

---

# **CS3402**

## ***Database Systems***

# ***Teaching Staff's Information***

---

- **Lecturer: Dr. LIAO Jing (Week 1-8)**

- ◆ Email: [jingliao@cityu.edu.hk](mailto:jingliao@cityu.edu.hk)

- ◆ Phone: 3442-8619

- ◆ Office: G6413

- **Lecturer: Dr. HUANG Heqing (Week 9,10,12,13)**

- ◆ Email: [heqhuang@um.cityu.edu.hk](mailto:heqhuang@um.cityu.edu.hk)

- ◆ Phone: 3442-8504

- ◆ Office: G6429

# *Teaching Staff's Information*

---

## ■ Tutorial Teaching:

- ◆ TA1/TC1: Mr. WANG Can [cwang355-c@my.cityu.edu.hk](mailto:cwang355-c@my.cityu.edu.hk)
- ◆ TA2/TC2 : Mr. DU Hao [haodu8-c@my.cityu.edu.hk](mailto:haodu8-c@my.cityu.edu.hk)
- ◆ TA3/TC3 : Mr. ZHANG Jingbo [jbzhang6-c@my.cityu.edu.hk](mailto:jbzhang6-c@my.cityu.edu.hk)
- ◆ TA4/TC4 : Mr. HUANG Zhitong [zhithuang2-c@my.cityu.edu.hk](mailto:zhithuang2-c@my.cityu.edu.hk)

## ■ Other TAs:

---Homework and midterm grading

- ◆ Mr. WAN Ziyu [ziyuwan2-c@my.cityu.edu.hk](mailto:ziyuwan2-c@my.cityu.edu.hk)
- ◆ Ms. ZHANG Peiying [peiyzhang9-c@my.cityu.edu.hk](mailto:peiyzhang9-c@my.cityu.edu.hk)
- ◆ Mr. WU Ronghuan [ronghwu2-c@my.cityu.edu.hk](mailto:ronghwu2-c@my.cityu.edu.hk)
- ◆ Mr. ZHANG Zhiyuan [zzhang452-c@my.cityu.edu.hk](mailto:zzhang452-c@my.cityu.edu.hk)

# Course Overview

---

## ■ Course Format:

### ◆ Lecture

◆ Time: Monday 12:00pm - 2:50pm

◆ Venue: YEUNG LT-2

◆ Delivery mode: Face-to-Face

### ◆ Tutorial

◆ Time: Week 6-13 (2 Q&A and 6 Labs)

◆ Venue: MMW 2450

◆ Delivery mode: Face-to-Face

# Assessment

---

- Coursework -- 40% :
  - ◆ Mid-term -- 25%
    - ◆ Date: **Week 7, Mon (Mar. 4th)**
    - ◆ Time: Lecture time
    - ◆ Format: Face-to-face, Open-book exam
  - ◆ Homework assignments (3 times) -- 15%
  - ◆ Lab attendance (At least 5 of the 6 labs) -- 3% *bonus*  
(*such that coursework is capped to 40%*)
- Final examination (open-book)-- 60%
  - ◆ *Get 30 out of 100 to pass*

# Course Materials

---

## ■ Text books

- ◆ “Fundamentals of Database Systems”, 6<sup>th</sup> edition (*or later*), by R. Elmasri, S.B. Navathe, Addison-Wesley.
- ◆ “Database System Concepts”, 5<sup>th</sup> edition (*or later*), by A. Silberschatz, H. Korth, S. Sudarshan, McGraw-Hill Companies Inc.

- Notations may vary in different books. Please stick to the ones used in this lecture notes!

# ***Database is everywhere***

---

- You cannot avoid it and it's everywhere!
- You can say it actually makes the current society and your life work!
- Banking/Credit card /Social Security Info...
- Online shopping/booking...



# Personal Instalment Loan

Enjoy up to **HK\$500** Cash Rebate  
for online or mobile application

[Apply Now >](#)

Annualised Percentage Rate as low as 2.68%.  
Terms and conditions apply to the offers and services.  
To borrow or not to borrow? Borrow only if you can repay!

Always with You!

Monthly flat rate  
as low as

0.12%



[Register FPS Now >](#)

To receive money with JUST a mobile  
number / email address / FPS ID

Follow global market  
trends with our investment  
outlook video

[All in "Investment  
Corner" >](#)

Investment involves risks.



- › Exchange Rates – T/T
- › Exchange Rates – Notes
- › Gold Prices
- › Interest Rates
- › HKD Prime Rate
- › Property Valuation

[Personalized settings](#)



LOGON

- › Personal e-Banking
- › Business e-Banking
- › Hang Seng HSBCnet
- › Stock Trading



**NEW!** **PRICE ASSURANCE**

**\$350** YOUR FARE  
**-\$250** LOWER FARE  
**\$100** CASH REFUND

**Never overpay.  
Simple arithmetic,  
Orbitz style.**

**ORBITZ**

**Quick Search** **Vacation Packages** **Hotels** **Flights** **Cars & Rail** **Cruises** **Activities** **Deals**

- ☒ **Flight**  
☐ **Hotel**  
☐ **Car**  
☐ **Activities**  
☐ **Cruises**
- ☐ **Flight + Hotel**  
☐ **Flight + Car**  
☐ **Hotel + Car**  
☐ **Flight + Hotel + Car**

**Winter Escape**  
**SAVE UP TO 50%**  
**ON HOTELS**  
[Get Details](#)

**From** City name or [airport](#)

**To** City name or [airport](#)

**Leave**  
 mm/dd/yy

**Return**  
 mm/dd/yy

**Travelers** ([Children or seniors?](#))  
 Adult (18-64)

**Flight preference**  
☐ I prefer non-stop flights

[Find Flights](#)

[Find Flight + Hotel](#)

[Expand search options](#) (Multi-city, preferred airlines, etc.)  
[One-way](#) | [Flexible dates](#)

Welcome to Orbitz.  
[Sign in](#) | [Register now](#)  
[Site feedback](#)

Search (Atlantis, beach...)  
 [Go](#)

**MY STUFF**  
[My Trips](#) [My Account](#) [Home](#) [Traveler Update](#) [Customer Support](#)

**WINTER SALE**  
 Save up to **50%**  
[Learn More](#)

Sponsored by:  
**PUERTO RICO**

**Explore**

<b>DESTINATIONS</b>	<b>INTERESTS</b>
<a href="#">Cancun</a>	<a href="#">Beach</a>
<a href="#">Las Vegas</a>	<a href="#">Romance</a>
<a href="#">New York City</a>	<a href="#">Last Minute</a>
<a href="#">San Francisco</a>	<a href="#">Family</a>
<a href="#">SEE MORE</a>	<a href="#">SEE MORE</a>

**More featured deals**

[View all deals](#)

From\*

[Student? You can travel cheaper!](#)

[Save](#)

NCL - UP to \$500 Free Onboard Spending

\$199



Entrez, The Life Sciences Search Engine

[HOME](#)
[SEARCH](#)
[SITE MAP](#)
[PubMed](#)
[All Databases](#)
[Human Genome](#)
[GenBank](#)
[Map Viewer](#)
[BLAST](#)

Search across databases  [GO](#) [Clear](#) [Help](#)

### Welcome to the Entrez cross-database search page

- |  |  |
|--|--|
| <b>PubMed:</b> biomedical literature citations and abstracts | <b>Books:</b> online books                           |
| <b>PubMed Central:</b> free, full text journal articles      | <b>OMIM:</b> online Mendelian Inheritance in Man     |
| <b>Site Search:</b> NCBI web and FTP sites                   | <b>OMIA:</b> online Mendelian Inheritance in Animals |

- |   |  |
|---|--|
| <b>Nucleotide:</b> Core subset of nucleotide sequence records | <b>dbGaP:</b> genotype and phenotype                             |
| <b>EST:</b> Expressed Sequence Tag records                    | <b>UniGene:</b> gene-oriented clusters of transcript sequences   |
| <b>GSS:</b> Genome Survey Sequence records                    | <b>CDD:</b> conserved protein domain database                    |
| <b>Protein:</b> sequence database                             | <b>3D Domains:</b> domains from Entrez Structure                 |
| <b>Genome:</b> whole genome sequences                         | <b>UniSTS:</b> markers and mapping data                          |
| <b>Structure:</b> three-dimensional macromolecular structures | <b>PopSet:</b> population study data sets                        |
| <b>Taxonomy:</b> organisms in GenBank                         | <b>GEO Profiles:</b> expression and molecular abundance profiles |



amazon.com

Hello, Ruoming Jin. We have [recommendations](#) for you. (Not Ruoming?)

FREE 2-Day Shipping, No Minimum Purchase

[Ruoming's Amazon.com](#)

[Today's Deals](#)

[Gifts & Wish Lists](#)

[Gift Cards](#)

[Your Account](#) | [Help](#)

Search All Departments

GO

Cart

Your Lists

Shop All Departments

- Books
- Movies, Music & Games
- Digital Downloads
- Computers & Office
- Electronics
- Home & Garden
- Grocery, Health & Beauty
- Toys, Kids & Baby
- Apparel, Shoes & Jewelry
- Sports & Outdoors
- Tools, Auto & Industrial

100% More



Books, Newspapers, and Blogs

We've doubled our selection since launch to over 200,000 titles—all available wirelessly in under 60 seconds:

- New York Times bestsellers are \$9.99, unless otherwise marked
- Expanded newspaper selection, including *The New York Times*, *Wall Street Journal*, and *Financial Times*
- More magazines and blogs, from *Newsweek* to *Time* to *Gawker*

amazonkindle [Learn more](#)

Amazon Daily BLOG

5 posts since yesterday  
Posts for Ruoming

Treat Yourself



[Combinatorial Optimization](#)

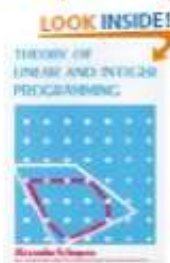
From the reviews: What a book! Actually it is three books which come... [Read more](#)  
\$135.00 **\$107.51**

[See more in your Wish List](#)

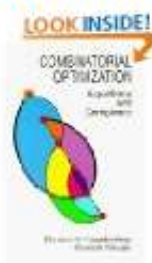
Check This Out

- Tax Central**  
Find tax software, books, and more
- 2009 Cast Your Vote**  
See our Top 10 New Year's Resolutions. Which one's for you?
- Holiday Returns**  
Visit our self-service Online Returns Center.
- For the Photography Fan**  
Zoom in on top

Frequently Bought with Items in Your Wish List



[Theory of Linear and Integer Programming](#)  
Paperback by Alexander



[Combinatorial Optimization...](#)  
Paperback by Christos H.



[Approximation Algorithms](#) Hardcover by Vijay V. Vazirani

Get Your Motor Running  
Explore Our New Motorcycle & ATV Store



[Shop now at Amazon.com](#)



## CMU Database Group

1.86万位订阅者 · 244 个视频

Carnegie Mellon University Database Group More Info: <http://db.cs.cmu.edu>.

订阅



### What is Database & SQL?

Guru99 · 163万次观看 · 7年前

<https://www.guru99.com/introduction-to-database-sql.html> This Database tutorial explains the concept of DBMS (Database ...

字幕



### MySQL IN 10 MINUTES (2020) | Introduction to Databases, SQL, & MySQL

365 Data Science · 5.2万次观看 · 1年前

Download Our Free Data Science Career Guide: <https://bit.ly/3fYJOKp> Sign up for Our Complete Data Science ...

字幕



### Microsoft Access 2019 - Full Tutorial for Beginners [+ General Overview]

bai · 47万次观看 · 1年前

Get into a new Way of Learning Microsoft Access 2019. Getting started, basics. MORE at <https://theskillsfactory.com/>. Full Guide ...

4K 字幕



### Relational Database Concepts

Prescott Computer Guy · 75万次观看 · 8年前

Basic Concepts on how relational databases work. Explains the concepts of tables, key IDs, and relations at an introductory level.

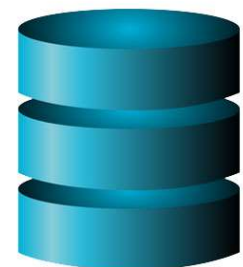


# Motivating Example

**Data:** in Computer system, Data is any sort of information which is stored in computer memory.

Student ID	Student name	Course	Course Name	Grade
50000000	Peter Wong	CS3402	Database Systems	B+
50000000	Peter Wong	CS2302	Data Structures	A
50000001	Mary Tsui	CS3402	Database Systems	A-
50000002	Bob Lee	CS3402	Database Systems	B

- This way of storing data good?
- How to improve?



# *Introduction to DB Systems*

---

## ■ Motivations

### ◆ File-processing Systems

- ◆ permanent records stored in various files
- ◆ application programs written to extract & add records

### ◆ Disadvantages of traditional file-processing systems

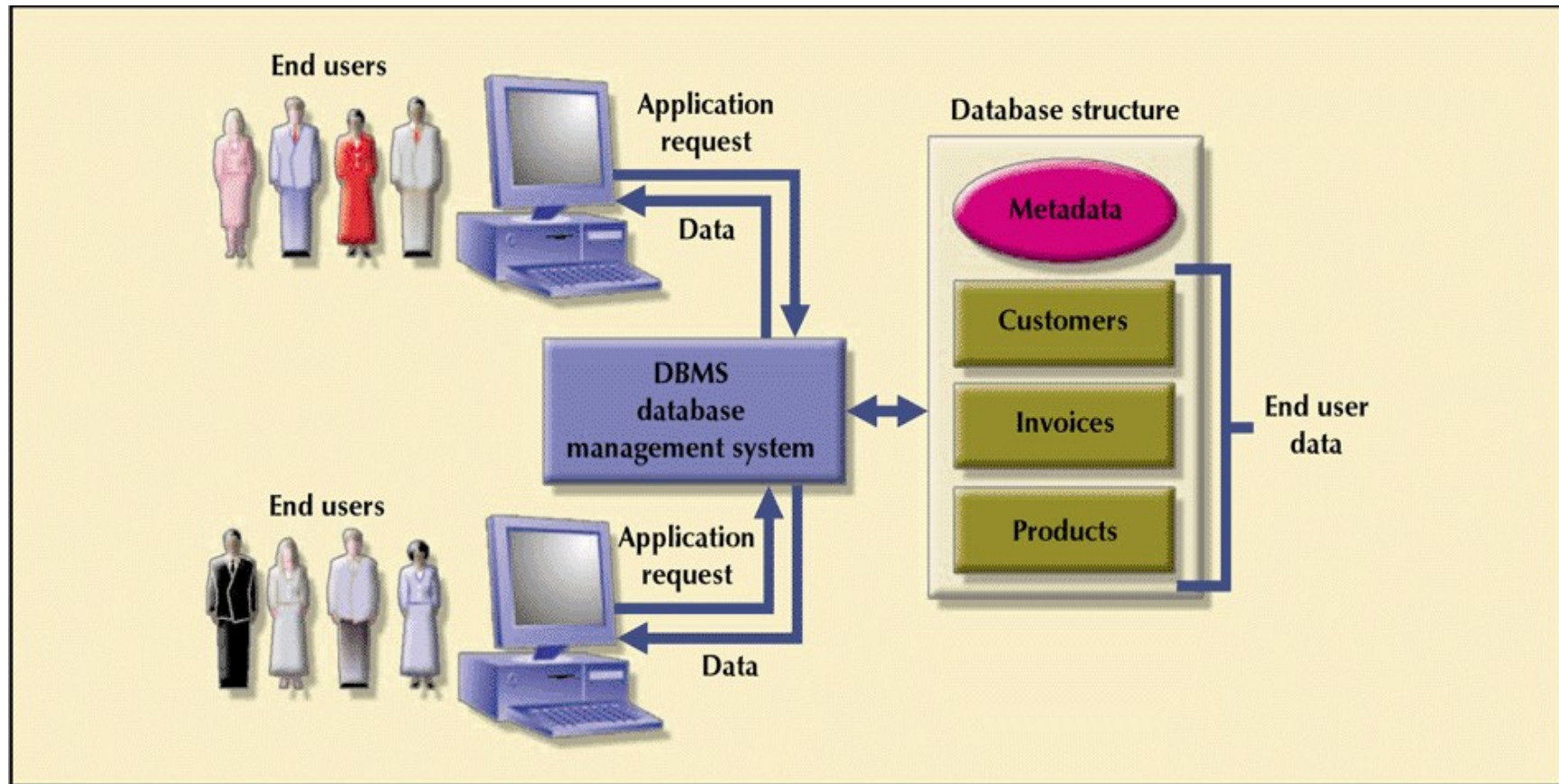
- ◆ data redundancy & inconsistency
- ◆ difficulty in accessing data
- ◆ data isolation & different data formats
- ◆ concurrent access anomalies
- ◆ security problem
- ◆ integrity problem

# *Introduction to DB Systems*

---

- What is a Database (DB)?
  - ◆ A non-redundant, persistent collection of logically-related data that are structured to support various processing and retrieval needs.
  
- Database Management System (DBMS)
  - ◆ A set of software programs for creating, storing, updating, and accessing the data of a DB.
  - ◆ E.g.: Oracle, Mysql, Oceanbase

# Database Management System



Database Systems: Design, Implementation, & Management: Rob & Coronel

- manages interaction between end users and database

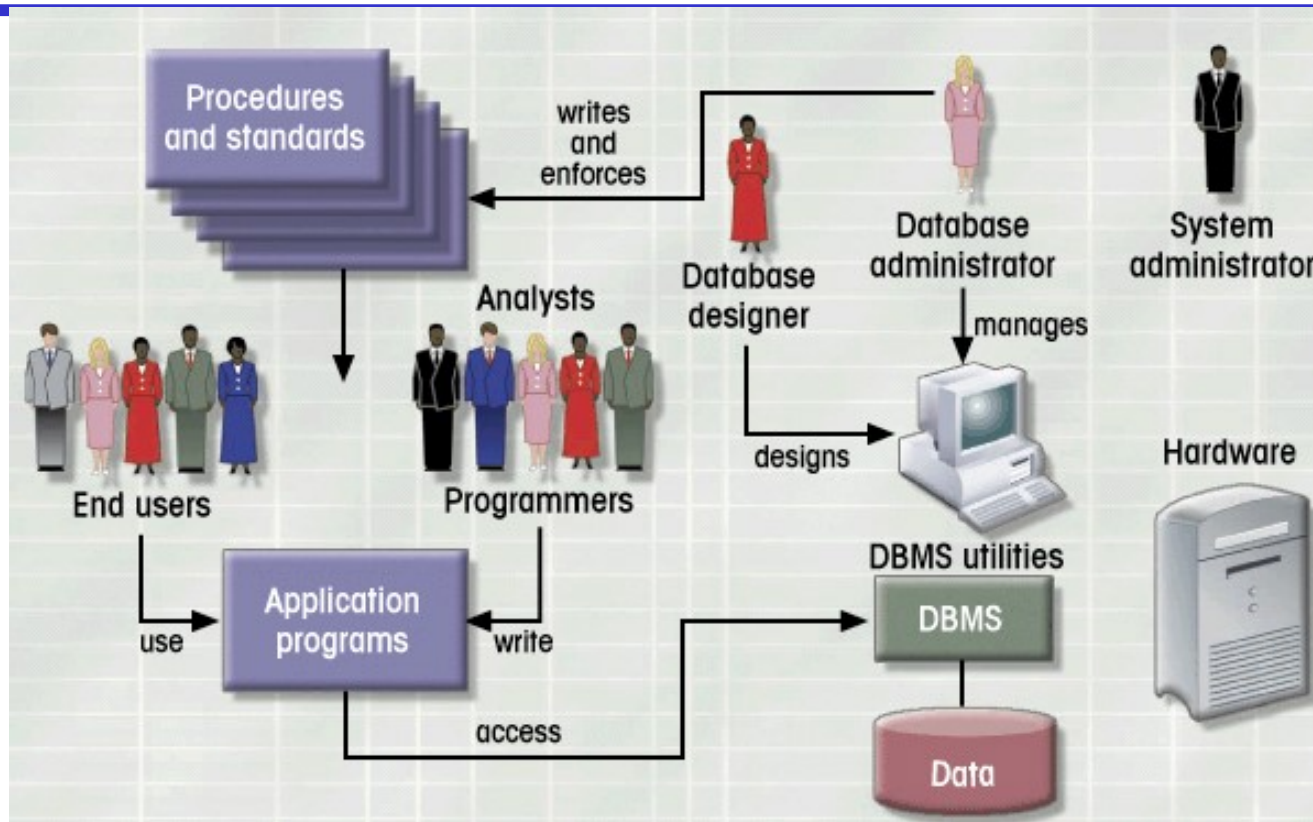


# *Database Management System*

---

- Difference between DBMS & other programming systems
  - ◆ the ability to manage persistent data
  - ◆ primary goal of DBMS: to provide an environment that is convenient, efficient, and robust to use in retrieving & storing data
- Other DBMS capabilities
  - ◆ data modeling
  - ◆ high-level languages to define, access and manipulate data
  - ◆ transaction management & concurrency control
  - ◆ access control
  - ◆ recovery

# Database System

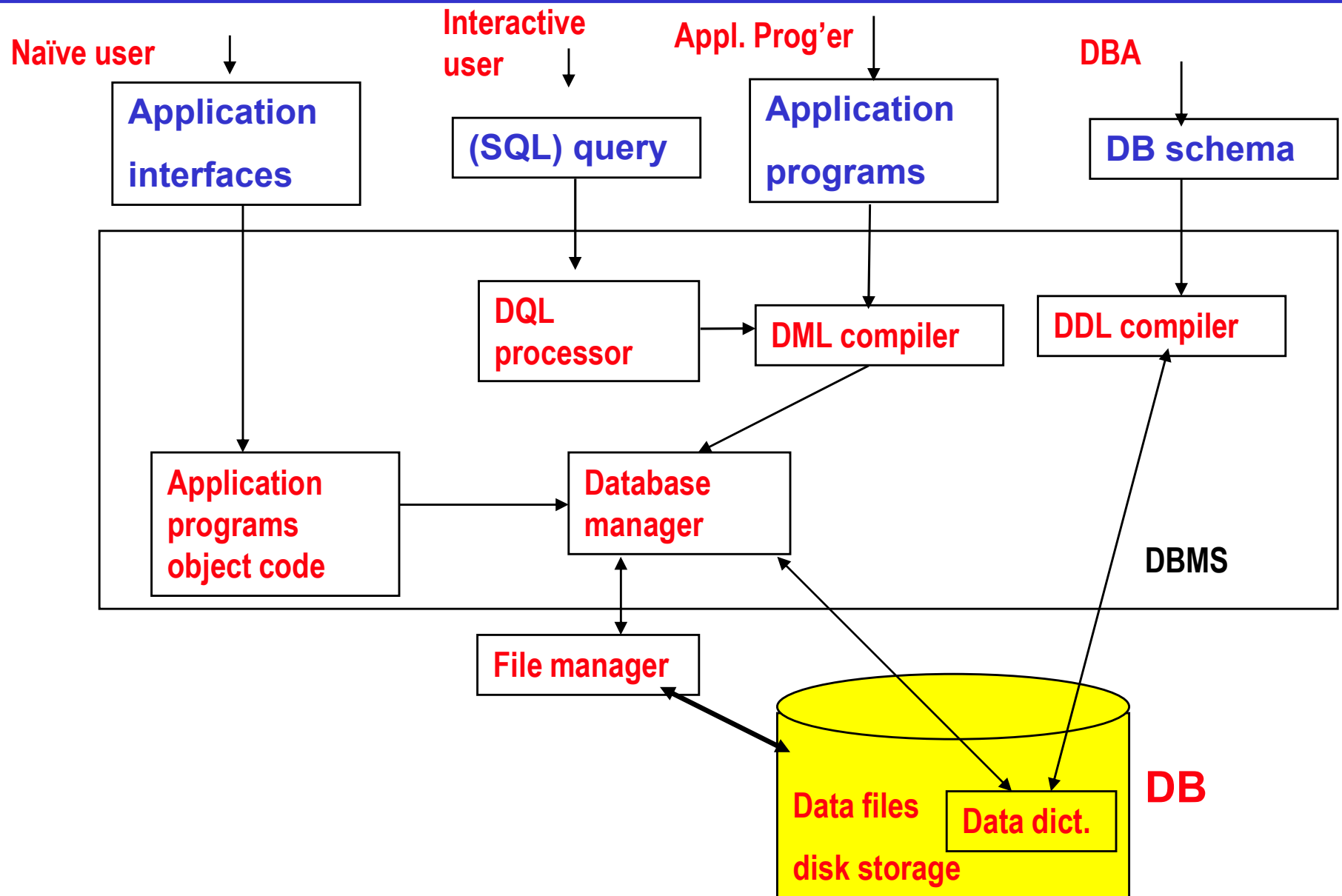


- **Hardware**
- **Software**
  - OS
  - DBMS
  - Applications
- **People**
- **Procedures**
- **Data**

## ■ Database System

- ◆ an **integrated system** of hardware, software, people, procedures, and data
- ◆ that define and regulate the collection, storage, management, and use of data within a database environment

# DB System Architecture



# ***DB System Architecture***

---

- *Database Users*

- ◆ Naive Users:

- ◆ Running application programs

- ◆ Interactive Users:

- ◆ Using query languages

- ◆ Application Programmers

- ◆ Writing embedded DML in a host language

# ***DB System Architecture***

---

- *Database Administrator (DBA)*

- ◆ DBA is the person who has central control over the DB
- ◆ Main functions of DBA:
  - ◆ schema definition
  - ◆ storage structure and access method definition
  - ◆ schema and physical organization modification
  - ◆ granting of authorization for data access
  - ◆ integrity constraint specification

# ***DB System Architecture***

---

- *Data Query Language (DQL)*
  - a language used to make queries in databases
  - e.g. search records with giving conditions (sex="Female")
- *Data Manipulation Language (DML)*
  - a language that enables users to manipulate data
  - e.g. insert or delete records
- *Data Definition Language (DDL)*
  - a language for defining DB schema
  - e.g., create, modify, and remove database objects such as tables, indexes, and users.

# ***DB System Architecture***

---

## ■ DB Manager

- ◆ interface between stored data and application programs/queries
- ◆ translate conceptual level commands into physical level ones
- ◆ responsible for
  - ◆ access control
  - ◆ concurrency control
  - ◆ backup & recovery
  - ◆ integrity

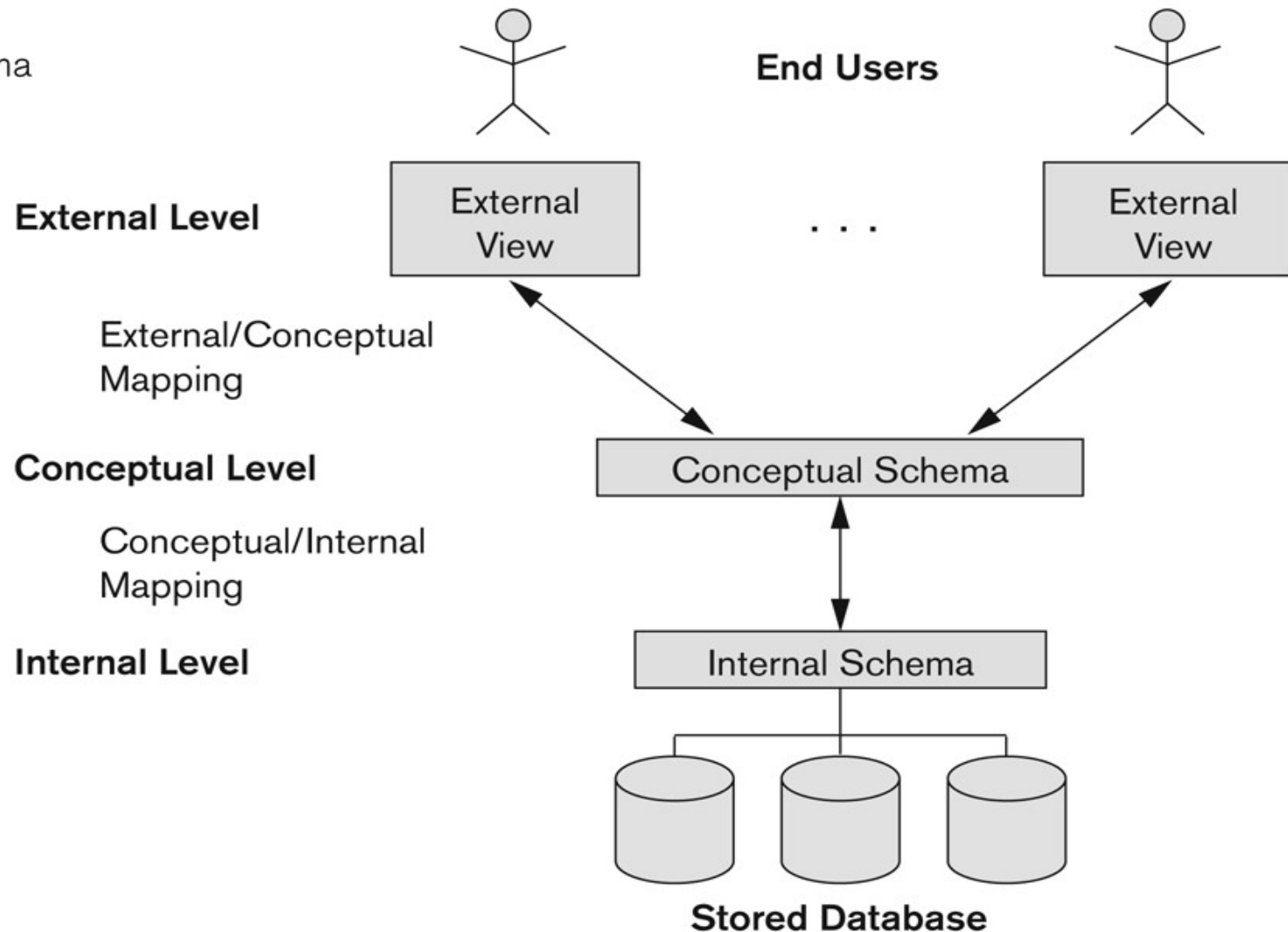
## ■ File Manager

- ◆ allocation of space
- ◆ operations on files

# ***Data Abstraction: 3-level architecture***

**Figure 2.2**

The three-schema architecture.





# ***Data Abstraction***

---

## ■ Data Abstraction

- ◆ Abstract view of the data
  - ◆ simplify interaction with the system
  - ◆ hide details of how data is stored and manipulated
- ◆ Levels of abstraction
  - ◆ Physical/internal level: **internal schema** uses a physical data model and describes the complete details of data storage and access paths for the database.
  - ◆ Conceptual level: **conceptual schema** describes the structure of the whole database for a community of users and hides the details of physical storage structures.
  - ◆ View/external level: **external schema** describes the part of the database that a particular user group is interested in and hides the rest from that group.

# ***Data Independence***

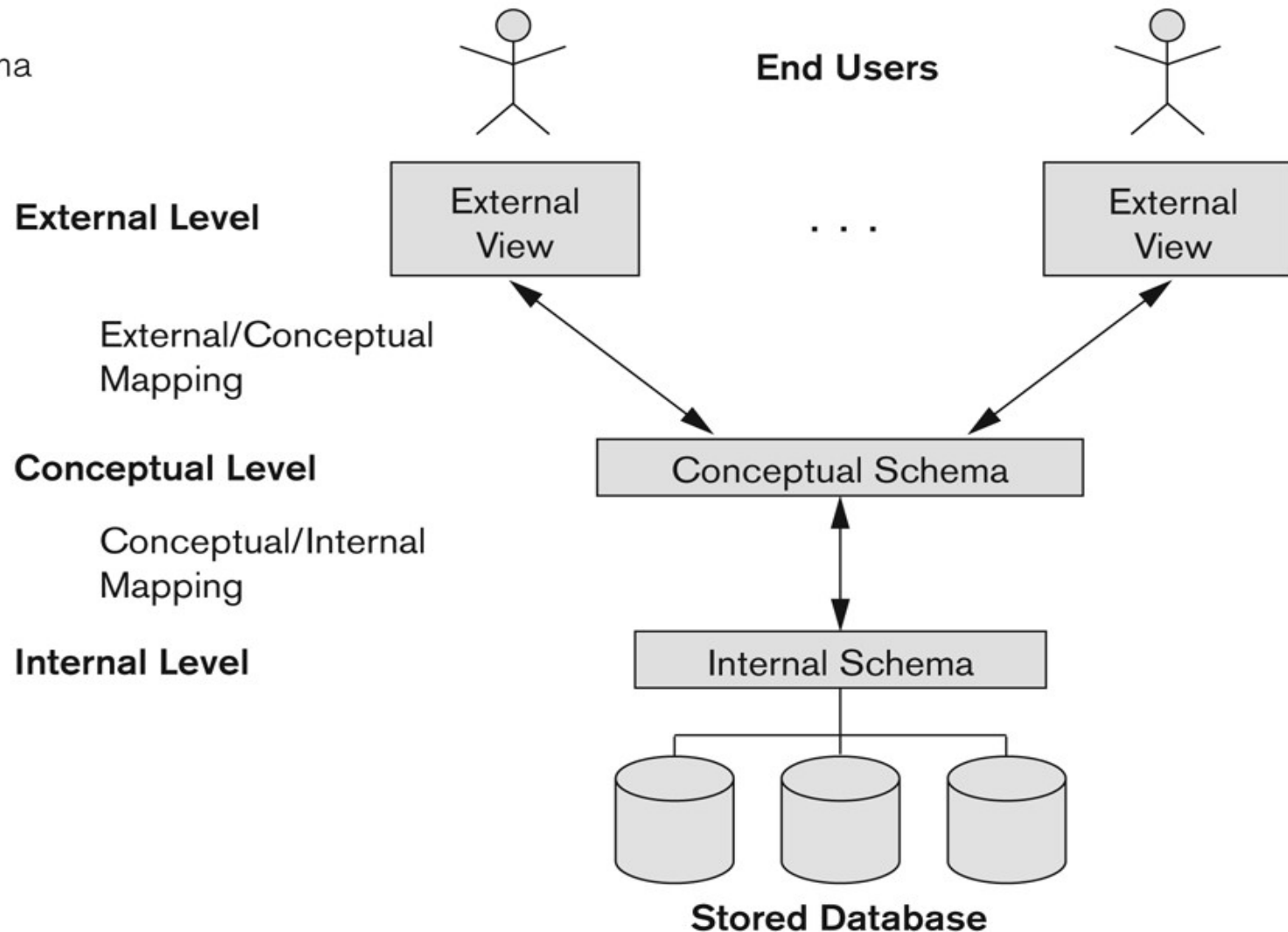
---

- ◆ the ability to modify a schema definition in one level without affecting a schema in the next higher level
- ◆ there are two kinds (a result of the 3-level architecture):
  - ◆ ***physical data independence***
    - *the ability to modify the physical schema without altering the conceptual schema and thus, without causing the application programs to be rewritten*
  - ◆ ***logical data independence***
    - *the ability to modify the conceptual schema without causing the application programs to be rewritten*

# ***Data Abstraction: 3-level architecture***

**Figure 2.2**

The three-schema architecture.



# *Data Models*

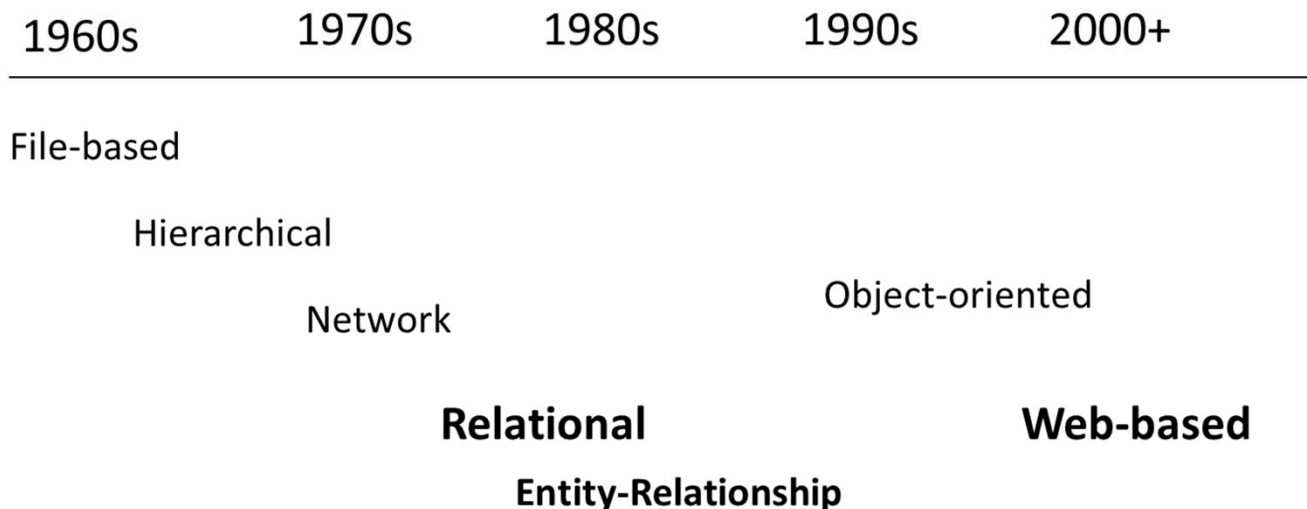
---

## ■ Data Model (conceptual level)

- ◆ A collection of conceptual tools for describing data, data relationships, operations, and consistency constraints
- ◆ the “core” of a database

## Evolution of Data Models

### • Timeline



# *Course Objectives*

---

- ER model: characterize relationships among entities
- Relational model: transform from ER diagram to tables
- SQL: language for writing queries
- Relational Algebra: logical way to represent queries
- Normal Forms: how to design good tables
- File Organization: provide file level structure to speed up query
- Query Optimization: transform queries into more efficient ones
- Transactions and Concurrency Control: handle concurrent operations and guarantee correctness of the database

# *Course Objectives*

---

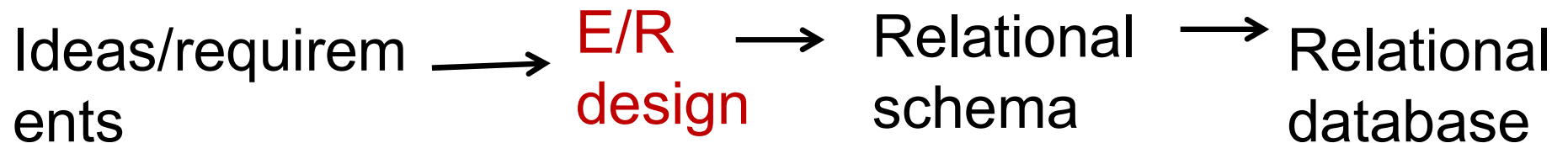
- ER model: characterize relationships among entities
- Relational model: transform from ER diagram to tables
- SQL: language for writing queries
- Relational Algebra: logical way to represent queries
- Normal Forms: how to design good tables
- File Organization: provide file level structure to speed up query
- Query Optimization: transform queries into more efficient ones
- Transactions and Concurrency Control: handle concurrent operations and guarantee correctness of the database

# ***The Entity-Relationship Model***

---

## ■ Preliminaries

- ◆ Proposed by P. Chen in 1976
- ◆ Direct, easy-to-understand graphical notation
- ◆ Translates readily to relational schema for database design



## ■ Three basic concepts:

Entity, Attribute, Relationship

# ER Model Concepts

---

## ■ Entity

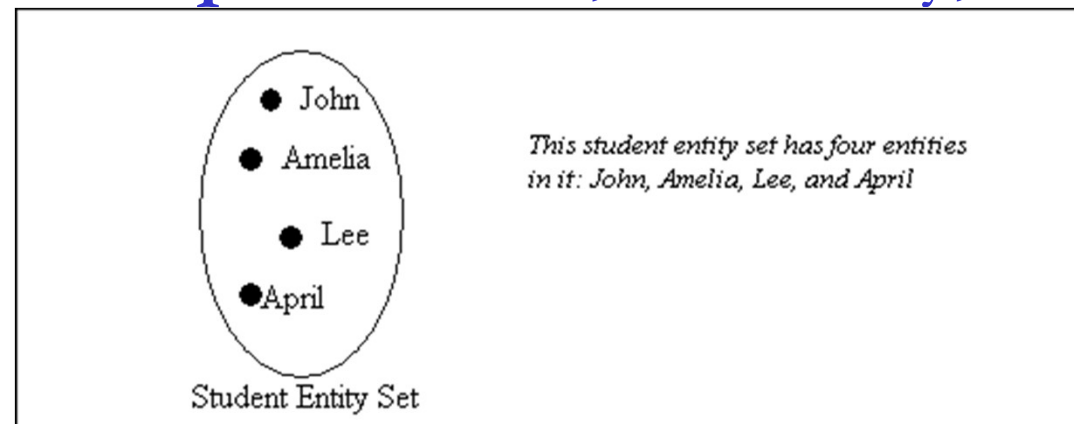
- ◆ a distinguishable object with an independent existence

Example: John Chan, CityU, HSBC, ...

## ■ Entity Set

- ◆ a set of entities of the same type

Example: Student, University, Bank, ...





# ***ER Model Concepts***

---

- **Attribute(Property)** -- a piece of information describing an entity
  - ◆ Example: Name, ID, Address, Sex are attributes of a student entity
- **Each attribute can take a value from a domain**
  - Example: Name  $\in$  Character String,  
ID  $\in$  Integer, ...

# Types of Attributes

---

## ■ Simple

- ◆ Each entity has a **single atomic value** for the attribute. For example, SSN or Sex, name...

## ■ Composite

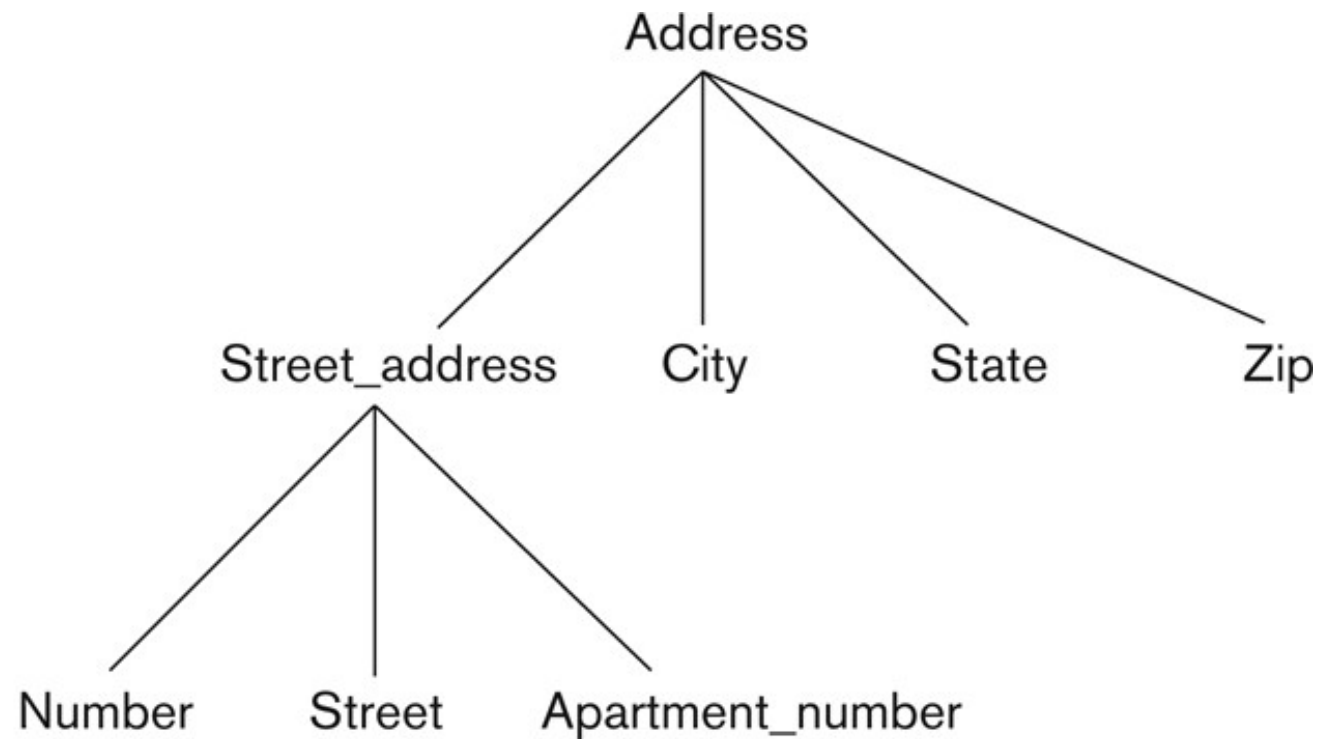
- ◆ The attribute may be composed of several components. For example:
  - ◆ Address(Flat, Block, Street, City, State, Country)
  - ◆ Composition may form a **hierarchy** where some components are themselves composite

## ■ Multi-valued

- ◆ An entity may have multiple values for that attribute. For example, Color of a CAR or PreviousDegrees of a STUDENT
  - ◆ Denoted as {Color} or {PreviousDegrees}
  - ◆ E.g., {{BSc, 1990}, {MSc, 1993}, {PhD, 1998}}

## *Example of a composite attribute*

---



**Figure 3.4**

A hierarchy of composite attributes.

# *Types of Attributes*

---

- In general, composite and multi-valued attributes may be **nested** to any number of levels
  - ◆ For example, PreviousDegrees of a STUDENT is a composite multi-valued attribute denoted by {PreviousDegrees (Degree, Year)}
  - ◆ Multiple PreviousDegrees values can exist
  - ◆ Each has two subcomponent attributes:
    - ◆ Year, Degree

# *Key Attributes*

---

◆ **Key**: attribute to uniquely identify an entity in an entity set

◆ Example:

studentID is a key of the STUDENT entity set: since there are no two students having the same value on studentID.

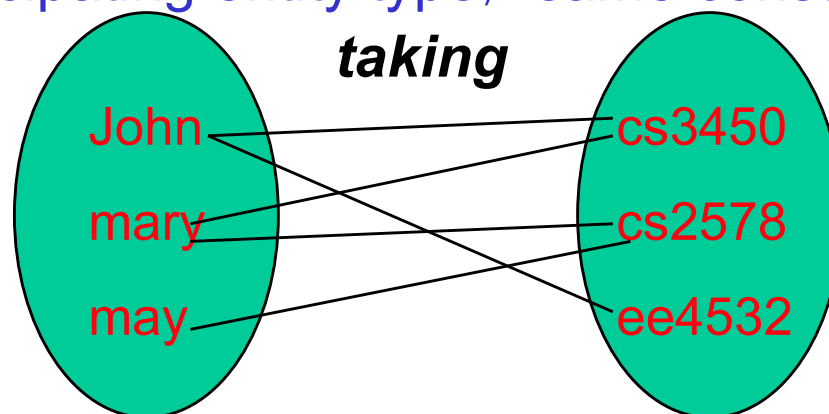
HKID is another key of the STUDENT entity set: since there are no two students having the same value on HKID.

Name is not a key of the STUDENT entity set: since two students may have the same name.

# ER Model Concepts

---

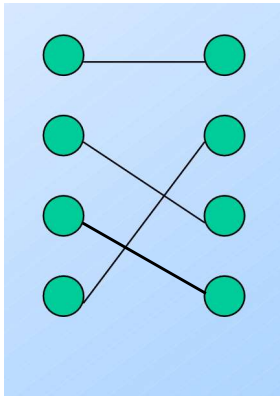
- **Relationship** -- an association among several entities
  - ◆ Example: Patrick and Eva are friends  
Patrick is taking cs3450
- a relationship can carry attributes: properties of the relationship
  - ◆ Example: Patrick takes cs3450 with a grade of B+
- **Relationship Set** -- a set of relationships of the same type (same attribute, same participating entity type, same constraints)
  - ◆ Example:



# Constraints on relationship

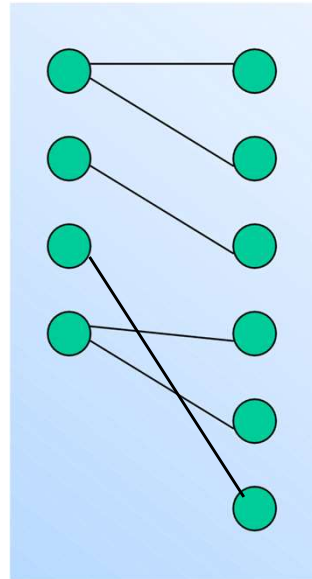
## ◆ Cardinality

COURSE LECTURER



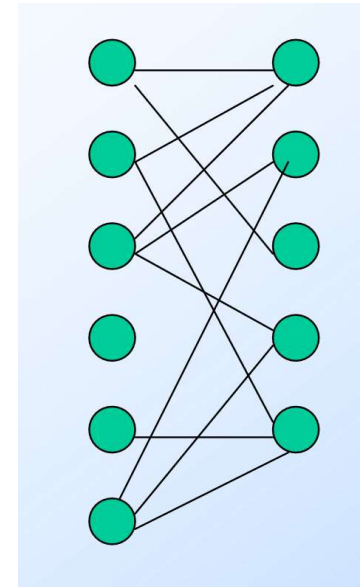
**One-to-one(1:1)**

COURSE TA



**One-to-many  
(1:N)**

STUDENTS COURSE



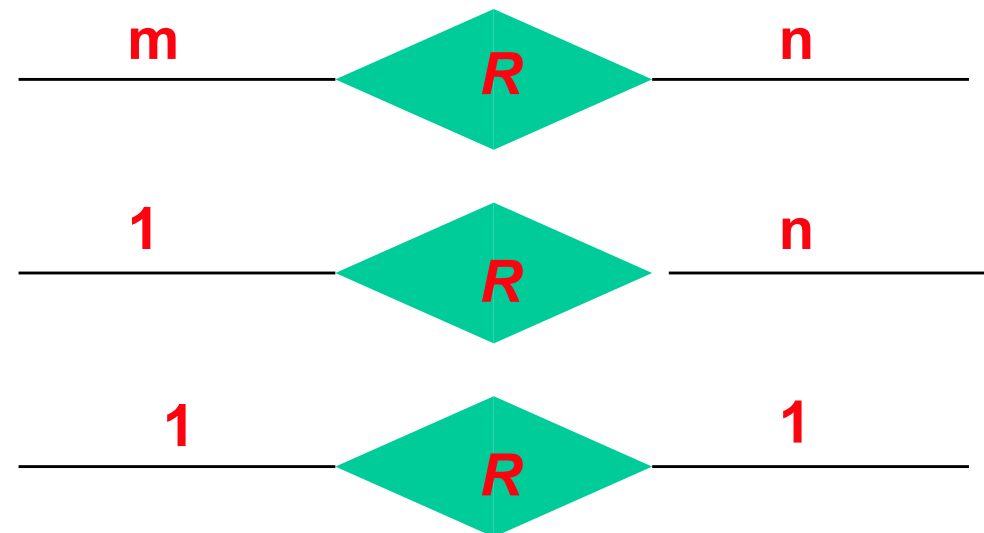
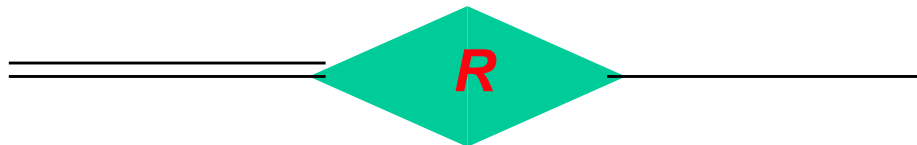
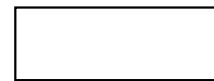
**Many-to-many (M:N)**

◆ Participation: whether every entity in the entity set participates in the relationship set: total v.s. partial

# ER Model Diagram

## ■ ER Diagram

- ◆ Rectangles: Entity Sets
- ◆ Oval: Attributes
- ◆ Diamonds: Relationship Sets
- ◆ Lines: Attributes to Entity/Relationship Sets  
or, Entity Sets to Relationship Sets





# Example COMPANY Database

---

- We need to create a database schema (definition) based on the following (simplified) application requirements of the COMPANY Database:
  - ◆ The company is organized into DEPARTMENTS
  - ◆ Each DEPARTMENT has a unique name, unique number, several locations, and an EMPLOYEE who *manages* the DEPARTMENT
  - ◆ We keep track of the start date of the department manager.
  - ◆ Each DEPARTMENT may *control* a number of PROJECTs
  - ◆ Each PROJECT has a unique name, unique number and is located at a single location, and many EMPLOYEEs working on it.

# *Example COMPANY Database*

---

- The database will store each EMPLOYEE's social security number (ssn), name(first name, last name and middle init), address, salary, sex, and birthdate
  - ◆ Each EMPLOYEE *works for* one DEPARTMENT *but work on* several PROJECTs
  - ◆ The DB will keep track of the number of hours per week that an employee currently works on each project
  - ◆ Each EMPLOYEE, except for the boss, is *supervised* by his/her direct supervisor (another EMPLOYEE)
- Each EMPLOYEE may *have* a number of DEPENDENTS
  - ◆ For each dependent, the DB keeps a record of name, sex, birthdate, and relationship to the employee

# *Initial Conceptual Design of Entity Sets for the COMPANY Database Schema*

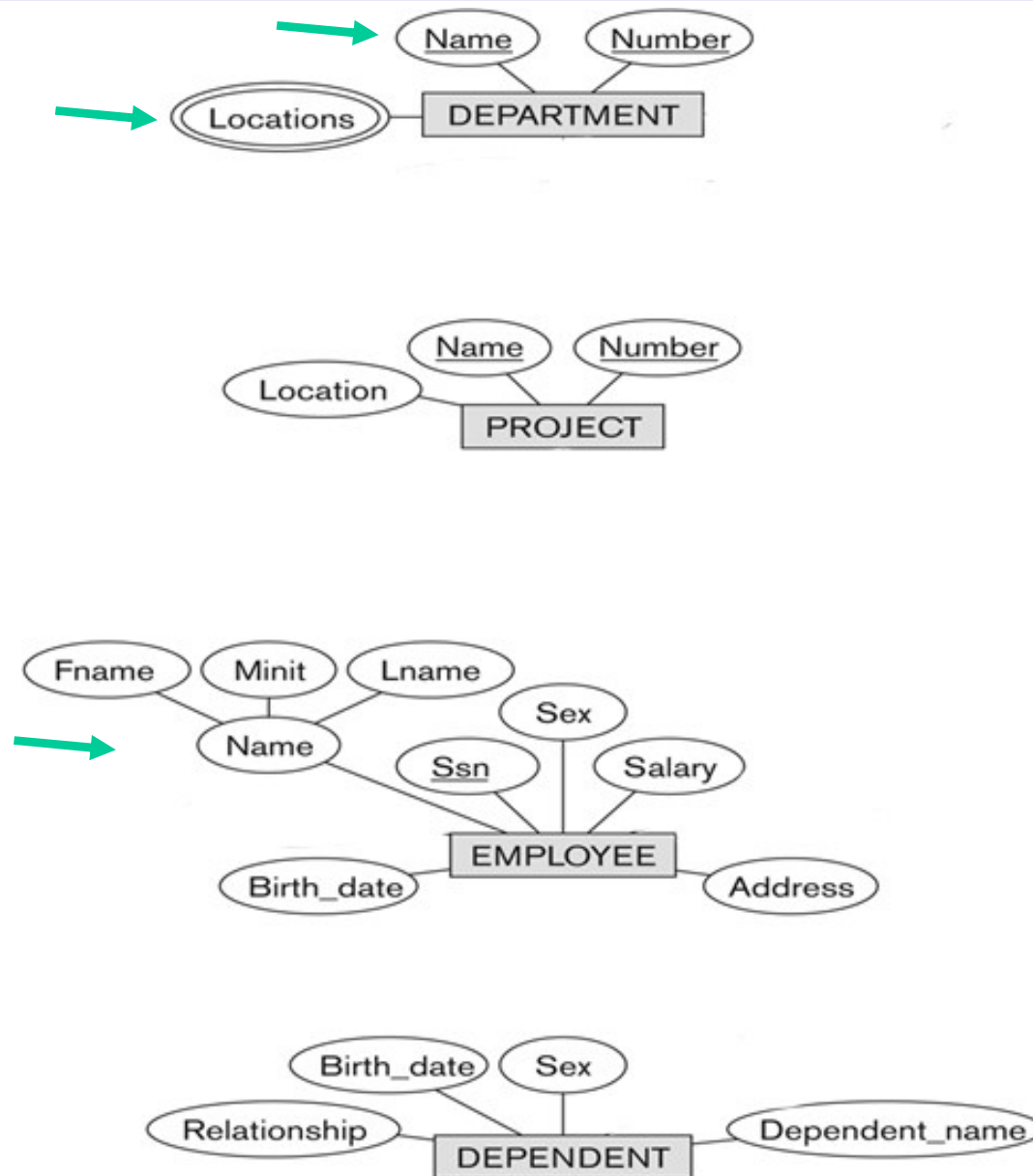
---

- Based on the requirements, we can identify four initial entity sets in the COMPANY database:
  - ◆ DEPARTMENT
  - ◆ PROJECT
  - ◆ EMPLOYEE
  - ◆ DEPENDENT
- The initial attributes shown are derived from the requirements description:

# Initial Design of Entity Sets:

## EMPLOYEE, DEPARTMENT, PROJECT, DEPENDENT

---



**Figure 3.8**

Preliminary design of entity types for the COMPANY database. Some of the shown attributes will be refined into relationships.

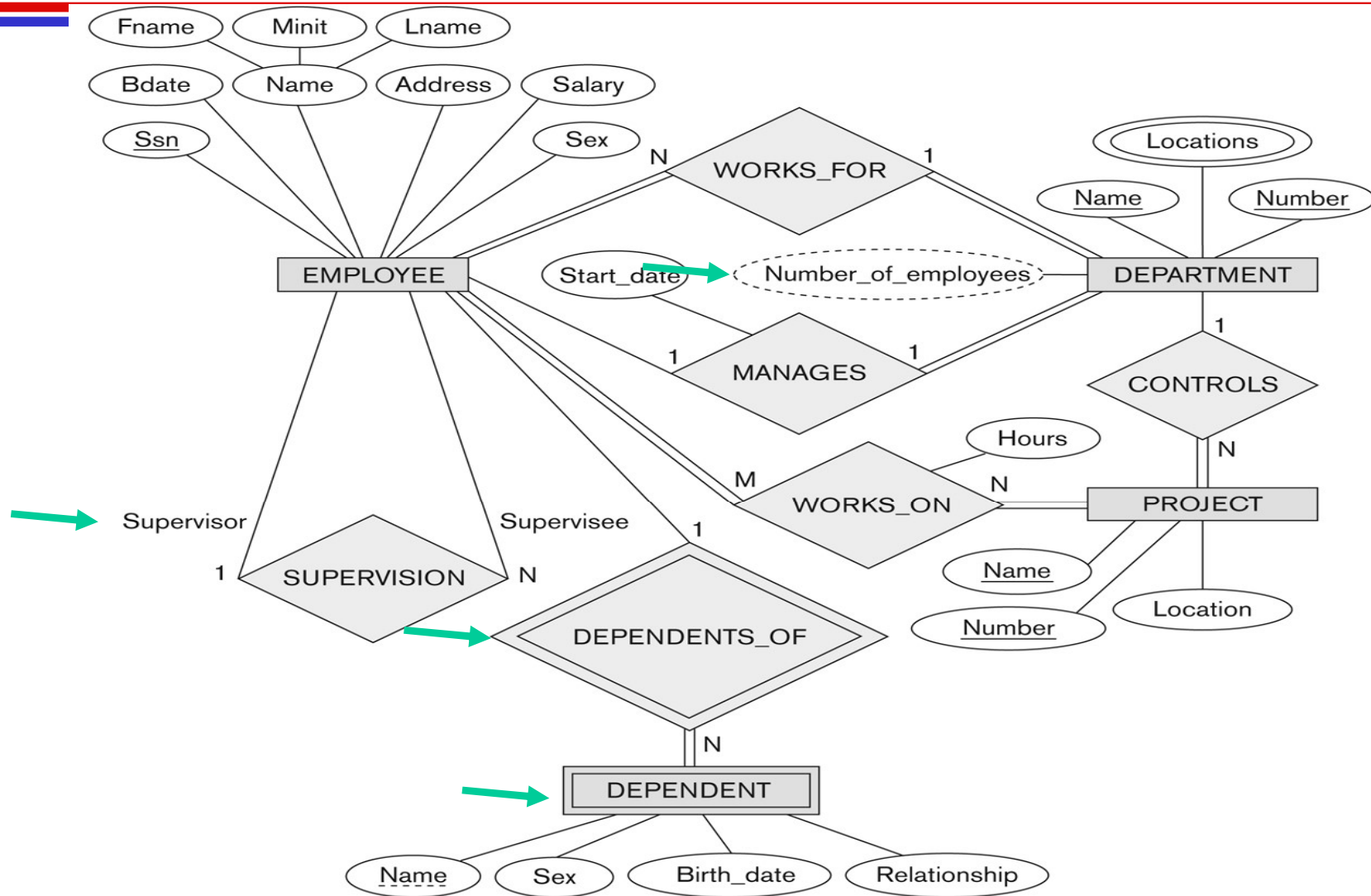
# *Refining the initial design by introducing relationships*

---

- Find **relationships** relating two or more distinct entities/entity types with a specific meaning
- By examining the requirements, six relationship types are identified:
  - ◆ EMPLOYEE *works on* PROJECT
  - ◆ EMPLOYEE *works for* DEPARTMENT
  - ◆ EMPLOYEE *manages* DEPARTMENT
  - ◆ EMPLOYEE *supervises* EMPLOYEE
  - ◆ DEPENDENT *depends on* EMPLOYEE
  - ◆ DEPARTMENT *controls* PROJECT

# ER DIAGRAM – Relationship Types are:

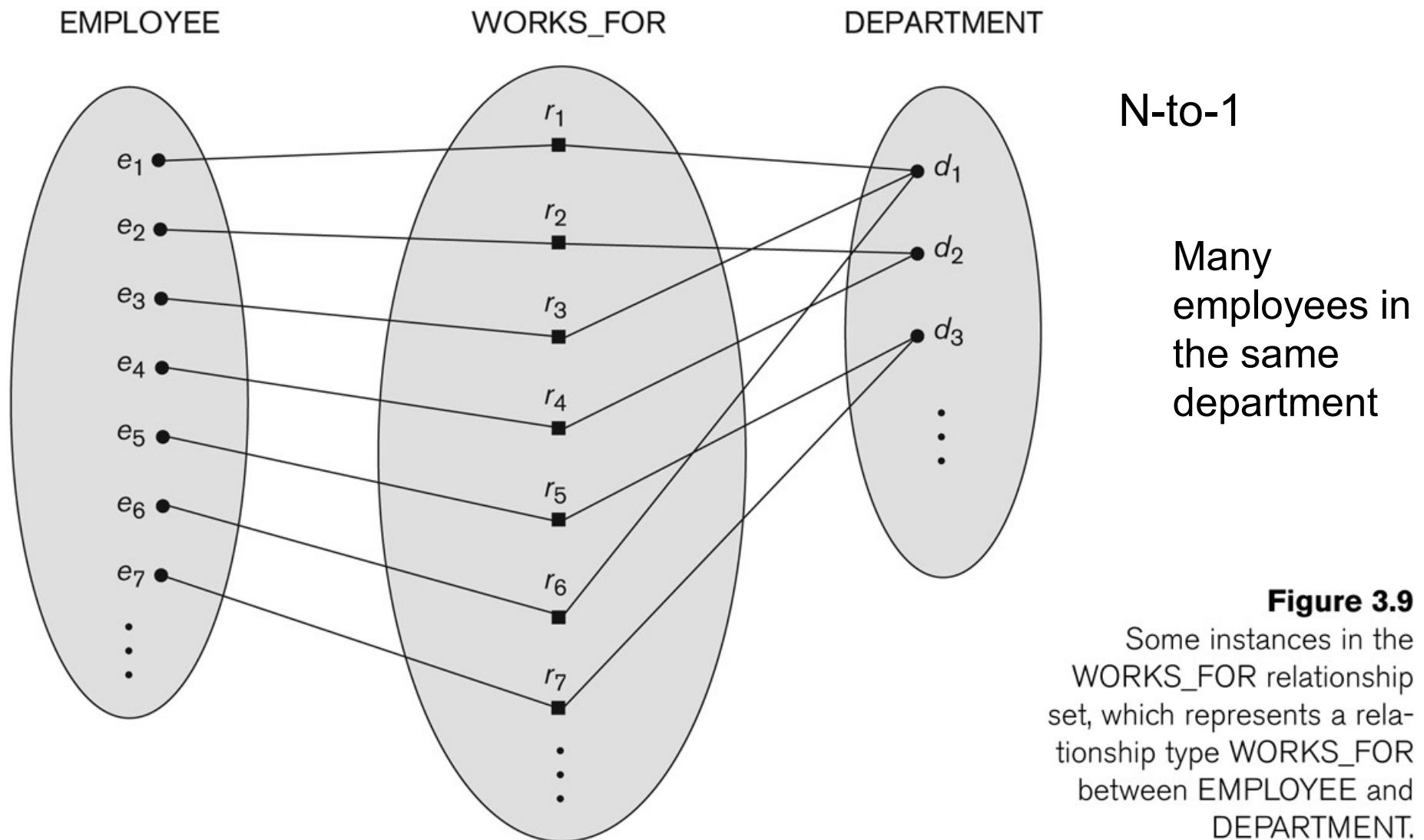
WORKS\_FOR, MANAGES, WORKS\_ON, CONTROLS, SUPERVISION, DEPENDENTS\_OF



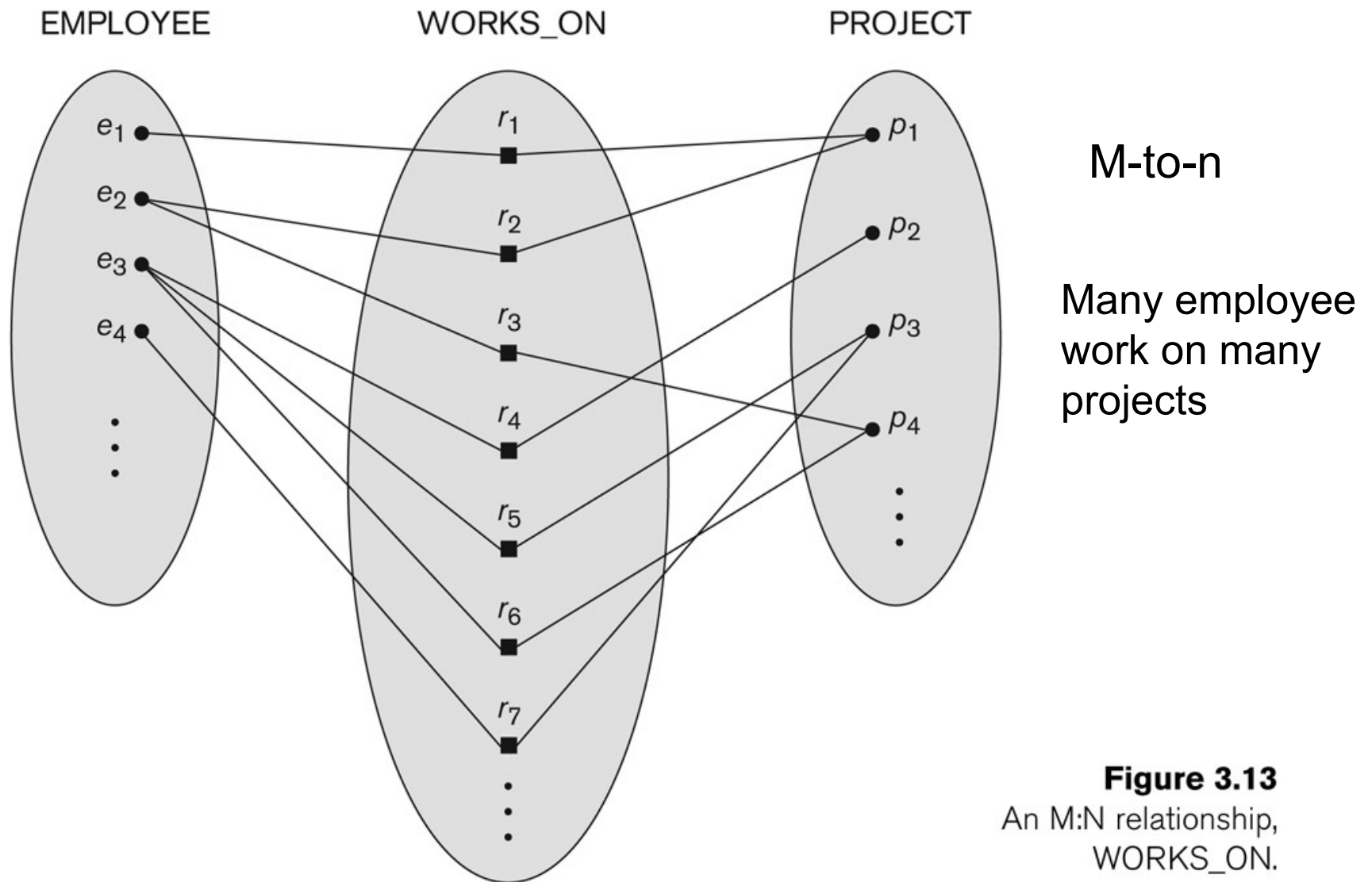
**Figure 3.2**

An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter.

# *Relationship instances of the WORKS\_FOR N:1 relationship between EMPLOYEE and DEPARTMENT*

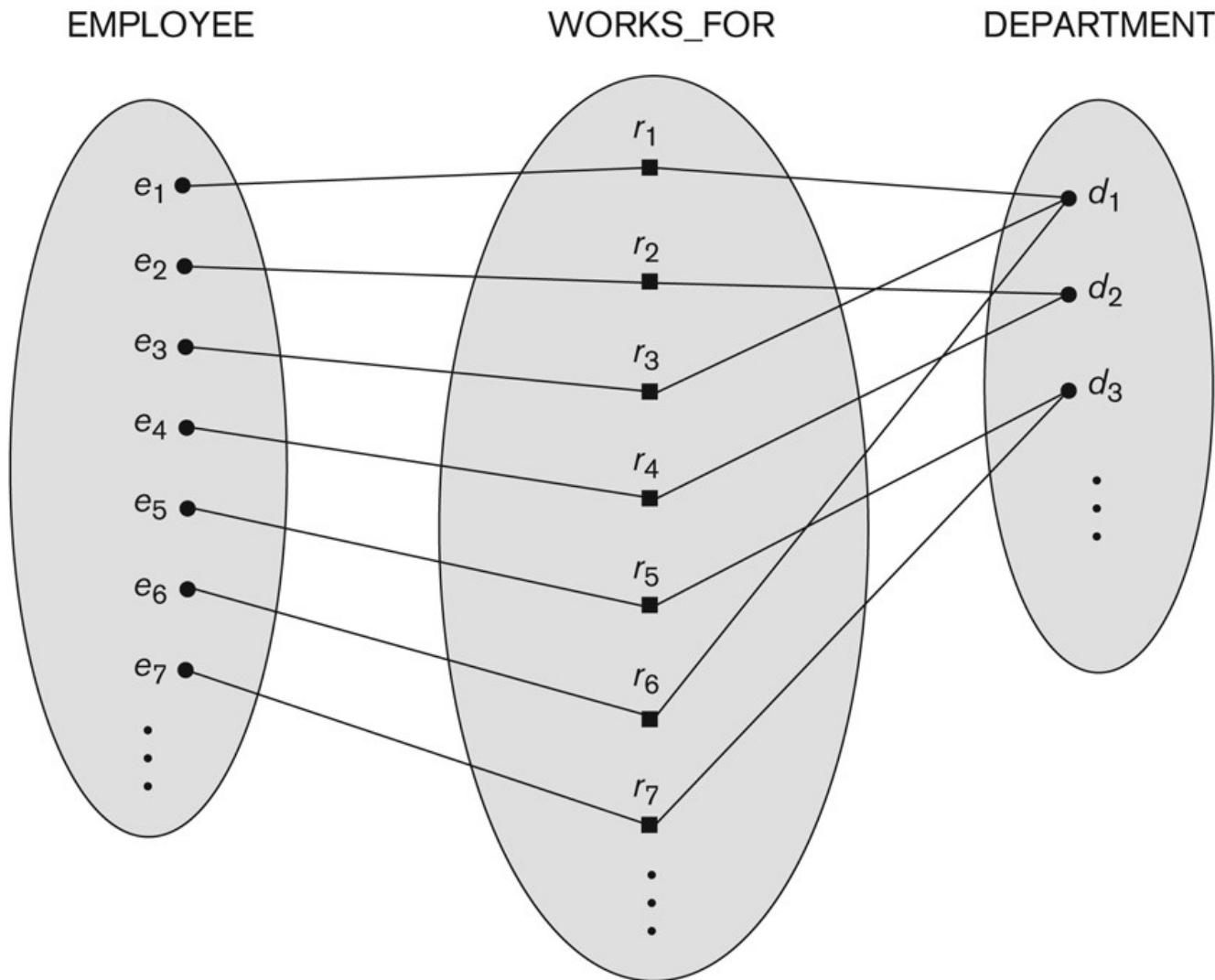


# *Relationship instances of the M:N WORKS\_ON relationship between EMPLOYEE and PROJECT*





# Many-to-one (N:1) Relationship



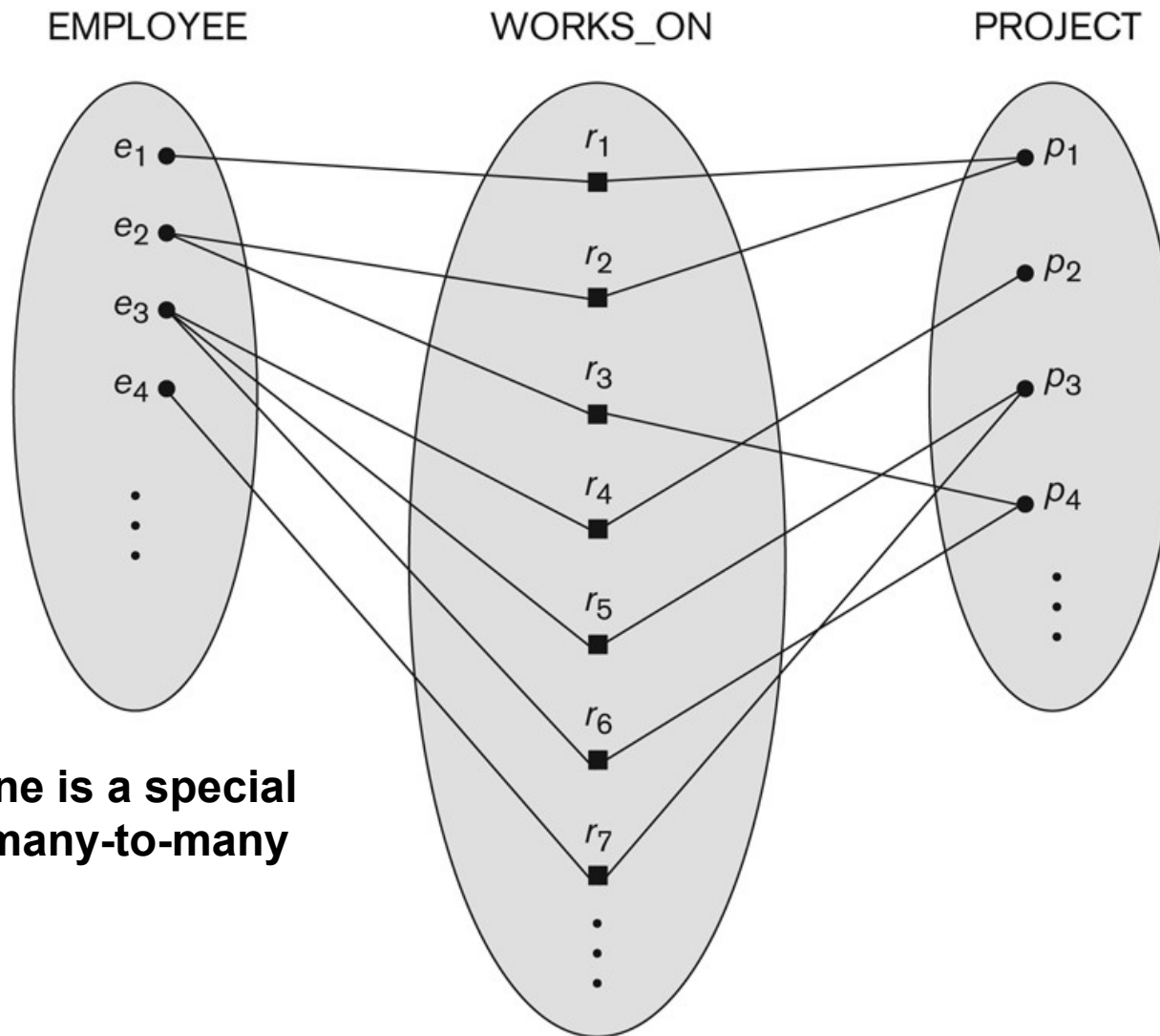
**Select anyone in  
EMPLOYEE, one can be  
identified in DEPARTMENT**

**One-to-one is special case  
of Many-to-one**

**Figure 3.9**

Some instances in the  
WORKS\_FOR relationship  
set, which represents a rela-  
tionship type WORKS\_FOR  
between EMPLOYEE and  
DEPARTMENT.

# Many-to-many (M:N) Relationship



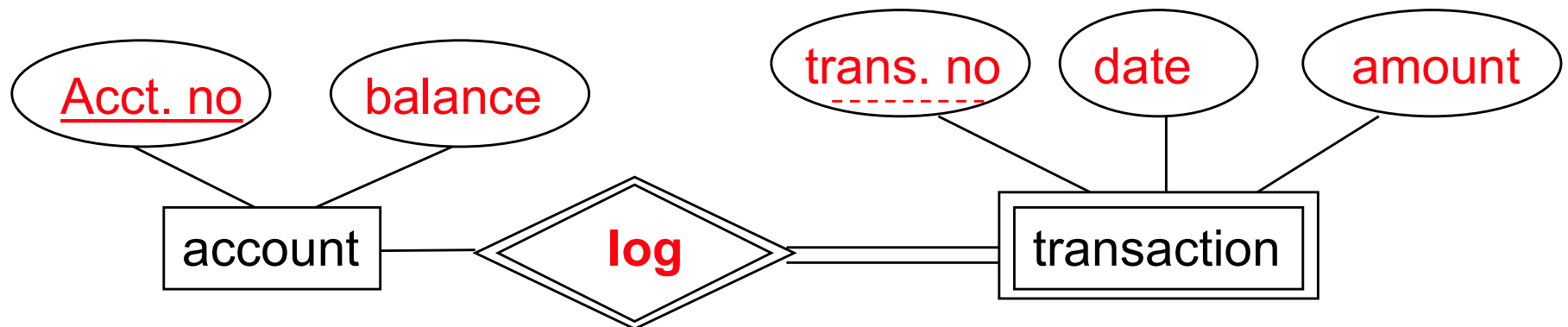
**Many-to-one is a special case of a many-to-many**

**Figure 3.13**  
An M:N relationship,  
WORKS\_ON.

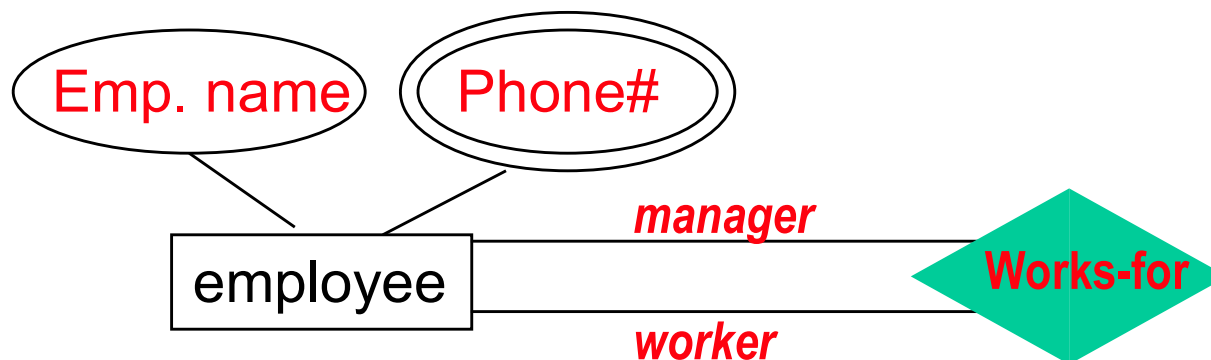
# ER Model Diagram

## ■ Weak Entity Set

◆ an entity set that does NOT have enough attributes to form a key



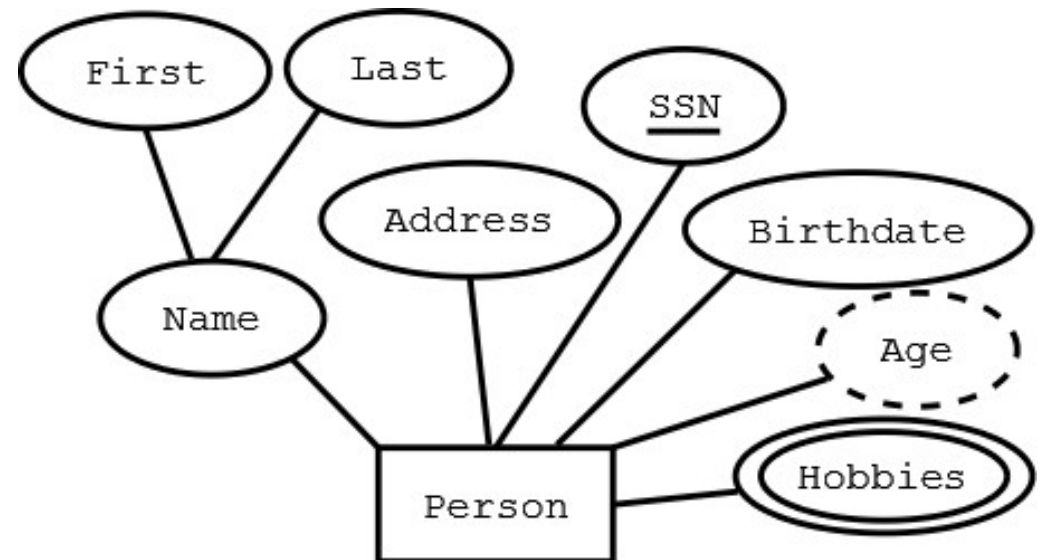
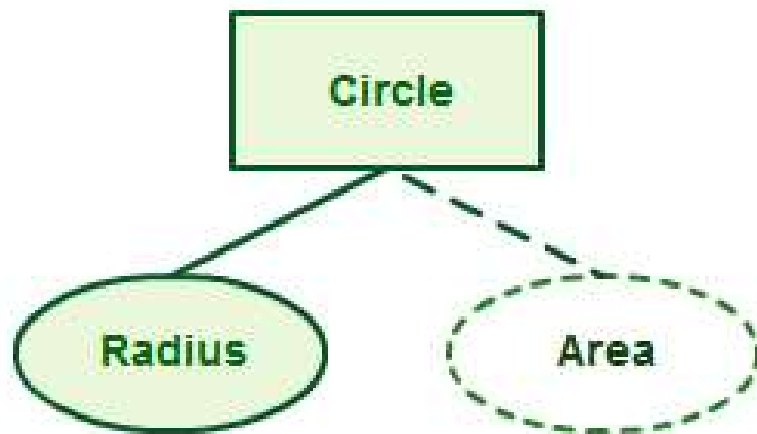
## ■ Role Indicators




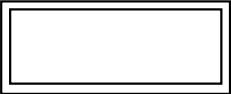

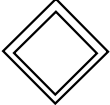



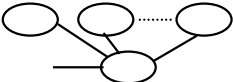

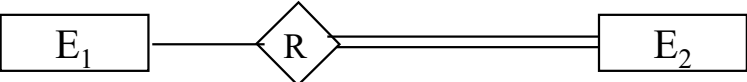
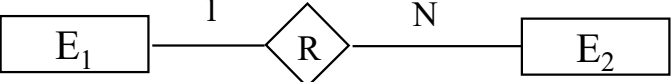
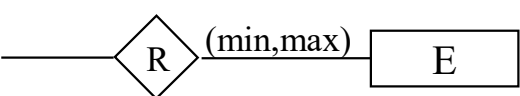
# ER Model Diagram

---

- **Derived Attribute:** An attribute which can be derived from other attributes is known as **derived attribute**.



# Summary of ER-Diagram Notation

Symbol	Meaning
	ENTITY TYPE
	WEAK ENTITY TYPE
	RELATIONSHIP TYPE
	IDENTIFYING RELATIONSHIP TYPE
	ATTRIBUTE
	KEY ATTRIBUTE
	MULTIVALUED ATTRIBUTE
	COMPOSITE ATTRIBUTE
	DERIVED ATTRIBUTE
	TOTAL PARTICIPATION OF E <sub>2</sub> IN R
	CARDINALITY RATIO 1:N FOR E <sub>1</sub> :E <sub>2</sub> IN R
	STRUCTURAL CONSTRAINT (min, max) ON PARTICIPATION OF E IN R

# Summary

---

- ◆ Concepts of database system, DBMS, data abstraction, Data model.
- ◆ The ER Model is regarded as the 1st “conceptual/semantic” model centered around relationships, not attributes
- ◆ It combines successfully the best features of the previous data models
- ◆ simple and easy to understand
- ◆ can be mapped to tables (relational model) in a straightforward manner  
*(to be studied in the coming lecture series)*