Intro to DB

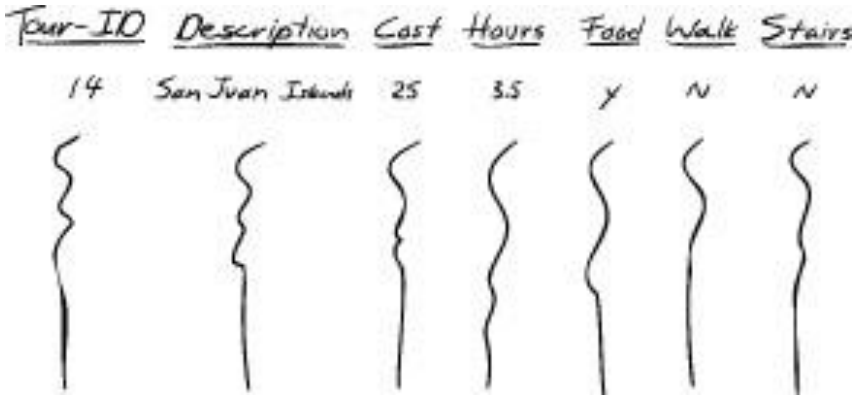
CHAPTER 6
DATABASE DESIGN & THE E-R MODEL

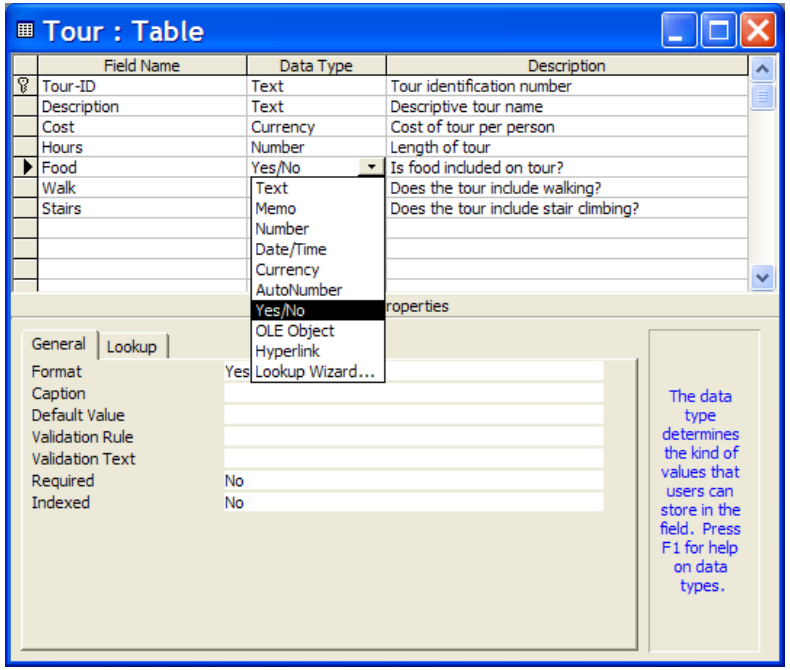
Chapter 6. DB Design & the E-R Model

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

Creating a Database

- Consider your needs
  - 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, …
- Create the table
  - 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

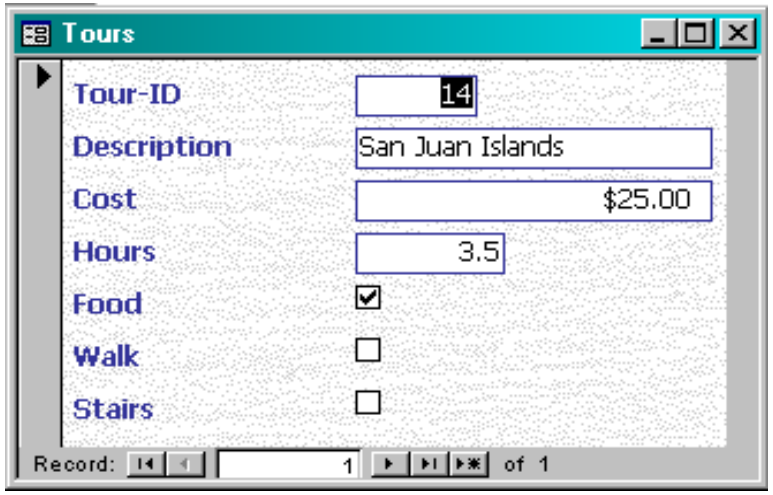
Microsoft Access - [Tour : Table]

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

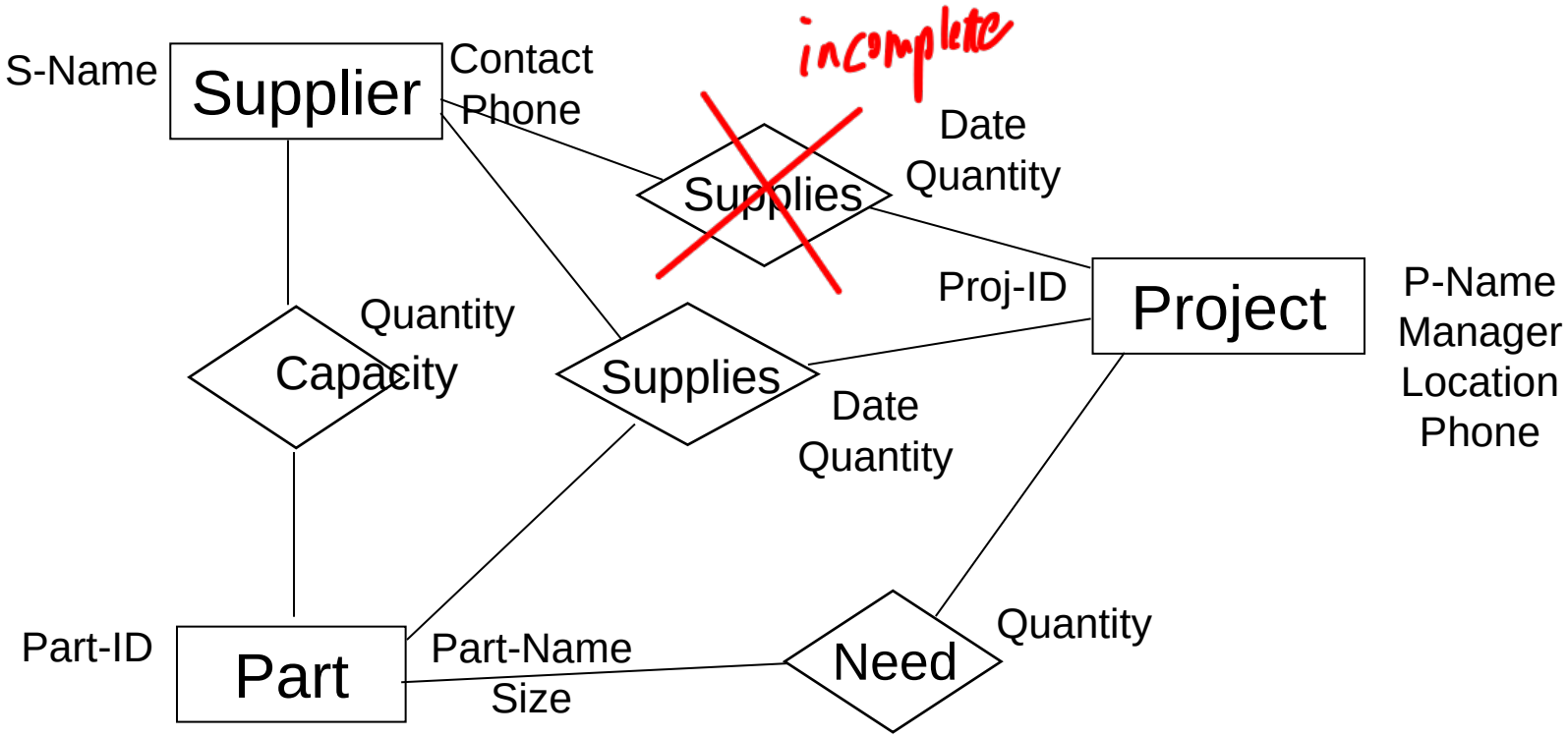
Supplier	*S-contact*
Sammi	555-5555
ABC	777-7777
…	

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

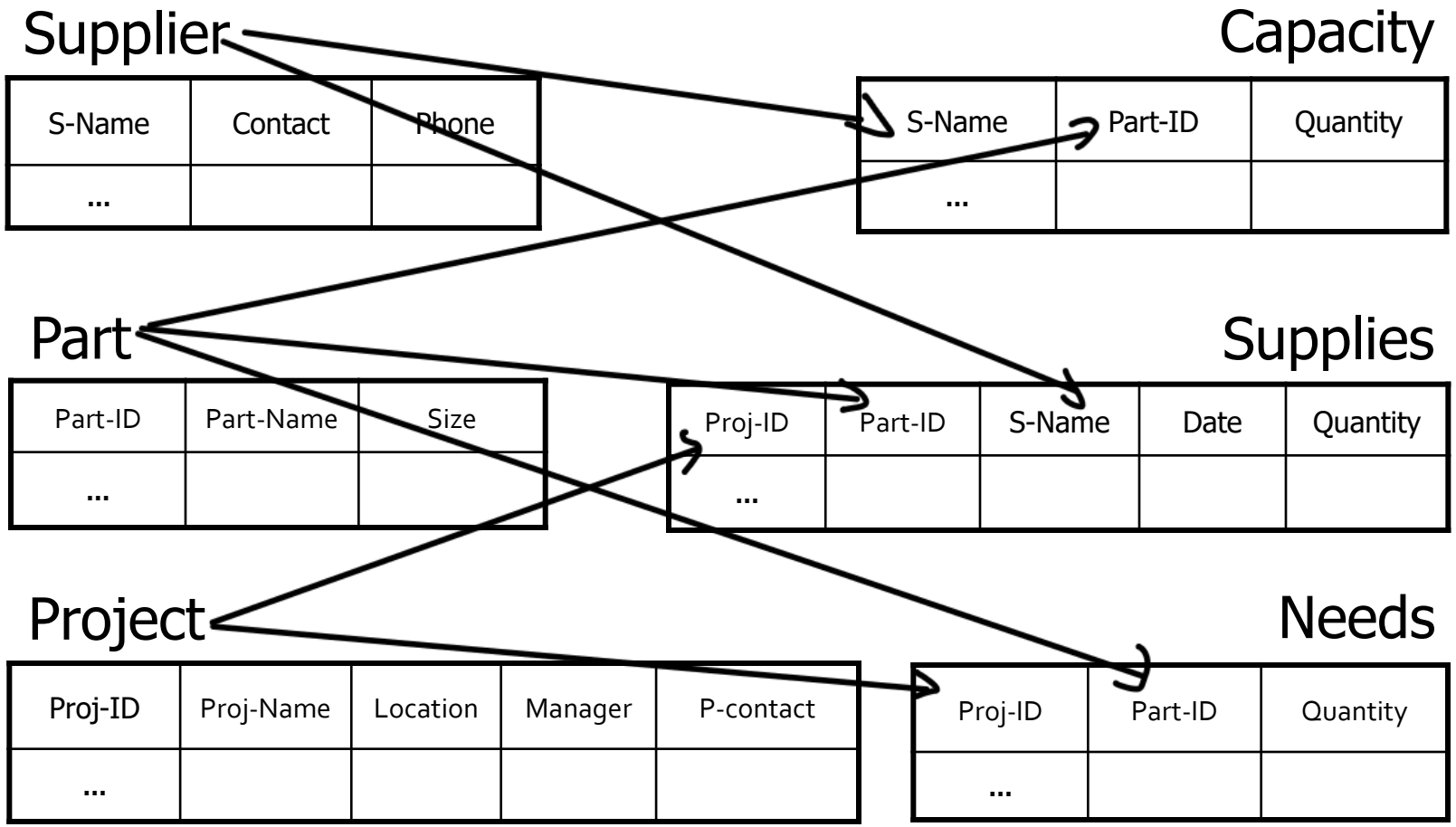
Proj-ID	*Location*	*Manager*	*P-contact*
P002	Seocho	K. Kang	222-1234
P003	Mapo	S. Choi	333-2345
P004	Jamsil	R. Smith	444-3456
…			

⇒ data seperation, and now...
have to combine these

Design 3 <ER Diagram>

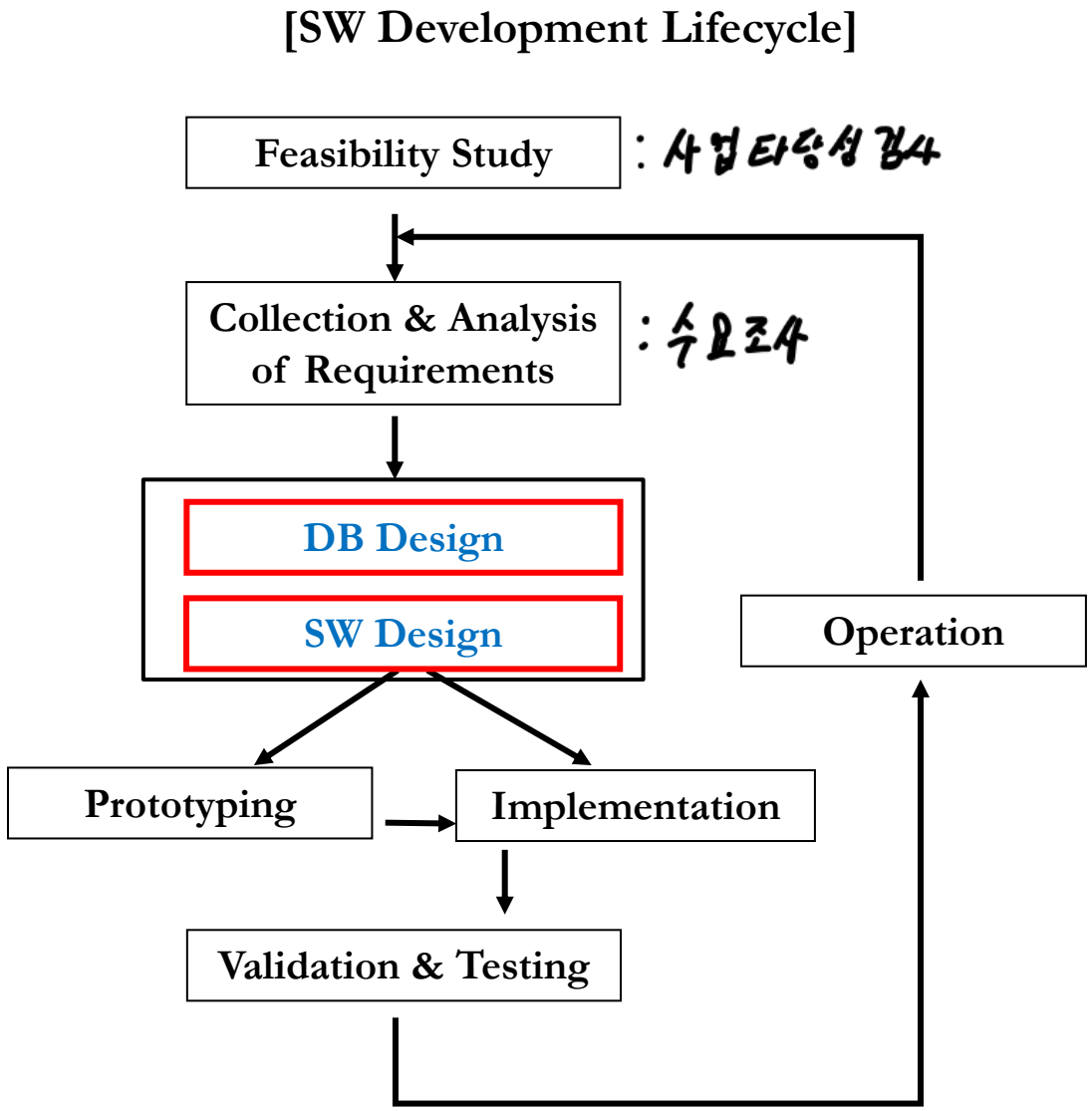


Design 3 <Relations>



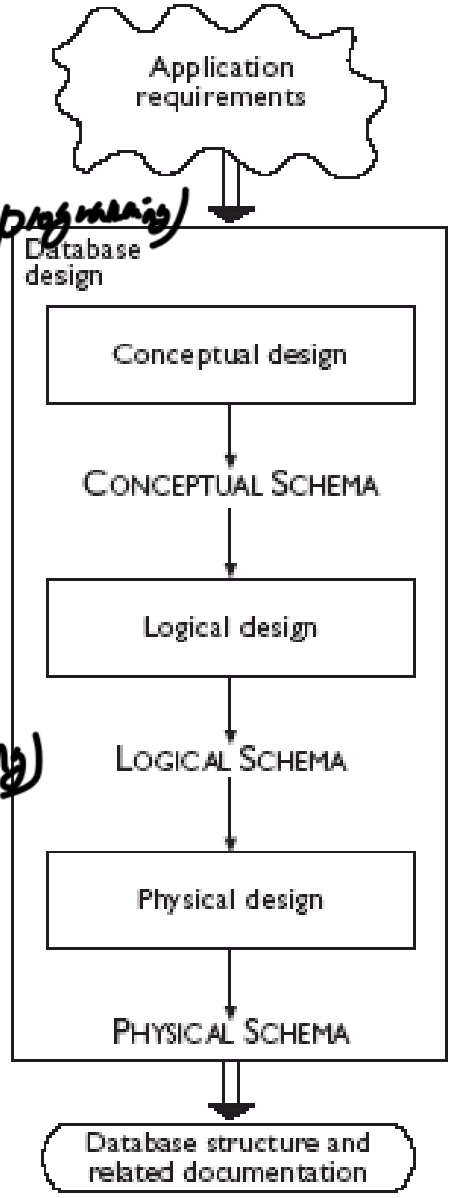
Database Design

- 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



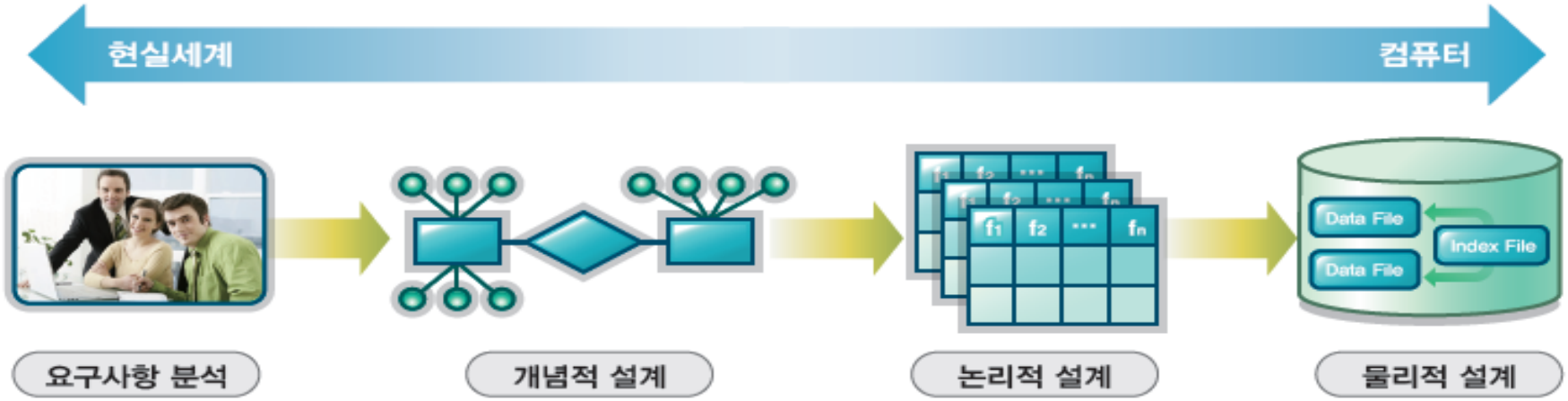
Phases of Database Design

- Three phases
- *Conceptual design* (schema setting, Before programming)
  - Construction of an ER schema
  - to provide an optimal description of the user requirements
- *Logical design*
  - map onto the implementation data model of the DBMS (such as RDB) (programming)
- *Physical design* (optimization)
  - 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
  - Effective means of communication between user, designer, and implementer

- E-R model is not an implementation model
  - i.e., there is no DBMS whose internal structures are based on the E-R model

not a programming sequence

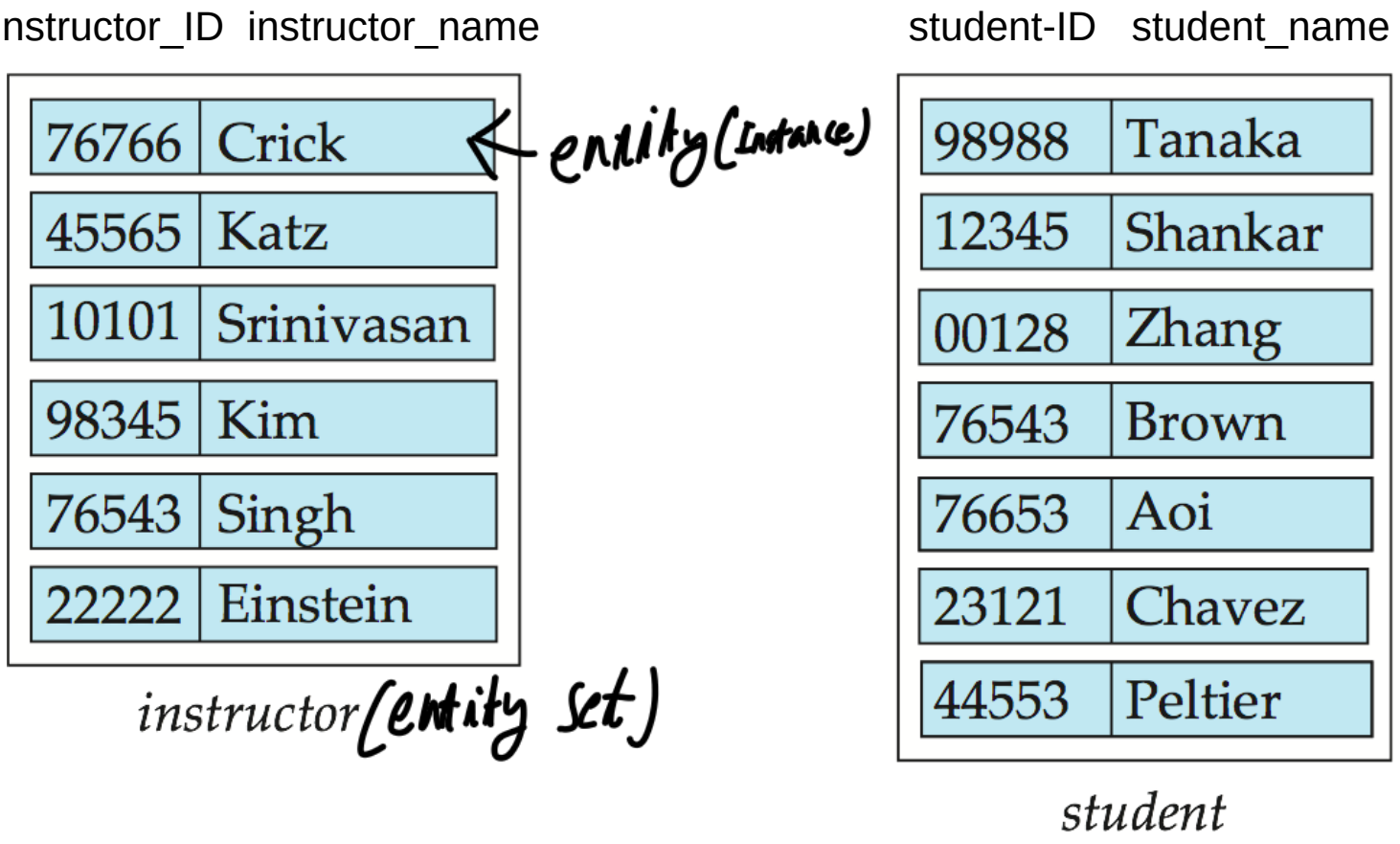
Database Modeling

- A *database* can be modeled as:
  - a collection of entities, and
  - relationships among entities.
- ***Entity*** (instance ≈ column)
  - an object that exists and is distinguishable from other objects
  - entity instance

    ex. specific person, company, event, plant
- ***Attributes***
  - Entities have attributes : specific data of entity
  - ex. people have *names* and *addresses*

- ***Entity set*** ≈ table
  - 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

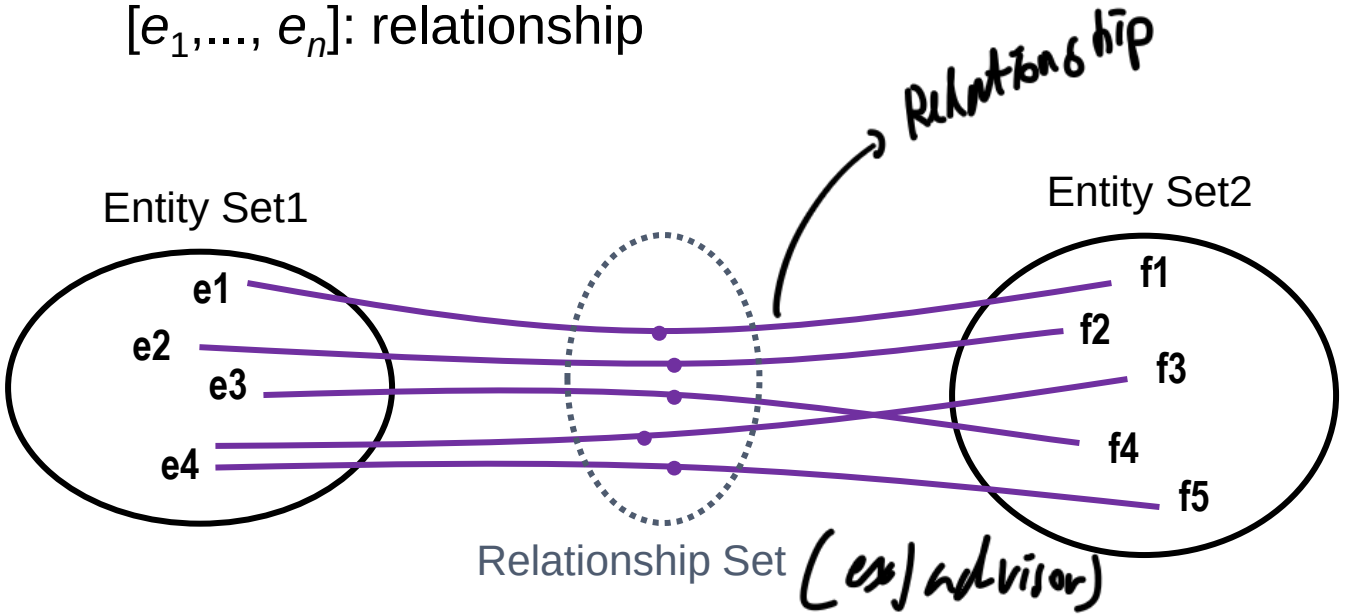


Relationships

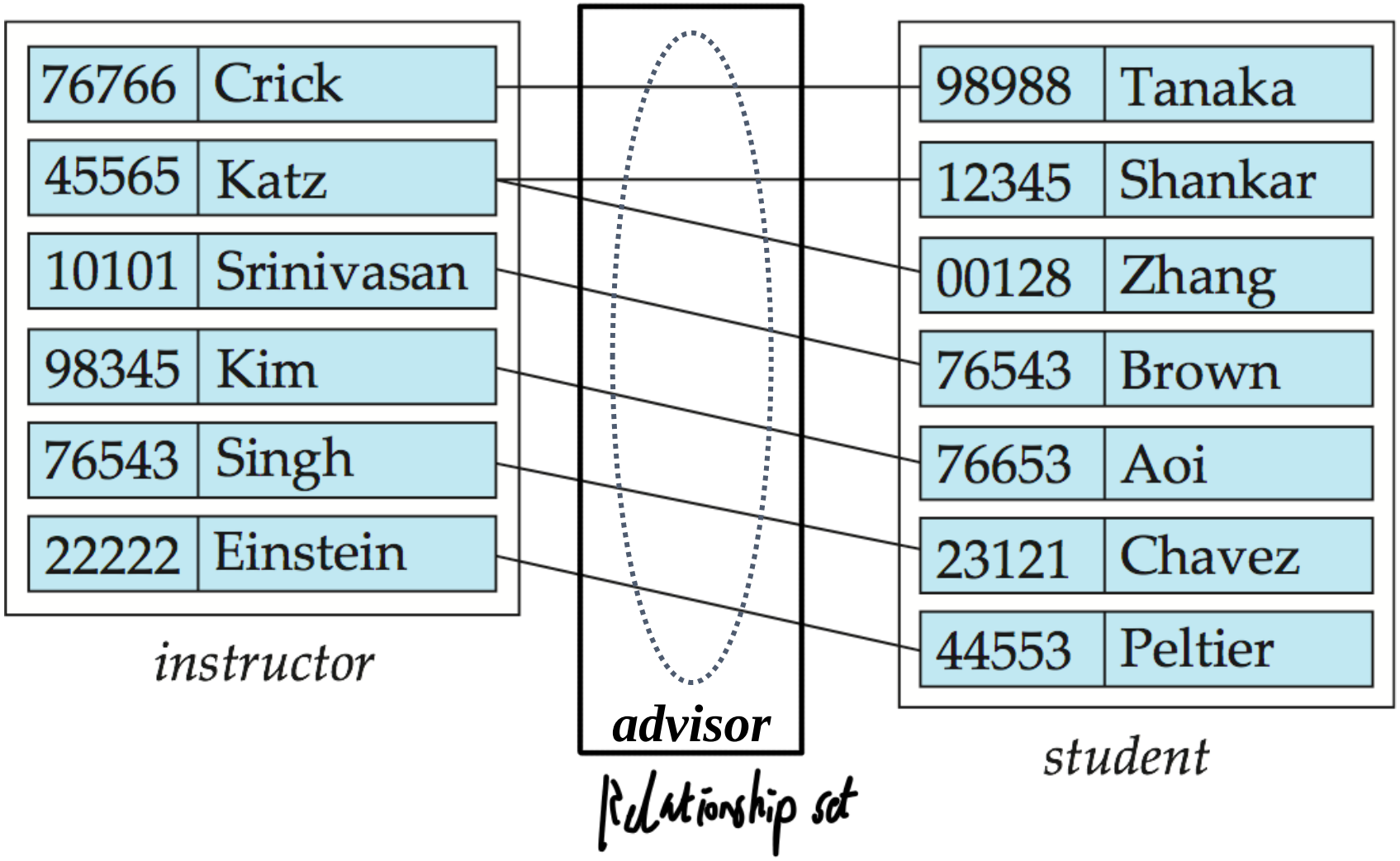
- ***Relationships***
  - Association between two or more entities
- ***Relationship set:***

$$R = \{ [e_1, ..., e_n] \mid e_1 \in E_1, \ldots, e_n \in E_n \}$$

where E_i: entity set,
$[e_1, ..., e_n]$: relationship

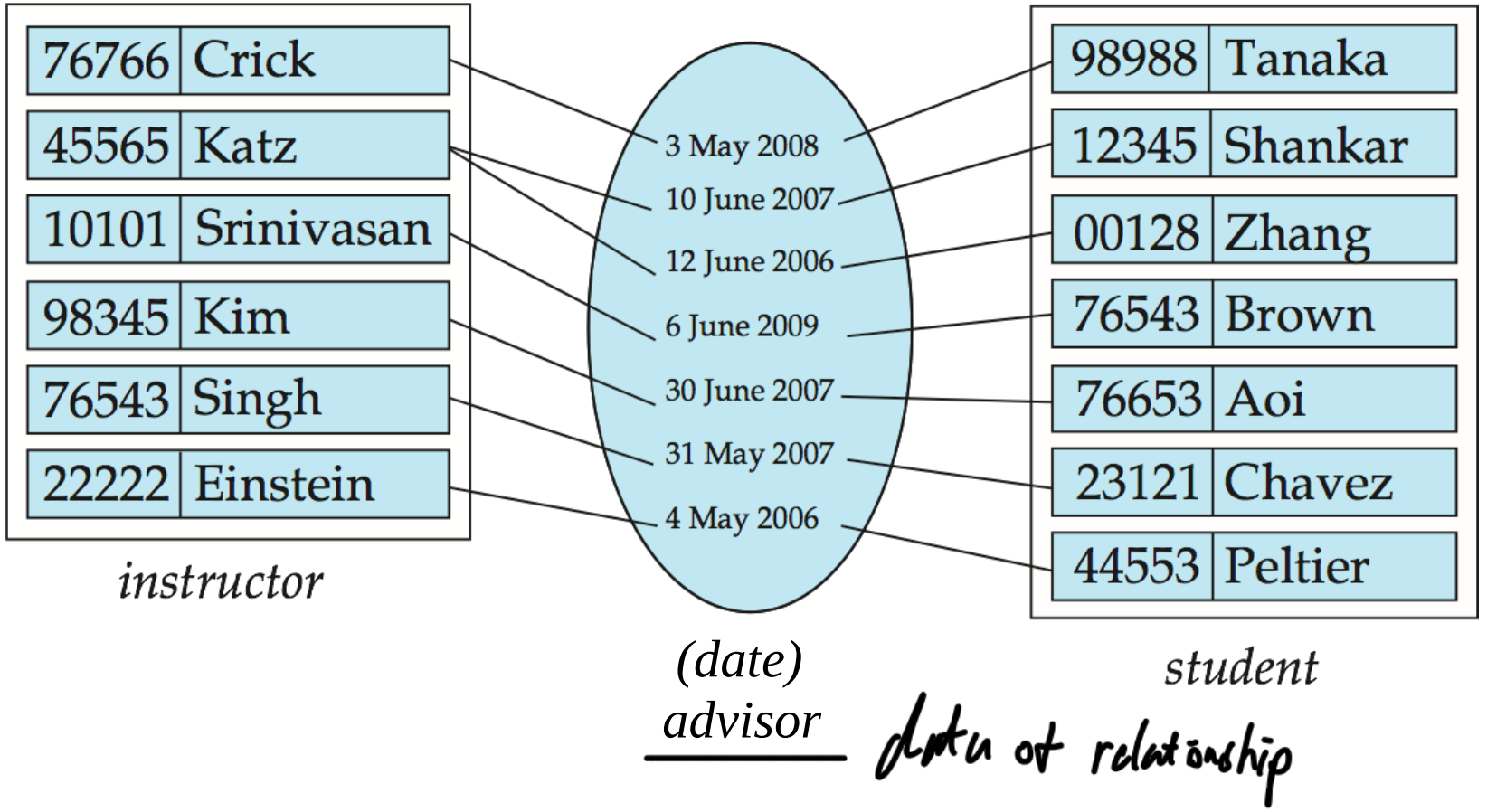


Relationship Set *advisor*

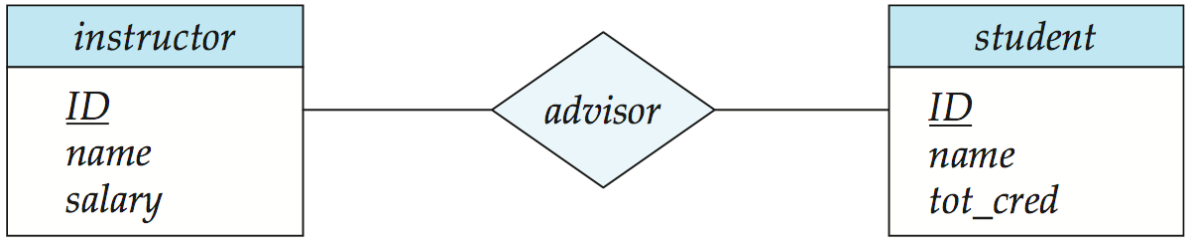


Attributes of Relationships

- Relationships can have attributes



E-R Diagrams



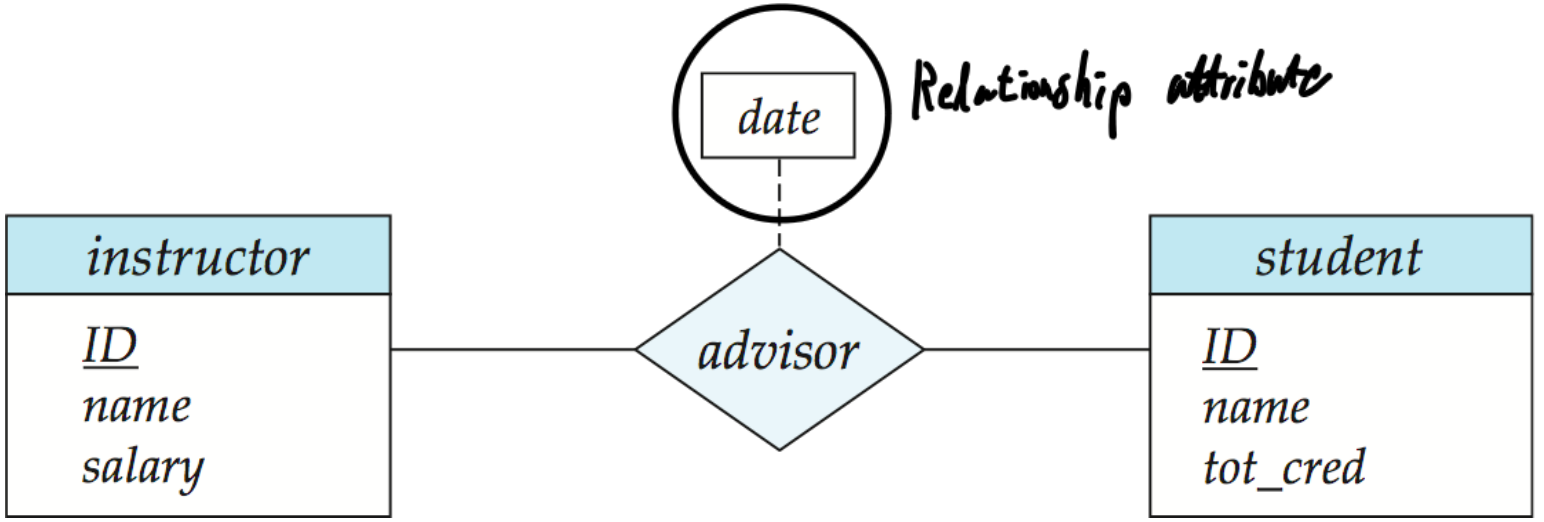
Rectangles represent entity sets.

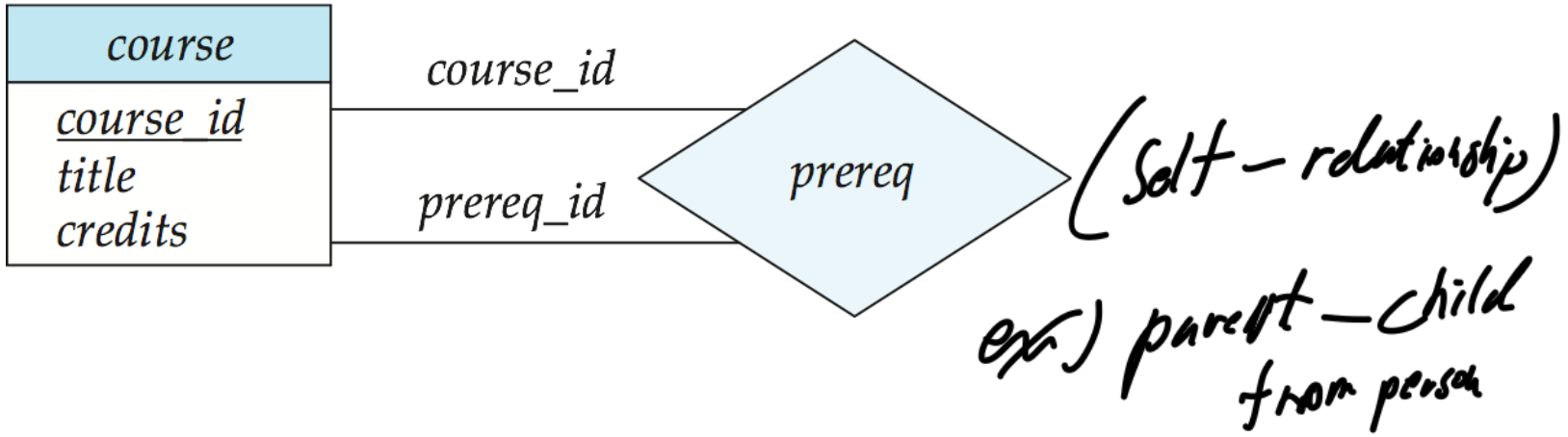
Diamonds represent relationship sets.

Attributes listed inside entity rectangle

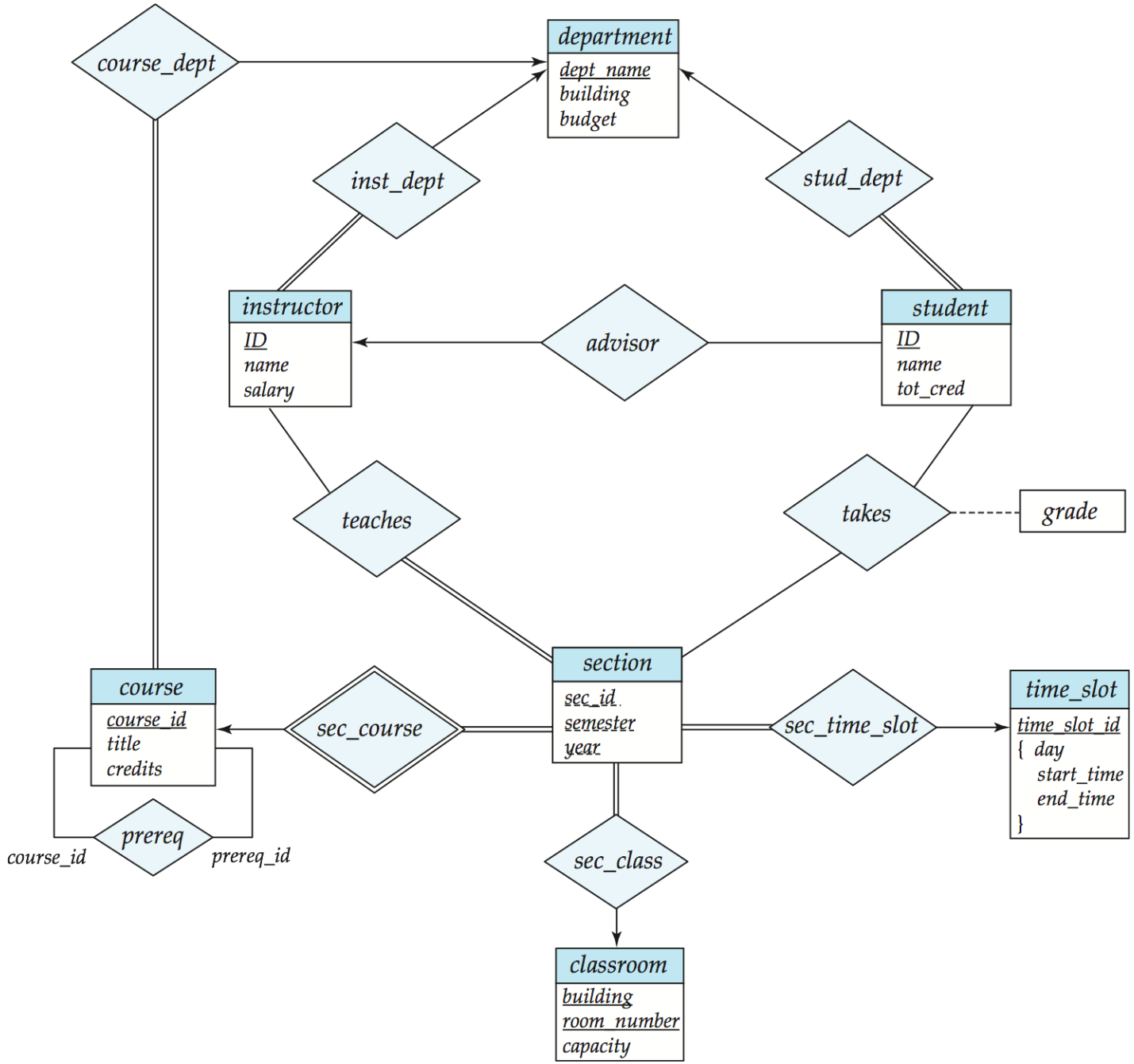
Underline indicates primary key attributes

Relationship Sets: Attributes & Roles

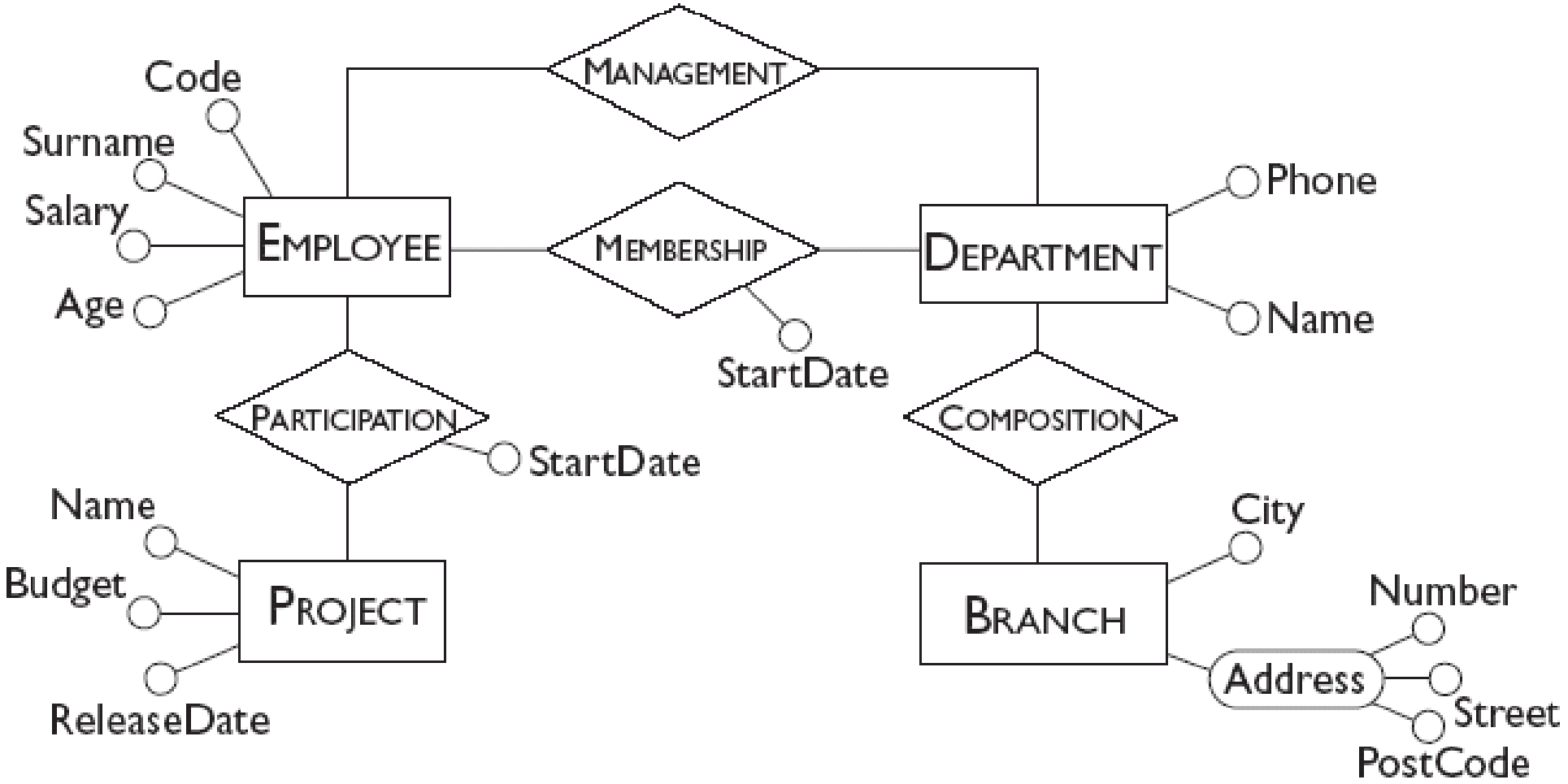




An ER Schema


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

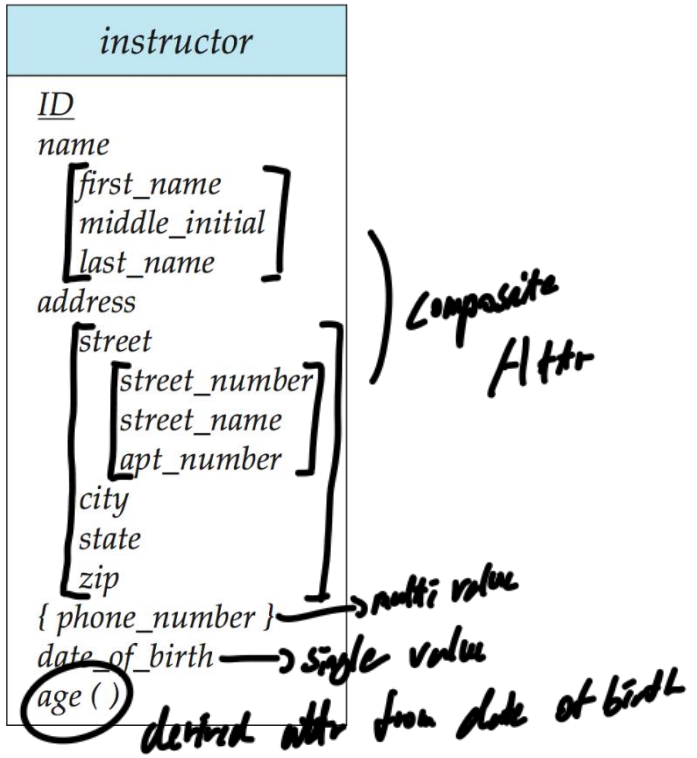
- Simple vs Composite attributes
  - Simple attribute (relational model)
    - values cannot be divided into subparts : atomic
    - *firstname, lastname, phone#*
  - Composite attribute (E-R model)
    - composed of multiple parts
    - *name = (lastnm, firstnm)*
    - *phone# = (number, extension)*

Why E-R use this?
⇒ Conceptual Design stage
more attention to requirement fulfilling.
∴ not closer to actual implementation

- 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 (variable)
    - *id, name, dept*
  - Multivalued attribute
    - an attribute may have more than one value for an instance (tuple)
    - *children = {john, tom}, phone#={5567, 5568}*

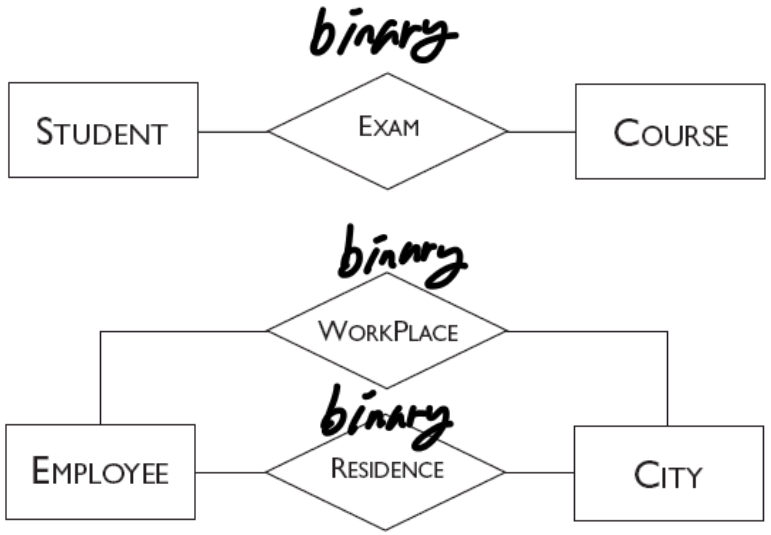


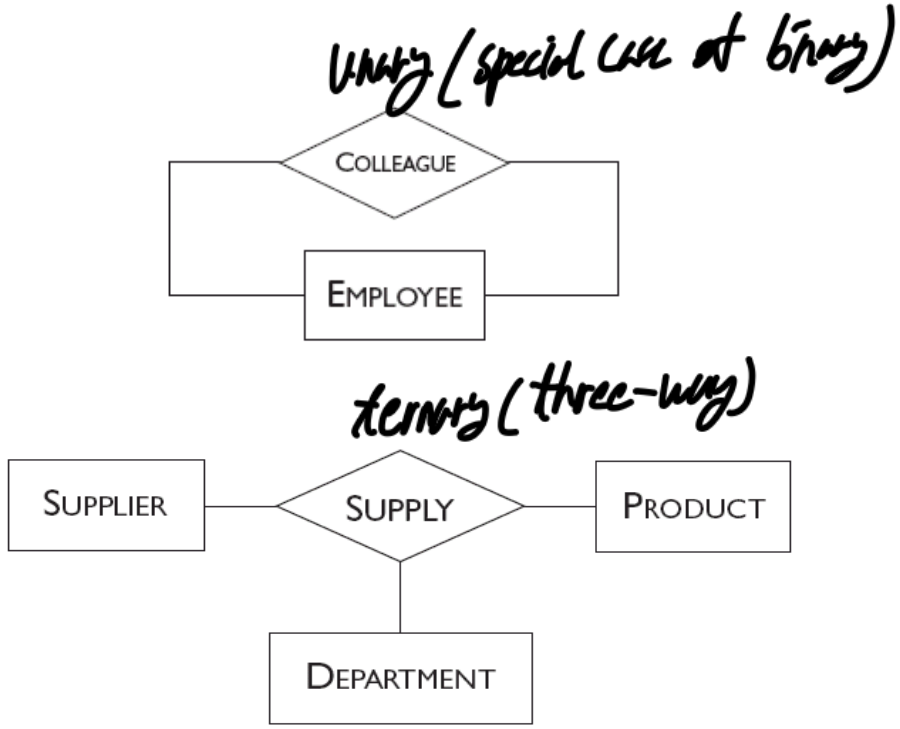
- Derived attributes
  - value can be derived from the values of other related attributes or entities
  - *duration, count, sum, …*

Degree of a Relationship Set

= Dimension of entity set vector

- 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

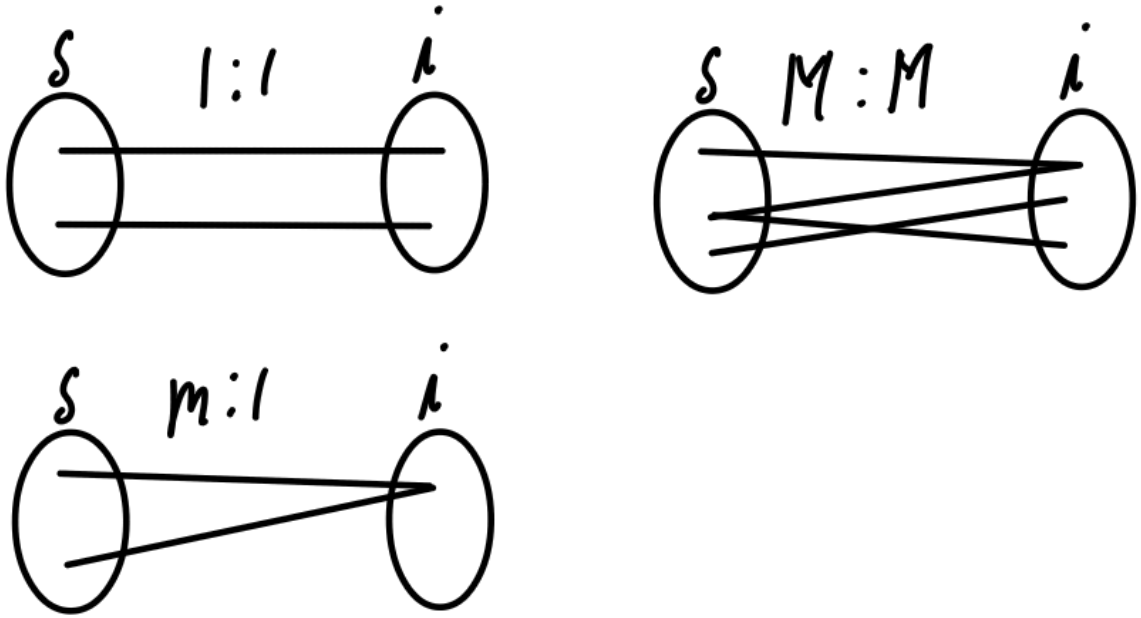




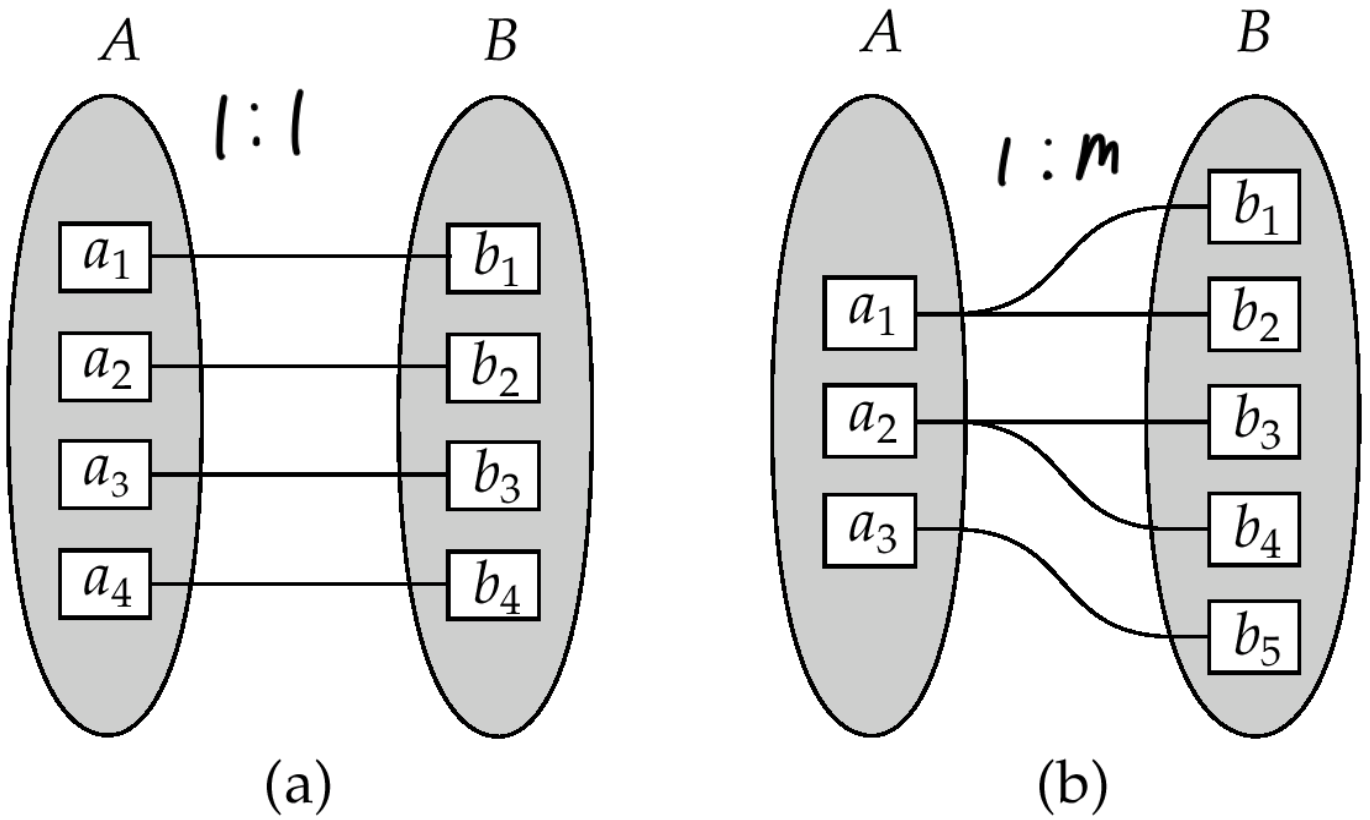
Mapping Constraints

- Relationship *cardinality* : correspondence
  - Number of entities to which another entity can be associated via a relationship set

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



Mapping Cardinalities

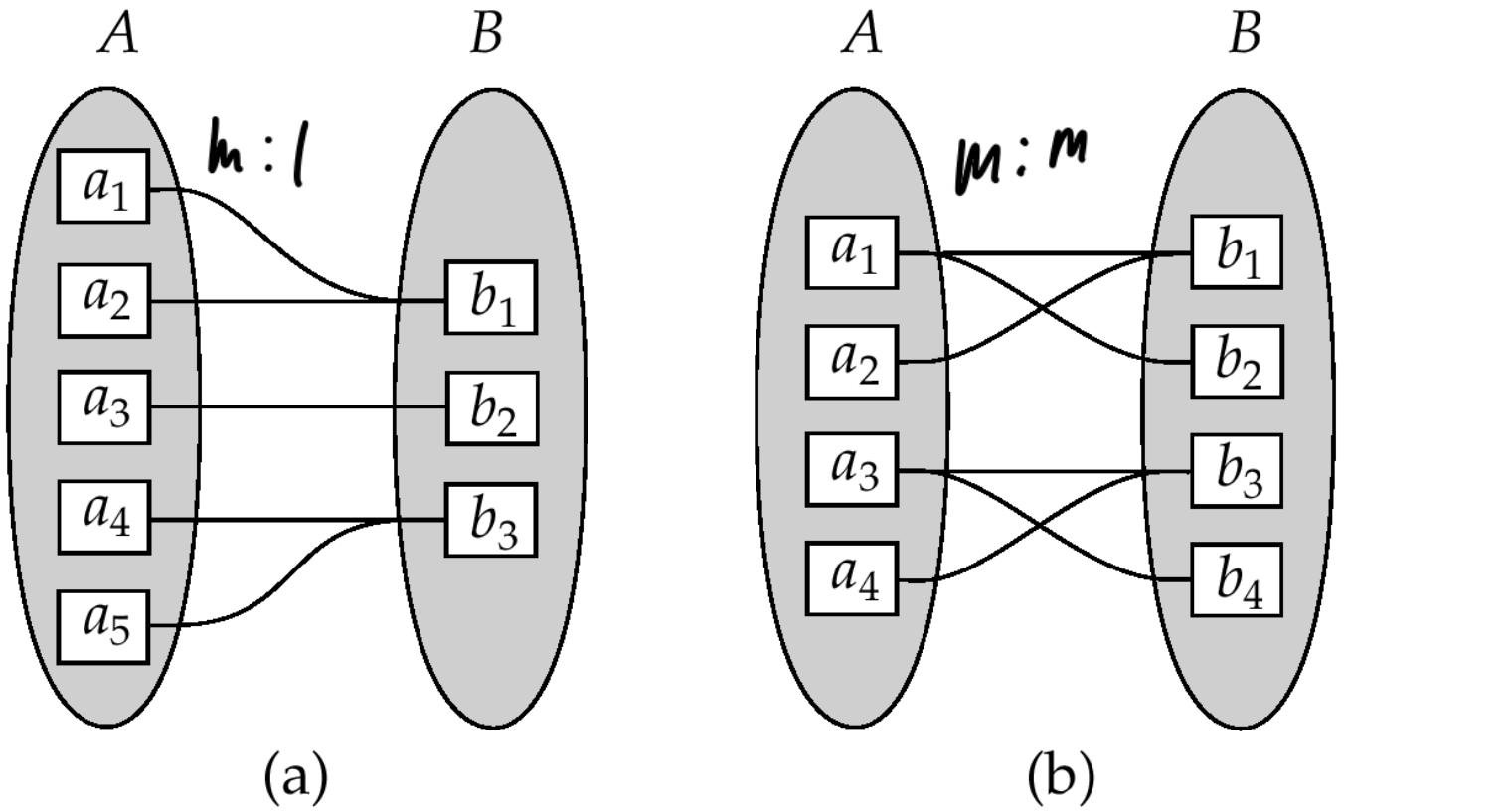


One to one

One to many

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

Mapping Cardinalities (cont.)

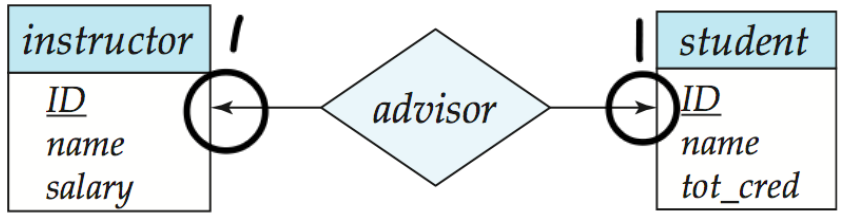


Many to one

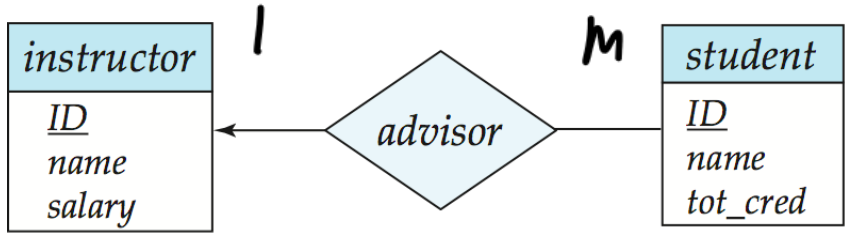
Many to many (no constraints)

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

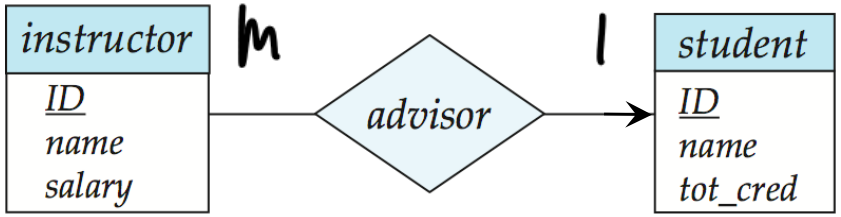
Mapping (Cardinality) Constraints : arrow means single correspondence



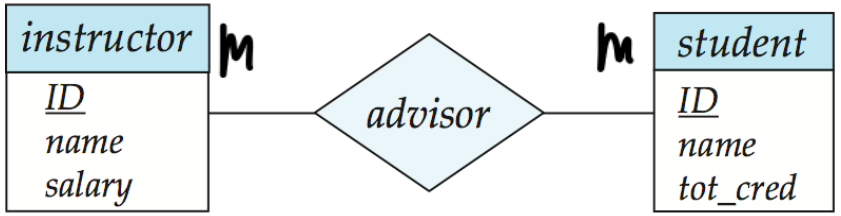
one-to-one



one-to-many



many-to-one



many-to-many

Participation of an Entity Set in a Relationship Set

Total participation (double line)

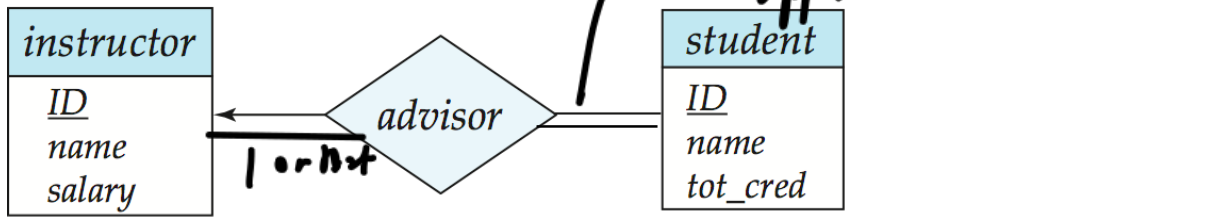
every entity in the entity set participates in at least one relationship in the relationship set

eg.: every *section* must have an associated course

Partial participation

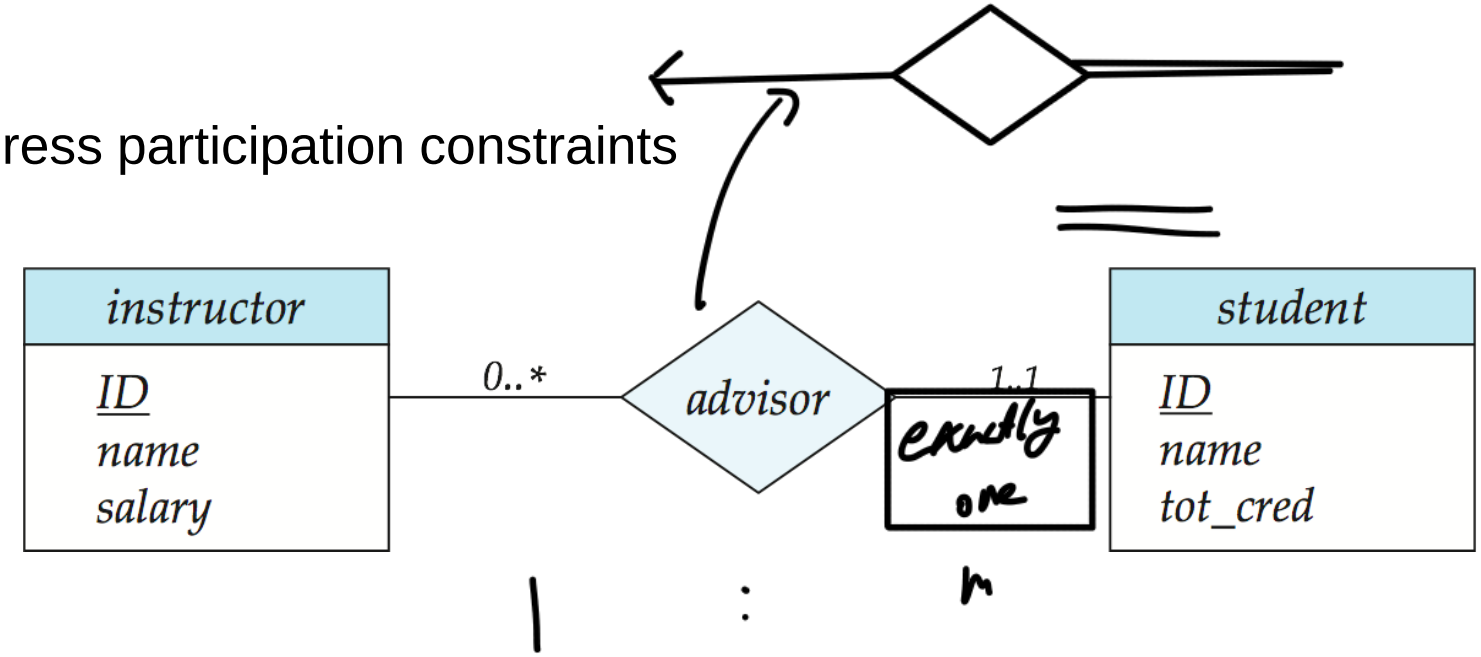
some entities may not participate in any relationship in the relationship set

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



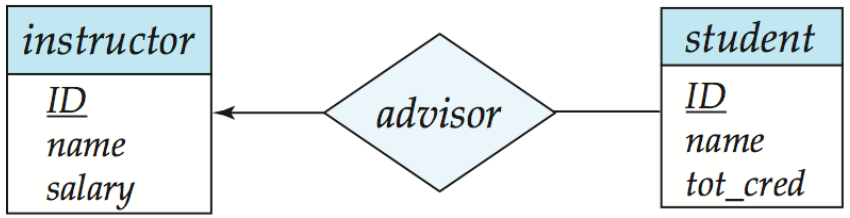
Keys

unique minimal representative

$S \supseteq C \supseteq P$

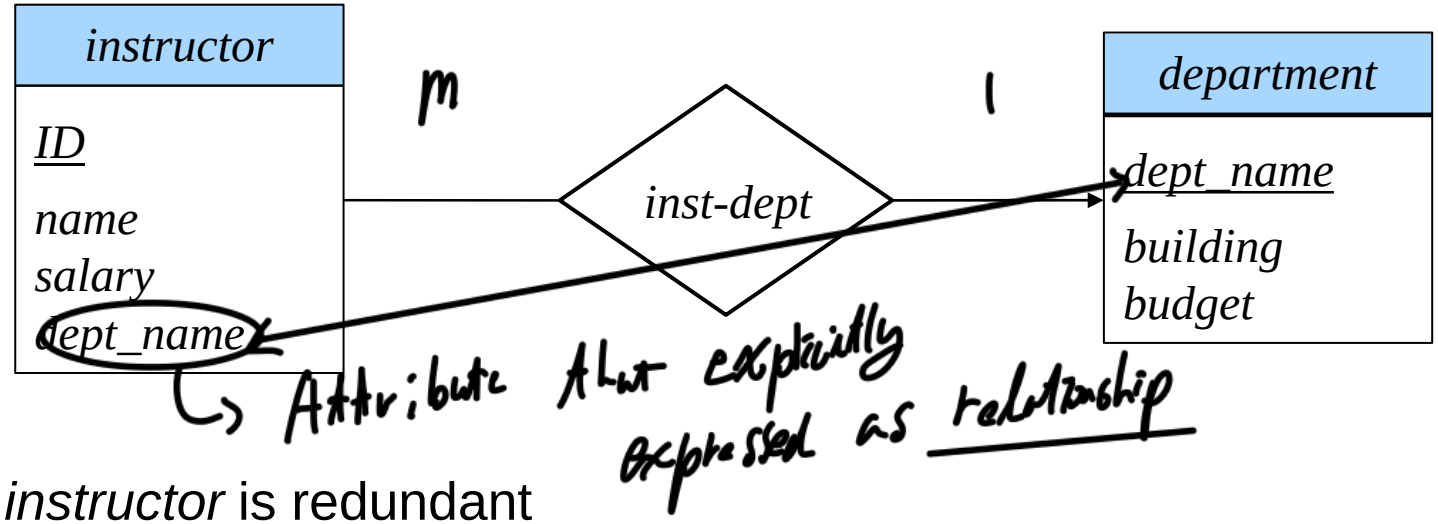
- Key for an Entity (Set)
  - Same as keys in relational models: super key, candidate key, primary key
  - Set of attributes whose values can distinguish entities from each other

- 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



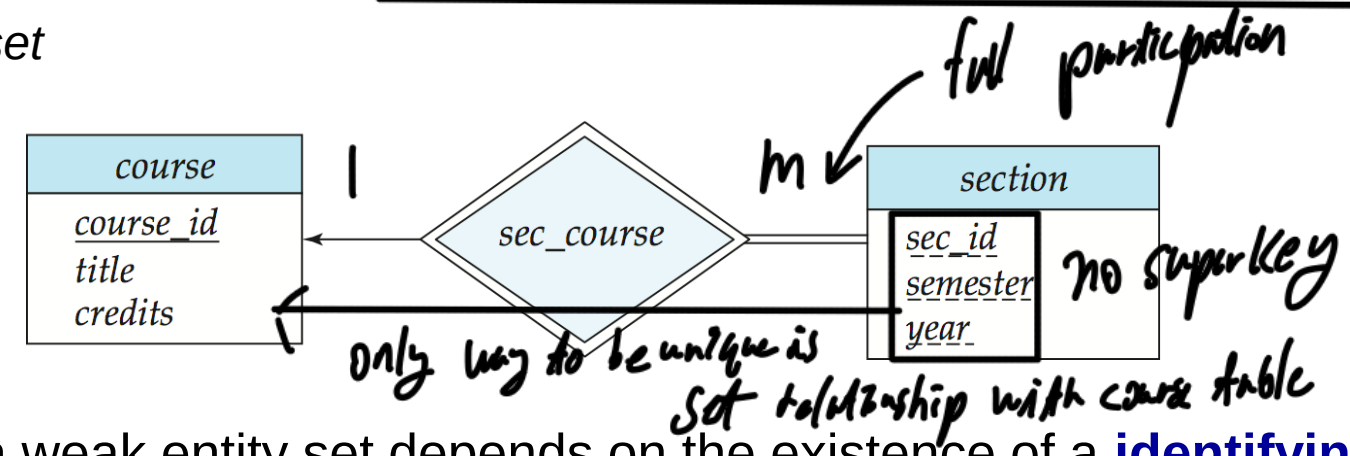
- *dept_name* in *instructor* is redundant

  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*
  - BUT: wasn't this a foreign key (which is important in schema diagrams)?!

↳ not important in E-R diagram

Weak Entity Sets

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



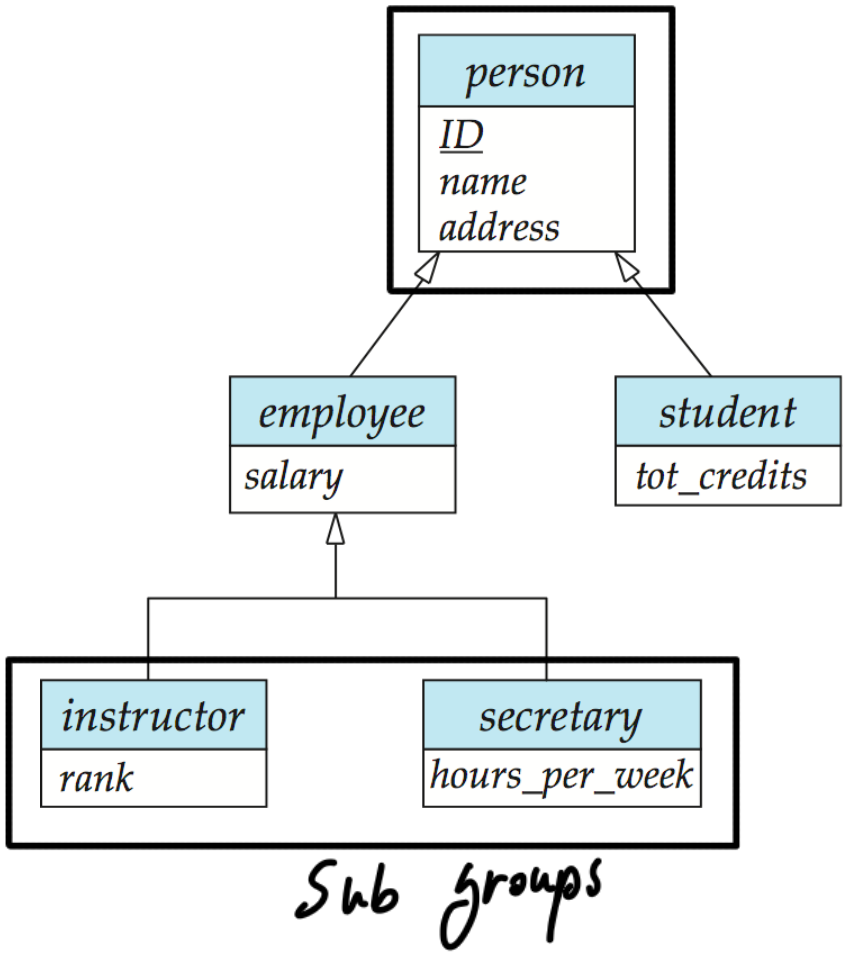
- The existence of a weak entity set depends on the existence of a **identifying entity set**
  - **Identifying relationship** depicted using a double diamond
- The **discriminator** (*or partial key*) of a weak entity set: set of attributes that distinguishes among all the weak entities related to the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

Extended E-R Features

- **Specialization** (top-down)
  - subgroupings within an entity set
  - Sub entities share common attributes
  - Each sub entity set may have its own specific attributes
- **Generalization** (down-top)
  - 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

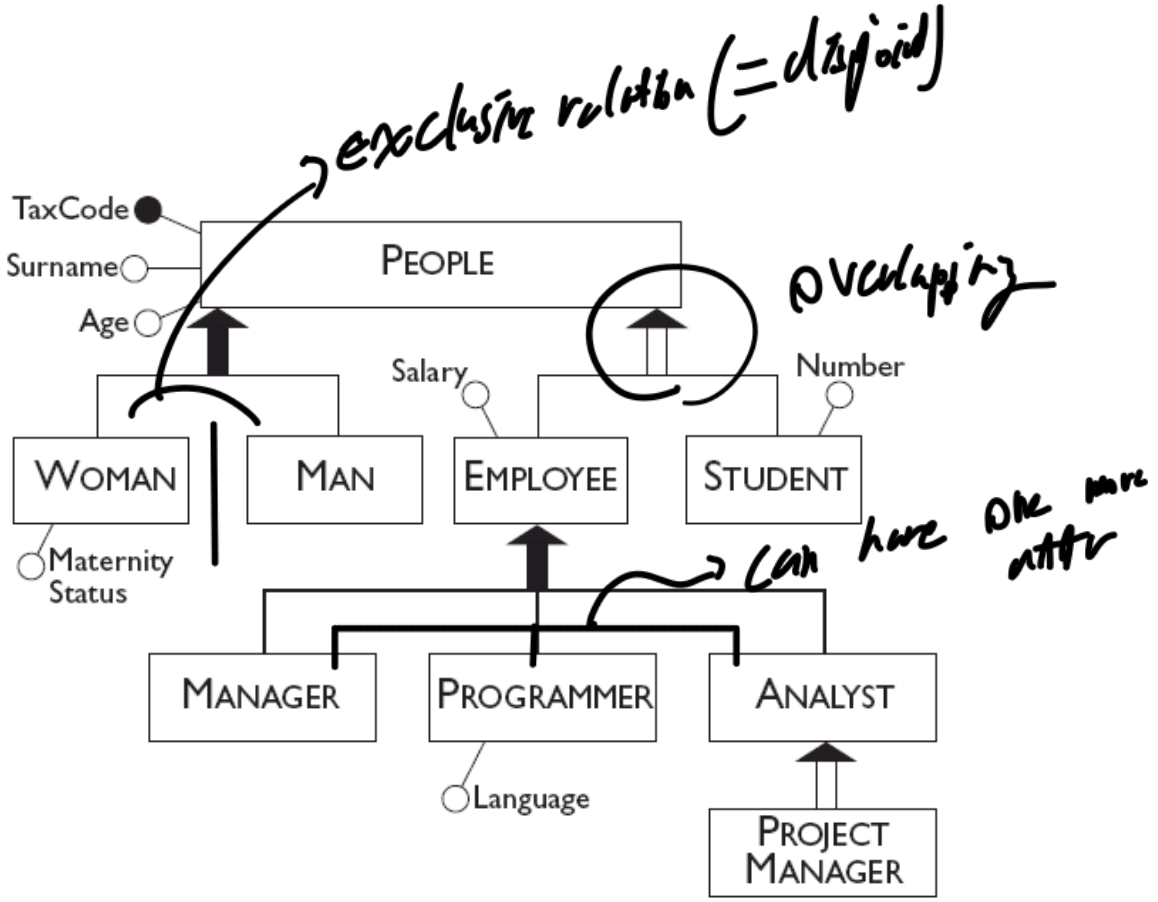
just method approach difference

common attribute

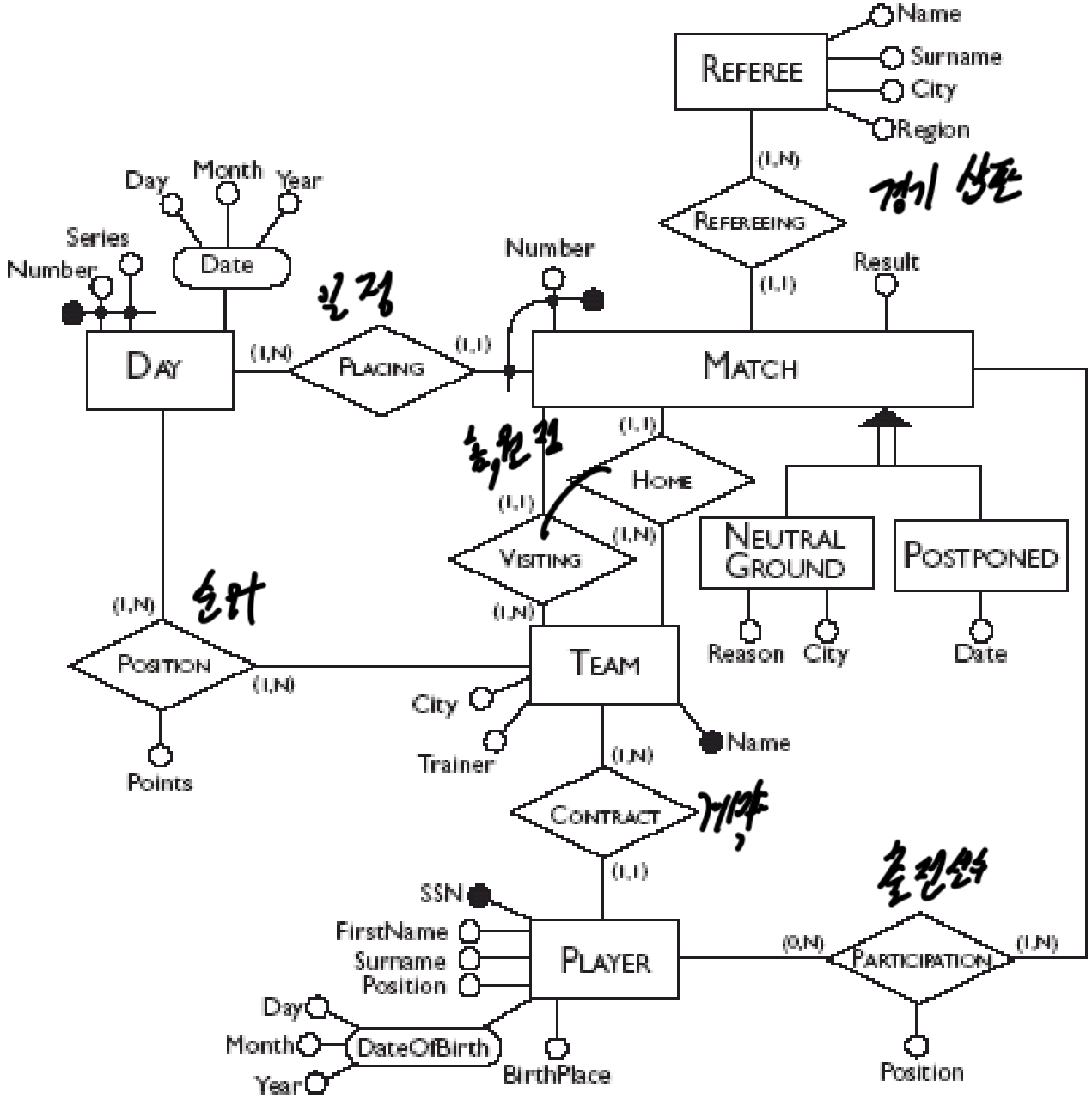


Extended E-R Features (cont.)

- 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:*
    - whether an entity can belong to more than two sub entity set
  - *total vs partial:*
    - whether every higher level entity belong to a lower level entity set



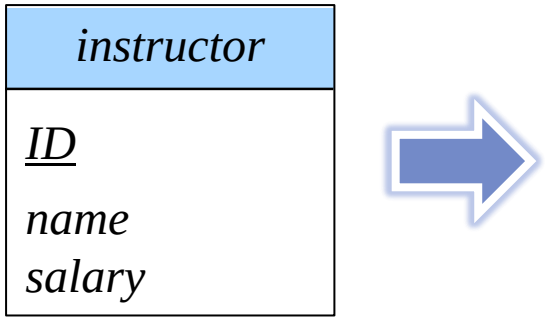
Exercise: Interpret the ER Diagram



[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
  - each entity set => unique table
  - each relationship set => unique table
- 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.)

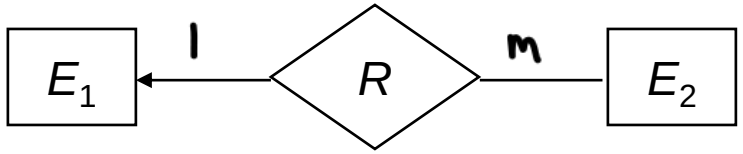
- Relationship set *R*
  - involving entities E_1, ..., E_k (k-degree)
    => table *r* with columns corresponding to

    $PK(E_1) \cup ... \cup PK(E_k) \cup attr(R)$: Union
  - *supplies* relating *supplier*, *client*, *part* C
  - *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

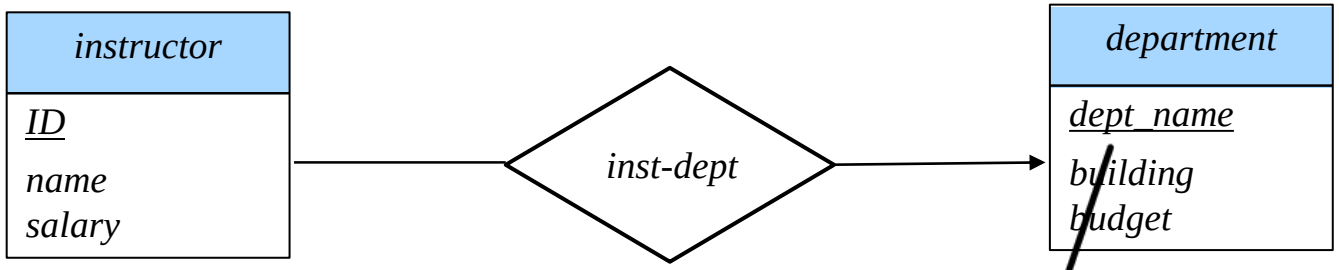
Combination of primary keys

  - Special cases: *one-to-many* or *one-to-one*



=> add columns representing $PK(E_1) \cup attr(R)$ to table representing E_2 ($PK(E_1)$ works as foreign key of E_2)

if 1 on 1 => $PK(E_1) \cup PK(E_2)$



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

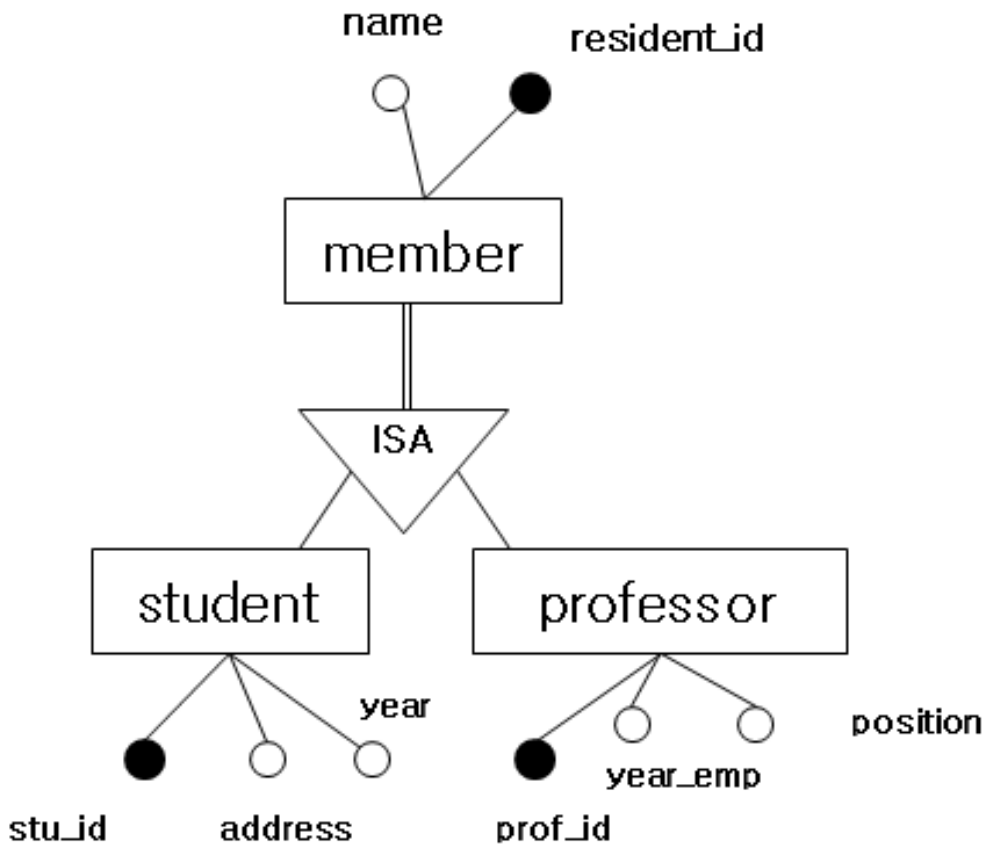
ID	dept_name
10101	Comp. Sci
12121	Finance
15151	Music
…	…

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

foriegn key

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

foreign key

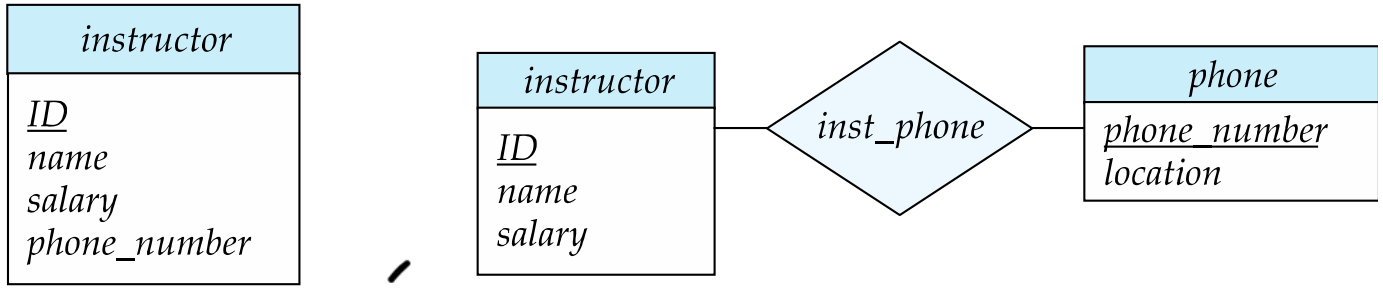
member(resident_id, name)
student(stu_id, resident_id, address, year)
professor(prof_id, resident_id, year_emp, position)

- 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)

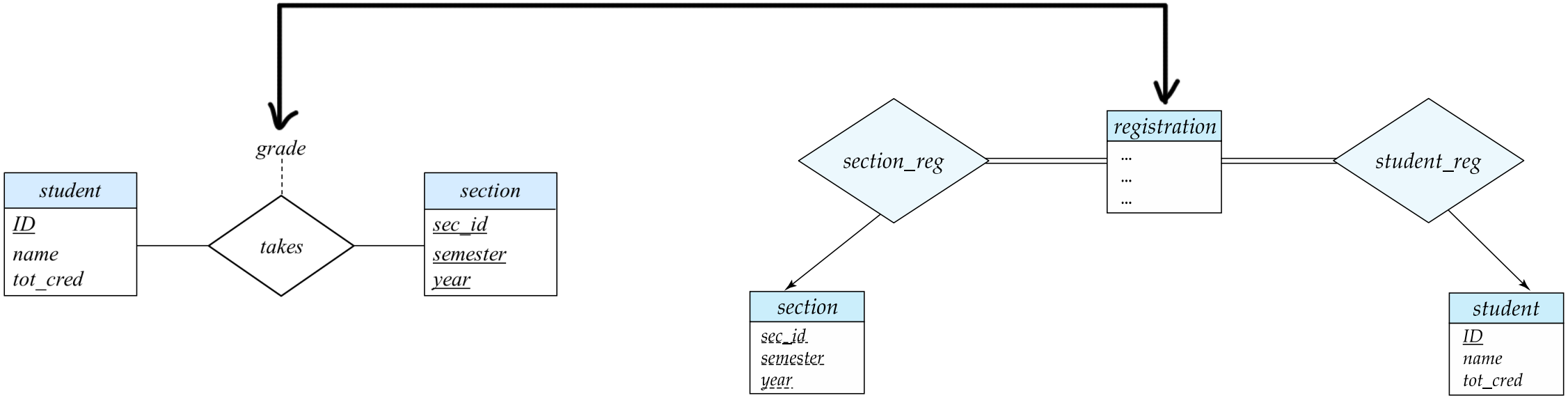
Design Issues

- *Entity vs Attribute* : is this information using frequently ?
  - 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



Design Issues (cont.)

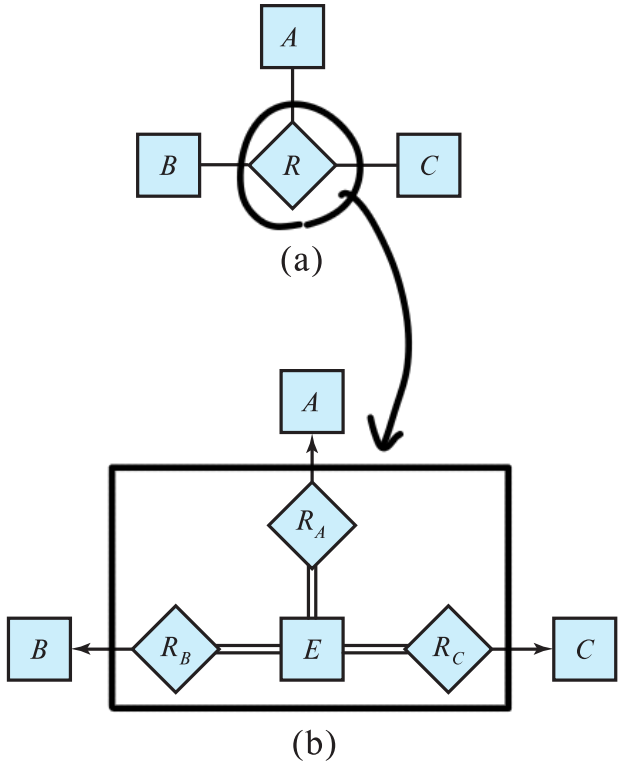
- *Entity vs Relationship*
  - "*student takes* (*has registration for*) a *section* (class)"
  - *takes* as relationship: simple but limited (cannot participate in other relationships)
  - *takes (registration)* as entity: can act as separate (independent) entity



Design Issues (cont.)

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

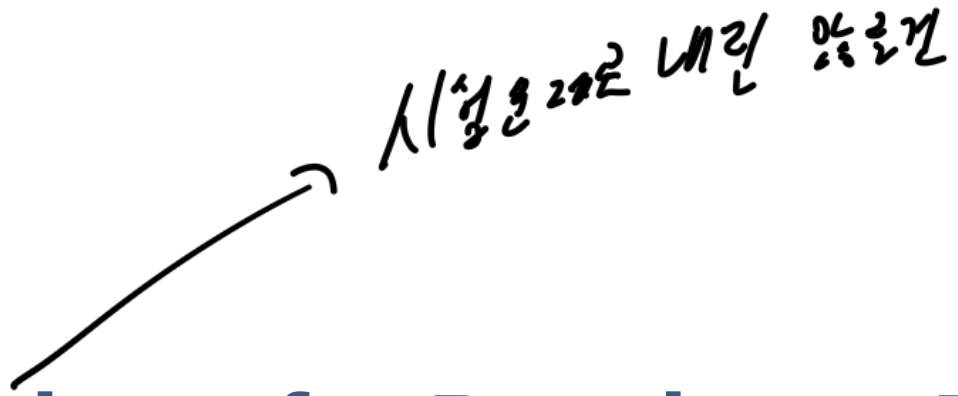
- *Generalization and specialization* : not important



Decision should be based on how the model

best represents the real world situation

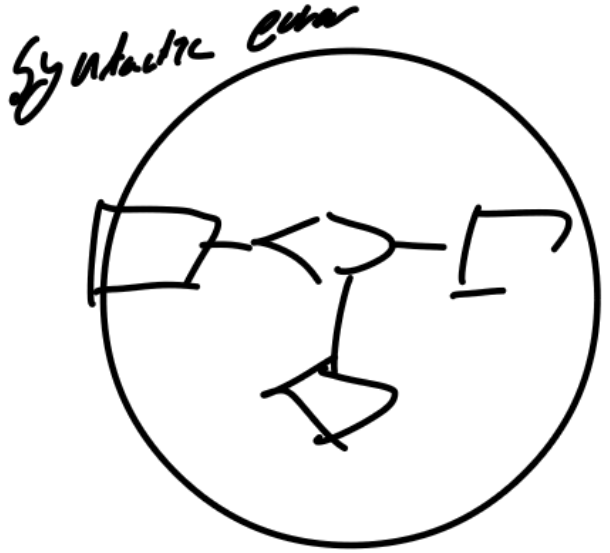
+2 : extensive/flexible



Qualities of a Database Design

Materials and examples from [Atzeni et al, 2000] (or [Batini, et al 1992])

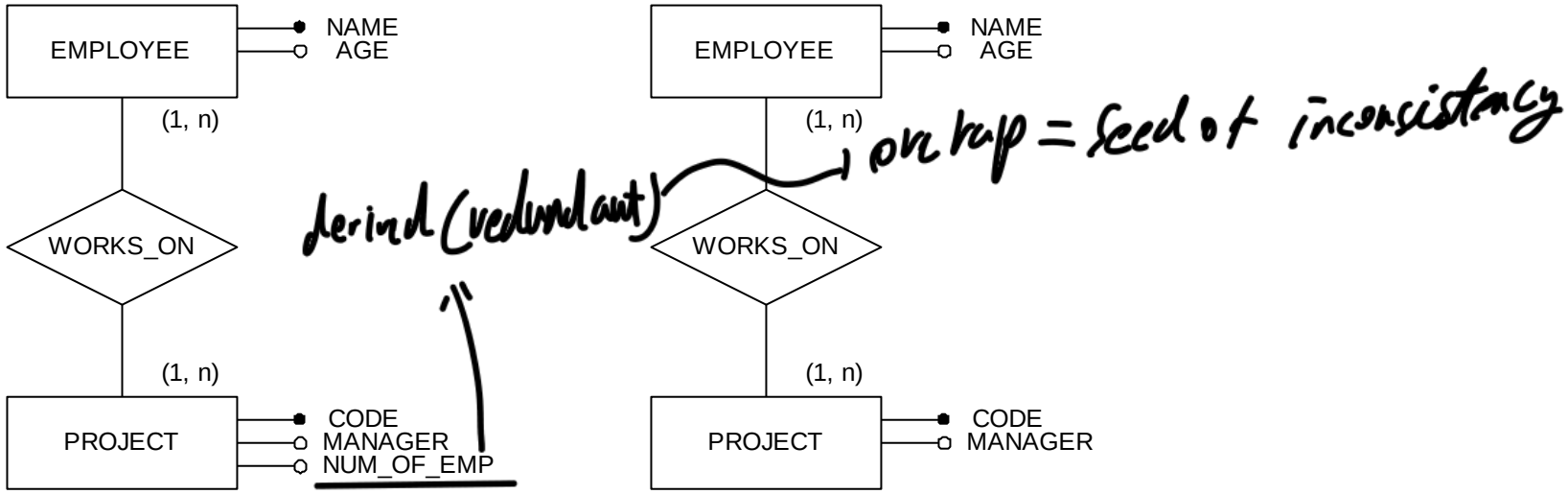
Completeness & Correctness



- **Completeness**
  - A schema represents all relevant features of the application domain.
- **Correctness**
  - A schema properly uses the concepts of the ER model.
  - Types of correctness
    1) Syntactic correctness
       - Concepts are properly defined in the schema.
    2) Semantic correctness
       - Concepts are used according to their definitions.
       - For example, To represent products, using an attribute when we need to represent several properties of products. (e.g., code, price, parts…)

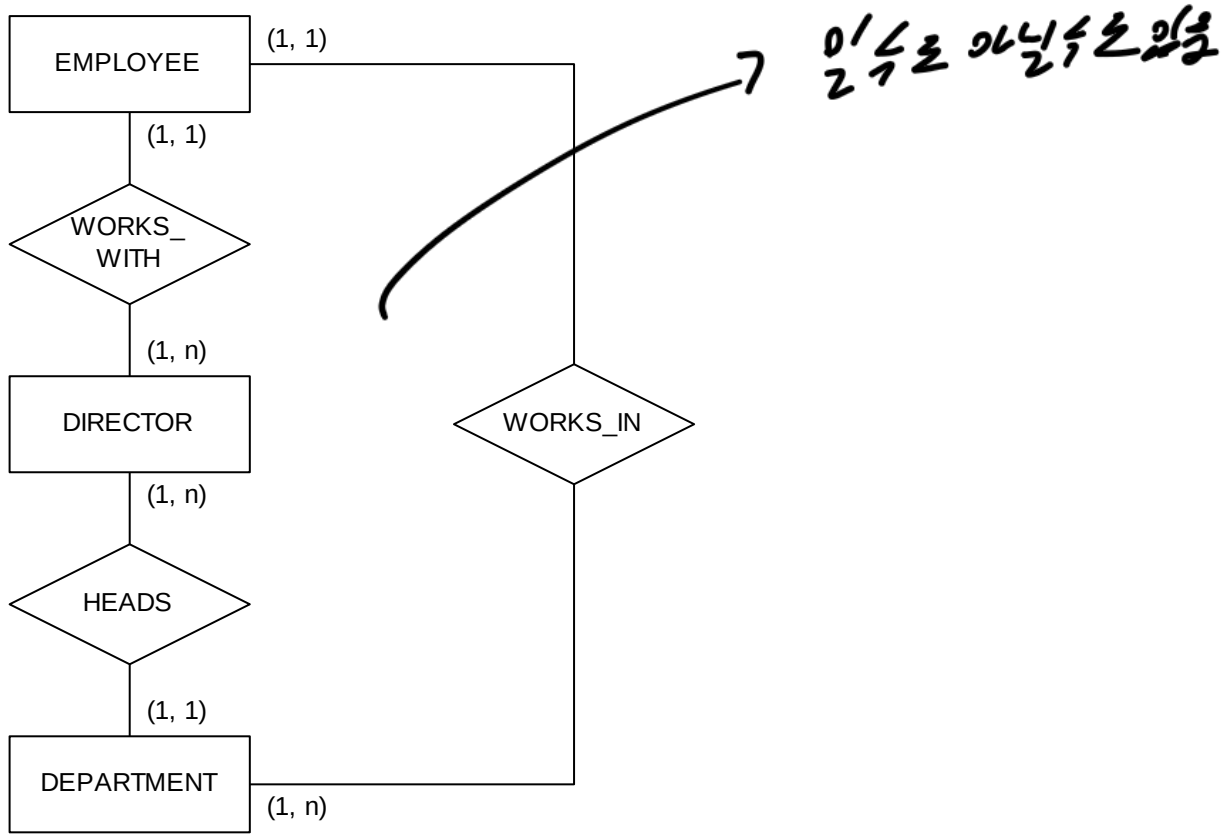
Minimality (no-duplicate) / necessary

- Every aspect of the requirements appears *only once* in the schema.
- No concept can be deleted from the schema without losing some information.



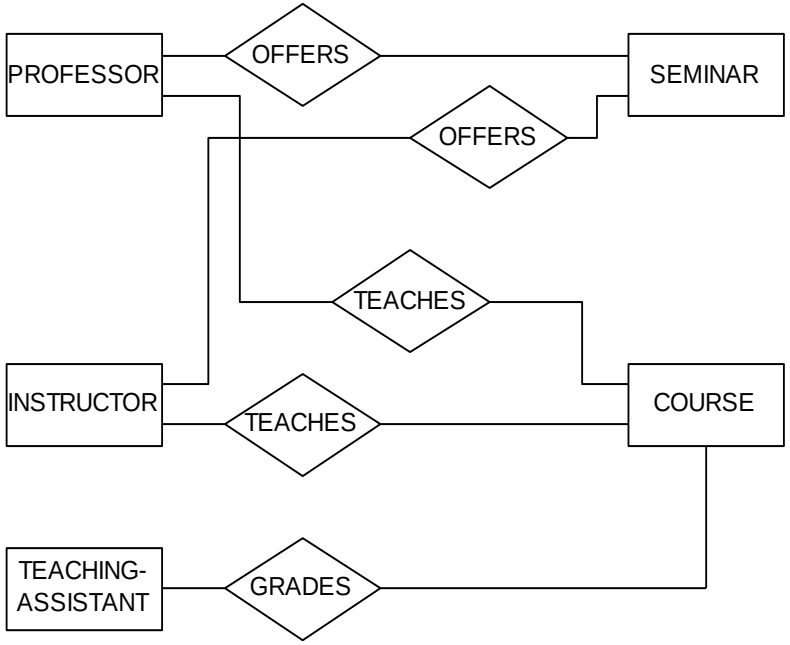
Examples – Minimality

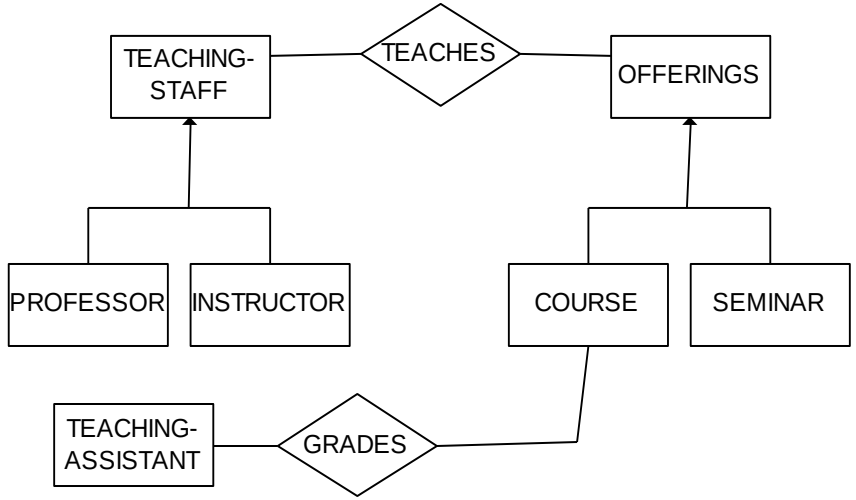
- Cycles of Relationships



Expressiveness (better way to represent)

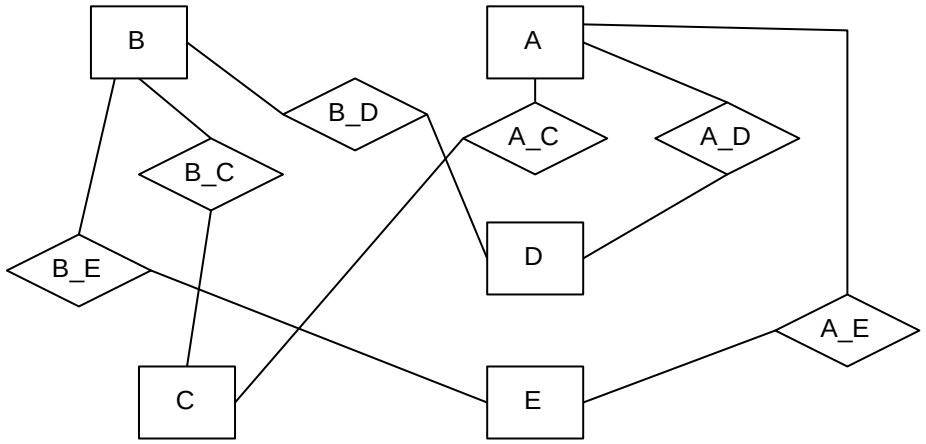
- A schema represents requirements in a natural way and can be easily understood through the meaning of ER, without the need for further explanation.

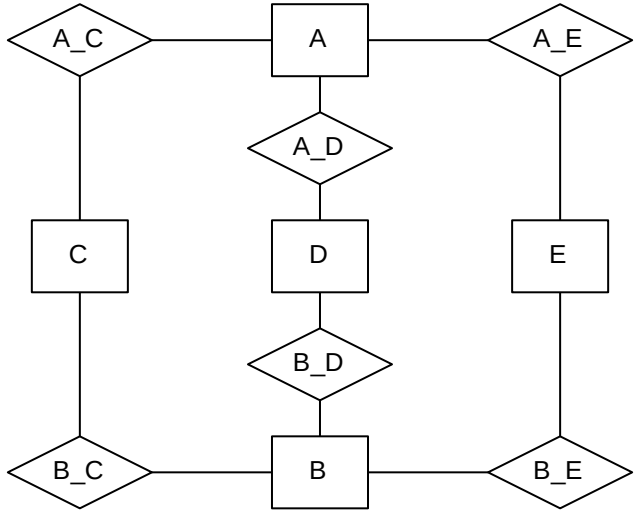




Readability

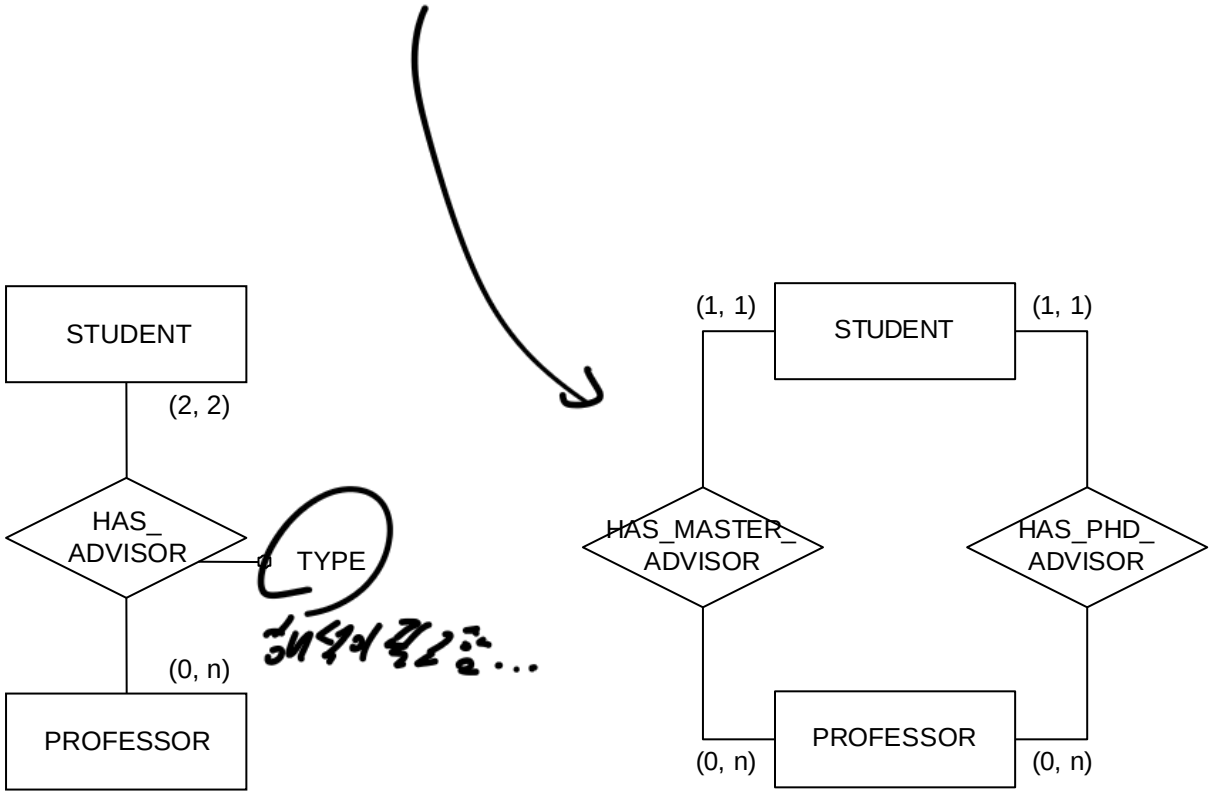
- A schema respects certain aesthetic criteria that make the diagram graceful



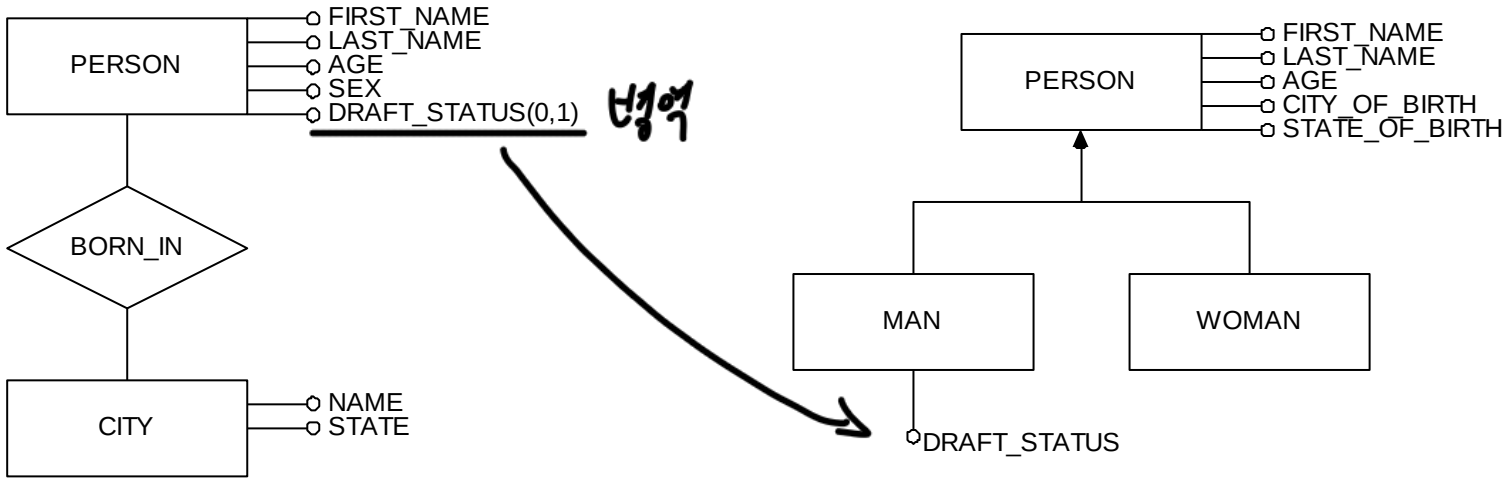


Self-Explanation

- A large number of properties can be represented using the conceptual model itself, without other formalisms (e.g., annotations in natural language.)

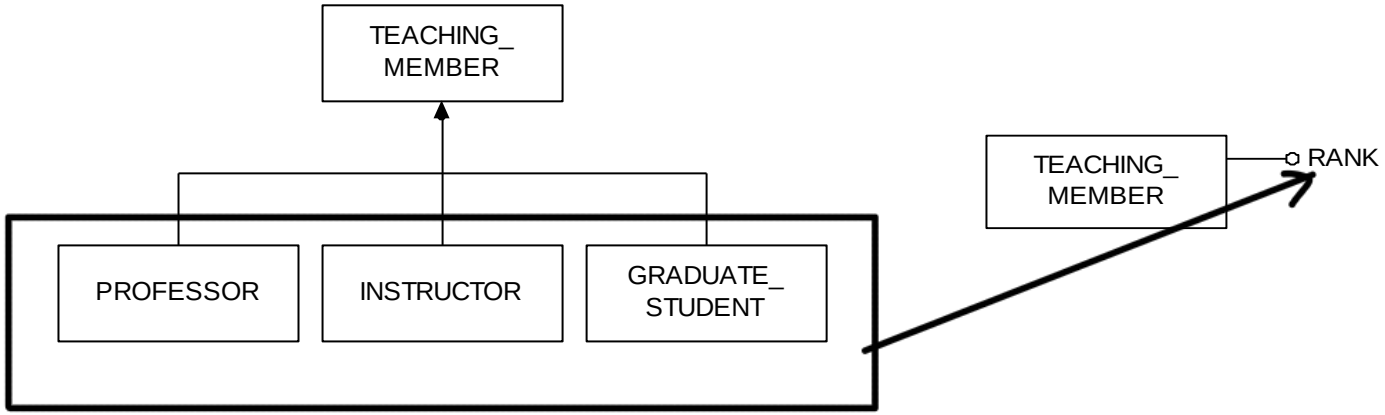


Examples – Self Explanation

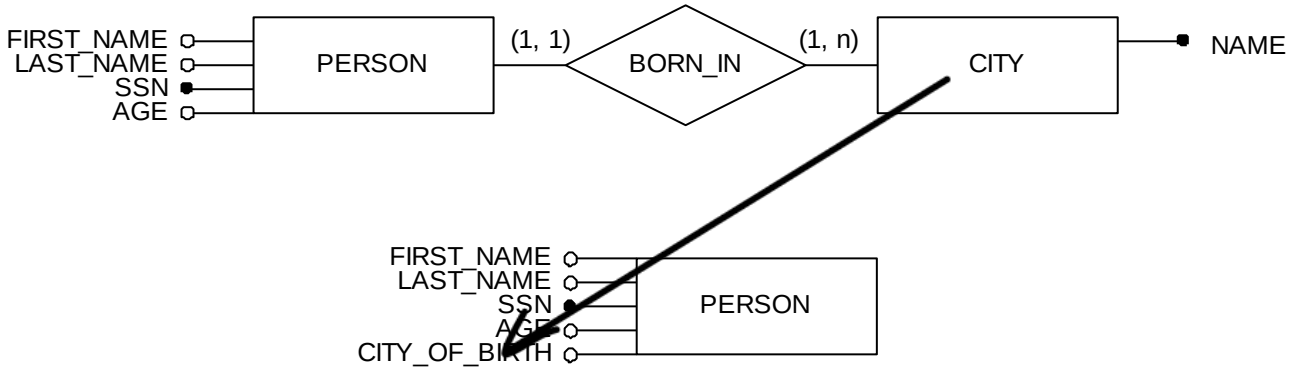


Examples – Expressiveness / Self-Explanation

- Elimination of Dangling Sub-entities in Generalization Hierarchies

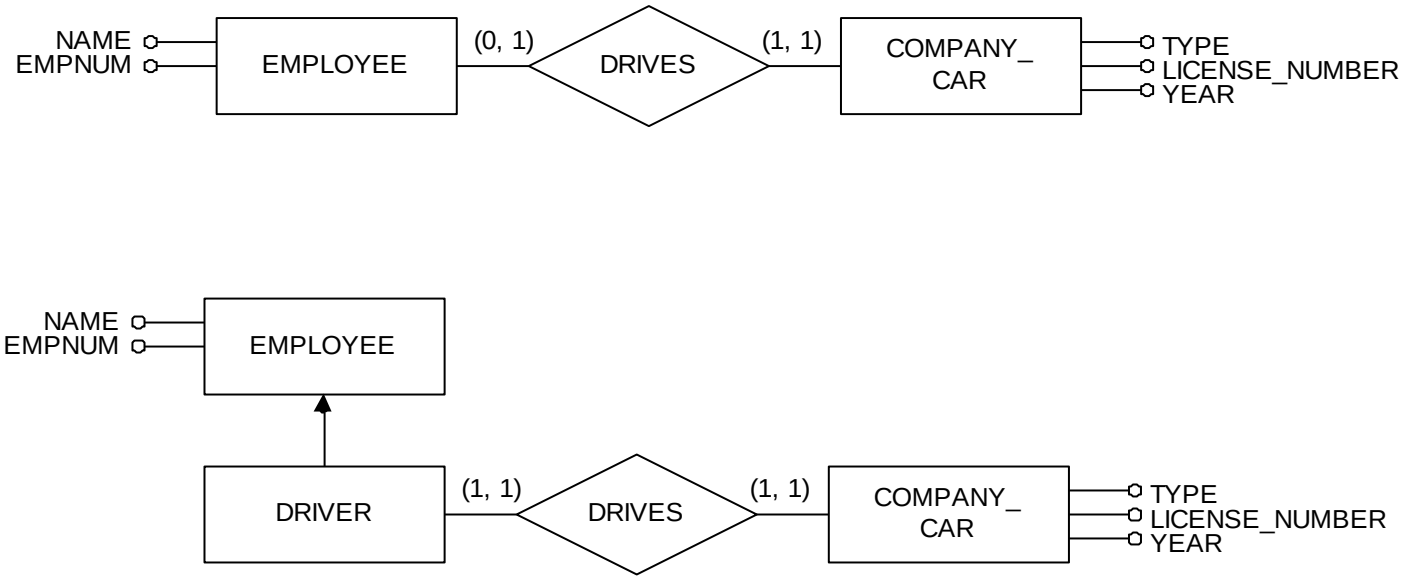


- Elimination of Dangling Entities



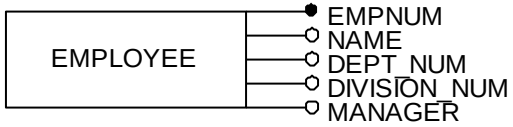
Examples – Expressiveness / Self-Explanation

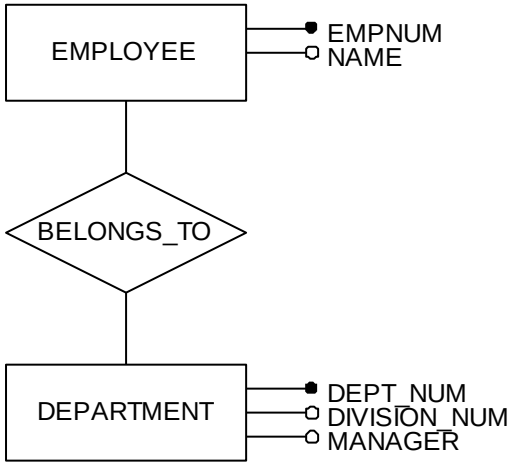
- Creation of Generalization/Specialization

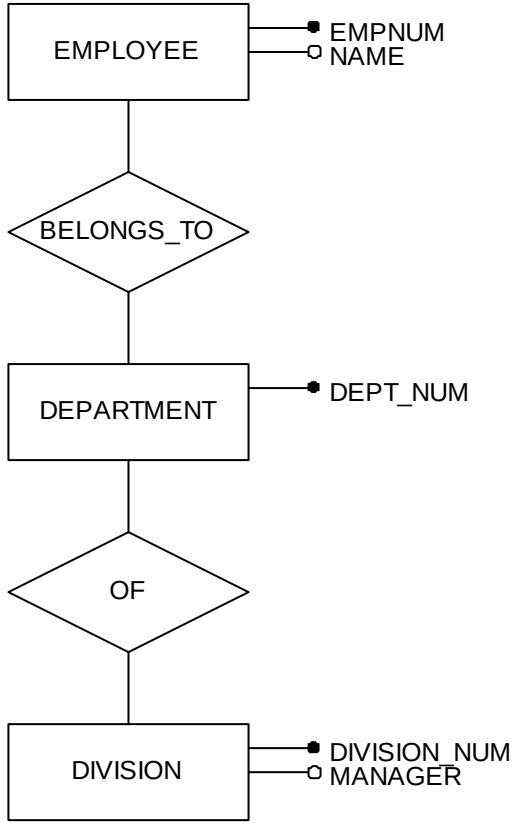


Normality (BCNF & 3NF)

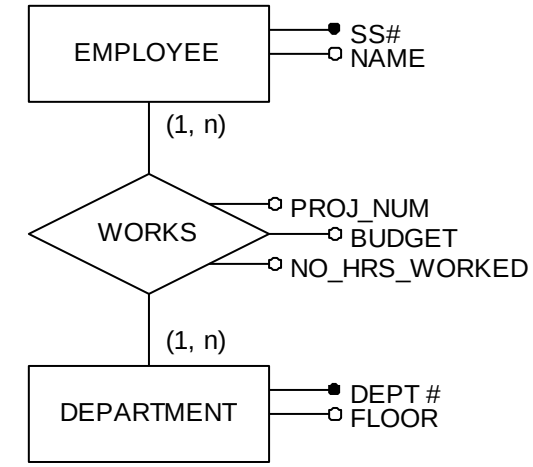
- By use of functional dependency

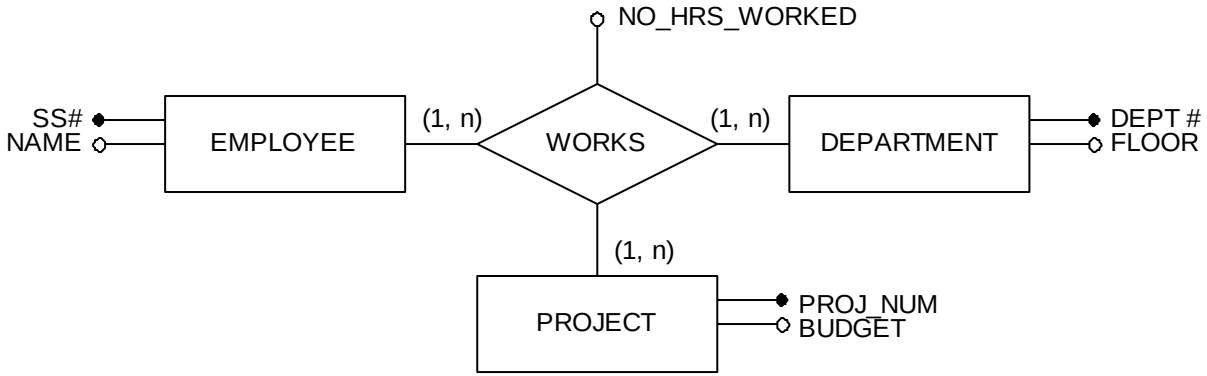






Examples – Normality



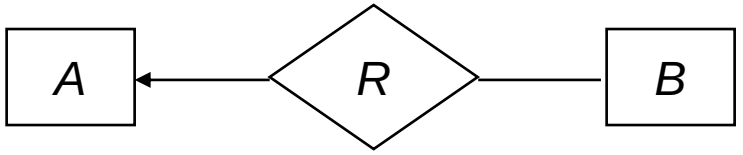


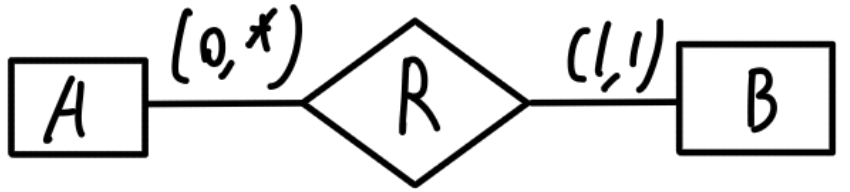
DISCUSSIONS – CHAPTER 6

Discussion 6-1

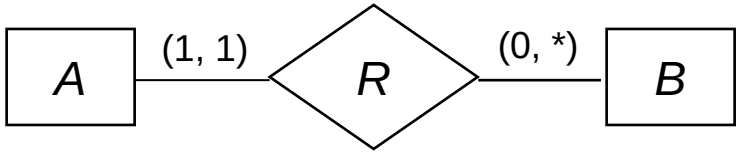
- ERD에서 relationship cardinality를 표현하는 방식으로 화살표와 선의 종류를 이용하는 '기호 방식'과 min과 max값을 구체적으로 표시하는 'min/max 방식' 두 가지를 설명하였다. 아래 각각을 표시된 방식으로 표현하라.

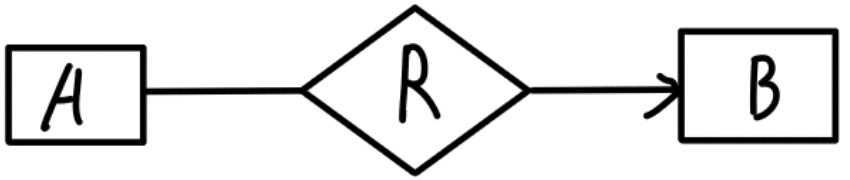
A. Min/Max 방식으로





B. 기호 방식으로





Discussion 6-2

Design an ER schema for the following requirements.

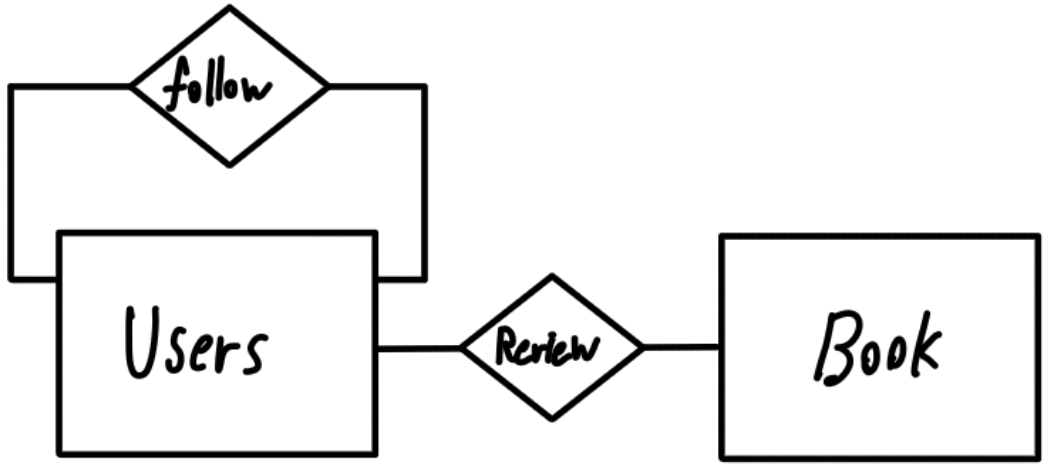
은행 DB: 각 도시마다 지점을 여러 개씩 두고 있으며, 그 중 하나는 그 도시의 대표지점이 된다. 지점마다 영업부, 총무부, 기획부의 조직이 있다.

Discussion 6-3

Design an ER schema for the following requirements.

북클럽(동호회) 정보 관리 데이터베이스

1. 회원정보와 책 정보 관리
2. 각 회원은 책에 대해 선호(별점) 및 서평을 남길 수 있음
3. 회원 간 follow 관계를 맺을 수 있음



Discussion 6-4

아래 요구사항에 대한 ER schema를 설계함에 **있어서 *weak entity set*으로 표현하는 것이 적절할 개념은?** 해당 개념을 ER schema로 표현하시오 (관련 strong entity set과 함께).

은행 DB: 각 도시마다 지점을 여러 개씩 두고 있으며, 그 중 하나는 그 도시의 대표지점이 된다.

지점마다 영업부, 총무부, 기획부의 조직이 있다.

Discussion 6-5

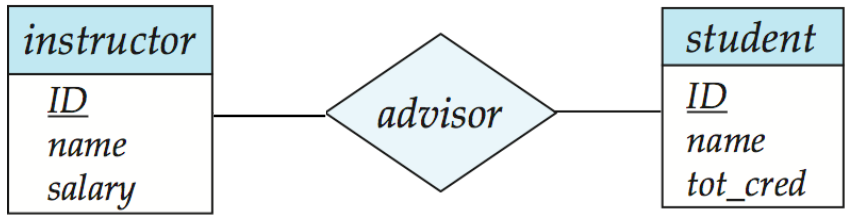
Construct an E-R diagram for the following requirements of a movie database. **Use *generalization/specialization* where appropriate.**

1. Actors star in movies.
2. Directors direct movies.
3. Some actors are also directors.
4. Some movies are sequels of other movies.
5. Some movies are remakes of other movies.
6. Some actors are related to other actors (married, parent, …)
7. Some actors are related to directors (married, parent, …)

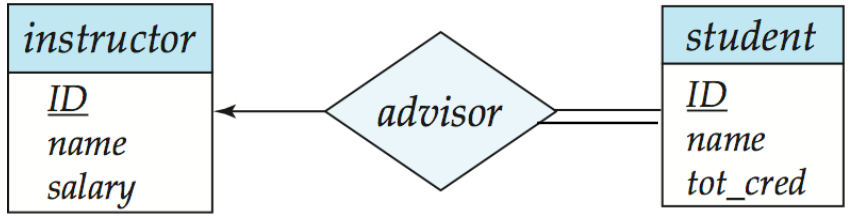
Discussion 6-6

- 아래 ERD에 해당하는 적절한 table들을 생성하라.

A.



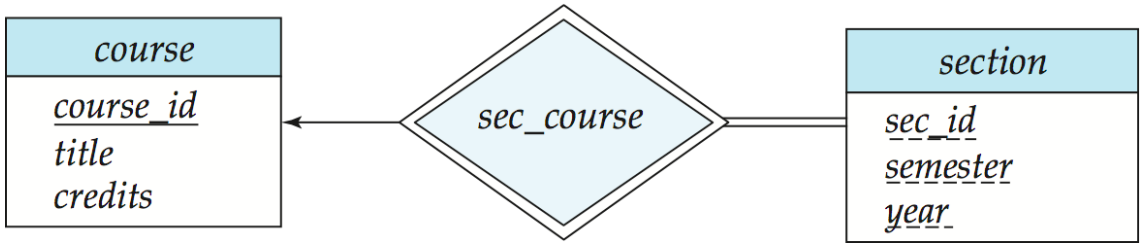
B.



Discussion 6-7

Create tables in SQL for the following ERD with a weak entity set *section*.

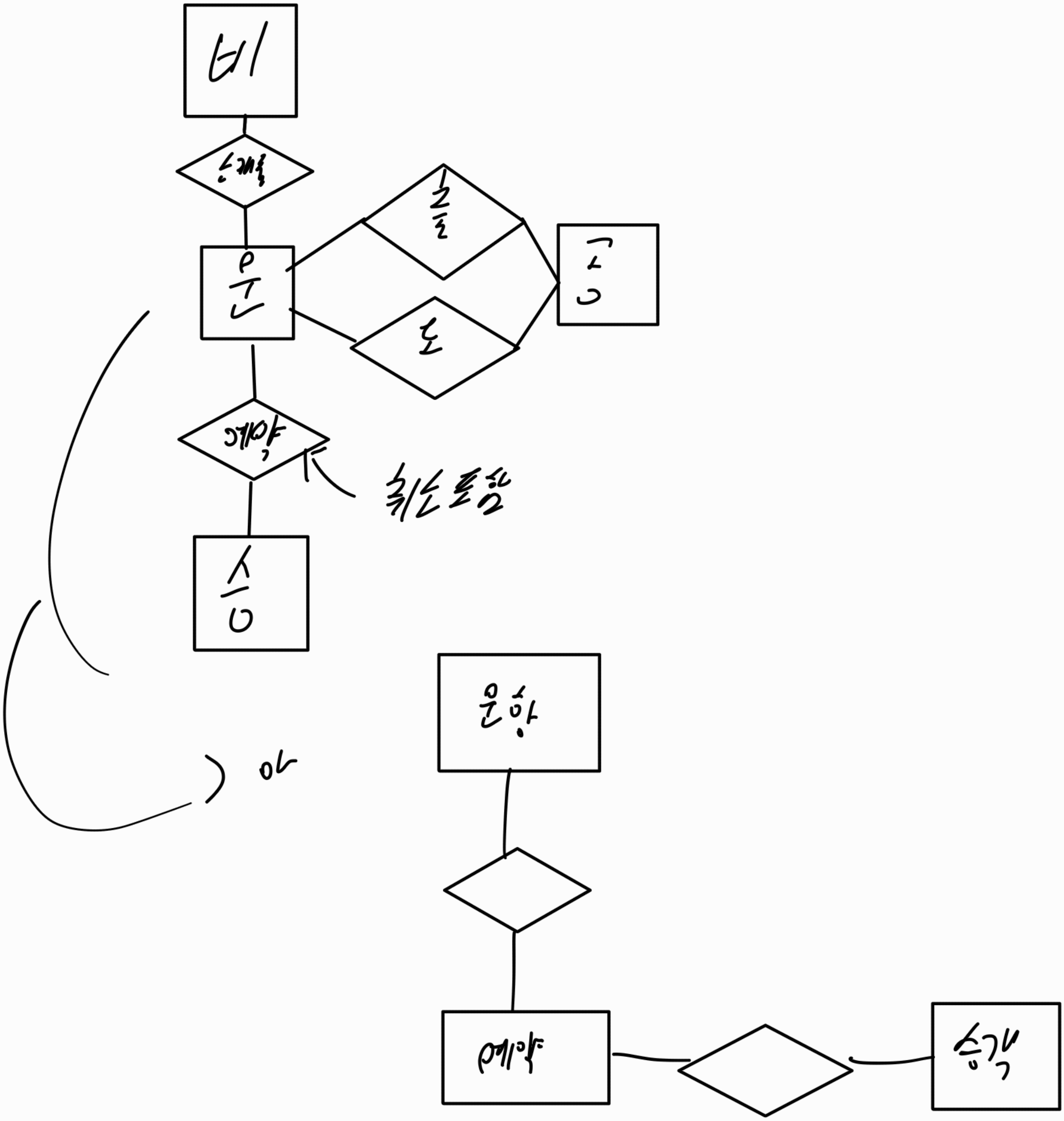
Specify appropriate constraints on the relations so that the *many-to-one* cardinality constraint and *total participation* constraint can be enforced.



Discussion 6-8

Produce a conceptual schema for the following ***airline reservation system****:*

항공사의 예약 데이터베이스는 비행기와 승객 예약에 대한 데이터를 저장한다. 항공사는 보유한 비행기의 기종, 엔진종류, 생산 년도, 좌석 수에 대한 정보를 유지한다. 운항 스케쥴이 잡힌 각 비행기에 대해서는 출발지와 도착지 및 각각의 일시 및 요일 정보를 관리한다. 각 운항 비행기는 중간 기착지 없이 하나의 출발지로부터 하나의 도착지로 가는 것으로 가정한다. 출발지와 도착지는 공항 코드로 구별되어지는데 각각의 공항 이름, 소재 국가, 도시 이름에 대한 정보가 있다. 승객에 대해서는 이름, 성별, 전화번호, 흡연 여부, 마일리지 정보를 관리한다. 승객은 복수 예약을 할 수 있고 또 예약을 취소할 수도 있다. 예약 및 취소 여부, 그 일자와 시각 등은 저장 관리된다.


예약
취소포함
운항
예약
승객

Discussion 6-9

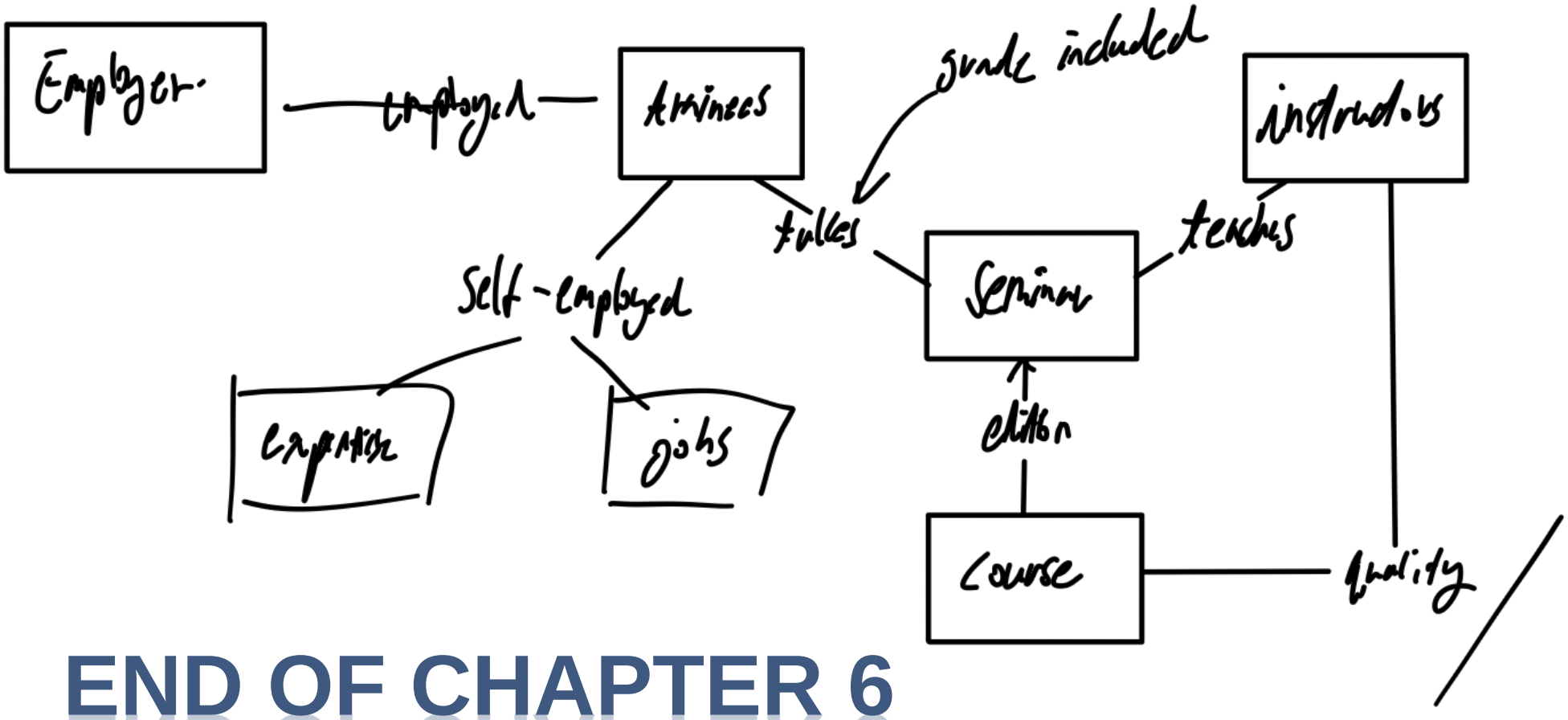
Produce a conceptual schema for the following ***hospital database****:*

The hospital stores data about patients, their admission and discharge from the hospital's departments and their treatments, For each patient, we know the name, address, sex, and ID number. For each department, we know the department's name, its location, the name of the doctor who heads it, the number of beds available, and the number of beds occupied. Each patient gets admitted at a given date and discharged at a given date. Each patient goes through multiple treatments during hospitalization; for each treatment, we store its name, duration, and the possible reactions to it that the patient may have.

Discussion 6-10

*Produce a conceptual schema for the following **training company**:*

We wish to create a database for a company that runs training courses. For this, we must store data about the trainees and the instructors. For each course participant (about 5000), identified by a code, we want to store the social security number, surname, age, sex, place of birth, employer's name, address and telephone number, previous employers (and period employed), the courses attended (there are about 200 courses) and the final assessment of each course. We need also to represent the seminars that each participant is attending at present and, for each day, the places and times the classes are held. Each course has a code and a title and any course can be given any number of times. Each time a particular course is given, we will call it an 'edition' of the course. For each edition, we represent the start date, the end date, and the number of participants. If a trainee is a self-employed professional, we need to know his or her area of expertise, and, if appropriate, his or her title. For somebody who works for a company, we store the level and position held. For each instructor (about 300), we will show the surname, age, place of birth, the edition of the course taught, those taught in the past and the courses that the tutor is qualified to teach. All the instructors' telephone numbers are also stored. An instructor can be permanently employed by the training company or can be freelance.



END OF CHAPTER 6