SOFT130015
Database Design

Course Information

About Instructor

- 吴毅坚
  - 软件工程实验室，江湾二号交叉学科楼D2014-1
- Email
  - wuyijian@fudan.edu.cn
- 电话/微信→
  - 13817374753
- QQ
  - 2745859 ←
- 欢迎联系

About TA

- 王睿智 23212010029@m.fudan.edu.cn
- 林明生 22210240434@m.fudan.edu.cn
- 刘人豪 22210240031@m.fudan.edu.cn
- 卞凌峰 20110240028@fudan.edu.cn

About the course

- Textbook
  - Database System Concept (7th ed.)
  - English version
- References
  - Database System Concept (6th ed.)
  - Practice-based (Oracle, MySql…)
- Focuses
  - Basic concepts of relational database
  - Application related issues & practical skills
- Assessment
  - 1 Project and 1 Lab
  - After-class assignments and in-class quizzes
  - Final examination (open book, open notes)

About the course (cont.)

- Final score is given according to your overall performance.
  - Practices (PJ, Labs): ~40%
  - Homework, Quizzes: ~10%
  - Final: ~50%
  - Actual percentage may vary.

Lecture 1

Introduction

Content

- Concepts based on the textbook
  - Why database ? An Introduction CH1
  - Relational Model and SQL
    - Relational Model CH2
    - SQL CH3-5
    - Other query languages (briefly)
  - Database Design
    - Entity-Relationship Model CH6
    - Relational Database Design CH7
    - Application Design and Development CH8-9
  - Database System Issues
    - Storage CH12-13
    - Indexing CH14 (CH24)
    - Transaction CH17
    - Query Processing and Optimization Ch15-16

WHY Database?

- Database Management System (DBMS)
  - Collection of interrelated data (usually referred to as the Database)
  - Set of programs to access the data
  - An environment that is both ***convenient*** and ***efficient*** to use
- Database systems are used to manage collections of data that are:
  - Highly valuable
  - Relatively large
  - Accessed by multiple uses and applications, often at the same time
- A modern database system is a complex software system whose task is to manage a large, complex collection of data

WHY Database?

- Database Everywhere
  - Enterprise Information
    - Sales: customers, products, purchases
    - Human resources: employee records, salaries, tax deductions
  - Banking
    - customers, accounts, transactions
  - Manufacturing
    - production, inventory, orders, supply chain
  - Universities
    - registration, grades, students

……

Why Database?

- Managing Data?
  - File System
    - In the early days, database applications were built directly on top of file systems
    - Drawbacks of using file system to store data
      - See next page

Why Database?

- Drawbacks of using file systems to store data
  - Data redundancy and inconsistency
    - Multiple file formats, duplication of information in different files
  - Difficulty in accessing data
    - Need to write a new program to carry out each new task
  - Data isolation — multiple files and formats
  - Integrity problems
    - Integrity constraints (e.g., account balance > 0) become "buried" in program code rather than being stated explicitly
    - Hard to add new constraints or change existing ones

Why Database?

- Drawbacks of using file systems to store data (cont.)
  - Atomicity problems of updates
    - Failures may leave data in an inconsistent state with partial updates carried out
      - Example: Transfer of funds from one account to another should either complete or not happen at all; otherwise inconsistency occurs
  - Concurrent access anomalies
    - Concurrent access by multiple users needed for performance
    - Uncontrolled concurrent accesses can lead to inconsistencies
  - Security problems
    - Hard to provide user access to some, but not all, data

Database systems offer solutions to all the above problems!

WHAT is a Database?

It may look like …

It consists of:

- Tables

- Attributes (Fields)

- Tuples (Records)

- Constraints (Logical relationship among data, which is not shown here)

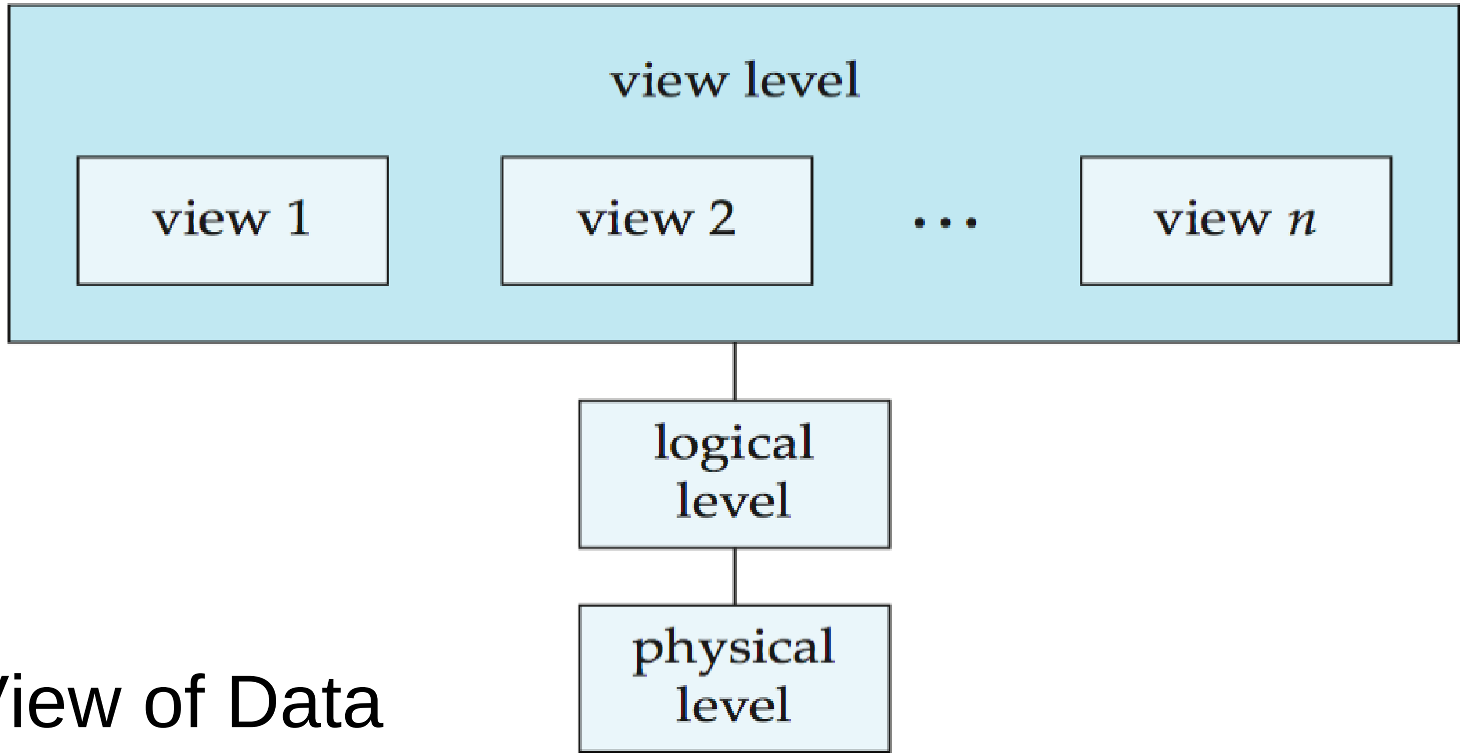
ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music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

WHAT is a Database?



- View of Data
  - Physical level: how a record is stored
  - Logical level: relationships among data, incl. data structure, data type
  - View level: data visited by application programs; hide data for security purposes

WHAT is a Database?

- Data Abstraction
  - Physical: How a record is stored
  - Logical: data, and the relationships among data

type *instructor* = **record**

ID : string;
name : string;
dept_name : string;
salary : integer;

end;

Database designers

- Similar to data structure and types

  - View: Provide application access; Information hiding

Application developers

WHAT is a Database?

- Schema (模式) - **the logical structure of the database**
  - **Analogous to type information of a variable in a program (data structure)**
  - **Physical schema: database design at the physical level**
  - **Logical schema: database design at the logical level**
- Instance (实例) - **the actual content of the database at a particular point in time**
  - **Analogous to the value of a variable**

WHAT is a Database?

- Physical Data Independence
  - The ability to **modify** the physical schema **without changing** the logical schema
  - Applications depend on the logical schema
  - Interfaces between levels
    - In general, the interfaces between the various levels and components should be well defined so that changes in some parts do not seriously influence others.

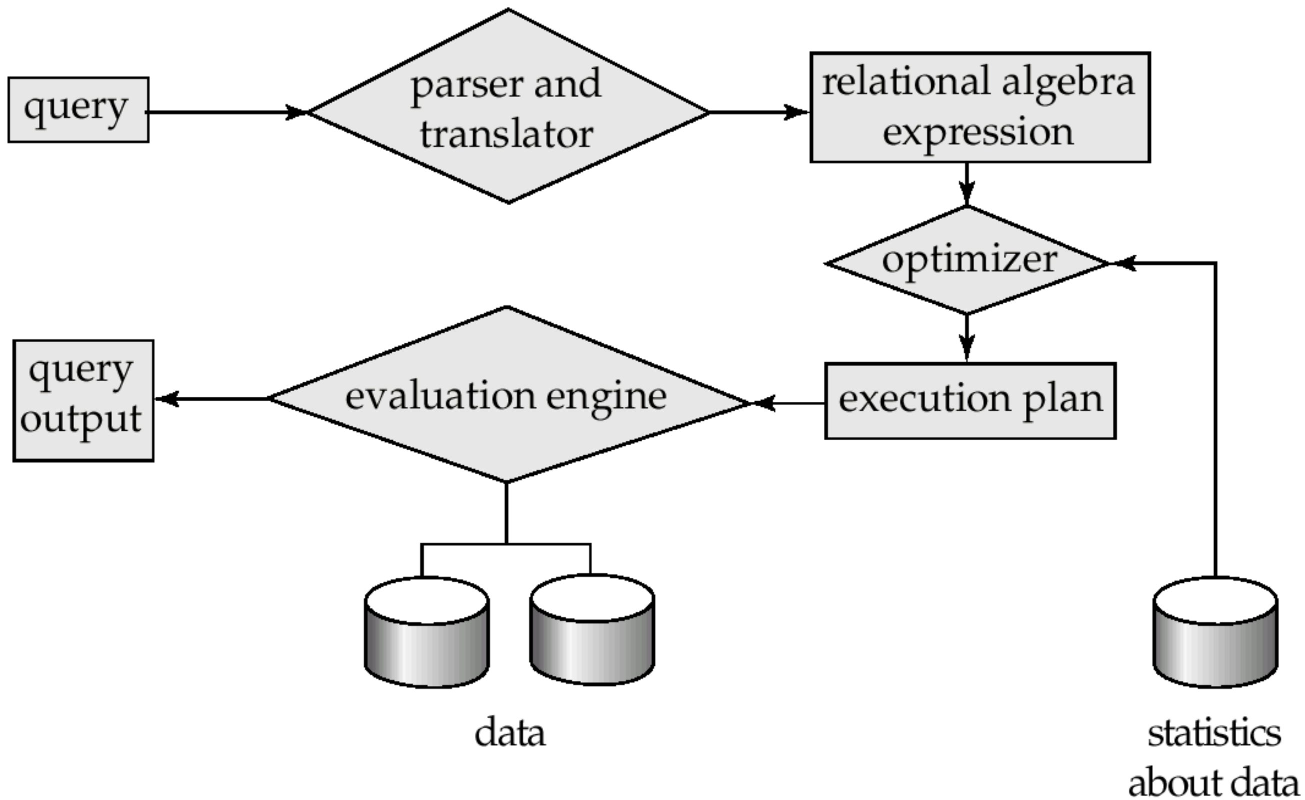
WHAT is a Database?

- Storage Manager
  - Provides **the interface between the low-level data** stored in the database **and the application programs and queries** submitted to the system
  - Interaction with the file manager
  - Efficient storing, retrieving and updating of data
- Design Issues of Storage Manager
  - **Storage access**
  - **File organization**
  - **Indexing and hashing**

WHAT is a Database?

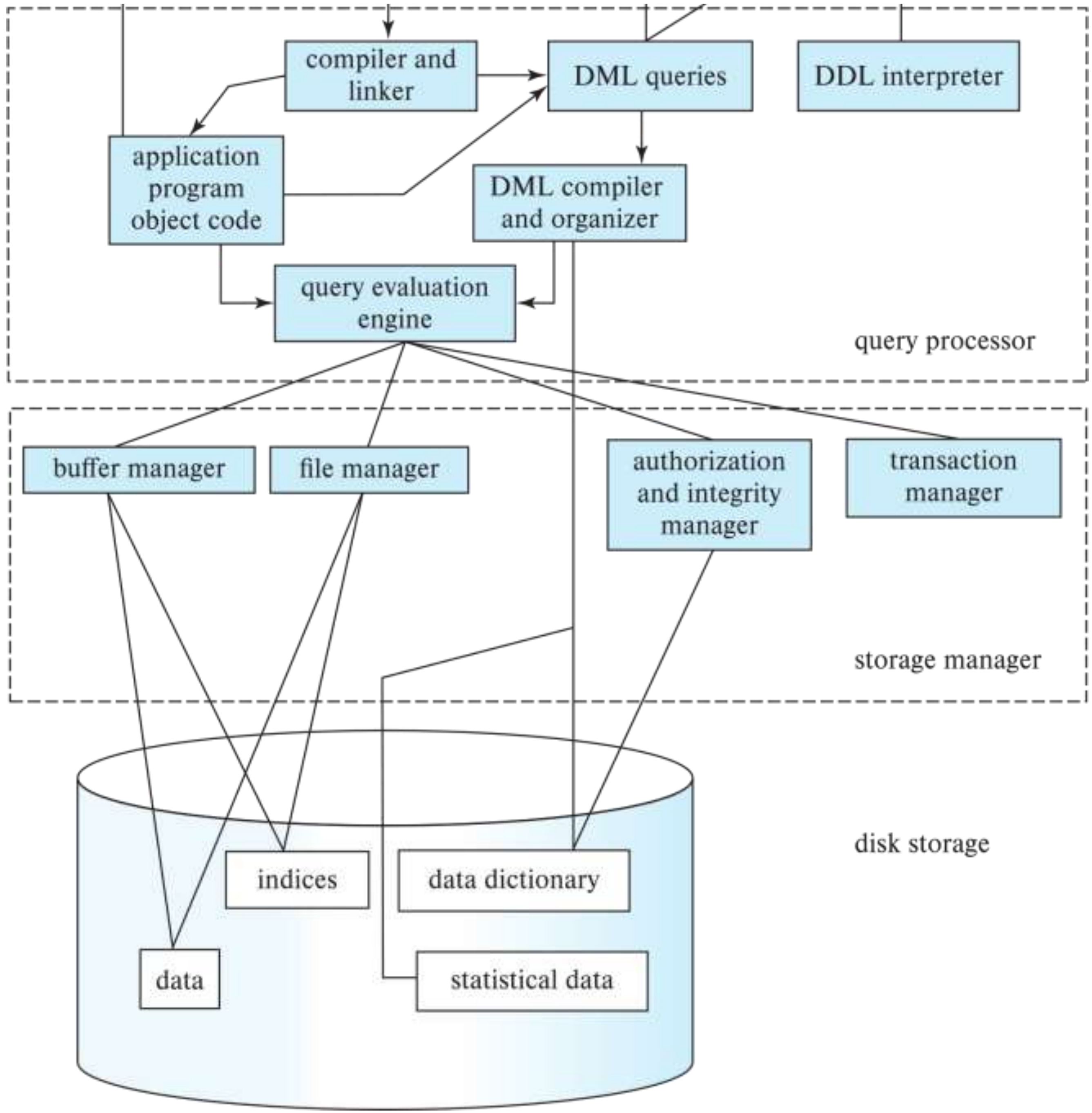
- Query Processing

1. Parsing and translation
2. Optimization
3. Evaluation



WHAT is a Database?

- Transaction Management
  - Transaction-management component
    - **Database survives system failure and transaction failure**
    - **Database remains in a consistent (correct) state despite of failures**
  - Concurrency-control management
    - **Controls the interaction among the concurrent transactions**
    - **Ensure the consistency of the database**


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
disk storage
indices
data dictionary
data
statistical data

WHO uses a Database?

- Database administrator
  - Schema definition
  - Storage structure and access method definition
  - Schema and physical organization modification
  - Granting user authority to access the database
  - Specifying integrity constraints
  - Acting as liaison with users
  - Monitoring performance and responding to changes in requirements
- Other users
  - **Naïve users**
  - **Application programmers**
  - **Sophisticated users**
  - **Specialized users**

A Database and the Applications

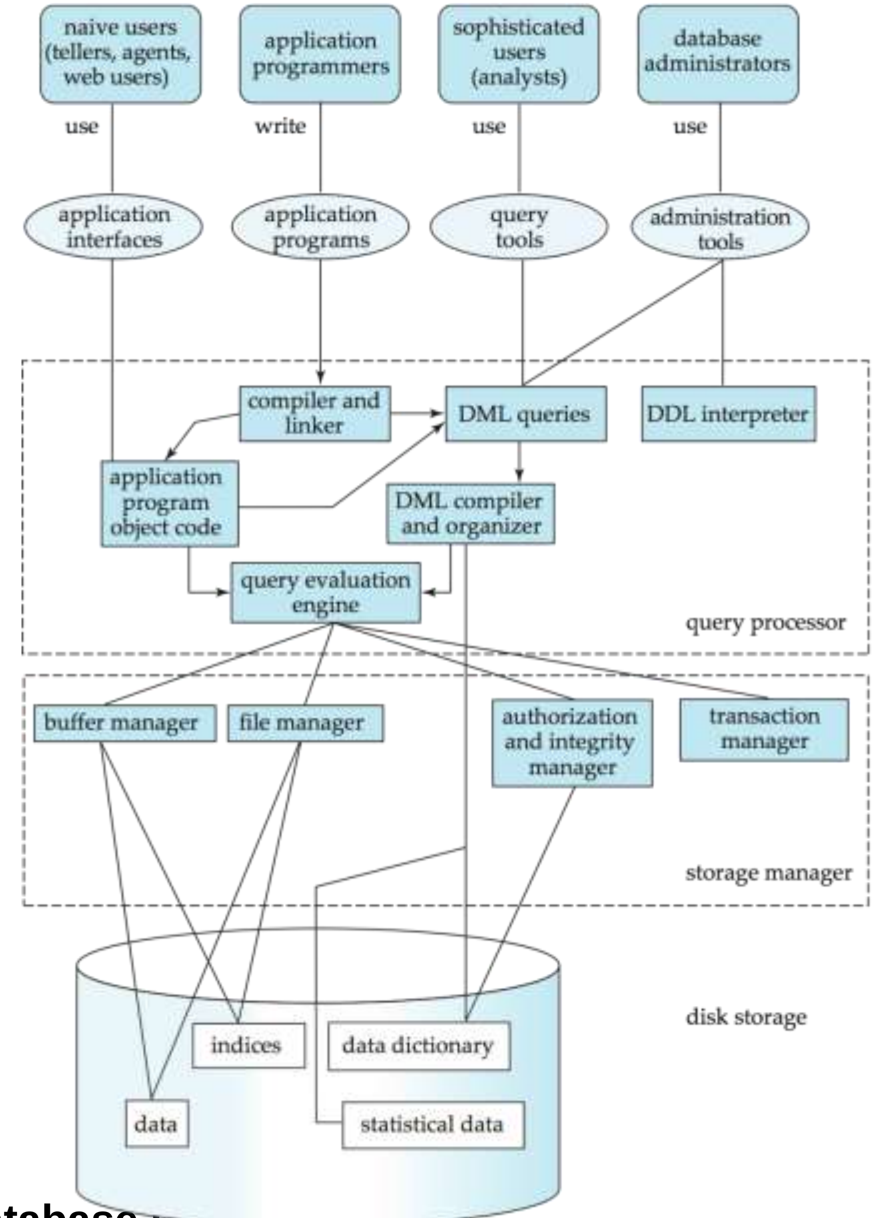
Administrators and Users

Applications, Tools

Query Processing

Storage Management

Disk Storage



HOW do we make it work?

- DDL (Data Definition Language)
  - Specification notation for defining the database schema
  - Maintains database structure, incl. table structure, setting up constraints, etc.
    - **Create table**
    - **Domain constraints**（数据、字段类型）
    - **Referential integrity**（数据之间的相关完整性）
    - **Assertions**（数据本身的完整性）
    - **Authorization**（数据授权）
  - Data dictionary
    - **DDL compiler generates a set of tables stored in a data dictionary which contains metadata**

HOW do we make it work?

- DML (Data Manipulation Language)
  - **Language for accessing and manipulating the data organized by the appropriate data model**
  - **Mainly querying and insert/delete/update data**
  - **DML also known as “query language”**
- Procedural DML
  - **user specifies what data is required and how to get those data**
- Declarative DML (nonprocedural)
  - **user specifies what data is required without specifying how to get those data**

From Theory to Practice

- Data Models
  - A collection of tools for describing: **Data**, **Data relationships**, **Data semantic** and **Data constraints**
  - **Relational Model**
    - **Relational Algebra**
    - **Mathematical basis of SQL**
  - **Entity-Relationship data model**
    - **A graphical tool for database design**
  - **Object-based data model**
    - **OO and object-relational**
  - **Semi-structured data model**
    - **XML**

From Theory to Practice

- SQL
  - **Widely used non-procedural language**
  - **Syntax, usage…**
- Application programs generally access databases through one of these:
  - **Language extensions to allow** *embedded SQL*
  - *A***pplication** *P***rogram** *I***nterface (e.g., ODBC/JDBC) which allow SQL queries to be sent to a database**

From Theory to Practice

- Database Design
  - Any problem with this design?

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

From Theory to Practice

Database Design: The process of designing the general structure of the database

- Logical Design – Deciding on the database schema. Database design requires that we find 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

From Theory to Practice

- Database Design
  - Problems
    - What is a “good” collection of relation schemas?
    - What rules may we follow when we design schemas?
    - What problems may we encounter in our design process?
  - Solutions
    - Normalization
    - Design process (ERM)

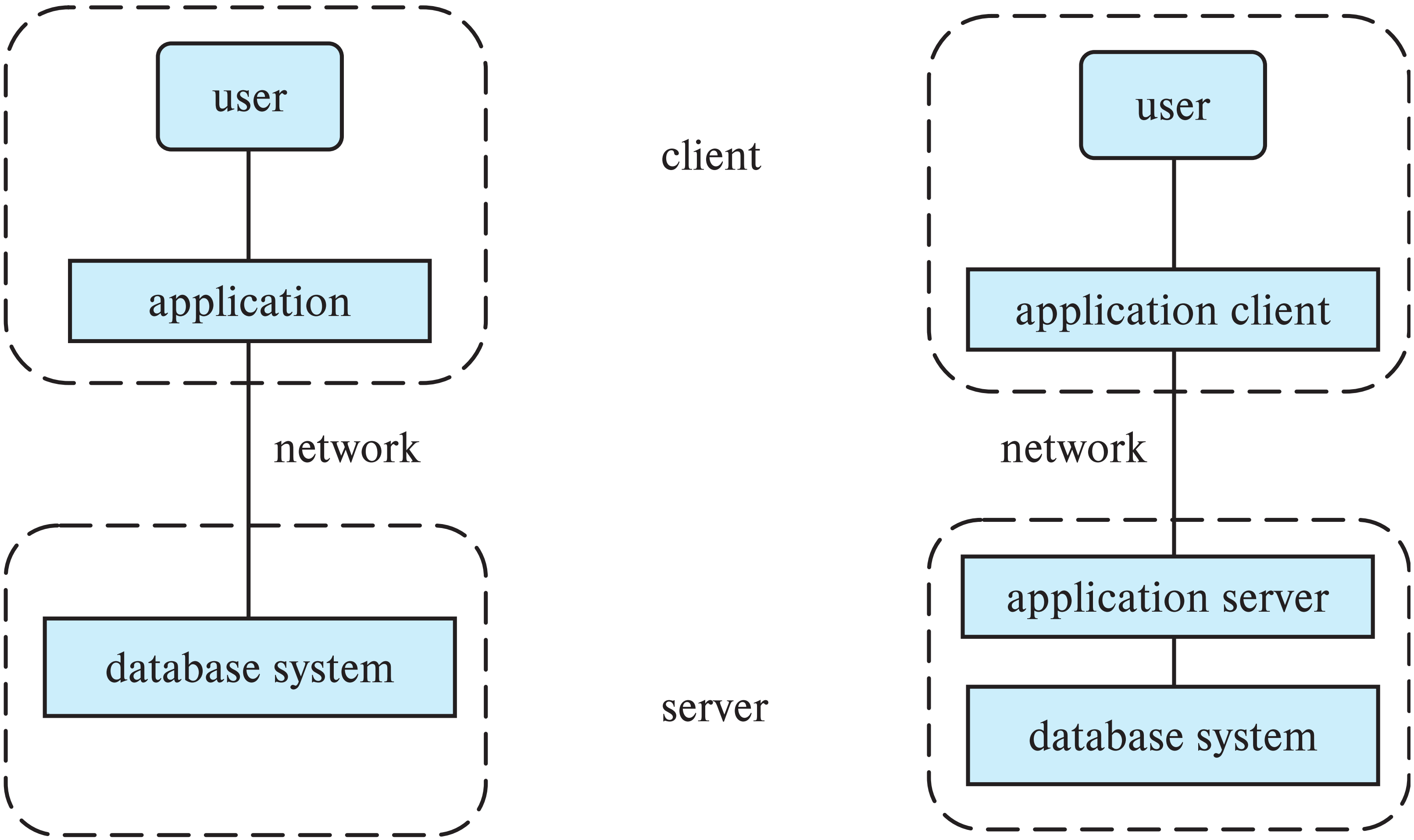
From Theory to Practice

- Database organization and relative concepts
  - Storage
  - Indexing
  - Transactions
  - Query Processing and Optimization *

Application Design

- Client/Server Examples
  - **Applications using Open Database Connection (ODBC), Borland Database Engine (BDE), etc.**
  - **Two-tier**
- Browser/Server Examples
  - **Web-based applications using Java servlets, JSP, ASP, etc.**
  - **Typically with an application server; three-tier**

Application Design



(a) Two-tier architecture (b) Three-tier architecture

History

- 1950s and early 1960s:
  - Data processing using magnetic tapes for storage
    - Tapes provide only sequential access
  - Punched cards for input
- Late 1960s and 1970s:
  - Hard disks allow direct access to data
  - Network and hierarchical data models in widespread use
  - Ted Codd defines the relational data model
    - Would win the ACM Turing Award for this work
    - IBM Research begins System R prototype
    - UC Berkeley begins Ingres prototype
  - High-performance (for the era) transaction processing

History

- 1980s:
  - Research relational prototypes evolve into commercial systems
    - SQL becomes industrial standard
  - Parallel and distributed database systems
  - Object-oriented database systems
- 1990s:
  - Large decision support and data-mining applications
  - Large multi-terabyte data warehouses
  - Emergence of Web commerce

History

- 2000s:
  - Big data storage systems
    - Google BigTable, Yahoo PNuts, Amazon…
    - “NoSQL” systems
  - Big data analysis: beyond SQL
    - Map reduce and friends
- 2010s
  - SQL reloaded
    - SQL front end to Map Reduce systems
    - Massively parallel database systems
    - Multi-core main-memory databases

Think it over

1. 请思考数据库管理系统设计与数据库应用设计的区别。
2. 如同数据结构设计不当将使得算法设计困难一样，数据库设计不当也会使得应用程序的设计变得困难。你认为在实际应用中，数据库设计可能遇到哪些问题？（不讨论数据库管理系统的设计）

Next Lecture

- Relational Model

End of

Lecture 1

Introduction