
四川大學
SICHUAN UNIVERSITY

# Database System Concepts

## Introduction

伍元凯
College of Computer Science (Software), Sichuan University
*wuyk0@scu.edu.cn*

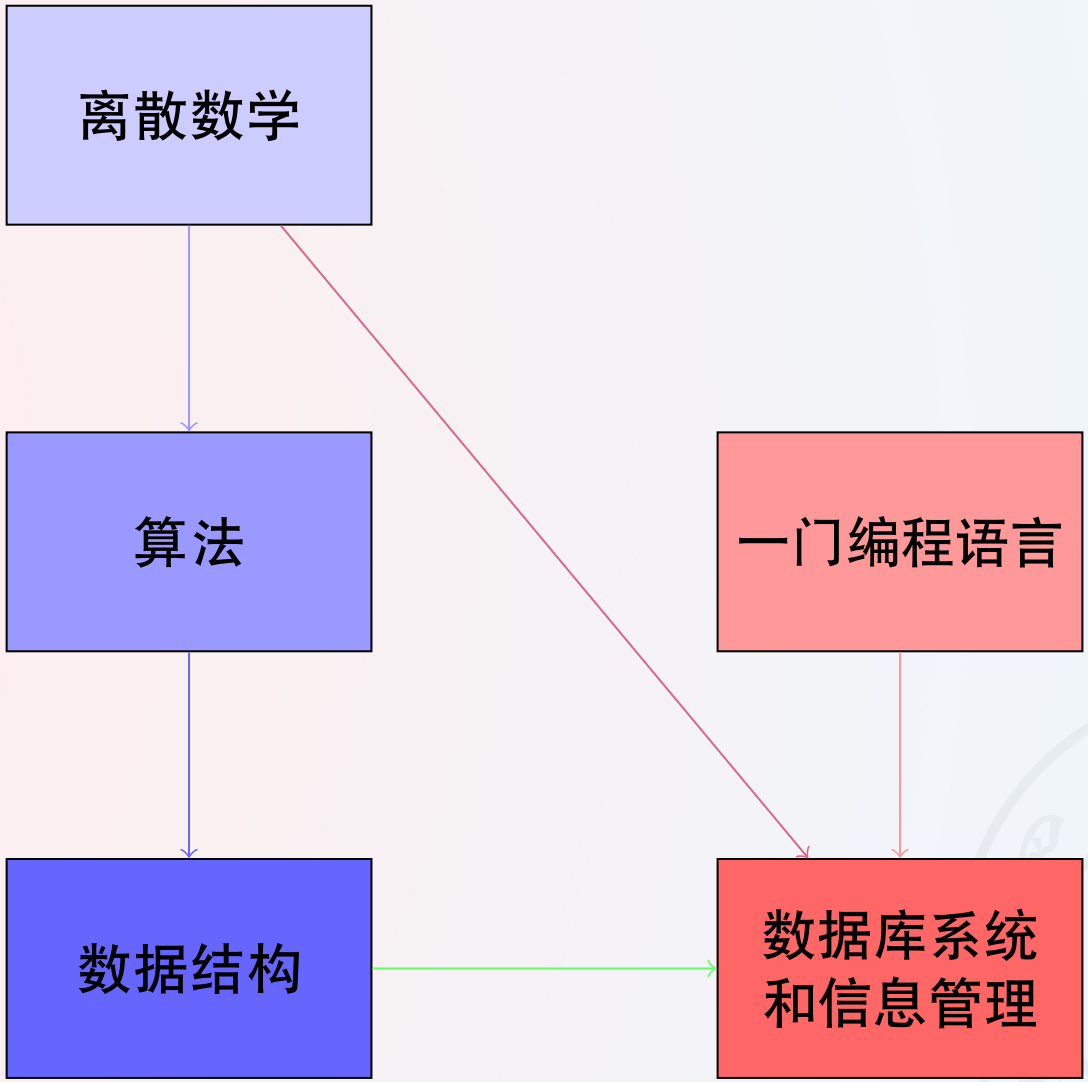
2023/03/01

- Textbook: Database System Concepts (7th Edition)
- Instructor: Wu Yuankai（伍元凯）(Tenure-Track Professor)
  Office: Yifu Science Building 223 （逸夫科学馆 223）
  Wangjiang Campus（望江校区）
  Email: wuyk0@scu.edu.cn
- Course TA: Yuan Xinyun（袁新云）
  Email: yuanxinyun@stu.scu.edu.cn

## Course Grading Policy（课程成绩评定细则）

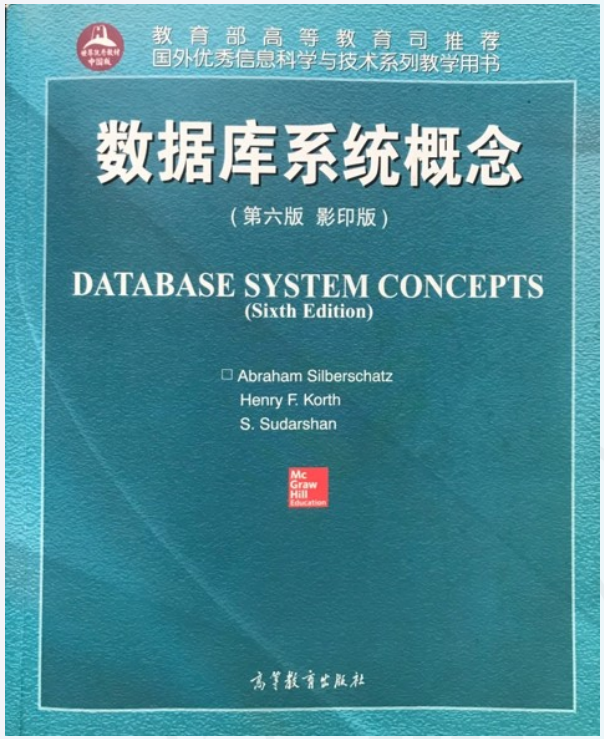
- class attendance （课堂考勤）6%;
- Lab Project （项目实践）24%;
- Final Exam （期末考试）40%;
- Quiz (期中考试 (第 7 周或者第 8 周)+ 平时表现) 30%;


**Figure 1:** Wechat Group



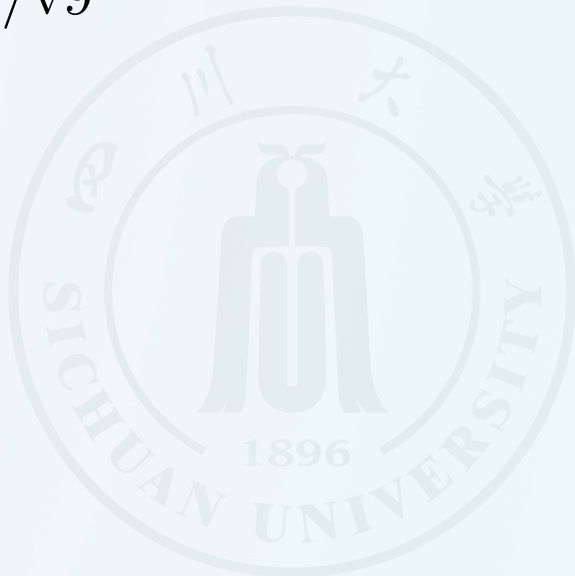

学习数据库之前需要学习的课程

- Database System Concepts (Sixth Edition)
  Abraham Silberschatz
  Henry F.Korth
  S.Sudarshan
- 高等教育出版社，2014



## References


- Thomas, Database Systems: A Practical Approach to Design Implementation and Management. Pearson Education Limited, 2014-09. ISBN: 9781292061184
- https://www.db-book.com/
- http://www.ibm.com/developerworks/cn/db2/v9
- http://www.oracle.com/database/index.html
- http://www.postgresql.org
- http://dev.mysql.com

- Teaching content arrangement

Chap 1 Introduction
Chap 2 Introduction to the Relational Model
Chap 3 Introduction to SQL
Chap 4 Intermediate SQL
Chap 5 Advanced SQL - (**6 weeks, 18 hours, Exam**)
Chap 6 Database Design and the E-R Model
Chap 7 Relational-Database Design
Chap 9 Application Development - (**6 weeks, 18 hours**)
Chap 17 Transactions (9 hours)
Chap 20 Database-System Architectures (**20 hours**)

- Database is a collection of **interrelated** data about an enterprise, which is managed by a DBMS(Database Management System).
- The primary goal of a DBMS is to provide a way to **store and retrieve** database information that is both convenient and efficient.
- Management of data involves both **defining structures** for storage of information and providing mechanisms for **the manipulation** of information.
- The database system must ensure the **safety** of the information stored, despite system crashes or attempts at unauthorized access
- If data are to be shared among several users, the system must provide **concurrency control mechanisms** to avoid possible anomalous results.

# Database Applications

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

Examples 目录

| Book ID | Title | Author | Publisher | Genre | ISBN | Nur |
|---|---|---|---|---|---|---|
| 001 | To Kill a Mockingbird | Harper Lee | HarperCollins | Fiction | 9780061120084 | |
| 002 | 1984 | George Orwell | Signet Classic | Science Fiction | 9780451524935 | |
| 003 | The Catcher in the Rye | J.D. Salinger | Little, Brown and Company | Fiction | 9780316769488 | |

Books table

| Member ID | Name | Address | Phone Number | Email |
|---|---|---|---|---|
| 001 | John Smith | 123 Main St. | 555-555-5555 | john.smith@email.com |
| 002 | Jane Doe | 456 Maple St. | 555-555-5555 | jane.doe@email.com |
| 003 | Bob Johnson | 789 Oak St. | 555-555-5555 | bob.johnson@email.com |

Library Members table

keep track of which books are checked out, who has checked them out, and when they are due back. For example, the library could create a third table that links the Book ID to the Member ID and the due date for each checked-out book.

| Book ID | Member ID |
|---|---|
| 001 | 001 |
| 002 | 002 |
| 003 | 001 |

Books borrowed table

## Vehicles Table

| Vehicle ID | Make and Model | Capacity |
|---|---|---|
| 001 | Bus A | 50 |
| 002 | Bus B | 60 |
| 003 | Train C | 100 |
| 004 | Subway D | 120 |

## Stops Table

| Stop ID | Location | Nearby Landmarks |
|---|---|---|
| 001 | Station A | City Hall |
| 002 | Station B | Library |
| 003 | Station C | Park |
| 004 | Station D | Museum |
| 005 | Station E | Shopping Center |
| 006 | Station F | Stadium |

## Routes Table

| Route ID | Start Point | End Point | Schedule |
|---|---|---|---|
| 001 | Station A | Station B | 7:00 AM |
| 002 | Station C | Station D | 8:30 AM |
| 003 | Station E | Station F | 10:00 AM |

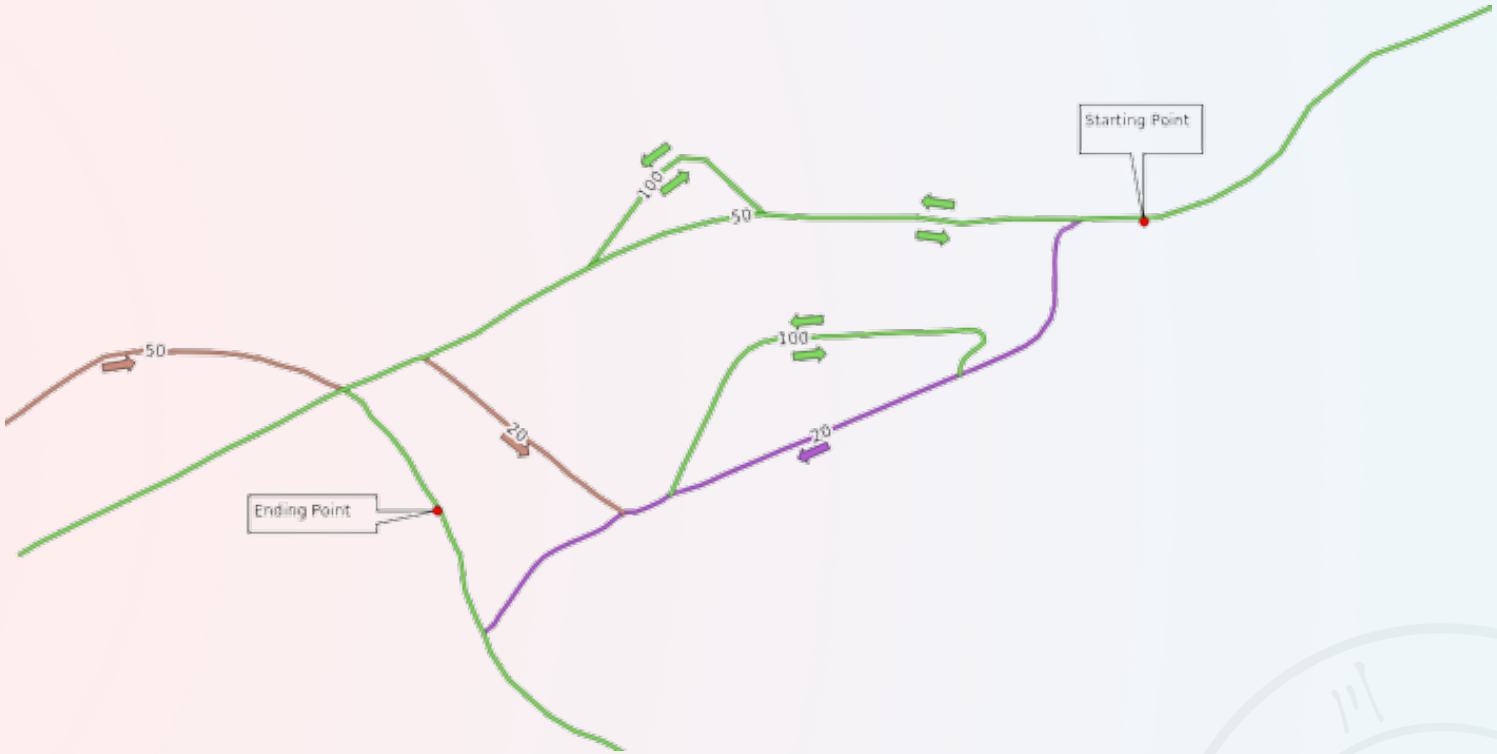
## Fares Table

| Fare ID | Passenger Type | Cost |
|---|---|---|
| 001 | Regular | $2.50 |
| 002 | Student | $1.50 |
| 003 | Senior | $1.00 |

| Vehicle ID | Route ID | Date | Passengers |
|---|---|---|---|
| 001 | 001 | 2023-02-25 | 40 |
| 002 | 002 | 2023-02-25 | 50 |
| 003 | 003 | 2023-02-25 | 80 |

Usage Table

- Public transportation planning: plan public transportation schedules and routes for a city or region
- Fare collection and pricing: Fare collection and pricing
- Transportation usage tracking: track the number of passengers using each vehicle on each route on a daily, weekly, or monthly basis.
- Transit system performance analysis
- Emergency response planning: By knowing the capacity of each vehicle and the location of each stop




The routes table could be used to create a network dataset in a GIS system, which could be used to perform network analysis tasks such as finding the shortest route between two locations or identifying the optimal location for a new bus stop.

In the early days, database applications were built directly on top of file systems，which leads to:

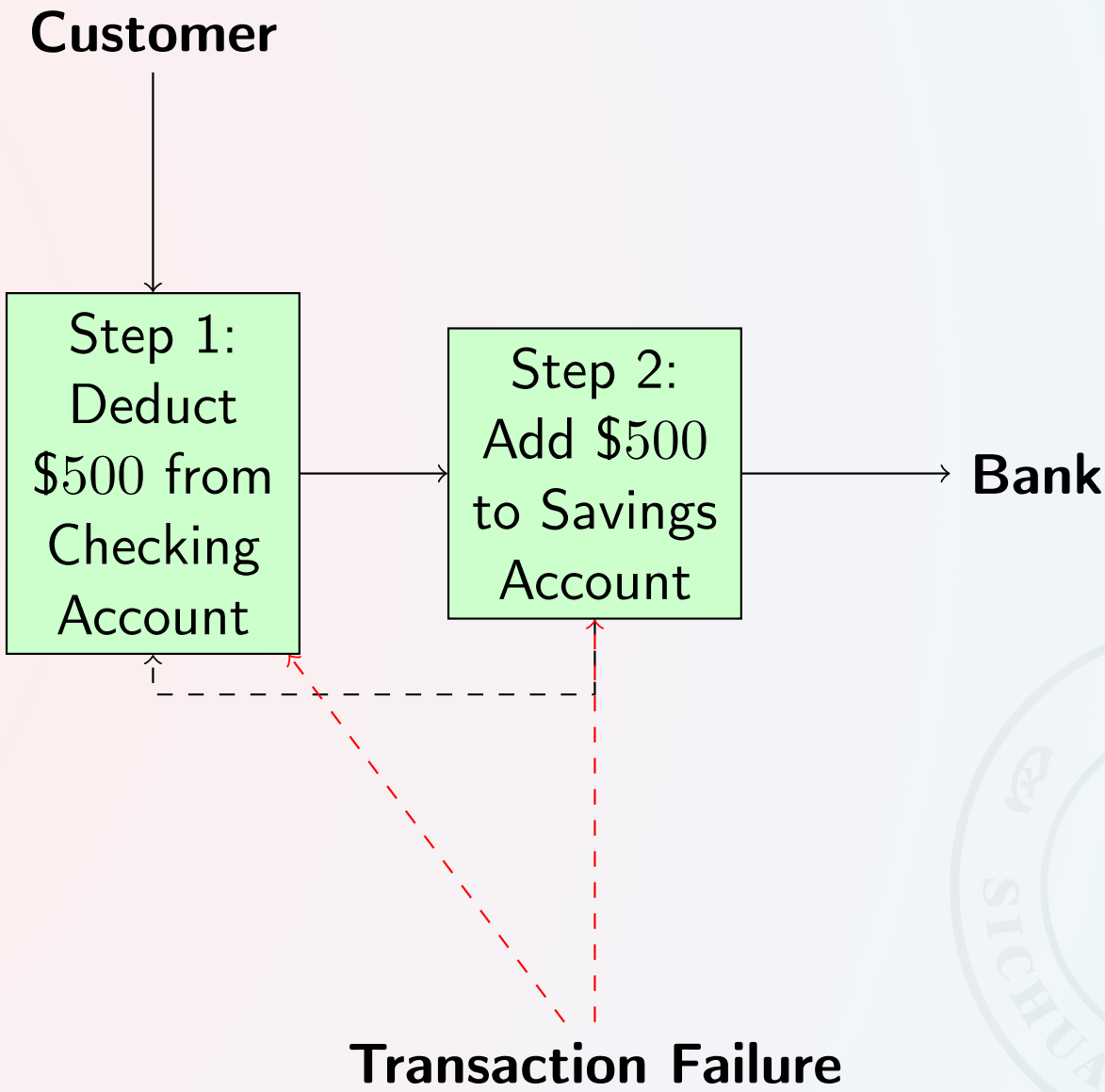
- Data redundancy（数据冗余）and inconsistency（不一致） Multiple file formats, duplication of information in different files
- Data isolation（数据孤立，数据孤岛） —multiple files and formats
- Difficulty in accessing data （存取数据困难） Need to write a new program to carry out each new task

Integrity problems（完整性问题）:

- Hard to add new constraints or change existing ones

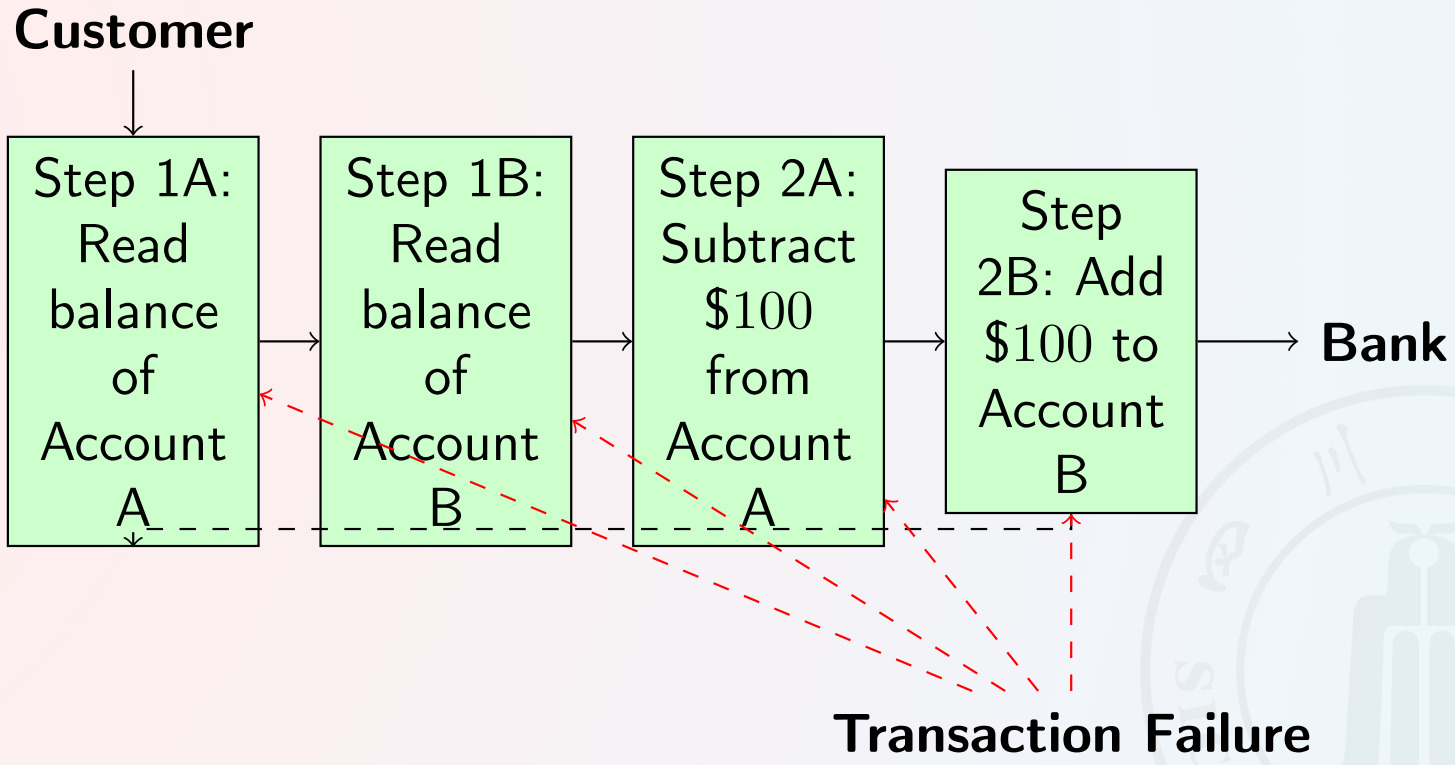
Atomicity problems（原子性问题）

- Example: Transfer of funds from one account to another should either complete or not happen at all


Customer
Step 1: Deduct $500 from Checking Account
Step 2: Add $500 to Savings Account
Bank
Transaction Failure

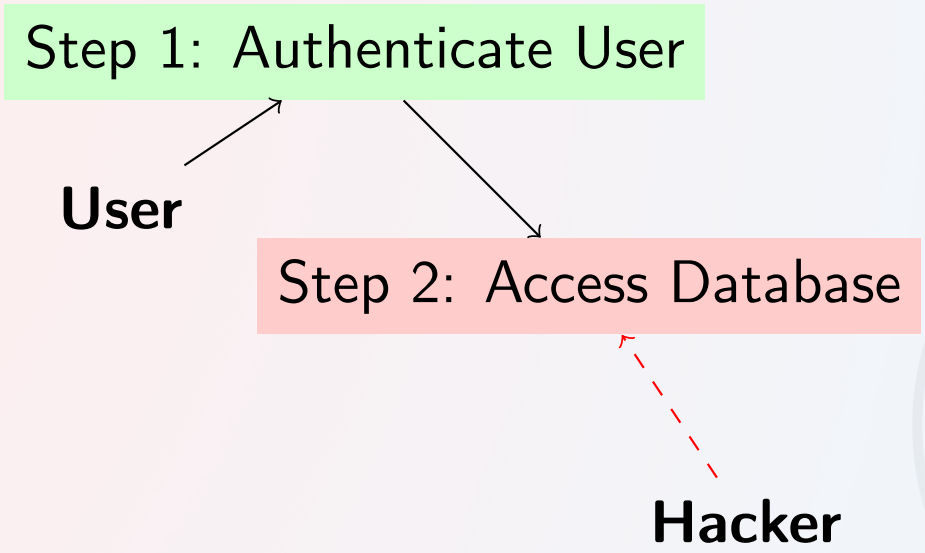
Concurrent access anomalies（并发访问异常）

- Uncontrolled concurrent accesses can lead to inconsistencies

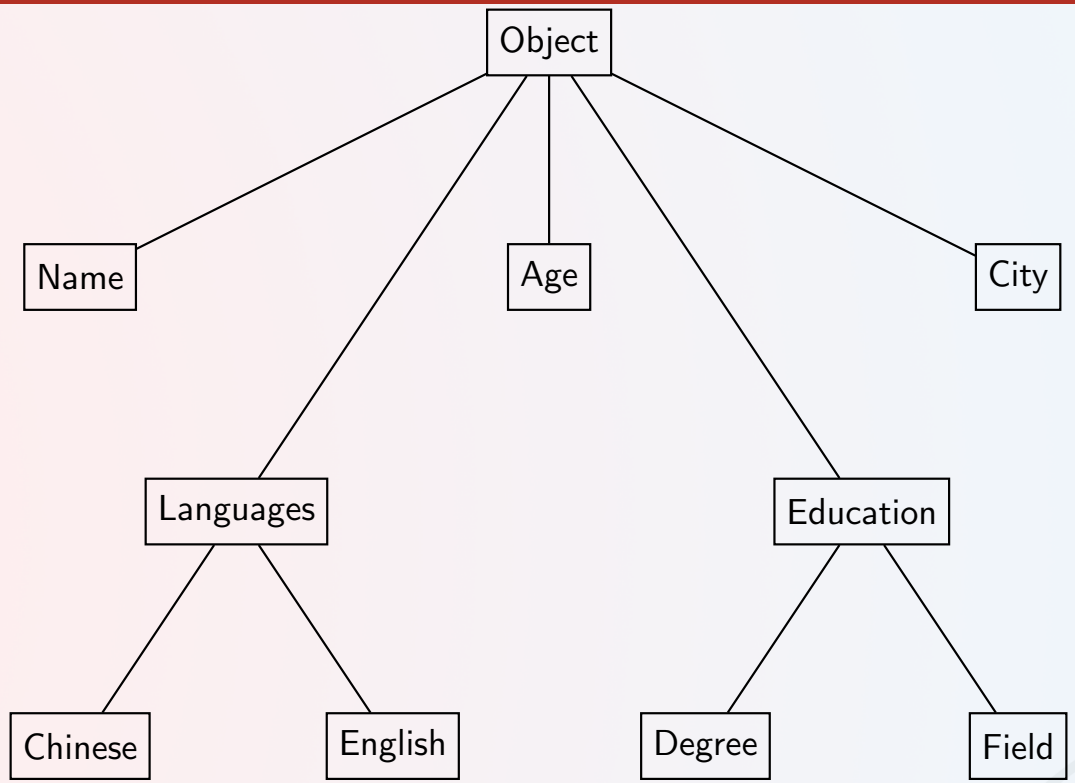


Security problems（安全性问题）

- Hard to provide user access to some, but not all, data
- Authentication(认证)
- Priviledge （权限）
- Audit（审计）



- Relational Model(关系模型): It organizes data into tables or relations, with each table consisting of rows and columns. Relationships between tables are established through keys.
- Object-Oriented Model: This model stores data as objects, with properties and methods. It is commonly used in object-oriented programming languages.
- Semi-structured Model: Data is stored in a format that doesn't conform to the strict rules of a traditional relational database. Instead, it allows for varying degrees of structure and flexibility, while still maintaining some organization and coherence. (XML,Json)




## JSON definition.

```
data = {
    "name": "刘备",
    "age": 30,
    "city": "成都",
    "languages": ["Chinese", "English"],
    "education": ["Master", "CS"]
}
```

Images:

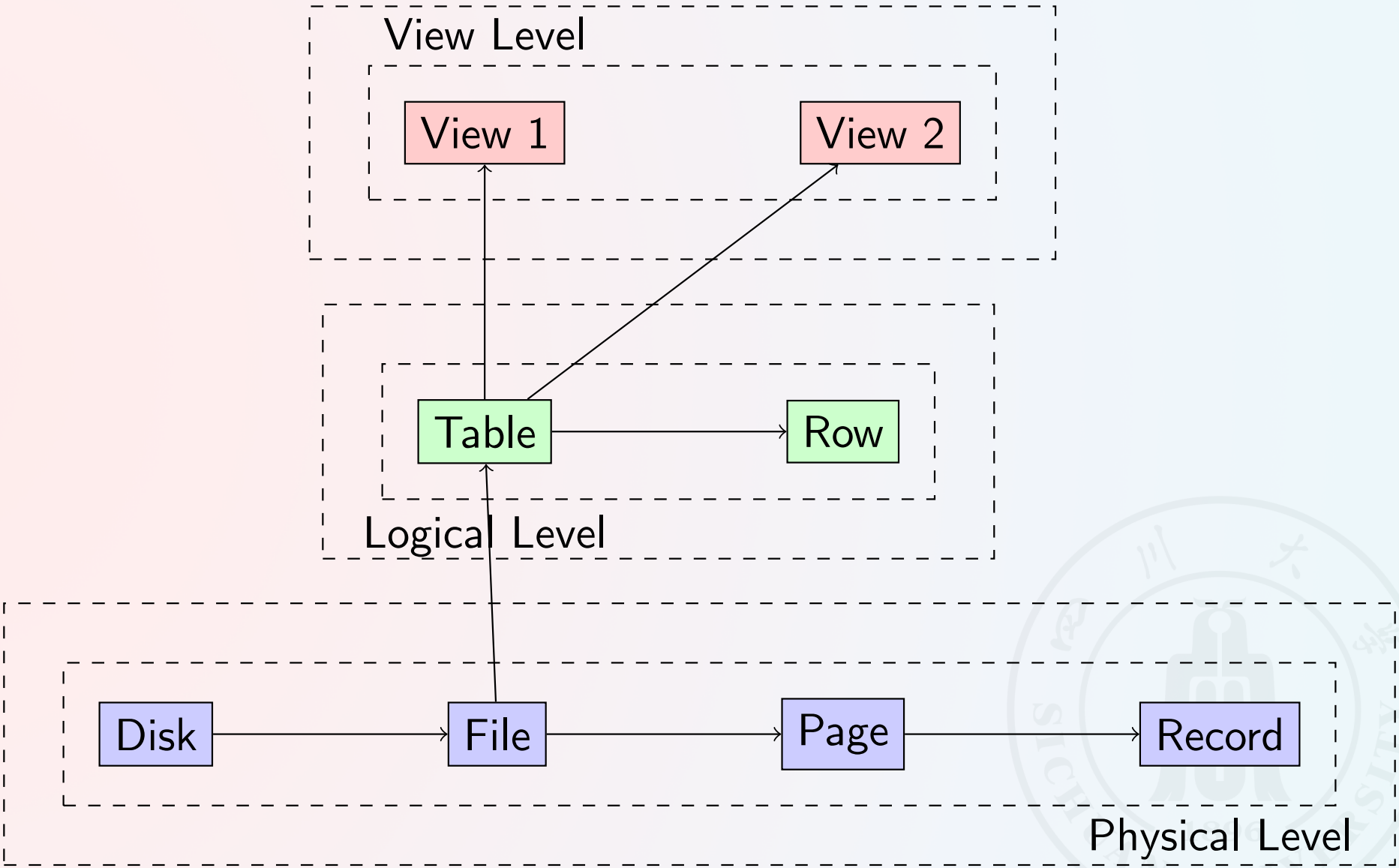
- File system storage: In this approach, the images are stored as files on the file system of the server hosting the database. The database stores only the path or location of the image file in the file system.
- Binary Large Object (BLOB) storage: BLOB is a data type that can be used to store large binary data, such as images, in a database.

Relational Model

| Attribute 属性 1 | Attribute 属性 2 | Attribute 属性 3 |
|---|---|---|
| Tuple 元组 1 Value 值 1 | 元组 1 值 2 | 元组 1 值 3 |
| 元组 2 值 1 | 元组 2 值 2 | 元组 2 值 3 |

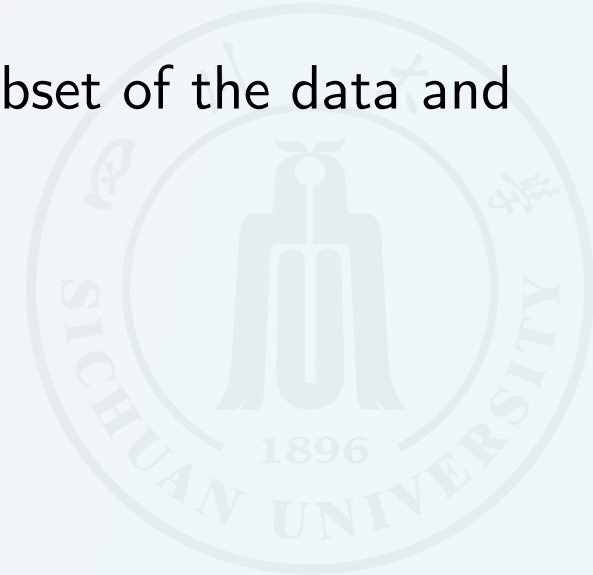
Detailed in Chap 2 and 6

# 目录


View Level
View 1
View 2
Table
Row
Logical Level
Disk
File
Page
Record
Physical Level

Data Abstraction

- Physical Level (the lowest level): At this level, the data is stored in files and records with specific formats and layouts.
- Logical Level: how the data is organized and presented to the user. It includes the definition of tables, views, indexes, and other database objects.
- View Level: Views are created by selecting a subset of the data and defining its presentation format.



目录

Schema （模式） the logical structure of the database

- Physical schema（物理模式）: database design at the physical level
- Logical schema（逻辑模式）: database design at the logical level

Instance（实例） the actual content of the database at a particular point in time

# 目录

Specification notation for defining the database schema

**TABLE definition.**

```sql
CREATE TABLE employees (
    employee_id INT PRIMARY KEY,
    name VARCHAR(50),
    salary DECIMAL(10,2)
);
```

The `PRIMARY KEY` constraint is used to indicate that the `employee_id` column is the primary key for the table, which means that it uniquely identifies each row in the table.

## 目录

## Example 1: Adding salary.

```sql
UPDATE employees
SET salary = salary * 1.05
WHERE department = 'Sales';
```

## Example 2: Finding low salary.

```sql
SELECT name, salary
FROM employees
WHERE department = 'Marketing'
ORDER BY salary DESC;
```

Detailed in Chap 3, 4 and 5

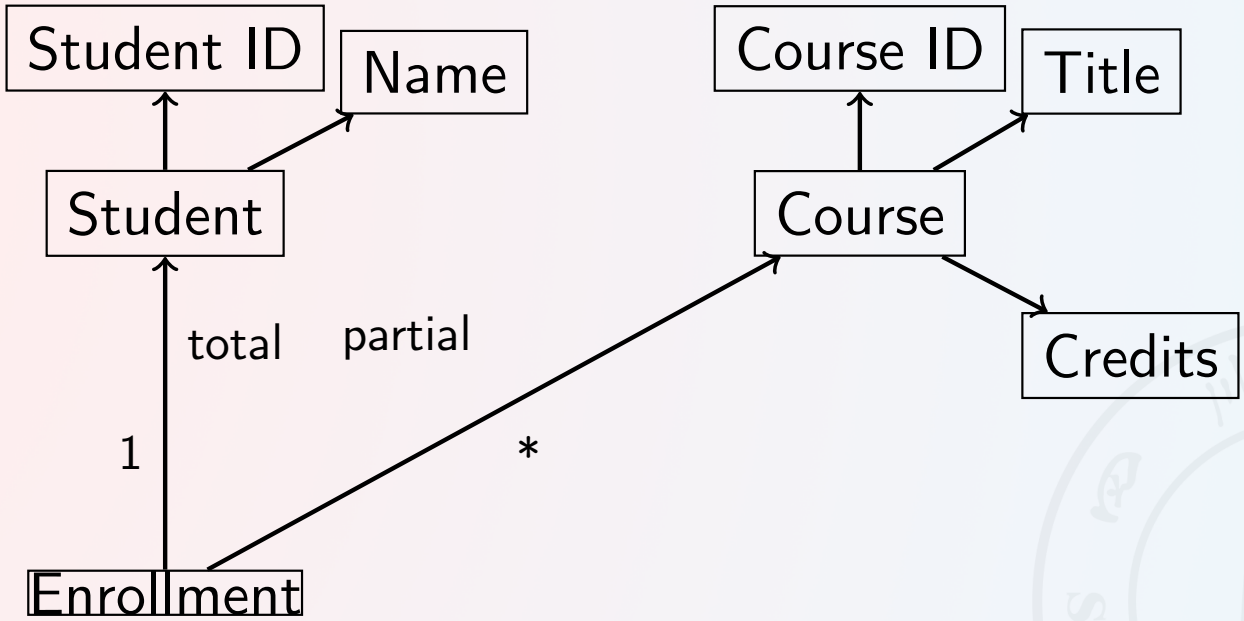
# 目录

## Access via API.

```python
import sqlite3
# Connect to the database
conn = sqlite3.connect('example.db')
# Create a cursor object to execute SQL commands
cur = conn.cursor()
# Execute a SELECT statement
cur.execute('SELECT * FROM employees')
# Fetch the results of the query
results = cur.fetchall()
# Print the results
for row in results:
    print(row)
# Close the cursor and database connection
cur.close()
conn.close()
```

## Entity Relationship Model （实体-联系模型）

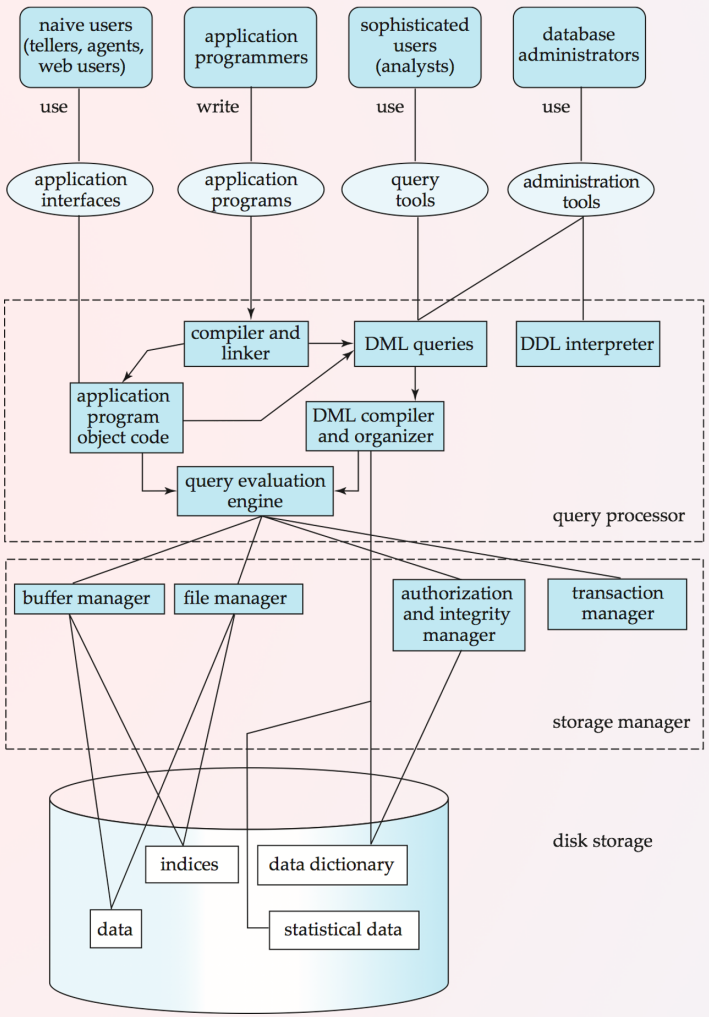
(Chap 7)



## Is this design good?

```sql
CREATE TABLE Sales (
    Sales_ID INT PRIMARY KEY,
    Product_ID INT,
    Product_Name VARCHAR(50),
    Category VARCHAR(20),
    Supplier_Name VARCHAR(50),
    Supplier_Country VARCHAR(50),
    Supplier_City VARCHAR(50),
    Sale_Date DATE,
    Customer_Name VARCHAR(50),
    Customer_Country VARCHAR(50),
    Customer_City VARCHAR(50),
    Quantity INT,
    Price DECIMAL(10,2)
);
```

Detailed in Chap 8




The functional components of a database system can be divided into

- The storage manager
- The query processor component
- The transaction management component.

∗1

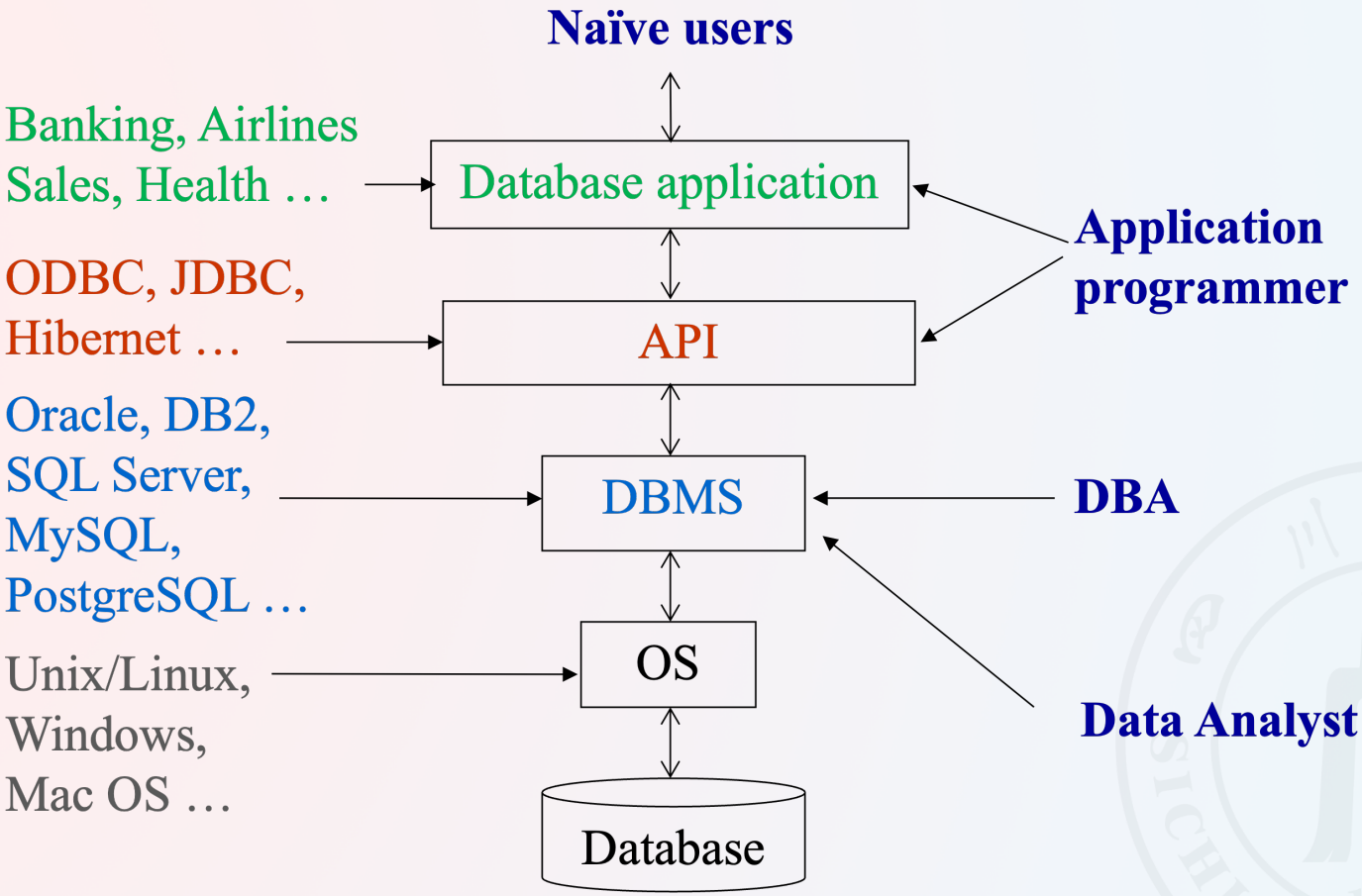
[1] https://www.bilibili.com/video/BV1ve411F794/?spm_id_from=333.999.0.0, some interesting videos.

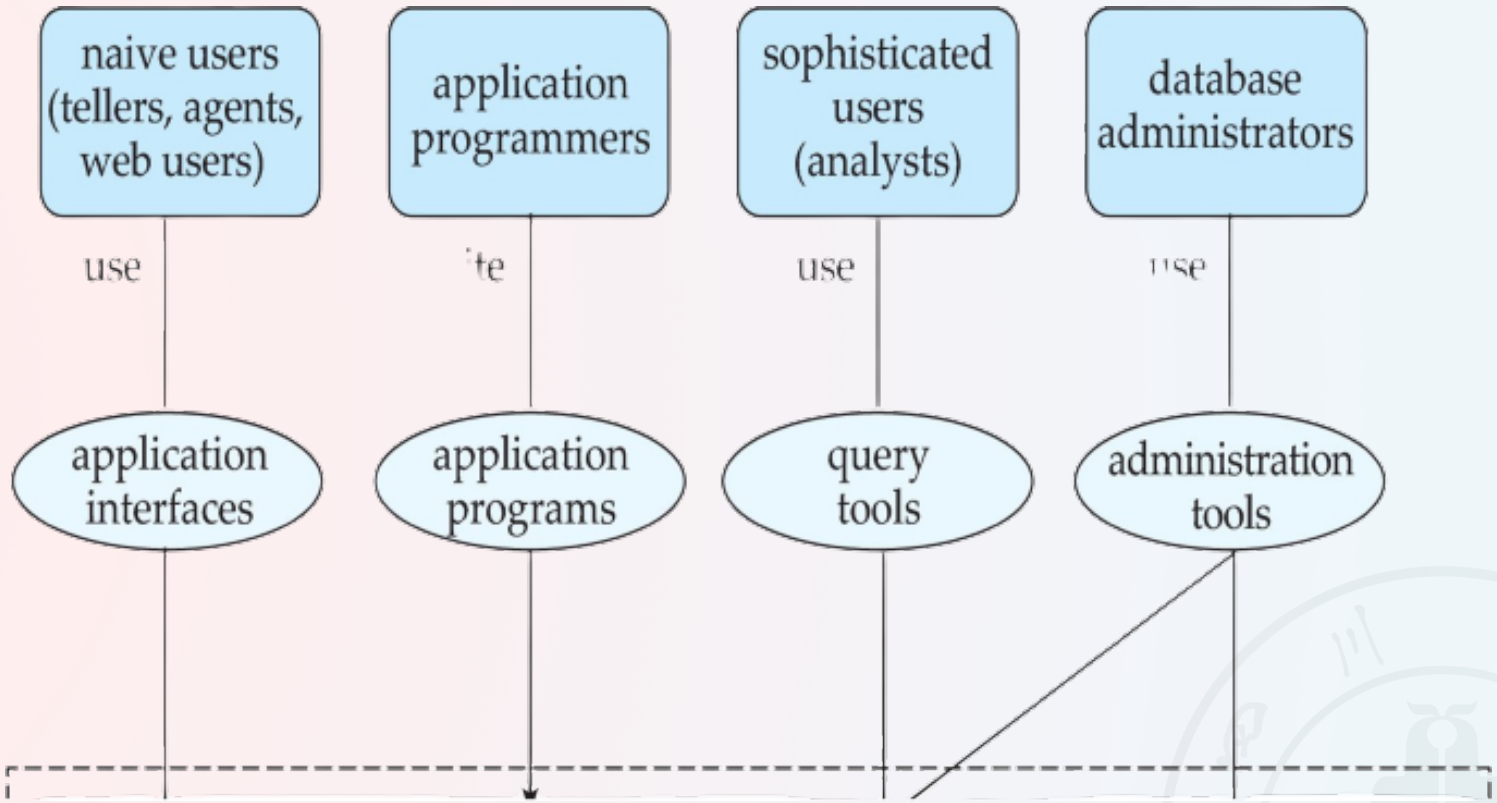
# 目录

- A transaction（事务） is a collection of operations that performs a single logical function in a database application.
- Recover Manager ensures that the database remains in a consistent (correct) state despite system failures (e.g., power failures and operating system crashes) and transaction failures.
- Concurrency-control manager controls the interaction among the concurrent transactions, to ensure the consistency of the database.

Chap 17

# Database users




naive users (tellers, agents, web users)
application programmers
sophisticated users (analysts)
database administrators
use
use
application interfaces
application programs
query tools
administration tools
Database

Database users

- Application programmers  interact with system through DML calls
- Naive users  people accessing database over the web, bank tellers, clerical staff
- Database Administrator  Coordinates all the activities of the database system.

Database administrator's duties include:

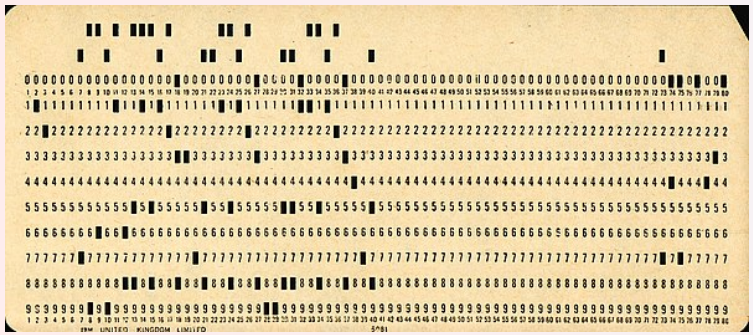
- Schema definition
- Storage structure and access method definition
- Schema and physical organization modification
- Granting user authority to access the database
- Specifying integrity constraints
- Acting as liaison with users
- Monitoring performance and responding to changes in requirements

Tokens from Jarmo, Iraq, 6500 BC. Courtesy Denise Schmandt–Besserat. The token shapes represented various kinds of merchandise prevalent in the farming economy of the time[2].

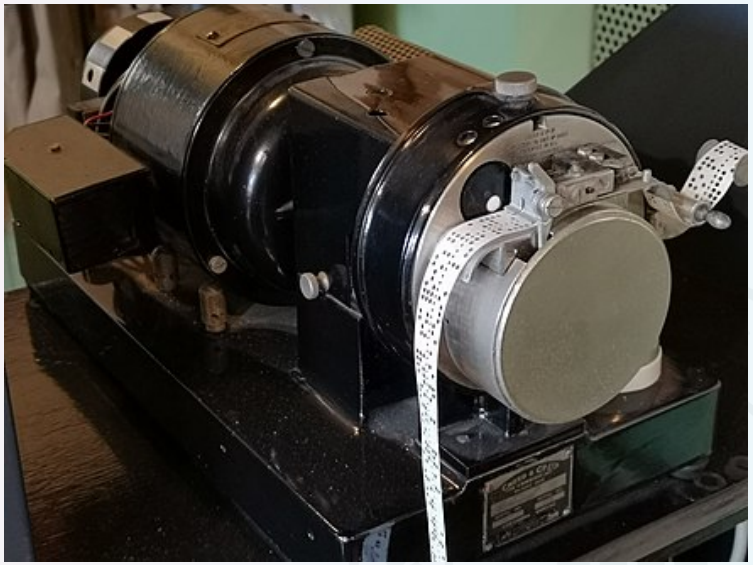
[2] https://sites.utexas.edu/dsb/tokens/from-accounting-to-writing/

1950s and early 1960s:

- Data processing using magnetic tapes for storage
- Punched cards for input
- Tapes provide only sequential access



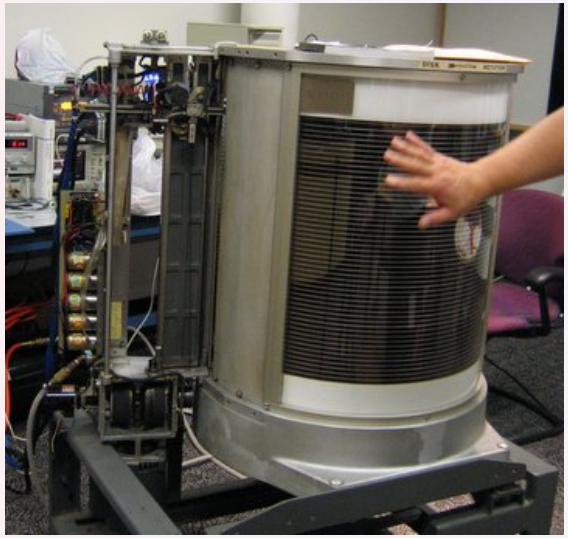
**(a)** A 12-row/80-column IBM punched card from the mid-twentieth century.



**(b)** Creed model 6S/2 5-hole paper tape reader

1960s :

- Hard disks allow direct access to data



Partially disassembled IBM 350 (RAMAC), 69 years old.

Integrated Data Store (The first DBMS in a batch-oriented, serial-file-processing, load and execute one-program-at-a-time world[3].).

- A direct access database was implemented on a virtual memory basis
- The network data model
- **A data description language**
- **A data storage and retrieval language**
- An exclusive ‘‘working storage’’ area for each record type

Charles Bachman at the 2012 ACM Turing Centenary Celebration (December 11, 1924 –July 13, 2017).

[3]Bachman, Charles W. ”The origin of the integrated data store (IDS): The first direct-access DBMS.” IEEE Annals of the History of Computing 31.4 (2009): 42-54.

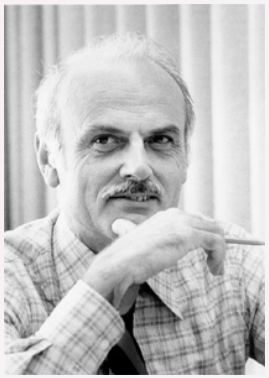
1970s :

- Business Applications OLTP (Online Transaction Processing)
- **Edgar F. Codd** defines the relational data model
- IBM Research begins System R prototype(1974, Jim Gray as a key player)
- UC Berkeley begins Ingres prototype(1974, leaded by Michael Stonebraker)

## Turing Award: Edgar F. Codd (1923-2003)

**A Relational Model of Data for Large Shared Data Banks**, CACM 1970.

The central idea of a relational model is to describe a database as a collection of predicates over a finite set of predicate variables, describing constraints on the possible values and combinations of values.



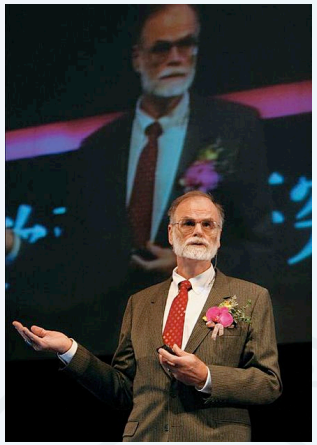
1980s:

- RDBMS implementation
- Research relational prototypes evolve into commercial systems
- Parallel database systems
- Distributed database systems
- Object-oriented database systems
- Object-relational Database systems
- Extended to Engineering Applications

## Turing Award: Jim Gray (1944-2007(Lost in the sea))

IMS、System R、SQL/DS、DB2

**《Transaction Processing: Concepts and Techniques》**



1998 ACM Turing Award for his seminal contribution to database and transaction processing research and technical leadership in system implementation.

Disappearance on January 28, 2007 at sea in his sloop Tenacious, during a short solo sailing trip to the Farallon Islands near San Francisco to scatter his mother's ashes.

1990s:

- Business intelligence(BI)
- Large decision support and data-mining applications
- Large multi-terabyte data warehouses
- OLAP(Online Analytical Processing)
- Emergence of Web commerce
- The Web changes everything
- New workloads –performance, **concurrency**, availability

2000s:

- Web Era
- Big data
- XML and XQuery standards
- Automated database administration
- NoSQL(MongoDB, Cassandra, HBase)

2010s:

- NewSQL
- Cloud database
- Blockchain
- Autonomous Database (AI powered Database)

List at least two reasons why DBMS use a declarative language such as SQL instead of just providing a library of functions in languages such as C, Python, or Java for data manipulation. Write some code using these languages for data manipulation and discuss their weaknesses.

列出两个原因，为什么 DBMS 使用类似 SQL 这样的声明式语言，而不仅仅是提供一组 C、Python 或 Java 函数库来进行数据操作。使用这些语言编写一些数据操作代码，并讨论它们的缺点。

## Reading

Bachman, Charles W. "The programmer as navigator." Communications of the ACM 16.11 (1973): 653-658.

# Thanks
# End of Chapter 1