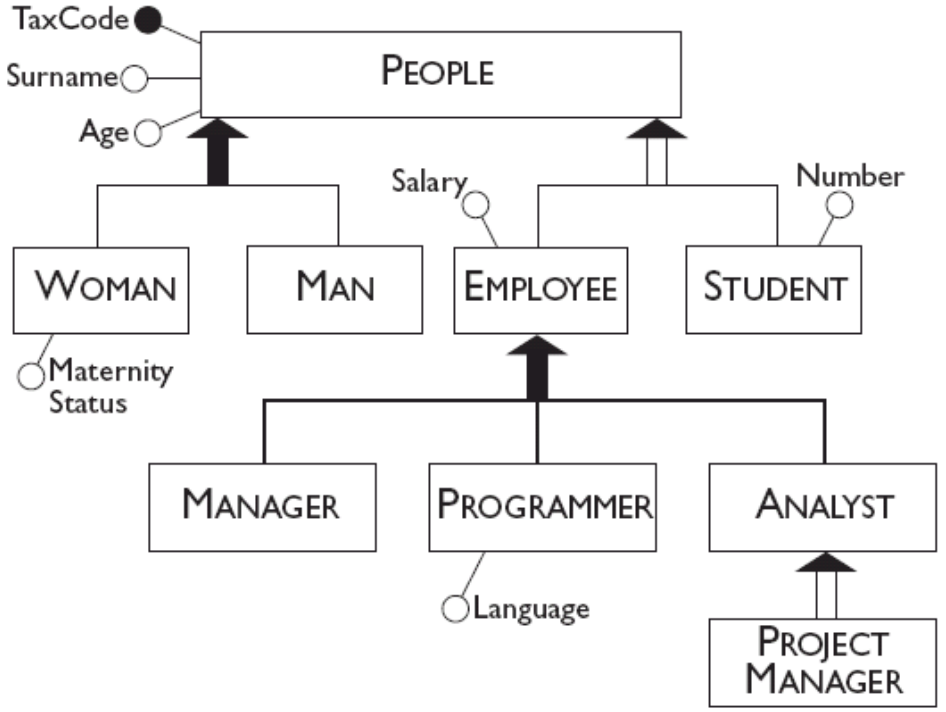
Extended E-R Features

-
  - subgroupings within an entity set
  - Sub entities share common attributes
  - Each sub entity set may have its own specific attributes
-
  - combine a number of entity sets that share the same features into a higher-level entity set
  - Opposite of specialization - depends on where you start

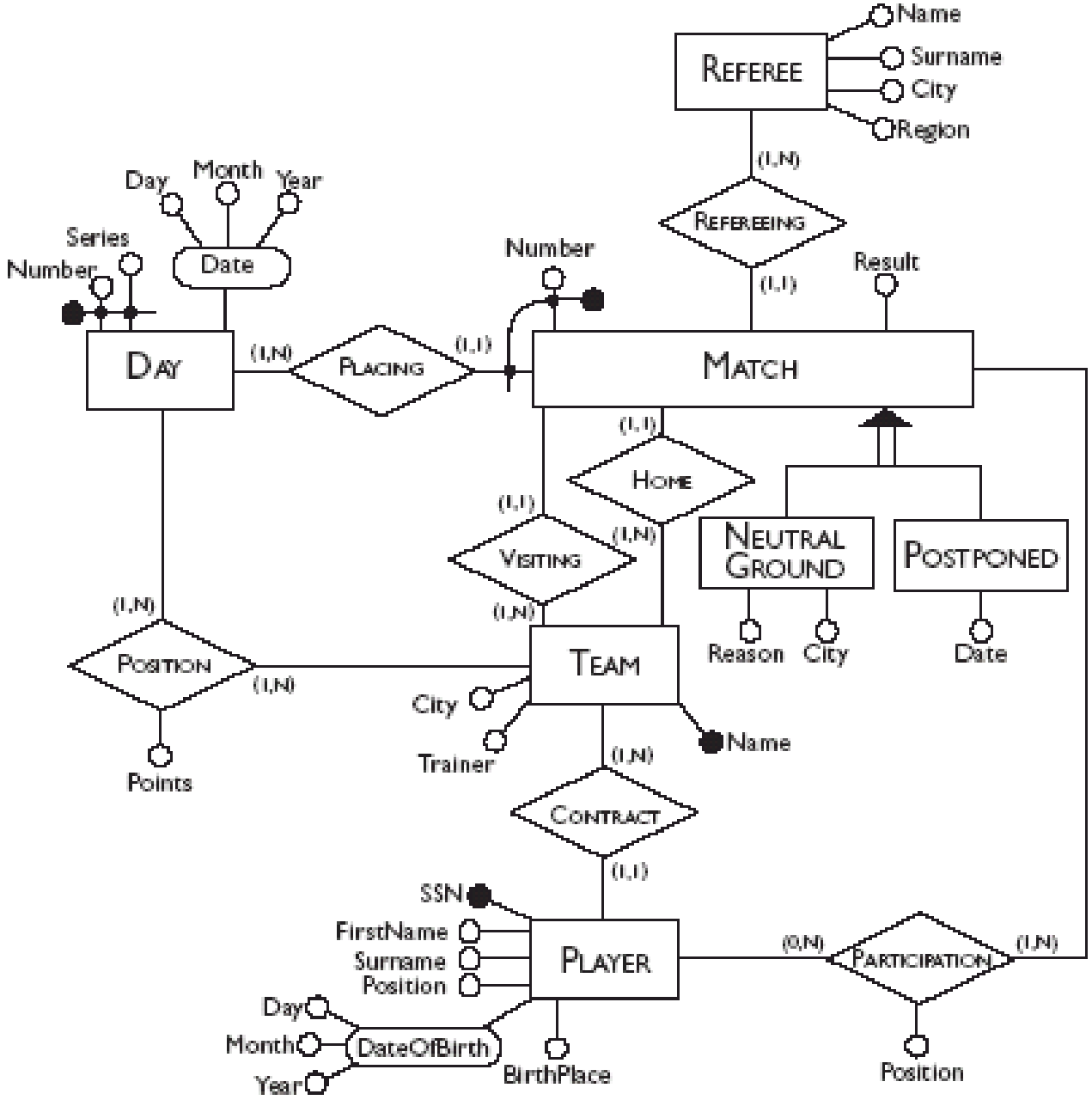


Extended E-R Features (cont.)

- Inheritance
  - The attributes and relationships of the higher-level entity sets are inherited by (applies to) the lower-level entity sets
- Types of generalization (super-sub entities)
  - *disjoint vs overlapping:*
    -
  - *total vs partial:*
    -



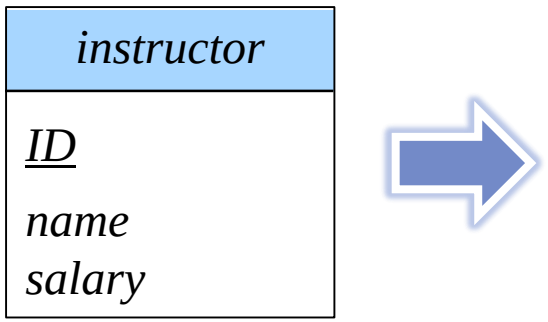
Exercise: Interpret the ER Diagram



[From Atzeni, et al, *Database Systems: Concepts, Languages and Architectures*, 2000]

Reducing ER schema to tables

- *Logical design*
  - map onto the implementation data model of the DBMS
- Basic rule
  -
  -
- Entity set E with attributes $a_1, ..., a_n$
  - table $r(E)$ with schema $E(a_1, ..., a_n)$
  - column a_i has domain D_i (as defined by the entity)
    - $r(E) \subseteq D_1 \text{ X } . . . \text{ X } D_n$



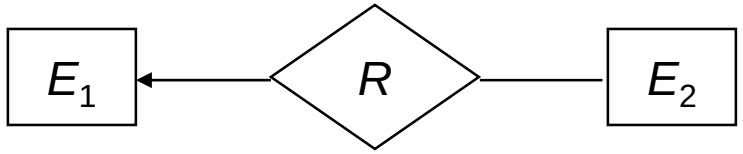
ID	*name*	*salary*
10101	Srinivasar	65000
12121	Wu	90000
15151	Mozart	40000
22222	Einstein	95000
32343	El Said	60000
33456	Gold	87000
45565	Katz	75000
58583	Califieri	62000
76543	Singh	80000
76766	Crick	72000
83821	Brandt	92000
98345	Kim	80000

ER schema to tables (cont.)

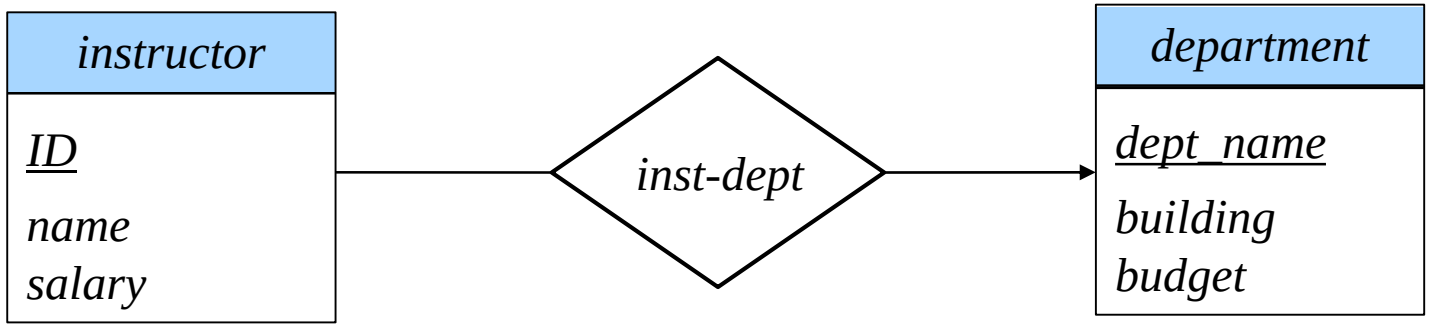
- Relationship set *R*
  - involving entities E_1, ..., E_k
    => table *r* with columns corresponding to

    $$PK(E_1) \cup ... \cup PK(E_k) \cup attr(R)$$
  - *supplies* relating *supplier*, *client*, *part*
  - *attends* relating *student* and *course-offering*

customer-id	*loan-number*
019-28-3746	L-11
019-28-3746	L-23
244-66-8800	L-93
321-12-3123	L-17
335-57-7991	L-16
555-55-5555	L-14
677-89-9011	L-15
963-96-3963	L-17



=> add columns representing $PK(E_1) \cup attr(R)$ to table representing E_2



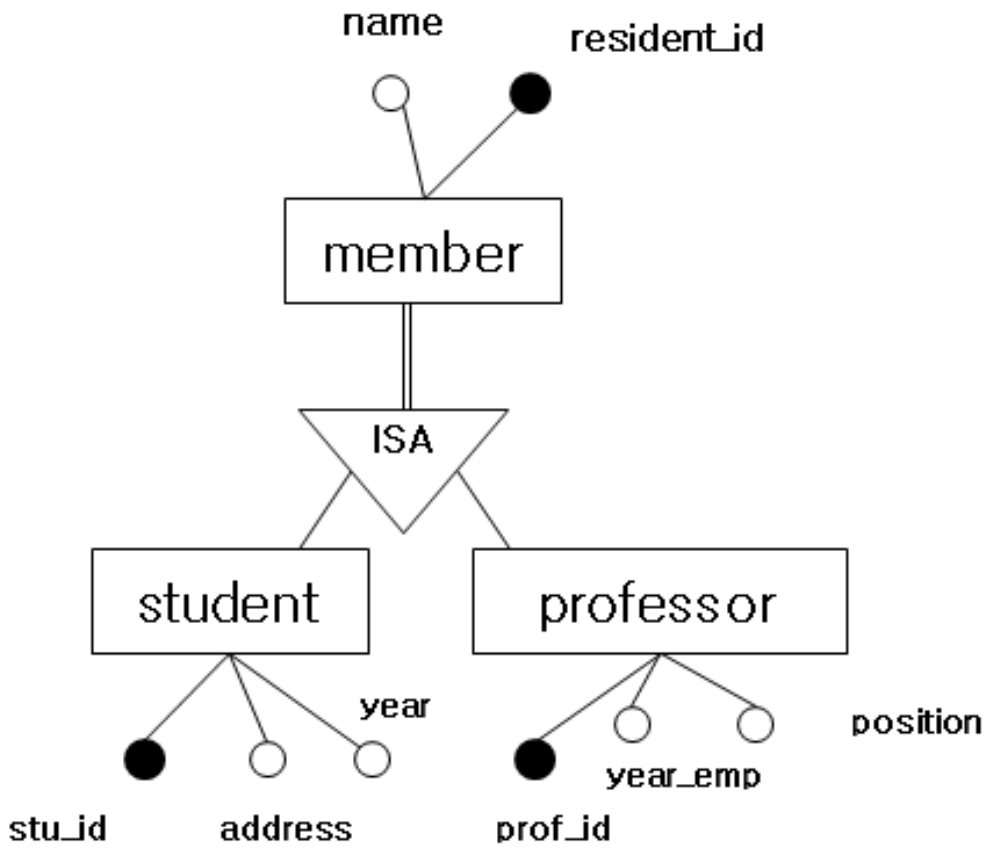
ID	name	salary
10101	Srinivasar	65000
12121	Wu	90000
15151	Mozart	40000
22222	Einstein	95000
32343	El Said	60000
33456	Gold	87000
45565	Katz	75000
58583	Califieri	62000
76543	Singh	80000
76766	Crick	72000
83821	Brandt	92000
98345	Kim	80000

ID	dept_name
10101	Comp. Sci
12121	Finance
15151	Music
…	…

dept_name	building	budget
Biology	Watson	90000
Comp. Sci.	Taylor	100000
Elec. Eng.	Taylor	85000
Finance	Painter	120000
History	Painter	50000
Music	Packard	80000
Physics	Watson	70000

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Generalization/Specialization to Relation Schema



- Option 1: A relation for each entity set
  - ISA relationship translated to *foreign key*
  - Good for *partial* and/or *overlapping* generalizations

member(<u>resident_id</u>, name)
student(<u>stu_id</u>, resident_id, address, year)
professor(<u>prof_id</u>, resident_id, year_emp, position)

- Option 2: Keep only the lower level entity sets
  - Merge attributes of higher level entity set onto each lower level entity set
  - Good for *total* and/or *disjoint* generalizations

student(resident_id, name, stu_id, address, year)
professor(resident_id, name, prof_id, year_emp, position)

Design Issues

-
  - an employee’s telephone
    - as an attribute: simple
    - as an entity: independent
  - decision should be based on
    - whether the telephone must be treated as an independent entity
    - the number of telephones an employee can have
    - whether telephones are shared between employees
-

  "customer having an *account* at a branch"
  - *account* as relationship: simple but limited (cannot participate in other relationships)
  - *account* as entity: account can act as separate entity

Design Issues (cont.)

-
  - all *n*-ary relationships can be represented by binary relationships by adding additional entities and corresponding relationships
  - however, this is not always desirable
-

Decision should be based on how the model

best represents the real world situation

END OF CHAPTER 7