資料庫系統
Class 1: Introduction

逢甲資工 許懷中

What is Data

- From Wikipedia
  - Any sequence of **one** or **more** symbols **given meaning** by specific act(s) of interpretation

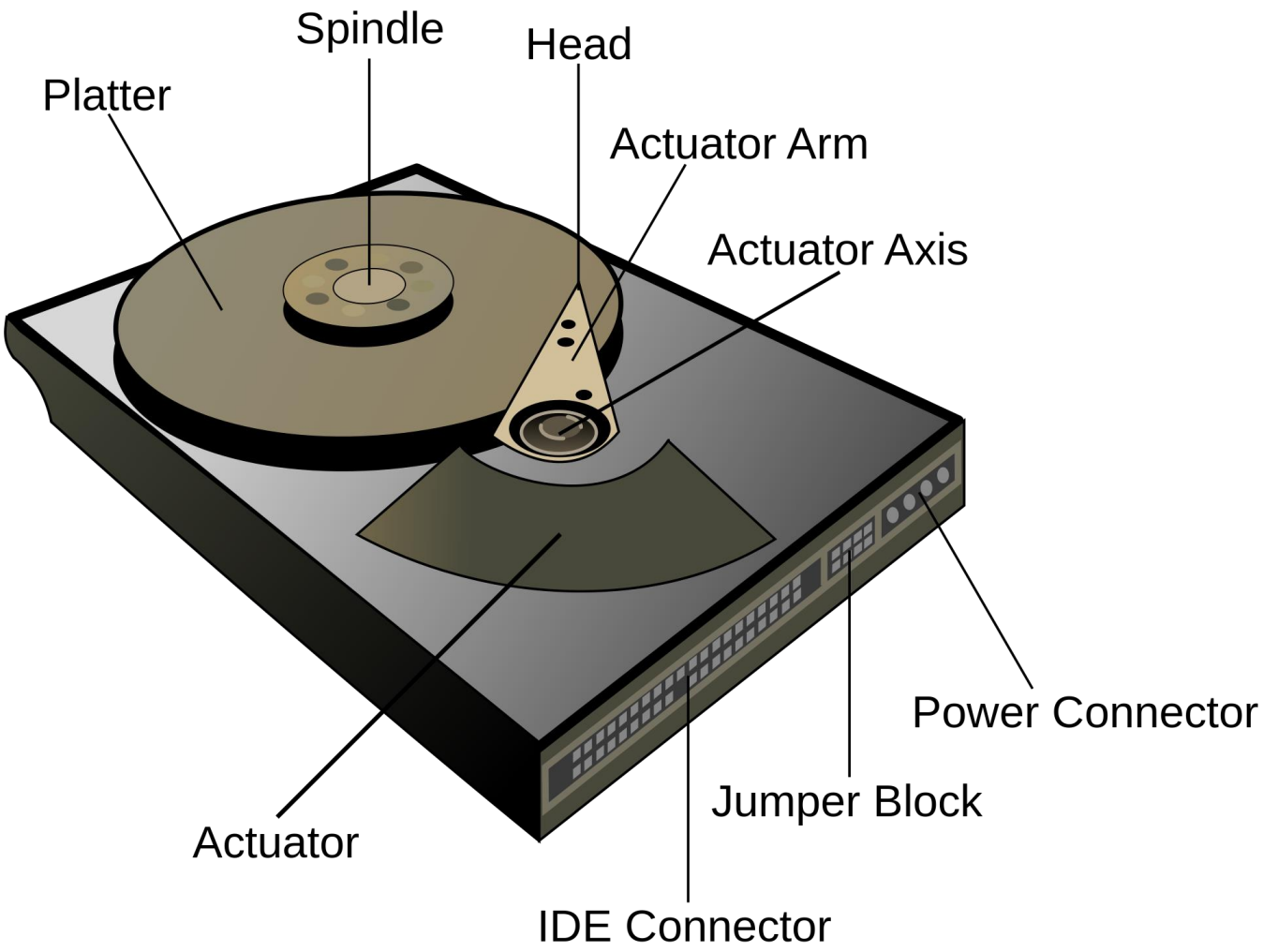
牛肉麵　大滷麵　便當

What is Database

- Collection of data
- Database Management System (DBMS)
  - Collection of **interrelated data**
  - **Set of tools** to **access** the data
  - **Store** and **retrieve** with **convenience** and **efficiency**

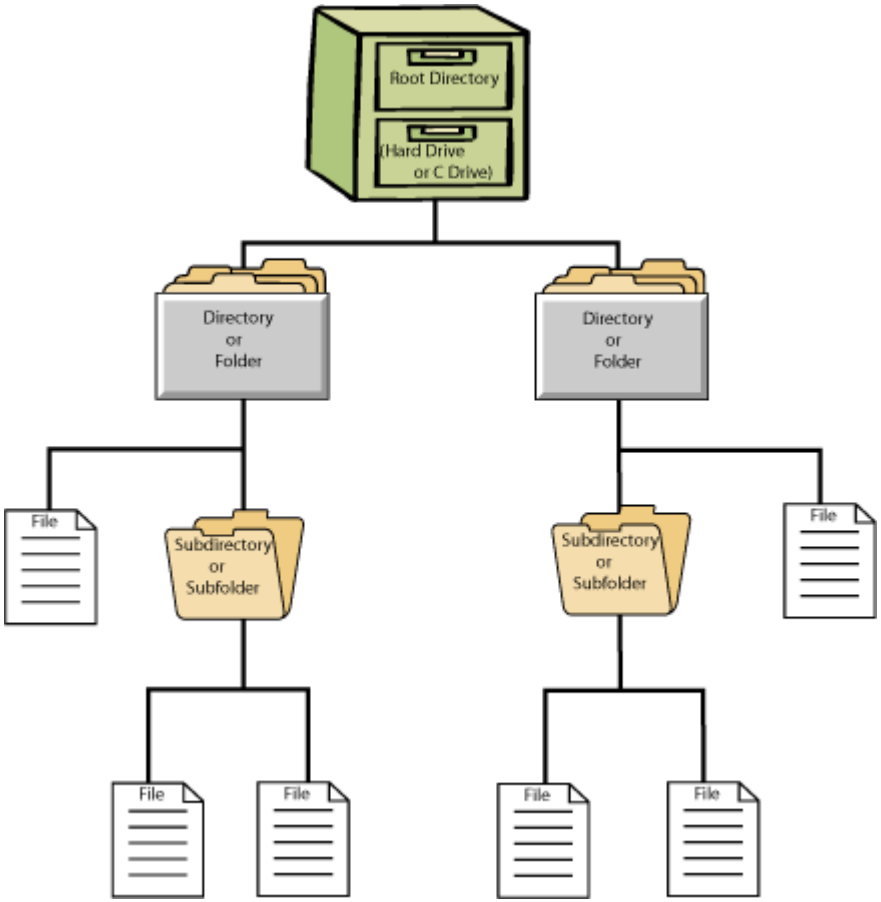
In the early ages

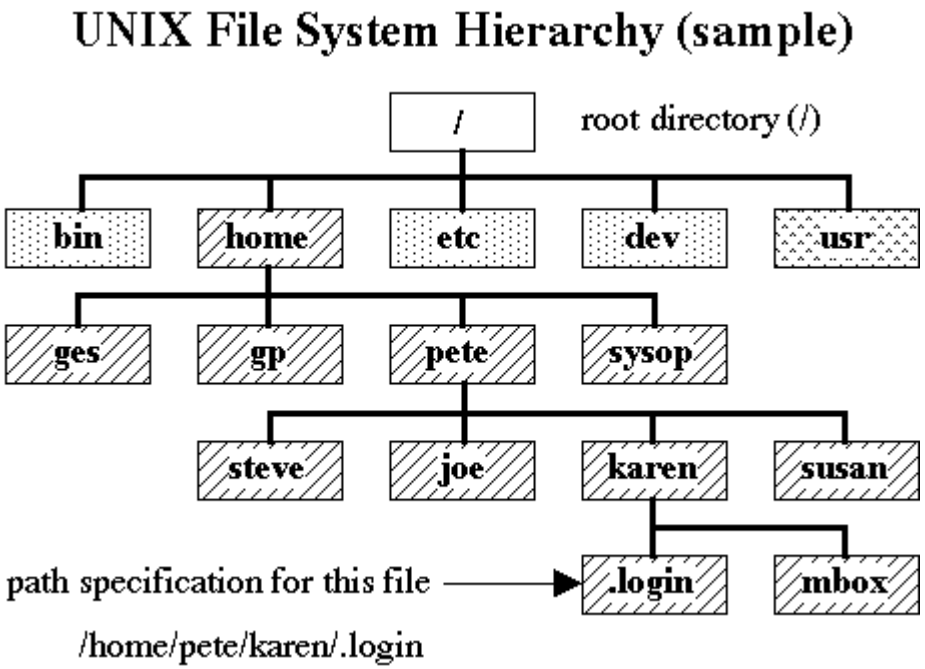
After 1960’


Spindle
Head
Platter
Actuator Arm
Actuator Axis
Power Connector
Jumper Block
Actuator
IDE Connector

After 2006

File System


Root Directory
(Hard Drive
or C Drive)
Directory
or
Folder
Directory
or
Folder
File
Subdirectory
or
Subfolder
Subdirectory
or
Subfolder
File
File
File
File
File


UNIX File System Hierarchy (sample)
/
root directory (/)
bin
home
etc
dev
usr
ges
gp
pete
sysop
steve
joe
karen
susan
.login
mbox
path specification for this file
/home/pete/karen/.login

Database Applications

- Banking: transactions
- Airlines: reservations, schedules
- Universities: registration, grades
- Sales: customers, products, purchases
- Online retailers: order tracking, customized recommendations
- Manufacturing: production, inventory, orders, supply chain
- Human resources: employee records, salaries, tax deductions
- …

University Example

- Students and instructors
- Departments
- Courses
- Actions
  - Assign grades to students
  - Compute GPA
- Organizing the data on file systems

Why DBMS is Necessary

- File system is not enough!
- Redundancy and inconsistency
  - Multiple file formats
  - Duplication among different files
- Difficulty in accessing data
  - Gathering data separated in different files
  - Gathering data with certain constraints
- Integrity problems
  - Implicit integrity constraints
  - Difficulty in adding new constraints

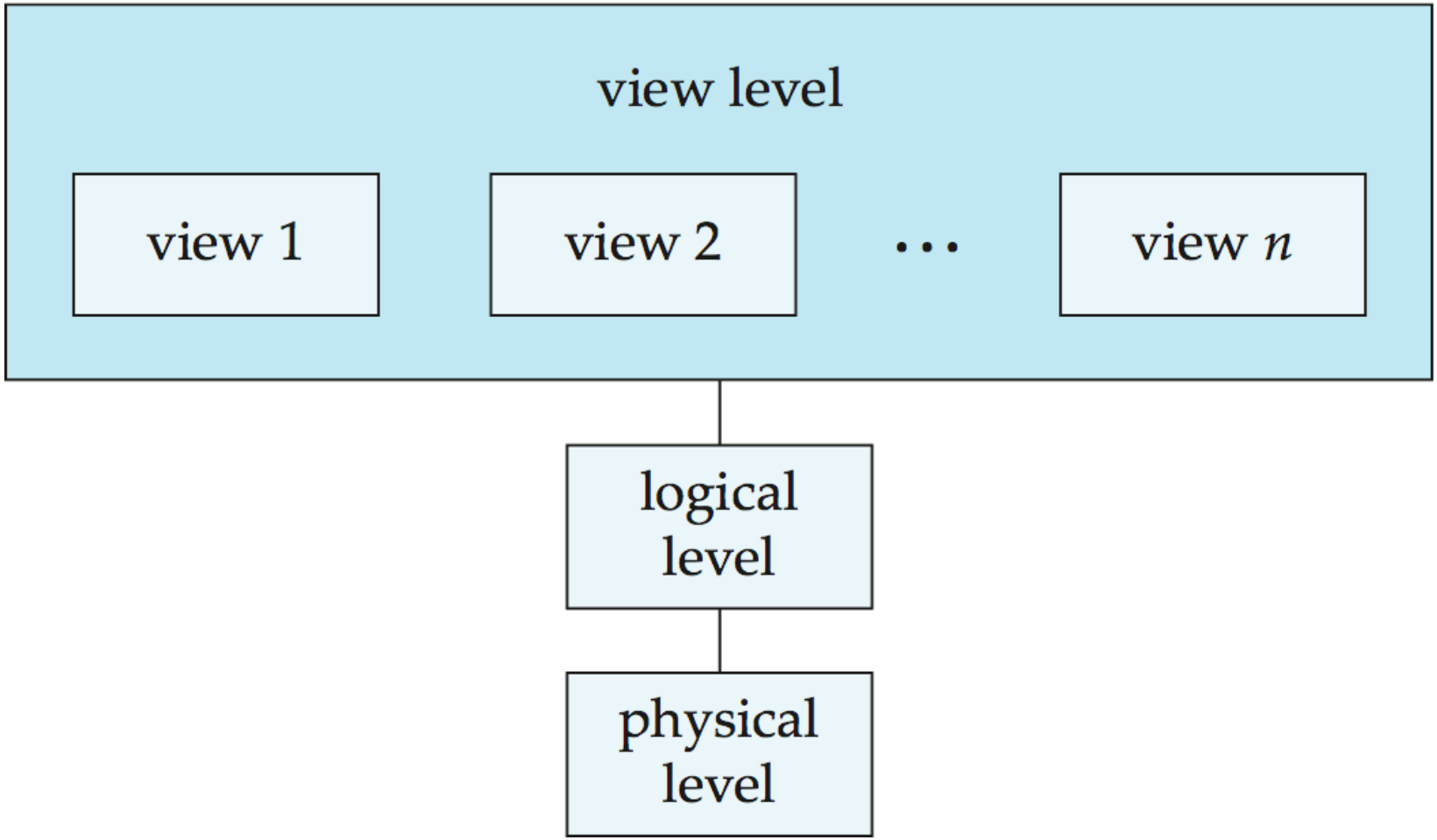
Why DBMS is Necessary (cont.)

- Atomicity of updates
  - Inconsistent state leaded by failure
  - Transfer of funds from one account to another needs being either completed or not happened
- Concurrent access by multiple users
  - Performance requirement
  - Inconsistency leaded by concurrent access
- Security problem
  - Authorization of data access

How DBMS Works

- Data Abstraction
  - Physical level
    - How data are stored
  - Logical Level
    - What data are stored
    - Relationships among data
    - **Physical data independence**
  - View Level
    - Different aspect of data
    - Partial description

Data Abstraction



Schemas & Instances

- Schema
  - Overall design of database
    - Customer profile, account information, and relationships among them
  - Physical schema
  - Logical schema
- Instances
  - Actual content of the database at a particular point of time
- Schema vs. Instances => Class vs. Object

Physical Data Independence

- Modification in physical schema without changing the logical schema
- Applications depend only on logical schema
- Separation of concerns!

Data Models

- A Collection of tools for describing
  - Data
  - Relationships among data
  - Semantics
  - Constraints

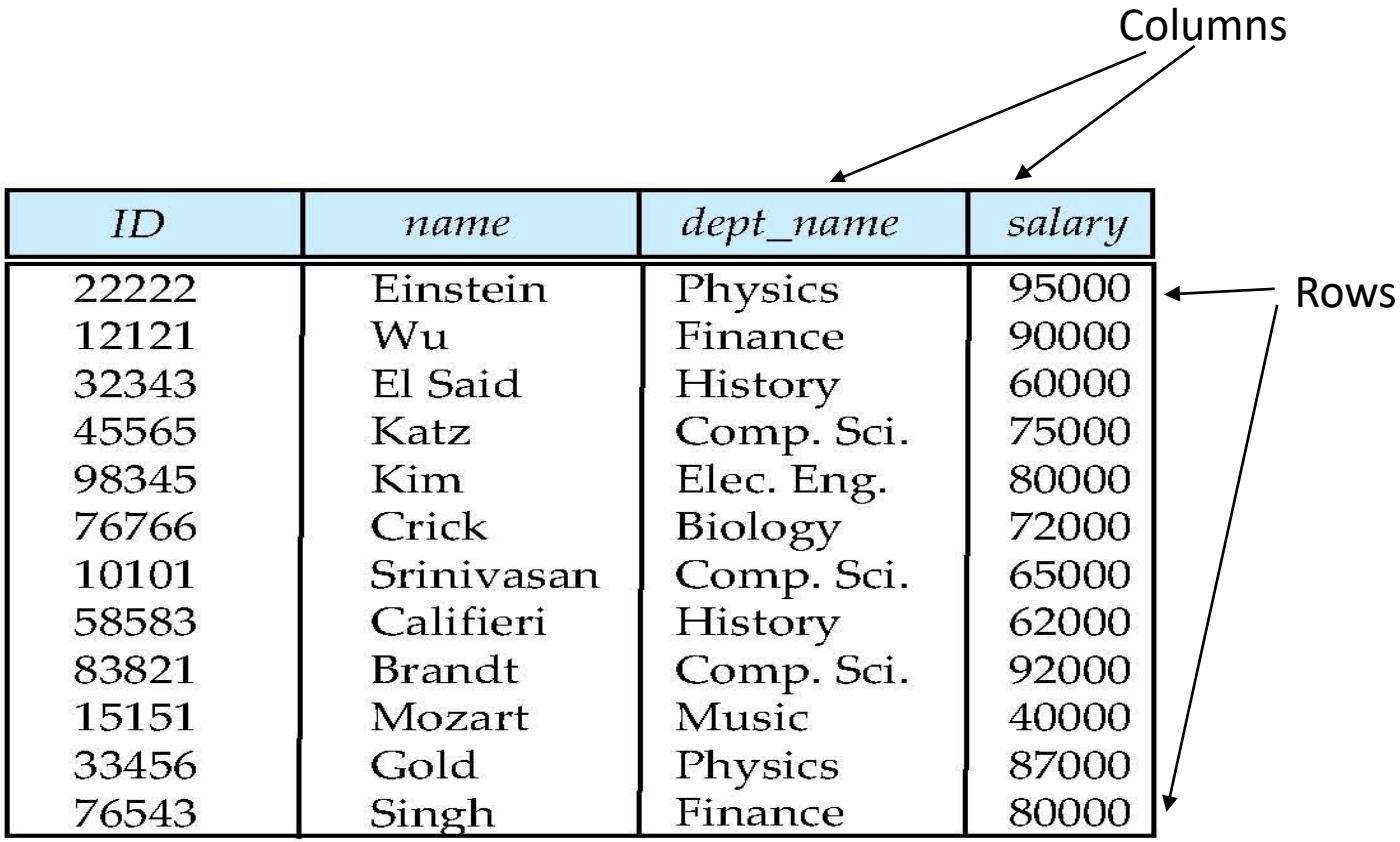
Data Models

- Relational model
- Entity-Relationship data model (E-R Model)
- Object-based data model
- Semi-structured data model
- Other older models
  - Network model
  - Hierarchical model

Relational Model

- Collection of tables
  - Representing both data and relationships among the data
  - Table is known as relations
  - Structured data

Relational Model (cont.)



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

(a) The *instructor* table

Relational Model (cont.)

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

(a) The *instructor* table

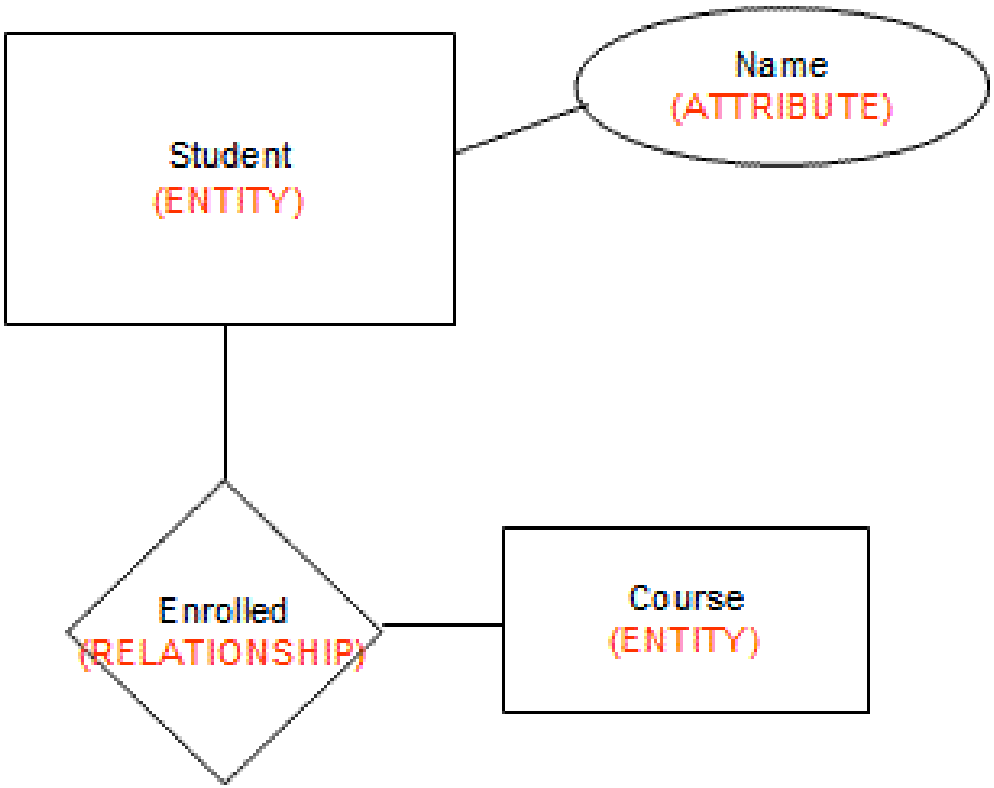
dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

Entity-Relation Model

- Collection of entities and relationships
  - Entity is a thing or object
  - Attributes describe the detail of an entity
  - Entities are connected with relationships
  - Mainly used for database design

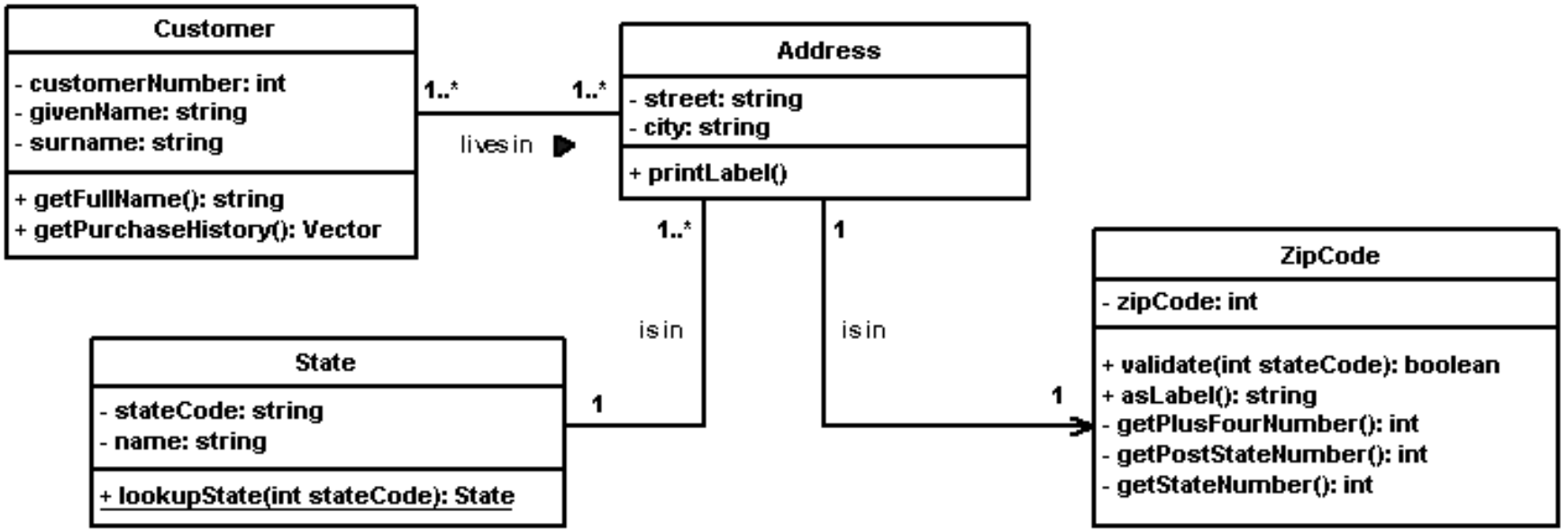
Entity-Relation Model



Object-based Data Model

- Originated from object-oriented programming
- Extended from E-R model
  - Encapsulation
    - Complex types
    - Non-atomic attributes
  - Methods (functions)
  - Object identity

Object-based Data Model



Semi-structured Data Model

- Data of the same type are allowed having different set of attributes
- Extensible Markup Language (XML)
- NoSQL
  - Not only SQL

```
<!-- bookstore.xml -->
- <bookstore>
  - <book ISBN="0123456001">
      <title>Java For Dummies</title>
      <author>Tan Ah Teck</author>
      <category>Programming</category>
      <year>2009</year>
      <edition>7</edition>
      <price>19.99</price>
    </book>
  - <book ISBN="0123456002">
      <title>More Java For Dummies</title>
      <author>Tan Ah Teck</author>
      <category>Programming</category>
      <year>2008</year>
      <price>25.99</price>
    </book>
  - <book ISBN="0123456010">
      <title>The Complete Guide to Fishing</title>
      <author>Bill Jones</author>
      <author>James Cook</author>
      <author>Mary Turing</author>
      <category>Fishing</category>
      <category>Leisure</category>
      <language>French</language>
      <year>2000</year>
      <edition>2</edition>
      <price>49.99</price>
    </book>
</bookstore>
```

Summary Quiz

- What composes a DBMS?
- Why file system is insufficient for DBMS?
- Naming three data models you have learned in the course and describe them briefly

Summary

- DBMS is composed of
  - Collection of interrelated data
  - Tools for data access
  - Store and retrieve data with convenience and efficiency
- Drawbacks in using file systems as DBMS
  - Data redundancy and inconsistency
  - Difficulty in accessing data
  - Integrity problems
  - Atomicity problems
  - Concurrent access problems
  - Security problems
- Relational, Entity-relation, object-based, and semi-structured data model

Database Languages

- Data manipulation language (DML)
  - For accessing and manipulating the data organized by a data model
  - Query language
- Data definition language (DDL)
  - Specifying a database schema or additional property of data

Data Manipulation Language

- Two classes of languages
  - **Procedural** – user specifies what data is required and how to get those data
  - **Declarative (nonprocedural)** – user specifies what data is required without specifying how to get those data
- SQL is the most widely used query language

DML Example

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

(a) The *instructor* table

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

(b) The *department* table

DML Example

- Find the name of the instructor with ID 22222

  select *name*
  from *instructor*
  where *instructor.ID* = '22222'

- Find the instructors belonging to the departments with budgets higher than 95000

  select *instructor.ID, department.dept name*
  from *instructor, department*
  where *instructor.dept name= department.dept name* **and** *department.budget* > 95000

Data Definition Language

- Specification notation for defining the database schema

create table *instructor* (

ID **char**(5),
name **varchar**(20),
dept_name **varchar**(20),
salary **numeric**(8,2))

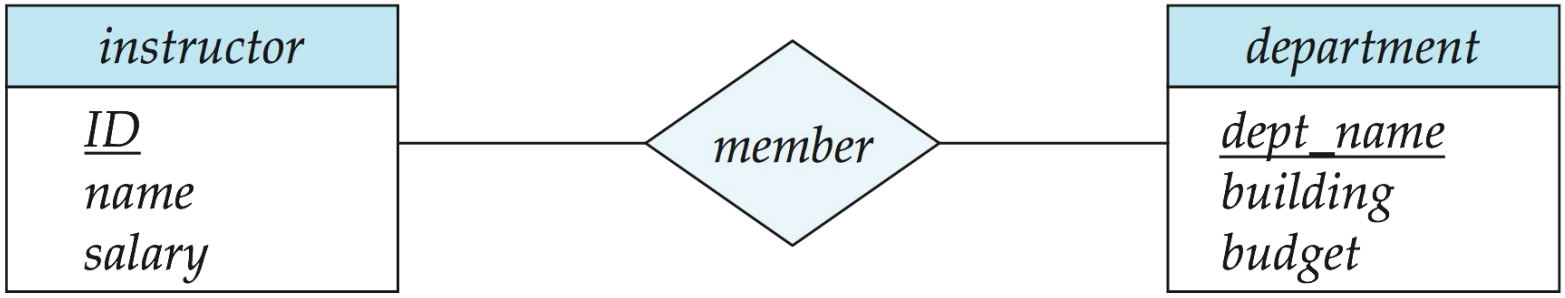
DDL (cont.)

- DDL compiler stores table information in a **data dictionary**
  - Containing metadata (i.e. data about data)
  - Integrity constraints
    - Primary key
      - E.g. unique instructor ID
    - Referential integrity (reference constraints)
      - E.g. *dept_name* in any *instructor* tuple must appear in *department* relation
    - Authorization

Database Design

- What attributes to be captured in the DB
  - Business decision
- How to group the attributes in various table
  - Computer-science problem
    - We want **GOOD** collection of relation schema
  - Entity-Relation model
  - Normalization

DB Design using E-R Model



DB Design using E-R Model

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

(a) The *instructor* table

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

(b) The *department* table

Normalization

ID	*name*	*salary*	*dept_name*	*building*	*budget*
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
83821	Brandt	92000	Comp. Sci	Taylor	100000
15151	Mozart	40000	Music	Packard	80000
33456	Gold	87000	Physics	Watson	70000
76543	Singh	80000	Finance	Painter	120000

Any thing wrong within this design?

Components in DBMS

- Storage manager
- Query processor

Storage Manager

- A program module for interfacing between
  - Low-level data store (physical level)
  - Queries (logical level)
  - Application programs (view level)
- Responsibility
  - Interaction with the file system
  - Efficiency in storing, retrieving, and updating of data
  - Maintaining authorization, integrity constraints, and transactions

Transaction Management

- **A**tomicity
  - Happen in its entirety or not at all
- **C**onsistency
  - Data must be consistent after a transaction
  - E.g. Sum of balance remains equal after transfer of funds between two accounts
- **I**solation
  - The state made by concurrent transactions remains equal if the transactions are executed sequentially
- **D**urability
  - Data must persist despite the possibility of system failure

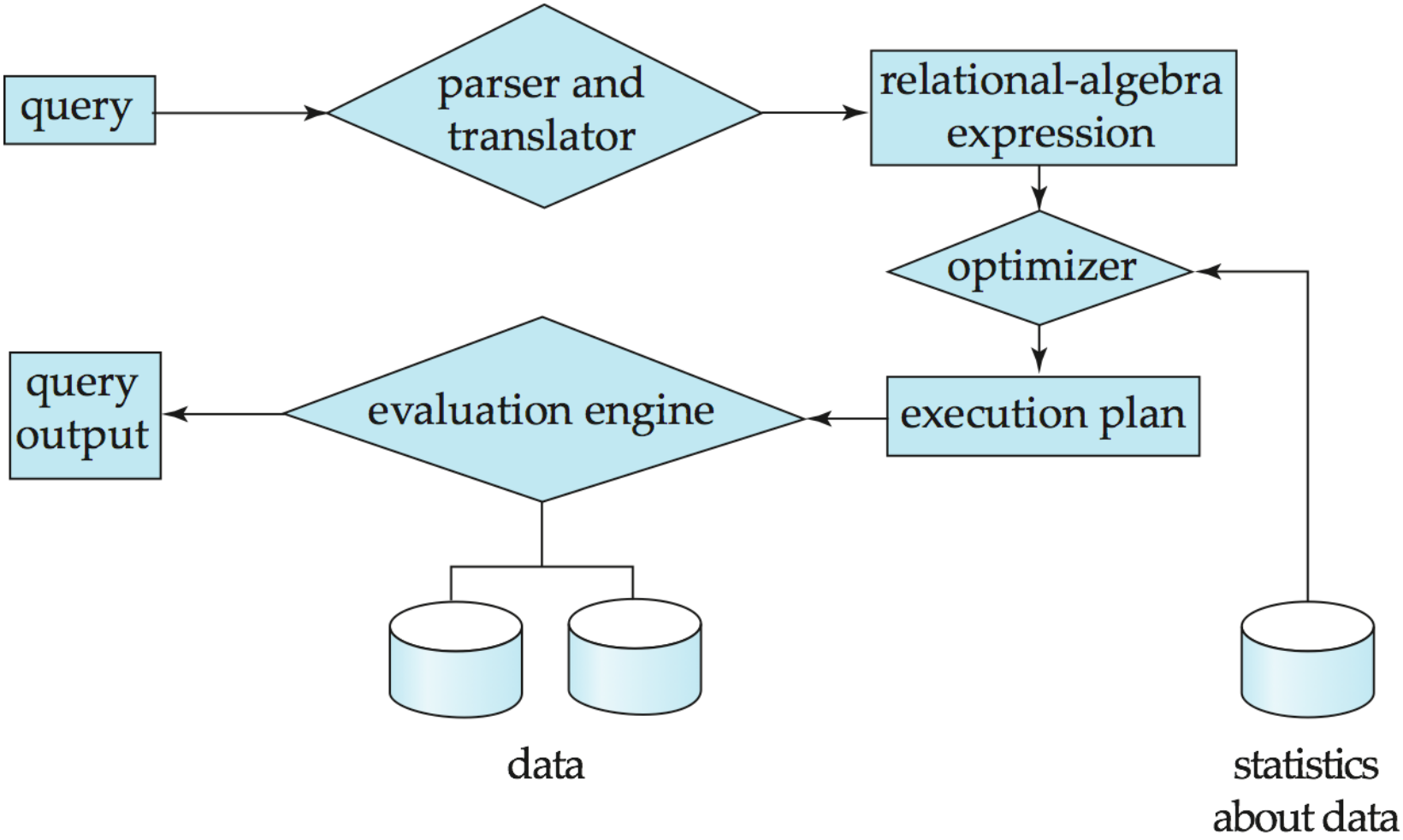
Transaction Management

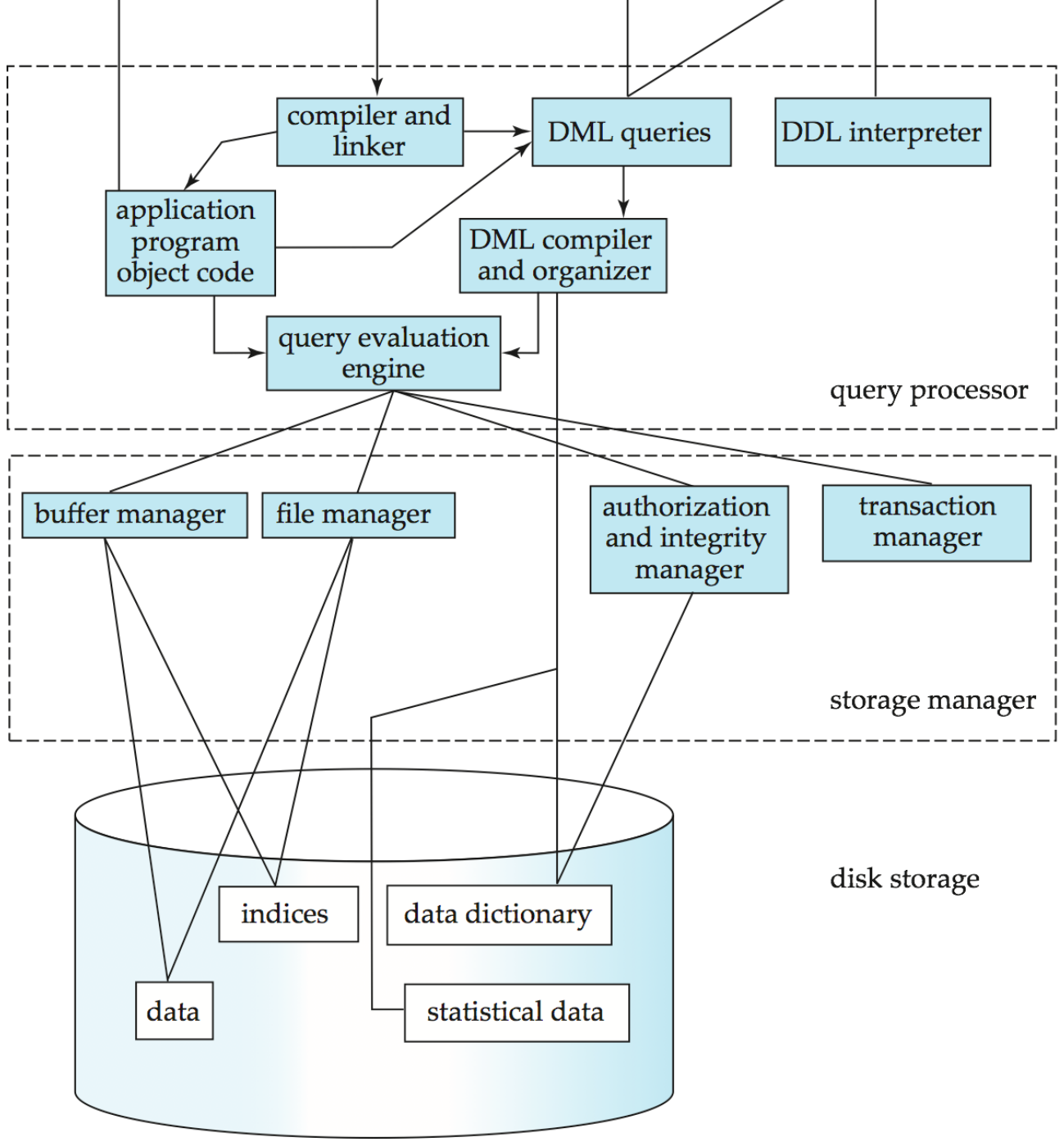
- Recovery manager
  - Failure recovery
- Concurrency-control manager
  - Lock
  - Performance issue

Query Processor

- Parsing and translation
  - DDL interpreter
  - DML compiler
- Optimization
  - Considering statistics of data
  - Considering intermediate results generated by complex operation
- Evaluation
  - Equivalent expressions
  - Different algorithms for each operation

Query Processor (cont.)




compiler and linker
DML queries
DDL interpreter
application program object code
DML compiler and organizer
query evaluation engine
query processor
buffer manager
file manager
authorization and integrity manager
transaction manager
storage manager
indices
data dictionary
data
statistical data
disk storage

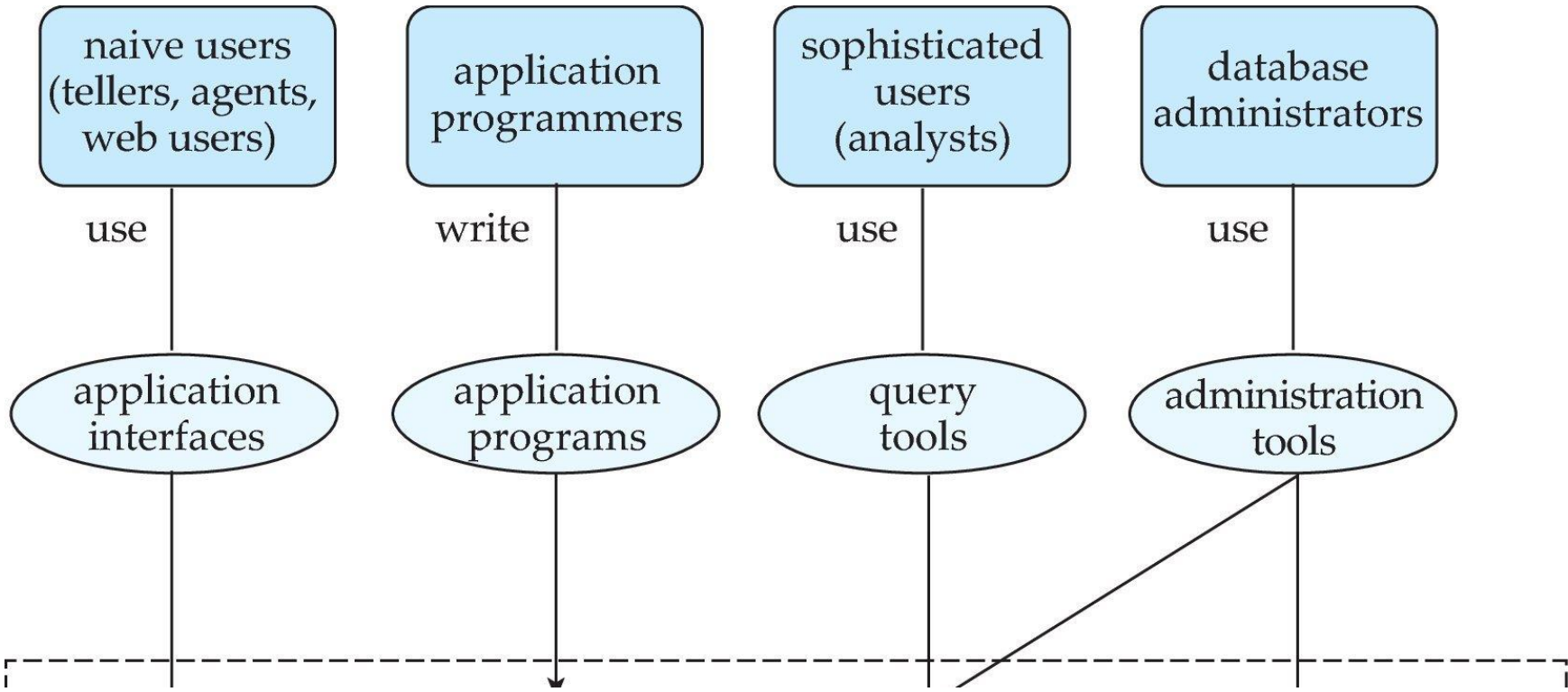
DB Users

- Naïve user
  - Access DB using applications
- Application programmers
  - Programming applications
  - Access DB using API (ODBC, JDBC)
- Sophisticated user
  - Experts
  - Access DB using DML directly

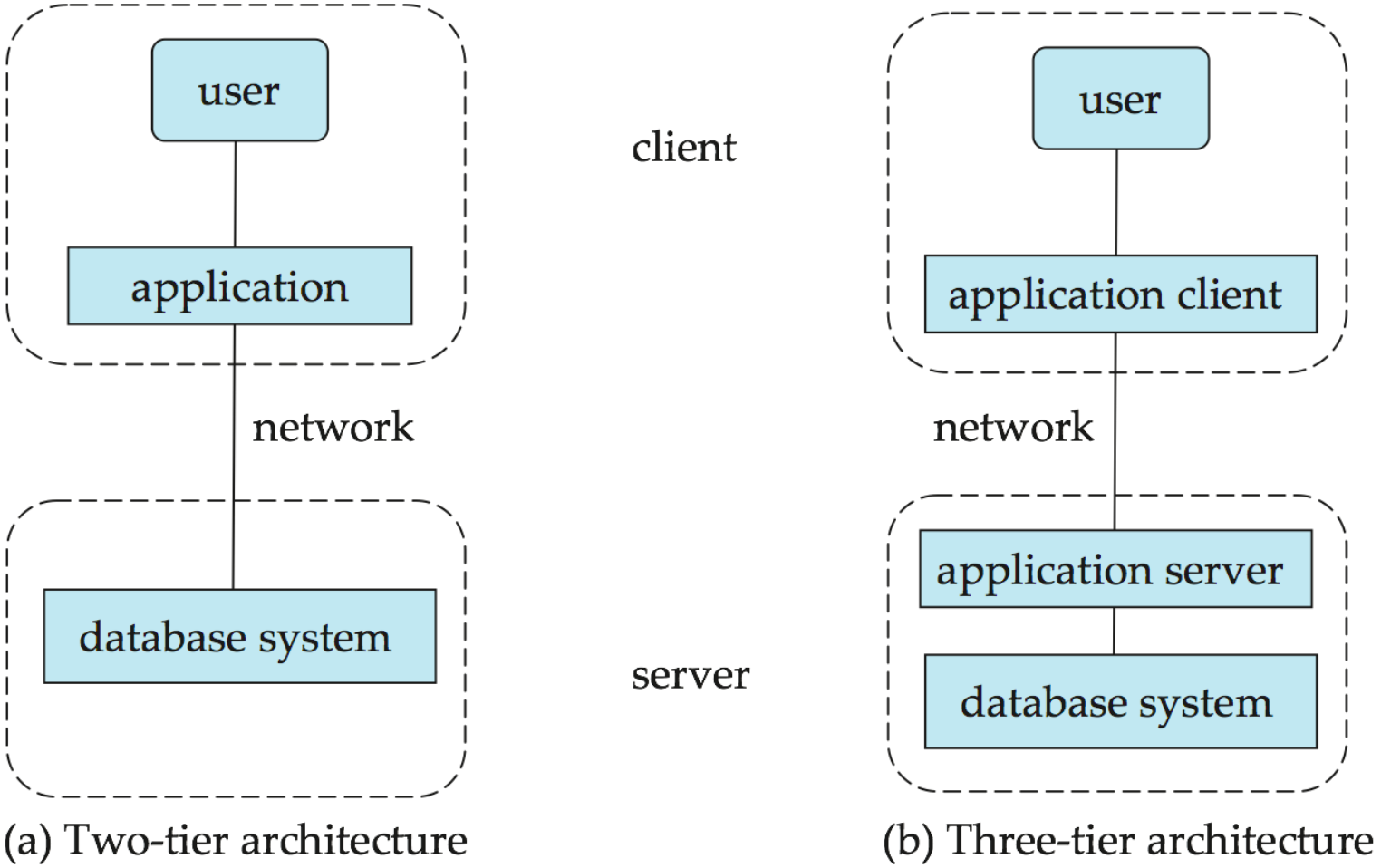
DB Administrator

- Schema definition
- Storage structure and access-method definition
  - Improving performance
- Granting of authorization
- Routine maintenance
  - Backup
  - Ensuring free space
  - Monitoring on job running and ensuring the performance

DB Users and DBA



DBMS Architecture



(a) Two-tier architecture

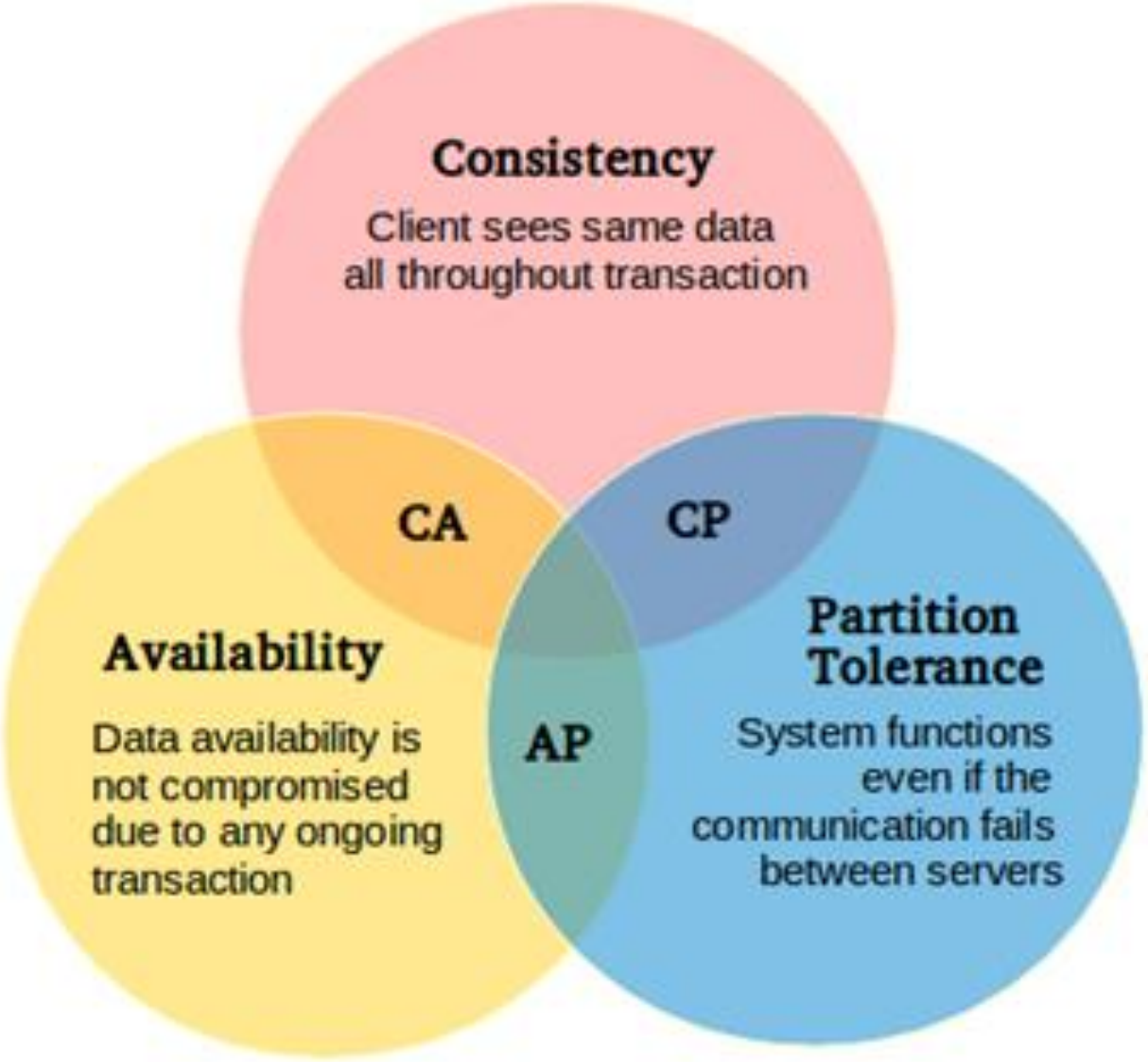
(b) Three-tier architecture

NoSQL DB

- Specialize DB for data analysis
- Extraordinary large volume data
  - Must be distributed
- Ultra fast incoming data
  - Storage performance matters
- Highly varied data
  - Must be semi-structured

NoSQL DB (cont.)

- Data Model
  - Key-value, ordered key-value, graph
  - document-specialized for full-text search
- Usually supporting only partial SQL
  - Preventing Join
- Guarantee only weak consistency or eternal consistency


Consistency
Client sees same data
all throughout transaction
CA
CP
Availability
Data availability is
not compromised
due to any ongoing
transaction
AP
Partition
Tolerance
System functions
even if the
communication fails
between servers

Summary Quiz

- What a bad design of DB may result?
- Briefly describe the process of query processing
- What are the key factors of transaction?
- Why NoSQL DBMS usually yield complete consistency?