# 数据库系统概念

### Chapter1 引言

#### 总结

·**数据库管理系统**（DataBase-Management System，DBMS）由相互关联的数据集合以及一组用于访问这些数据的程序组成。数据描述特定的企业。

·DBMS的主要目标是为人们提供方便、高效的环境来存储和检索数据。

·如今数据库系统无处不在，很多人每天直接或者间接地与数据库系统打交道。

·数据库系统设计用来存储大量的信息。数据的管理既包括信息存储结构的定义，也包括提供处理信息的机制。另外数据库系统还必须提供所存储信息的安全性，以处理系统崩溃或者非授权访问企图，如果数据在多个用户之间共享，系统必须避免可能的异常结果。

·数据库系统的一个主要目的是为用户提供数据的抽象视图，也就是说，系统隐藏数据存储和维护的细节。

·数据库结构的基础是**数据模型**（data model）：一个用于描述数据、数据之间的联系、数据语义和数据约束的概念工具的集合。

·关系数据模型是最广泛使用的将数据存储到数据库中的模型。其他的数据模型有面向对象模型、对象-关系模型和半结构化数据模型。

·**数据操纵语言**（Data-Manipulation Language，DML）是使得用户可以访问和操纵数据的语言。当今广泛使用的是非过程化的DML，他只需要用户指明需要什么数据，而不需要指明如何获得这些数据。

·**数据定义语言**（Data-Definition Language，DDL）是说明数据库模式和数据的其他特性的语言。

·数据库设计主要包括数据库模式的设计。实体-联系（E-R）数据模型是广泛用于数据库设计的数据模型，他提供了一种方便的图形化的方式来观察数据、联系和约束。

·数据库系统由几个子系统构成：

··**存储管理器**（storage manager）子系统在数据库中存储的低层数据与应用程序和向系统提交的查询之间提供接口。

··**查询处理器**（query processor）子系统编译和执行DDL和DML语句。

·**事务管理**（transaction management）负责保证不管是否有故障发生，数据库都要处于一致的（正确的）状态。事务管理器还保证并发食物的执行互不冲突。

·数据库系统的体系结构受支持其运行的计算机系统的影响很大。数据库系统可以是集中式的，或者客户-服务器方式的，即一个服务器为多个客户机执行工作。数据库系统还可以设计成具有充分利用并行计算机结构的能力。分布式数据库跨越多个地理上分布的互相分离的计算机。

·典型地，数据库应用可被分为运行在客户机上的前端和运行在后端的部分。在两层的体系结构中，前端直接和后端运行的数据库进行通信。在三层结构中，后端又被分为应用服务器和数据库服务器。

·知识发现技术是同自动地从数据中发现统计规律和模式。**数据挖掘**（data mining）领域将人工智能和统计分析研究人员创造的只是发现技术，与使得知识发现技术能够在极大的数据库上高效实现的技术结合起来。

·有4种不同类型的数据库用户，按用户期望与数据库进行交互的不同方式来区分它们。为不同类的用户设计了不同的用户界面。

#### 术语回顾

·**数据库管理系统（DBMS）** 由相互关联的数据集合以及一组用于访问这些数据的程序组成。数据描述特定的企业。

·**数据库系统应用** 企业：销售、会计、人力资源、生产制造、联机零售；银行和金融：银行业、信用卡交易、金融业；大学；航空业；电信业。

·**文件处理系统** 永久记录被存储在多个不同的文件中，人们编写不同的应用程序来将记录从有关文件中取出或加入到适当的文件中。

·**数据不一致性** 相同的信息可能在几个地方重复存储，同一数据的不同副本不一致。

·**一致性约束** 发生故障时数据应该被恢复到故障发生以前一致的状态。

·**数据抽象** 对用户屏蔽复杂性，简化用户与系统的交互（物理层、逻辑层、视图层）

·**实例** 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

·**模式** 数据库总体的设计称作数据库模式

··**物理模式** 在物理层描述数据库的设计

··**逻辑模式** 在逻辑层描述数据库的设计

·**物理数据独立性** 应用程序不依赖于物理模式，物理模式隐藏在逻辑模式下，并且通常可以在应用程序丝毫不受影响的情况下被轻易地更改。

·**数据模型** 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。

··**实体-联系模型** 现实世界由一组称作实体的基本对象以及这些对象间的联系构成。

··**关系数据模型** 用表的集合来表示数据与数据间的联系。

··**基于对象的数据模型** 实体-联系模型增加了封装、方法（函数）和对象标识等概念后的扩展

··**半结构化数据模型** 相同类型的数据项含有不同的属性集的数据定义

·**数据库语言** 数据定义语言+数据操纵语言

··**数据定义语言** 说明数据库模式的定义，定义了数据库模式的实现细节。创建

··**数据操纵语言** 检索，插入，删除，修改

··**查询语言** 检索

·**元数据** 关于数据的数据，数据字典中的数据。

·**应用程序** 使用嵌入式的SQL查询来访问数据库中的数据来与数据库进行交互的程序

·**规范化** 生成一个关系模式集合，使我们存储信息时没有不必要的冗余，同时又能轻易地检索数据

·**数据字典** 包含了元数据，一种特殊的表，只能由数据库系统本身来访问修改

·**存储管理器** 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之间提供接口的部件（权限及完整性管理器，事务管理器，文件管理器，缓冲区管理器，数据文件，数据字典，索引）

·**查询处理器** （DDL解释器：解释DDL语句并将这些定义记录在数据字典中；DML编译器：将查询语言中的DML语句翻译为一个执行方案，包括一系列查询执行引擎能理解的低级指令；查询执行引擎：执行由DML编译器产生的低级指令）

·**事务** 数据库应用中完成单一逻辑功能的操作集合

··**原子性** 一个事务要么发生要么不发生

··**故障恢复** 检测系统故障并将数据库恢复到故障发生之前的状态

··**并发控制** 控制并发事务的相互影响，保证数据库的一致性

·**两层和三层数据库体系结构** 客户-数据库（通过查询语言表达式调用数据库功能：ODBC/JDBC）；客户-应用服务器-数据库（客户机只有前端，与应用服务器通信，应用服务器与数据库通信，应用程序的业务逻辑嵌入到应用服务器中）

·**数据挖掘** 半自动地分析大型数据库并从其中找到有用的模式的过程

·**数据库管理员（DBA）** 对数据库系统进行集中控制的人（模式定义，存储结构及存取方法定义，模式及物理组织的修改，数据访问授权，日常维护）

#### 实践习题

* 1. **这一章讲述了数据库系统的几个主要优点，他有哪两个不足之处？**

1. 数据库系统的建立需要大量的知识，金钱，技巧和时间。
2. 数据库的复杂性可能会导致性能不佳。
   1. **列出Java或C++之类的语言中的类型说明系统与数据库系统中使用的数据定义语言的5个不同之处。**
3. 数据库系统使用DDL语句的执行结果来说明类型，编程语言类型说明在程序中
4. DDL允许一致性约束，域约束、参照完整性，编程语言通常不允许
5. DDL支持授权，编程语言没有
6. 编程语言的类型比SQL类型多：数据，对象等，SQL只有数值，字符等等
7. DDL基于关系属性声明的类型，编程语言支持对象和对象集
   1. **列出为一个企业建立数据库的6个主要步骤。**

企业需求文档、数据类型和数据关系模型、数据完整性约束、物理层、用户界面、建立数据库

* 1. **大学要维护的至少三种不同类型的信息（除1.6.2节列出的）**

学生住宿及水电费缴纳情况、图书馆书籍和书籍借阅、校院系活动和活动开支

* 1. **你想建立YouTube类似的视频节点。考虑1.2节中列出的将数据保存在文件系统中的各个缺点，讨论每一个缺点与存储实际的视频数据和关于视频的元数据（诸如标题、上传他的用户、标签、观看他的用户）的关联。**

如果用文件系统储存视频数据而不是数据库的话：

1. 数据的冗余和不一致性：不同的用户有不同的订阅，每个用户文件夹下都要有这个视频，重复存储；视频可能被上传者更新而在其他的地方仍然存在旧版本
2. 数据访问困难：不同的视频搜索导致不同的查询，这要求不同的程序
3. 数据孤立：用户给自己的订阅视频重命名，导致检索困难
4. 完整性问题：并没有
5. 原子性问题：上传视频要么成功要么失败，不能传一半
6. 并发访问异常：上传视频时要是同时的话，给出的编码可能相同
7. 安全性问题：会员福利！
   1. **在Web查找中使用的关键字查询与数据库查询很不一样。请列出这两者在查询表达方式和查询结果是什么方面的主要差异。**

Web是关键字查询，DB查询是条件精确查询；Web结果是URL列表，DB结果是表格

* 1. **列出4个你使用过的很可能使用了数据库来存储持久数据的应用。**

同花顺股票、阿里旅行、bilibili

* 1. **列出文件处理系统和DBMS的四个主要区别**

1. 文件处理系统只能够物理访问，DBMS可以物理访问和逻辑访问
2. DBMS允许全部被授权者访问，文件处理系统数据只能对应唯一的程序
3. DBMS允许灵活访问，文件处理系统要预先写好程序
4. DBMS允许并行访问，文件处理系统不行
   1. **解释物理数据独立性的概念，在数据库中的重要性。**

改变物理数据但是不用重写应用程序，对用户隐藏了底层的东西

* 1. **数据库管理的5个职责，每个职责不能履行时会产生的问题。**

1. 与文件管理互动；否则数据库将没有意义，因为没有任何储存
2. 保证一致性；否则数据库会出现不合逻辑的错误
3. 保证数据安全；否则未授权的用户可能对数据库产生危害
4. 备份恢复；否则出现错误数据会丢失
5. 保证并发访问；否则数据出错
   1. **给出2个理由说明数据库系统使用声明性查询语言SQL，而不是只提供C/C++的函数库来执行数据操作。**
6. 容易学习
7. 写语句时不用考虑性能问题，由数据库系统主动选择
   1. **解释用图1-4的表来设计会导致的问题**
8. 更改院系预算时需要在每一个该院系教师后面的修改，增加复杂性；更改教师院系的时候需要同时更改预算和办公地点
9. 一个院系需要至少一个导师才能存在；如果导师没了，大楼和预算信息也没了
   1. **DBA的5个作用**
10. 模式定义
11. 存储结构和存取方式定义
12. 模式及物理组织的修改
13. 否则未授权的用户可能会干扰数据库
14. 日常维护：备份、保证运行空间、监视运行
    1. **两层和三层体系结构之间的区别，对Web来说那种合适？**

当然是三层，这样客户机和应用服务器发生关系而不是直接和数据库关联，这样可以简化客户操作

* 1. **可能被用于存储一个社会网络系统如Facebook中的信息的至少3个表。**

1. 用户和基本信息表
2. 用户所发的图片和状态表
3. 用户和其他用户关系表
4. 用户状态阅读授权表

## Part1 关系数据库

### Chapter2 关系模型介绍

#### 总结

·**关系数据模型**（relational data model）建立在表的集合的基础上。数据库系统的用户可以对这些表进行查询，可以插入新元组、删除元组以及更新（修改）元组。表达这些操作的语言有几种。

·关系的**模式**（schema）是指他的逻辑设计，而关系的**实例**（instance）是指他在特定时刻的内容。数据库的模式和实例的定义是类似的。关系的模式包括他的属性，还可能包括属性类型和关系上的约束，比如主码和外码约束。

·关系的**超码**（superkey）是一个或多个属性的集合，这些属性上的取值保证可以唯一识别出关系中的元组。候选码是一个最小的超码，也就是说，他是一组构成超码的属性集，但这组属性的任意子集都不是超码。关系的一个候选码被选作**主码**（primary key）。

·在参照关系中的**外码**（foreign key）是这样一个属性集合：对于参照关系中的每个元组来说，它在外码属性上的取值肯定等于被参照关系中某个元组在主码上的取值。

·**模式图**（schema diagram）是数据库中模式的图形化表示，它显示了数据库中的关系，关系的属性、主码和外码。

·**关系查询语言**（relational query language）定义了一组运算集，这些运算可作用于表上，并输出表作为结果。这些运算可以组合成表达式，表达所需的查询。

·**关系代数**（relational algebra）提供了一组运算，他们以一个或者多个关系为输入，返回一个关系作为输出。诸如SQL这样的实际查询语言是基于关系代数的，但增加了一些有用的句法特征。

术语回顾