



数据库系统原理

教师：江胜

jwt@hust.edu.cn



关于本课程

- 线上线下结合
 - <https://www.icourse163.org/course/HUST-1449788170?tid=1468242475>
- 两次课堂测验（15%）、平时作业等（15%，含）、考试（70%）
- 思政与课程内容的融合
- 工程认证的重要性
- CMU 15-445/645课程
 - Database Systems (<https://15445.courses.cs.cmu.edu/fall2022/>)
- 《数据库系统原理实践》
 - 头歌平台
- 与华为的合作
 - 华为在线课程系列<https://edu.huaweicloud.com/roadmap/colleges.html>
 - 华为在线课程数据库系列，通过技术领域选择“鲲鹏”后，查找数据库或者openGauss的课程进入，或者直接搜索数据库相关课程



关于本课程

- 本课堂腾讯QQ群



群名称：2022秋季计算机2010-2011
群 号：904112043

江胜的个人会议室

869 166 5037



请使用手机端「腾讯会议 App」扫码入会

— 腾讯会议 —



教材

- 本课程主讲教材

- 萨师煊, 王珊著. 数据库系统概念(第五版). 高等教育出版社

- 开发环境

- Mysql、SQL Server ...;
Java/Python/C#/Delphi/C++/Powerbuilder, etc.

教材



- 推荐教材

- 冯玉才, 《数据库系统基础》, 华中理工大学出版社
- Database System Concepts, Abraham Silberschatz Professor / Henry F Korth / S Sudarshan, McGraw-Hill Education
《数据库系统概念》, 机械工业出版社
- A First Course in Database System Jeffrey D.Ullman, Jennifer Widom
《数据库系统基础教程》 清华大学出版社
- Database System Implementation Hector Garcia-Molina, D.Ullman, Jennifer Widom
《数据库系统实现》 机械工业出版社
- Transaction Processing Concepts and Techniques Jim Gray
《事务处理 – 概念与技术》 机械工业出版社



课程目标

- 结合关系型数据库系统深入理解数据库系统的基本概念、处理引擎、原理和方法。
- 掌握关系数据模型及关系数据语言，能熟练应用SQL语言表达各种数据定义、操作及控制等。
- 掌握实体联系（E-R）模型的概念和方法，掌握关系数据库规范化理论和关系数据库设计方法。通过上机训练及相关实践课程，初步具备进行数据库应用系统设计和开发的能力。
- 较深入掌握数据库恢复、并发控制、数据库安全性控制、数据库完整性控制、数据库引擎、查询优化等技术。
- 对数据、数据库领域研究的深入课题有大致了解，激发在此领域中继续学习和研究的愿望，为学习数据库系统高级课程做准备。



学习方法

- 理论联系实际，勤于动手
- 学会举一反三
- 系统思考
- 团队精神，合作，讨论



一个简单的数据应用实例

商品编号	商品名	进库价格	出库价格	生产厂家
DQ001	电视机	1600	1900	长虹
DQ002	冰箱	1300	1500	海尔
DQ003	洗衣机	1000	1500	海尔
DQ004	空调	2200	2500	春兰

商场库存货物数据

- 货物进库（增加数据）
- 货物出库（删除数据）
- 调整货物出库价格（修改数据）
- 查找生产厂家为‘海尔’的货物信息（查询数据）



课程特点

内容较多，部分章节难以理解，理论联系实际，学以致用。

第一章的抽象方法、计算机方法论；

第二章的代数运算、谓词演算；

第三章的一种英文文法的掌握；

第四章的计算机安全的基本知识；

第五章完整性语义和机制；

第六章的函数依赖、集合论闭包的概念；

课程特点（续）

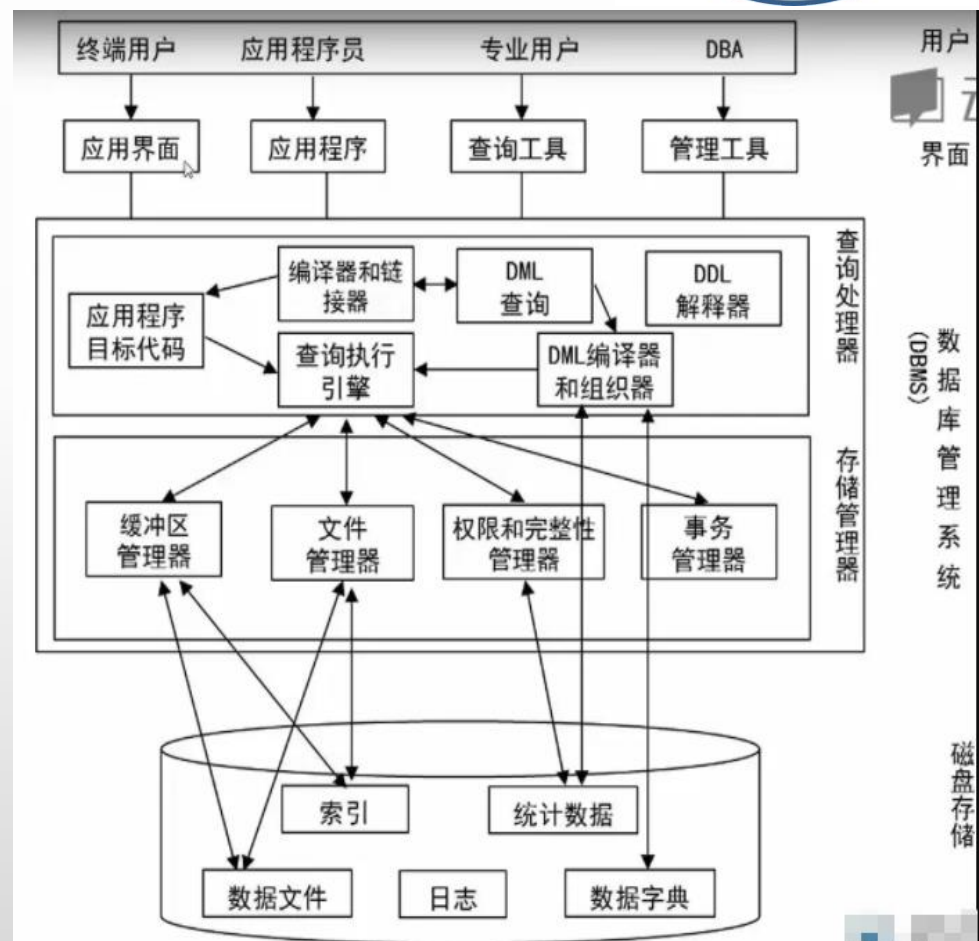
第七章涉及到软件工程的一些基本方法和经验；

第八章的“数据库引擎”数据存储、索引、缓存、哈希、查询处理等机制；

第九章的逻辑优化和物理优化；

第十章中事务之基本概念，保障系统可靠性的恢复策略；

第十一章关于并发不当产生的数据不一致问题，锁机制及其协议。





课程内容及课时分解

- 第一章 绪论 (7) (6-7)
- 第二章 关系数据库 (4) (2-3)
- 第三章 关系数据库标准语言SQL (8) (4-5)
- 第四章 数据库安全性 (2) (1)
- 第五章 数据库完整性 (2) (1)
- 第六章 关系数据理论 (8) (4-5)
- 第七章 关系数据库设计 (4) (3-4)
- 第八章 关系数据库引擎 (0) (7-8)
- 第九章 关系查询处理和查询优化 (3) (4-5)
- 第十章 关系数据库恢复 (3) (3-4)
- 第十一章 关系数据库并发控制 (3) (3-4)
- 课堂测验 (2)



第一章 绪论



1.1 数据库系统概述

1.2 数据模型

1.3 数据库系统结构

1.4 数据库系统的组成

1.5 数据库访问过程

1.6 数据库系统特点

1.7 小结



1.1 数据库系统概述

1.1.1 数据、数据库、数据库管理系统、数据库系统

1.1.2 数据管理技术的产生和发展 — 数据管理的三个阶段



1.1.1 数据、数据库、DBMS、DBS

■ 数据 (Data)

- 对现实世界中客观事物的符号表示
- 可以是数值数据，也可以是非数值数据，如声音、图像、结构化的记录等
- 计算机中数据
 - 能输入计算机，并能为其处理的符号序列
- 数据与其语义不可分

(0005794, 601, 周 济, 1, 1946.08.26, 01)

(工号, 部门编号, 姓名, 性别, 出生日期, 民族)

行政办公室

男

汉族



1.1.1 数据、数据库、DBMS、DBS(续)

■ 数据库 (Database)

长期储存在计算机内的、有组织的、可共享的数据集合

- “存放数据的仓库”
- 存储在计算机的存储设备上
- 按一定的格式组织、描述和存储
- 较小的冗余度
- 具有一定结构
- 高的数据独立性
- 可共享
- 易扩展
- 安全性、完整性



1.1.1 数据、数据库、DBMS、DBS(续)

■ 数据库管理系统(DBMS)

- 系统软件，数据库系统的一个重要组成部分
- 科学地组织和存储数据，高效地获取和维护数据
- 位于用户与操作系统之间
- 具有下述功能：
 - 数据定义功能 – DDL (如Create)
 - 数据操纵功能 – DML(如Select,Delete,Insert,Update)
 - 数据控制功能 – DCL(如Grant,Revoke)
 - 数据库的建立和维护功能
 - 数据库的运行管理



1.1.1 数据、数据库、DBMS、DBS(续)

■ 数据库系统(DBS)

- 计算机系统引入数据库后的系统
- 操作系统、数据库管理系统DBMS(及开发工具)、数据库、应用系统、数据库管理员(DBA)、用户

1.1.1 数据、数据库、DBMS、DBS(续)

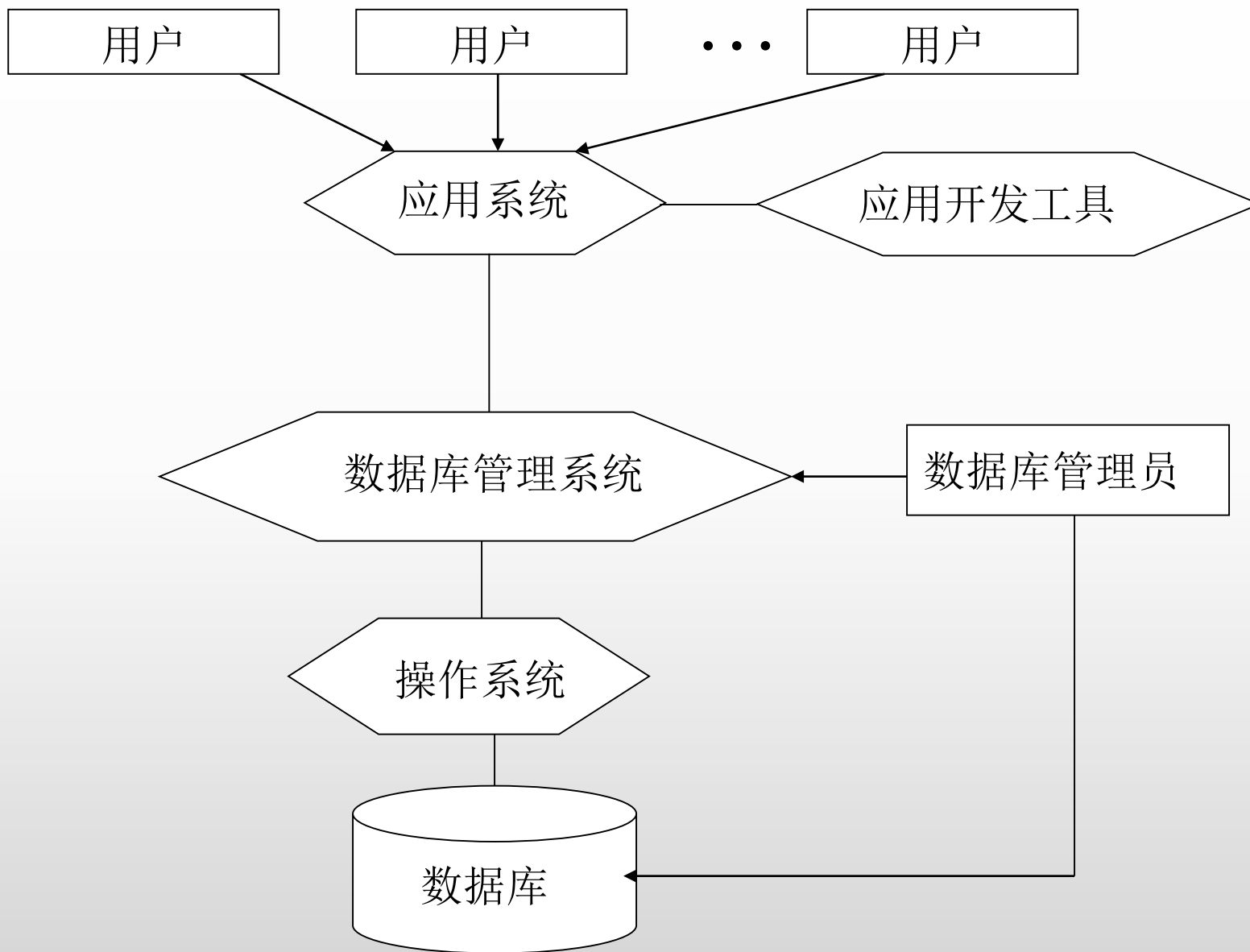


图1—1 数据库系统体系结构图



1.1.2 数据管理技术的产生和发展

- 数据库技术是应数据管理任务的需要而产生的
- 数据管理：数据分类、组织、编码、存储、检索和维护
- 数据管理技术经历了三个阶段
 - 人工管理阶段(50年代中期以前)
 - 文件系统阶段(50年代后期-60年代中期)
 - 数据库系统阶段(60年代后期开始)



1.1.2 数据管理技术的产生和发展(续)

一人工管理阶段

■ 背景

- 计算机主要用于科学计算
 - 数据量小、结构简单，如高阶方程、曲线拟和等
- 外存为顺序存取设备
 - 磁带、卡片、纸带，没有磁盘等直接存取设备
- 数据处理方式：批处理
- 没有操作系统，没有数据管理软件
 - 硬件资源都需要由用户自己管理。用户用机器指令编码，通过纸带机输入程序和数据，程序运行完毕后，由用户取走纸带和运算结果，再让下一用户上机操作



1.1.2 数据管理技术的产生和发展(续)

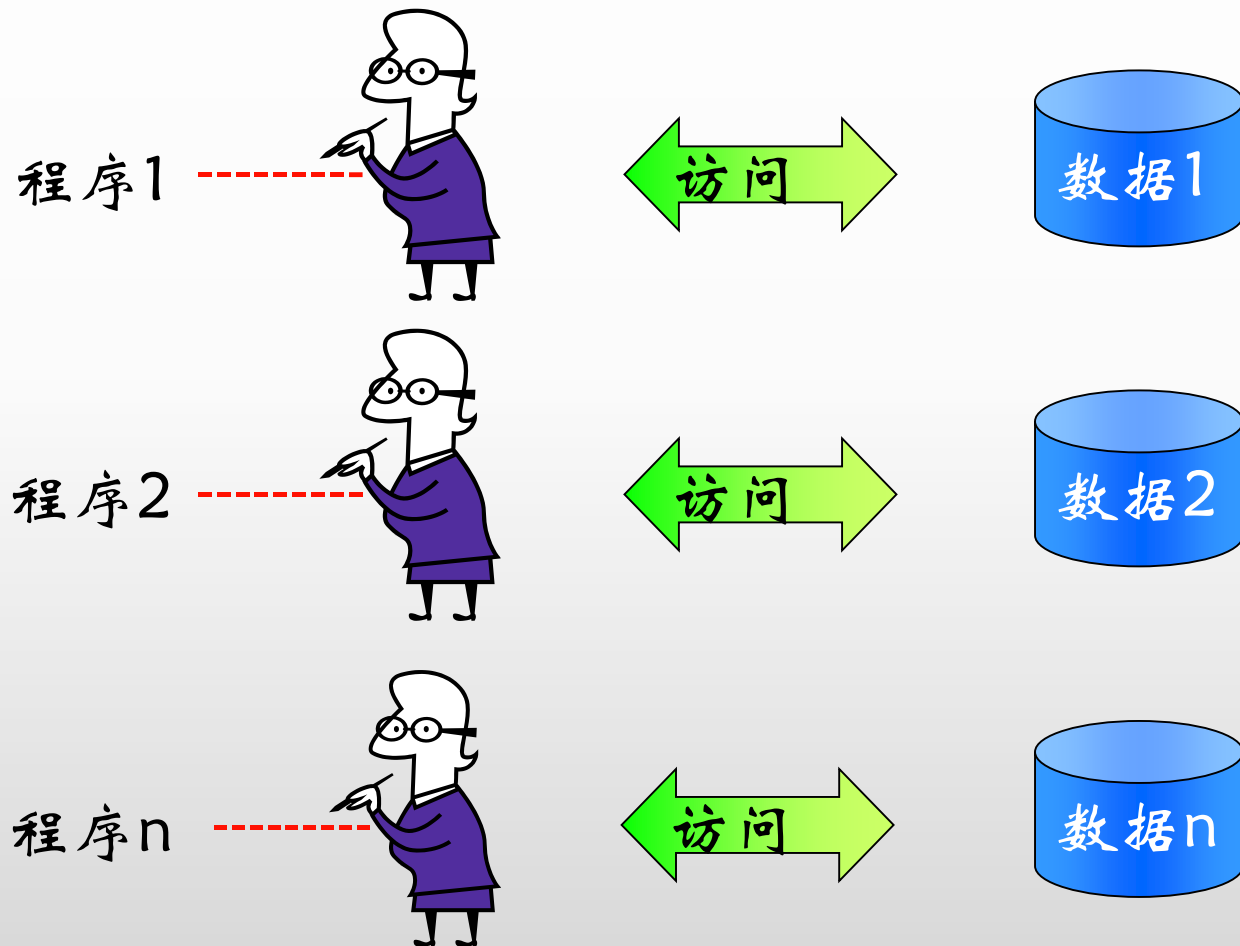
一人工管理阶段

■ 特点

- 应用程序完全负责数据管理工作
 - 数据的组织、存储结构、存取方法、输入输出等
- 数据完全面向特定的应用程序（无共享性）
 - 不同程序之间的数据具有巨大的冗余
- 数据与程序没有独立性（无独立性）
 - 程序中存取数据的子程序随着存储结构的改变而改变
- 数据不保存
 - 每个用户使用自己的数据，数据不保存，用完就撤走

1.1.2 数据管理技术的产生和发展(续)

—人工管理阶段





1.1.2 数据管理技术的产生和发展(续)

一文件系统阶段

■ 背景

- 计算机不但用于科学计算，还用于管理
- 外存有了磁盘、磁鼓等直接存取设备
 - 直接存取设备（DASD）
 - 无须顺序存取
 - 由地址直接访问所需记录
- 有了专门管理数据的软件，一般称为文件系统
 - 文件存储空间的管理
 - 目录管理
 - 文件读写管理
 - 文件保护
 - 向用户提供操作接口



1.1.2 数据管理技术的产生和发展(续)

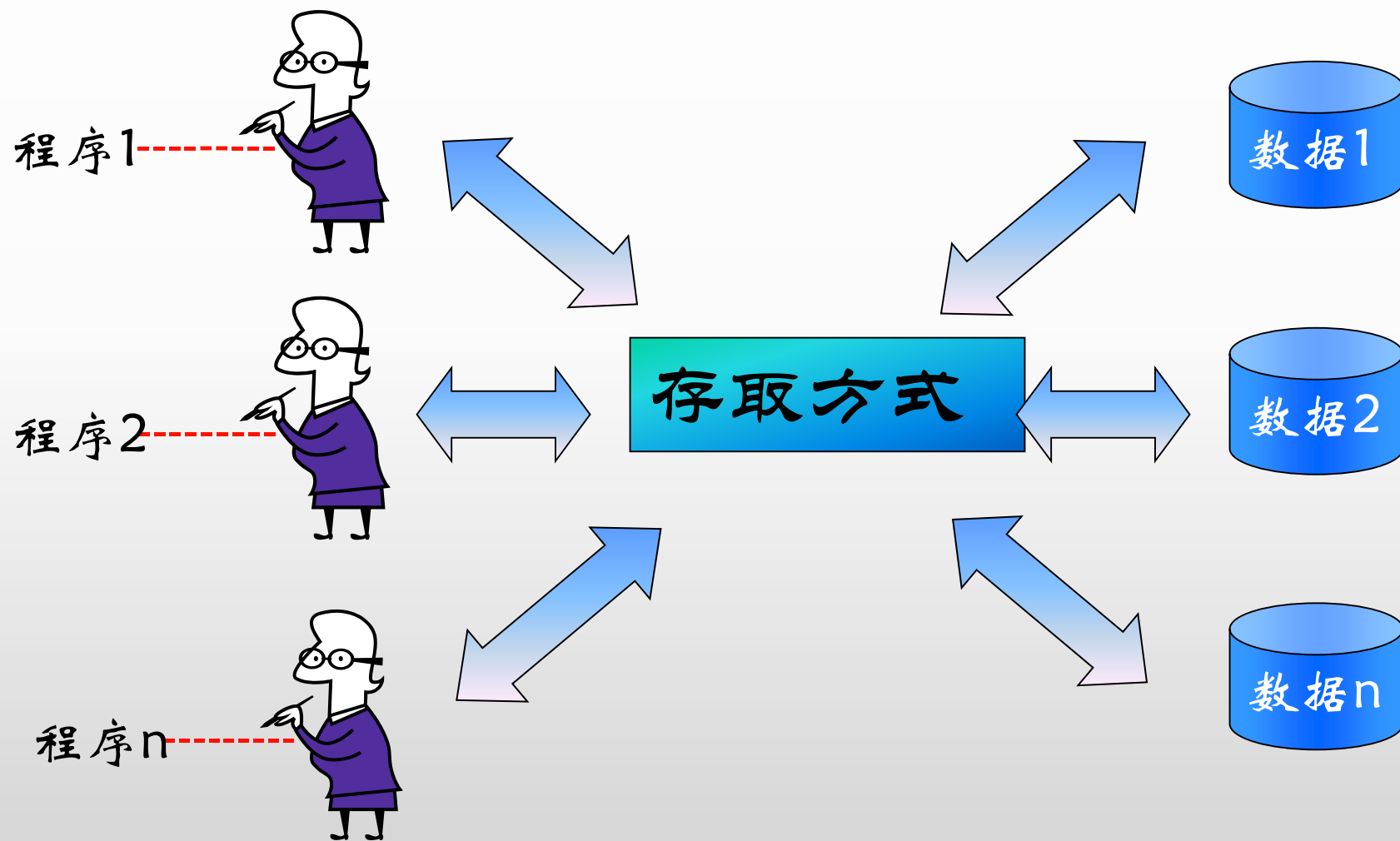
一文件系统阶段

■ 特点

- 系统提供一定的数据管理功能
 - 存取方法（按文件名访问，按记录进行存取）
 - 支持对文件的基本操作（增、删、改、查等），用户程序不必考虑物理细节
 - 数据的存取基本上以记录为单位
- 数据仍是面向应用的
 - 一个数据文件对应一个用户程序
- 数据与程序有一定的独立性
 - 文件的逻辑结构与存储结构由系统进行转换，数据在存储上的改变不一定反映在程序上，但是扩展新的应用时仍然很困难

1.1.2 数据管理技术的产生和发展(续)

一文件系统阶段





1.1.2 数据管理技术的产生和发展(续)

—文件系统阶段

■ 数据与程序的独立性差

- 文件系统的出现并没有从根本上改变数据与程序紧密结合的状况，数据的逻辑结构改变则必须修改应用程序
- 文件系统只是解脱了程序员对物理设备存取负担，它并不负责数据语义的解释，只负责存储
- 数据的语义信息只能由程序来解释，也就是说，数据收集以后怎么组织，以及数据取出来之后按什么含义应用，只有全权管理它的程序知道。
- 一个应用若想共享另一个应用生成的数据，必须同另一个应用沟通，了解数据的语义与组织方式



文件系统处理数据的方式（举例）

例程1.1

```
#include "stdio.h"

main()
{
    FILE *fp;
    fp=fopen("file1.c","w");
    fputs("Chenwei",fp);
    putw(20000101,fp);
    putw(20,fp);
    fputs("Linzi",fp);
    putw(20000102,fp);
    putw(21,fp);
    fclose(fp);
}
```

例程1.2

```
#include "stdio.h"
#define SIZE 2
Struct student_type
{char name[8];
  int num;
  int age;}stud[SIZE]
main()
{int i;
  FILE *fp;
  fp=fopen("file1.c","r");
  for(i=0;i<SIZE;i++)
  {fread(&stud[i],sizeof(struct student_type),1,fp);
   printf("%8s%d%d\n",stud[i].name,
           stud[i].num,stud[i].age);}
}
```



1.1.2 数据管理技术的产生和发展(续)

—关于数据独立性

■ 为什么需要“数据独立性”

- 软件系统日益复杂、庞大。“状态空间”爆炸
- 软件需要人来维护。需求、设计错误；BUG；需求变动；软件扩展。。。
- 软件危机
- 软件全生命周期的成本控制

■ 怎么做到“数据独立性”

- 软件分层。从“架构”的层面，“解耦”
- 模块化。高内聚、低耦合
- 数据的表示、定义、存储、处理与“业务”分开



文件系统处理数据的方式

- 数据和程序分为不同的文件
- 数据文件由文件系统组织和存储
- 文件系统提供数据存取方法的接口
- 数据的逻辑结构与程序紧密相关



1.1.2 数据管理技术的产生和发展(续)

一文件系统阶段

- 数据的共享性差，冗余度大
 - 数据面向应用
 - 即使不同应用程序所需要的数据有部分相同时，也必须建立各自的文件，而不能共享相同的数据
 - 数据孤立
 - 数据分散管理，许多文件，许多数据格式
- 数据的不一致性
 - 由于数据存在很多副本，给数据的修改与维护带来了困难，容易造成数据的不一致性



1.1.2 数据管理技术的产生和发展(续)

一文件系统阶段



劳资科

学号	姓名	系别	补贴
----	----	----	----



房产科

学号	姓名	性别	系别	住址
----	----	----	----	----



学籍科

学号	姓名	系别	学分	学位
----	----	----	----	----



人事科

学号	姓名	性别	系别	年龄	学位	出身
----	----	----	----	----	----	----



1.1.2 数据管理技术的产生和发展(续)

一数据库系统阶段

■ 背景

- 计算机管理的数据量大，描述的对象众多，关系复杂，共享性要求强（多种应用、不同语言共享数据）
- 外存有了大容量磁盘，光盘
- 数据库管理系统的出现
- 软件价格上升，硬件价格下降，编制和维护软件及应用程序成本相对增加，其中维护的成本更高，力求降低
- 数据处理需要考虑分布处理



1.1.2 数据管理技术的产生和发展(续)

—数据库系统阶段

■ 特点

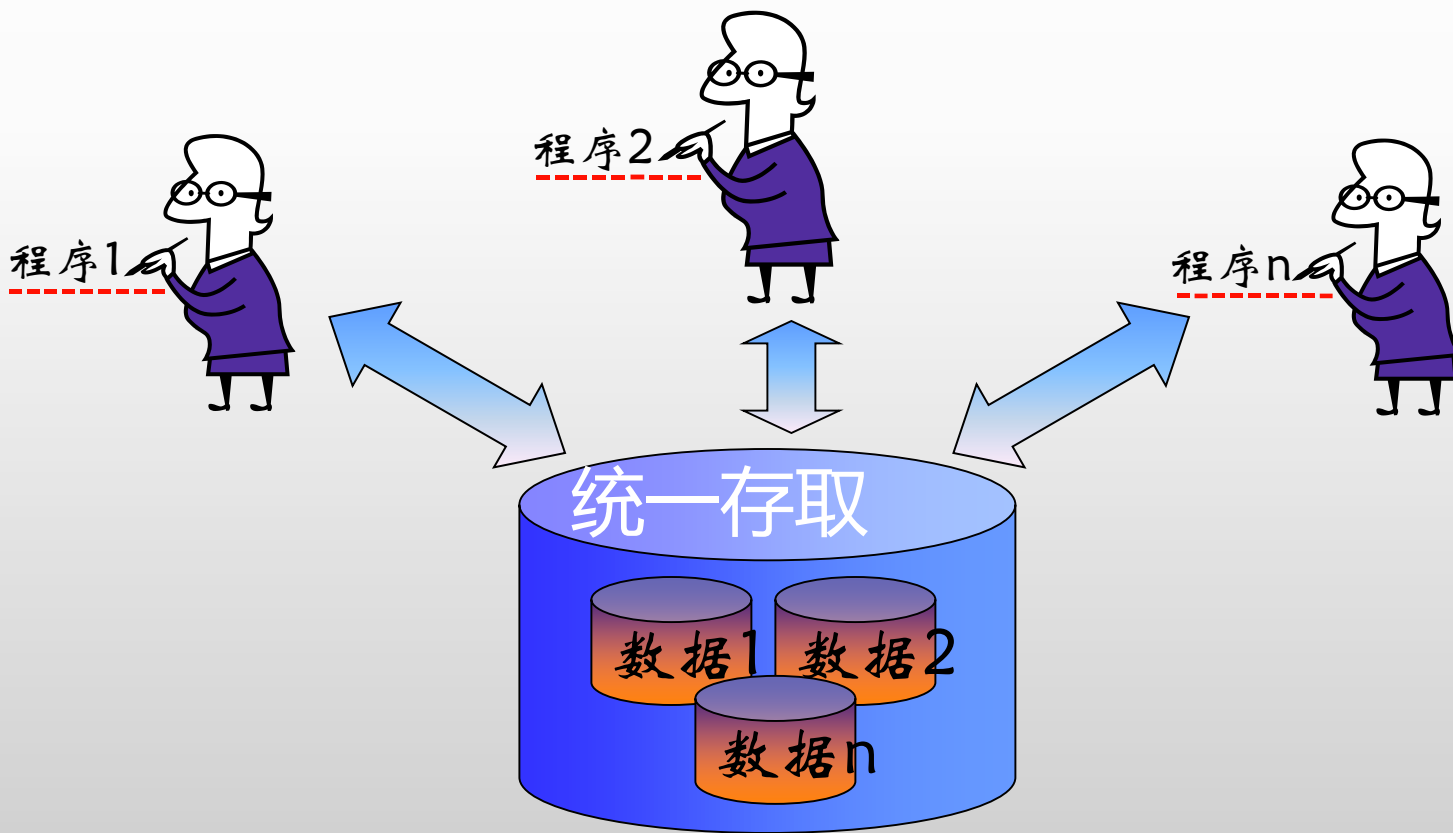
- 数据结构化
- 数据的共享性高，冗余度低，易扩充
- 数据独立性高
- 数据由**DBMS**统一管理和控制

1.1.2 数据管理技术的产生和发展(续)

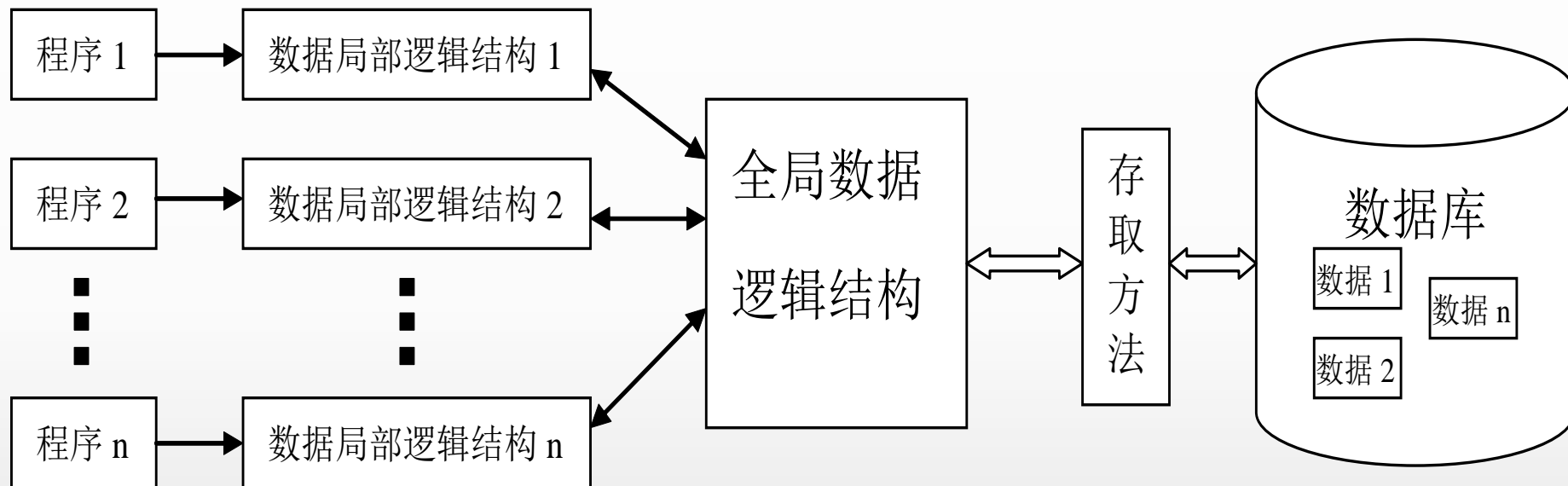
—数据库系统阶段

■ 数据库观点

- 以数据库为中心，使得数据既便于集中管理，也便于应用程序的研制和维护。



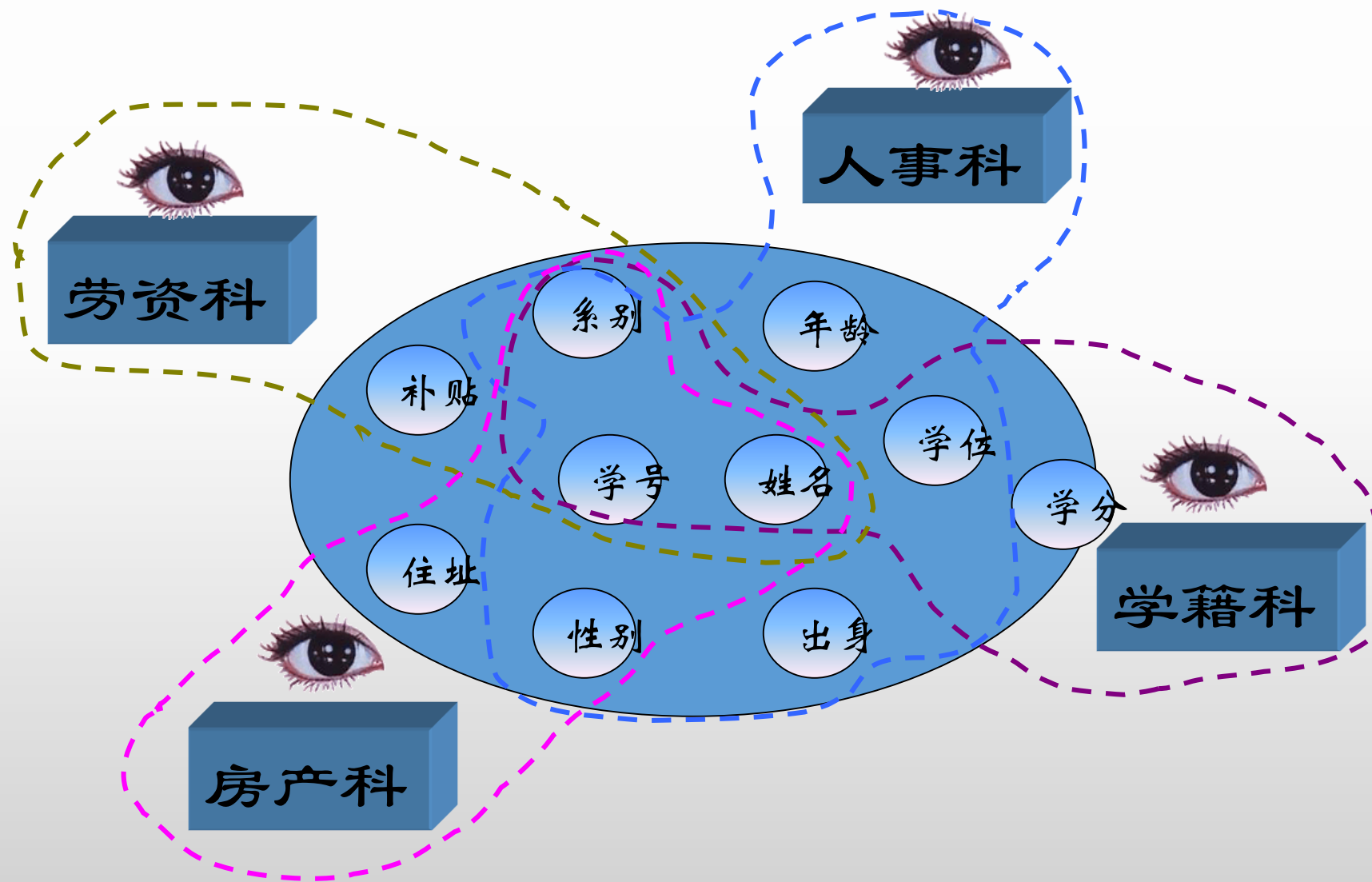
数据库系统阶段



数据库系统的数据应用方式

1.1.2 数据管理技术的产生和发展(续)

—数据库系统阶段





1.1.2 数据管理技术的产生和发展(续)

一数据库系统的特点

■ 数据结构化

- 数据结构化是数据库与文件系统的本质区别

学生人事记录

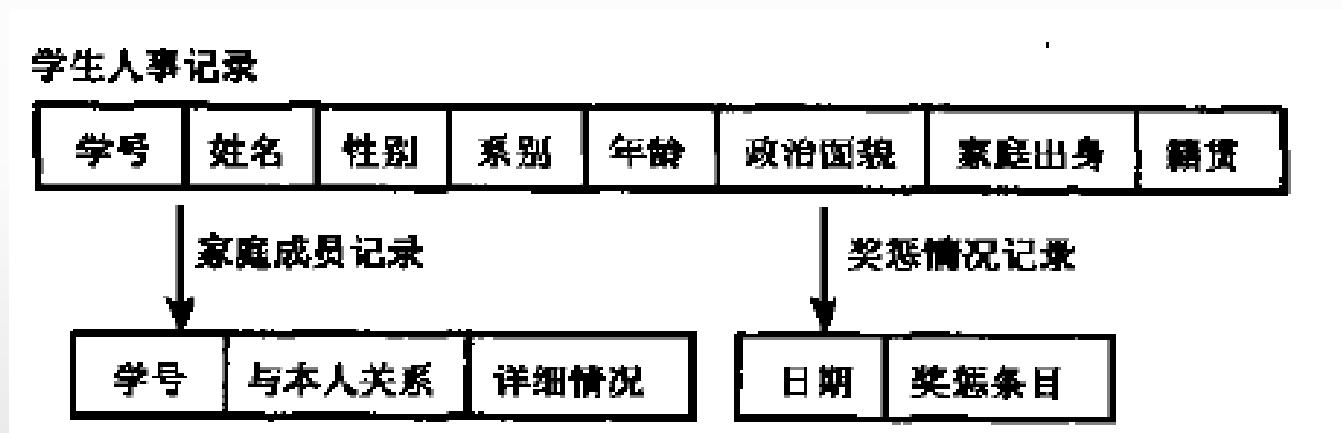
学号	姓名	性别	系别	年龄	政治面貌	家庭出身	籍贯	家庭成员	奖惩情况
----	----	----	----	----	------	------	----	------	------



1.1.2 数据管理技术的产生和发展(续)

一数据库系统的特点

- 数据结构化 – 续



主记录 – 详细记录格式示例



1.1.2 数据管理技术的产生和发展(续)

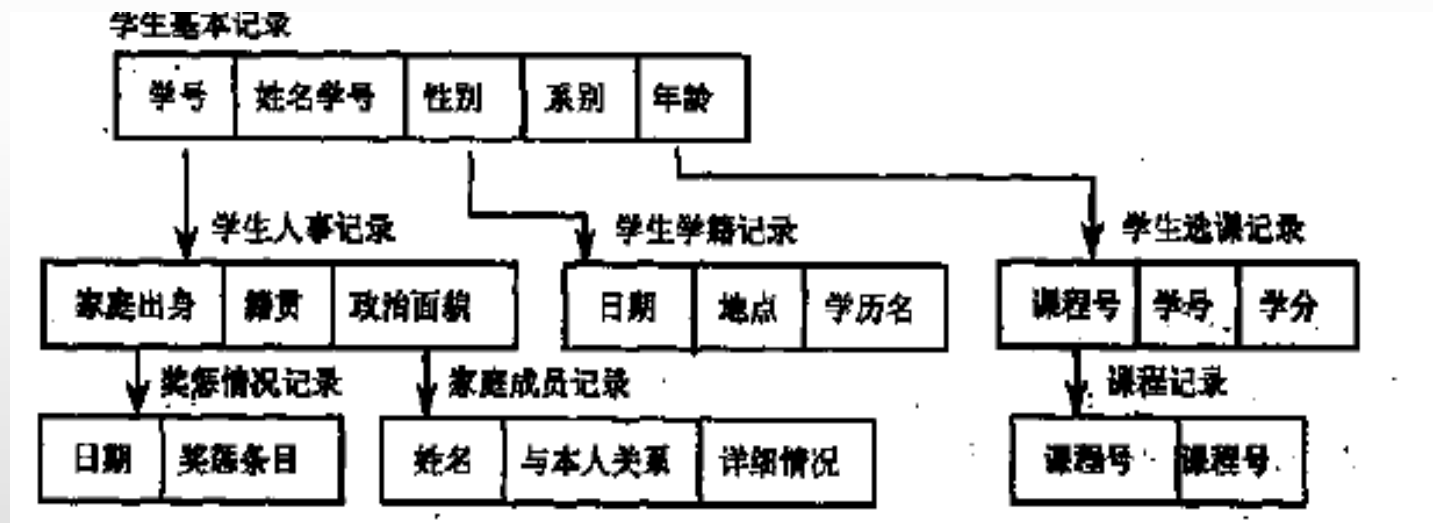
一数据库系统的特点

- 数据结构化 – 续
 - 数据反映了客观事物间的本质联系，而不是着眼于面向某个应用，是有结构的数据。这是数据库系统的主要特征之一，与文件系统的根本差别。文件系统只是记录的内部有结构，一个文件的记录之间是个线性序列，记录之间无联系

1.1.2 数据管理技术的产生和发展(续)

一数据库系统的特点

- 数据结构化 – 续
 - 支持全企业的应用而不是某一个应用



学生数据的组织



1.1.2 数据管理技术的产生和发展(续)

一数据库系统的特点

- 数据的冗余度小，易扩充
 - 数据面向整个系统，而不是面向某一应用，数据集中管理，数据共享，因此冗余度小
 - 节省存储空间，减少存取时间，且可避免数据之间的不一致性
 - 每个应用选用数据库的一个子集，只要重新选取不同子集或者加上一小部分数据，就可以满足新的应用要求，这就是易扩充性



1.1.2 数据管理技术的产生和发展(续)

一数据库系统的特点

- 具有较高的数据和程序的独立性
 - 在逻辑上，把数据库的定义和描述从应用程序中分离出去
 - 在物理上，数据的存取由系统管理，用户不必考虑存取路径等细节，从而简化了应用程序



1.1.2 数据管理技术的产生和发展(续)

—数据库系统的特点

- 统一的数据管理和控制功能，并发共享能力高
 - 数据的安全性控制（**Security**）
 - 保护数据以防止不合法的使用所造成的数据泄露和破坏
 - 措施：用户标识与鉴定，存取控制
 - 数据的完整性控制（**Integrity**）
 - 数据的正确性、有效性、相容性
 - 措施：完整性约束条件定义和检查
 - 并发控制（**Concurrency**）
 - 对多用户的并发操作加以控制、协调，防止其互相干扰而得到错误的结果并使数据库完整性遭到破坏
 - 措施：封锁



1.1.2 数据管理技术的产生和发展(续)

—数据库系统的特点

- 统一的数据控制功能，数据共享程度高（续）
 - 数据库恢复（Recovery）
 - 把数据库从错误状态恢复到某一已知的正确状态(亦称一致性状态或完整性状态)
 - 措施：转储，镜像，日志



第一章 绪论

1.1 数据库系统概述

➔ 1.2 数据模型

1.3 数据库系统结构

1.4 数据库系统的组成

1.5 数据库访问过程

1.6 数据库系统特点

1.7 小结

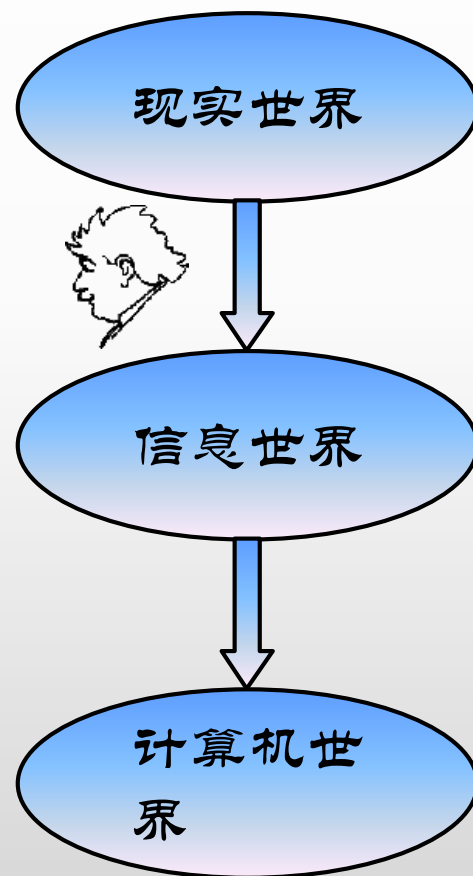
1.2 数据模型(data model)

1.2.1 数据模型简介

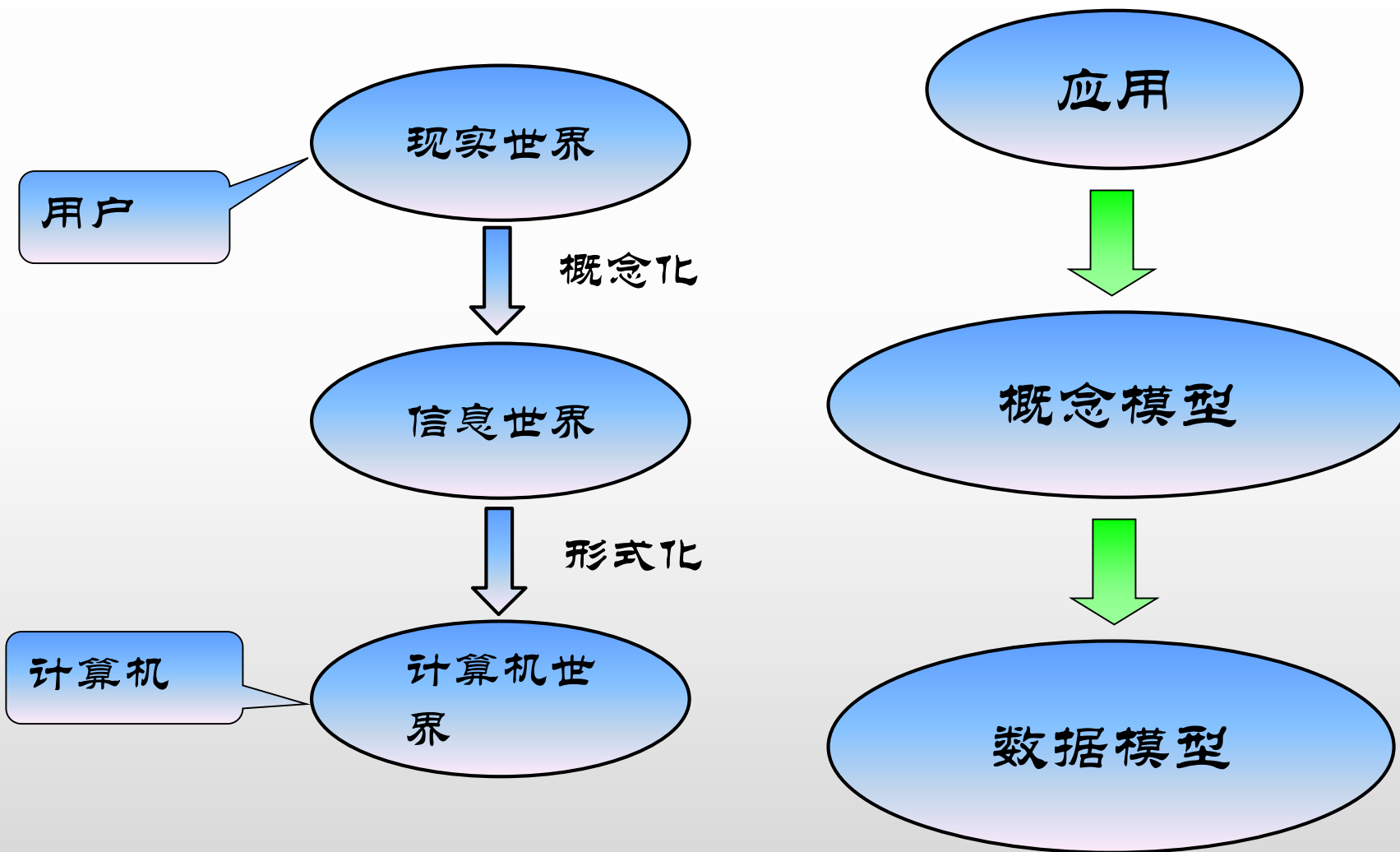
- 对现实世界特征的模拟和抽象
 - 真实地模拟现实世界
 - 易为人所理解
 - 便于计算机实现

数据模型

数据及数据间联系
的表示形式(现实世界的模拟)



1.2 数据模型





1.2 数据模型

■ 现实世界

- 存在于人们头脑之外的客观世界
- 例

- 仓库管理中涉及的货物管理，货物、货物的进出以及相应的报表、图表、表格、卡片、单据等

■ 信息世界

- 现实世界在人脑中的反映

■ 计算机世界

- 信息在计算机中以数据形式存储



1.2 数据模型

■ 概念模型

- 按用户的观点来对数据和信息建模
- 用于组织信息世界的概念，表现从现实世界中抽象出来的事物以及它们之间的联系
- 这类模型强调其语义表达能力，概念简单、清晰，易于用户理解
- 它是现实世界到信息世界的抽象，是用户与数据库设计人员之间进行交流的语言
- 如E-R(实体-联系模型)模型



1.2 数据模型

■ 数据模型

- 从计算机实现的观点来对数据建模
- 是信息世界中的概念和联系在计算机世界中的表示方法
- 一般有严格的形式化定义，以便于在计算机上实现
- 如层次模型、网状模型、**关系模型**等



1.2 数据模型

- 数据模型

数据模型是严格定义的一组概念。这些概念精确的描述了系统的静态特性、动态特性和完整性约束。



1.2 数据模型

■ 数据结构

- 描述系统的静态特性，即组成数据库的对象类型
- 数据本身
 - 类型、内容、性质。如：关系模型中的域、属性，关系等
- 数据之间的联系
 - 例如：关系模型中的外码



1.2 数据模型

■ 数据操作

- 描述系统的动态特性，即对数据库中对象的实例允许执行的操作的集合，包括操作及操作规则
- 一般有检索、更新（插入、删除、修改）操作
- 数据模型要定义操作含义、操作符号、操作规则，以及实现操作的语言

■ 数据约束

- 数据的约束条件是完整性规则的集合，规定数据库状态及状态变化所应满足的条件，以保证数据的正确、有效、相容(例如：工龄<年龄， $YL \leq 150$ 。)



1.2 数据模型

典型DM

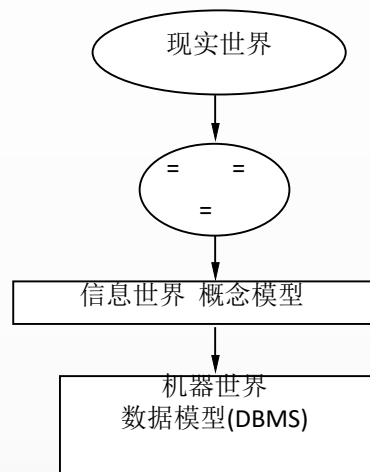
- ① 层次模型(Hierarchical Model)
- ② 网状模型(Network Model)
- ③ 关系模型(Relational Model)

1.2.2 概念模型

1. 定义

独立于特定DBMS的现实世界的抽象模型。(用户与DB设计人员进行交流的语言)。

1.2 数据模型



2. 特点

- 1) 较强语义表达能力;
- 2) 便于直接表示应用语义;
- 3) 简单、清晰, 易于理解。



1.2 数据模型

3. 信息世界概念

认识 → 表示 → 处理

1) 实体(entity)

——客观存在可相互区别的事物、事件和概念。(静态、动态、物质、精神、联系等)

2) 属性(Attribute)

——实体具有的特性。

Student (XH, XM, XB, YL)

3) 实体型 (entity type)

——具有相同特征和性质的实体其实体名和属性命名序列。



1.2 数据模型

Student

XH	XM	XB	YL
----	----	----	----

Course

KH	KM
----	----

S-C

XH	KH	CJ
----	----	----

4) 实体值 (entity value)

——实体型的具体实例。





1.2 数据模型

Student

XH	XM	XB	YL
2001	于得水	男	20
2002	牛得草	男	21
2003	白雪	女	21
2004	田野	男	18
.....

5) 实体集 (entity set)
——同型实体值的集合。





1.2 数据模型

6) 域 (domain)

——属性的取值范围。

例：YL为小于150的三位整数，XB为（男，女）

7) 码 (KEY)

——唯一标识一个实体集中任何实体值又不含多余属性的属性集。

Student (KEY): XH

SC (KEY) : (XH, KH)

- 至少一个属性
- 至多n个属性
- 不含多余属性

(XH, XM) KEY?

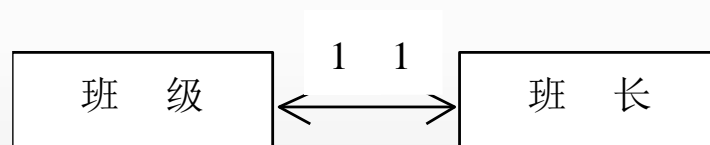


1.2 数据模型

8) 联系 (relationship)

① 1: 1 (一对一联系)

定义：设有实体集A、B，若其中任何一个实体集中每一实体值至多与另一实体集中的一个实体值有联系，则称A、B间存在一对一联系。



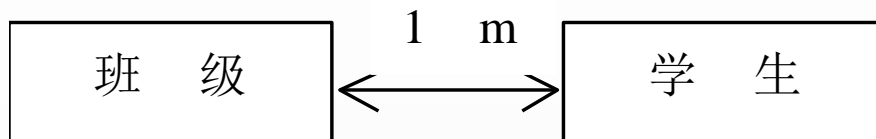
② 1: m

定义：设有实体集A、B，若A中的每一个实体值，与B中的n个实体值 ($n \geq 0$) 有联系，反之，对于B中的每一个实体值，至多与A中的一个实体值有联系，则称A、B间存在一对多联系。



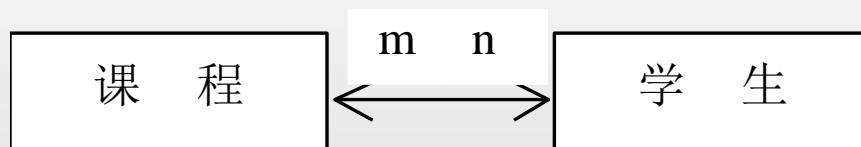


1.2 数据模型



③ m:n联系

定义：设有实体集A、B，若其中任何一个实体集中的每一个实体值均与另一个实体集中的n个实体值（ $n \geq 0$ ）有联系，则称A、B间存在多对多联系。

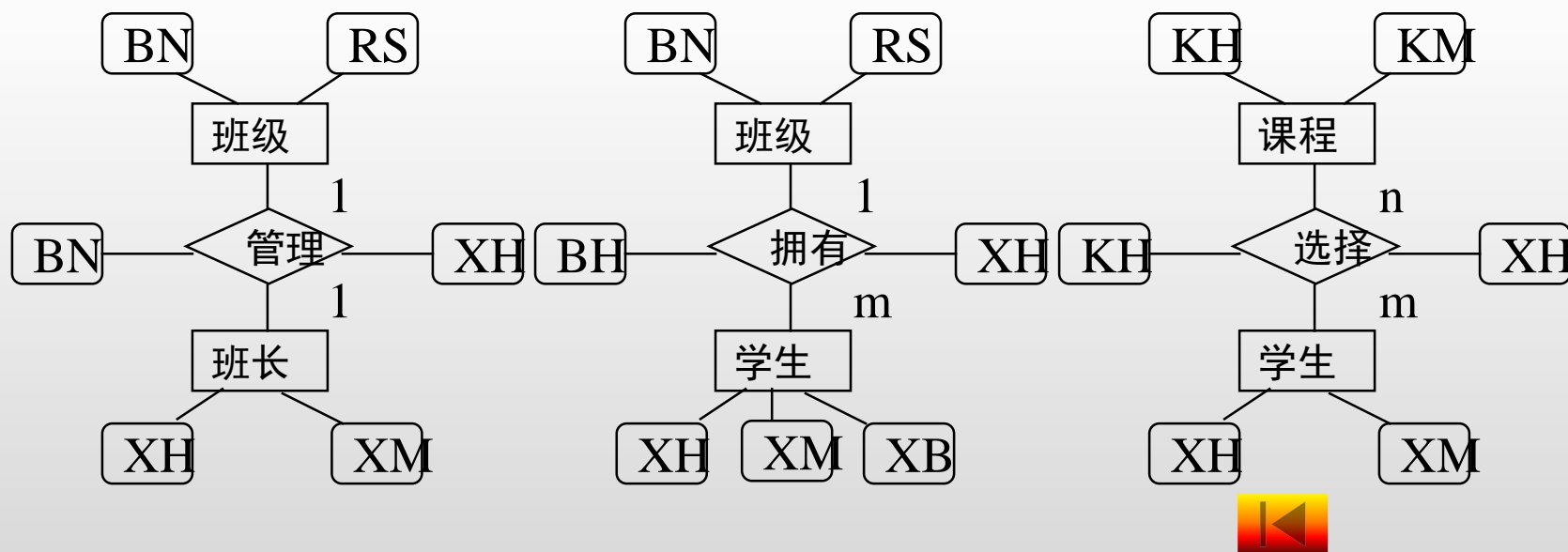


1.2 数据模型

4. 概念模型表法方法

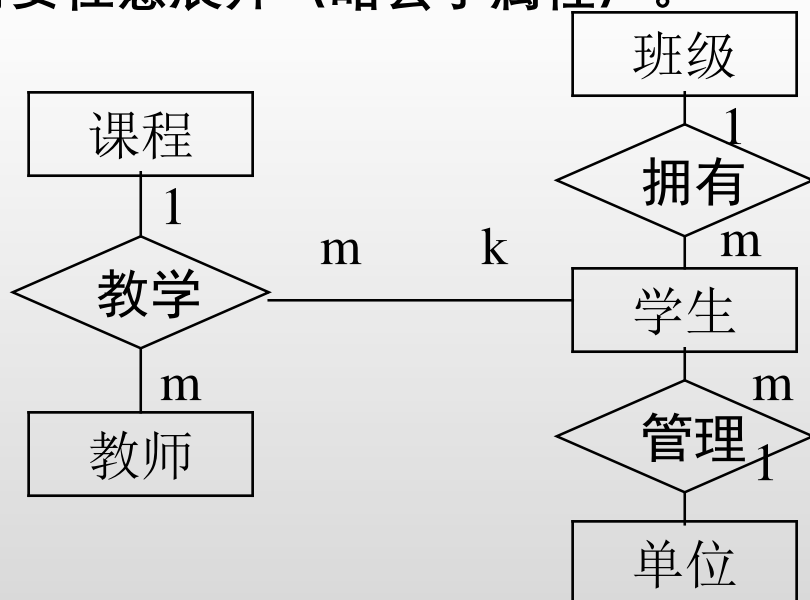
E-R方法 (Entity Relationship Approach)

1) 构成形式



1.2 数据模型

- (1) 矩形表示实体型，框内标明实体名；
- (2) 椭圆表示属性，用无向边与其相应实体连接；
- (3) 菱形表示联系，内标明联系名，用无向边与相关实体连接；
- (4) 无向边上标明联系的类型（1: 1, 1: m, m: n）；
- (5) 可据需要任意展开（略去了属性）。





1.2 数据模型

2) 特征

(1) 直接表示 $m:n$ 联系

(2) 与特定DBMS无关

- 更一般；
- 更抽象；
- 更接近现实。

(3) 易于向特定DBMS支持的DM转换





1.2 数据模型

1.2.3 层次DM

——用树形结构表示实体及实体间联系的DM。

1. 数据结构

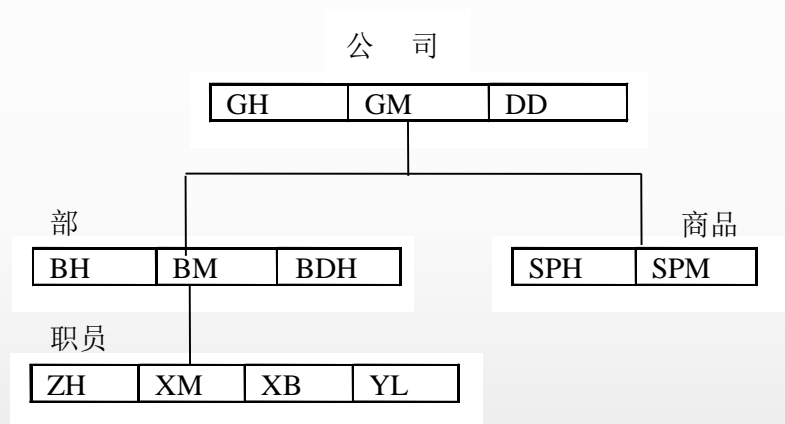


图1-4 层次模型简例

- 1) 一个结点表示一个实体（一个片段：fragment）；
- 2) 无向连线表示实体间联系；
- 3) 结点内含字段(field)，表示属性；



1.2 数据模型

4) 片段、字段须命名；

5) 特征

①有且仅有一个结点无双亲结点，称之为根结点（root）；

②余下子女结点有且仅有一个双亲结点。

2. 操作

I、D、U、Q .

3. 约束

1) 无双亲不能插入子女结点值；

2) 删去双亲结点值，则子女结点值同时删去。

4. 优点

1) 简单易用（几条操作命令）；

1.2 数据模型

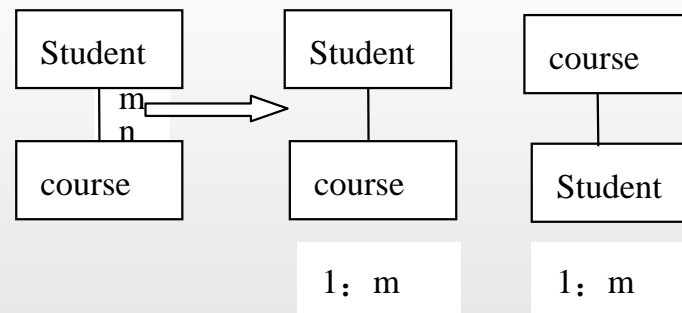
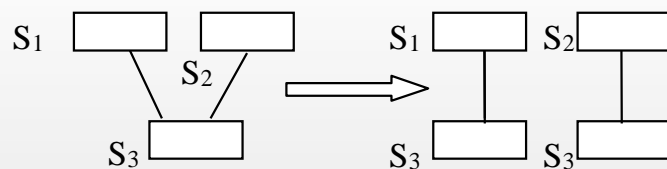
2) 自然表示1: M联系;

3) 速度较快。

5. 缺点

1) 不能直接表示m: n联系;

(须引进冗余结点)



2) 插入, 删除操作限制多;

3) 查子女须经过双亲。

(从上到下, 从左到右)

1.2 数据模型

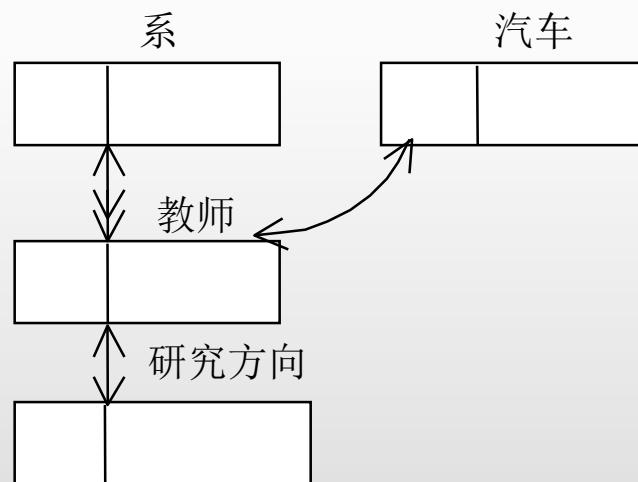
1.2.4 网状模型

——用网状结构表示实体及实体间联系的数据模型（DBTG：Database Task Group）。

CODASYL(Conference of Data System Language)

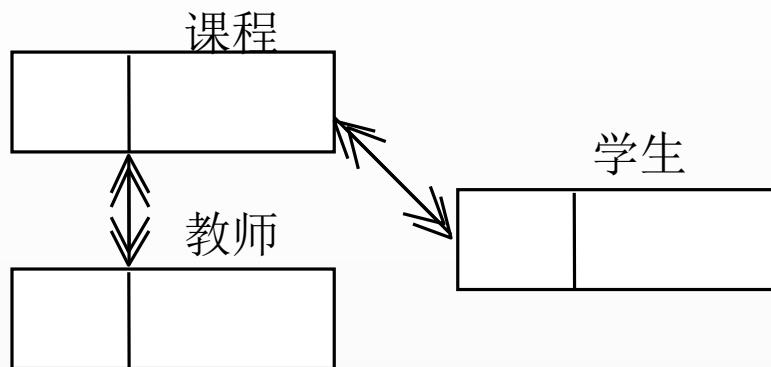
1. 数据结构

a: 简单网

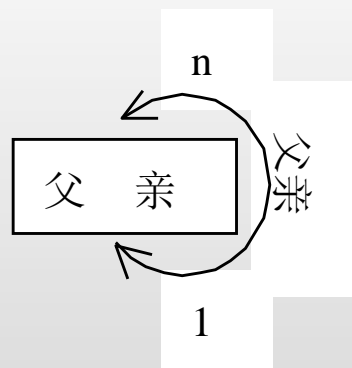


1.2 数据模型

b: 复杂网



c: 简单环网

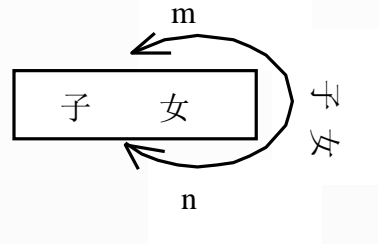


- 一个父亲有多个已为人父的儿子；
- 已为人父的儿子只有一个父亲。



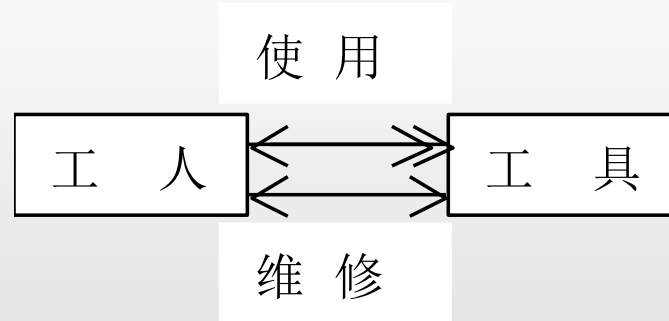
1.2 数据模型

d: 复杂环网



- 每个子女可多个子女;
- 每个为人子女者又可有多多个子女。

e: 多种联系



1.2 数据模型

- 1) 结点表示实体，称为记录类型；
- 2) 结点内含数据项，表示属性；
- 3) 有向连线表示实体间联系；(系类型)
- 4) 属性可嵌套。

XH	XH	CJ		
		CJ ₁	CJ ₂	CJ ₃
901	易难	59	73	66
904	高飞	81	81	81
.....

5) 特征

- (1) 可多个结点无双亲结点；
- (2) 子女结点可多个双亲结点；
- (3) 两记录间可多种联系。





1.2 数据模型

2. 操作

Q、I、D、U .

3. 约束

- 1) 插入不受限制；
- 2) 删去双亲，子女不受影响。





1.2 数据模型

4. 优点

- 1) 直接表示的m: n联系;
- 2) 存取效率高。

5. 缺点

- 1) 结构复杂;
- 2) DDL复杂;
- 3) 一次存取一个记录值;
- 4) 应用程序与数据结构相互依赖;
- 5) 过程化语言。





1.2 数据模型

1.2.5 关系模型(Relational Model)

——用二维表格表示实体及其间联系的DM。

1. 数据结构

XH	XM	XB	YL
2001	马千里	男	20
2002	草上飞	男	18
2005	凌燕	女	18
2003	高升	男	18
...	↑	↓	
...	列	域 (domain)	

← 属性名
← 元组 (4元元组)

:属性取值范围

1) 表格表示实体，内含属性；



1.2 数据模型

2) 表格表示实体间联系；

student (XH,XM) course(KH,KM) sc(XH,KH,CJ)

3) 行、列次序无关。

4) 每一个分量均不可再分。

XH	KH	CJ		
		CJ ₁	CJ ₂	CJ ₃
.....

5) 至少一个KEY。

2. 操作

Q、I、D、M、.....

3. 约束——完整性约束 (integrity)

1) 实体完整性；

2) 参照完整性；





学号	姓名	性别
X01	张三	男
X02	李四	女

课程号	课程名	学分
K01	马哲	2.0
K02	数据库	3.5

学号	课程号	成绩
X01	K01	81
X01	K02	76
X02	K01	85
X02	K02	92



1.2 数据模型

3) 用户定义完整性。

4. 优点

1) 建立在严格的数学理论的基础之上。

2) 结构简单易用。

3) 应用程序与数据说明独立：

- 将DB定义从应用程序中独立出来；

- DB存储物理细节透明；

- 存取路经透明；

- 非过程化语言。

4) 集合操作。

5. 缺点

1) 查询效率慢。

2) 复杂数据类型表示能力弱。





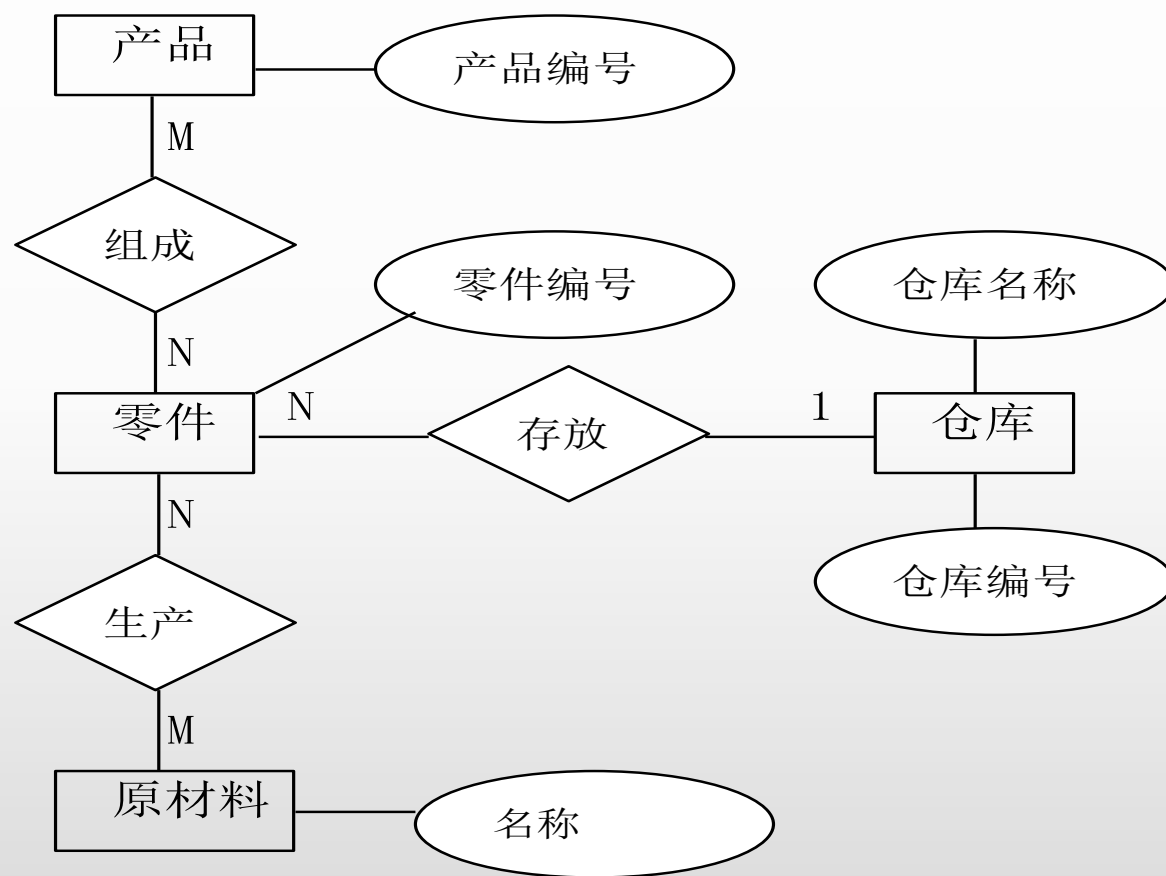
课堂作业

- 设有如下应用

某工厂生产若干产品，每种产品由不同的零件组成，零件可用在不同的产品上。这些零件由不同的原材料制成，不同零件所使用的材料可以相同。这些零件存放在若干仓库中，一类零件存放于一个仓库中。

请用E-R图画出此工厂产品，零件，材料，仓库的概念模型

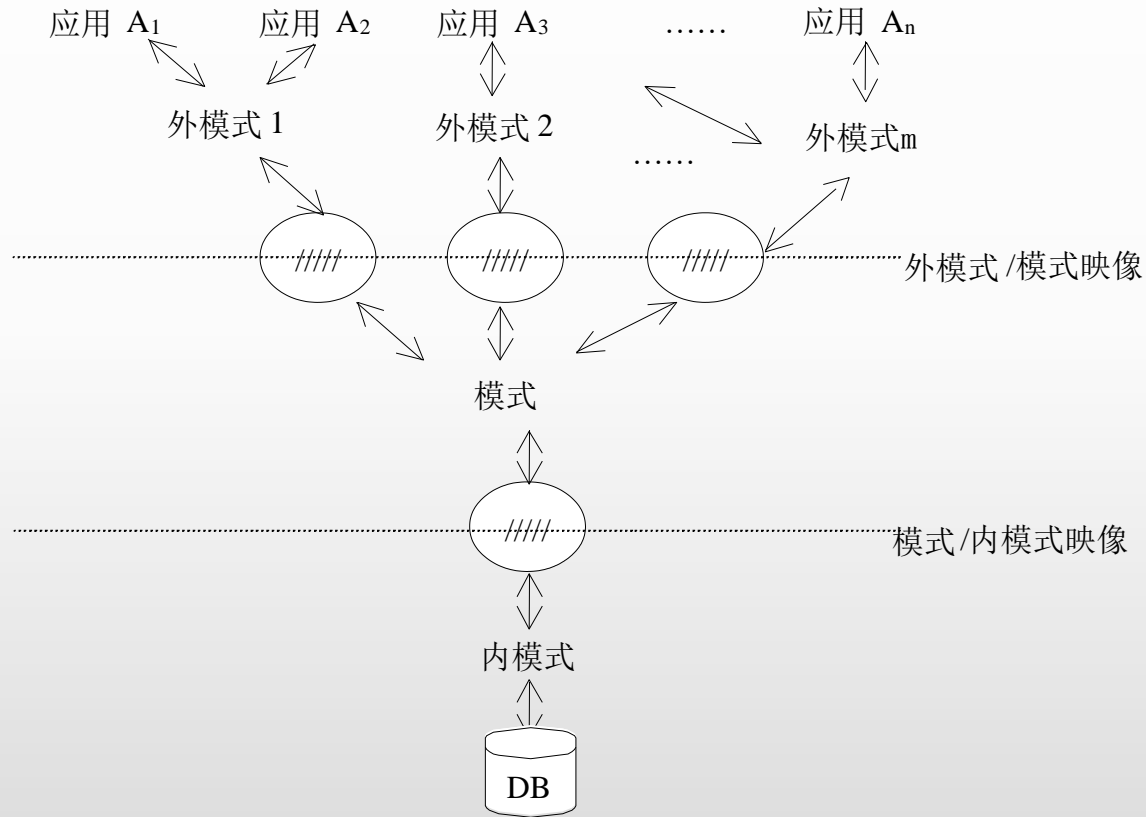
课堂作业



产品，零件，材料，仓库 E-R 概念模型

1.3 数据库系统结构(the structure of DBS)

1.3.1 三层模式结构





1.3 数据库系统结构

1.3.1.1 有关概念

1. 模式 (schema) (概念模式/逻辑模式)

——DB中全体数据的逻辑结构及其特征的说明。

- 全局性：针对DB中整体数据
- 逻辑性：student(XH, XM, YL)
- 特征性：名称、数据、类型、长度、约束
- 说明性：上述结构及特征的表示程序。

用特定语言编写的表达上述结构及特征的程序。

2. 外模式 (External schema) (subschema) (子模式/视图)

——DB中局部 (局部用户) 数据的逻辑结构及其特征的说明。





1.3 数据库系统结构

1) 外模式是模式的子集。

Student (XH, XM, XB, YL)

Course (KH, KM)

SC(XH, KH, CJ)

①单关系子集

student (XH, XM)

②多关系子集

SCI(XH, KH, XM, CJ)

2)子模式间可相互重叠

3)可不同于模式

命名、数据类型、安全约束、结构





1.3 数据库系统结构

- 4) 虚结构：数据仍按原关系模式存储。
- 5) 一个模式可多个外模式
- 6) 一个应用程序只能使用一个外模式
- 7) 多个应用程序可共用一个外模式。

3.内模式 (Internal Schema) (storage-schema) (存储模式)

- DB物理结构、存取路经及存取方法的说明
- 一个模式对应一个内模式。





1.3 数据库系统结构

1.3.1.2 映像

1. 外模式/模式映像

——说明外模式与模式间的对应联系（外模式中说明）。

2. 模式/内模式映像

——说明模式与内模式的对应关系（模式中说明）。

（LS在内部如何组织）





1.3 数据库系统结构

1.3.1.3作用

1. 子模式作用

- 1) 支持不同用户建立适应局部应用特征的结构；
- 2) 简化应用处理；
- 3) 提高安全性；
- 4) 实现数据逻辑独立性：

·分隔应用程序与模式

·模式变，由DBA改变外模式/模式映像，外模式不变，应用程序不变。





1.3 数据库系统结构

2. 模式作用

1) 支持数据少冗余，共享；

Student (XH,XM)

2) 支持数据逻辑独立性；

3) 支持数据物理独立性：

- 分模式与内模式；

- 内模式变，由DBA改变模式/内模式映射，模式不变，子模式不变，应用程序不变。

3. 内模式作用

1) 支持用户建立适应需求的物理结构等；

2) 实现数据物理独立：

- 程序中屏蔽物理细节；

- 内模式变，DBA改变映像，模式不变，外模式不变，应用程序不变。

1.3 数据库系统结构

1.3.2 主从式结构（集中式结构）

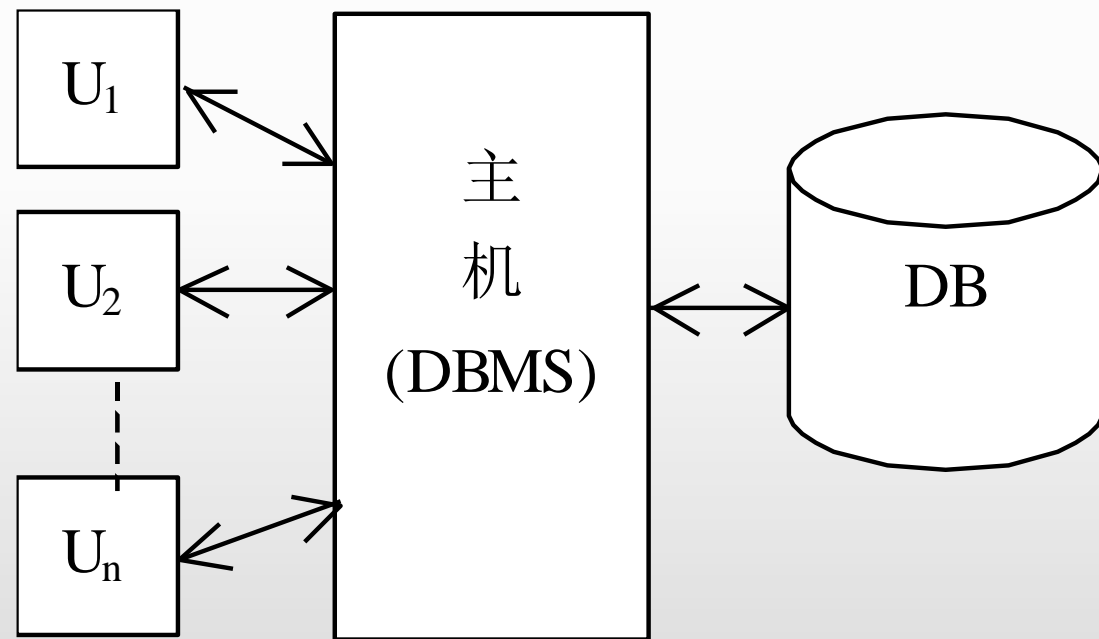
集中式数据库系统体系架构指运行在独立的计算机系统、不和其他计算机系统交互的数据库体系架构。

1、优点

- 1) 结构简单；
- 2) 资源共享性高（外理及数据均由主机完成）
- 3) 数据易于管理与维护。

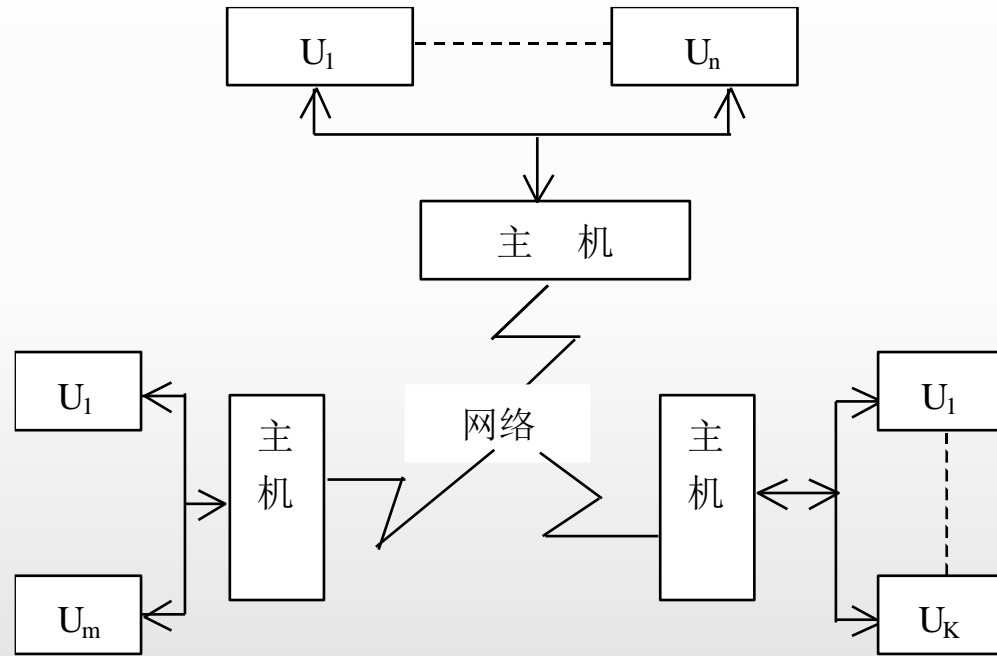
2、缺点

- 1) 主机负担重：用户数增多，I/O瓶颈；
- 2) 可靠性弱（主机故障）。



1.3 数据库系统结构

1.3.3 分布式结构（distribution）





1.3 数据库系统结构

1、优点

- 1) 自治与协调;
- 2) 独立能力;
- 3) 异地数据访问;
- 4) 可靠性高;
- 5) 可用性好。

2、缺点

- 1) 结构与管理复杂;
- 2) 效率受网速影响。



1.3 数据库系统结构

具有多个并行计算单元的计算机架构

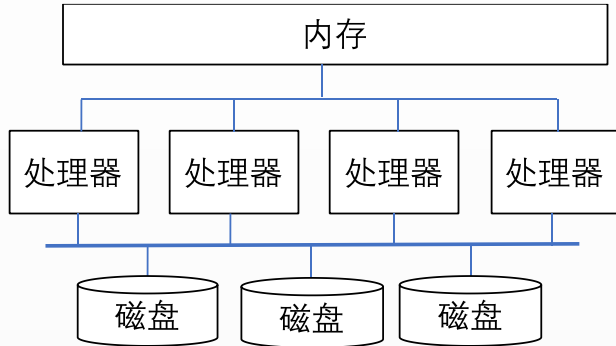
优点

- 高算力、高性能
- 高可靠
- 高并发
- 低成本解决大问题（如双十一。利用便宜的X86服务器组成超大规模nodes运算集群）

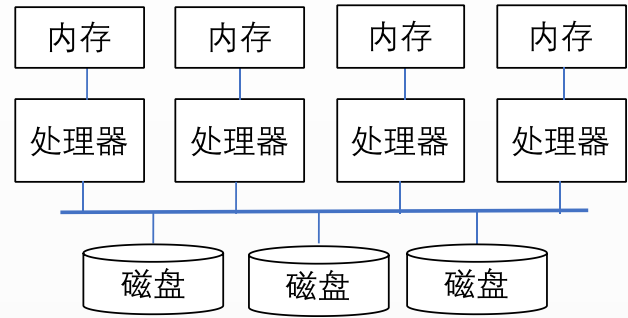
缺点

- 结构与管理复杂

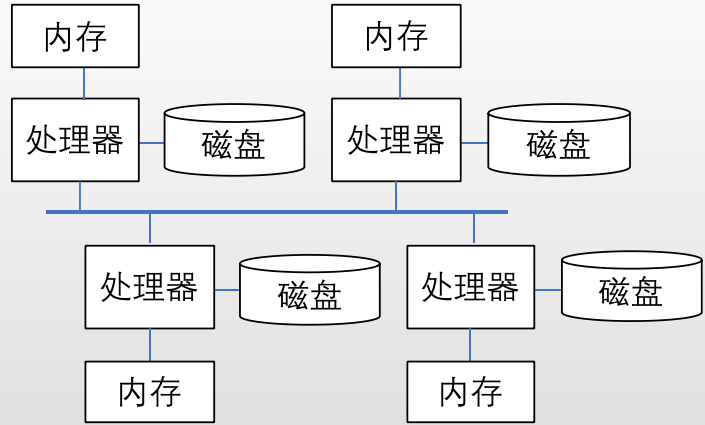
1.3 数据库系统结构



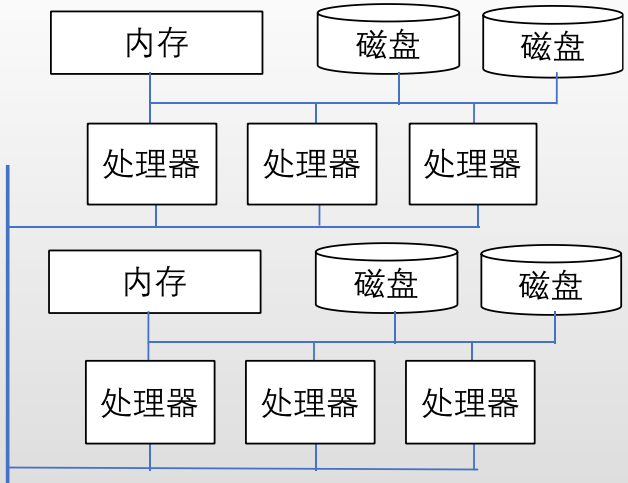
(a) 共享内存类型



(b) 共享磁盘类型



(c) 无共享类型



(d) 混合类型



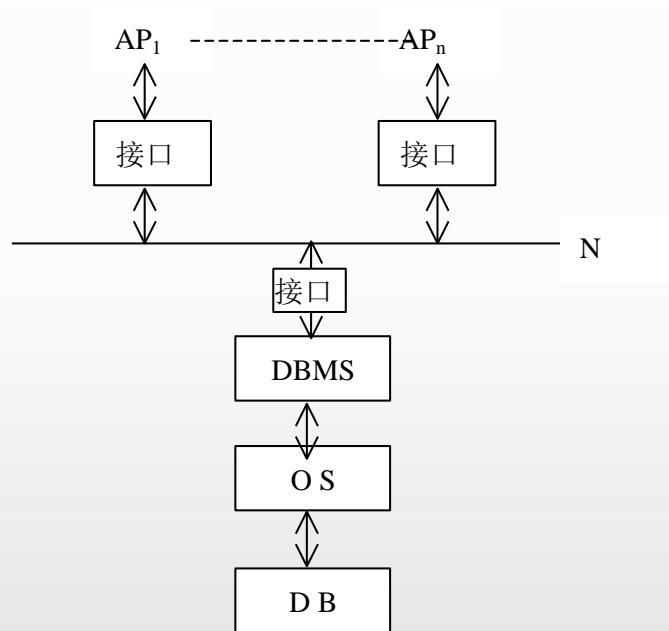
1.3 数据库系统结构

关于分布式和并行

- 客观现实中“分布”和“并行”的广泛存在
- 两者在一定意义上的“等同性”，分布即并行，并行即分布
- 针对具体应用的不同方案，数据密集型、计算密集型

1.3 数据库系统结构

1.3.4 客户/服务器结构 (client/server)



客户发请求到S端，结果返回到C端。





1.3 数据库系统结构

1、优点

- 1) 负载相对均衡，效率提高；
- 2) 减少网络传输量；
- 3) 提高吞吐率；
- 4) 开放性较好。

2、缺点

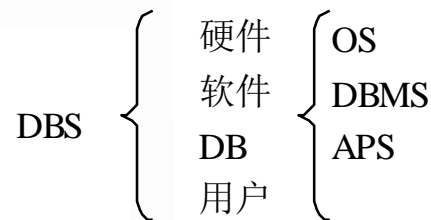
- 1) DB访问瓶颈(若数据集中在S端，要求高性能的服务器)；
- 2) 服务器负担重（数据管理及其应用处理都集中在S上）。





1.4 数据库系统组成

1.4.1 DB系统的组成



1、定义：支持定义、使用和维护DB的系统软件。

2、功能

1) DB定义

模式、子模式、内模式、映射、约束

2) DB操纵 (Manipulation)

I、M、D、Q ……

3) DB存储

存储结构、存取路径、I/O

4) DB运行管理





1.4 数据库系统组成

安全性、完整性检查，DD、索引维护、事务调度、
并发控制

5) DB建立

初始数据输入，数据转换。

6) DB维护

转储与恢复、性能监视与分析、重组、重构。

7) DB通信

OS、Netware、其它DBMS。





1.4 数据库系统组成

3、组成

1) DB定义、操纵语言及编译程序（含预处理及解释）

2) DB运行控制程序

初启程序、安全控制、完整性控制、并发控制、事务管理、I/O、存取路径管理、缓冲区管理、日志管理。

3) 实用程序（utility）

查询、更新、初装、转储、恢复、监测、转换、重组、重构、通讯。





1.4 数据库系统组成

1.4.2 应用程序

主语言+DML下面给出带有嵌入式SQL的一小段C程序。

EXEC SQL INCLUDE SQLCA;..... (1)定义SQL通信区

EXEC SQL BEGIN DECLARE SECTION;..... (2)主变量说明开始

CHAR Sno(5)

CHAR Cno(3)

INT Grade;

EXEC SQL END DECLARE SECTION;.....主变量说明结束

main()

{

EXEC SQL DECLARE C1 CURSOR FOR.....(3)游标操作(定义游标)

SELECT Sno,Cno,Grade

FROM SC;

/*从表中查询Sno,Cno,Grade*/





1.4 数据库系统组成

```
EXEC SQL OPEN C1;.....(4)游标操作(打开游标)
for(;;)
{
    EXEC SQL FETCH C1 INTO :Sno,:Cno,:Grade;
                                .....(5)游标操作(推进游标指针并
    将当前数据放入主变量)
                                if(sqlca.sqlcode<>SUCCESS).....(6)利用SQLCA
    中的状态信息决定何时退出循环
                                break;

    printf("Sno:%s,Cno:%s,Grade:%d", :Sno,:Cno,:Grade);

                                /*打印查询
    结果*/
}
EXEC SQL CLOSE C1;.....(7)游标操作(关闭游标)
```





1.4 数据库系统组成

1.4.3 用户

1、DBA (Database Administrator)

DBMS、DB及其它软件管理与维护（安全授权、监测和改进性能）

2、系统分析员

分析用户需求，确定数据库事务

3、应用程序员

应用软件编码、调试和维护

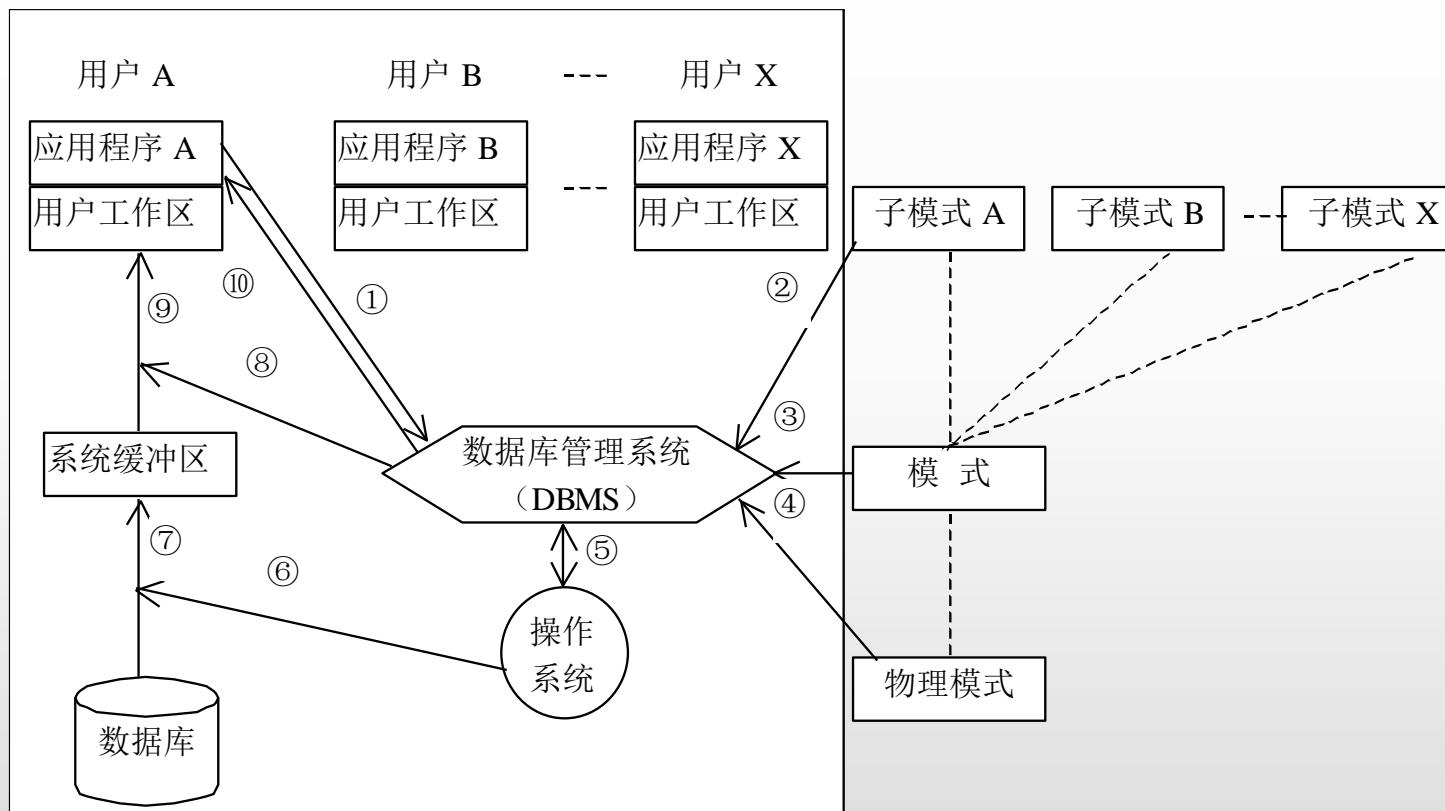
4、终端用户

使用DB



1.5 数据库访问过程

1.5.1 从数据库中读取记录的过程





1.5 数据库访问过程

在数据库系统中，当一个应用程序或用户需要存取数据库中的数据时，应用程序、DBMS、操作系统、硬件等几个方面必须协同工作，共同完成用户的请求。这是一个较为复杂的过程，其中DBMS起着关键的中介作用。

应用程序（或用户）从数据库读取一个数据通常需要以下步骤：

1. 应用程序（或用户）A向DBMS发出从数据库中读数据记录的命令；
2. DBMS对该命令进行语法检查、语义检查，并调用应用程序A对应的子模式，检查A的存取权限，决定是否执行命令，如果拒绝执行，则向用户返回错误信息；





1.5 数据库访问过程

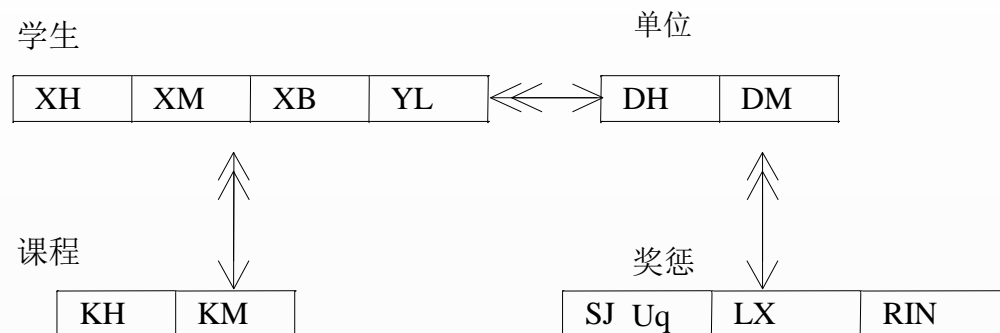
3. 在决定执行该命令后，DBMS调用模式，依据子模式/模式映象的定义，确定应读入模式中的哪些记录；
4. DBMS调用物理模式，依据模式/物理模式映象的定义，决定从哪个文件、用什么存取方式、读入哪个或哪些物理记录；
5. DBMS向操作系统发出执行读取所需物理记录的命令；
6. 操作系统执行读数据的有关操作；
7. 操作系统将数据从数据库的存储区送到系统缓冲区；
8. DBMS依据子模式/模式映象的定义，导出应用程序A所要读取记录的格式；
9. DBMS将数据记录从系统缓冲区传送到应用程序A的用户工作区；
10. DBMS向应用程序返回命令执行情况的状态信息。

图中显示了应用程序（用户）从数据库中读取记录的过程。执行其他操作的过程也与此类似。



1.6 数据库系统特点(the characteristic of DBS)

1、数据结构化



·表示数据间的联系

·可交叉使用

2、数据共享性高

(a)数据项一级；

(b) 模式数据全体共享（授权）；

(c) 新的应用。

3、数据冗余低（redundancy）

——相同数据对象的重复构造与存放。

1) 问题：花费空间，修改麻烦，潜在数据不一致性。

2) 优点：可减少并发冲突。





1.6 数据库系统特点

4、数据独立性高 (independence)

——应用程序独立于其所使用数据的说明的特性。

1) 分类

①逻辑数据独立性

——模式变、变模式/子模式映像，子模式不变，应用程序不变。

②物理数据独立性。

——内模式变，应用程序不变。

2) 目标

①数据定义从应用程序中分离出来；

②编程不考虑逻辑、物理细节；

③简化编程；

④提高应用程序稳定性，应变能力强，减少维护修改。





1.6 数据库系统特点

5、数据安全性 (security)

——防止非授权使用或破坏DB中的数据。

- 1) 身份鉴别;
- 2) 操作授权;
- 3) 加密存储。

6、数据完整性 (integrity)

——数据的正确性，有效性、相容性。

- 实体完整性
- 参照完整性
- 用户完整性

工龄<年龄，身高<3米（用户定义完整性）。

原因：输入不当、修改不当、故障。





1.6 数据库系统特点

7、恢复能力强（recovery）

——将DB从不正确状态恢复到某一正确状态。

备份+日志+回复处理技术

8、数据一致性（consistency）

——任何时刻对同一DB中相同数据的并发访问所获得的值应该是一致的（往往须相同）。

飞机订票系统问题。





1.6 小结

- 数据管理技术的三个阶段及特点
- 数据模型及其三要素
- 概念模型：**E-R模型**
- 三个主要的经典数据模型
- 关系模型
- 数据库的三级模式和两级映象
- 数据独立性
- 数据库系统的组成
- 数据库访问过程

培养学生勇攀高峰、精益求精的责任感、使命感、大国工匠精神

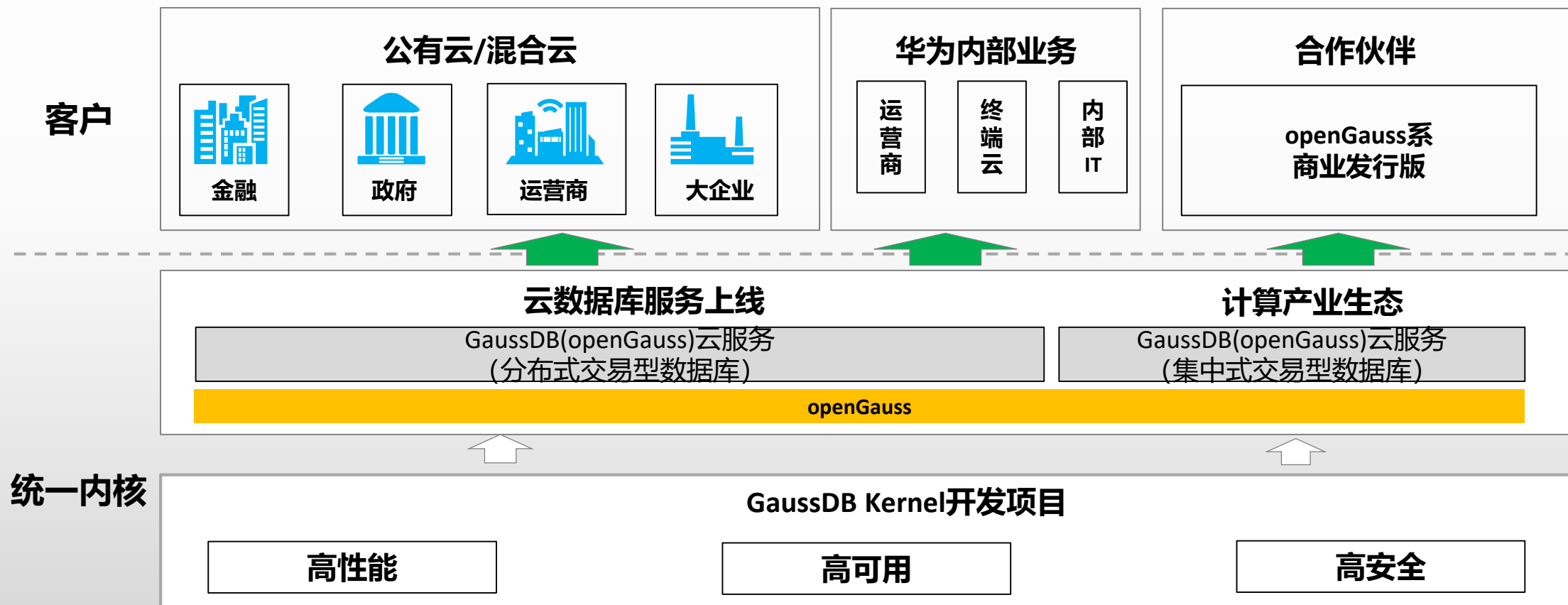


<https://edu.huaweicloud.com/roadmap/colleges.html>



openGauss产品：商用+自用+开源相结合，内核将长期演进

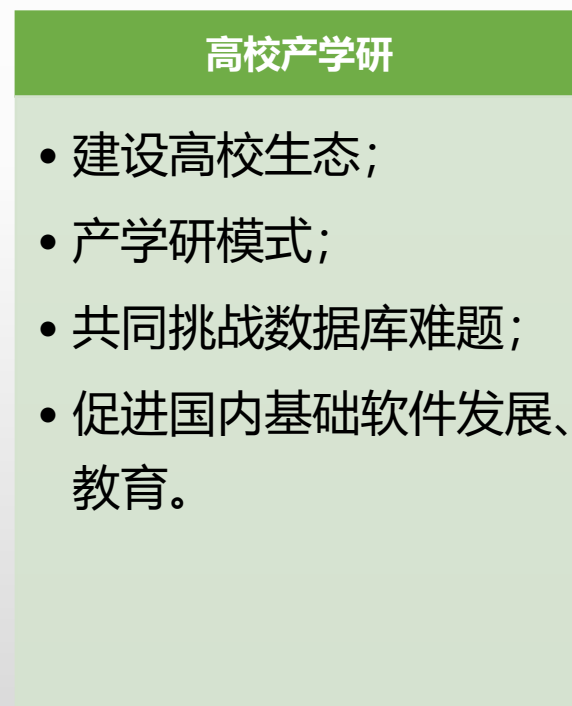
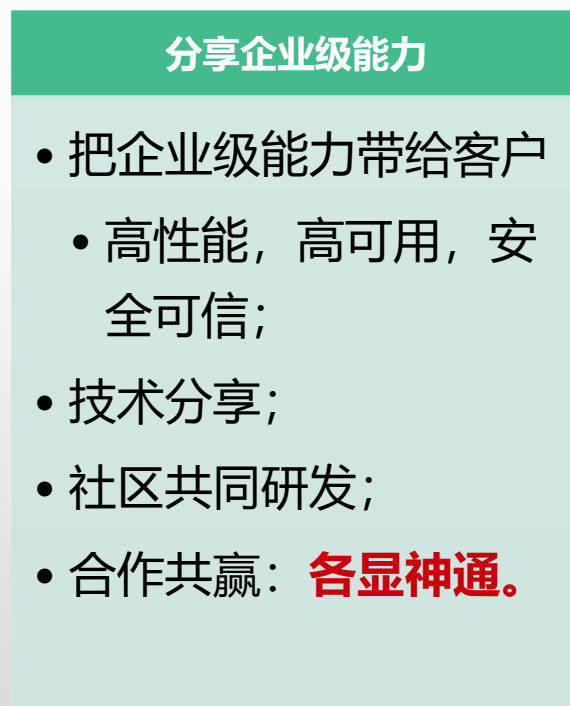
- 华为公司内部配套和公有云的GaussDB服务均基于openGauss，内核将保持长期演进。





openGauss为什么开源? (1)

- 公司战略：硬件开放、软件开源、使能伙伴；
- 通过openGauss开源社区运作，推广华为自有数据库生态，助力鲲鹏计算产业生态构建；
- 聚国内数据库人才，携手并进，共筑国产数据库事业。



openGauss体系架构





openGauss架构 VS PostgreSQL架构 关键技术对比

- openGauss是衍生自PostgreSQL-XC，单机逻辑架构与PG接近。
- openGauss和PG在架构和关键技术上有根本性差异，尤其是存储引擎和优化器两大核心能力。

关键差异化因素		openGauss	PostgreSQL
运行时模型	执行模型	线程池模型，高并发连接切换代价小、内存损耗小，执行效率高，一万并发连接比最优性能损耗<5%。	进程模型，数据库进通过共享内存实现通讯和数据共享。每个进程对应一个并发连接，存在切换性能损耗，导致多核扩展性问题。
事务处理	并发控制	64位事务ID，使用CSN解决动态快照膨胀问题；NUMA-Aware引擎优化改造解决“五把大锁”。	事务ID回卷，长期运行性能因为ID回收周期大幅波动；存在“五把大锁”的问题，导致事务执行效率和多处理器多核扩展性存在瓶颈。
	日志和检查点	增量Checkpoint机制，实现性能波动<5%。	全量checkpoint，性能短期波动>15%。
	鲲鹏NUMA	NUMA改造、cache-line padding、原生spin-lock。	NUMA多核能力弱，单机两路性能TPMC <60w。
数据组织	多引擎	行存、列存、内存引擎，在研DFV存储和原位更新。	仅支持行存。
SQL引擎	优化器	支持SQL Bypass, CBO吸收工行等企业场景优化能力。	支持CBO，复杂场景优化能力一般。
	SQL解析	ANSI/ISO标准SQL92、SQL99和SQL2003和企业扩展包。	ANSI/ISO标准SQL92、SQL99和SQL2003。

openGauss 竞争力总览



把企业级数据库能力带给用户和伙伴

价值

openGauss提供面向多核的极致性能、全链路的业务和数据安全、基于AI的调优和高效运维的能力，全面友好开放，共同打造全球领先的企业级开源关系型数据库；

关键特性

高性能

①

- **两路鲲鹏性能150万tpm**
- 面向多核架构的并发控制技术；
- NUMA-Aware数据结构；
- SQL-Bypass智能选路执行技术；
- ④ ➢ 面向实时高性能场景的内存引擎。

高可用 & 高安全

②

- 业务无忧，故障切换时间RTO<10 s；
- 精细安全管理：细粒度访问控制、多维度审计；
- 全方位数据保护：存储&传输&导出加密、动态脱敏、全密态计算。
- ⑤

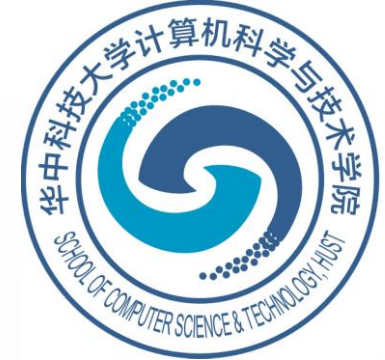
易运维

③

- 基于AI的智能参数调优，提供AI参数自动推荐；
- 慢SQL诊断，多维性能自监控视图，实时掌控系统性能表现；
- 提供在线自学习的SQL时间预测、快速定位，急速调优。

全开放

- 采用木兰宽松许可证协议，允许对代码自由修改，使用、引用；
- 数据库内核能力完全开放；
- 开放运维监控、开发和迁移工具；
- 开放伙伴认证、培训体系及高校课程。

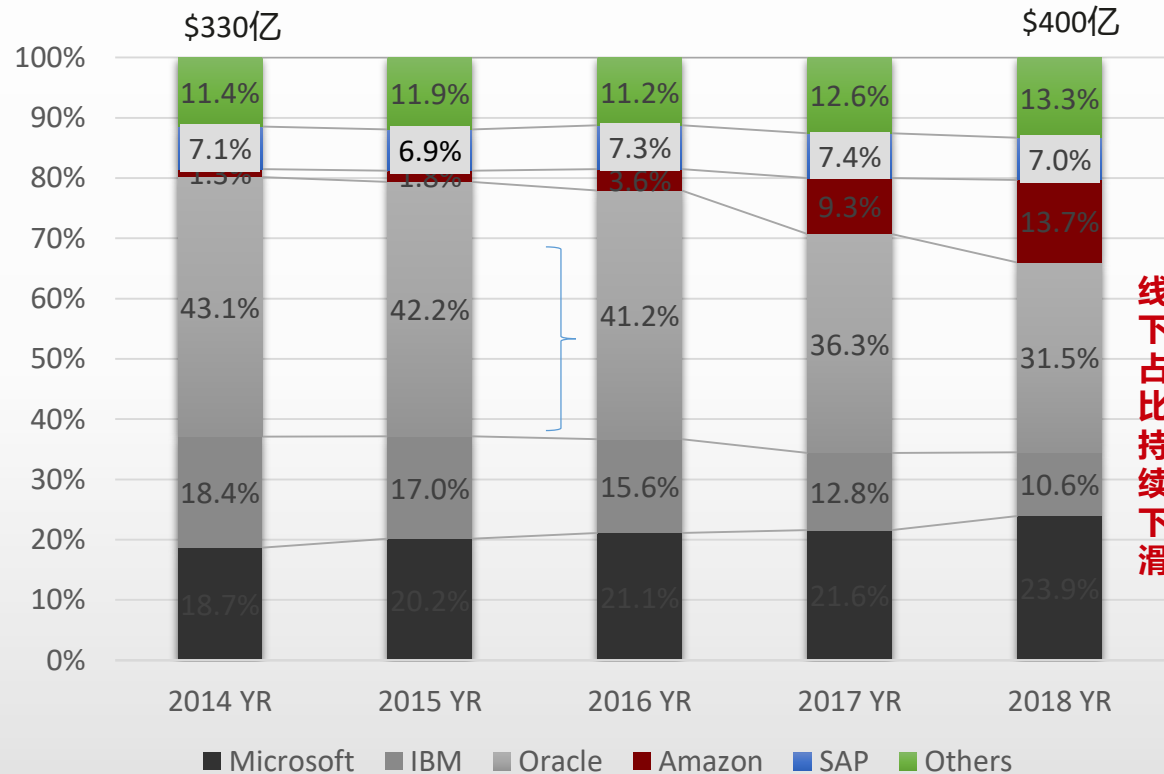


云化数据库是大势所趋 (1)

数据库技术革新正在打破现有秩序，**云化，分布式，多模处理**是未来主要趋势

Source: Gartner

2014年-2018年数据库市场份额变化



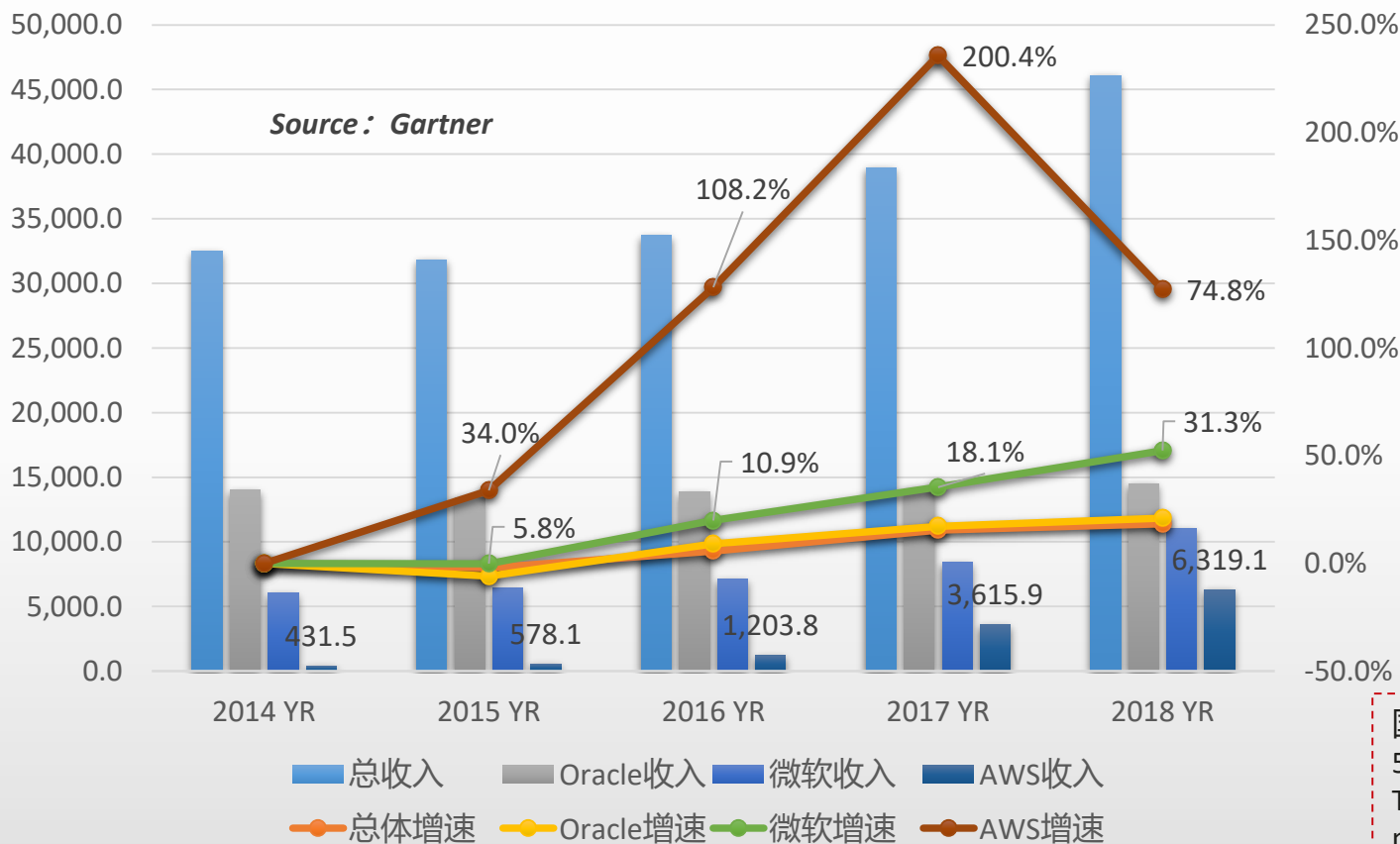
传统线下数据库市场（以IBM+Oracle为代表）占比持续下滑；
2024年，数据库市场份额将达到\$650亿，**全球75%数据库以云服务的方式存在**

国际权威研究机构Gartner 2019年5月发布《The Future of the Database Management System (DBMS) Market Is Cloud》报告，鲜明指出：数据库的未来是上云



云化数据库是大势所趋 (2)

2014年-2018年数据库市场Leader收入及增速



传统厂商Oracle增长停滞VS云厂商AWS增速平均每年90%，营收增加14倍
Gartner预计2024年，AWS的云数据库会超过Oracle

国际权威研究机构Gartner 2019年5月发布《The Future of the Database Management System (DBMS) Market Is Cloud》报告，鲜明指出：数据库的未来是上云



GaussDB数据库升级为全场景云服务

✦ 表示处于待规划状态



GaussDB全场景服务

7大全球区域研究所从事基础研究

10+年数据库领域技术积累

1000+数据库专项人才

30000+全球数据库应用量

