# CS33503: 数据库系统

邹兆年

哈尔滨工业大学
计算机科学与技术学院
海量数据计算研究中心
电子邮件: znzou@hit.edu.cn

2019年春



## 为什么要学习数据库系统?

- 当你得到一个数据库，并需要对它进行管理时，你需要了解数据库系统的基本概念，掌握数据库语言，具备数据库系统的使用技能。
- 当你面对一个数据密集型应用设计与开发需求时，你需要掌握数据库的设计方法，了解如何评估设计方案的优劣，具备数据库系统应用开发能力。
- 当你接手一个性能低下的数据密集型应用时，你需要了解数据库系统的工作原理，知道如何对系统进行优化和重新设计。
- 当你参与一种新型数据库管理系统的研发时，你需要了解多种数据库管理系统的工作原理和设计方案，并具备一定的研究能力。

# 授课安排

课堂授课

- 20节课，共40学时
- 第1–6, 8–11周，周一，第1–2节，正心34
- 第1–6, 8–11周，周三，第1–2节，正心34

实验

- 2次实验，共8学时
- 第8、12周，周六，第1–4节，格物208 (1603101、1603102、1603103班)
- 第8、12周，周日，第5–8节，格物207 (1637101、1637102班)



# 教学团队

主讲: 邹兆年 副教授、博导

- 办公室: 综合楼810
- 电子邮件: znzou@hit.edu.cn
- QQ: 12943596

助教: 张子超、袁怡轮、刘丹丹

- QQ群: 544017070 (请将成员昵称修改为"学号-姓名")

2019春数据库系统
扫一扫二维码，加入该群。

## 教材与参考书

- Ramez Elmasri, and Shamkant B. Navathe. Fundamentals of Database Systems (Fourth Edition)
- Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom. Database System Implementation (Second Edition)
- Abraham Silberschatz, Henry F. Korth, and S. Sudarshan. Database System Concepts (Sixth Edition)
- Jeffrey D. Ullman, and Jennifer Widom. A First Course in Database Systems (Third Edition)
- 王珊, 萨师煊. 数据库系统概论(第5版).
- Peter Bailis, Joseph M. Hellerstein, and Michael Stonebraker. Readings in Database Systems (Fifth Edition). `http://www.redbook.io/`



## 考核方式

- 期末考试: 占60%，考试时间待定
- 实验: 占20%
- 作业: 占20%

# 第1章: 绪论

邹兆年

哈尔滨工业大学
计算机科学与技术学院
海量数据计算研究中心
电子邮件: znzou@hit.edu.cn

2019年春



## Outline[1]

- Basic Concepts
- Functions of Database Systems
- Features of Database Systems
- Data Models
- Database Schemas
- DBMS Languages
- DBMS Architectures

[1]课件更新于2019年3月27日

## What is Data?

Data: known facts that can be *recorded* and have an implicit *meaning*

- "Chomolungma": the name of the highest mountain in the world
- 8,848: the height of the Chomolungma in metres
- 29,029: the height of the Chomolungma in feet
- "Asia": the continent where the Chomolungma locates



## Types of Data

- Structured data: relational data
- Semi-structured data: XML, JSON
- Text data: documents, news reports, papers
- Time series: stock market data
- Sequence data: DNA sequence
- Spatial data: maps, GIS
- Graph data: social networks, knowledge graphs
- Sensor data: Internet of Things (IoT), sensor networks
- Multimedia data: images, audio, video
- Web data: Web pages, Web graph, Web logs
- ...

Many applications need to manage data

## What is Data Management?

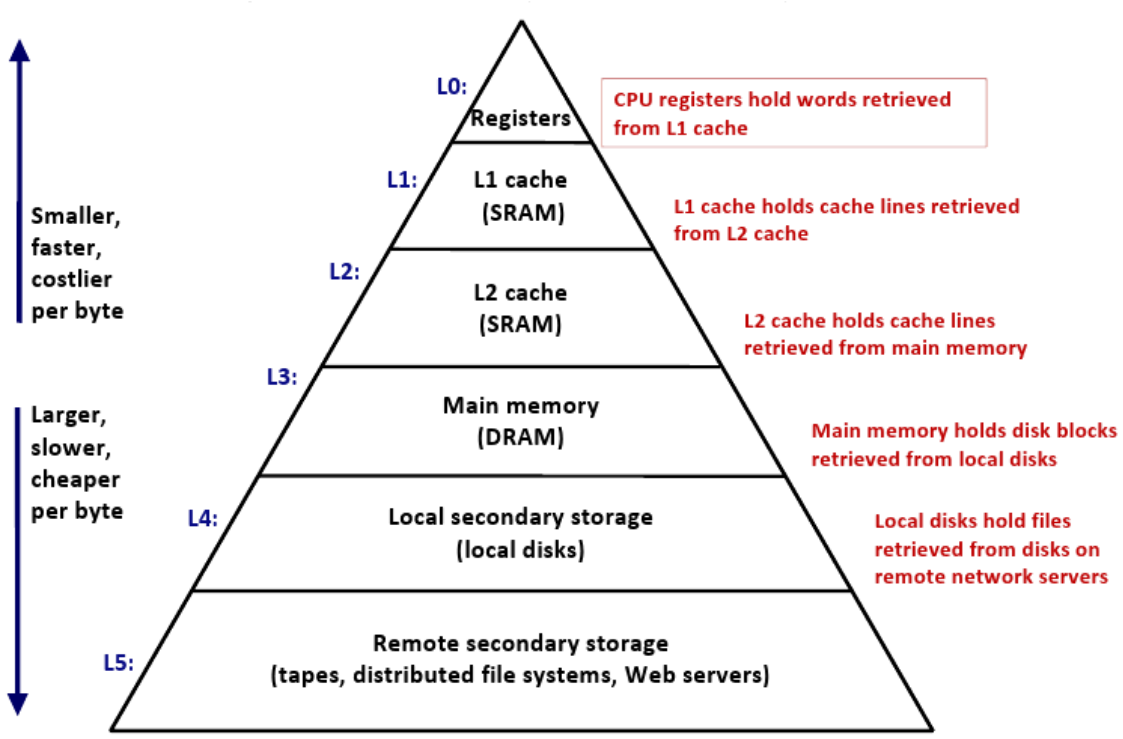
1. Defining data types, structures, and constraints
2. Constructing or loading initial data on a non-volatile storage medium
3. Manipulating data
   - Data retrieval: Querying, generating reports
   - Data modification: Insertions, deletions and updates to its content
4. Processing and sharing data by a set of concurrent users and application programs–yet, keeping all data valid and consistent
5. Security, access control, archiving, etc.



## Data Management Approaches

- File system approach
- Database system approach
- NoSQL (No SQL → Not Only SQL → Not yet SQL) approach
- Big data approach (data science)

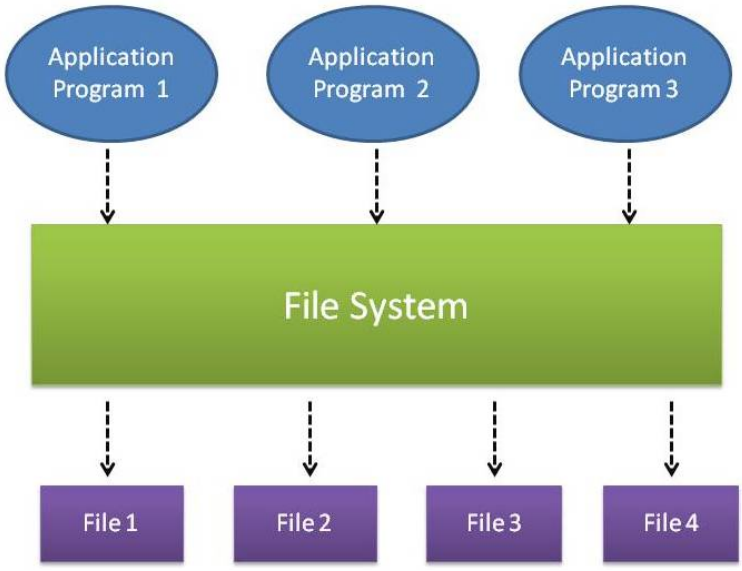
# Memory Hierarchy






# File System Approach

- Data is stored in *files*
- Data is managed by *application programs*



# File System Approach: Example

student.txt

| | | | | |
|---|---|---|---|---|
| Alice | F | CS | 001 | 90 |
| Alice | F | CS | 002 | 91 |
| Bob | M | CS | 001 | 60 |
| Jack | M | MA | 001 | 85 |
| Jill | F | MA | 002 | 80 |

course.txt

| | |
|---|---|
| 001 | 'Database Systems' |
| 002 | 'Data Mining' |

- Find all girls
- Find all students in Department of Computer Science (CS)
- Find all students who study the course # 001
- Find all students who study the course "Database Systems"

Programs have to be changed when data change



# Limitations of File System Approach

student.txt

| | | | | |
|---|---|---|---|---|
| Alice | F | CS | 001 | 90 |
| Alice | F | CS | 002 | 91 |
| Bob | M | CS | 001 | 60 |
| Jack | M | MA | 001 | 85 |
| Jill | F | MA | 002 | 80 |

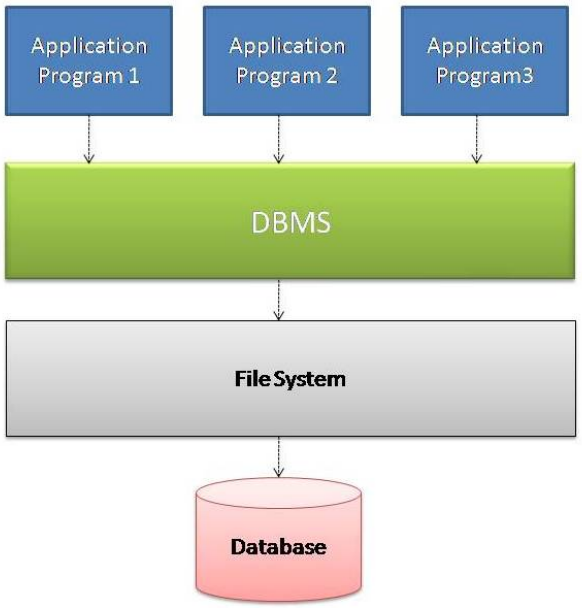
course.txt

| | |
|---|---|
| 001 | 'Database Systems' |
| 002 | 'Data Mining' |

- Data redundancy
- Data inconsistency
- Data integrity problems
- Difficulty in accessing data: no indexes
- Security problems: no access control
- Concurrency problems: no concurrency control

# Database System Approach

- Data is stored in *databases*
- Data is managed by *database management systems (DBMS)*

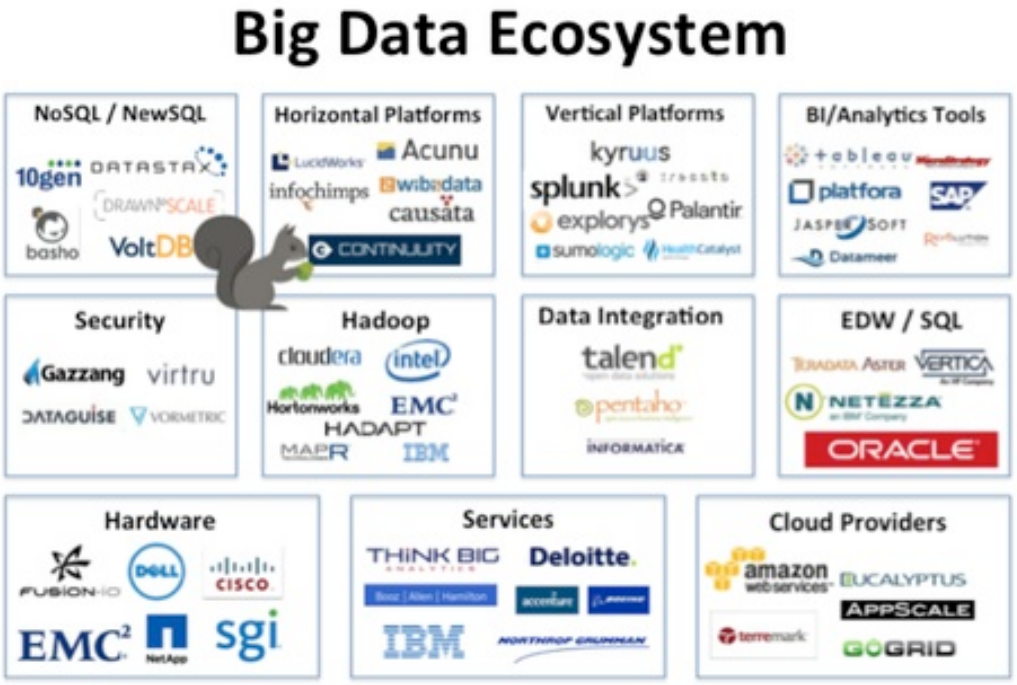





# File System Approach vs. Database System Approach

| | File system approach | DB approach |
|---|---|---|
| Data redundancy | High | Low |
| Data consistency | No | Yes |
| Accessing methods | No | Yes |
| Data integrity | No | Yes |
| Access control | No | Yes |
| Concurrency control | No | Yes |

# Big Data Approach






# 1.1 Basic Concepts

# Databases (DB)

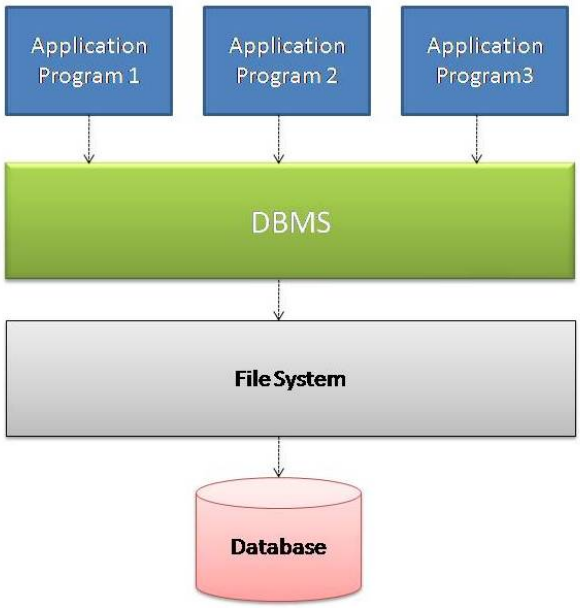
A database (DB) is a set of related data that is organized, shared, and persistent.

| | Organization | Sharing | Persistence |
|---|---|---|---|
| Database | Organized | High | Yes |
| File | Unorganized | Low | Yes |



# Database Management Systems (DBMS)

A database management system (DBMS) is a general-purpose system software that facilates the organization, storage, manipulation, control, and maintainence of databases among various users and application programs.



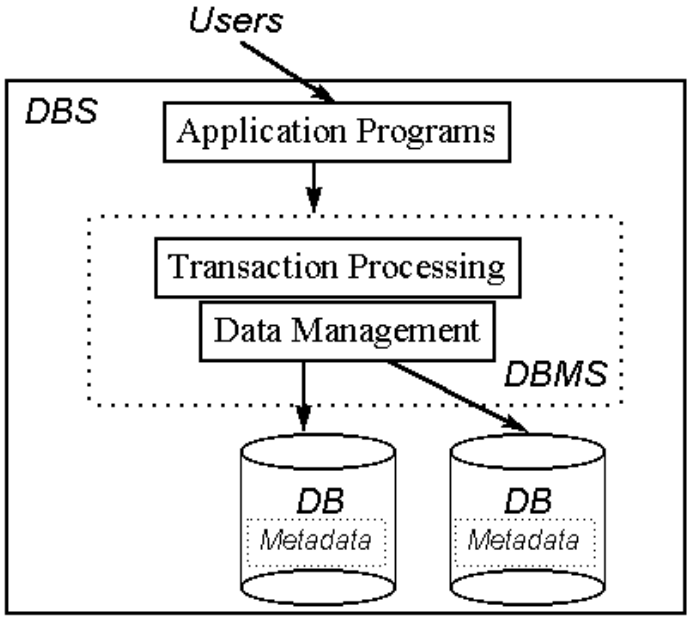
# Database Users

- Database administrators (DBA): responsible for authorizing access to the database, coordinating and monitoring its use, acquiring software and hardware resources, controlling its use and monitoring efficiency of operations
- Database designers: responsible for defining the content, the structure, the constraints, and functions or transactions against the database
- End users: use the data for queries, reports and some of them update the database content
  - Casual end users
  - Naive or parametric end users
  - Sophisticated end users
  - Stand-alone users



# Database Systems (DBS)

- A database system (DBS) is a system composed by DB's, a DBMS, application programs, users, and DBA
- Database systems are normally known as databases



# 1.2 Functions of DBMS



# 1.2 Functions of DBMS

- Data definition: defining data schemas
- Data storage: storage structures and access methods
- Data manipulation: data queries, data updates
- Data control: transaction management, data integrity, data security, concurrency control, failure recovery
- Data maintenance: data entry, data transformation, data archiving, data restoring, performance monitoring

# 1.3 Features of Database Systems



# 1.3 Features of Database Systems

- Structured data organization
- Data sharing
- Low data redundancy
- Data independency: physical independency and logical independency
- Data is managed by the DBMS

# 1.4 Data Models



## Data Abstraction

Data abstraction is the processes of mapping the real world to the computer world

Real world $\longrightarrow$ Information world $\longrightarrow$ Computer world

- Real world: human, animals, machines, ...
- Information world: entities, attributes, relationships
- Computer world: records, fields, references

# Data Models

Data models are important tools for *data abstraction*.

- A set of concepts to describe the structure of a database
- the operations for manipulating these structures
- certain constraints that the database should obey



# Categories of Data Models

- Conceptual (high-level, semantic) data models: provide concepts that are close to the way many users perceive data
- Physical (low-level, internal) data models: provide concepts that describe details of how data is stored in the computer.
- Implementation (representational) data models: provide concepts that fall between the above two, used by many commercial DBMS implementations (e.g. relational data models used in many commercial systems)

# Categories of Implementation Data Models

- Hierarchical data model
- Network data model
- Relational data model
- Object-oriented data model
- Relational-object data model
- Semi-structured data model
- Document data model
- Graph data model



# Constraints

- Entity constraints
- Referential constraints
- User-defined constraints

**COURSE**

| Course_name | Course_number | Credit_hours | Department |
|---|---|---|---|
| Intro to Computer Science | CS1310 | 4 | CS |
| Data Structures | CS3320 | 4 | CS |
| Discrete Mathematics | MATH2410 | 3 | MATH |
| Database | CS3380 | 3 | CS |

**SECTION**

| Section_identifier | Course_number | Semester | Year | Instructor |
|---|---|---|---|---|
| 85 | MATH2410 | Fall | 04 | King |
| 92 | CS1310 | Fall | 04 | Anderson |
| 102 | CS3320 | Spring | 05 | Knuth |
| 112 | MATH2410 | Fall | 05 | Chang |
| 119 | CS1310 | Fall | 05 | Anderson |
| 135 | CS3380 | Fall | 05 | Stone |

**GRADE_REPORT**

| Student_number | Section_identifier | Grade |
|---|---|---|
| 17 | 112 | B |
| 17 | 119 | C |
| 8 | 85 | A |
| 8 | 92 | A |
| 8 | 102 | B |
| 8 | 135 | A |

**PREREQUISITE**

| Course_number | Prerequisite_number |
|---|---|
| CS3380 | CS3320 |
| CS3380 | MATH2410 |
| CS3320 | CS1310 |

**Figure 1.2**
A database that stores student and course information.

# 1.5 Database Schemas



## Database Schemas

A database schema is the description of a database including the descriptions of the database structure, data types, and the constraints on the database

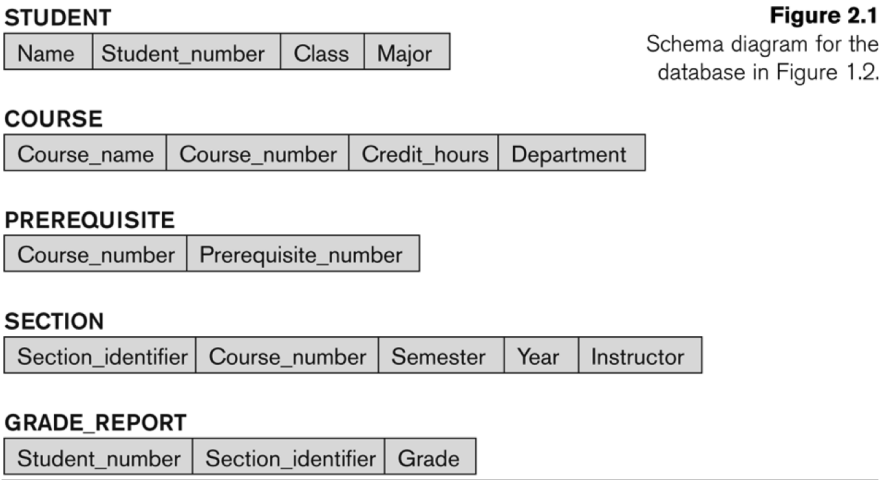



**Figure 2.1**
Schema diagram for the database in Figure 1.2.

# Database Instances

A database instance (database state or snapshot) is the actual data stored in a database at a particular moment in time

**COURSE**

| Course_name | Course_number | Credit_hours | Department |
|---|---|---|---|
| Intro to Computer Science | CS1310 | 4 | CS |
| Data Structures | CS3320 | 4 | CS |
| Discrete Mathematics | MATH2410 | 3 | MATH |
| Database | CS3380 | 3 | CS |

**SECTION**

| Section_identifier | Course_number | Semester | Year | Instructor |
|---|---|---|---|---|
| 85 | MATH2410 | Fall | 04 | King |
| 92 | CS1310 | Fall | 04 | Anderson |
| 102 | CS3320 | Spring | 05 | Knuth |
| 112 | MATH2410 | Fall | 05 | Chang |
| 119 | CS1310 | Fall | 05 | Anderson |
| 135 | CS3380 | Fall | 05 | Stone |

**GRADE_REPORT**

| Student_number | Section_identifier | Grade |
|---|---|---|
| 17 | 112 | B |
| 17 | 119 | C |
| 8 | 85 | A |
| 8 | 92 | A |
| 8 | 102 | B |
| 8 | 135 | A |

**PREREQUISITE**

| Course_number | Prerequisite_number |
|---|---|
| CS3380 | CS3320 |
| CS3380 | MATH2410 |
| CS3320 | CS1310 |

**Figure 1.2**
A database that stores student and course information.



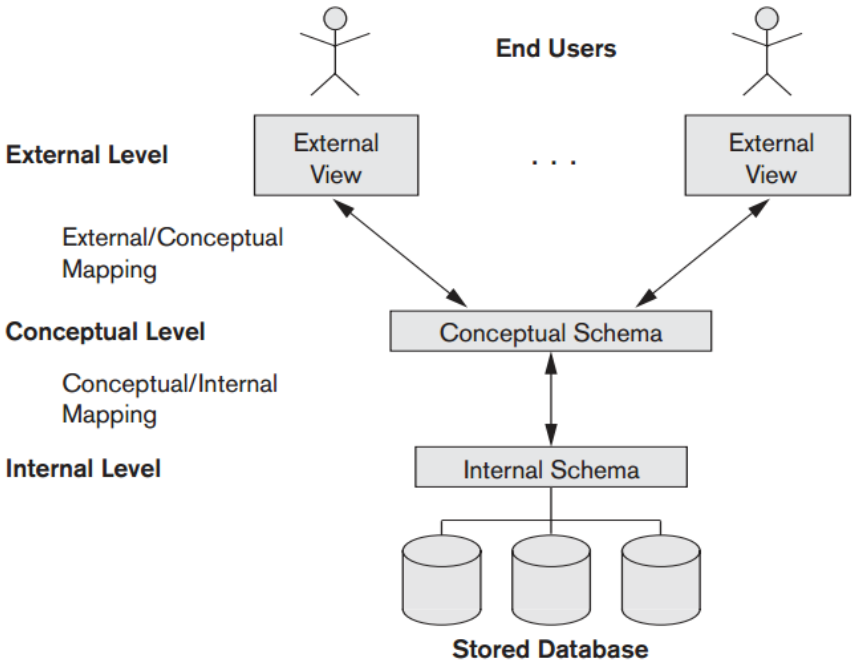
# Database Schemas vs. Database Instances

- The database schema is the "type"
- The database instance is the "value"

- The database schema changes very infrequently
- The database instance changes every time the database is updated

# Three-Schema Architecture

In the three-schema architecture, DBMS schemas are defined at three levels






# Three Levels of Schemas

## Internal schema (storage schema)

- Describe physical storage structures and access methods
- There is only one internal schema
- Typically use a physical data model

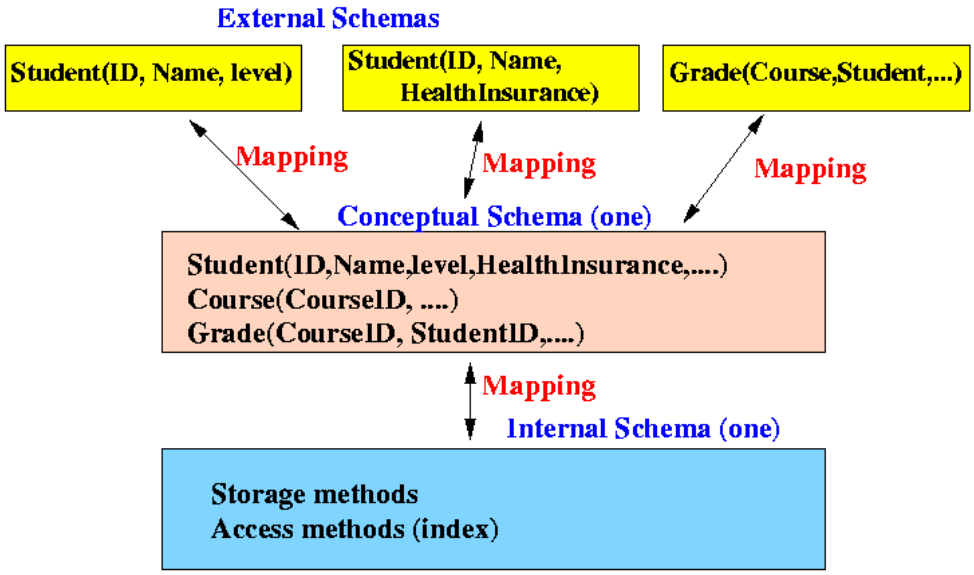
## Conceptual schema

- Describe the structure and constraints for the whole database for a community of users
- There is only one conceptual schema
- Use a conceptual or an implementation data model

## External schema

- Describe various user views
- There can be more than one external schemas
- Usually use the same data model as the conceptual schema

# Three-Schema Architecture: Example

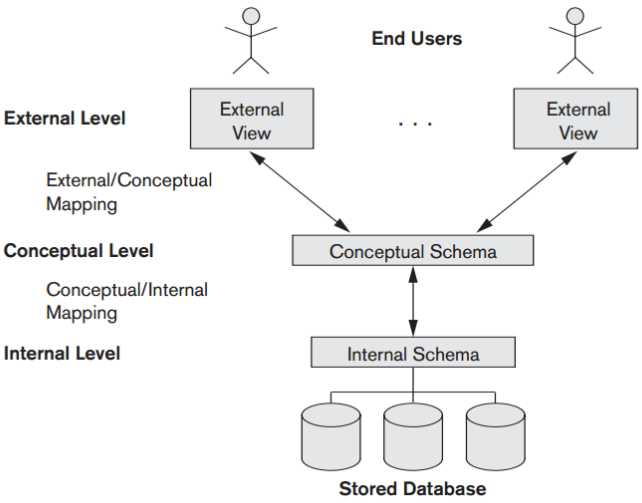





# Schema Mappings

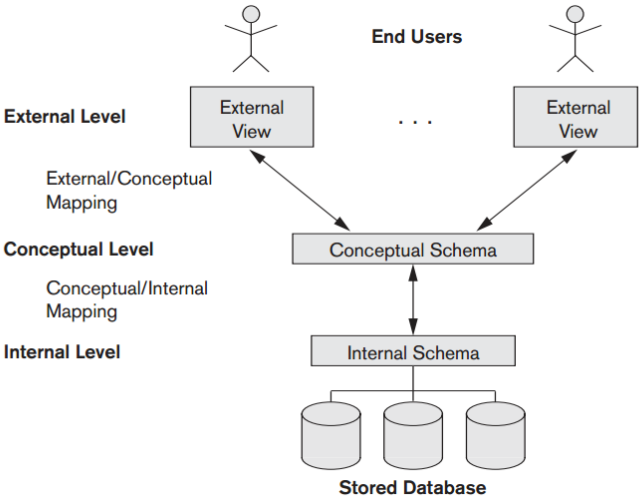
Mappings among schema levels are needed to transform requests and data

- Request transformation: Programs refer to an external schema, and are mapped by the DBMS to the internal schema for execution
- Data transformation: Data extracted from the internal DBMS level is reformatted to match the user's external view



# Categories of Schema Mappings

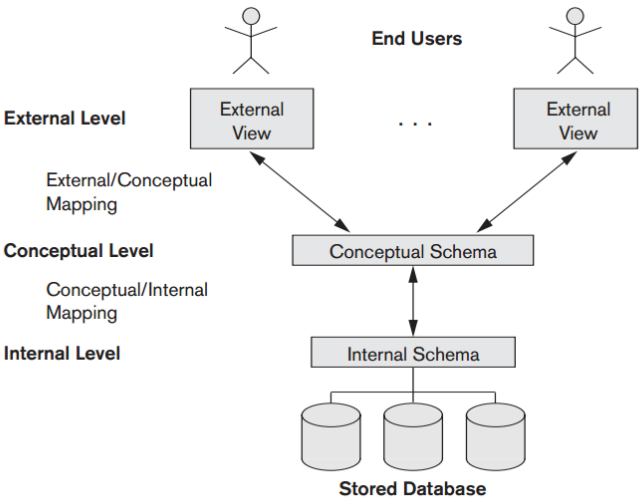
- External/conceptual mapping: mapping from an external schema to the conceptual schema
- Conceputual/internal mapping: mapping from the conceptual schema to the internal schema





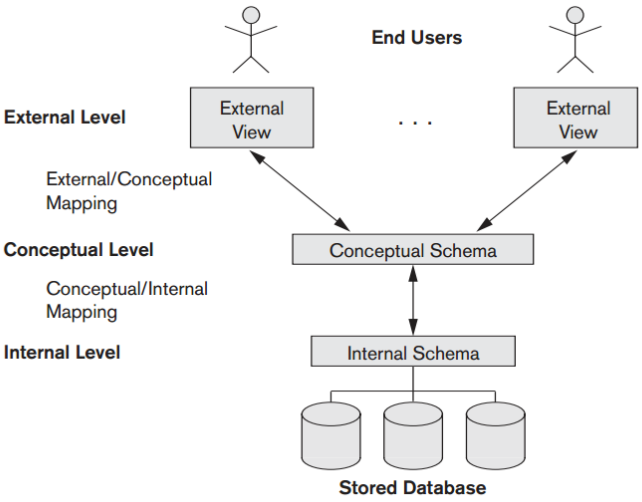

# Data Independence

- When a schema at a lower level is changed, only the mappings between this schema and the high-level schemas need to be changed
- The high-level schemas themselves are unchanged
- The application programs referring to the external schemas need not to be changed



## Categories of Data Independence

- Logical data independence: the capability to change the conceptual schema without having to change the external schemas as well as the application programs referring to them
- Physical data independence: the capability to change the internal schema without having to change the conceptual schema






# 1.6 Database Languages

## Categories of Database Languages

- Data definition languages (DDL): used by the DBA and database designers to specify the conceptual schema of a database
- Data manipulation languages (DML): used to specify queries and updates to a database



## Data Definition Languages (DDL)

- The DDL is used by the DBA and database designers to specify the conceptual schema of a database
- In many DBMSs, the DDL is also used to define internal and external schemas (views), e.g., SQL
- In some DBMSs, separate storage definition language (SDL) and view definition language (VDL) are used to define internal and external schemas

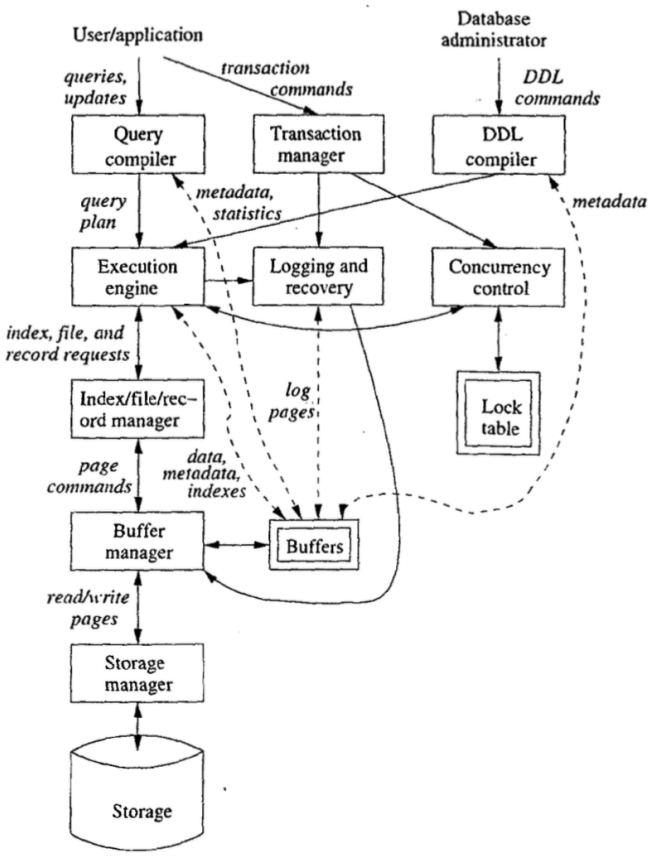
## Data Manipulation Languages (DML)

- The DML is used to specify database retrievals and updates
- DML commands (data sublanguage) can be embedded in a general-purpose programming language (host language)
- Stand-alone DML commands can be applied directly (called a query language)



# 1.7 DBMS Architectures

# DBMS Archetictures






# Syllabus

Part I: Fundamentals of Database Systems

- Chapter 1: Introduction
- Chapter 2: Relational Databases
- Chapter 3: Structured Query Language (SQL)

Part II: Database Design

- Chapter 4: Conceptual Database Design (The E-R Model)
- Chapter 5: Logical Database Design
- Chapter 6: Physical Database Design

Part III: Implementations of Database Management Systems

- Chapter 7: Storage Management
- Chapter 8: Query Processing
- Chapter 9: Query Optimization
- Chapter 10: Concurrency Control
- Chapter 11: Failure Recovery

# Summary



# Summary

- Basic Concepts
- Functions of Database Systems
- Features of Database Systems
- Data Models
- Database Schemas
- DBMS Languages
- DBMS Architectures