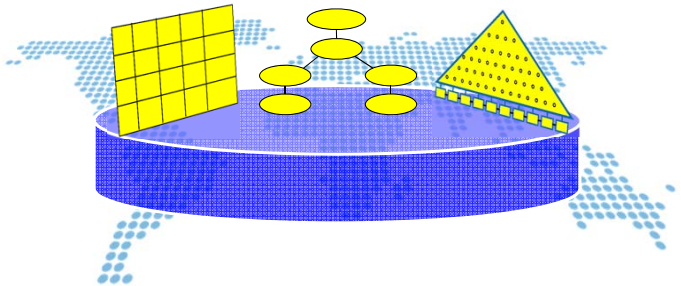
数据库系统

概述

陈世敏
（中科院计算所）



Outline

- 数据库的概念
- 计算机硬件的发展
- 数据库系统的发展

概念？

- 怎么定义一个概念？
  - 内涵：里面有什么，描述它的属性
  - 外延：什么不是？有什么区别？
- 例如：
  - 大学
  - 计算机科学

主要概念

- 所以这里我们的目标是说清楚概念的内涵和外延
  - 数据库（Database）
  - 数据库系统（Database Management System）
  - 关系数据库（Relational Database）
  - 关系数据库系统（Relational Database Management System）
  - 关系模型（Relational Model）

数据库（Database）

- 内涵
  - 包含一组数据
  - 描述了实体（Entity）和它们之间的联系（Relationship）
    - 例如，实体：学生、教师、课程、教室、班级等
    - 联系：学生选课、老师教课、上课占用教室、老师指导学生等
- 外延
  - 不同于文件（File）
    - 从文件系统的角度，不关心内容，而数据库关心数据的内容
    - 当然数据库可能存储在文件中
  - 不同于数据集（Data Set）
    - 没有强调数据的组织，可能是混乱的，未经整理和清洗的
    - 而数据库中的数据都是通过某种方式组织起来的
    - 数据库可以被认为是一种数据集
  - 不同于数据结构（Data Structure）
    - 不关心数据的内容

数据库系统（DBMS, Database Management System）

- 内涵
  - 管理数据库的系统
  - 管理：存储、修改、查询、运算
- 外延
  - 不同于文件系统
    - 管理文件，存储/读取/修改文件
  - 不同于运算系统
    - 例如：Matlab（矩阵运算），Mathematica（公式推导）MapReduce（云计算），MPI（高性能计算）
    - 为了一种特定的运算模型设计的系统，方便编程和运行
  - 不同于网站
    - 提供网页服务，当然网页可以访问后台的数据库系统

关系数据库和关系数据库系统

- 关系模型（Relational Model）
- 关系数据库（Relational Database）
  - 采用关系模型的数据库
- 关系数据库系统（RDBMS, Relational Database Management System）
  - 支持关系模型为核心模型的数据库系统
- 什么是关系模型？

关系模型

- 关系模型中的实体和联系都可以用“表”来表示
- Table/Relation （表）

举例：学生信息表

每一列是一个属性（~10）

每一行是一个学生记录（~10^4）

ID	Name	Birthday	Gender	Major	Year	GPA
131234	张飞	1995/1/1	男	计算机	2013	85
145678	貂蝉	1996/3/3	女	经管	2014	90
129012	孙权	1994/5/5	男	法律	2012	80
…	…	…	…	…	…	
…	…	…	…	…	…	
…	…	…	…	…	…	
…	…	…	…	…	…	
…	…	…	…	…	…	
…	…	…	…	…	…	
…	…	…	…	…	…	

Table/Relation （表）

- 列(Column)：一个属性，有明确的数据类型
  - 例如：数值类型（e.g., int, double），字符串类型(varchar)，类别类型(有些像程序语言中的enum)
  - 必须是原子类型，不能够再进一步分割，没有内部结构
- 行(Row)：一个记录（tuple, record）
  - 表是一个记录的集合
  - 记录之间是无序的
- 通常是一个很瘦长的表
  - 几列到几十列
  - 成千上万行，很大的表可以有亿/兆行

举例：学生信息表

原子类型

数值 字符串 日期 类别 字符串年 数值

每一行是一个学生记录（~10^4）

ID	Name	Birthday	Gender	Major	Year	GPA
131234	张飞	1995/1/1	男	计算机	2013	85
145678	貂蝉	1996/3/3	女	经管	2014	90
129012	孙权	1994/5/5	男	法律	2012	80
…	…	…	…	…	…	
…	…	…	…	…	…	
…	…	…	…	…	…	
…	…	…	…	…	…	
…	…	…	…	…	…	
…	…	…	…	…	…	
…	…	…	…	…	…	

什么是原子类型？无内部嵌套结构

√ Int
√ Double
√ Char string
√ Int 基础上表达的类型：Date, Enum, …

X 程序语言中的struct
X class
X array
X list, set, map …

举例：学生信息表

每一列是一个属性（~10）

每一行是一个学生记录（~10^4）

记录无序

ID	Name	Birthday	Gender	Major	Year	GPA
131234	张飞	1995/1/1	男	计算机	2013	85
145678	貂蝉	1996/3/3	女	经管	2014	90
129012	孙权	1994/5/5	男	法律	2012	80
…	…	…	…	…	…	
…	…	…	…	…	…	
…	…	…	…	…	…	
…	…	…	…	…	…	
…	…	…	…	…	…	
…	…	…	…	…	…	
…	…	…	…	…	…	

举例：学生信息表

每一列是一个属性（~10）

每一行是一个学生记录（~10^4）

非常瘦长

ID	Name	Birthday	Gender	Major	Year	GPA
131234	张飞	1995/1/1	男	计算机	2013	85
145678	貂蝉	1996/3/3	女	经管	2014	90
129012	孙权	1994/5/5	男	法律	2012	80
…	…	…	…	…	…	
…	…	…	…	…	…	
…	…	…	…	…	…	
…	…	…	…	…	…	
…	…	…	…	…	…	
…	…	…	…	…	…	
…	…	…	…	…	…	

表的数学定义

- K列的表：$\{<t_1, \ldots, t_k> \mid t_1 \in D_1, \ldots, t_k \in D_k\}$
  - 整个表是一个集合$\{<t_1, \ldots, t_k>\}$
  - 集合的每个元素有这样的形式$<t_1, \ldots, t_k>$
    - 第j个部分t_j
  - D_j是第j列的类型所对应的所有可能取值的集合（域）

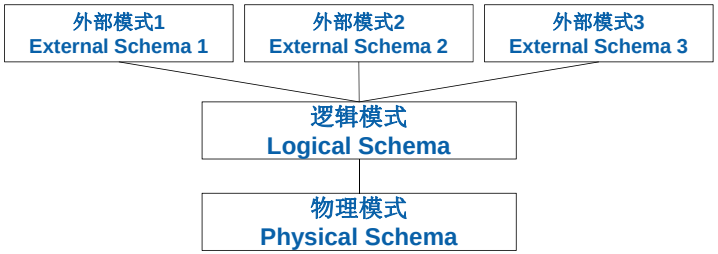
Schema vs. Instance

- Schema: 模式/类型
  - 一个表的类型是由每个列的类型决定的
  - 例如： *Student* (*ID* integer, *Name* varchar(20), *Birthday* date, *Gender* enum(M, F), *Major* varchar(20), *Year* year, *GPA* float)
- Instance: 实例/具体取值
  - 具体存储哪些记录，每个列的具体值
  - 由具体应用决定的
- 这样区分的意义
  - Schema只需要定义一次
  - 可以对应多个instance
  - 随着时间推移，新的修改增删操作，表的内容不断变化，而类型不变

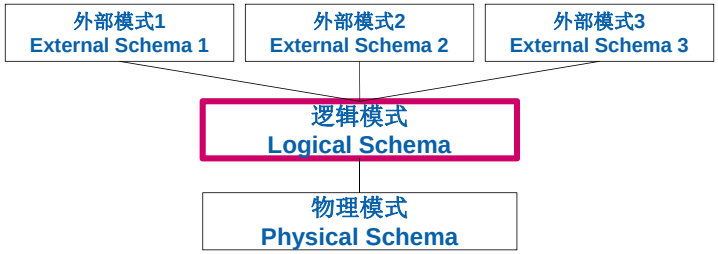
关系模型的优点：数据独立性

- 数据独立性（Data Independence）
  - DBMS内部的数据组织和存储的改变，不影响上层的应用
  - 应用不加修改就可以继续正确地工作
  - 换言之，数据组织和存储的改变对于应用是隔离的
- 如何实现呢？
  - ☞抽象层次

抽象层次

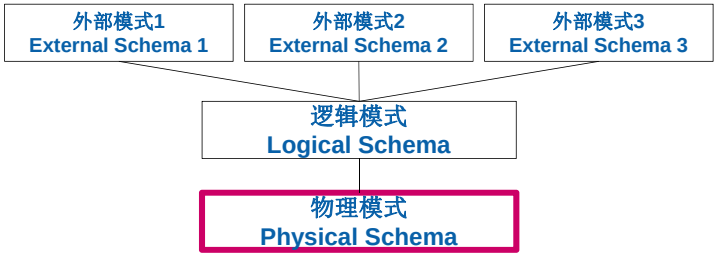


Logical Schema



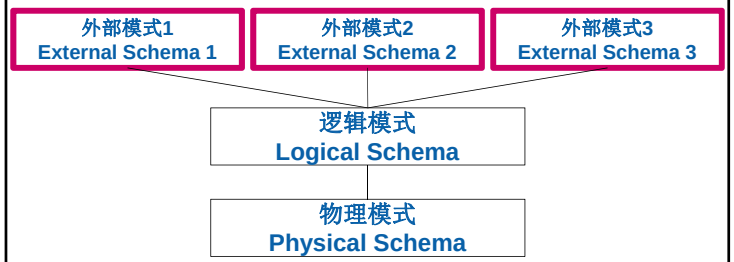
- 这个就是我们前面讲的Schema
- 根据关系模型对数据的定义，也就是数据表的类型定义

Physical Schema



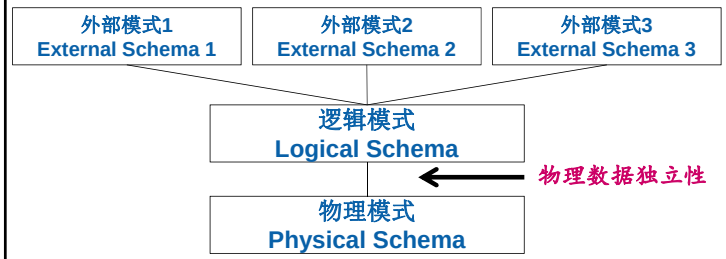
- 确定逻辑模式是如何存储和组织的
- 文件组织和结构？索引结构？

External Schema



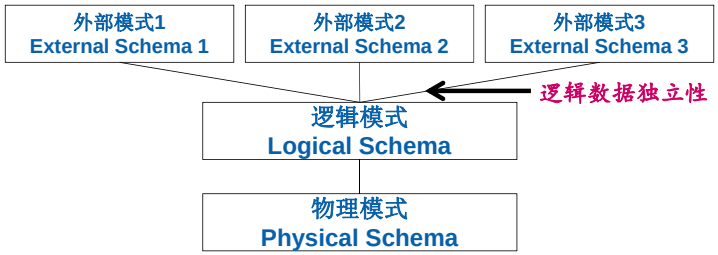
- 出于安全和数据独立性的需要，对于不同的用户（群体），通常只使其看到逻辑模式的一个部分
  - 例如：学生可以看到自己的选课信息，但不能看到其他同学的选课信息
- （会讲到）可以在逻辑模型上使用View来得到

两种数据独立性



- 物理数据独立性（Physical Data Independence）
  - 逻辑模式可以屏蔽物理模式的变化
  - 只要逻辑模式不变，即使物理存储方式发生变化，也不会影响应用

两种数据独立性



- 逻辑数据独立性（Logical Data Independence）
  - 当逻辑模式变化了（例如，增加了新的列）
  - 可以定义与过去相同的外部模式，来使上层应用不受影响

其它模型

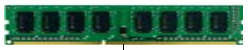
- ER模型
  - 一种外部概念模型，易于表达应用中的数据实体与关系
  - 可以转化为关系模型
- 早于关系模型出现的模型
  - 层次模型（Hierarchical Model）、网状模型（Network Model）
  - 已经基本被淘汰
- 关系模型后出现的模型
  - Object-Oriented data model（面向对象的模型），Object-Relational data model（关系对象模型），XML
  - 获得了有限的成功
- 近期出现的模型：图模型等

Outline

- 数据库的概念
- 计算机硬件的发展
- 数据库系统的发展

80年代的计算机系统

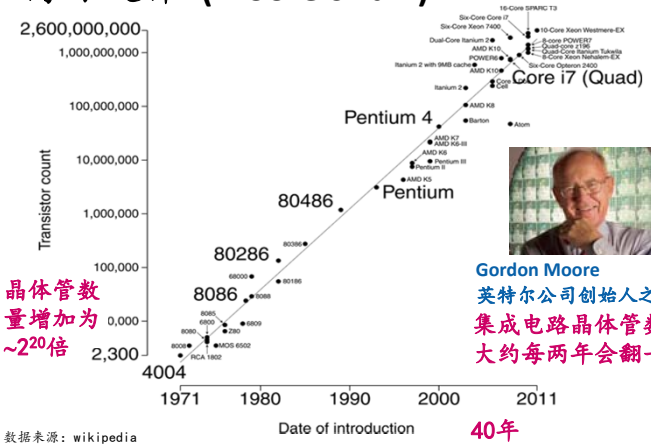
CPU 处理器



DRAM 主存

HDD （Hard Disk Drive）
硬盘

摩尔定律 (Moore's Law)



晶体管数量增加为 ~2^{20}倍

Gordon Moore
英特尔公司创始人之一
集成电路晶体管数量大约每两年会翻一番

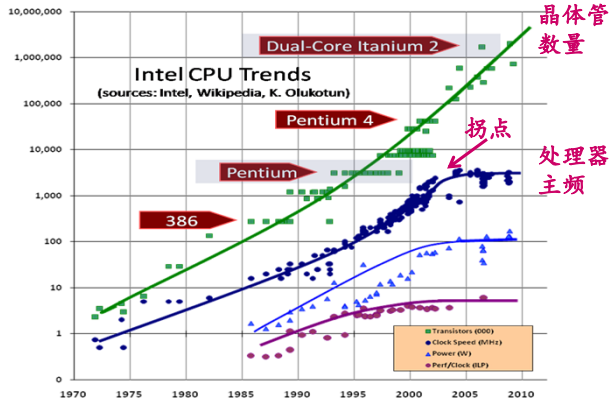
数据来源：wikipedia

40年

CPU体系结构的发展(2004年前)

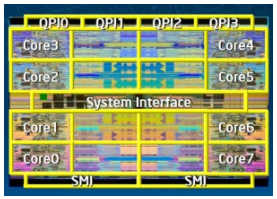
- 提高串行程序效率
  - 提高主频
  - 流水线（Pipeline）
  - 超标量（Super-Scalar）
  - 乱序执行（Out-of-order Execution）
  - 向量指令（SIMD/Vector Instructions）
  - 多级高速缓存（Multi-level Cache）

主频增加这一趋势于2004-2005年间结束



单核单线➔多核多线➔众核

- 功耗、散热等限制处理器主频的进一步增加
- 业界不得不转向使用多核
  - 从主频增加➔核的增加
  - 双核、4核、6核、8核…、28核
- 研究片上网络、高速缓存等

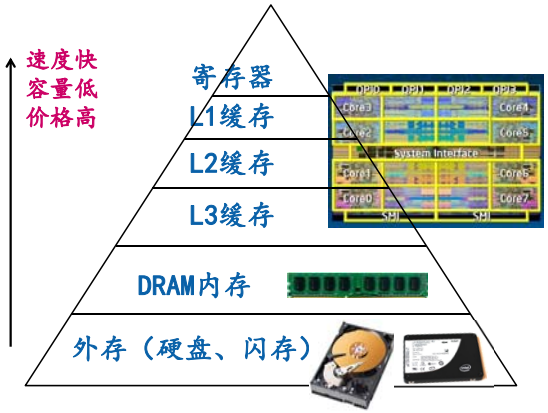


Nehalem EX

多种类型的处理器

- GPU
- Intel Xeon Phi
- ARM

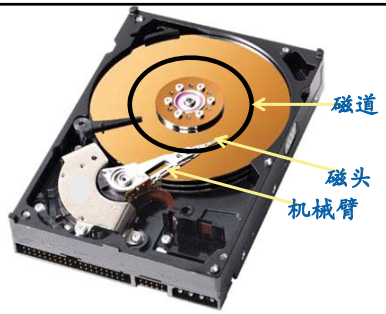
存储层次结构



内存

- 容量➔符合摩尔定律，指数级增加
- 带宽:有一定的办法增加
- 访问速度: 比指令执行慢100倍
  - 访存墙问题

硬盘访问



- 硬盘容量:
  呈指数级增长
- 硬盘的性能
  - 访问速度：受限于机械臂的移动，盘片的转速
    - 大约每次访问需要10ms
  - 带宽：受限于盘片的转速
    - 传输速度大约为100MB/s
  - 顺序访问比随机访问好很多

新型存储/内存设备

- 闪存(Flash)与固态硬盘(SSD: Solid State Drive)
  - 闪存：发明于1980年，与DRAM技术有一定的相似性
  - 最早用于取代ROM作BIOS存储，后来用于数字电子设备：相机、手机、U盘、microSD卡等，大量生产，价格降低
  - 固态硬盘：2009年开始出现以闪存为存储介质的固态硬盘
    - 优点：没有机械装置，随机读性能比硬盘高100倍，顺序读或顺序写性能好于硬盘
    - 缺点：随机写性能差，重写次数有限制(例如，5000次)，超过即报废
- 新型的非易失存储技术：NVM
  - Phase Change Memory，STT-RAM，Memristor，3D-Xpoint
  - 与DRAM的读写速度相似，支持的读写次数相似
  - 非易失：不需要定时刷新，节能，可靠
  - 对计算机系统的各方面都会产生深远的影响

体系结构和硬件技术的巨大发展

计算机系统的发展

苹果2型电脑

80286 PC

Outline

- 数据库的概念
- 计算机硬件的发展
- 数据库系统的发展

关系型数据库

- E.F. Codd于1970年提出了数据管理的关系模型，并因此于1981年获得图灵奖

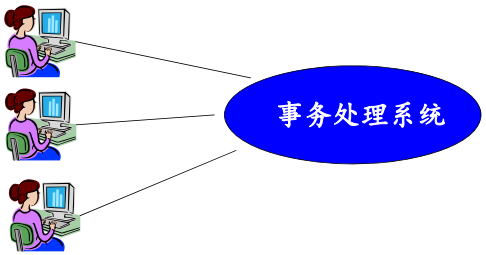
- Jim Gray参与了第一个关系型数据库原型系统（System R）的实现，并由于对数据库和事务处理的多项贡献获得了1998年的图灵奖

- Michael Stonebraker主持了另一个早期关系型数据库原型系统（Ingress）的实现，并实现了一系列的系统（Postgres, C-Store, H-Store, 等），2015年获得图灵奖

ORACLE Microsoft SQL Server IBM DB2

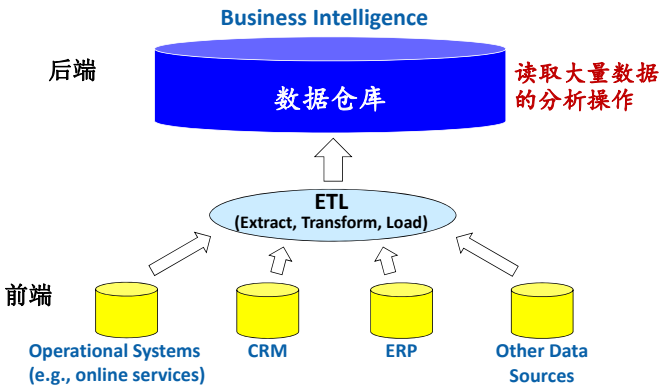
- 三大数据库产品Oracle, Microsoft SQL Server, IBM DB2的最初实现都是在1970末到1980年代

事务处理 (Transaction Processing)



- 早期的数据库系统主要针对事务处理应用
- 典型例子：银行业务，订票，购物等
- 大量并发用户，少量随机读写操作

数据仓库 (Data Warehouse)



多种发展

- 数据流处理
- 地理信息系统
- 多媒体数据库
- 用于Web的后端
- 。 。 。 。 。 。
- 大数据系统

数据库系统课程内容

- 如何使用数据库？
- 数据库是如何工作的？
- 新进展介绍

数据库系统课程内容

- 如何使用数据库？
  - ER模型、关系模型、SQL
  - 实验1：数据库安装与使用
  - 实验2：数据库应用设计
- 数据库是如何工作的？
  - 数据存储与访问路径
  - 查询处理
  - 查询优化
  - 事务处理系统
  - 数据仓库
  - 实验3：分析型数据库系统实现
- 新进展介绍
  - 内存数据库
  - 大数据初步

小结

- 数据库系统的基本概念
- 计算机体系结构和硬件技术的发展
- 数据管理系统的发展