数据库系统

主讲：高宏

Who is Who

- 主讲: 高宏 (海量数据计算研究中心)
- 办公室: 科创大厦K1419
- 手机: 13945092842
- Email: honggao@hit.edu.cn

- 研究方向: 大数据、物联网

  时空序列数据分析、图与社交网络分析、
  数据质量、物联网数据采集与分析

- 学术兼职:

  大数据科学与工程黑龙江省重点实验室主任
  中国计算机学会 数据库专委会 副主任
  ACM China常务理事

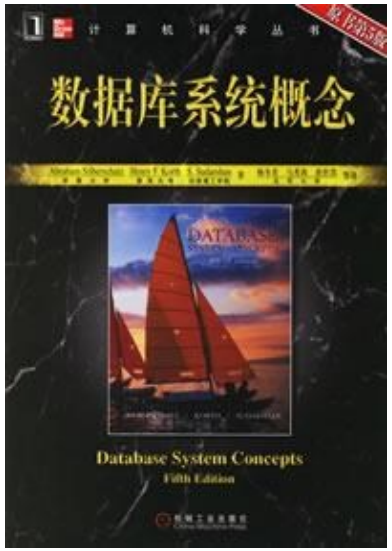
Who is Who

- 助教: 刘佳艺
- 办公室: 科创大厦 K1413房间
- Email: hitdatabase2021@163.com
- QQ群:

考试与成绩

- 成绩
  - 平时成绩： 20%
  - 课程实践： 20%
  - 期末考试： 60% 闭卷

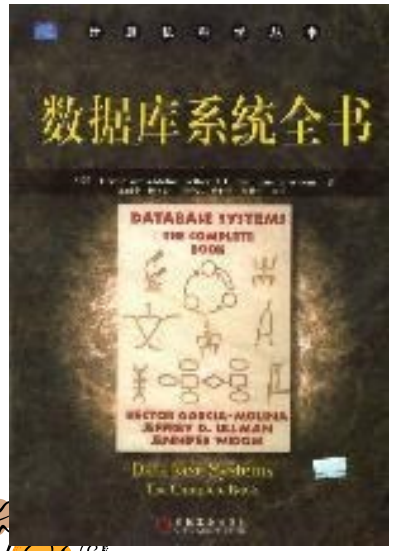
参考教材



Database Systems Concepts (最新到6th Edition)

原出版社： McGraw-Hill
作者： Abraham Silberschatz,
Henry F.Korth,
S.Sudarshan

译者： 杨冬青 马秀莉 唐世渭 等
出版社：机械工业出版社

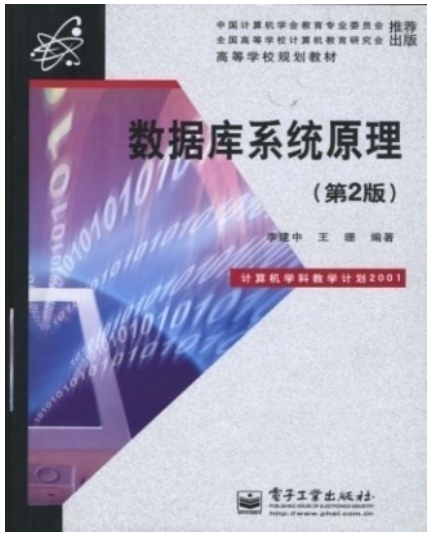


Database Systems: The Complete Book

原出版社： Prentice Hall/Pearson
作者： （美）**Hector Garcia-Molina,**
Jeffrey D.Ullman,
Jennifer Widom

译者： 岳丽华 杨冬青 龚育昌
唐世渭 徐其钧
出版社：机械工业出版社

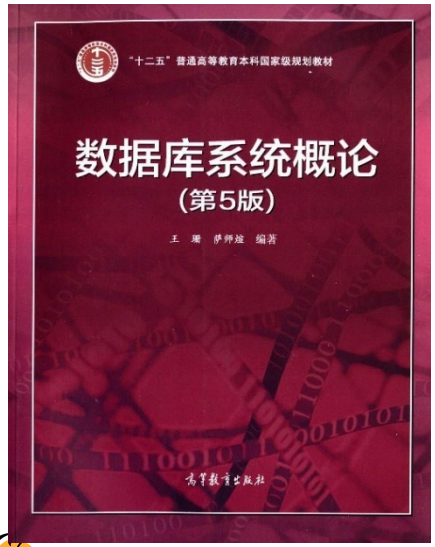
参考教材



数据库系统原理（第二版）

出版社：电子工业出版社

作者：李建中, 王 珊



数据库系统概论（第五版）

出版社：高等教育出版社

作者：王珊、萨师煊

- 美国斯坦福大学Jennifer Widom教授的教学视频

- 中国人民大学王珊教授教学团队MOOC视频
  - https://www.icourse163.org/

Top Conferences and Journals

- SIGMOD: ACM International Conference on Management of Data
- VLDB: International Conference on Very Large Data Bases
- ICDE: IEEE International Conference on Data Engineering
- SIGKDD: ACM Conference on Knowledge Discovery and Data Ming
- PODS: ACM Symposium on Principles of Database Systems

1. ACM Transactions on Database Systems
2. ACM Transactions on Information Systems
3. VLDB Journal
4. IEEE Transactions on Knowledge and Data Engineering

第一章　绪论

Why?
What?
How?

Outline

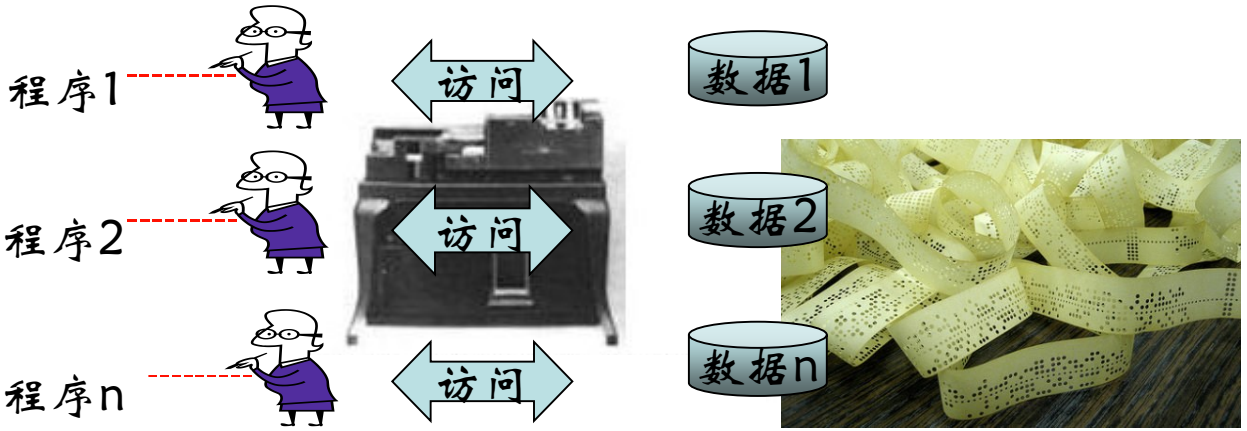
- Why数据库系统?
- 相关概念
- 数据独立性
- 数据模型与数据库模式
- 数据库系统的发展

Outline

- Why数据库系统?
- 相关概念
- 数据独立性
- 数据模型与数据库模式
- 数据库系统的发展

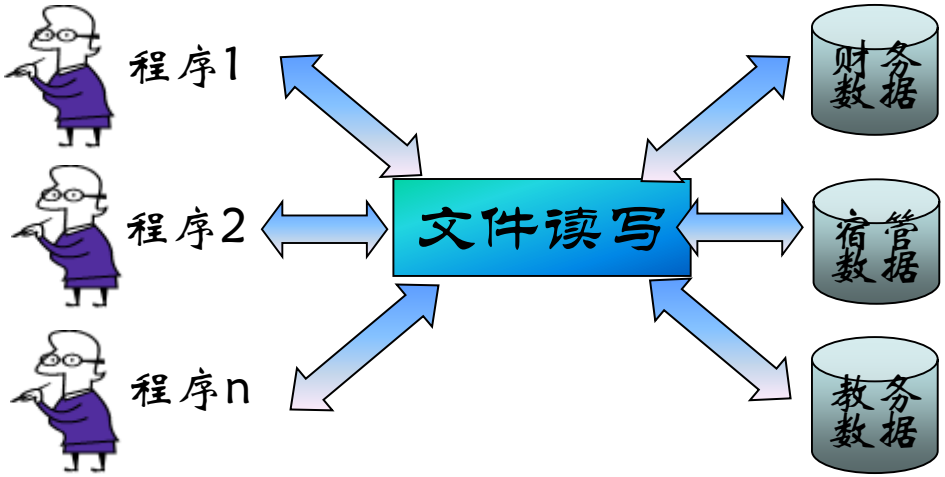
Why数据库系统？

- 数据管理技术的发展
  - 人工管理阶段(40年代中-50年代中)
    - 需求：科学计算
    - 环境：无直接存取的存储设备、无操作系统
    - 应用程序与数据关系：
      - 数据组织与访问与存储设备紧密耦合
      - 数据难以共享



Why数据库系统？

- 数据管理技术的发展
  - 文件系统阶段(50年代末-60年代中)
    - 需求：科学计算、管理
    - 环境：出现磁盘、文件系统
    - 应用程序与数据关系：
      - 屏蔽了存储设备差异，但数据访问与数据组织紧耦合
      - 数据冗余不一致、文件相互孤立、缺少完整性约束



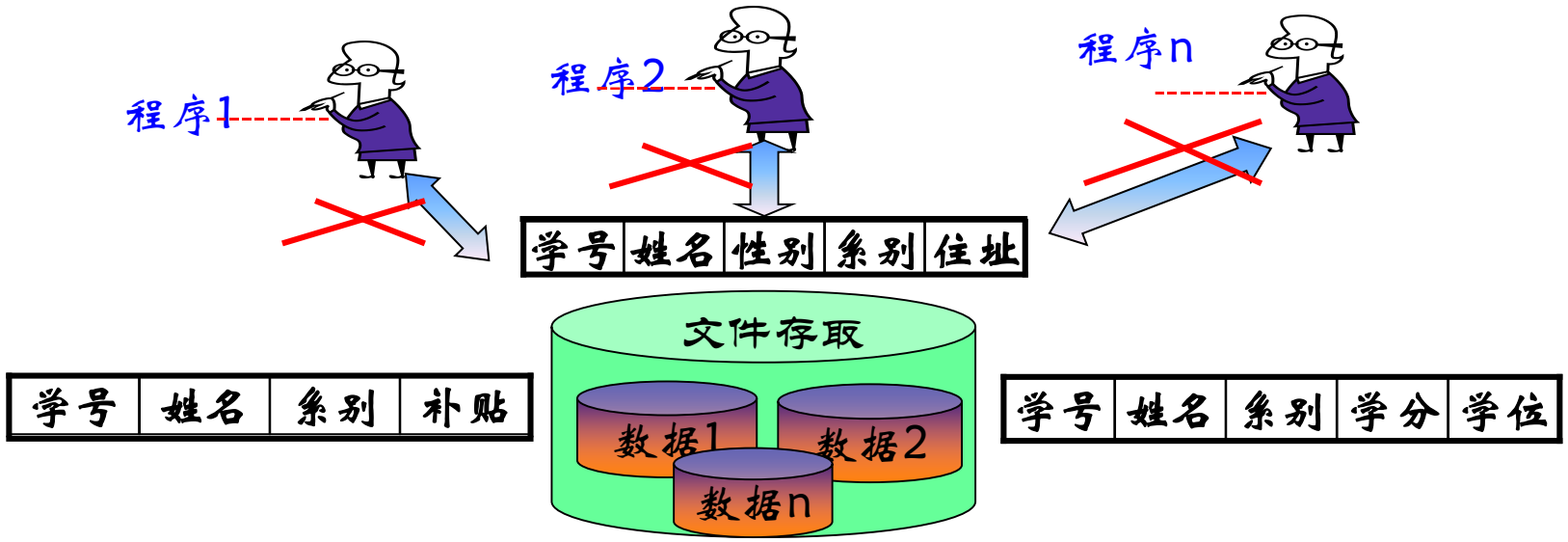
学号	姓名	系别	补贴

学号	姓名	性别	系别	住址

学号	姓名	系别	学分	学位

Why数据库系统？

- 数据管理技术的发展
  - 数据库系统产生(60年代末)
    - 需求：大规模数据访问与数据共享
    - 环境：大容量磁盘



数据维护代价巨大！ 例如：更新系别？ 增加学生？

Why数据库系统？

- 数据管理技术的发展
  - 数据库系统产生(60年代末)
    - 需求：大规模数据访问与数据共享
    - 环境：大容量磁盘

学号	姓名	性别	年龄	系别	住址	学位	补贴	学分	出生地

视图

财务部视图

学号	姓名	系别	补贴

宿管科视图

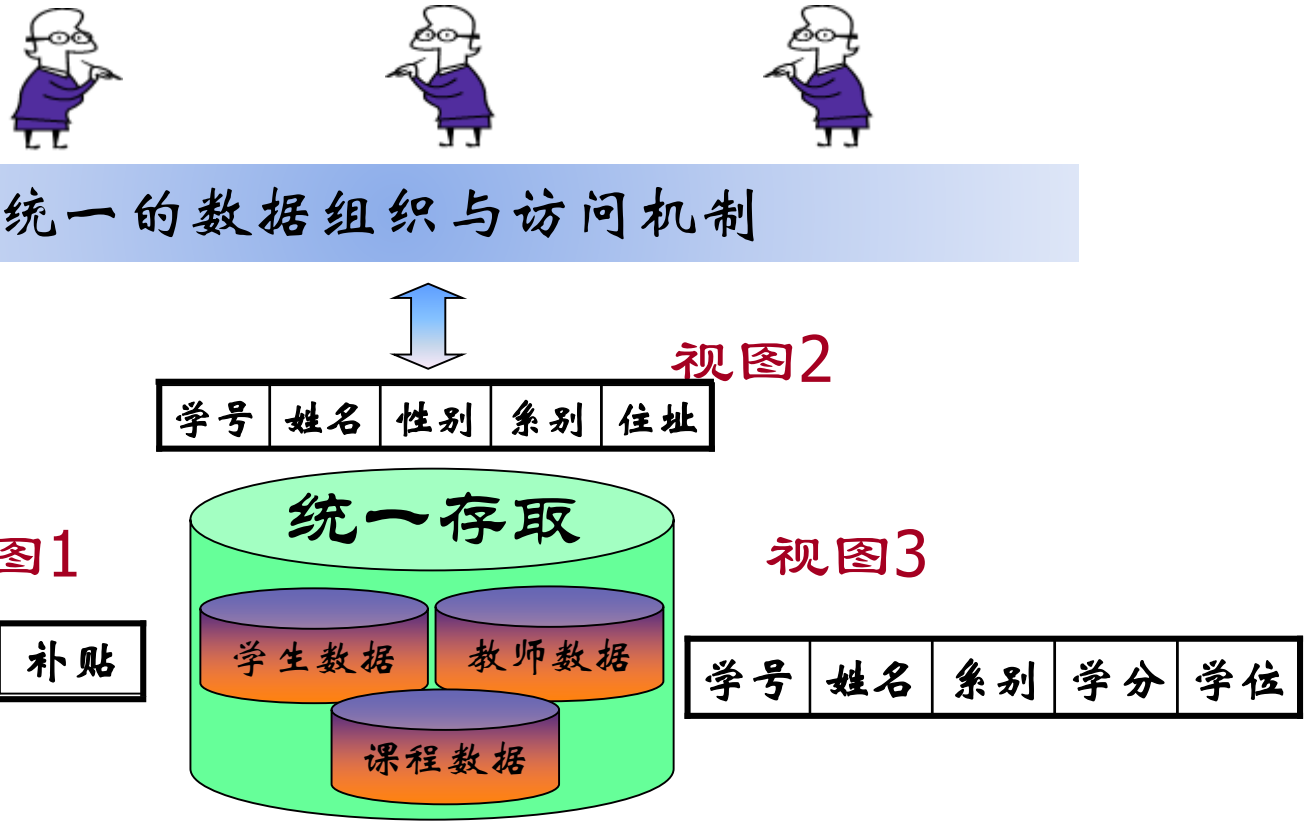
学号	姓名	性别	系别	住址

教务处视图

学号	姓名	系别	学分	学位

虚文件：
只有文件名及数据模式
没有数据内容

- 数据管理技术的发展
  - 数据库系统产生(60年代末)
    - 应用程序与数据关系：统一的数据组织与访问机制

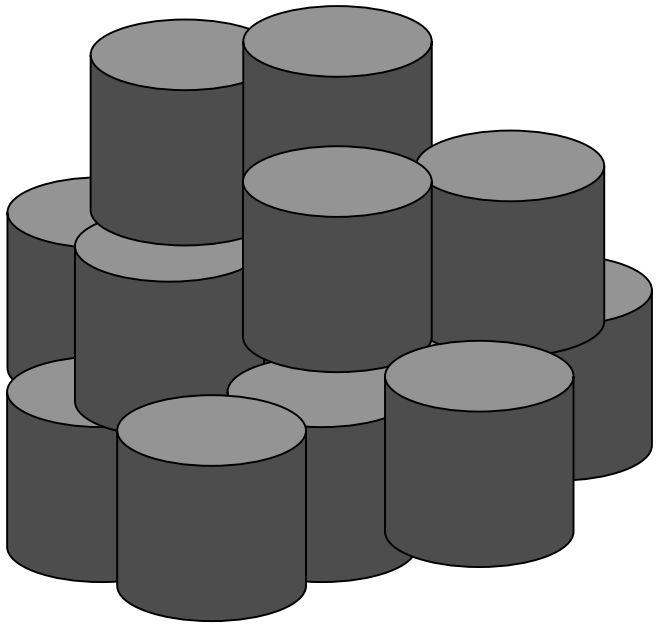


Outline

- Why数据库系统?
- 相关概念
- 数据独立性
- 数据模型与数据库模式
- 数据库系统的发展

什么是数据库?

- 数据库
  - 是长期储存在计算机内、有组织的、可共享的数据的集合
  - 若干不同的数据集合组成
- 数据库的特征
  - 长期储存，不是临时的
  - 数据按一定的形式组织、描述和储存
  - 可为各种用户共享



什么是数据库？

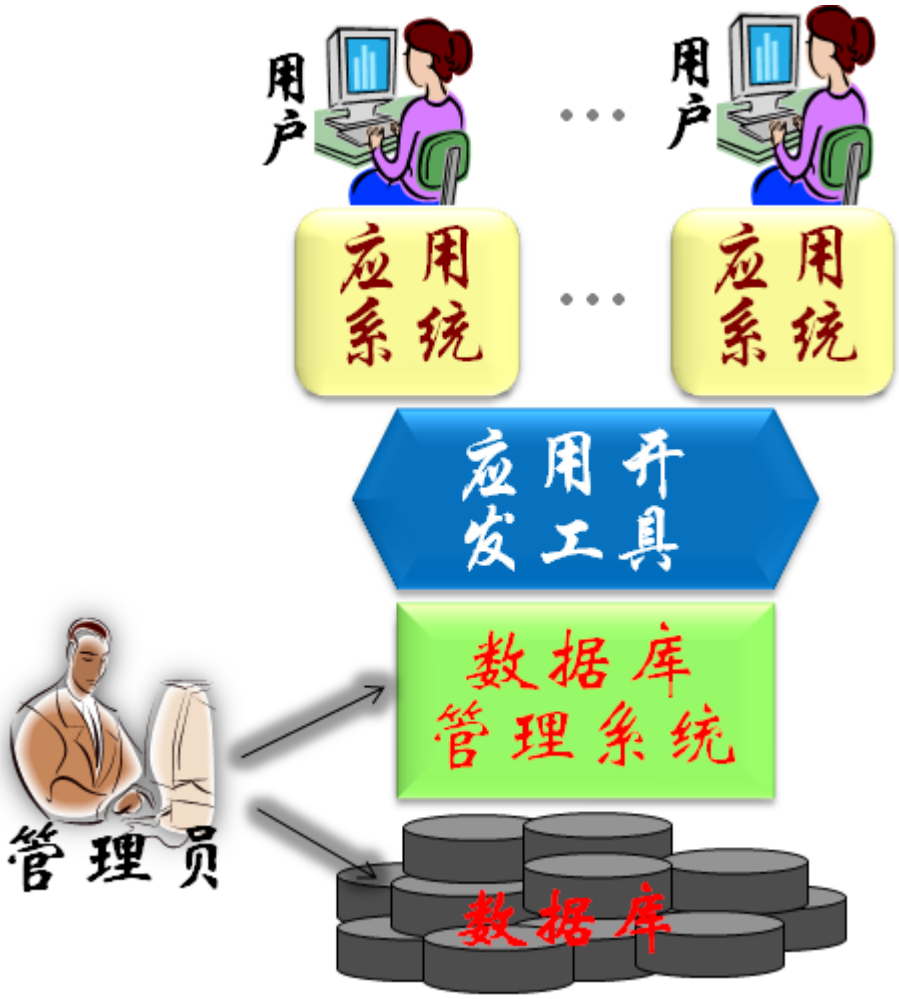
- 数据库的类型
  - 按照数据的类型
    - 简单结构数据库：如关系数据库，时空数据库
    - 复杂结构数据库：如图数据库
    - 半结构化数据：如XML数据库
    - 非结构化数据：如文本、音视频、图像等多媒体数据库
  - 数据存储的方式
    - 单机数据库
    - 分布式数据库
    - 并行数据库
    - 网络数据库，……
  - 数据存储的介质、时长
    - 外存数据库、内存数据库、流数据库

什么是数据库管理系统?

- 数据库管理系统:管理数据库，支持应用的软件系统
  DataBase Management System (DBMS)
  - 定义、存储、维护数据
  - 数据查询
  - 确保数据一致、安全、完整
  - 并发控制与事务处理
- 数据字典 (Meta Data)
  - 是用来描述数据库管理系统中各种信息的数据

什么是数据库系统?

- 数据库系统
  - 观点一：
    - 数据库
    - 数据库管理系统
  - 观点二：
    - 数据库
    - 数据库管理系统
    - 数据库管理员
    - 应用开发工具
    - 应用系统
    - 数据库系统用户
      - 数据库管理员
      - 数据库设计员
      - 应用程序员
      - 最终用户



数据库、数据库管理系统、数据库系统

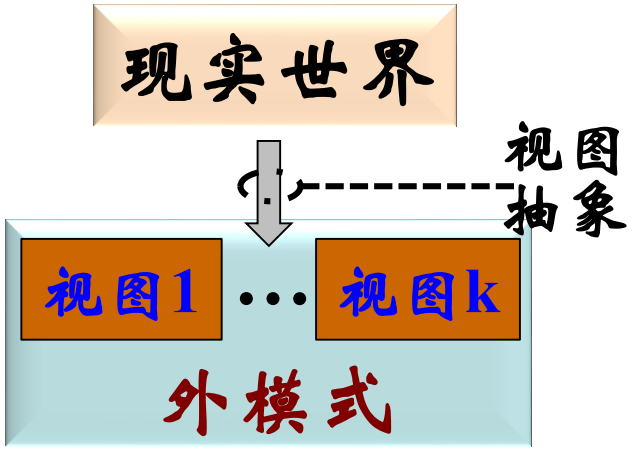
- 多种看法
  - 数据库管理系统=数据库+一组管理软件
    - 数据库管理系统=数据库系统
  - 数据库系统=数据库+数据库管理系统
  - 数据库=数据库系统

Outline

- Why数据库系统？
- 相关概念
- **数据独立性**
- 数据模型与数据库模式
- 数据库系统的发展

数据抽象与三级模式

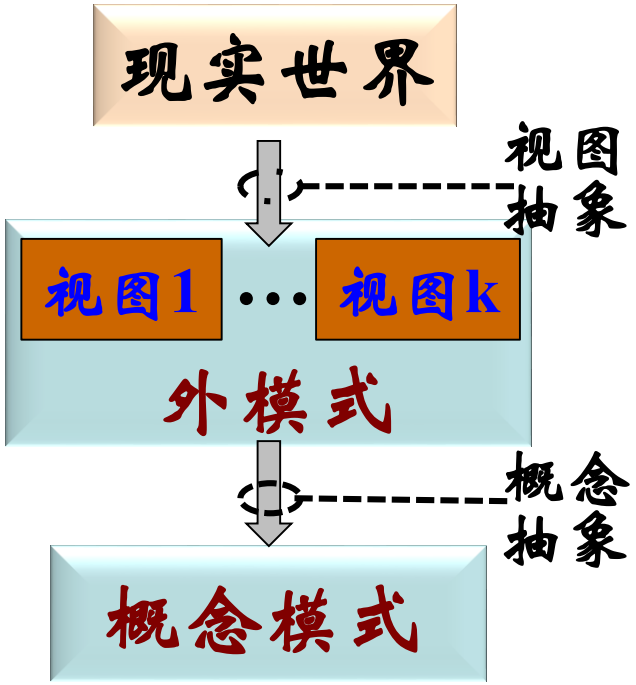
- 视图抽象
  - 把现实世界信息按不同用户观点抽象为多个逻辑数据结构，每个逻辑结构称为一个视图，描述了每个用户所关心的数据
  - 所有视图的集合形成了数据库的外模式



财务处	学号	姓名	系别	补贴			
宿管科	学号	姓名	性别	系别	住址		
教务处	学号	姓名	系别	学分	学位		
学生处	学号	姓名	性别	系别	年龄	学位	出生地

数据抽象与三级模式

- 概念抽象
  - 综合外模式中所有视图，把所有用户关心的现实世界抽象为概念模式，形成数据库整体逻辑结构

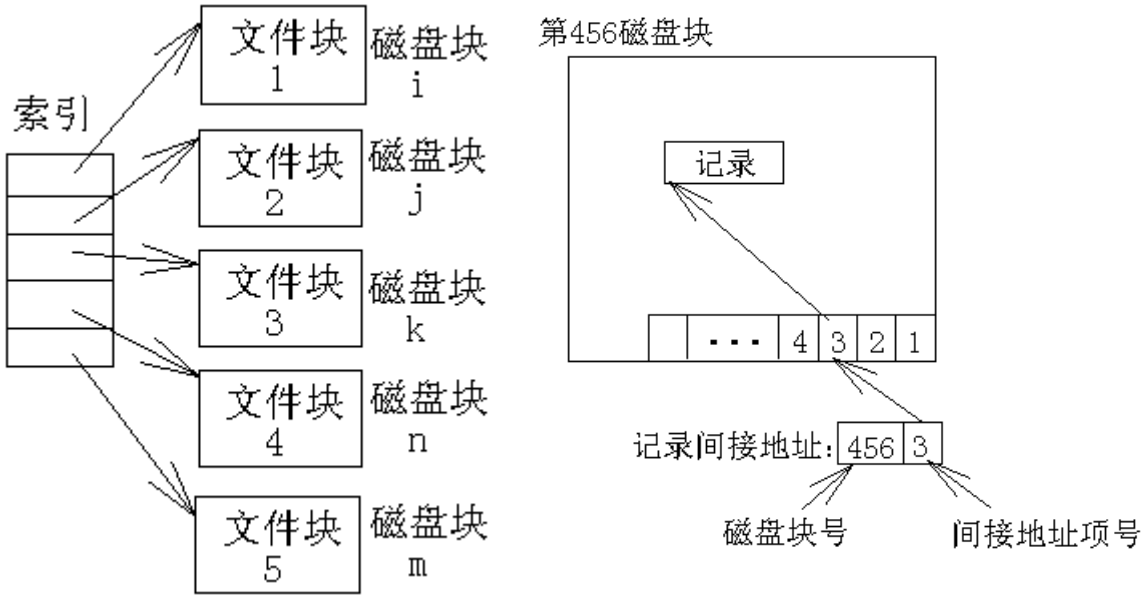


财务处	学号	姓名	系别	补贴		
宿管科	学号	姓名	性别	系别	住址	
教务处	学号	姓名	系别	学分	学位	
学生处	学号	姓名	性别	系别	年龄	学位	出生地

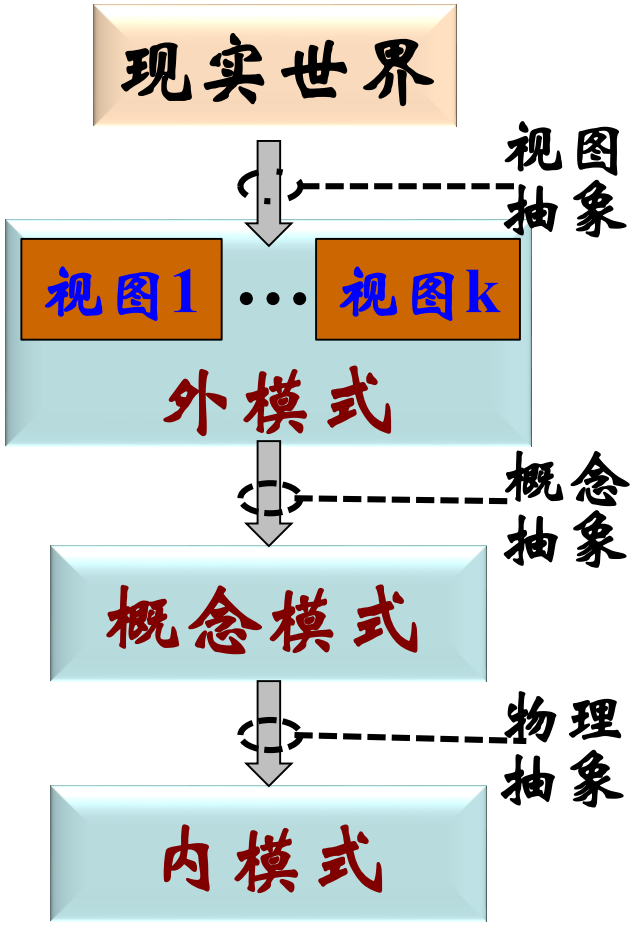
学号	姓名	性别	年龄	系别	住址	学位	补贴	学分	出生地

数据抽象与三级模式

- 物理抽象
  - 对概念模式进行抽象成为数据库的内模式，确定如何在物理存储设备上存储数据库



索引存储方法



数据的三级模式

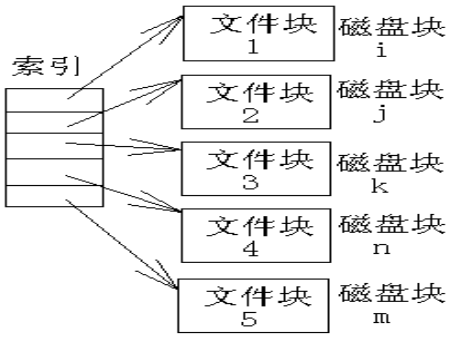
财务处 | 学号 | 姓名 | 系别 | 补贴 |

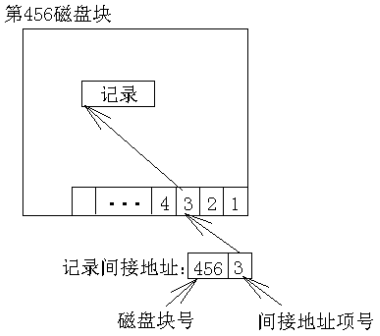
宿管科 | 学号 | 姓名 | 性别 | 系别 | 住址 |

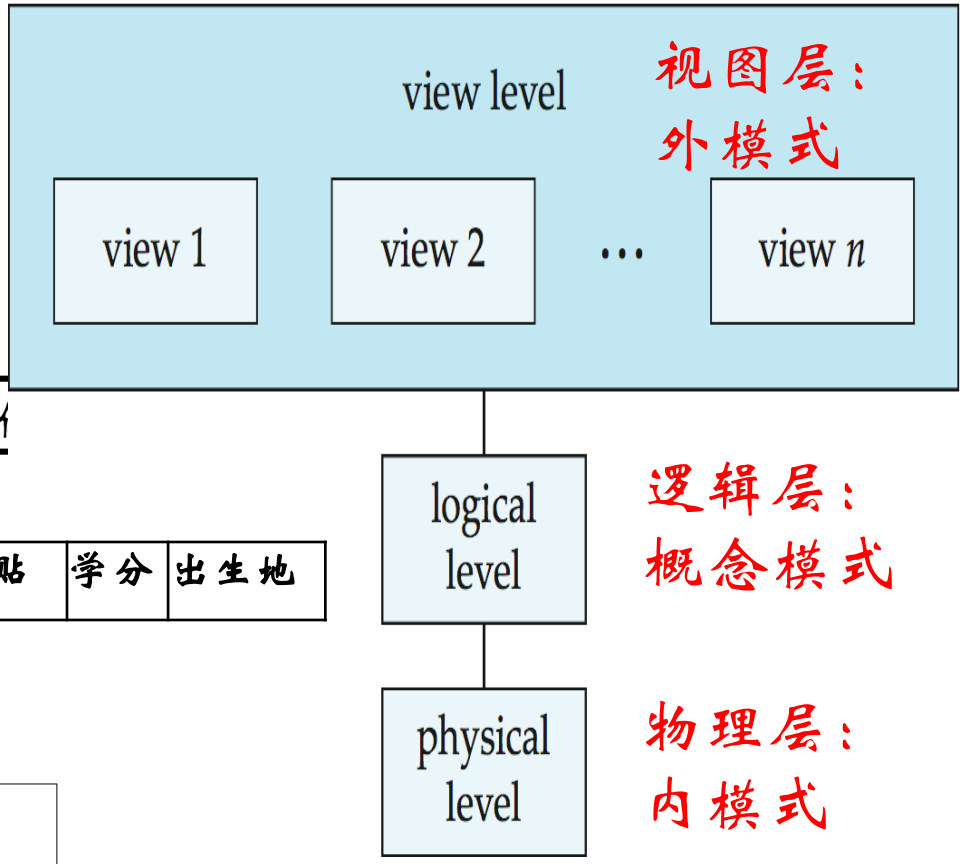
教务处 | 学号 | 姓名 | 系别 | 学分 | 学位 |

学生处 | 学号 | 姓名 | 性别 | 系别 | 年龄 | 学…

学号	姓名	性别	年龄	系别	住址	学位	补贴	学分	出生地

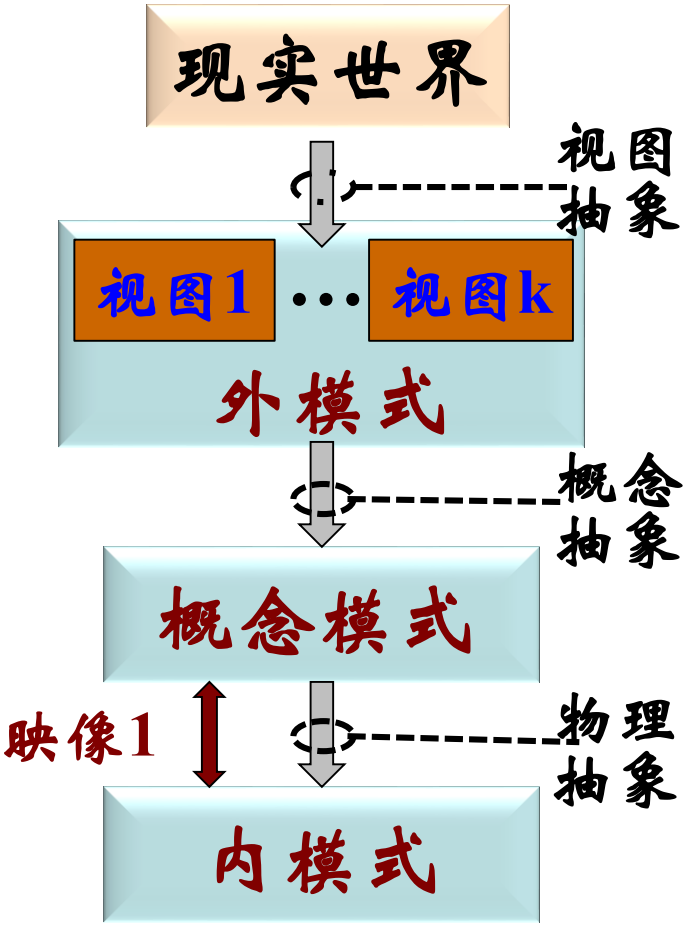






两层映像与数据独立性

- 物理数据独立性
  - 由内模式/概念模式映像实现
  - 数据库内模式发生改变时
    - 仅需修改内模式/逻辑模式映像
    - 数据的逻辑结构不变
    - 应用程序可以不变
  - 例如
    - 按行存储→按列存储
    - 编码存储
    - 压缩存储
    - ……

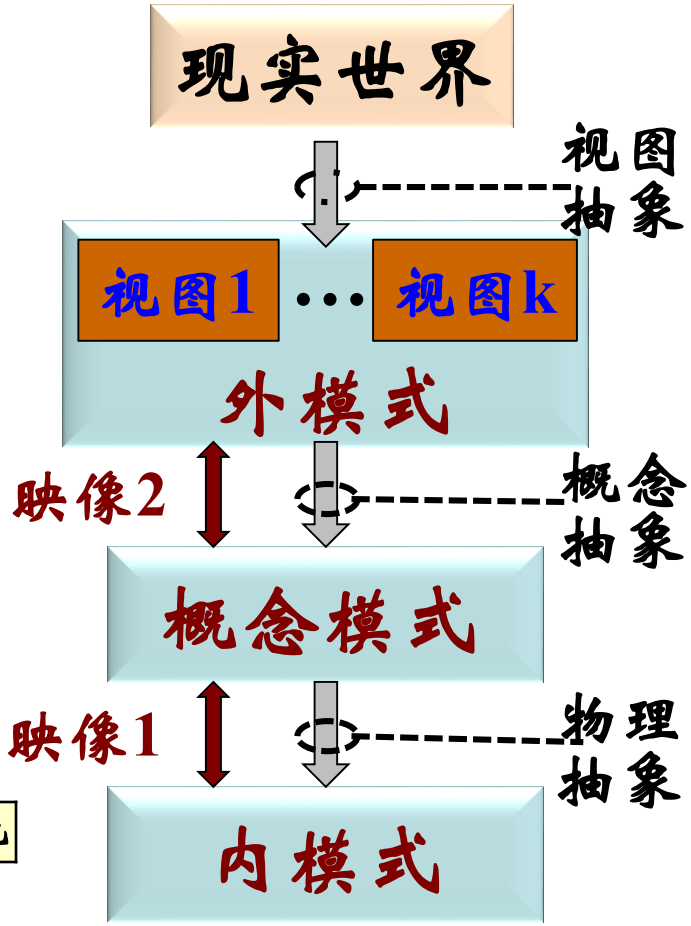


两层映像与数据独立性

- 逻辑数据独立性
  - 由概念模式/外模式映像实现
  - 当概念模式发生改变时
    - 仅需修改逻辑模式/外模式映像
    - 数据库的外模式不变
    - 应用程序可以不变
  - 例如：
    - 根据需要，增加属性

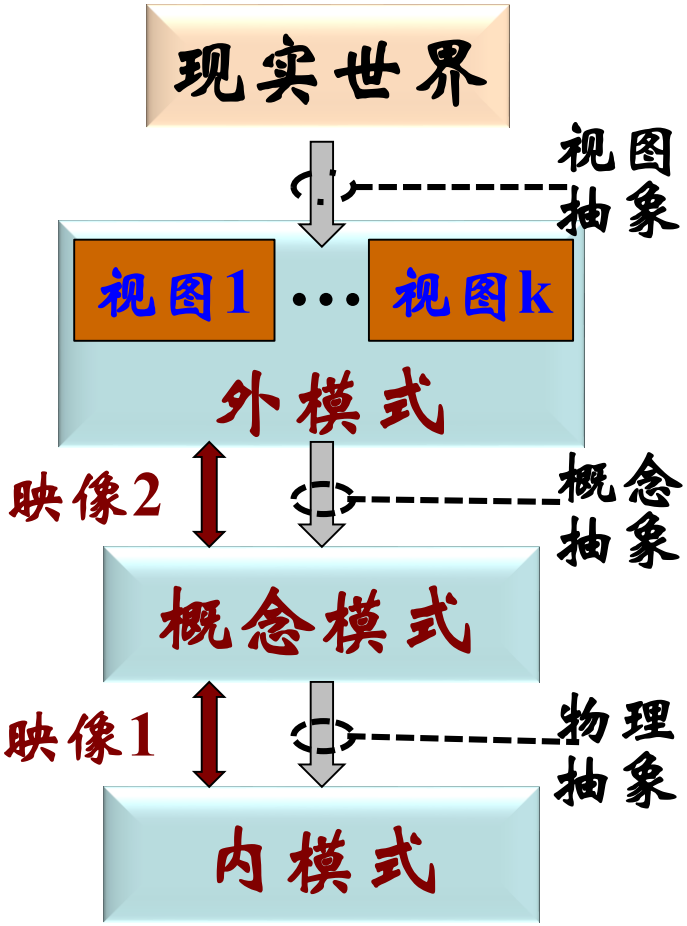
学号	姓名	性别	年龄	系别	住址	学位	补贴	学分	出生地

学号	姓名	性别	年龄	系别	住址	学位	补贴	学分	电话	出生地



三级模式与数据独立小结

- 三层抽象
  - 视图抽象→外模式
  - 概念抽象→概念模式
  - 物理抽象→内模式
- 两级独立性
  - 物理独立性
    - 内模式发生改变时，概念模式可以不变
  - 逻辑独立性
    - 概念模式发生改变时，外模式可以不变



Outline

- Why数据库系统？
- 相关概念
- 数据独立性
- **数据模型与数据库模式**
- 数据库系统的发展

数据模型及基本要素

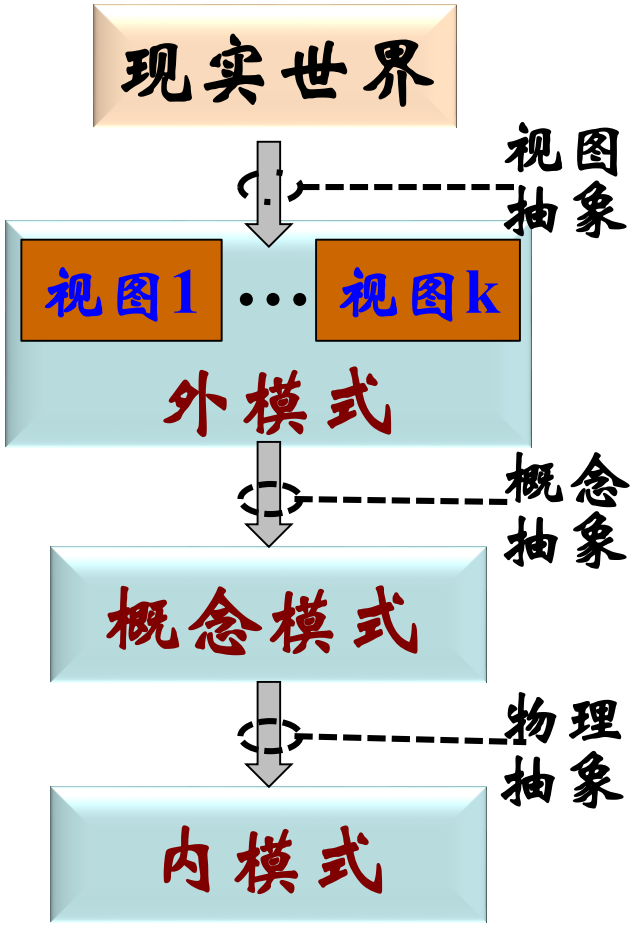
- 数据模型

  描述数据及数据之间关系、基本操作、满足的约束等

- 基本要素
  - 数据结构
    - 描述现实世界对象的信息结构
      - 对象的每个属性的数据类型、长度等
    - 描述对象之间联系的信息结构
  - 数据操作
    - 数据查询、维护等操作
  - 数据的完整性约束
    - 完整性规则的集合
    - 规定了数据必须遵守的语义约束条件

常用的数据模型

- 实体关系（E-R）数据模型
  - 用于视图抽象和外模式的定义
- 面向对象（O-O）数据模型
  - 用于视图抽象和外模式的定义
  - 用于概念抽象和逻辑模式定义
- 关系数据模型
  - 用于概念抽象和逻辑模式定义
- 对象关系 (O-R)数据模型
  - 用于概念抽象和逻辑模式定义
- 层次和网络数据模型
  - 用于概念抽象和逻辑模式定义



从不同角度、不同层面对数据建模

数据库模式

- 模式(Schema)
  - 用给定的数据模型刻画一组特定数据集合的逻辑结构与特征
  - 即特定数据集合的结构定义
    - 是数据间关系的抽象描述，与具体值无关

例如：学生成绩数据集合的模式(姓名,课程编号,成绩,学期,年度)

- 实例(Instance)
  - 是模式在某一时刻的一个具体值，即：数据集合的

Grade_Report

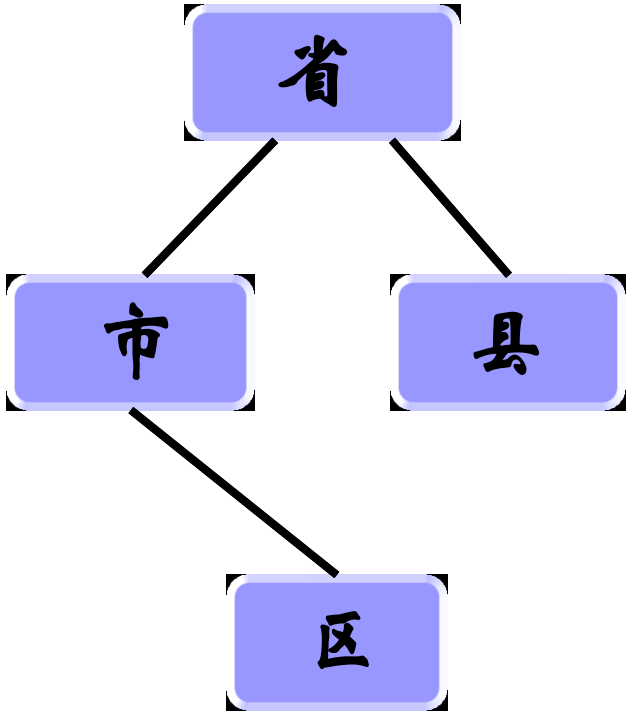
姓名	课程编号	成绩	学期	年度
常红	计 14	86	一	1994
常红	计 15	93	一	1994
都薇	数 1	89	二	1995
都薇	数 13	90	一	1995
都薇	计 14	85	一	1995

Prof-Course

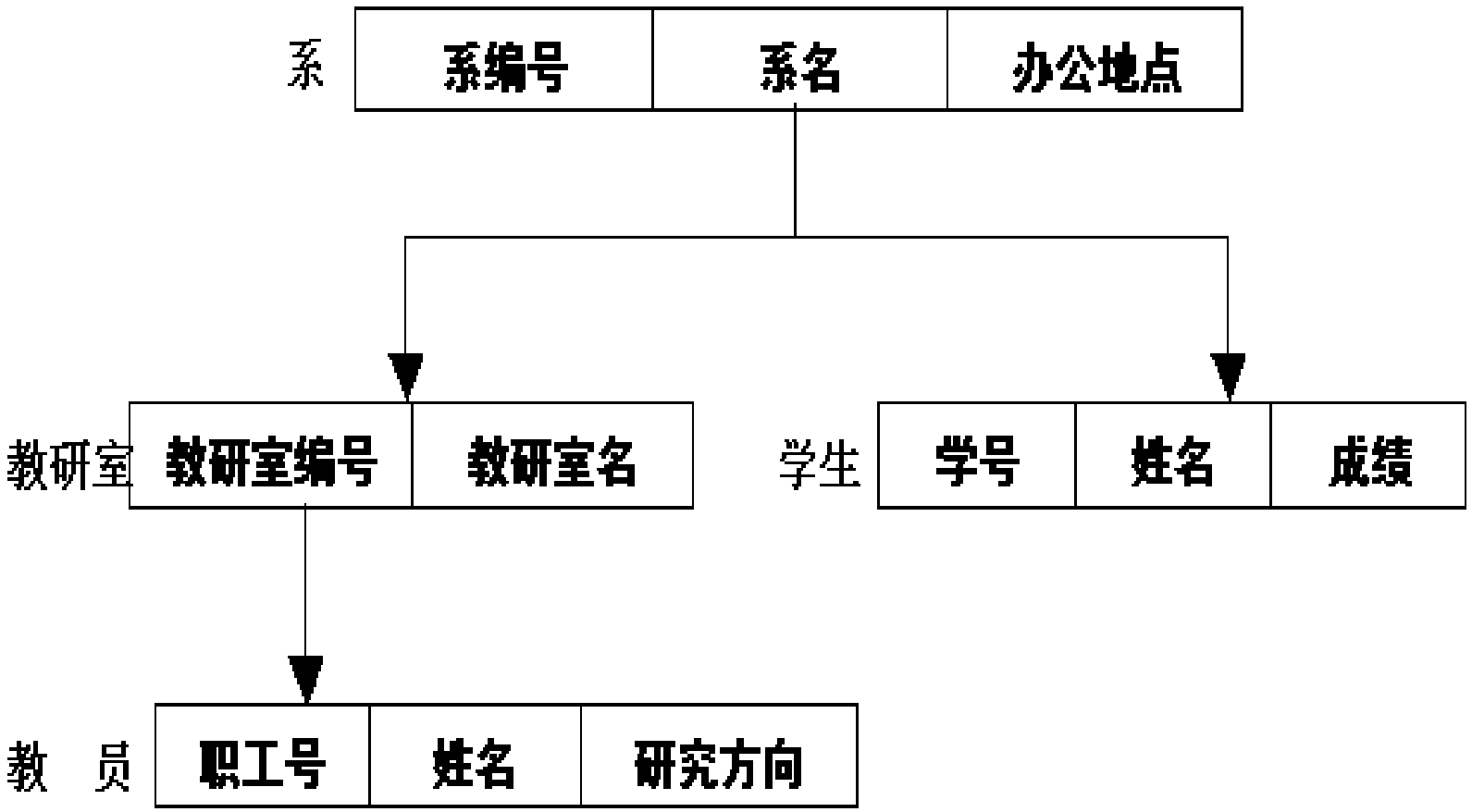
教师名	课程编号
刘德祥	数 01
刘德祥	数 13
陈庆奎	计 15
李金宝	计 17
孙文隽	计 14

层次数据模型的数据结构

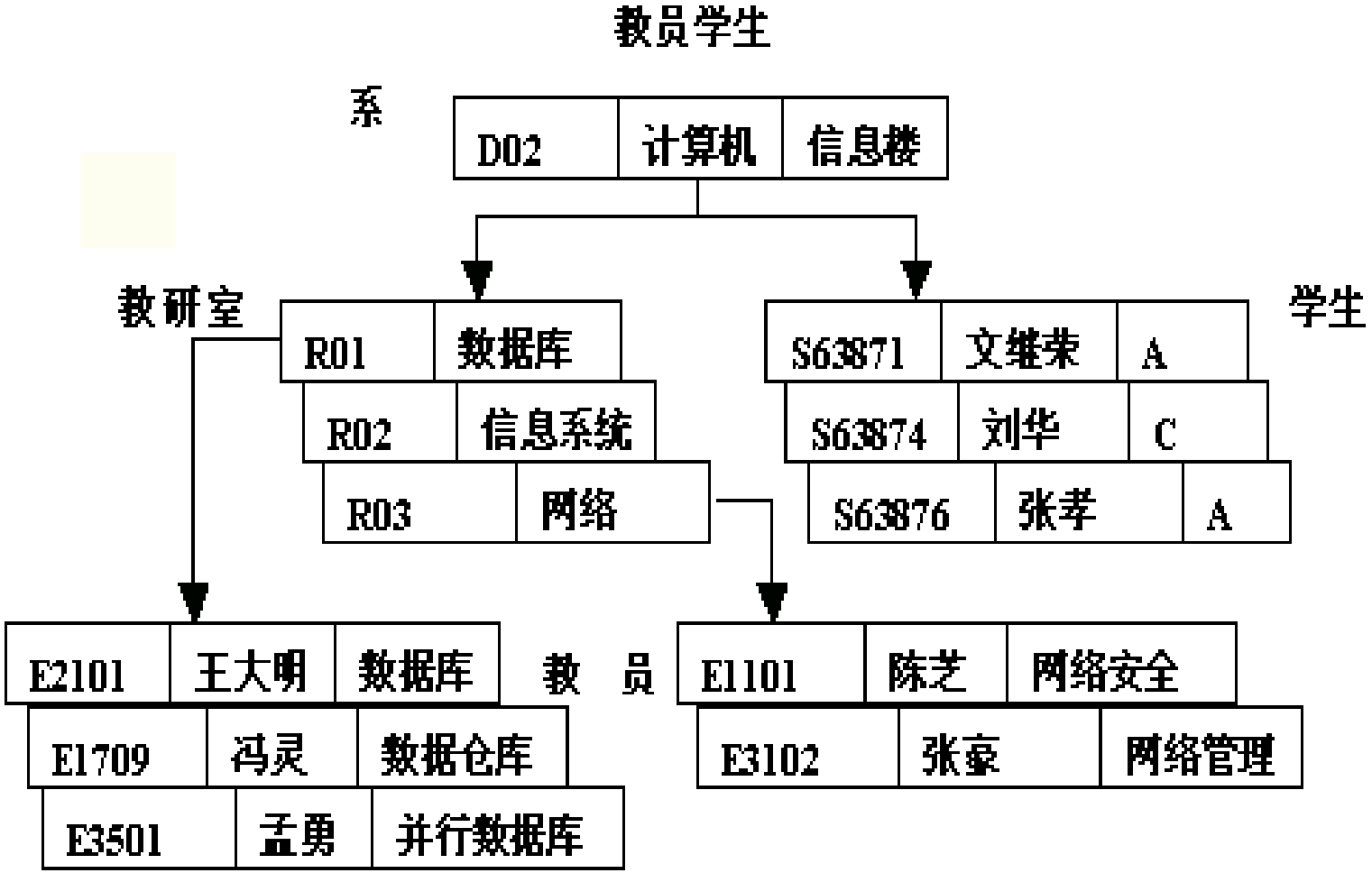
- 层次模型的数据结构是满足下列条件的树
  - 每个结点：现实世界的对象的一个抽象（一类对象），又称实体。
    - 结构:可包含若干个属性
      数据记录型表示
    - 属性：各种数据类型描述
  - 边表示现对象之间的联系
    - 表示对象(实体)之间一对多联系



“系-教员-教研室-学生”一个层次数据库模式



数据库的一个实例


教员学生
系
D02
计算机
信息楼
教研室
R01
数据库
R02
信息系统
R03
网络
学生
S63871
文维荣
A
S63874
刘华
C
S63876
张孝
A
教 员
E2101
王大明
数据库
E1709
冯灵
数据仓库
E3501
孟勇
并行数据库
E1101
陈芝
网络安全
E3102
张豪
网络管理

层次模型的优缺点

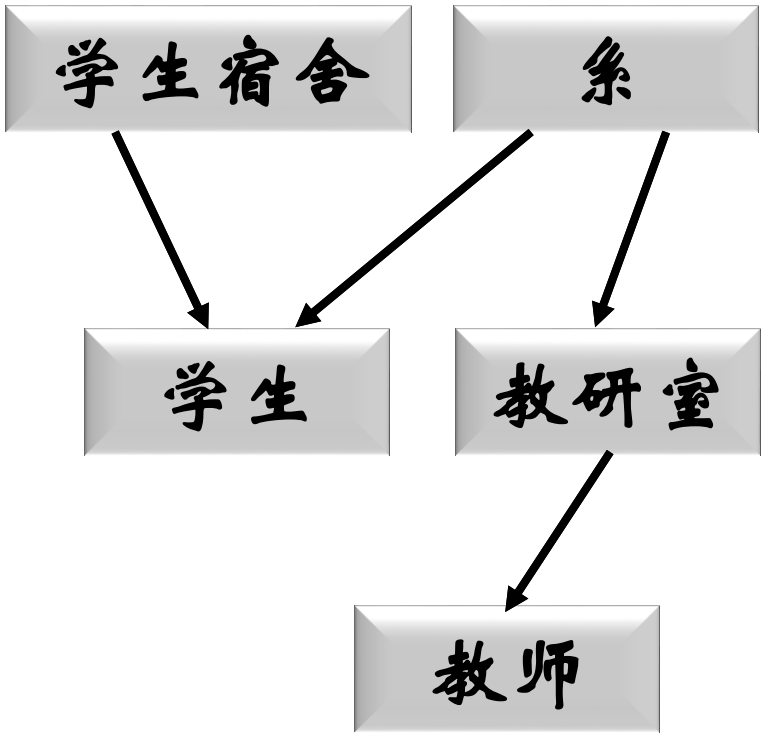
- 优点
  - 层次数据模型简单，对具有一对多的层次关系的现实世界描述自然、直观，容易理解
  - 性能优于关系模型，不低于网状模型
  - 层次数据模型提供了良好的完整性支持
- 缺点
  - 多对多联系表示不自然
  - 对插入和删除操作的限制多
  - 查询子女结点必须通过双亲结点
  - 面向过程

具有代表性的层次数据库系统

- IMS数据库管理系统
  - 第一个大型商用DBMS
  - 1968年推出
  - IBM公司研制

网状数据模型的数据结构

- 网状模型的数据结构是满足下列条件的图
  - 每个结点是一个对象记录
    - 实体型：数据记录型表示
    - 每个实体型可包含若干个实体属性
    - 实体属性：用字段描述
  - 边表示对象之间的联系
    - 表示实体之间一对多联系
    - 容易间接表示实体之间的多对多联系
  - 允许多个结点无双亲结点
  - 允许结点有多个双亲结点
  - 允许两个结点之间有多种联系（复合联系）
- 层次模型是网状模型特例



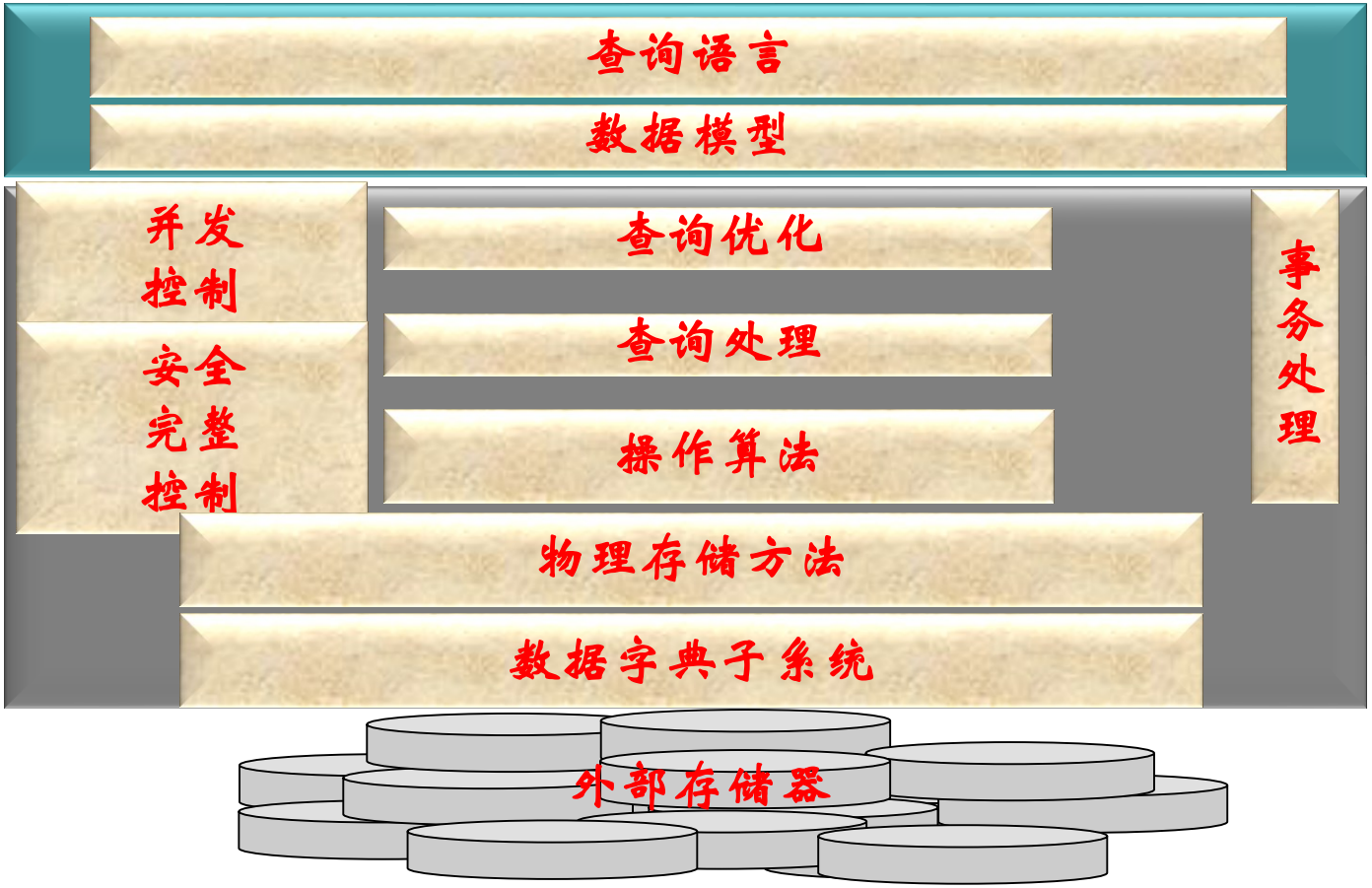
网状数据模型的优缺点

- 优点
  - 能够更为直接地描述现实世界，如一个结点可以有多个双亲
  - 具有良好的性能，存取效率较高
- 缺点
  - 结构比较复杂，不利于最终用户掌握
  - 定义与操作语言复杂，用户不容易使用
  - 面向过程

典型的网状数据库系统

- 最早的网状数据库管理系统IDS(1964)
  - Charles W．Bachman(网状数据库之父)
  - 美国数据系统语言委员会CODASYL下属的数据库任务组DBTG于1971年推出了第一个正式报告——DBTG报告，成为数据库历史上具有里程碑意义的文献
    - 奠定了网状数据库系统的概念、方法和技术
    - 基于IDS的经验所确定的方法称为DBTG方法或CODASYL方法，所描述的网状模型称为DBTG模型或CODASYL模型

数据库管理系统结构



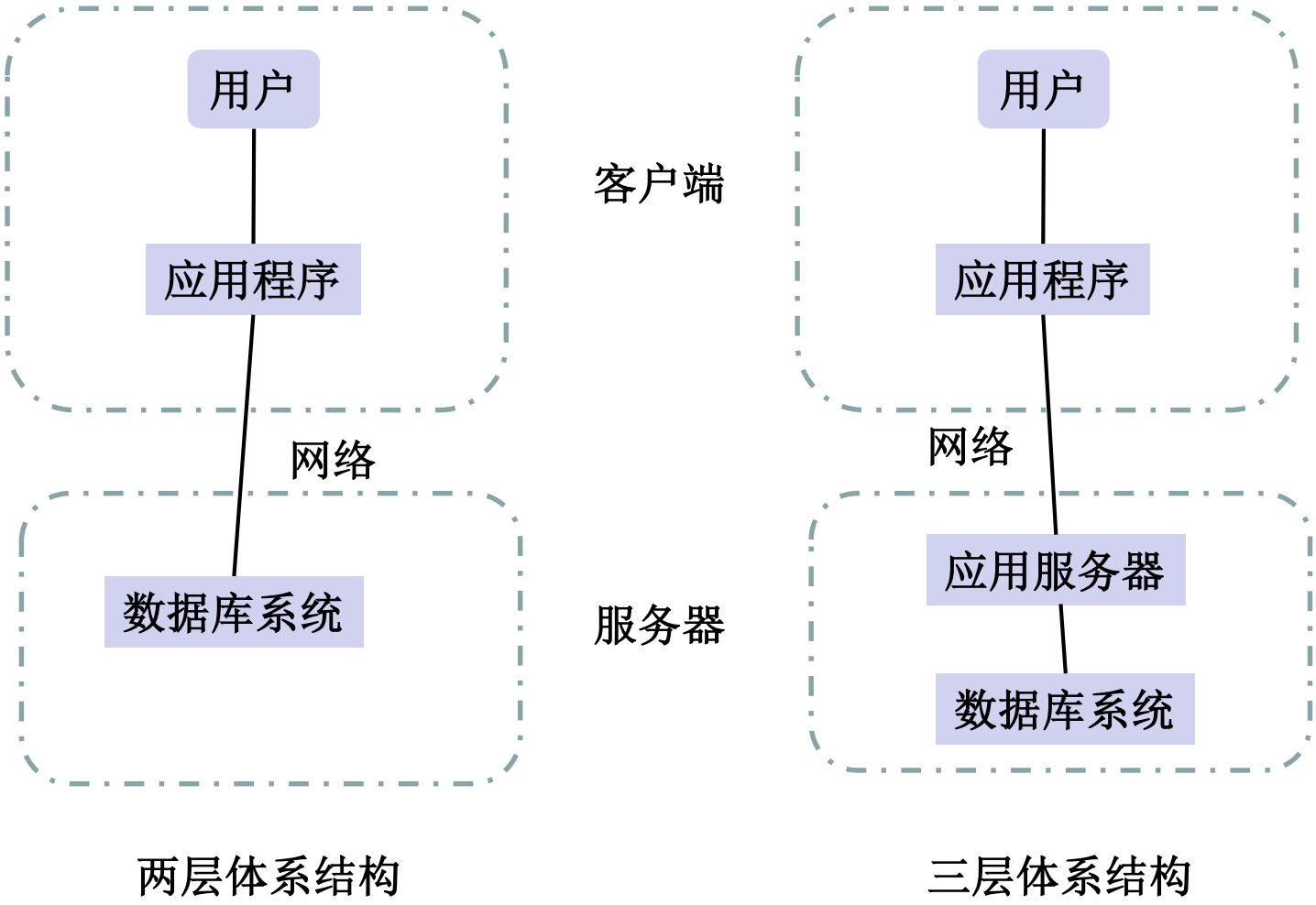
涉及到计算机学科的多个领域：

操作系统、语言与编译、软件工程、计算机网络等

DBMS的主要特点

- 数据独立性(Data independence)
- 高效数据访问(Efficient data access)
- 数据完整性与安全性(Data integrity and security)
- 数据管理(Data administration)
  - 系统易用性
- 并发控制机制(Concurrency control)
  - 权限管理、视图机制、多用户共享
- 健壮(Robustness)
  - 死锁、故障恢复

数据库系统结构



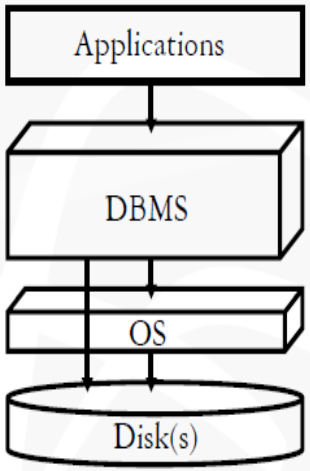
两层体系结构

三层体系结构

Major DBMS today

- Oracle
- IBM DB2 (from System R, System R*, Starburst)
- Microsoft SQL Server
- NCR Teradata
- Sybase
- Informix (acquired by IBM)
- PostgreSQL (from UC Berkeley's Ingres, Postgres)
- Tandem NonStop (acquired by Compaq, now HP)
- ? MySQL and Microsoft Access

Modern DBMS architecture



- OS layer is bypassed for performance and safety
- Many details will be filled in the DBMS box

Outline

- Why数据库系统？
- 相关概念
- 数据独立性
- 数据模型与数据库模式
- 数据库系统的发展

数据库系统的发展

- 第一代数据库系统 (1960s)
  - 层次和网状数据库系统
- 第二代数据库系统 (1970s)
  - 关系数据库系统

    1970年，E.F.Codd提出关系数据模型和理论，获图灵奖

    IBM 开始研发System R、UC Berkeley 研发Ingres
- 第三代数据库系统 (1980s)
  - 关系数据库系统占据统治地位
  - 面向对象数据模型，面向对象数据库系统
  - 并行与分布式数据库系统

数据库系统的发展

- Internet时代的数据库技术
  - 数据仓库、OLAP分析、数据挖掘
  - 数字图书馆、电子商务、Web医院、远程教育
  - 面向Ad Hoc和Moblie的移动数据库
  - Web上的数据管理与信息检索
  - 数据流管理
- 大数据时代的数据管理技术
  - 非关系的大规模并行分布式数据管理系统
  - NoSQL
  - NewSQL

数据库系统的发展

- 关系数据库out了么？为什么学习关系DBMS原理？
  - 关系模型出现之初只是一套理论，当时很多人怀疑它是否可行，是否能够被高效实现
  - 1980s关系数据库占据数据库系统主导地位，且持续了30-40年，在计算机史上可称为一段传奇
  - 关系数据库的核心在于结构化数据处理，近40年期间出现很多竞争技术
    - 对象数据库、XML数据库等曾聒噪一时，无一持续
    - 关系模型不断扩展，以适应多种应用
      - 例如，当前在网络大部分应用（在线发布、论坛、社交网络、电子商务、游戏、SaaS等）仍然由关系数据库所支撑
      - 关系模型简单易用
  - 在可预见的未来，关系数据库可能仍将继续与各种非关系数据存储一起使用

- **DB-Engines 2020年8月数据库排名**

359 systems in ranking, August 2020

Rank			DBMS	Database Model	Score		
Aug 2020	Jul 2020	Aug 2019			Aug 2020	Jul 2020	Aug 2019
1.	1.	1.	Oracle	Relational, Multi-model	1355.16	+14.90	+15.68
2.	2.	2.	MySQL	Relational, Multi-model	1261.57	-6.93	+7.89
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	1075.87	+16.15	-17.30
4.	4.	4.	PostgreSQL	Relational, Multi-model	536.77	+9.76	+55.43
5.	5.	5.	MongoDB	Document, Multi-model	443.56	+0.08	+38.99
6.	6.	6.	IBM Db2	Relational, Multi-model	162.45	-0.72	-10.50
7.	↑ 8.	↑ 8.	Redis	Key-value, Multi-model	152.87	+2.83	+8.79
8.	↓ 7.	↓ 7.	Elasticsearch	Search engine, Multi-model	152.32	+0.73	+3.23
9.	9.	↑ 11.	SQLite	Relational	126.82	-0.64	+4.10
10.	↑ 11.	↓ 9.	Microsoft Access	Relational	119.86	+3.32	-15.47

11.	↓ 10.	↓ 10.	Cassandra	Wide column	119.84	-1.25	-5.37
12.	12.	↑ 13.	MariaDB	Relational, Multi-model	90.92	-0.21	+5.96
13.	13.	↓ 12.	Splunk	Search engine	89.91	+1.64	+4.03
14.	↑ 15.	↑ 15.	Teradata	Relational, Multi-model	76.78	+0.81	+0.14
15.	↓ 14.	↓ 14.	Hive	Relational	75.29	-1.14	-6.51
16.	16.	↑ 18.	Amazon DynamoDB	Multi-model	64.75	+0.17	+8.18
17.	↑ 18.	↑ 25.	Microsoft Azure SQL Database	Relational, Multi-model	56.85	+4.22	+28.85
18.	↓ 17.	↑ 20.	SAP Adaptive Server	Relational	53.96	+0.09	-1.90
19.	↑ 20.	↑ 21.	SAP HANA	Relational, Multi-model	53.12	+1.78	-2.31
20.	↓ 19.	↓ 16.	Solr	Search engine	51.69	+0.05	-7.43
21.	↑ 22.	↑ 22.	Neo4j	Graph	50.18	+1.26	+1.79
22.	↑ 23.	↓ 19.	HBase	Wide column	49.11	+0.45	-7.42
23.	↓ 21.	↓ 17.	FileMaker	Relational	48.04	-1.41	-9.98
24.	↑ 25.	↑ 28.	Google BigQuery	Relational	32.60	+2.95	+8.13
25.	↓ 24.	↓ 24.	Microsoft Azure Cosmos DB	Multi-model	30.73	+0.32	+0.79
26.	26.	↓ 23.	Couchbase	Document, Multi-model	29.06	+0.36	-4.77
27.	27.	↓ 26.	Memcached	Key-value	25.96	+0.12	-1.05
28.	28.	↓ 27.	Informix	Relational, Multi-model	24.37	-0.27	-1.30
29.	↑ 30.	↑ 34.	InfluxDB	Time Series	22.88	+1.01	+4.53
30.	↓ 29.	30.	Amazon Redshift	Relational	22.37	-0.04	-0.24

数据库系统的发展

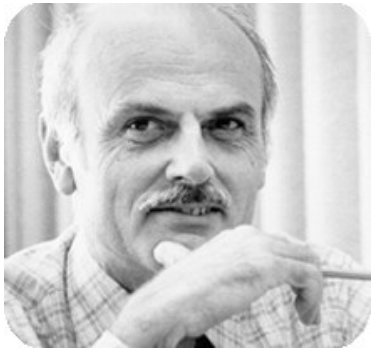
- 数据库发展至今50多年的历史，产生了4位图灵奖得主

Charles W. Bachman

网状数据库之父

1964年在通用电气研制了IDS，世界上第一个网状数据库系统

1973年获得图灵奖，表彰他在数据库领域，尤其是网状数据库管理系统方面的杰出贡献



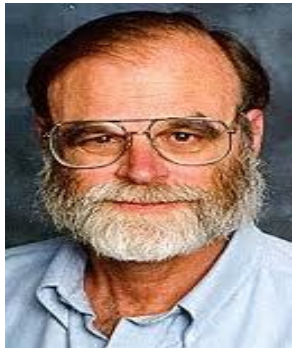
Edgar F. Codd

关系数据库之父

1970年提出关系模型概念

在IBM领导开发System R，IBM以System R为基础推出DB2

1981年获得图灵奖



James Gray

数据库和事务处理研究领域具有开创性的贡献，以及在系统实现方面具有领导地位

1998年获得图灵奖

M. R. Stonebraker

早期在UC Berkly领导研制Ingres、Postgres

1992年提出对象关系数据库模型

是众多数据库公司的创始人之一，其中包括Ingres、StreamBase Systems和Vertica等

2014年获得图灵奖

Now let's go to

Next Chapter