

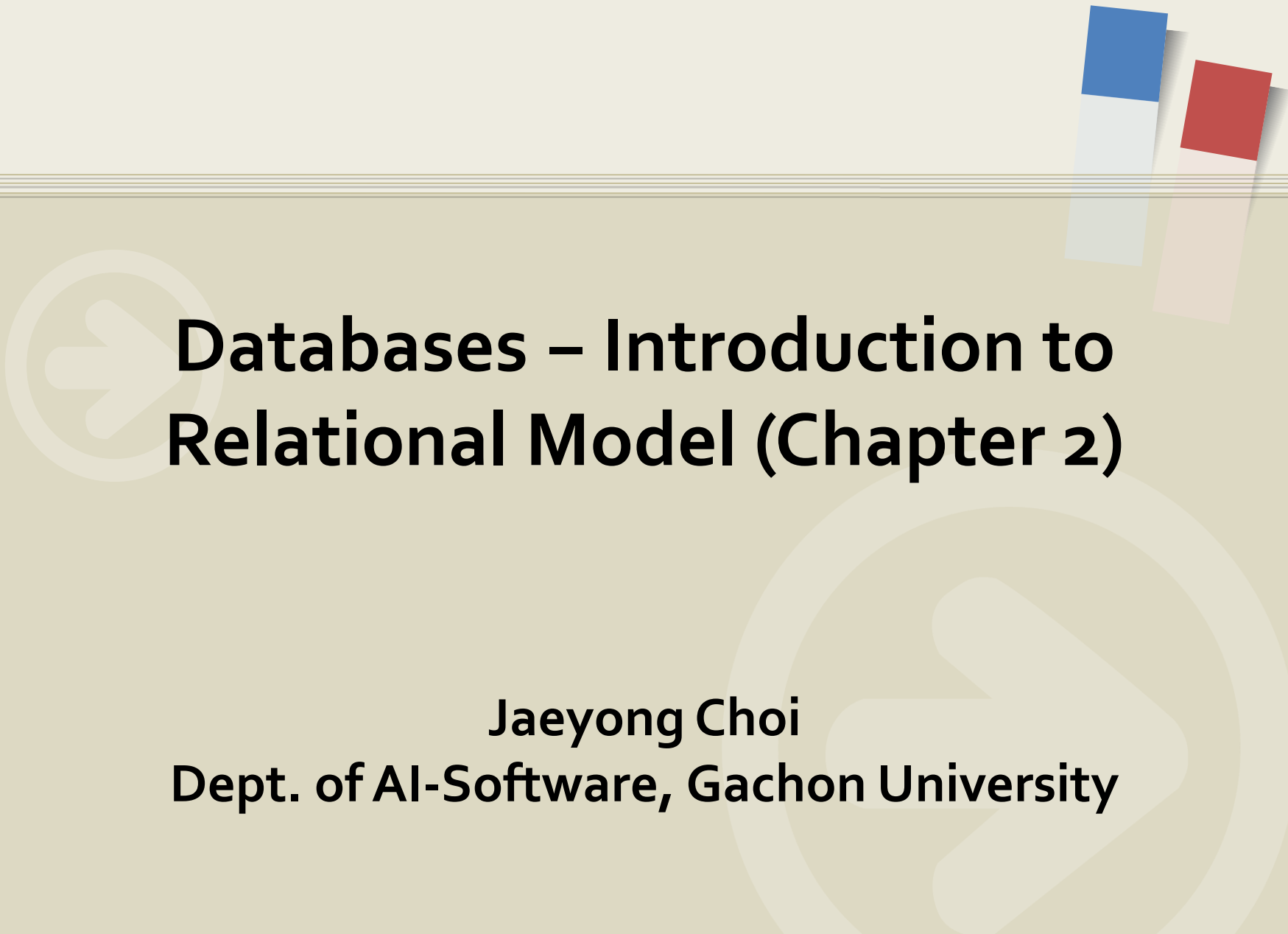
Databases – Database Design & E-R Model (Chapter 6)

Jaeyong Choi
Dept. of Software, Gachon University



❑ Schedule

Week	Topic	Chapter	Note
1	Introduction to DBMS, Relational Model	1	
2	Relational Algebra : - Concept of Key - Relational algebra operators - Relational algebra expressions	2	추석
3	Introduction to SQL	3	
4	Advanced SQL : - Advanced expression of SQL - Nested SQL queries	4, 5	MOOC
5	Entity/Relationship Model	6	
6	Relational Database Design 1 Relational Database Design 2 (추석보강)	7	MOOC
7	Storage and File Structure	12, 13	
8	Mid-term Exam		



Databases – Introduction to Relational Model (Chapter 2)

Jaeyong Choi
Dept. of AI-Software, Gachon University



Equivalent Queries

- ❑ There is more than one way to write a query in relational algebra.
- ❑ Example: Find instructors in the Physics department with salary greater than 90,000

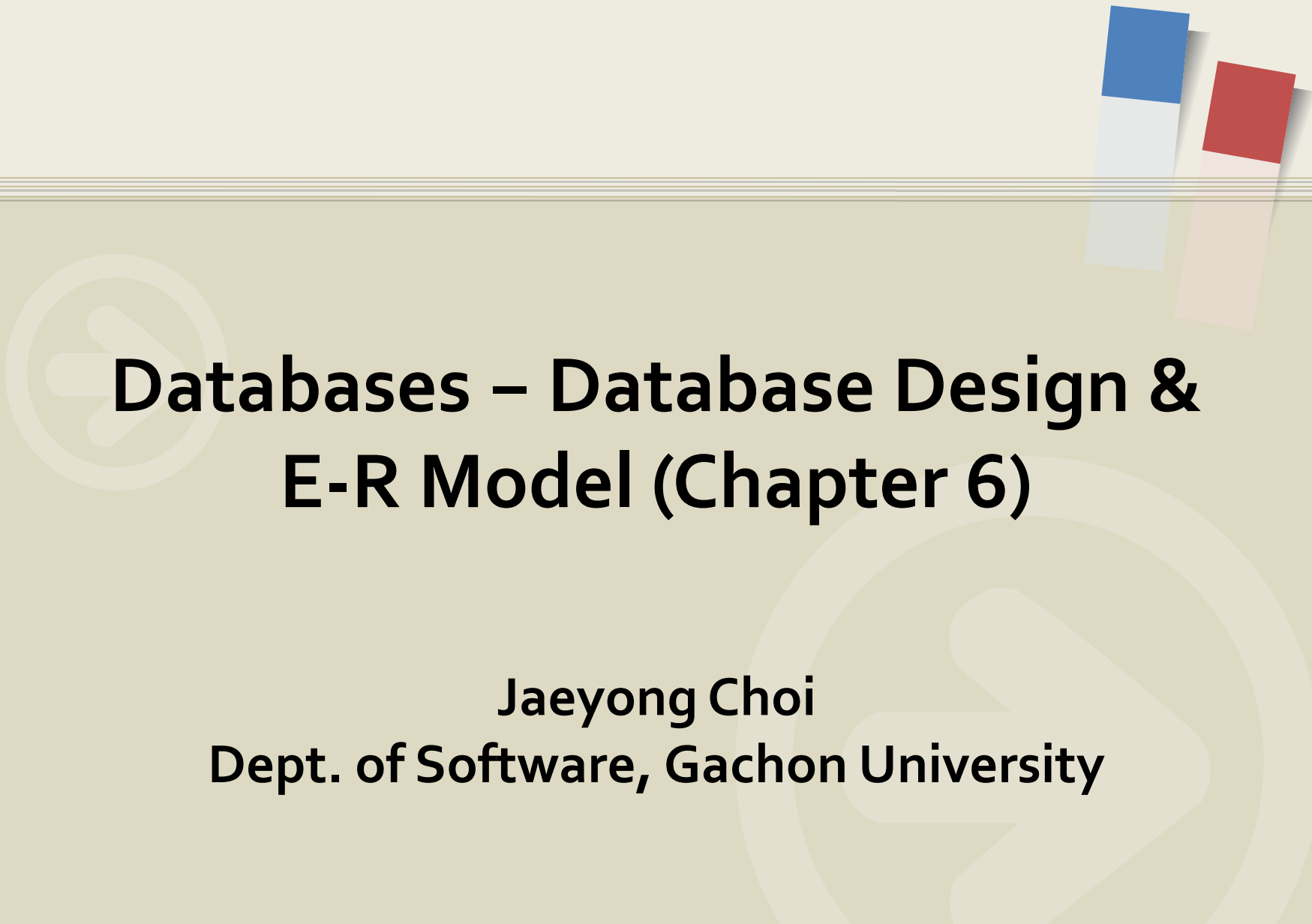
❑ Query 1

$\sigma_{\text{dept_name} = \text{"Physics"} \wedge \text{salary} > 90,000}(\text{instructor})$

❑ Query 2

$\sigma_{\text{dept_name} = \text{"Physics"}}(\sigma_{\text{salary} > 90,000}(\text{instructor}))$

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

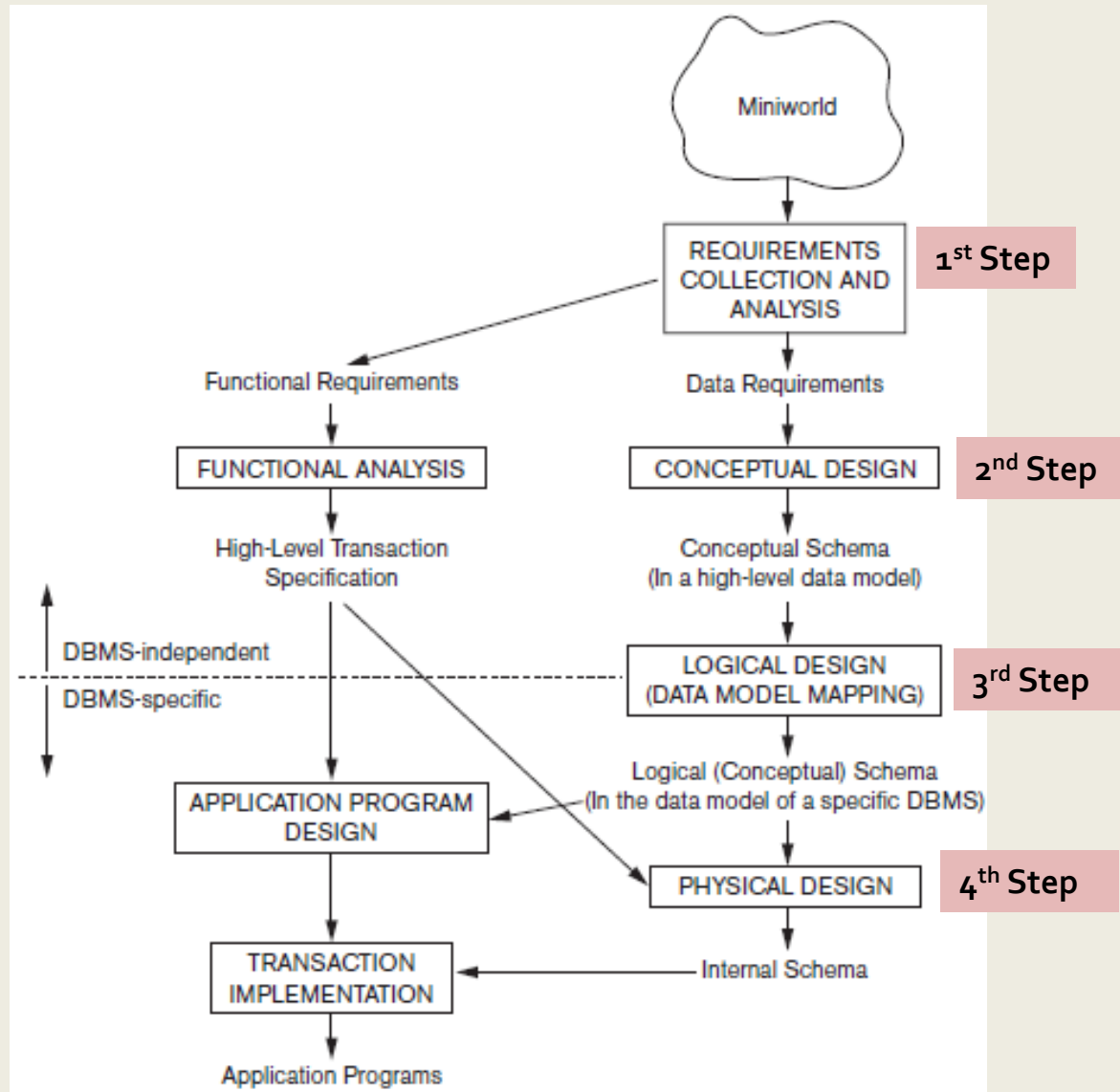


Databases – Database Design & E-R Model (Chapter 6)

Jaeyong Choi
Dept. of Software, Gachon University



Database Design Overview





Overview of Design Process (1/3)

□ Design phases

- Characterize fully the data needs of the prospective database users
 - Outcome → specification of user requirements
- Choose a data model and translate these requirements into a conceptual schema of the database
 - Entity-Relationship model → conceptual design
- *Conceptual schema* also indicates the functional requirements of the enterprise
 - *Specification of functional requirements* describe the kinds of operations (or transactions) that will be performed on the data



Overview of Design Process (2/3)

□ Design phases *cont'd*

- *Logical design* – deciding on the database schema
 - Database design requires a “good” collection of **relation schemas**
 - Business decision – What attributes should we record in the database?
 - Computer science decision – What relation schemas should we have and how should the attributes be distributed among the various relation schemas?
- *Physical design* – deciding on the physical layout of the database

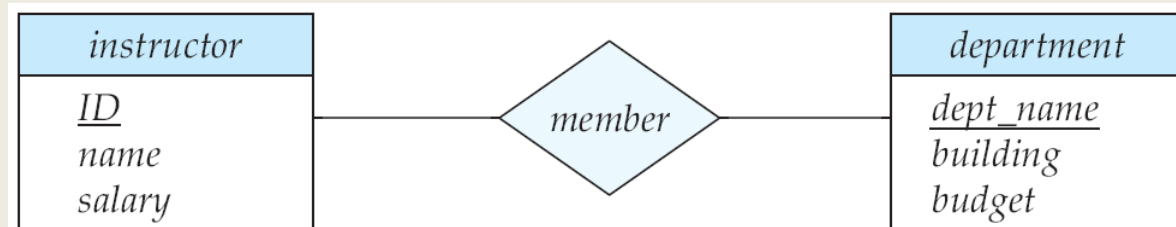
Real World to Computer World (1/3)



Entity

Conceptual schema
E-R model

Database schema



Example of E-R Diagram

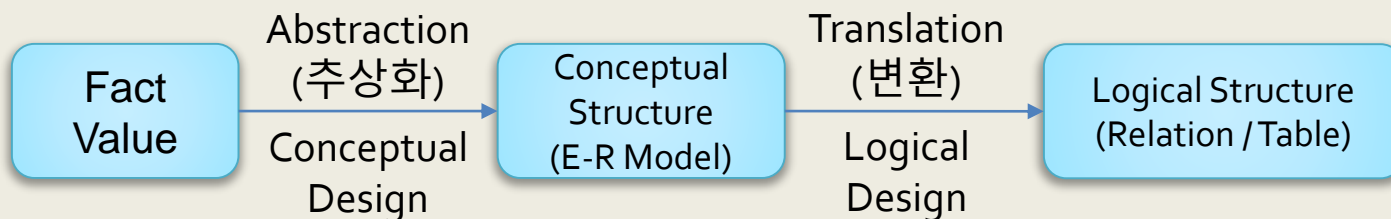
Real World to Computer World (2/3)



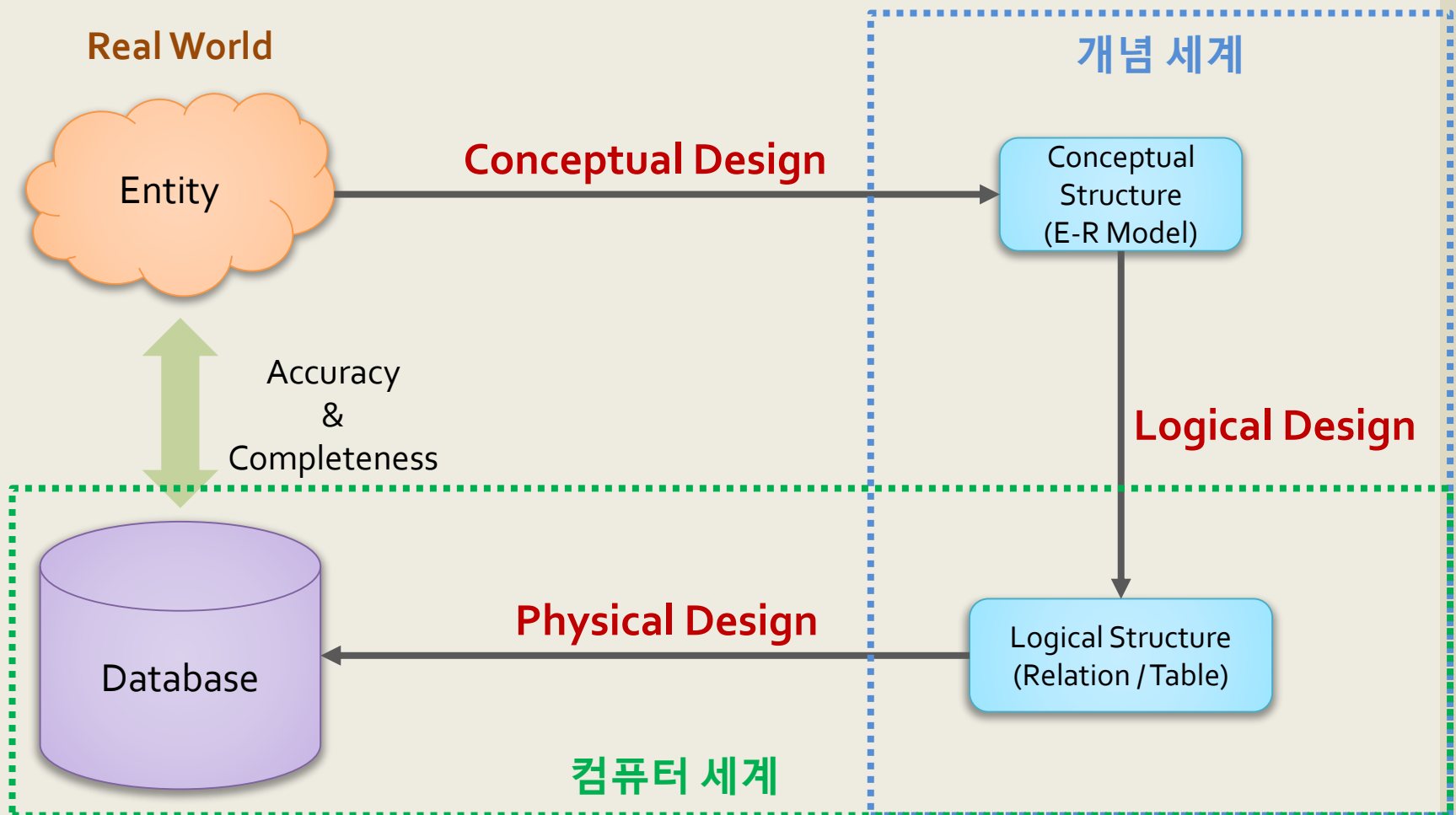
Entity

Conceptual schema
E-R model

Database schema



Real World to Computer World (3/3)



Overview of Design Process (3/3)

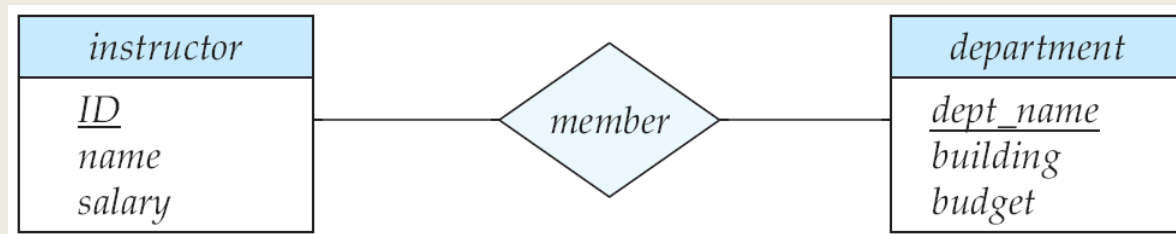
□ Design approaches

■ Entity-relationship Model

- Models an enterprise as a collection of *entities* and *relationships*
 - *Entity*: a “thing” or “object” distinguishable from others; described by a set of attributes
 - *Relationship*: an association among several entities
- Represents diagrammatically by an *entity-relationship diagram*

■ Normalization (Chapter 7)

- Formalizes what designs are bad, and test for them



Example of E-R Diagram



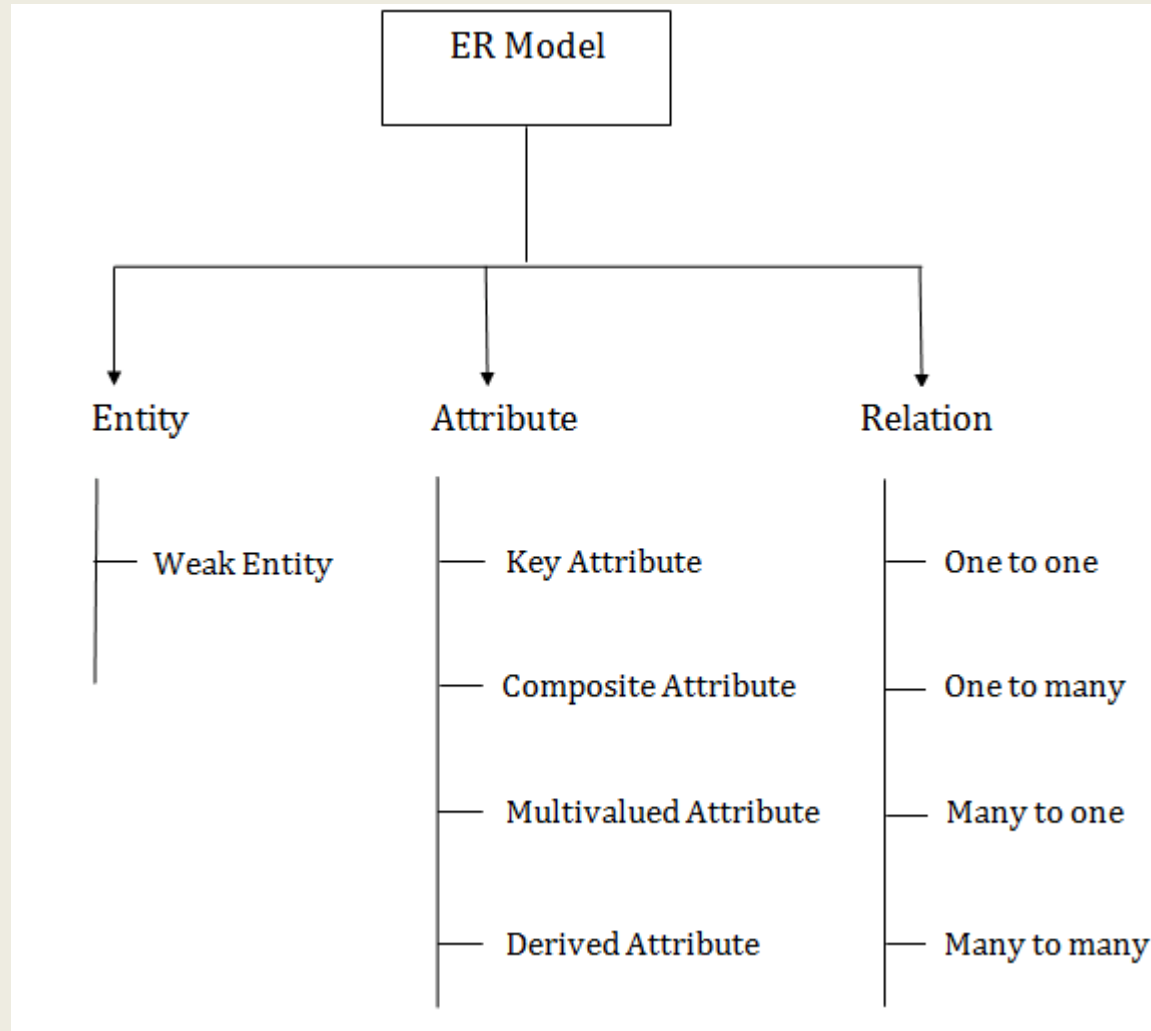
Entity-Relationship Model



❑ E-R model – database modeling

- ❑ Facilitates database design by allowing specification of an *enterprise schema* that represents the overall *logical structure* of a database
- ❑ Very useful in mapping the meanings and interactions of real-world enterprises onto a conceptual schema
- ❑ Employs three basic concepts:
 - ❑ Entity sets
 - ❑ Relationship sets
 - ❑ Attributes
- ❑ Has an associated diagrammatic representation, the *E-R diagram*, which can express the overall logical structure of a database graphically

Three basic concepts of E-R Model





Three basic concepts of E-R Model

□ Entity sets

- An *entity* is an object that exists and is distinguishable from other objects
 - E.g., specific person, company, event, plant
- An *entity set* is a set of entities of the same type that share the same properties
 - E.g., set of all persons, companies, trees, holidays
- An entity is represented by a set of *attributes*; i.e., descriptive properties of all members of an entity set
 - E.g., instructor = (ID, name, street, city, salary)
- A subset of attributes form a *primary key* of the entity set; i.e., uniquely identifying each member of the set

Three basic concepts of E-R Model

Entity sets: *instructor* and *student*

<i>instructor</i>	
<u>ID</u>	name

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

instructor

<i>student</i>	
<u>ID</u>	name

98988	Tanaka
12345	Shankar
00128	Zhang
76543	Brown
76653	Aoi
23121	Chavez
44553	Peltier

student

Three basic concepts of E-R Model

Relationship sets

- A *relationship* is an association among several entities

- E.g.,



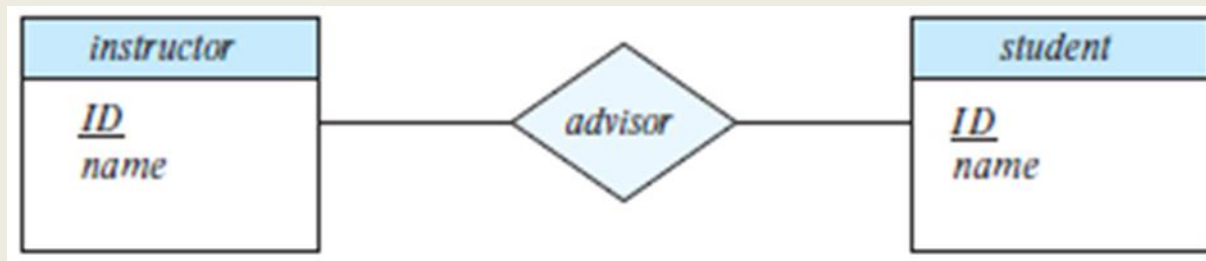
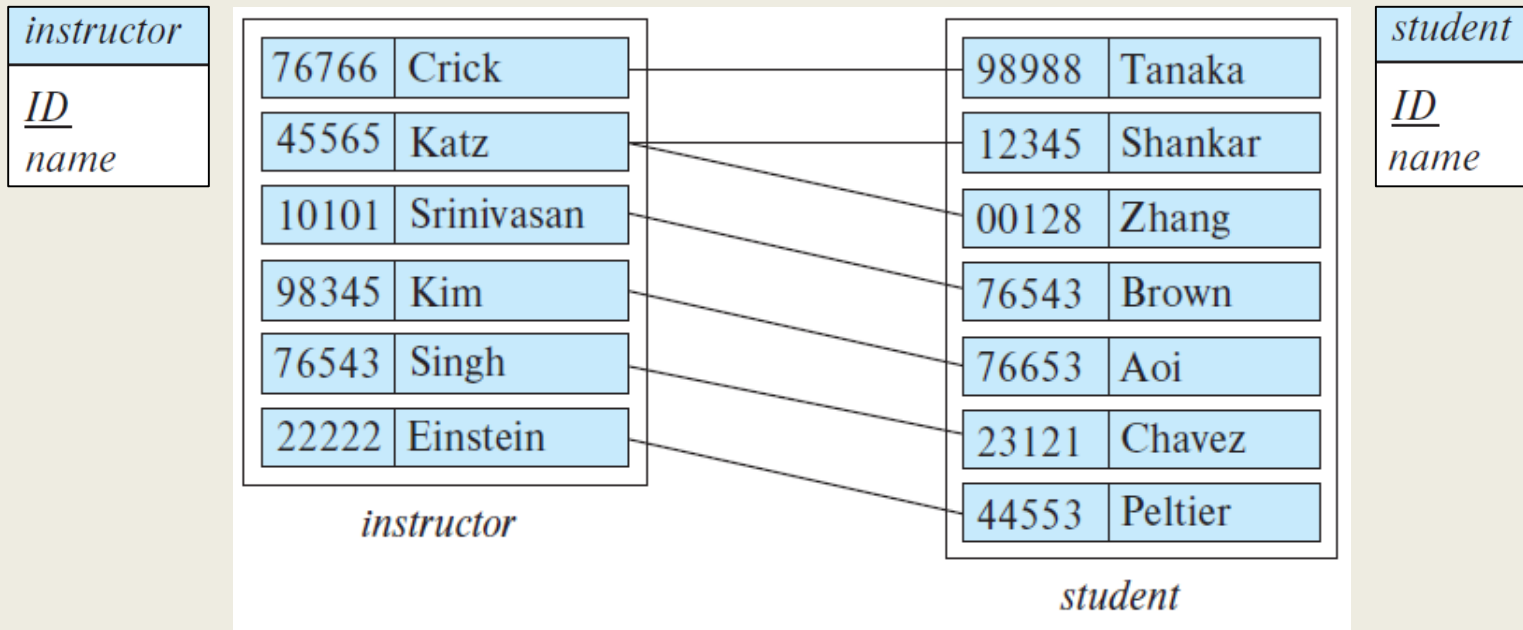
- A *relationship set* is a mathematical relation among $n \geq 2$ entities, each taken from entity sets

$$\{(e_1, e_2, \dots, e_n) \mid e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n\}$$

- E.g., $(44553, 22222) \in \text{advisor}$

Three basic concepts of E-R Model

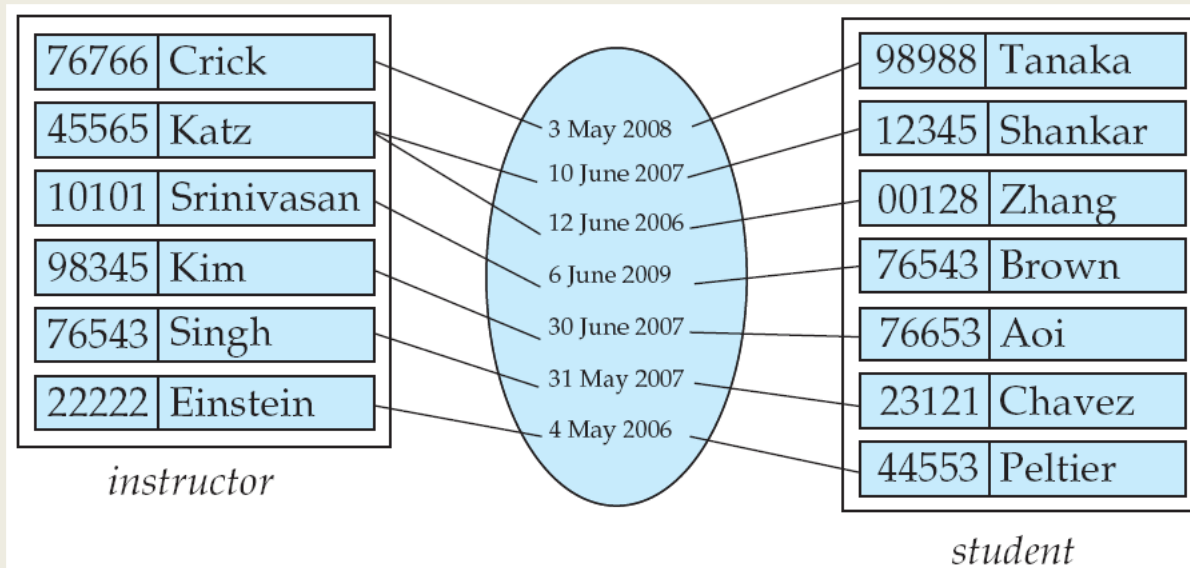
❑ Relationship set: *advisor*



Three basic concepts of E-R Model

❑ Relationship sets (*cont'd*)

- An attribute can also be associated with a relationship set
- E.g., *advisor* relationship set may have an attribute *date* when the student was associated with the advisor

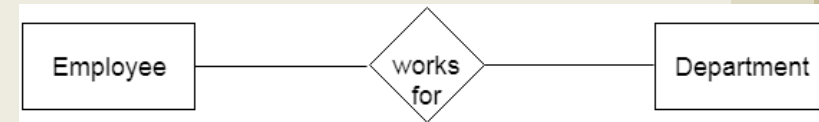


Three basic concepts of E-R Model

❑ Degree of a relationship set

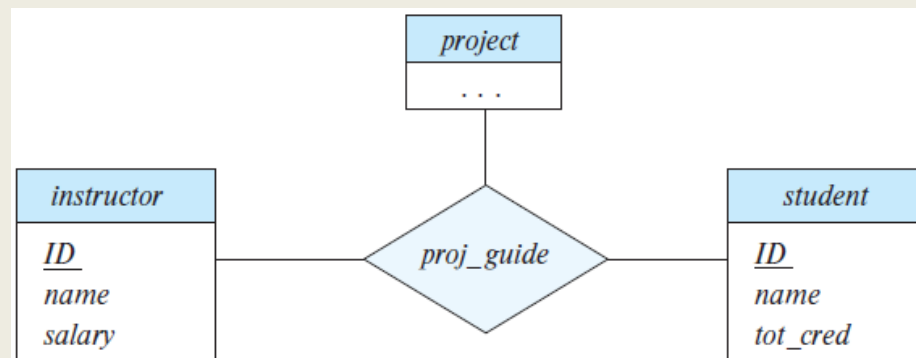
❑ Binary relationship

- ❑ Involves two entity sets (or degree two)
- ❑ Most relationship sets in a database system are binary



❑ Relationships between more than two entity sets are rare

- ❑ E.g., students work on research projects under the guidance of an instructor
- ❑ Relationship *proj_guide* is a ternary relationship between instructor, student, and project

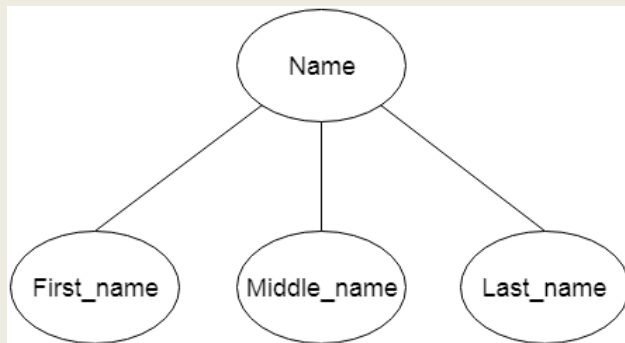


Three basic concepts of E-R Model

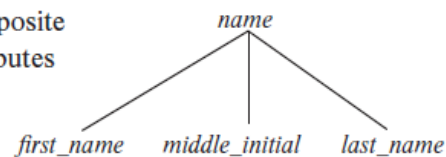
Attribute type 1 :

Simple and composite attributes

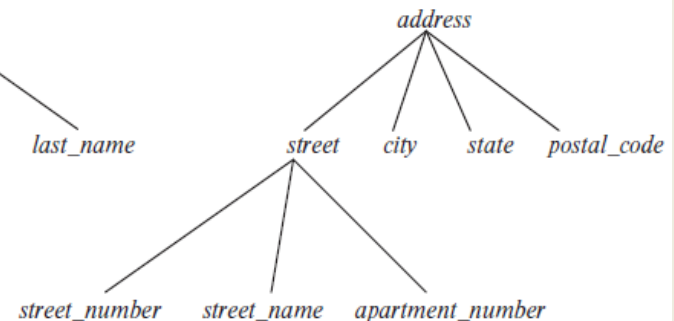
- Simple : have not been divided into subparts
- Composite : can be divided into subparts



composite
attributes



component
attributes



composite attributes

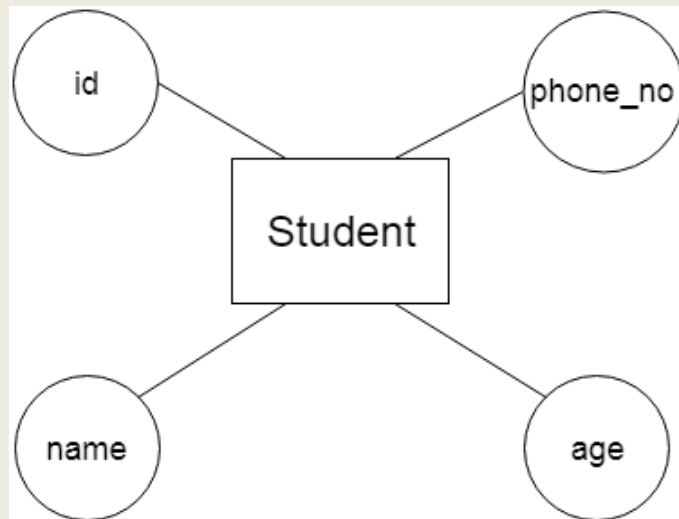
Three basic concepts of E-R Model

□ Attribute type 2:

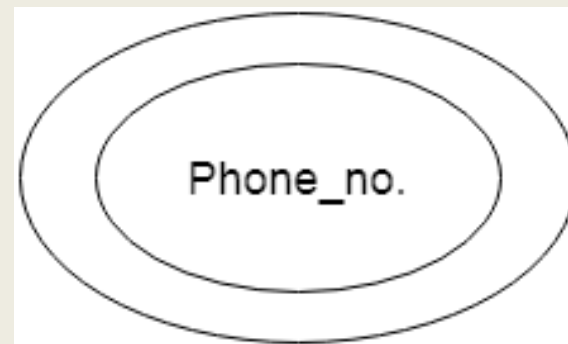
□ *Single valued* and *multivalued* attributes

□ E.g., single valued attribute: *student_ID*

□ E.g., multivalued attribute: *phone_numbers*



Single valued attributes



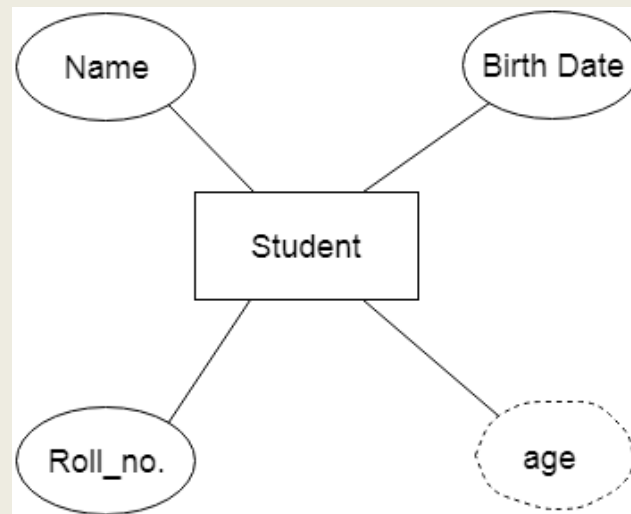
multivalued attributes

Three basic concepts of E-R Model

□ Attribute type 3:

□ *Derived* attributes

- Can be computed from other attributes
- E.g., *age*, given *date_of_birth*
 - *Derived attribute = age*
 - *Base / Stored attribute = date_of_birth*



Derived attributes



❑ Domain:

- ❑ A set of possible values for an attribute (the type of the attribute)
- ❑ Examples
 - ❑ The domain of customer name might be strings of some fixed length
 - ❑ The domain of social security number might be 9 digit positive integers
 - SSN : XXX – XX – XXXX



Constraints



❑ Mapping cardinality constraints

- ❑ Express the number of entities to which another entity can be associated via a relationship set
- ❑ Most useful in describing binary relationship sets
- ❑ For a binary relationship set, the mapping cardinality must be one of the following types:
 - ❑ One-to-one
 - ❑ One-to-many, many-to-one
 - ❑ Many-to-many



❑ Mapping cardinalities

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

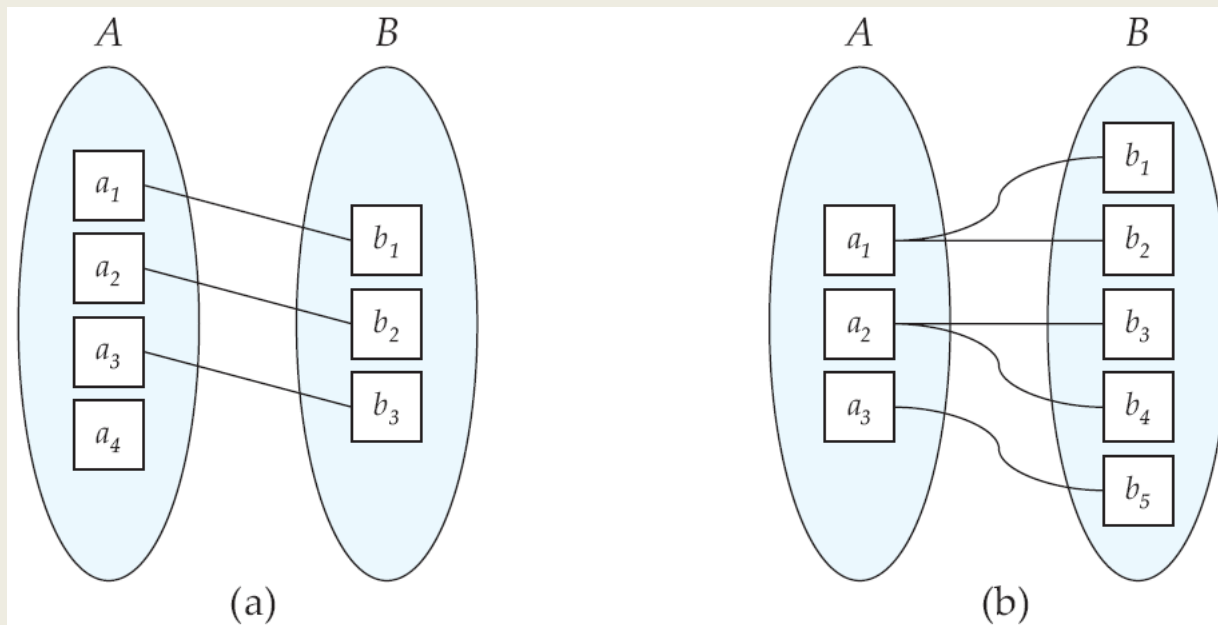


Figure 7.5 Mapping cardinalities. (a) One-to-one. (b) One-to-many.



❑ Mapping cardinalities *cont'd*

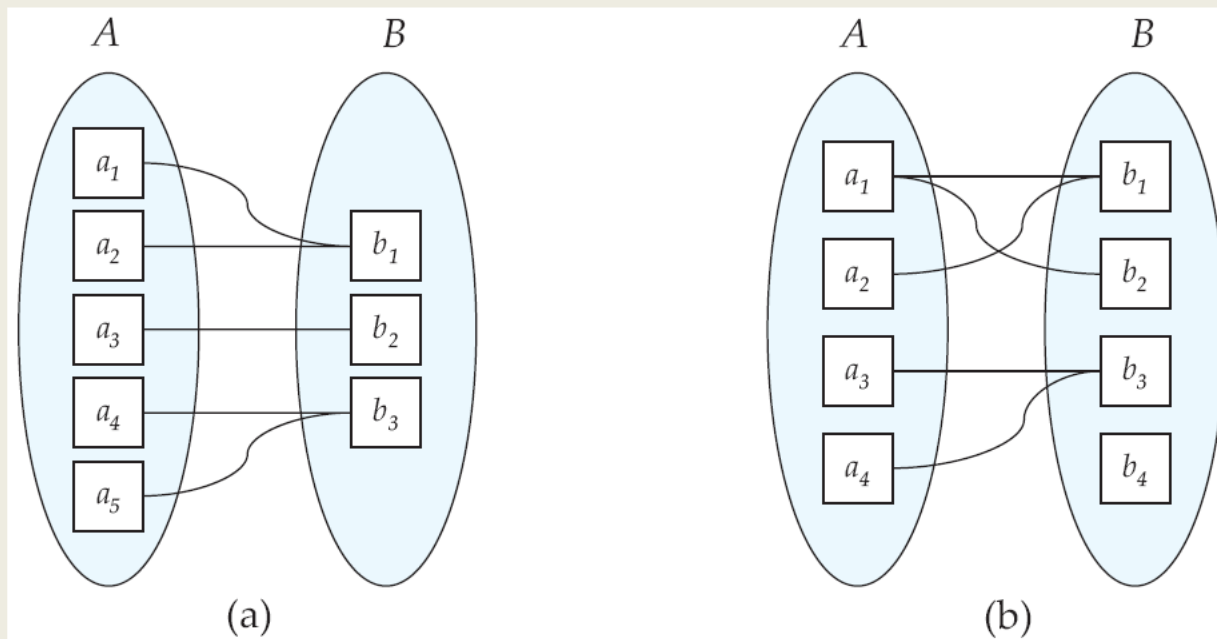


Figure 7.6 Mapping cardinalities. (a) Many-to-one. (b) Many-to-many.



Entity-Relationship Diagrams

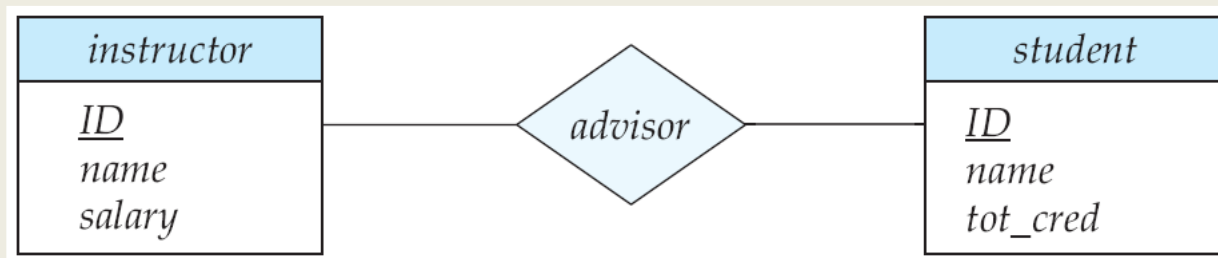
Basic structure

Entities are represented graphically as follows:

- Rectangles represent entity sets
- Attributes listed inside entity rectangle
- Underline indicates primary key attributes

Relationship sets

- Diamonds represent relationship sets

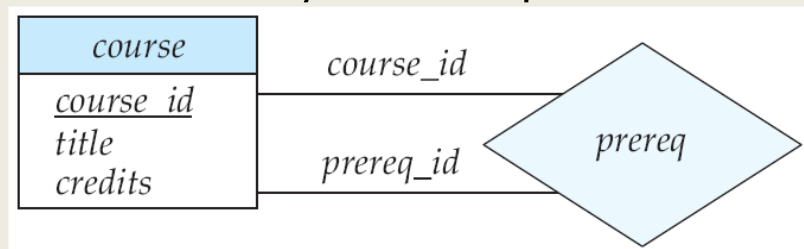




Entity-Relationship Diagrams

❑ Roles : function that an entity plays in a relationship

- ❑ Since entity sets participating in a relationship set are generally distinct, roles are implicit & not usually specified.
- ❑ What if the meaning of a relationship needs clarification?
- ❑ The same entity set participates in a relationship set more than once, in different roles.
- ❑ Example
 - ❑ Roles are represented with the labels “*course_id*” and “*prereq_id*”
 - ❑ One course (C2) is a prerequisite for another course (C1) we have relationship set *prereq* that is modeled by ordered pairs of course entities. (C1, C2) pairs





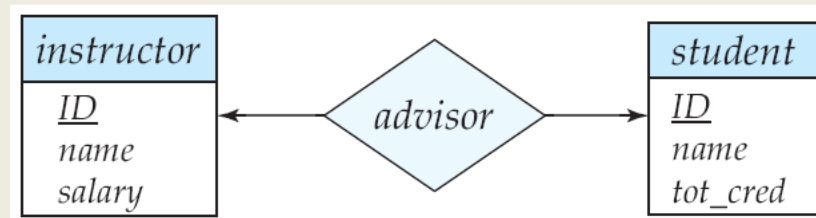
Entity-Relationship Diagrams

❑ Cardinality constraints

- Expressed by drawing either a directed line (\rightarrow), signifying “one”, or an undirected line ($—$), signifying “many”, between the relationship set and the entity set

❑ One-to-one relationship

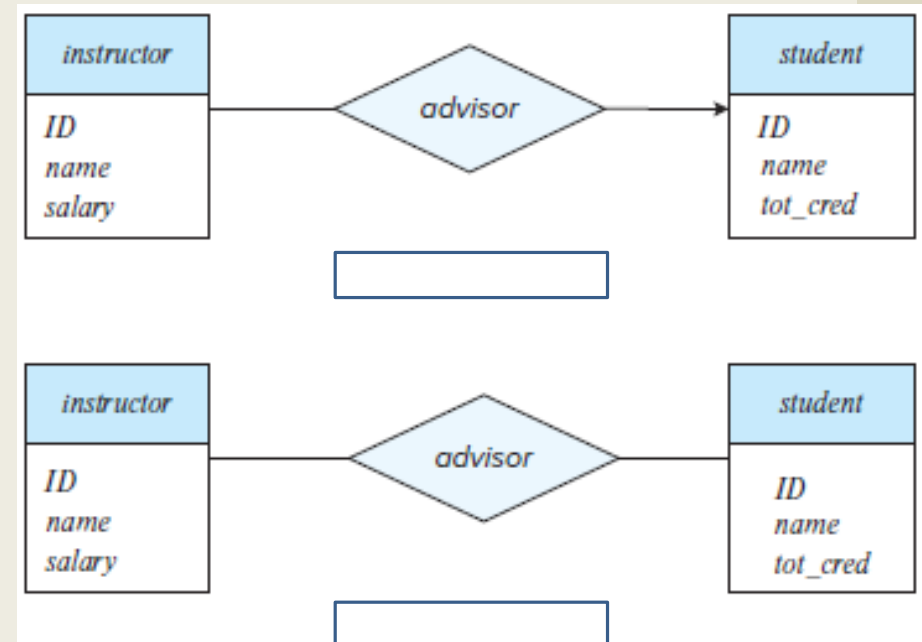
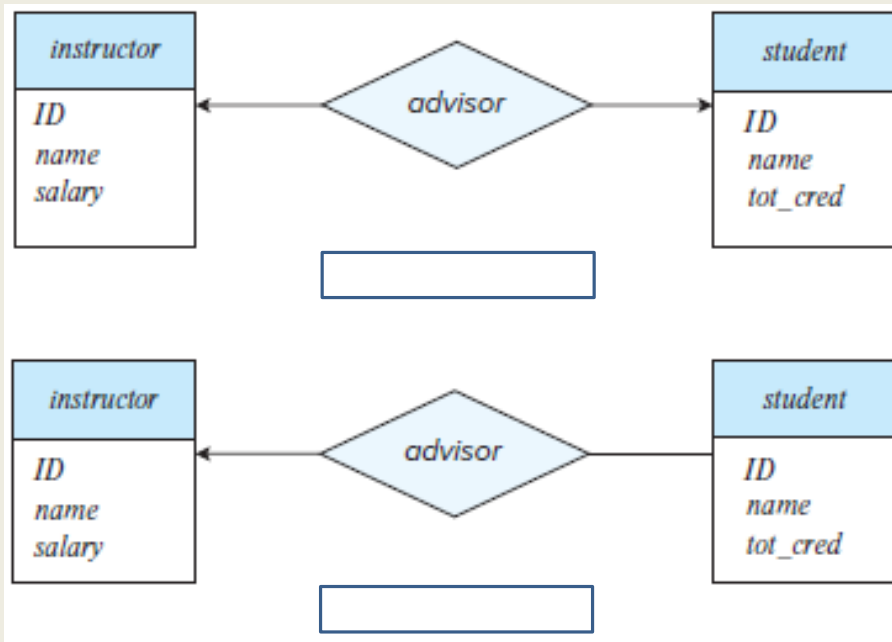
- A *student* is associated with at most one *instructor* via the relationship *advisor*





Entity-Relationship Diagrams

Exercise

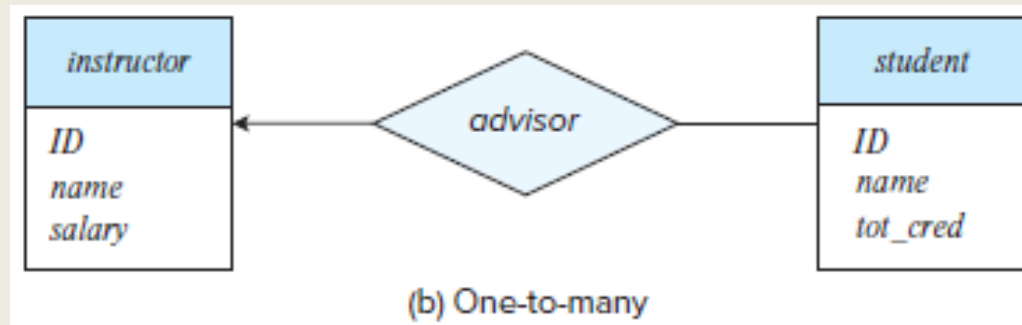




Entity-Relationship Diagrams

❑ One-to-many relationship

- An *instructor* is associated with several (possibly 0) *students* via *advisor*
- A *student* is associated with at most one *instructor*

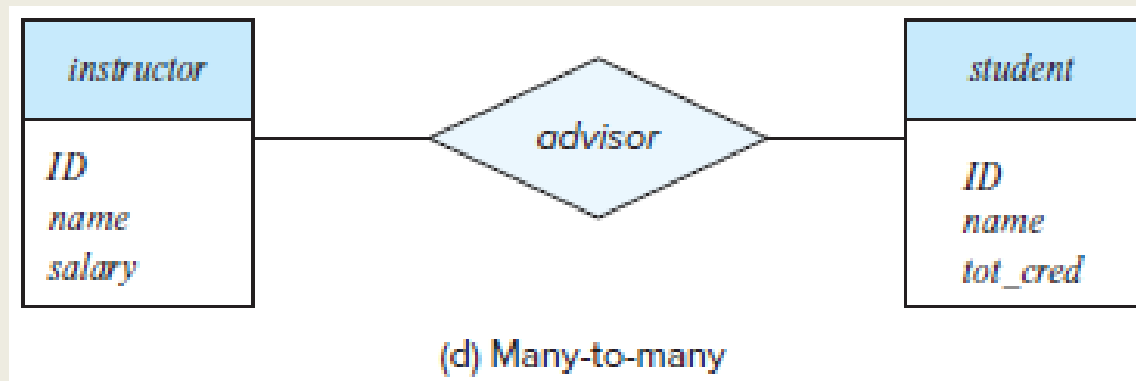




Entity-Relationship Diagrams

❑ Many-to-many relationship

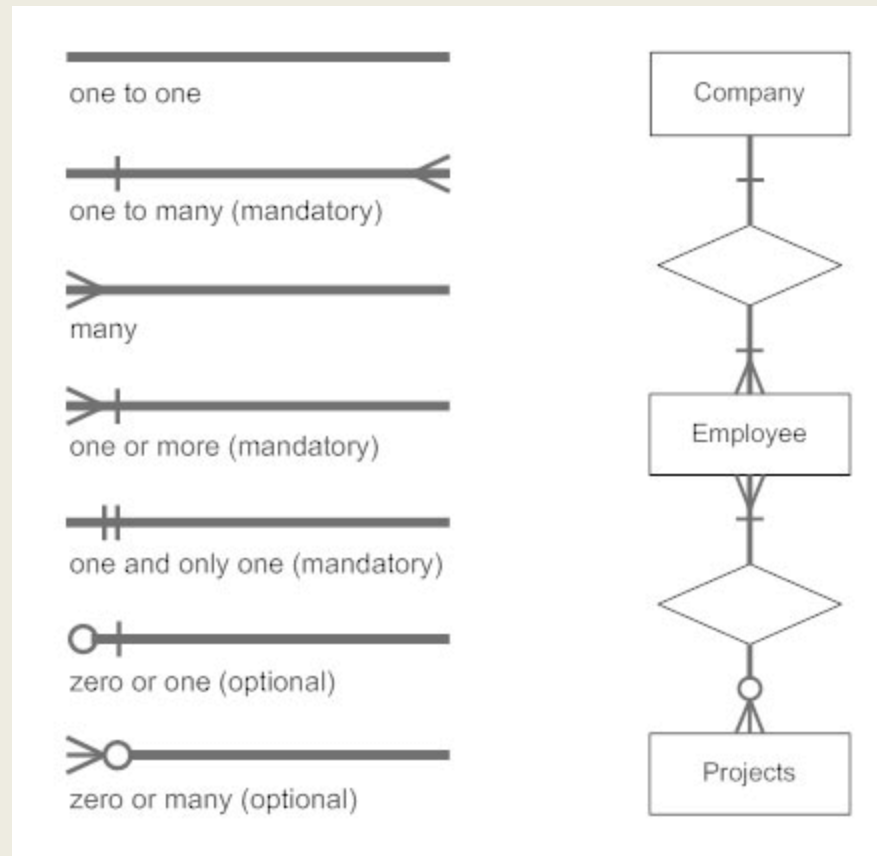
- ❑ An *instructor* is associated with several (possibly 0) *students* via *advisor*
- ❑ A *student* is associated with several (possibly 0) *instructors* via *advisor*





Entity-Relationship Diagrams

- ❑ In ER diagram, many notations are used to express the cardinality.

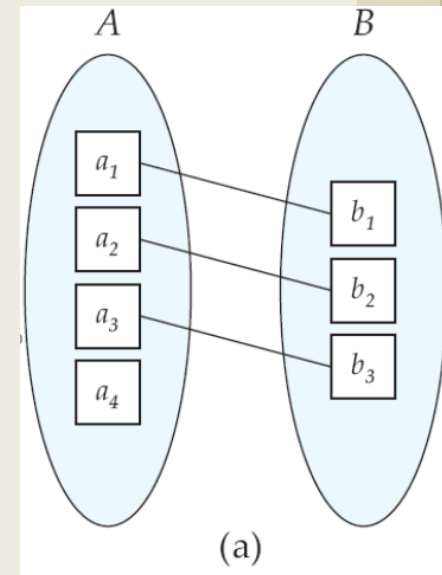
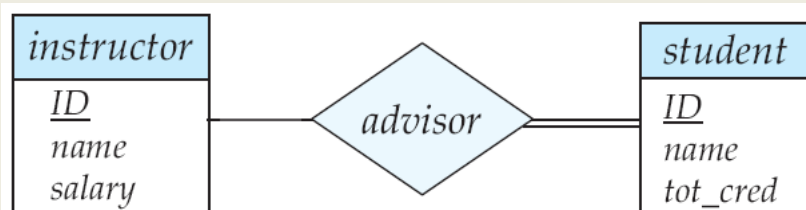




Entity-Relationship Diagrams

❑ Total and partial participation

- ❑ *Total participation* (indicated by a double line)
 - ❑ Every entity in the entity set participates in at least one relationship in the relationship set
 - ❑ E.g., **participation of student in advisor relation is total**; every student must have an associated instructor
- ❑ *Partial participation*
 - ❑ Some entities may not participate in any relationship in the relationship set
 - ❑ E.g., participation of instructor in advisor is partial

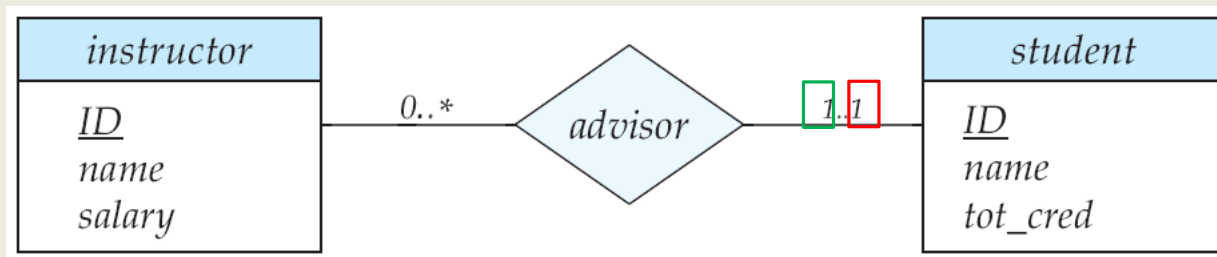




Entity-Relationship Diagrams

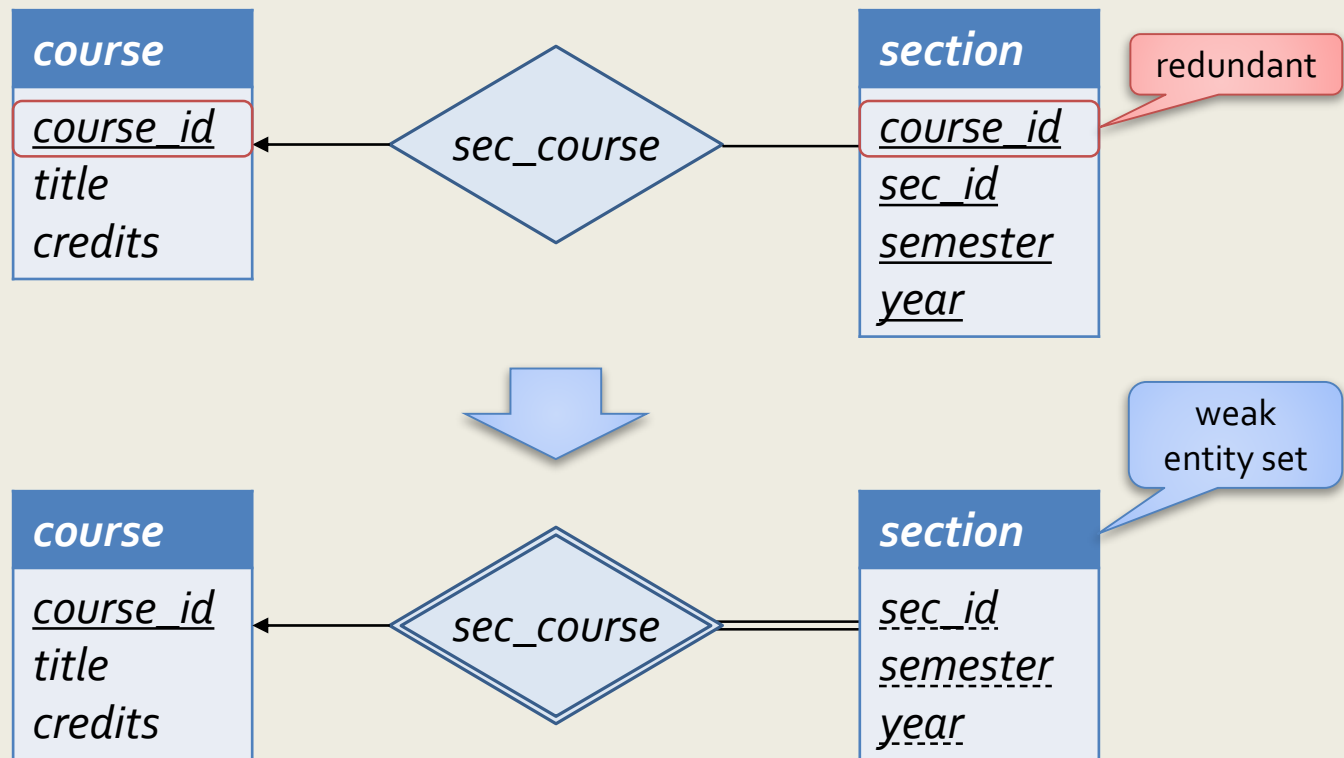
Expressing more complex constraints

- A line may have an associated minimum and maximum cardinality, shown in the form $l..h$, where l is the minimum and h the maximum cardinality
 - A minimum value of 1 indicates total participation
 - A maximum value of 1 indicates that the entity participates in at most one relationship
 - A maximum value of * indicates no limit





❑ Weak entity set





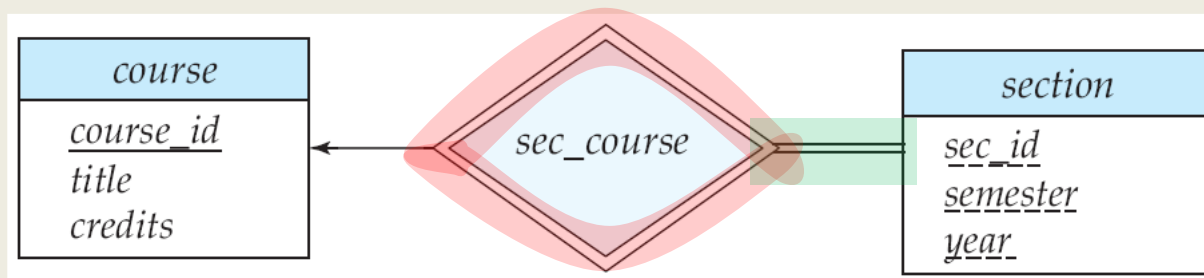
❑ Weak entity set *cont'd*

- ❑ An entity set that does not have sufficient attributes to form a primary key
 - ❑ cf. *strong entity set*: an entity set that has a primary key
 - ❑ The *discriminator* of a weak entity set is a set of attributes that make distinction among all weak entities
- ❑ It must be associated with another (strong) entity set, called the *identifying entity set*
 - ❑ Every weak entity must be associated with an identifying entity
 - ❑ The weak entity set is said to be *existence dependent* on the identifying entity set
 - ❑ The relationship associating the weak entity set with the identifying entity set is called the *identifying relationship*



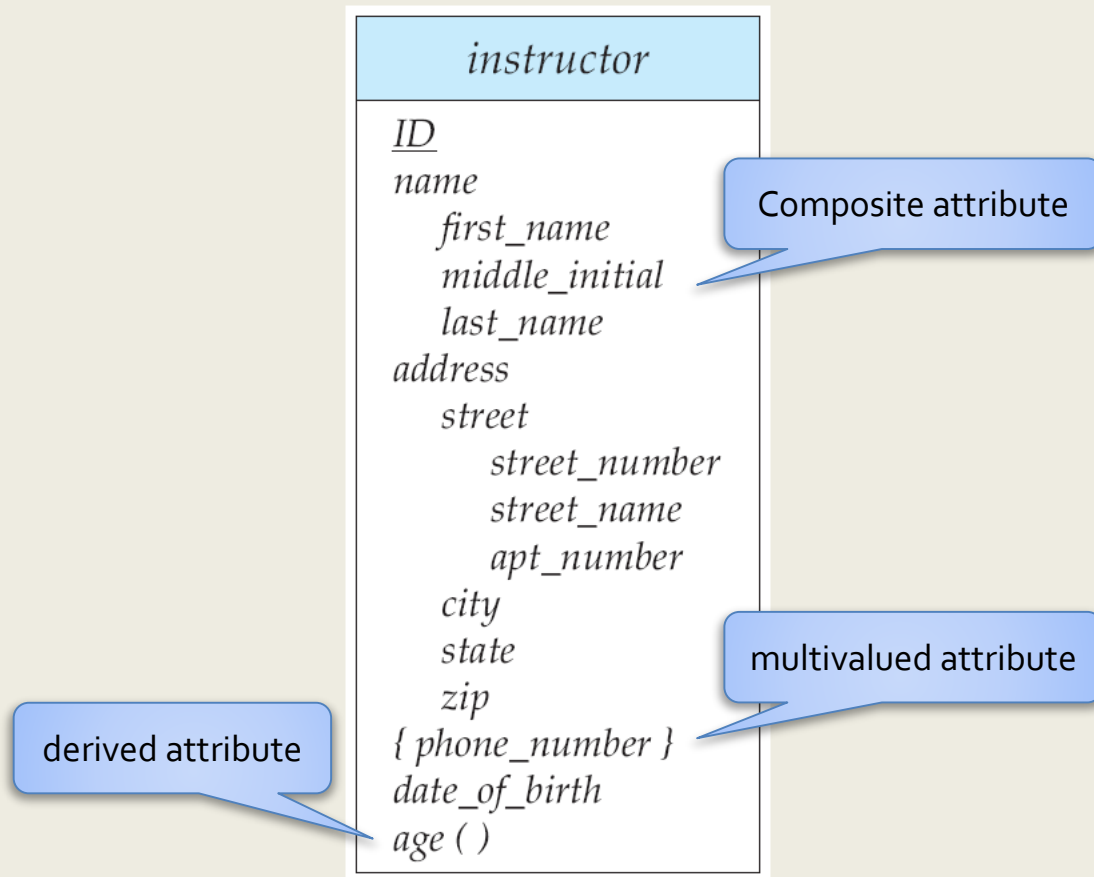
❑ Expressing weak entity sets

- ❑ We underline the discriminator of a weak entity set with a dashed line
- ❑ The relationship set connecting the weak entity set to the identifying entity set is depicted by a double diamond
- ❑ Primary key of a weak entity set is: primary key of identifying entity set + discriminator of weak entity set
 - ❑ E.g., *section* – (course_id, sec_id, semester, year)





❑ Entity with complex attributes



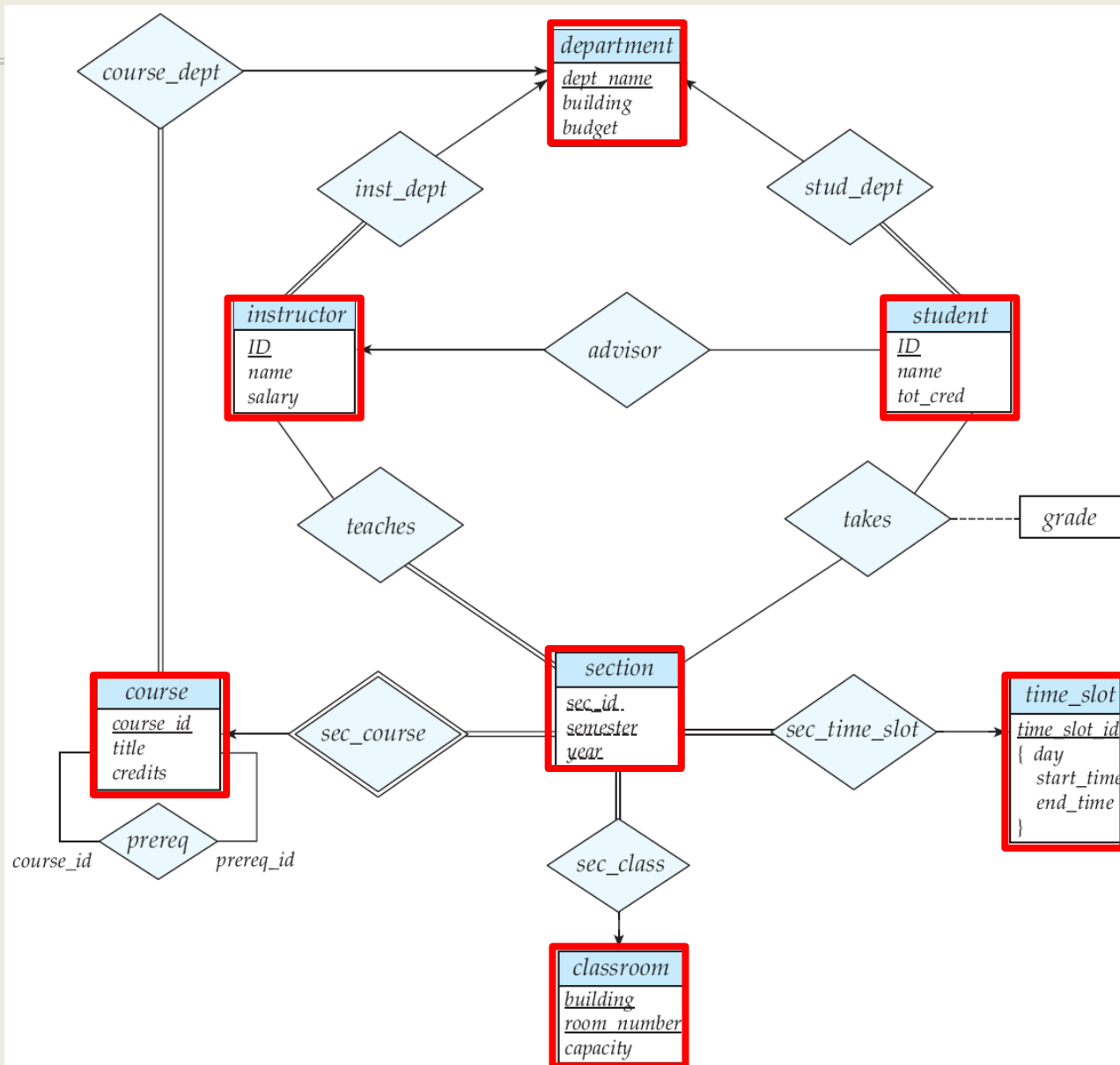


Let's design university enterprise

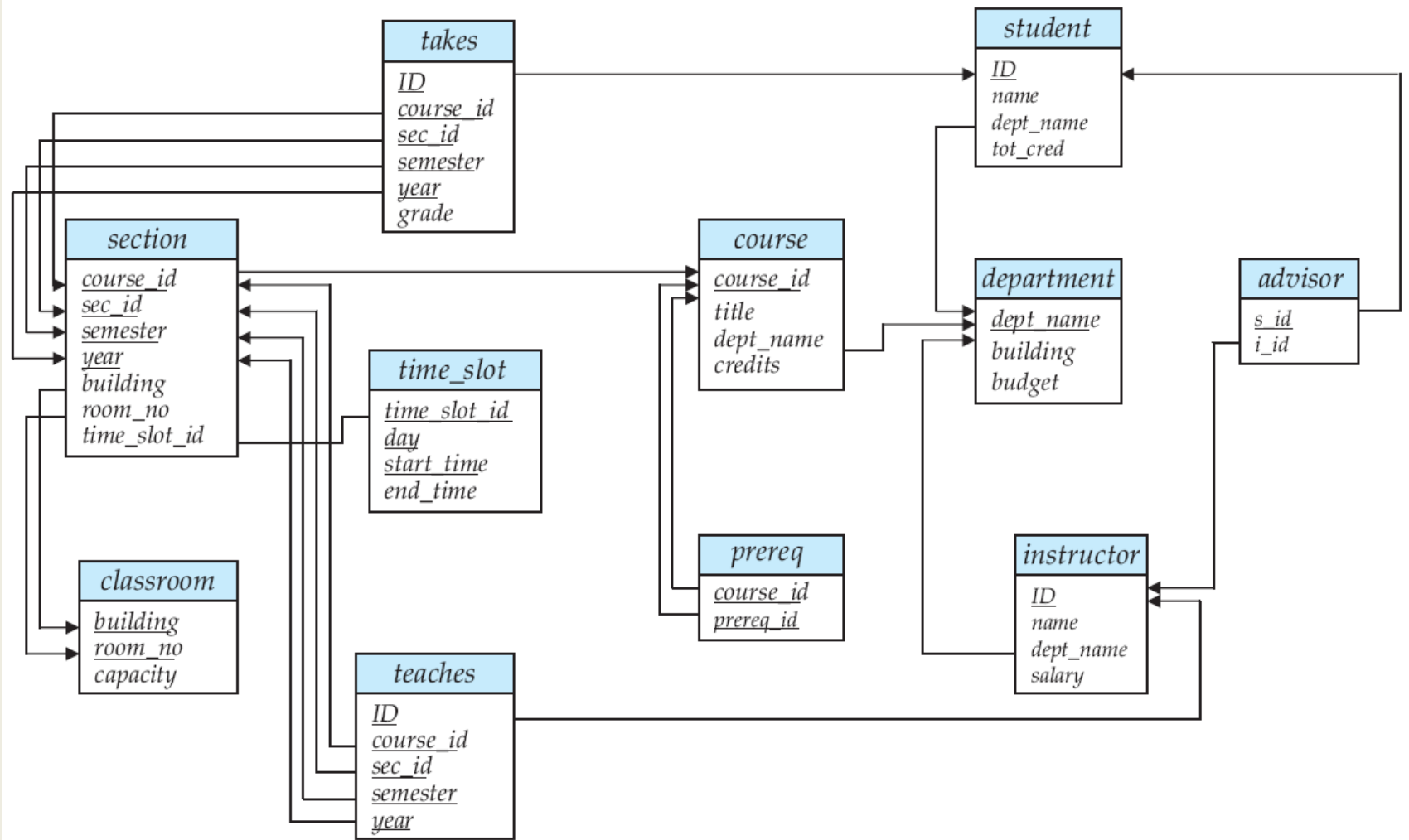
□ How to design a database using the E-R model

- We usually start by identifying those **entity sets** that should be included
 - Entity sets as *student*, *instructor*, etc.
- We must choose the appropriate **attributes**.
 - The entity set *instructor* includes the attributes *ID*, *name*, *dept_name*, *salary*, *phone number*, *office number*, *home page*, etc. with *ID* forming the primary key.
 - The entity set *department* includes the attributes *dept_name*, *building*, and *budget*, with *dept_name* forming the primary key.
- The **relationship** sets among the various entities are formed
 - The attribute *dept_name* appears in both entity sets. Since it is the primary key for the entity set *department*, it is redundant in the entity set *instructor* and needs to be removed.

E-R diagram for a university enterprise



Data schema for a university enterprise





Let's make Relational Schemas using ER model



❑ Reduction to relation schemas

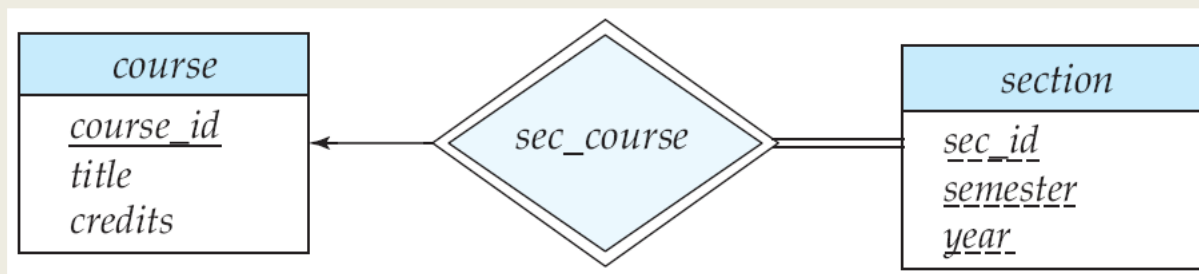
(ER 모델을 Relation Schema로 만드는 방법)

- A database which conforms to an E-R diagram can be represented by a collection of relation schemas
- For each entity set and relationship set, there is a **unique schema** that is assigned the name of the corresponding entity set or relationship set
- **Each schema** has a number of columns (generally corresponding to attributes), which **have unique names**



❑ Representing entity sets

- ❑ A **strong entity set** reduces to a schema with the same attributes
 - ❑ E.g., *student* (ID, name, tot_cred)
- ❑ A **weak entity set** becomes a table that includes a column for the primary key of the identifying strong entity set
 - ❑ E.g., *section* (course_id, sec_id, semester, year)





❑ Entity sets with **composite attributes**

- ❑ Composite attributes are flattened out by creating a separate attribute for each component attribute

- ❑ E.g., the schema corresponding to *instructor* has two attributes *name_first_name* and *name_last_name*
- ❑ Prefix omitted if no ambiguity; i.e., *name_first_name* could be *first_name*

- ❑ Extended *instructor* schema is:

- ❑ *instructor* (ID, first_name, middle_initial, last_name, street_number, street_name, apt_number, city, state, zip, date_of_birth)

<i>instructor</i>	
<u>ID</u>	
name	
first_name	
middle_initial	
last_name	
address	
street	
street_number	
street_name	
apt_number	
city	
state	
zip	
{ phone_number }	
date_of_birth	
age ()	



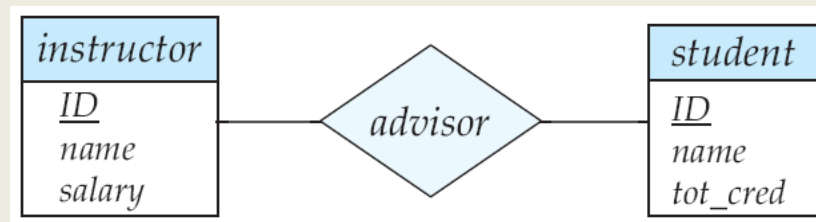
❑ Entity sets with **multivalued attributes**

- ❑ A multivalued attribute M of an entity set E is represented **by a separate schema EM**
- ❑ Schema EM has attributes corresponding to the primary key of E and an attribute corresponding to multivalued attribute M
 - ❑ E.g., multivalued attribute *phone_number* of *instructor* is represented by a schema *inst_phone* (ID, phone_number)
- ❑ Each value of the multivalued attribute maps to a separate tuple of the relation on schema EM
 - ❑ E.g., an *instructor* entity with primary key 22222 and phone numbers 456-7890 and 123-4567 maps to two tuples: (22222, 456-7890) and (22222, 123-4567)



❑ Representing relationship sets

- A many-to-many relationship set is represented as a schema with **attributes for the primary keys** of the two participating entity sets, and any descriptive attributes of the relationship set
- E.g., *advisor* (s_id, i_id)

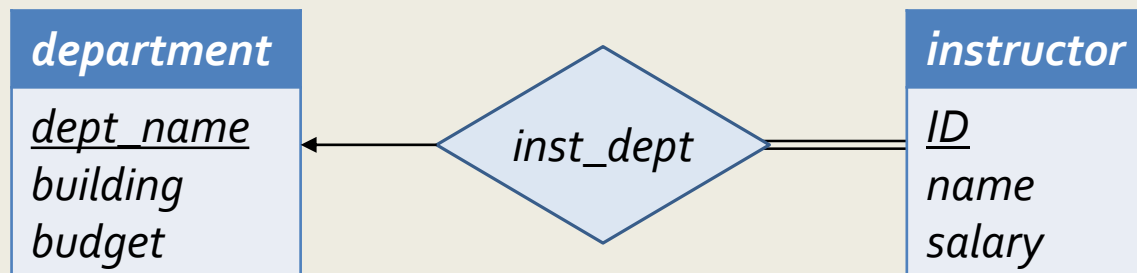




Redundancy of schemas

❑ Many-to-one relationship

- ▣ Sets that are total on the many-side can be represented by adding an extra attribute to the many-side, containing the primary key of the one-side
- ▣ E.g.,
 - ▣ case 1: creating a schema for relationship set *inst_dept*
 - ▣ Case2: add an attribute *dept_name* to the schema arising from entity set *instructor*

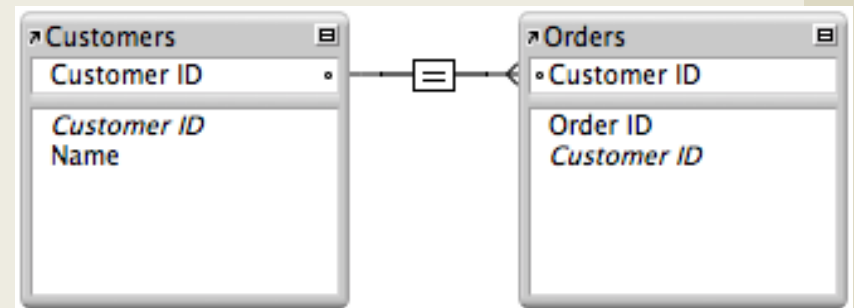
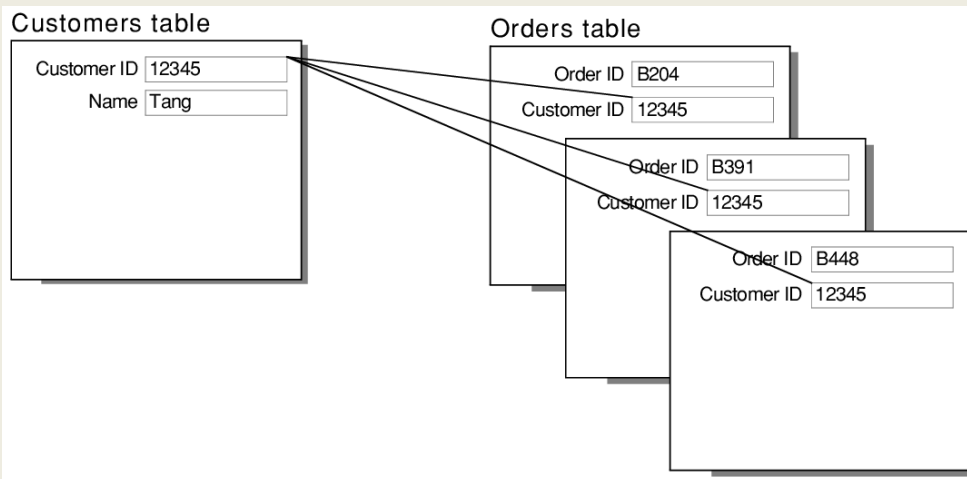




Redundancy of schemas

❑ Many-to-one relationship

- ❑ Primary key field in the Customers table, Customer ID, is designed to contain unique values.
- ❑ The foreign key field in the Orders table, Customer ID, is designed to allow multiple instances of the same value.





Redundancy of schemas



❑ One-to-one relationship

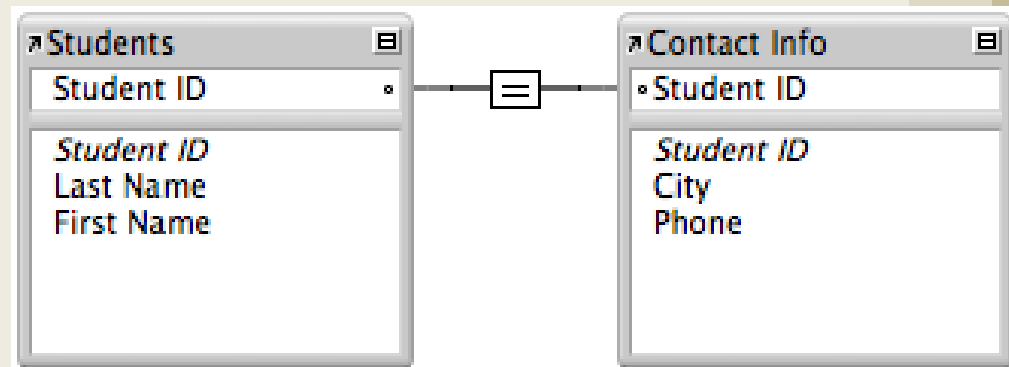
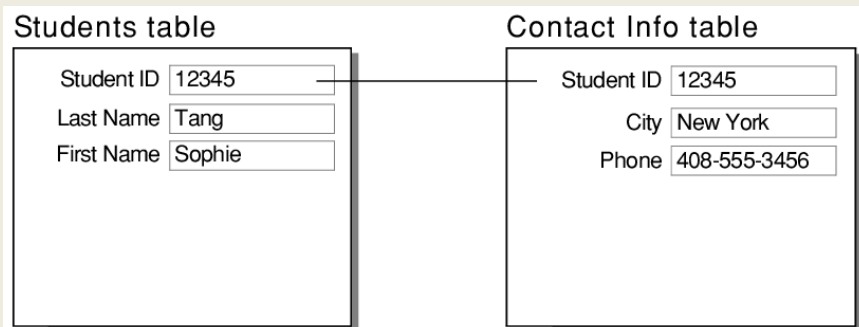
- ❑ Either side can be chosen to act as the many-side
 - ❑ That is, an extra attribute can be added to either of the tables corresponding to the two entity sets
- ❑ If participation is partial on the many-side, replacing a schema by an extra attribute in the schema corresponding to the many-side could result in null values



Redundancy of schemas

❑ One-to-one relationship

- ❑ The key field in each table, Student ID, is designed to contain unique values
- ❑ In the Students table, the Student ID field is the primary key
- ❑ In the Contact Info table, the Student ID field is a foreign key.

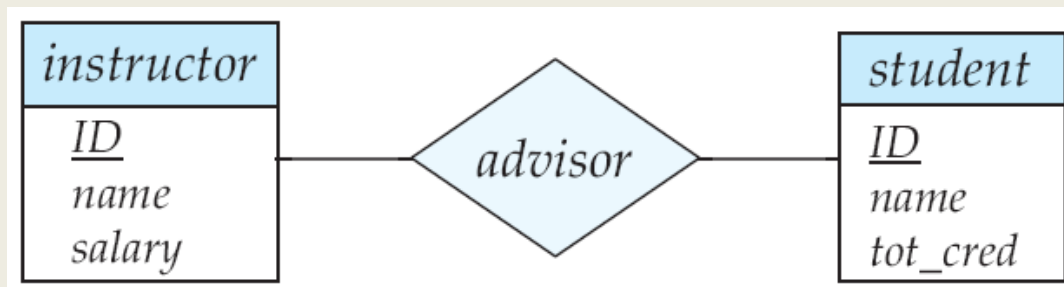




Redundancy of schemas

❑ Many-to-many relationship

- ❑ An *instructor* is associated with several (possibly 0) *students* via *advisor*
- ❑ A *student* is associated with several (possibly 0) *instructors* via *advisor*
- ❑ RDB usually don't allow you to implement a direct many-to-many relationship between two tables.
 - ❑ ➔ You can break the many-to-many relationship into two one-to-many relationships by using a third table, called a *join table*.

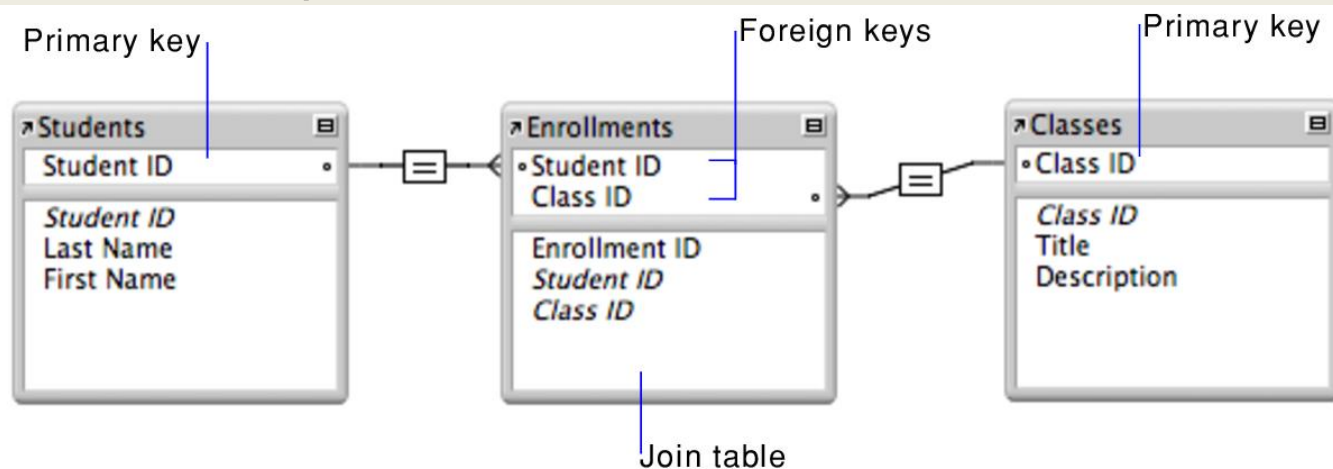




Redundancy of schemas

❑ Many-to-many relationship

- ❑ A student can register for many classes, and a class can include many students.
- ❑ Students table contains a record for each student.
- ❑ Classes table contains a record for each class.
- ❑ A join table, Enrollments, creates two one-to-many relationships—one between each of the two tables.

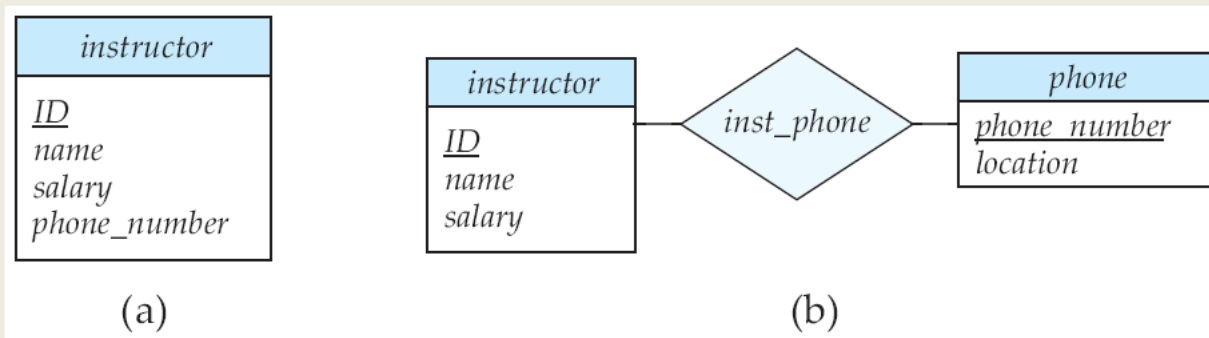




Entity-Relationship Design Issues

❑ Entities vs. attributes

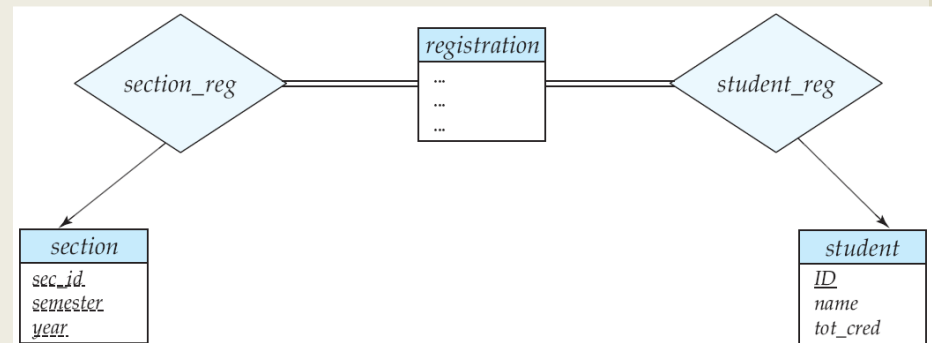
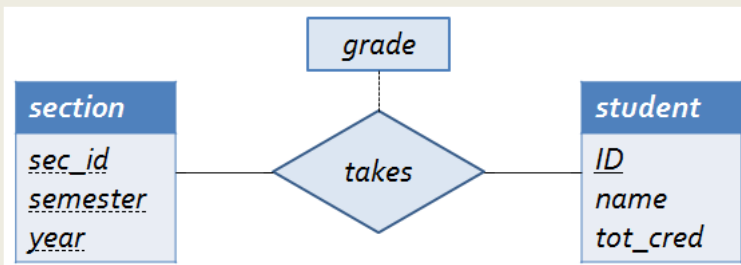
- ▣ Using phone as an entity allows extra information about phone numbers (plus multiple phone numbers)





❑ Entities vs. relationship sets

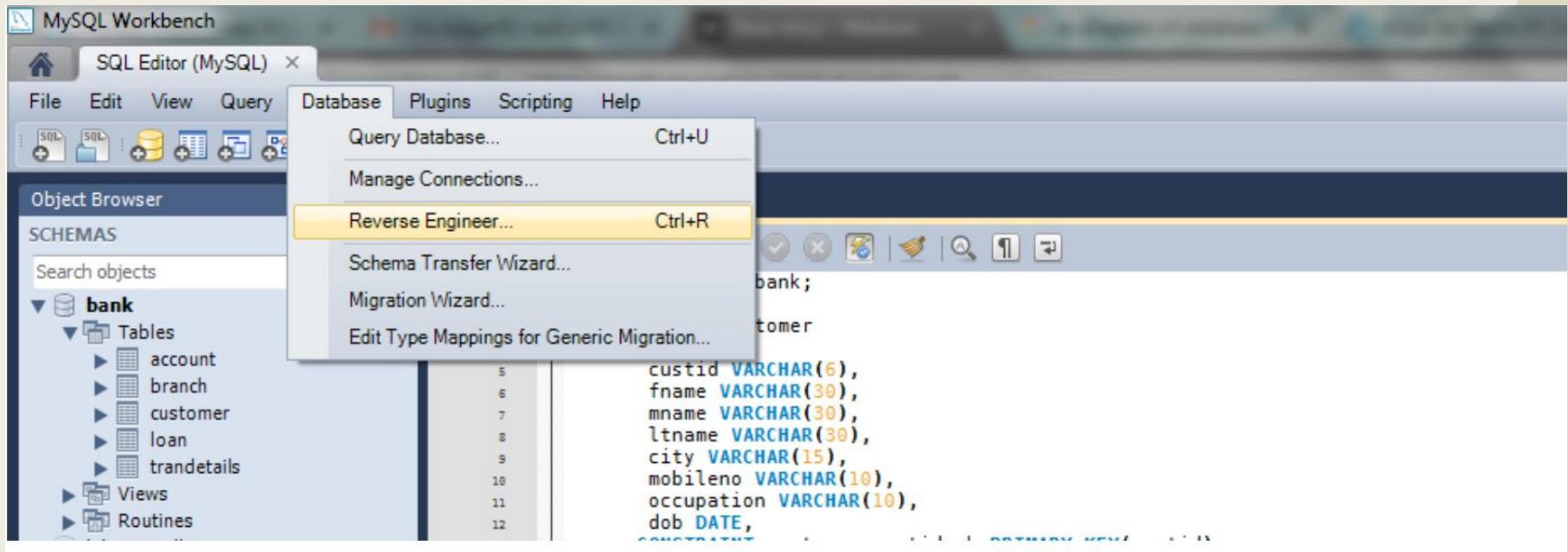
- ❑ If other information is associated with a course-registration record, it might be best to make it an entity
- ❑ Possible guideline is to designate a relationship set to describe an *action* that occurs between entities





Create EER diagram in workbench

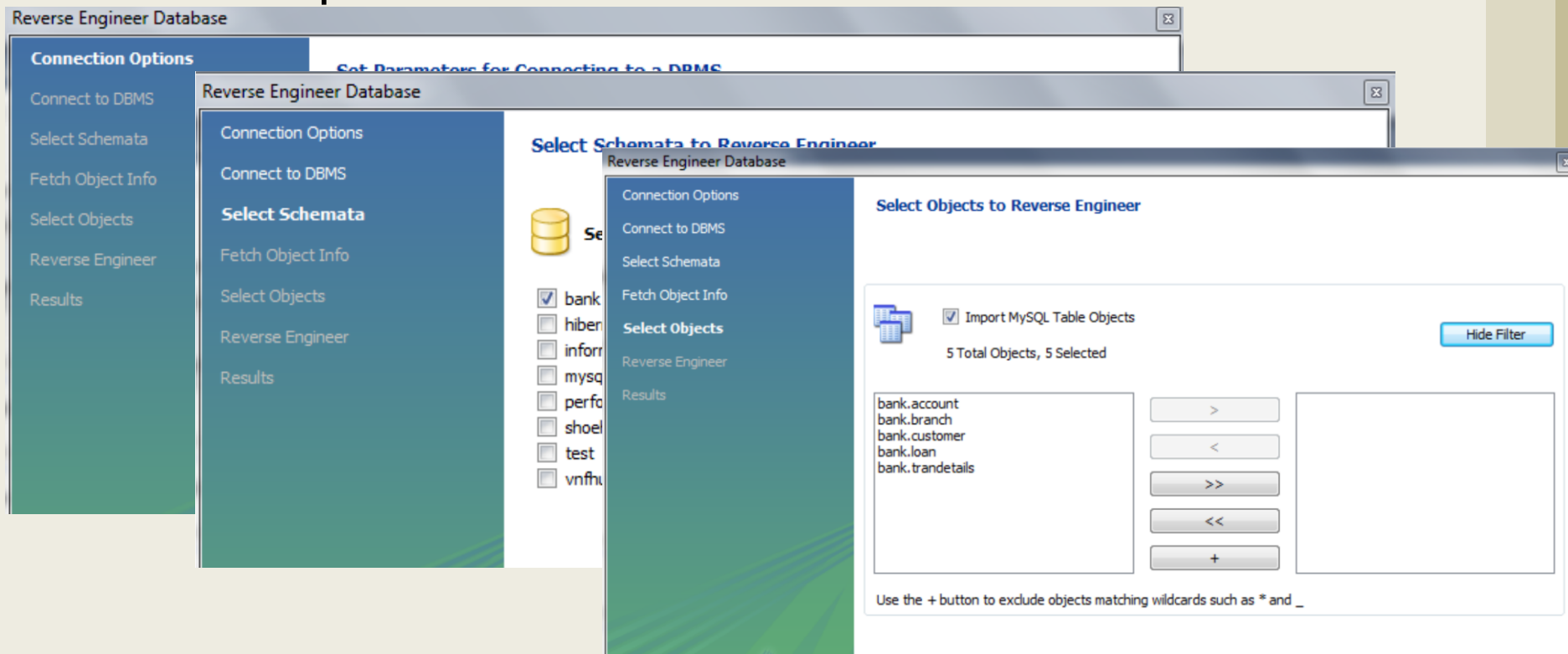
❑ Click on **Database -> Reverse Engineer**.





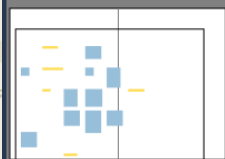
Create EER diagram in workbench

- ❑ Select your **stored connection** (for connecting to your MySQL Server in which database is present) from the dropdown. Then click **Next**.



Bird's Eye

Zoom: 100%



Catalog Tree

mydb
 Tables
 Views
 Routine Groups

Catalog Layers User Types

Description Editor

No Selection

Diagram



physics_fall_2009

departments_total_salary

faculty

history_instructors

tt1
 s1 INT(11)
 s2 CHAR(5)

department
 dept_name VARCHAR(20)
 building VARCHAR(15)
 budget DECIMAL(12,2)
 Indexes

t1
 s1 INT(11)
 s2 CHAR(5)

takes
 ID VARCHAR(5)
 course_id VARCHAR(10)
 sec_id VARCHAR(8)
 semester VARCHAR(6)
 year DECIMAL(4,0)
 grade VARCHAR(2)
 Indexes

teaches
 ID VARCHAR(5)
 course_id VARCHAR(10)
 sec_id VARCHAR(8)
 semester VARCHAR(6)
 year DECIMAL(4,0)
 Indexes

people
 id INT(11)
 first_name VARCHAR(50)
 last_name VARCHAR(50)
 birth_date DATE
 curr_time TIMESTAMP
 Indexes

instructor
 ID VARCHAR(5)
 name VARCHAR(20)
 dept_name VARCHAR(20)
 salary DECIMAL(8,2)
 Indexes

section
 course_id VARCHAR(10)
 sec_id VARCHAR(4)
 semester VARCHAR(6)
 year DECIMAL(4,0)
 building VARCHAR(8)
 room_no INT(11)
 time_slot_id VARCHAR(4)
 Indexes

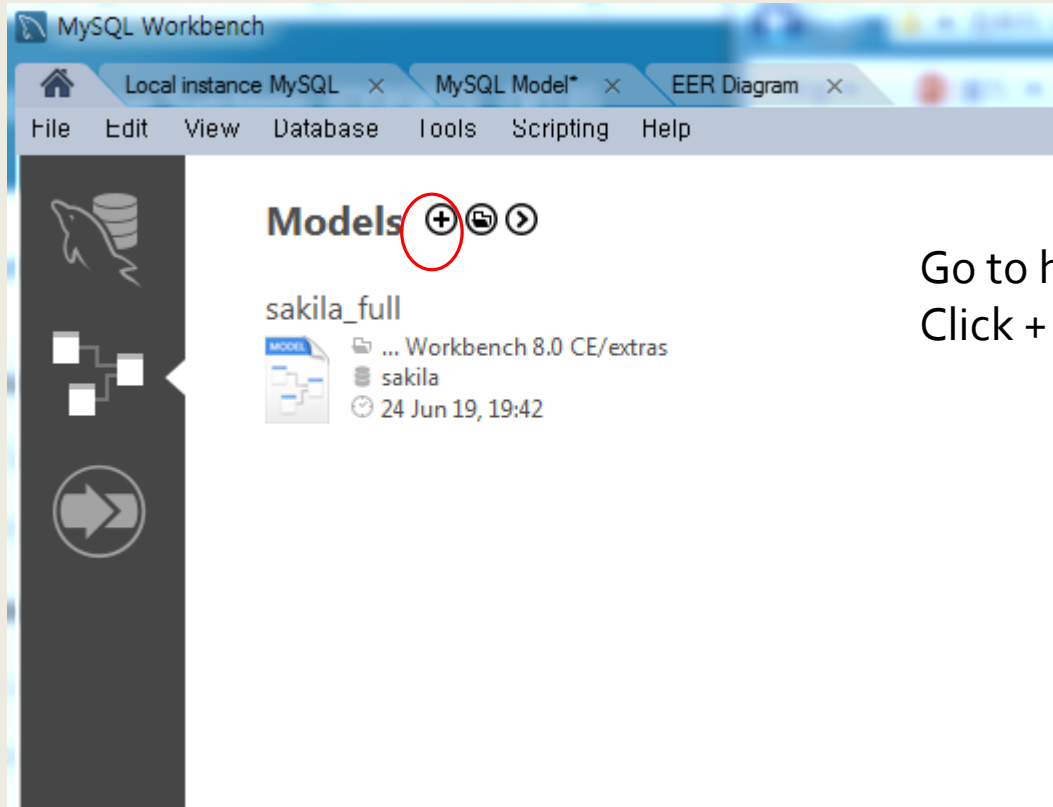
student
 ID VARCHAR(5)
 name VARCHAR(20)
 dept_name VARCHAR(20)
 tot_cred DECIMAL(3,0)
 Indexes

course
 course_id VARCHAR(10)
 title VARCHAR(50)
 dept_name VARCHAR(20)
 credits DECIMAL(2,0)
 Indexes

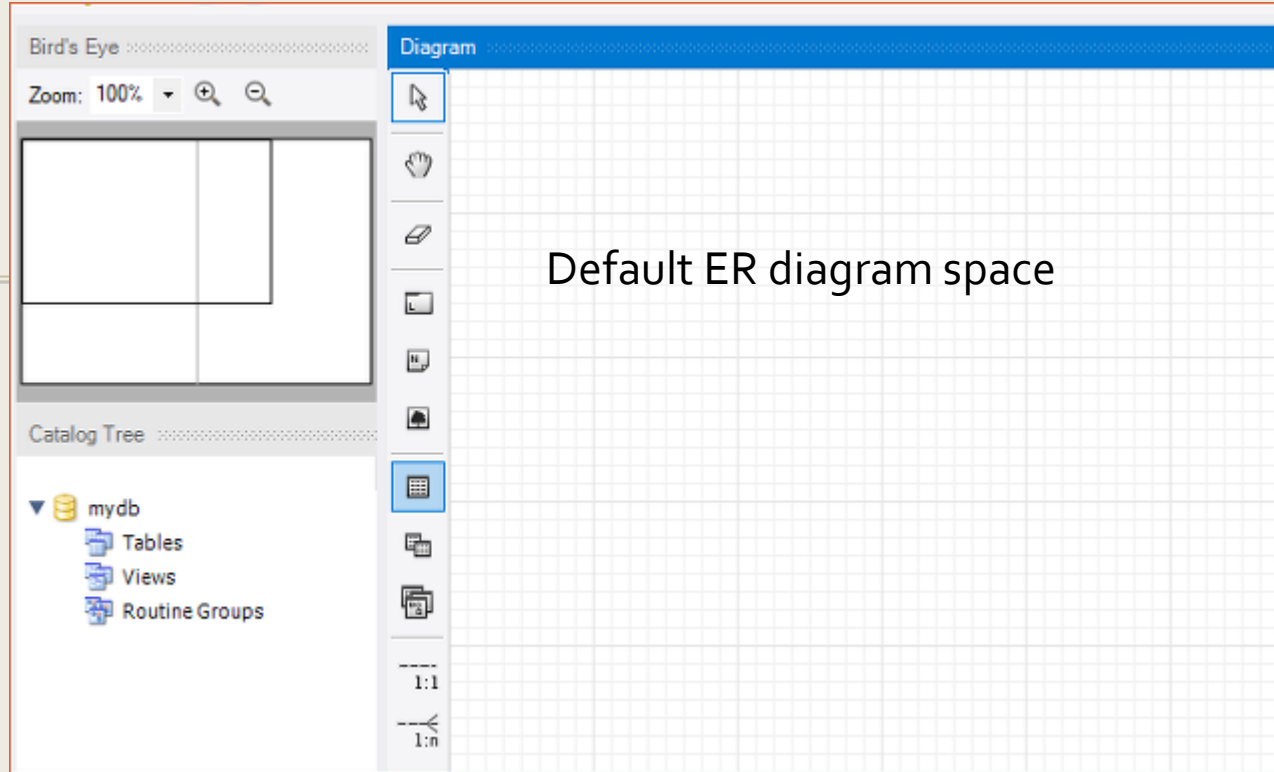
instructor_info



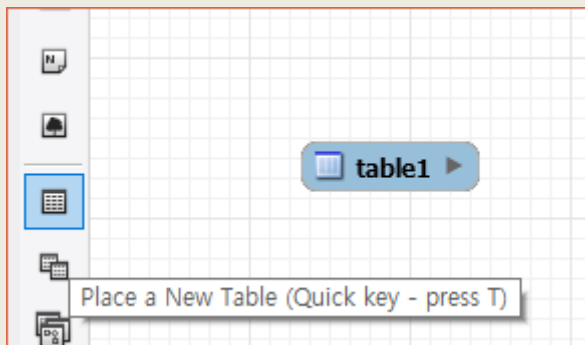
Create ERD in workbench



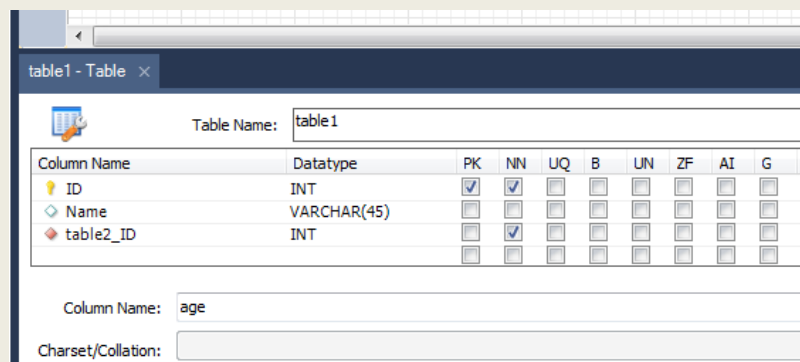
Go to home and click model menu
Click + button to make ER model



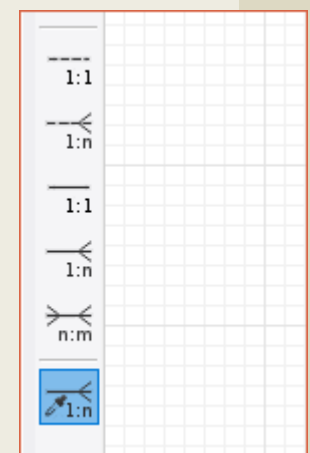
Add table1 and table 2



Click table twice and add attributes

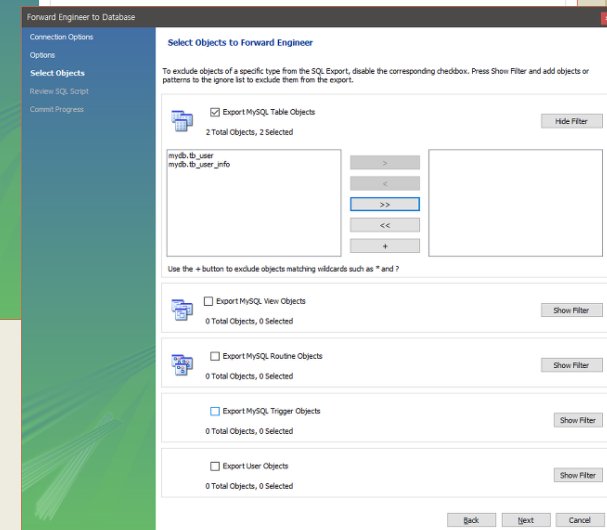
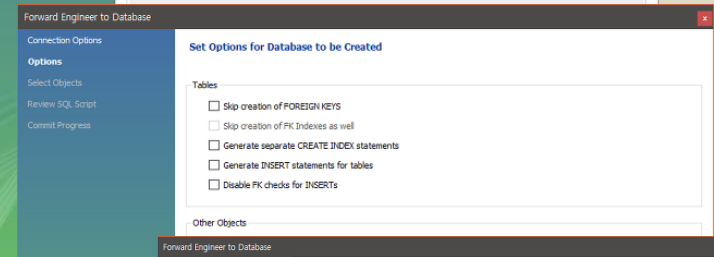
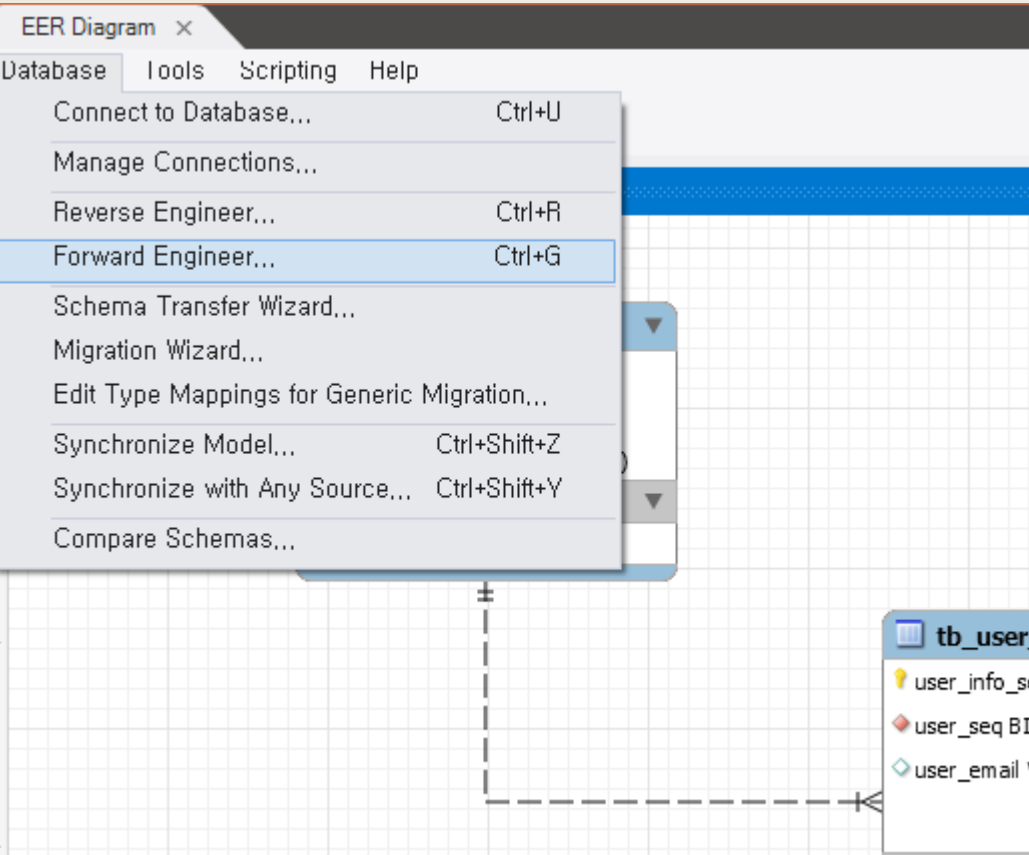


Add relationship





Convert ER model to schema



Click! ➔ NEXT ➔ click! ➔ NEXT ➔...

MySQL Workbench

Local instance MySQL x MySQL Model* (test2.mwb) x EER Diagram x

File Edit View Arrange Model Database Tools Scripting Help

Bird's Eye

Zoom: 100%

Catalog Tree

mydb

Tables

table1

table2

tb_user

Views

Routine Groups

Diagram

table1

ID INT

Name VARCHAR(45)

table2_ID INT

Indexes

table2

ID INT

age VARCHAR(45)

Indexes

1:1

1:n

1:1

1:n

n:m

1:n

table2 - Table

Table Name: table2

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
ID	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
age	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name: age

Charset/Collation: Default Charset

Default Collation

Comments:

MySQL Workbench

Local instance MySQL (mydb) x MySQL Model* (test2.mwb) x EER Diagram x Local instance MySQL

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

mydb

Tables

course

department

instructor

people

section

student

t1

table1

table2

takes

tb_user

teaches

tt1

Views

Stored Procedures

Functions

sakila

sys

test

test2

world

script1

Limit to 1000 rows

1. show databases;
2. use test;
3. use mydb;
- 4.
5. show tables;
6. desc table2;
- 7.
8. insert into course val
9. desc student;
10. desc instructor;
11. show tables;



ER modeling tools



- ❑ Online ER modeling

- ❑ <https://online.visual-paradigm.com/drive/#diagramlist:proj=o&new=ERDiagram>

- ❑ Erwin

- ❑ <https://go.erwin.com/request-erwin-EDGE-trial-or-demo>