1. 目录
2. Chapter 1
   1. 概览

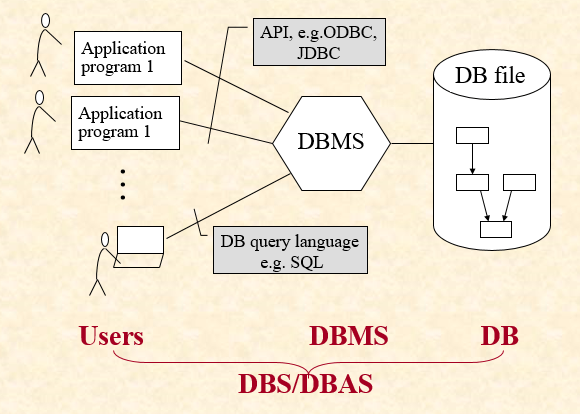
* DBS概念和组成
* 数据抽象与数据独立性
* DBS设计阶段与数据模型
* Database Languages（DDL, DML）
* 数据库背景知识
  1. DBS概念和组成
     1. 数据库系统

数据库系统（DBS）是由一个相互关联的数据的集合和一组用以访问这些数据的程序组成

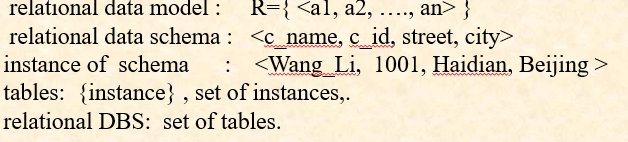
这个数据集合成为数据库（database）

* + 1. 目标

提供一种可以方便、高效地存取数据库信息的途径



* 1. 数据视图
* 数据库系统是一些互相关联的数据以及一组使得用户可以访问和修改这些数据的程序的集合。
* 数据库系统的一个主要目标是给用户提供数据的抽象视图，也就是说，系统隐藏关于数据存储和欸和的某些细节。
  + 1. 数据抽象
       1. 物理层（physical level）：描述数据实际上怎样存储的
       2. 逻辑层（logical level）：描述数据库中存储什么数据以及这些数据间存在什么关系。虽然逻辑层的实现可能涉及复杂的物理层结构，但逻辑层用户不必知道这样的复杂性，即数据独立性
       3. 视图层（view level）：只描述整个数据库的某个部分
    2. 实例和模式
* 特定时刻存储在数据库中的信息的集合称作数据库的一个实例（instance）
* 数据库的总体设计称作数据库模式（schema）;数据库模式即使发生变化，也不频繁
* 应用程序若不依赖于物理模式，即具有物理数据独立性（Physical data independence），即使物理模式改变了它们也无需重写



* + - 1. 几种模式
         1. 物理模式（Physical schema）:在物理层描述数据库的设计
         2. 逻辑模式（logical schema）：在逻辑层描述数据库设计
         3. 子模式（subschema）:在视图层，描述了不同的视图
    1. 数据模型
* 数据库结构的基础是数据模型
* 数据模型是一个描述数据、数据联系、数据语义、以及一致性约束的概念工具的集合。a collection of conceptual tools for describing data, data relationship, data semantics, consistency
  + - 1. 分类
         1. 关系模型（relational model）：用表的集合来表示数据和数据间的关系。每个表有多个列，每列有唯一的列名。
         2. 实体-联系模型（entity-relationship model）:现实世界由一组称为实体的基本对象及这些对象件的联系构成
         3. 基于对象的数据模型（object-based data model）
         4. 半结构化数据模型
  1. 数据库语言
     1. 数据操纵语言

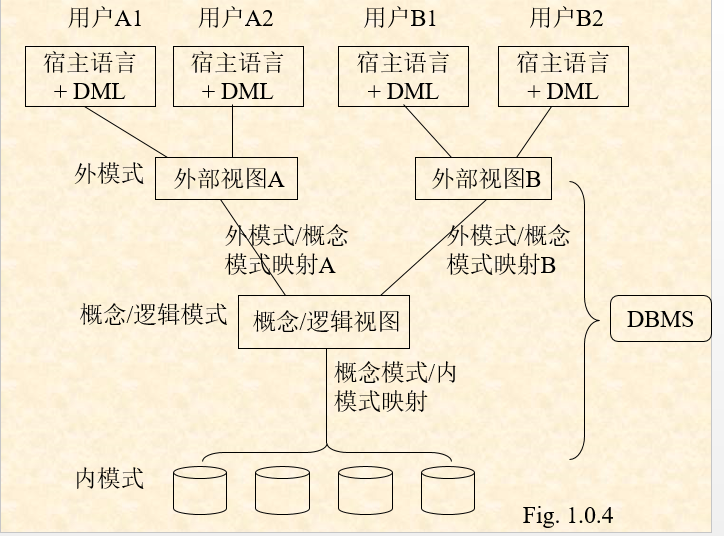
Data-Manipulation Language,DML 是这样一种语言：它使得用户可以访问或操纵那些按照某种适当的数据模型组织起来的数据

* 信息检索
* 向数据库插入新的信息
* 删除信息
* 修改信息
  + - 1. 过程化DML(procedural DML)：要求用户指定需要什么数据以及如何获得这些数据
      2. 声明化DML(declarative DML)：只要求用户指定需要什么数据，而不声明如何获得这些数据
    1. 数据定义语言
* 数据库模式是通过一系列定义来说明的，这些定义由数据定义语言（Data Defenition Language，DDL）来表达
* 存储在数据库中的数据必须满足某些一致性约束（consistency constraint）
* DDL以一些指令（语句）作为输入，生成一些输出。DDL的输出放在数据字典中，数据字典包含了元数据
  + - 1. 域约束（domin constraint）:每个属性都必须对应于一个所有可能的取值构成的域
      2. 参照完整性（referential integrity）:一个关系中给定属性集上的取值也在另一关系的某一属性集中的取值中出现；数据库的修改会导致参照完整性的破坏
      3. 断言（assertion）：一个断言就是数据库需要时刻满足的某一条件。域约束和参照完整性约束是断言的特殊形式。
      4. 授权（authorization）：对不同用户在数据库中的而不同数据值上允许不同的访问类型
  1. **数据库设计**
     1. 设计过程

从保持data independence（数据无关性/独立性）角度出发，根据data models所定义的数据规范形式，在view、logical 、 physical三个层次，采用三种data abstraction方法，通过DB概念设计、DBS逻辑设计、 DBS物理设计三个阶段，构造 面向具体应用领域的DBS的 external schema、logical schema 、internal schema的集合, 从而得到conceptual DBS 、logical DBS 、physical DBS的设计结果

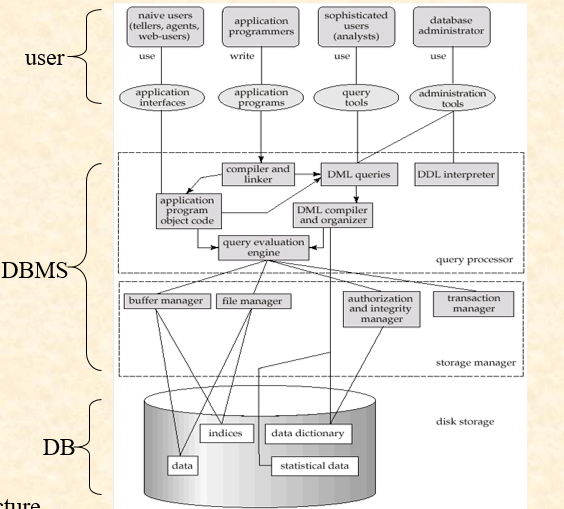
* 初始阶段：全面刻画预期的数据库用户的数据需求
* 概念设计阶段：选取一个数据模型，运用该模型的概念将需求转化为数据库的概念模式。涉及到决定数据库中该包含哪些属性，以及如何将这些属性组织到多个表中
  + 后者主要有两种方法：使用实体-联系模型；引入一套算法（规范化），这套算法将所有属性集作为输入，生成一组关系表
  + 逻辑设计阶段：设计者将高层的概念模式映射到将要使用的数据库的实现数据模型上
  + 物理设计阶段





* 1. **数据存储和查询**
     1. 存储管理器
* 存储管理器是数据库系统中负责在数据库中存储的低层数据与应用程序以及向系统提交的查询间提供接口的部件。
* 负责数据库中的存储、检索和更新
  + 1. 查询处理器

子系统编译执行DDL和DML语句

* 1. 数据库体系结构
* DBS = User/DBA + DBMS + DB
* DBMS

query processing

storage manager

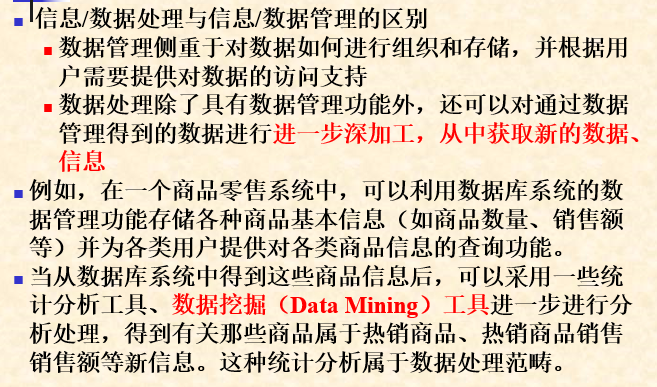
* DB files

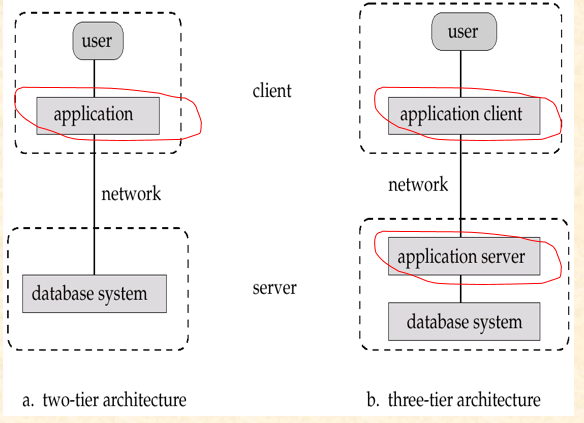
application data

data dictionary/directory

indices

statistic data



1. Chapter 2
   1. 关系数据库的结构

* 关系数据库由表的集合构成，每张表有唯一的名字
* 关系relation用来指代表，元组tuple用来指代行，属性attribute指代表中的列
* 关系实例 relation instance 表示一个关系的特定实例
* 元组在关系中出现的顺序是无关紧要的
* 关系中属性的允许取值的集合：域 domain
* Null表示未知或不存在
  1. 数据库模式