第 5 章：数据库系统　　作者：希赛教育软考学院　来源：希赛网　2014年01月26日

# 数据库管理系统的功能和特征

**第5章 数据库系统**

在考试大纲中，有关数据库系统的知识点包括：

数据库管理系统的功能和特征；

数据库模型（概念模式、外模式、内模式）；

数据模型，ER图，第一范式、第二范式、第三范式；

数据操作（集合运算和关系运算）；

数据库语言（SQ）；

数据库的控制功能（并发控制、恢复、安全性、完整性）；

数据仓库和分布式数据库基础知识。

## 5.1 数据库管理系统的功能和特征

数据管理技术的发展大致经历了人工管理阶段（20世纪50年代中期前）、文件系统阶段（20世纪50年代后期到60年代中期）、数据库阶段（20世纪60年代末到70年代末）和高级数据库技术阶段（20世纪80年代初开始）。

数据库是长期存储在计算机内的、有组织的、可共享的数据的集合。

数据库管理系统（DBMS）是一种负责数据库的定义、建立、操作、管理和维护的软件系统。其目的是保证数据安全可靠，提高数据库应用的简明性和方便性。DBMS的工作机理是把用户对数据的操作转化为对系统存储文件的操作，有效地实现数据库三级之间的转化。数据库管理系统的主要职能有：数据库的定义和建立、数据库的操作、数据库的控制、数据库的维护、故障恢复和数据通信。

数据库系统（DBS）是实现有组织地、动态地存储大量关联数据方便多用户访问的计算机软件、硬件和数据资源组成的系统。一个典型的数据库系统包括数据库、硬件、软件（应用程序）和数据库管理员（DBA）四个部分。根据计算机的系统结构，DBS可分成集中式、客户/服务器式、并行式和分布式4种。

与文件系统阶段相比，数据库技术的数据管理方式具有以下特点。

采用复杂的数据模型表示数据结构，数据冗余小，易扩充，实现了数据共享。

具有较高的数据和程序独立性，数据库的独立性有物理独立性和逻辑独立性。

数据库系统为用户提供了方便的用户接口。

数据库系统提供四个方面的数据控制功能，分别是并发控制、恢复、完整性和安全性。数据库中各个应用程序所使用的数据由数据库系统统一规定， 按照一定的数据模型组织和建立，由系统统一管理和集中控制。

增加了系统的灵活性。

高级数据库技术阶段的主要标志是分布式数据库系统和面向对象数据库系统的出现。

集中式系统的弱点是随着数据量的增加，系统相当庞大，操作复杂，开销大，而且因为数据集中存储，大量的通信都要通过主机，造成拥挤。分布式数据库系统的主要特点是数据在物理上分散

存储，在逻辑上是统一的。分布式数据库系统的多数处理就地完成，各地的计算机由数据通信网络相联系。

面向对象数据库系统是面向对象的程序设计技术与数据库技术相结合的产物。面向对象数据库系统的主要特点是具有面向对象技术的封装性和继承性，提高了软件的可重用性。

从目前的数据库系统来看，主要存在以下缺点：

采用静态数据模型，数据类型和操作简单、固定，只能处理短寿命事务；

不能适应计算机辅助设计、计算机辅助软件工程、图像处理、超文本、多媒体等新的应用。

数据库的未来发展趋势如下：

分布式数据管理；

支持面向对象的数据模型；

体系结构适应功能扩展，能处理复杂数据类型和长寿命事务，能和以前数据库共存；

数据库技术与其他学科相结合（分布式数据库、并行数据库、多媒体数据库、Internet数据库、知识库、演绎数据库、主动数据库）。



上一节　本书简介　下一节

第 5 章：数据库系统　作者：希赛教育软考学院　来源：希赛网　2014年01月26日

## 数据库模型

### 5.2 数据库模型

数据库系统的设计目标是允许用户逻辑地处理数据，而不必涉及这些数据在计算机中是怎样存放的，在数据组织和用户应用之间提供某种程度的独立性。



上一节　本书简介　下一节

第 5 章：数据库系统　作者：希赛教育软考学院　来源：希赛网　2014年01月26日

## 数据库系统的三级结构

### 5.2.1 数据库系统的三级结构

数据库技术中采用分级的方法，将数据库的结构划分为多个层次。最著名的是美国ANSI/SPARC数据库系统研究组于1975年提出的三级划分法，如图5-1所示。

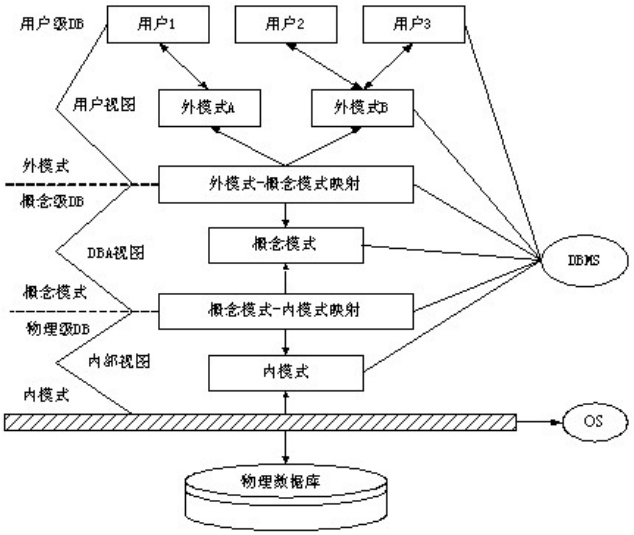


**图5-1 数据库系统结构层次**

数据库系统划分为3个抽象级：用户级、概念级、物理级。

**1.用户级数据库**

用户级数据库对应于外模式，是最接近于用户的一级数据库，是用户看到和使用的数据库，又称用户视图。用户级数据库主要由外部记录组成，不同用户视图可以互相重叠，用户的所有操作都是针对用户视图进行的。

**2.概念级数据库**

概念级数据库对应于概念模式，介于用户级和物理级之间，是所有用户视图的最小并集，是数据库管理员看到和使用的数据库，又称DBA视图。概念级数据库由概念记录组成，一个数据库可有多个不同的用户视图，每个用户视图由数据库某一部分的抽象表示所组成。一个数据库应用系统只存在一个DBA视图，它把数据库作为一个整体的抽象表示。概念级模式把用户视图有机地结合成一个整体，综合平衡考虑所有用户要求，实现数据的一致性，最大限度降低数据冗余，准确地反映数据间的联系。

**3.物理级数据库**

物理级数据库对应于内模式，是数据库的低层表示，它描述数据的实际存储组织，是最接近于物理存储的级，又称内部视图。物理级数据库由内部记录组成，物理级数据库并不是真正的物理存储，而是最接近于物理存储的级。



上一节　本书简介　下一节



作者：希赛教育软考学院　来源：希赛网　2014年01月26日

## 数据库系统的三级模式

### 5.2.2 数据库系统的三级模式

数据库系统的三级模式如图5-1所示。

**1.概念模式**

概念模式（模式、逻辑模式）用以描述整个数据库中数据库的逻辑结构，描述现实世界中的实

体及其性质与联系，定义记录、数据项、数据的完整性约束条件及记录之间的联系，是数据项值的框架。

数据库系统概念模式通常还包含有访问控制、保密定义、完整性检查等方面的内容，以及概念/物理之间的映射。

概念模式是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共数据视图。一个数据库只有一个概念模式。

**2.外模式**

外模式（子模式、用户模式）用以描述用户看到或使用的那部分数据的逻辑结构，用户根据外模式用数据操作语句或应用程序去操作数据库中的数据。外模式主要描述组成用户视图的各个记录的组成、相互关系、数据项的特征、数据的安全性和完整性约束条件。

外模式是数据库用户（包括程序员和最终用户）能够看见和使用的局部数据的逻辑结构和特征的描述，是数据库用户的数据视图，是与某一应用有关的数据的逻辑表示。一个数据库可以有多个外模式。一个应用程序只能使用一个外模式。

**3.内模式**

内模式是整个数据库的最低层表示，不同于物理层，它假设外存是一个无限的线性地址空间。内模式定义的是存储记录的类型、存储域的表示、存储记录的物理顺序，指引元、索引和存储路径等数据的存储组织。

内模式是数据物理结构和存储方式的描述，是数据在数据库内部的表示方式。一个数据库只有一个内模式。

**4.三级模式的关系**

模式是数据库的中心与关键；

内模式依赖于模式，独立于外模式和存储设备；

外模式面向具体的应用，独立于内模式和存储设备；

应用程序依赖于外模式，独立于模式和内模式。



上一节　本书简介　下一节

第 5 章：数据库系统　作者：希赛教育软考学院　来源：希赛网　2014年01月26日

## 数据库系统两级独立性

**5.2.3 数据库系统两级独立性**

数据库系统两级独立性是指物理独立性和逻辑独立性。三个抽象级间通过两级映射（外模式/模式映射，模式/内模式映射）进行相互转换，使得数据库的三级形成一个统一的整体。

**1.物理独立性**

物理独立性是指用户的应用程序与存储在磁盘上的数据库中的数据是相互独立的。当数据的物理存储改变时，应用程序不需要改变。

物理独立性存在于概念模式和内模式之间的映射转换，说明物理组织发生变化时应用程序的独立程度。

**2.逻辑独立性**

逻辑独立性是指用户的应用程序与数据库中的逻辑结构是相互独立的。当数据的逻辑结构改变时，应用程序不需要改变。

逻辑独立性存在于外模式和概念模式之间的映射转换，说明概念模式发生变化时应用程序的独立程度。

值得注意的是：逻辑独立性比物理独立性更难实现。



上一节　本书简介　下一节

第 5 章：数据库系统　作者：希赛教育软考学院　来源：希赛网　2014年01月26日

## 数据模型

**5.3 数据模型**

本节将介绍数据模型。



上一节　本书简介　下一节

第 5 章：数据库系统　作者：希赛教育软考学院　来源：希赛网　2014年01月26日

## 数据模型的分类

**5.3.1 数据模型的分类**

数据模型主要有两大类，分别是概念数据模型（实体联系模型）和基本数据模型（结构数据模型）。

概念数据模型是按照用户的观点来对数据和信息建模，主要用于数据库设计。概念模型主要用实体联系方法（Entity-Reationship Approach）表示，所以也称ER模型。

基本数据模型是按照计算机系统的观点对数据和信息建模，主要用于DBMS的实现。基本数据模型是数据库系统的核心和基础。基本数据模型通常由数据结构、数据操作和完整性约束三部分组成。其中数据结构是对系统静态特性的描述，数据操作是对系统动态特性的描述，完整性约束是一组完整性规则的集合。

常用的基本数据模型有层次模型、网状模型、关系模型和面向对象模型。

层次模型用树型结构表示实体类型及实体间联系。层次模型的优点是记录之间的联系通过指针来实现，查询效率较高。层次模型的缺点是只能表示1:n联系，虽然有多种辅助手段实现m:n联系，但较复杂，用户不易掌握。由于层次顺序的严格和复杂，使得数据的查询和更新操作很复杂，应用程序的编写也比较复杂。

网状模型用有向图表示实体类型及实体间联系。网状模型的优点是记录之间的联系通过指针实现，m:n联系也容易实现，查询效率高。其缺点是编写应用程序比较复杂，程序员必须熟悉数据库的

逻辑结构。

关系模型用表格结构表达实体集，用外键表示实体间联系，其优点有：

建立在严格的数学概念基础上；

概念单一（关系），结构简单、清晰，用户易懂易用；

存取路径对用户透明，从而数据独立性、安全性好，简化数据库开发工作。

关系模型的缺点主要是由于存取路径透明，查询效率往往不如非关系数据模型。







# 关系模型

### 5.3.2 关系模型

先学习几个基本概念。

域：一组具有相同数据类型的值的集合。

笛卡儿积：给定一组域D1,D2,…，Dn,其中可以有相同的域。D1,D2,…，Dn的笛卡儿积为：

D1×D2×…×Dn={（d1,d2,…，dn）|dj∈Dj,j=1,2,…，n}

其中每一个元素（d1,d2,…，dn）叫做一个n元组（简称为元组）。元组中的每一个值dj叫做一个分量。

关系：D1×D2×…×Dn的子集叫做在域D1,D2,…，Dn上的关系，表示为：

R（D1,D2,…，Dn）

这里R表示关系的名字，n是关系的目或度。

关系中的每个元素是关系中的元组，通常用t表示。关系是笛卡儿积的子集，所以关系也是一个二维表，表的每行对应一个元组，表的每列对应一个域。由于域可以相同，为了加以区分，必须为每列起一个名字，称为属性。

若关系中的某一属性组的值能唯一地标识一个元组，则称该属性组为候选码（候选键）。若一个关系有多个候选码，则选定其中一个为主码（主键）。主码的诸属性称为主属性。不包含在任何候选码中的属性称为非码属性（非主属性）。在最简单的情况下，候选码只包含一个属性。在最极端的情况下，关系模式的所有属性组是这个关系模式的候选码，称为全码。

关系可以有3种类型：基本关系（通常又称为基本表或基表）、查询表和视图表。基本表是实际存在的表，它是实际存储数据的逻辑表示。查询表是查询结果对应的表。视图表是由基本表或其他视图表导出的表，是虚表，不对应实际存储的数据。

基本关系具有以下6条性质。

列是同质的，即每一列中的分量是同一类型的数据，来自同一个域。

不同的列可出自同一个域，称其中的每一列为一个属性，不同的属性要给予不同的属性名。

列的顺序无所谓，即列的次序可以任意交换。

任意两个元组不能完全相同。但在大多数实际关系数据库产品中，例如Orace等，如果用户没有定义有关的约束条件，它们都允许关系表中存在两个完全相同的元组。

行的顺序无所谓，即行的次序可以任意交换。

分量必须取原子值，即每一个分量都必须是不可分的数据项。

关系的描述称为关系模式。一个关系模式应当是一个五元组。它可以形式化地表示为：R（U,D,DOM,F）。其中R为关系名，U为组成该关系的属性名集合，D为属性组U中属性所来自的域，DOM为属性向域的映像集合，F为属性间数据的依赖关系集合。关系模式通常可以简记为：R（A1,A2,…，An）。其中R为关系名，A1,A2,…，An为属性名。

关系实际上就是关系模式在某一时刻的状态或内容。也就是说，关系模式是型，关系是它的值。关系模式是静态的、稳定的，而关系是动态的、随时间不断变化的，因为关系操作在不断地更新着数据库中的数据。但在实际当中，常常把关系模式和关系统称为关系，读者可以从上下文中加以区别。

在关系模型中，实体及实体间的联系都是用关系来表示的。在一个给定的现实世界领域中，相应于所有实体及实体之间的联系的关系集合构成一个关系数据库。

关系数据库也有型和值之分。关系数据库的型也称为关系数据库模式，是对关系数据库的描述，是关系模式的集合。关系数据库的值也称为关系数据库，是关系的集合。关系数据库模式与关系数据库通常统称为关系数据库。





第 5 章：数据库系统　作者：希赛教育软考学院　来源：希赛网　2014年08月29日

# 关系规范化理论

### 5.3.3 关系规范化理论

**1.关系模式的存储异常问题**

设有一个关系模式R（SNAME,CNAME,TNAME TADDRESS），其属性分别表示学生姓名、选修的课程名、任课教师姓名和任课教师地址。仔细分析，这个模式存在着下列存储异常的问题：

数据冗余：如果某门课程有100个学生选修，那么在R的关系中就要出现100个元组，这门课程的任课教师姓名和地址也随之重复出现100次。

修改异常：由于上述冗余问题，当需要修改这个教师的地址时，就要修改100个元组中的地址值，否则就会出现地址值不一致的现象。

插入异常：如果不知道听课学生名单，则这个教师的任课情况和家庭地址就无法进入数据库；否则就要在学生姓名处插入空值。

删除异常：如果某门课程的任课教师要更改，那么原来任课教师的地址将随之丢失。

因此，模式R虽然只有4个属性，但却是性能很差的模式。如果把R分解成下列两个关系模式：R1（SNAME,CNAME）和R2（CNAME,TNAME,TADDRESS），则能消除上述提出的存储异常现象。

为什么会产生这些异常呢？与关系模式属性值之间的联系直接有关。在模式R中，学生与课程有直接联系，教师与课程有直接联系，而教师与学生无直接联系，这就产生了模式R的存储异常。因此，模式设计强调"每个联系单独表达"是一条重要的设计原则，把R分解成R1和R2是符合这条原则

的。

**2.函数依赖**

设R（U）是属性U上的一个关系模式，X和Y是U的子集，r为R的任一关系，如果对于r中的任意两个元组u,v,只要有u[X]=v[X],就有u[Y]=v[Y],则称X函数决定Y,或称Y函数依赖于X,记为X→Y。

从函数依赖的定义可以看出，如果有X→U在关系模式R（U）上成立，并且不存在X的任一真子集X′使X′→U成立，那么称X是R的一个候选键。也就是X值唯一决定关系中的元组。由此可见，函数依赖是键概念的推广，键是一种特殊的函数依赖。

在R（U）中，如果X→Y,并且对于X的任何一个真子集X′，都有X′→Y不成立，则称Y对X完全函数依赖。若X→Y,但Y不完全函数依赖于X,则称Y对X部分函数依赖。

在R（U）中，如果X→Y（Y不是X的真子集），且Y→X不成立，Y→Z，则称Z对X传递函数依赖。

设U是关系模式R的属性集，F是R上成立的只涉及到U中属性的FD集，则有以下3条推理规则。

自反性：若属性集Y包含于属性集X，属性集X包含于U，则X→Y在R上成立；

增广性：若X→Y在R上成立，且属性集Z包含于属性集U，则XZ→YZ在R上成立；

传递性：若X→Y和Y→Z在R上成立，则X→Z在R上成立。

这里XZ，YZ等写法表示X∪Z，Y∪Z，上述3条推理规则是函数依赖的一个正确的和完备的推理系统。根据上述3条规则还可以推出其他3条常用的推理规则。

并规则：若X→Y和X→Z在R上成立，则X→YZ在R上成立；

分解规则：若X→Y在R上成立，且属性集Z包含于X，则X→Z在R上成立；

伪传递规则：若X→Y和WY→Z在R上成立，则WX→Z在R上成立。

在关系模式R（U,F）中被F逻辑蕴含的函数依赖全体叫做F的闭包，记做$F^+$。

设F为属性集U上的一组函数依赖，X是U的子集，那么相对于F属性集X的闭包用$X^+$表示，它是一个从F集使用推理规则推出的所有满足X→A的属性A的集合：

$$X^+=\{属性A|X→A在F^+中\}$$

如果$G^+=F^+$,就说函数依赖集F覆盖G（F是G的覆盖，或G是F的覆盖），或F与G等价。

如果函数依赖集F满足下列条件，则称F为一个极小函数依赖集，也称为最小依赖集或最小覆盖。

F中任一函数依赖的右部仅含有一个属性；

F中不存在这样的函数依赖X→A，使得F与F–{X→A}等价；

F中不存在这样的函数依赖X→A，X有真子集Z使得F-{X→A}∪{Z→A}与F等价。

**3.范式**

第一范式（1NF）：如果关系模式R的每个关系r的属性值都是不可分的原子值，那么称R是第一范式的模式，r是规范化的关系。关系数据库研究的关系都是规范化的关系。

第二范式（2NF）：若关系模式R是1NF，且每个非主属性完全函数依赖于候选键，那么称R是2NF模式。

第三范式（3NF）：如果关系模式R是1NF，且每个非主属性都不传递依赖于R的候选码，则称R是3NF。

BC范式（BCNF）：若关系模式R是1NF，且每个属性都不传递依赖于R的候选键，那么称R是BCNF模式。

上述4种范式之间有如下联系：$1NF \supset 2NF \supset 3NF \supset BCNF$。

**4.关系模式分解**

如果某关系模式存在存储异常问题，则可通过分解该关系模式来解决问题。把一个关系模式分解成几个子关系模式，需要考虑的是该分解是否保持函数依赖，是否是无损连接。

无损连接分解的形式定义如下：设R是一个关系模式，F是R上的一个函数依赖（FD）集。R分解成数据库模式δ={R1,……，Rk}。如果对R中每一个满足F的关系r都有下式成立：

$$r = \pi_{R1}(r) \bowtie \pi_{R2}(r) \bowtie \cdots \bowtie \pi_{Rk}(r)$$

那么称分解δ相对于F是无损连接分解，否则称为损失连接分解。

下面是一个很有用的无损连接分解判定定理。

设ρ={R1,R2}是R的一个分解，F是R上的FD集，那么分解ρ相对于F是无损分解的充分必要条件是(R1∩R2)→(R1-R2)或(R1∩R2)→(R2-R1)。

设数据库模式δ={R1,……，Rk}是关系模式R的一个分解，F是R上的FD集，δ中每个模式Ri上的FD集是Fi。如果{F1,F2,…，Fk}与F是等价的（即相互逻辑蕴涵），那么我们称分解δ保持FD。如果分解不能保持FD，那么δ的实例上的值就可能有违反FD的现象。



上一节　本书简介　下一节

第 5 章：数据库系统　作者：希赛教育软考学院　来源：希赛网　2014年01月26日

## 数据操作

**5.4 数据操作**

在关系数据库中，数据操作主要包括查询和更新两大类。关系数据语言有关系代数语言，关系演算语言、具有关系代数和关系演算双重特点的语言三种。其中关系演算语言又包括元组关系演算语言和域关系演算语言。



上一节　本书简介　下一节

第 5 章：数据库系统　作者：希赛教育软考学院　来源：希赛网　2014年08月28日

## 集合运算

**5.4.1 集合运算**

传统的集合运算是二目运算，包括并、交、差、广义笛卡儿积四种运算。

**1.并**

设关系R和S具有相同的关系模式，R和S的并是由属于R或属于S的元组组成的集合，记为RS.形式定义如下：

式中t是元组变量（下同）。显然，R∪S=S∪R.

**2.差**

关系R和S具有相同的关系模式，R和S的差是由属于R但不属于S的元组组成的集合，记为R–S.形式定义如下：

**3.交**

关系R和S具有相同的关系模式，R和S的交是由既属于R又属于S的元组组成的集合，记为R∩S.形式定义如下：

显然，R∩S =R–（R–S），或者R∩S = S–（S–R）。

**4.笛卡儿积**

设关系R和S元数分别为r和s.R和S的笛卡儿积是一个r+s元的元组集合，每个元组的前r个分量来自R的一个元组，后 s个分量来自S的一个元组，记为R×S.形成定义如下：

若R有m个元组，S有n个元组，则R×S有m×n个元组。

**5.集合运算实例**

例如，设关系R和S如表5-1所示。则R∪S与R∩S如表5-2所示，R–S和S–R如表5-3所示，R×S如表5-4所示。

**表5-1 关系R和S**

| *R* 关系 | | | | *S* 关系 | | |
|---|---|---|---|---|---|---|
| *A*1 | *A*2 | *A*3 | | *A*1 | *A*2 | *A*3 |
| a | b | c | | a | b | a |
| b | a | d | | b | a | d |
| c | d | d | | c | d | d |
| d | f | g | | d | s | c |

**表5-2 R∪S与R∩S**

| *R*∪*S* | | | | *R*∩*S* | | |
|---|---|---|---|---|---|---|
| *A*1 | *A*2 | *A*3 | | *A*1 | *A*2 | *A*3 |
| a | b | c | | b | a | d |
| b | a | d | | c | d | d |
| c | d | d | | | | |
| d | f | g | | | | |
| a | b | a | | | | |
| d | s | c | | | | |

**表5-3 R–S和S–R**

| *R* - *S* | | | | *S* - *R* | | |
|---|---|---|---|---|---|---|
| *A*1 | *A*2 | *A*3 | | *A*1 | *A*2 | *A*3 |
| a | b | c | | a | b | a |
| d | f | g | | d | s | c |

**表5-4 R×S**

| A1 | A2 | A3 | A1 | A2 | A3 |
|---|---|---|---|---|---|
| a | b | c | a | b | a |
| b | a | d | a | b | a |
| c | d | d | a | b | a |
| d | f | g | a | b | a |
| a | b | c | b | a | d |
| b | a | d | b | a | d |
| c | d | d | b | a | d |
| d | f | g | b | a | d |
| a | b | c | c | d | d |
| b | a | d | c | d | d |
| c | d | d | c | d | d |
| d | f | g | c | d | d |
| a | b | c | D | s | c |
| b | a | d | d | s | c |
| c | d | d | d | s | c |
| d | f | g | d | s | c |





第 5 章：数据库系统　作者：希赛教育软考学院　来源：希赛网　2014年01月26日

# 关系运算

### 5.4.2 关系运算

在5.4.1节的集合运算基础上，关系数据库还有一些专门的运算，主要有投影、选择、连接、除法和外连接。它们是关系代数最基本的操作，也是一个完备的操作集。在关系代数中，由五种基本代数操作经过有限次复合的式子称为关系代数运算表达式。表达式的运算结果仍是一个关系。我们可以用关系代数表达式表示各种数据查询和更新处理操作。

**1.投影**

投影操作从关系R中选择出若干属性列组成新的关系，该操作对关系进行垂直分割，消去某些列，并重新安排列的顺序，再删去重复元组。记为：

$$\pi_A(R) \equiv \{t[A] \mid t \in R\}$$

其中A为R的属性列。

**2.选择**

选择操作在关系R中选择满足给定条件的所有元组，记为：

$$\sigma_F(R) \equiv \{t \mid t \in R \wedge F(t) = true\}$$

其中F表示选择条件，是一个逻辑表达式（逻辑运算符+算术表达式 ）。选择运算是从行的角度进行的运算。

**3.θ连接**

θ连接从两个关系的笛卡儿积中选取属性间满足一定条件的元组，记为：

$$R \underset{A\,\theta\,B}{\triangleright\triangleleft} S \equiv \{t_r t_s \mid t_r \in R \wedge t_s \in S \wedge t_r[A]\theta t_s[B]\}$$

其中A和B分别为R和S上度数相等且可比的属性组。θ为"="的连接，称为等值连接，记为：

$$R \underset{A=B}{\triangleright\triangleleft} S \equiv \{t_r t_s \mid t_r \in R \wedge t_s \in S \wedge t_r[A] = t_s[B]\}$$

如果两个关系中进行比较的分量必须是相同的属性组，并且在结果中把重复的属性列去掉，则

称为自然连接，记为：

$$R \bowtie S \equiv \{t_r t_s \mid t_r \in R \wedge t_s \in S \wedge t_r[A] = t_s[B]\}$$

**4.除法**

设两个关系R和S的元数分别为r和s（设r>s>0），那么R÷S是一个（r–s）元的元组的集合。R÷S是满足下列条件的最大关系：其中每个元组t与S中每个元组u组成新元组<t,u>必在关系R中。其具体计算公式如下：

$$R \div S = \pi_{1,2,\cdots,r-s}(R) - \pi_{1,2,\cdots,r-s}((\pi_{1,2,\cdots,r-s}(R) \times S) - R)$$

**5.外连接**

两个关系R和S进行自然连接时，选择两个关系R和S公共属性上相等的元组，去掉重复的属性列构成新关系。这样，关系R中的某些元组有可能在关系S中不存在公共属性值上相等的元组，造成关系R中这些元组的值在运算时舍弃了；同样关系S中的某些元组也可能舍弃。为此，扩充了关系运算左外连接、右外连接和完全外连接。

左外连接：R和S进行自然连接时，只把R中舍弃的元组放到新关系中。

右外连接：R和S进行自然连接时，只把S中舍弃的元组放到新关系中。

完全外连接：R和S进行自然连接时，只把R和S中舍弃的元组都放到新关系中。

**6.关系运算实例**

设两个关系模式R和S如表5-5所示，$\pi_{1,2}(R)$则的结果如表5-6所示，$\sigma_{1>2}(R)$的结果如表5-7所示，$R \bowtie S$的结果如表5-8所示，R与S的左外连接如表5-9所示，R与S的右外连接如表5-10所示，R与S的完全外连接如表5-11所示。

**表5-5 关系R和S**

| R关系 | | | | S关系 | | |
|---|---|---|---|---|---|---|
| A1 | A2 | A3 | | A1 | A2 | A4 |
| a | b | c | | a | z | a |
| b | a | d | | b | a | h |
| c | d | d | | c | d | d |
| d | f | g | | d | s | c |

**表5-6 对关系R求投影操作**

| A1 | A2 |
|---|---|
| a | b |
| b | a |
| c | d |
| d | f |

**表5-7 对关系R求选择操作**

| A1 | A2 | A3 |
|---|---|---|
| b | a | D |

**表5-8 对关系R和S的自然连接**

| A1 | A2 | A3 | A4 |
|---|---|---|---|
| B | a | d | h |
| C | d | d | d |

**表5-9 R与S的左外连接**

| A1 | A2 | A3 | A4 |
|---|---|---|---|
| A | b | c | null |
| B | a | d | h |
| C | d | d | d |
| D | f | g | null |

表5-10 R与S的右外连接

| A1 | A2 | A3 | A4 |
|---|---|---|---|
| A | z | null | a |
| B | a | d | h |
| C | d | d | d |
| D | s | null | c |

表5-11 R与S的完全外连接

| A1 | A2 | A3 | A4 |
|---|---|---|---|
| A | b | c | null |
| B | a | d | h |
| C | d | d | d |
| D | f | g | null |
| A | z | null | a |
| D | s | null | c |



上一节 本书简介 下一节



# 数据库语言

## 5.5 数据库语言

我们简单地介绍标准化数据库查询语言SQ.

SQ语言由Boyce和Chamberin于1974年提出，1975年-1979年，IBM San Jose Research ab的关系数据库管理系统原型System R实施了这种语言。SQ-86是第一个SQ标准，后续的有SQ-89、SQ-92（SQ2）、SQ-99（SQ3）等。现在大部分DBMS产品都支持SQ,但每个产品在具体使用时又有方言，支持程度不同。

SQ的特点主要体现在以下几个方面。

集数据定义语言、数据操纵语言、数据控制语言的功能于一体，语言风格统一。

存取路径的选择及SQ语句的操作过程由系统自动完成，减轻了用户负担，提高了数据独立性。

采用集合的操作方式。

既是自含式语言（联机交互），又是嵌入式语言（宿主语言）。

语言简捷，易学易用。只有10个动词（SEECT,CREATE,DROP,ATER,INSERT,UPDATE,DEETE,GRANT,REVOKE,MODIFY）。



上一节 本书简介 下一节



# 数据定义

**5.5.1 数据定义**

**1.定义基本表**

SQ语言使用动词CREATE定义基本表，其具体语法格式如下：

CREATE TABE <表名>

（ <列名><数据类型>[列级完整性约束条件][,

<列名><数据类型>[列级完整性约束条件]][,

<表级完整性约束条件>] ）；

例如，建立一个学生表student,它由学号Sno、姓名Sname、性别Ssex、年龄Sage、所在系Sdept5个属性组成。其中学号不能为空，值是唯一的，并且姓名取值也唯一。

CREATE TABE Student

（ Sno CHAR（ 5 ） NOT NU UNIQUE,

Sname CHAR（ 20 ） UNIQUE,

Ssex CHAR（ 2 ），

Sage INT,

Sdept CHAR（ 15 ））；

**2.修改基本表**

修改基本表的命令格式如下：

ATER TABE <表名>

[ADD <新列名><数据类型>[完整性约束]]

[DROP <完整性约束名>]

[MODIFY <列名><数据类型>];

例如，向Student表增加"入学时间"列，其数据类型为日期型。SQ命令如下：

ATER TABE Student Add Scome Date;

**3.删除基本表**

DROP TABE <表名>

例如，要删除Student表的命令为：

DROP TABE Student;

注意：基本表一旦删除，表中的数据、表上建立的索引和视图都将自动被删除。

**4.建立索引**

建立索引的命令格式如下：

CREATE [Unique][Custer]INDEX <索引名>

ON <表名>（ <列名>[<次序>][,<列名>[<次序>]]... ）；

其中<次序>可以为ASC（ 升序，默认 ）、DESC（ 降序 ）。

Unique:每一个索引值只对应唯一的数据记录。

Custer:聚簇索引，即索引项的顺序与表中记录的物理顺序一致。

例如，要在Student表的Sname列上建立一个聚簇索引，并按升序排列的命令为：

CREATE Custer INDEX Stuname ON Student（ Sname ）；

**5.删除索引**

删除索引的SQ命令格式如下：

DROP INDEX <索引名>

例如，要删除Student表的索引Stuname的命令为：

DROP INDEX Stuname;







## 数据查询

### 5.5.2 数据查询

在SQ语言中，只提供了一个动词SEECT用来进行数据查询操作，但这个动词的参数十分复杂，且能嵌套使用，所以，考试时往往就考这个功能。其通用格式如下：

SEECT [A | Distinct] <目标列表达式>[,<目标列表达式>]…

FROM <表名或视图名>[, <表名或视图名>]…

[WHERE <条件表达式>]

[GROUP BY <列名1>[HAVING <条件表达式>]]

[ORDER BY <列名2>[ASC | DESC]];

**1.单表查询**

下面，我们主要通过一些例子来说明SEECT语句的使用。假设有上述的student表，还有课程表course（cno,cname,credit,cpno）和选修表sc（sno,cno,grade）。其中cno为课程号，cname为课程名称，cpno为先修课程号，credit为学分，grade为成绩。

查询全体学生的学号与姓名的命令格式为：

SEECT Sno,Sname

FROM student;

查询全体男学生的详细记录的命令格式为：

SEECT *

FROM student

WHERE Ssex='男';

查询所有年龄大于21岁的学生的姓名、出生年份和所有系，要求用小写字母表示所有系名。其格式如下：

SEECT Sname, 'Year of Birth:',2004 – Sage,ower（Sdept）

FROM student

WHERE Sage>21;

查询IS系、MA系和CS系学生的姓名和性别的命令格式为：

SEECT Sname,Sex

FROM student

WHERE Sdept In（'IS','MA','CS'）；

查询名字中第2个字为"阳"的学生的姓名、学号的命令格式为：

SEECT Sname,Sno

FROM student

WHERE Sname IKE '_ _阳%';

其中的"_"代表一个字符，而"%"代表0到若干个字符。

查询DB_Design课程的课程号和学分的命令格式为：

SEECT Cno,credit

FROM Course

WHERE Cname IKE 'DB_Design' Escape '\';

查询选修了3号课程的学生的学号及成绩，查询结果按分数的降序排列所有有成绩的学生学号和课程号。

SEECT Sno,Grade

FROM SC

WHERE Cno='3'

ORDER BY Grade DESC;

在SQ语言中，也可以使用集函数：

Count（[Distinct|A]*）：统计元组个数；

Count（[Distinct|A]<列名>）：统计一列中值的个数；

Sum（[Distinct|A]<列名>）：计算一列值的总和；

Avg（[Distinct|A]<列名>）：计算一列值的平均值；

max（[Distinct|A]<列名>）：求一列值中的最大值；

Min（[Distinct|A]<列名>）：求一列值中的最小值。

求各个课程号及相应的选课人数。

SEECT Cno,Count（Sno）

FROM SC

GROUP BY Cno;

**2.连接查询**

查询每个学生及其选修课程的情况。

SEECT Student.*,SC.*

FROM Student,SC

WHERE Student.Sno= SC.Sno;

查询每一门课程的间接选修课。

SEECT F.Cno,S.Cpno

FROM Course F,Course S

WHERE F.Cpno = S.Cno;

其中的F和S称为course的别名。

查询每个学生及其选修课程的情况。

SEECT Student.Sno,Sname,Ssex,Sage,Cno,Grade

FROM Student eft Outer Join SC

ON Student.Sno = SC.Sno;

查询每个学生的学号、姓名、选修的课程名称及成绩。

```
SEECT Student.Sno,Sname,Cname,Grade
FROM Student,SC,Course
WHERE Student.Sno=SC.Sno And SC.Cno=Course.Cno;
```

**3.嵌套查询**

查询与"刘晨"在同一系学习的学生。

```
SEECT Sno,Sname
FROM Student
WHERE Sdept IN
（SEECT Sdept
FROM Student
WHERE Sname='刘晨'）；
```

查询选修了课程名为信息系统（MIS）的学生学号和姓名。

```
SEECT Sno,Sname
FROM Student
WHERE Sno IN
（Seect Sno
FROM SC
WHERE Cno IN
（SEECT Cno
FROM Course
WHERE Cname='MIS'））；
```

查询其他系中比信息系某一个学生年龄小的学生姓名和年龄。

```
SEECT Sname,Sage
FROM Student
WHERE Sage < Any
（SEECT Sage
FROM Student
WHERE Sdept='IS'）；
```

或者：

```
SEECT Sname,Sage
FROM Student
WHERE Sage <
（SEECT Max（Sage）
FROM Student
WHERE Sdept='IS'）
AND Sdept <> 'IS';
```

查询没有选修1号课程的学生姓名。

```
SEECT Sname
```

FROM Student

WHERE Not Exists

（SEECT *

FROM SC

WHERE Sno=Student.Sno And Cno='1'）；

查询至少选修了95002选修表的全部课程的学生的学号。

SEECT Distinct Sno

FROM SC SCX

WHERE Not Exists

（SEECT *

FROM SC SCY

WHERE SCY.Sno='95002'

And Not Exists

（SEECT *

FROM SC SCZ

WHERE SCZ.Sno=SCX.Sno And SCZ.Cno=SCY.Cno））；

4.集合查询

例如，要查询计算机系的学生及年龄不大于19岁的学生的命令格式为：

SEECT *

FROM Student

WHERE Sdept='CS'

UNION

SEECT *

FROM Student

WHERE Sage<=19;



上一节　本书简介　下一节



## 数据更新

### 5.5.3 数据更新

**1.插入数据**

插入单个元组的命令格式为：

INSERT INTO <表名>[（<属性列1>[,<属性列2>…]）

VAUES（<常量1>[,<常量2>]…）

例如，将一个新学生记录（95020,陈冬，男，IS,18）插入到Student表中。

INSERT INTO Student

VAUES（'95020', '陈冬', '男', 'IS',18）；

**2.修改数据**

修改数据的命令格式为：

UPDATE <表名>

SET <列名1>=<表达式1>[,<列名2>=<表达式2>]...

[WHERE <条件>]

例如，将学生95001的年龄改为22岁。

UPDATE Student

SET Sage=22

WHERE Sno='95001';

**3.删除数据**

删除表中数据的命令格式为：

DEETE FROM <表名>

[WHERE <条件>]

例如，删除学号为95019的学生记录为：

DEETE FROM Student

WHERE Sno='95019';



上一节 本书简介 下一节



## 视图

### 5.5.4 视图

视图不真正存在数据，只是把定义存于数据字典，在对视图进行查询时，才按视图的定义从基本表中将数据查出。若一个视图是从单个基本表导出的，并且只是去掉了基本表的某些行和某些列，但保留了码，则这个视图称为行列子集视图。

在DBMS中，视图的作用如下：

简化用户的操作；

使用户能从多种角度看待同一数据；

对重构数据库提供了一定程度的逻辑独立性；

能够对机密数据提供安全保护。

**1.定义视图**

建立视图的命令格式如下：

CREATE VIEW <视图名>[（<列名>[,<列名>]...）]

AS

子查询

[With Check Option]

其中With Check Option表示对视图进行Update、Insert和Deete操作时，要保证更新、插入或删除的行满足视图定义中的谓词条件。

例如，建立信息系学生的视图：

```
CREATE VIEW IS_Student
AS
SEECT Sno,Sname,Sage
FROM Student
WHERE Sdept='IS'
With Check Option;
```

**2.删除视图**

删除视图的命令格式为：

```
DROP 视图名
```

例如，要删除视图IS_S1:

```
DROP VIEW IS_S1;
```

**3.查询视图**

因为视图没有真实数据，所以，对视图的查询要转换为对相应表的查询，这个过程叫视图消解，视图消解过程由DBMS自动完成。

例如，在信息系学生的视图中找出年龄小于20岁的学生：

```
SEECT Sno,Sage
FROM IS_Student
WHERE Sage<20;
```

以上语句等价于：

```
SEECT Sno,Sage
FROM Student
WHERE Sage<20 And Sdept='IS';
```

**4.更新视图**

更新视图就是对相应表的更新。例如，将信息系学生视图IS_Student中学号为95002的学生姓名改为"刘辰":

```
UPDATE  IS_Student
SET Sname='刘辰'
WHERE Sno='95002';
```

以上语句等价于：

```
UPDATE Student
SET Sname='刘辰'
WHERE Sno='95002' And Sdept='IS';
```

# 数据控制

### 5.5.5 数据控制

**1.授权**

授权的命令格式如下：

GRANT <权限>[,<权限>]...

[ON <对象类型><对象名>]

TO <用户>[,<用户>]... [With Grant Option]

例如，把对Student表和Course表的全部操作权授予用户U2和U3:

GRANT A Privieges

ON Tabe Student,Course

TO U2,U3;

又如，把对表SC的Insert权限授予U5用户，并允许U5将此权限再授予其他用户：

GRANT Insert

ON Tabe SC

TO U5 With Grant Option;

**2.收回授权**

收回授权的命令格式如下：

REVOKE <权限>[,<权限>]...

[ON <对象类型><对象名>]

FROM <用户>[,<用户>]...

例如，把用户U4修改学生学号的权限收回：

REVOKE Update（Sno），Seect

ON Tabe Student

FROM U4;







# 数据库的控制功能

### 5.6 数据库的控制功能

本节将介绍数据库的控制功能。

# 并发控制

### 5.6.1 并发控制

数据库管理系统运行的基本工作单位是事务，事务是用户定义的一个数据库操作序列，这些操作序列要么全做，要么全不做，是一个不可分割的工作单位。事务具有以下特性。

原子性（Atomicity）：数据库的逻辑工作单位。

一致性（Consistency）：使数据库从一个一致性状态变到另一个一致性状态。

隔离性（Isoation）：不能被其他事务干扰。

持续性（永久性）（Durabiity）：一旦提交，改变就是永久性的。

事务通常以BEGIN TRANSACTION（事务开始）语句开始，以COMMIT或ROBACK语句结束。COMMIT称为"事务提交语句",表示事务执行成功地结束。ROBACK称为"事务回退语句",表示事务执行不成功地结束。从终端用户来看，事务是一个原子，是不可分割的操作序列。事务中包括的所有操作要么都做，要么都不做（就效果而言）。事务不应该丢失或被分割完成。

在多用户共享系统中，许多事务可能同时对同一数据进行操作，称为"并发操作",此时数据库管理系统的并发控制子系统负责协调并发事务的执行，保证数据库的完整性不受破坏，同时避免用户得到不正确的数据。

数据库的并发操作带来的问题：丢失更新问题，不一致分析问题（读过时的数据），依赖于未提交更新的问题（读了"脏"数据）。这三个问题需要DBMS的并发控制子系统来解决。

处理并发控制的主要方法是采用封锁技术。有两种封锁，X封锁和S封锁。

排他型封锁（简称X封锁）：其含义是如果事务T对数据A（可以是数据项、记录、数据集以至整个数据库）实现了X封锁，那么只允许事务T读取和修改数据A,其他事务要等事务T解除X封锁以后，才能对数据A实现任何类型的封锁。可见X封锁只允许一个事务独锁某个数据，具有排他性。

共享型封锁（简称S封锁）：X封锁只允许一个事务独锁和使用数据，要求太严。需要适当从宽，例如可以允许并发读，但不允许修改，这就产生了S封锁概念。S封锁的含义是如果事务T对数据A实现了S封锁，那么允许事务T读取数据A,但不能修改数据A,在所有S封锁解除之前决不允许任何事务对数据A实现X封锁。

在多个事务并发执行的系统中，主要采取封锁协议来进行处理。

一级封锁协议：事务T在修改数据R之前必须先对其加X封锁，直到事务结束才释放。一级封锁协议可防止丢失修改，并保证事务T是可恢复的。但不能保证可重复读和不读"脏"数据。

二级封锁协议：一级封锁协议加上事务T在读取数据R之前先对其加S锁，读完后即可释放S锁。二级封锁协议可防止丢失修改，还可防止读"脏"数据。但不能保证可重复读。

三级封锁协议：一级封锁协议加上事务T在读取数据R之前先对其加S锁，直到事务结束才释放。三级封锁协议可防止丢失修改、防止读"脏"数据与防止数据重复读。

两段锁协议：所有事务必须分两个阶段对数据项加锁和解锁。其中扩展阶段是在对任何数据进行读、写操作之前，首先要申请并获得对该数据的封锁；收缩阶段是在释放一个封锁之后，事务不能再申请和获得任何其他封锁。若并发执行的所有事务均遵守两段封锁协议，则对这些事务的任何并发调度策略都是可串行化的。遵守两段封锁协议的事务可能发生死锁。

下面讨论封锁的粒度。所谓封锁的粒度即是被封锁数据目标的大小，在关系数据库中封锁粒度

有属性值、属性值集、元组、关系、某索引项（或整个索引）、整个关系数据库、物理页（块）等几种。

封锁粒度小则并发性高，但开销大。封锁粒度大则并发性低但开销小，综合平衡照顾不同需求以合理选取适当的封锁的粒度是很重要的。

采用封锁的方法固然可以有效防止数据的不一致性，但封锁本身也会产生一些麻烦，最主要的就是"死锁"（deadock）问题。所谓死锁即是多个用户申请不同封锁，由于申请者均拥有一部分封锁权而又需等待另外用户拥有的部分封锁而引起的永无休止的等待。一般讲，死锁是可以避免的，目前采用的办法有如下几种。

预防法：此种方法是采用一定的操作方式以保证避免死锁的出现，顺序申请法、一次申请法等即是此类方法。所谓顺序申请法，即是对封锁对象按序编号，用户申请封锁时必须按编号顺序（从小到大或反之）申请，这样能避免死锁发生。所谓一次申请法，即是用户在一个完整操作过程中必须一次性申请它所需要的所有封锁，并在操作结束后一次性归还所有封锁，这样也能避免死锁的发生。

死锁的解除法：此种方法是允许产生死锁，并在死锁产生后通过解锁程序以解除死锁。使用这种方法需要有两个程序，一个是死锁检测程序，用它来测定死锁是否发生；另一个是解锁程序，一旦经测定系统已产生死锁则启动解锁程序以解除死锁。有关死锁检测及解锁技术请参阅相应的资料，这里不做进一步讨论。





第 5 章：数据库系统　作者：希赛教育软考学院　来源：希赛网　2014年01月26日

## 数据恢复

### 5.6.2 数据恢复

把数据库从错误状态恢复到某一已知的正确状态的功能，称为数据库的恢复。数据库的故障可以分为事务内部的故障、系统故障、介质故障和计算机病毒造成的故障等。

**1.事务内部的故障**

1）可预期的

例如把一笔金额从一个账户转给另一个账户：

Begin Transaction

Baance = Baance – Amount;

if（Baance < 0） Roback;

ese Baance1 = Baance1 + Amount;

Commit;

2）不可预期的

运算溢出、并发事务发生死锁、违反完整性约束等。

**2.系统故障**

系统故障包括硬件错误、操作系统错误、DBMS代码错误和突然停电等。

数据恢复的基本原理就是冗余，建立冗余的方法有数据转储和登录日志文件等。可根据故障的不同类型，采用不同的恢复策略。

1）事务故障的恢复

事务故障的恢复是由系统自动完成的，对用户是透明的，步骤如下。

①反向扫描文件日志，查找该事务的更新操作。

②对该事务的更新操作执行逆操作。

③继续反向扫描日志文件，查找该事务的其他更新操作，并做同样处理。

④如此处理下去，直至读到此事务的开始标记，事务故障恢复完成。

2）系统故障的恢复

系统故障的恢复在重新启动时自动完成，不需要用户干预。步骤如下。

①正向扫描日志文件，找出在故障发生前已经提交的事务，将其事务标识记入重做（Redo）队列。同时找出故障发生时尚未完成的事务，将其事务标识记入撤销（Undo）队列。

②对撤销队列中的各个事务进行撤销处理：反向扫描日志文件，对每个Undo事务的更新操作执行逆操作。

③对重做队列中的各个事务进行重做处理：正向扫描日志文件，对每个Redo事务重新执行日志文件登记的操作。

3）介质故障与病毒破坏的恢复

介质故障与病毒破坏的恢复步骤如下。

①装入最新的数据库后备副本，使数据库恢复到最近一次转储时的一致性状态。

②从故障点开始反向读日志文件，找出已提交事务标识将其记入重做队列。

③从起始点开始正向阅读日志文件，根据重做队列中的记录，重做所有已完成事务，将数据库恢复至故障前某一时刻的一致状态。

4）具有检查点的恢复技术

检查点记录的内容可包括：

建立检查点时刻所有正在执行的事务清单；

这些事务最近一个日志记录的地址。

采用检查点的恢复步骤如下。

①从重新开始文件中找到最后一个检查点记录在日志文件中的地址，由该地址在日志文件中找到最后一个检查点记录。

②由该检查点记录得到检查点建立时所有正在执行的事务清单队列（A）。

③建立重做队列（R）和撤销队列（U），把A队列放入U队列中，R队列为空。

④从检查点开始正向扫描日志文件，若有新开始的事务T1,则把T1放入U队列。若有提交的事务T2,则把T2从U队列移到R队列；直至日志文件结束。

⑤对U队列的每个事务执行Undo操作，对R队列的每个事务执行Redo操作。

### 5.6.3 安全性

在数据库系统中大量数据集中存放，而且多用户共享，系统安全保护措施是否有效是数据库系统主要的性能指标之一。数据库安全模型如图5-2所示。

| | 数据对象 | 操作类型 |
|---|---|---|
| 模式 | 模式<br>外模式<br>内模式 | 建立、修改、检索<br>建立、修改、检索<br>建立、修改、检索 |
| 数据 | 表<br>属性列 | 查找、插入、修改、删除<br>查找、插入、修改、删除 |

**图5-2 数据库安全模型**

**1.用户标识与鉴别**

用户标识和鉴定是系统提供的最外层的安全保护措施。其方法是每次用户要进入系统时由系统提供一定的方式让用户标识自己的名字或身份，系统对用户身份进行鉴定核实后才提供系统使用权，常用的方法有下列几种。

用户名或用户标识号：在定义外模式时为每个用户提供一个用户代号存放在数据字典中。用户使用系统时，系统鉴别此用户是否是合法用户，若是，则可进入下一步的核实，否则不能使用系统。

口令：为了进一步核实用户，系统常常要求用户输入口令。为保密起见，用户在终端上输入的口令不显示在屏幕上，系统核对口令以鉴别用户身份。以上的方法简单易行，但用户名、口令容易被人窃取，因此还可以用更可靠的方法。

随机数检验：用户根据预先约定好的计算公式求出一个数值作为动态口令送入计算机，当这个值与系统算出的结果一致时，才允许进入系统。

用户标识和鉴定可以重复多次。

**2.存取控制**

在数据库系统中，为了保证用户只能存取有权存取的数据，系统要求对每个用户定义存取权限。存取权限包括两方面的内容：一方面是要存取的数据对象；另一方面是对此数据对象进行操作的类型。对一个用户定义存取权限就是要定义这个用户可以在哪些数据对象上进行哪些类型的操作。在数据库系统中对存取权限的定义称为"授权",这些授权定义经过编译后存放在数据库中。对于获得使用权又进一步发出存取数据库操作的用户，系统就根据事先定义好的存取权限进行合法权检查，若用户的操作超出了定义的权限，系统拒绝执行此操作，这就是存取控制。

授权编译程序和合法权检查机制一起组成了安全性子系统。

在非关系系统中，用户只能对数据进行操作，存取控制的数据对象也仅限于数据本身。而关系数据库系统中，DBA可以把建立和修改基本表的权限授予用户，用户可利用这种权限来建立和修改基本表、索引、视图，因此，关系系统中存取控制的数据对象不仅有数据本身，还有模式、外模式、内模式等内容，如表5-12所示。

表5-12 关系系统中的存取权限

关系数据语言SQ除了数据定义和数据操作外，还提供了数据控制的功能，其授权和收回就是通过其提供的GRANT和REVOKE语句来实现的。

**3.视图机制**

视图机制可以将要保密的数据对无权存取这些数据的用户隐藏起来，这样就自动地提供了对数据的安全保护。

**4.审计**

审计是现代计算机系统中必不可少的功能之一，其主要任务是对用户（包括应用程序）使用系统资源（包括软硬件和数据）的情况进行记录和审查，一旦发现问题，审计人员通过审计跟踪，可望找出原因，追查责任，防止类似问题再度发生。因此，审计往往作为保证数据库安全的一种补救措施。

数据库系统中的审计工作包括如下几种。

设备安全审计。主要审查关于系统资源的安全策略、各种安全保护措施及故障恢复计划等。

操作审计。对系统的各种操作（特别是一些敏感操作）进行记录、分析。记录内容包括操作的种类、所属事务、所属进程、用户、终端（或客户机）、操作时间、审计日期等。

应用审计。审计建于数据库之上的整个应用系统的功能、控制逻辑、数据流是否正确。

攻击审计。对已发生的攻击性操作及危害系统安全的事件（或企图）进行检测和审计。

上述各种审计所用技术大致可分为以下3类。

1）静态分析系统技术

审计者通过查阅各种系统资源（软硬件、数据）的说明性文件，例如软件的设计说明书、流程图等来了解整个系统，甚至定位出一些易被攻击的薄弱环节。

2）运行验证技术

运行验证的目的是保证系统控制逻辑正确，各类事务能有效执行。该技术一般又细分为实际运行测试和性能测试两种。实现时，审计者既可根据审计需要，选择系统中-个实际事务作为样板进行审计跟踪，又可生成专门的测试用例，通过将测试用例在系统运行的实际结果与期望结果进行比较来评价系统；还可设计一个专门仿真系统的程序，让仿真系统与实际系统并行工作，比较它们的结果来评测系统。

3）运行结果验证技术

这种技术注意力放在运行结果--数据上。它主要涉及审计数据选择和收集、数据分析两类问题。常用的审计数据选择和收集的办法有：

在应用程序中插入一个审计数据收集模块

设置专门的审计跟踪事务

兼用系统的日志库

使用由随机抽取记录组成的专用审计库

……

一旦获得审计数据后，审计者可以检查各类控制信息、完整性约束等内容，以达到各种审计目的。

5.数据加密

对于那些保密程度极高的数据（如用户标识、绝密信息等）和在网络传输过程中可能被盗窃的数据除采用上述安全保护措施外，一般还需采用数据加密技术，以密文形式保存和传输，保证只有那些知道密钥的用户可以访问。数据加密是防止数据库中的数据在存储和传输中失密的有效手段。有关加密技术的详细内容请参考相关章节，在此不再详细叙述。

还有一种统计数据库安全性，举例说明如下。

例1:

（1）本公司共有多少女高级程序员？

（2）本公司女高级程序员的工资总额是多少？

问题：

如果第（1）个查询的结果为"1",那么第（2）个查询的结果就是该高级程序员的工资。

例2:设某用户A的工资是Z,他想知道用户B的工资。

（1）用户A和其他N个程序员的工资总额是多少？

（2）用户B和其他N个程序员的工资总额是多少？

问题：

如果第（1）个查询的结果为X,第（2）个查询的结果是Y,那么B的工资为Y–X+Z。





第 5 章：数据库系统　作者：希赛教育软考学院　来源：希赛网　2014年01月26日

# 完整性

### 5.6.4 完整性

**1.完整性约束条件**

保证数据库中的数据完整性的方法之一是设置完整性检查，即对数据库中的数据设置一些约束条件，这是数据的语义体现。数据的完整性约束条件一般在数据模式中结出，并在运行时检查，当不满足条件时立即向用户通报以便采取措施，如表5-13所示。

完整性约束条件一般指的是对数据库中数据本身的某些语法、语义限制，数据间的逻辑约束及数据变化时应遵守的规则等。所有这些约束条件一般均以谓词逻辑形式表示，即以具有真假值的原子公式及命题连接词（并且、或者、否定）所组成的逻辑公式表示。完整性约束条件的作用对象可以是关系、元组、列三种。

数据库中数据的语法、语义限制与数据间的逻辑约束称为静态约束。它反映了数据及数据间的固有的逻辑特性，是最重要的一类完整性约束。静态约束包括静态列级约束（对数据类型的约束、对数据格式的约束、对取值范围或取值集合的约束、对空值的约束、其他约束）、静态元组约束、静态关系约束（实体完整性约束、参照完整性约束、函数依赖约束、统计约束）。

数据库中的数据变化应遵守的规则称为数据动态约束，它反映了数据库状态变迁的约束。动态约束包括动态列级约束（修改列定义时的约束、修改列值时的约束）、动态元组约束、动态关系约束。

**表5-13 完整性约束条件**

| 粒度 / 状态 | 数据对象 | 元组级 | 操作类型 |
|---|---|---|---|
| 静态 | 列定义<br>● 类型<br>● 格式<br>● 值域<br>● 空值 | 元组值应<br>满足的条件 | 实体完整性约束<br>参照完整性约束<br>函数依赖约束<br>统计约束 |
| 动态 | 改变列定义或列值 | 元组新旧值之间应满足的约束条件 | 关系新旧状态间应满足的约束条件 |

**2.完整性控制**

**1）完整性控制机制应该具有的功能**

定义功能：提供定义完整性约束条件的机制。

检查功能：检查用户发出的操作请求是否违背了完整性约束条件。

如果发现用户的操作请求违背了约束条件，则采取一定的动作来保证数据的完整性。

如果在一条语句执行完后立即检查，则称立即执行约束；如果在整个事务执行结束后再进行检查，则称延迟执行约束。

2）完整性规则的五元组（D,O,A,C,P）

D:约束作用的数据对象。

O:触发完整性检查的数据库操作。

A:数据对象必须满足的断言或语义约束。

C:选择A作用的数据对象值的谓词。

P:违反完整性规则时触发的过程。

如学号不能为空可表示为：

D:约束作用的数据对象为Sno属性。

O:插入或修改元组时。

A:Sno定义为Not Nu.

C:无。

P:拒绝执行该操作。

**3）参照完整性**

外码能否接受空值问题根据实际应用决定。

在被参照关系中删除元组的问题。

级联删除（Cascades）：将参照关系中所有外码值与被参照关系中要删除元组主码值相同的元组一起删除。如果参照关系同时又是另一个关系的被参照关系，则这种删除操作会继续级联下去。

受限删除（Restrict 默认）：仅当参照关系中没有任何元组的外码值与被参照关系中要删除元组的主码值相同时，系统才可以执行删除操作，否则拒绝执行删除操作。

置空删除（Set Nu）：删除被参照关系的元组时，并将参照关系中相应元组的外码值置为空值。

在参照关系中插入元组的问题。

受限插入：仅当被参照关系中存在相应的元组时，其主码值与参照关系插入元组的外码值相同时，系统才执行插入操作，否则拒绝此操作。

递归插入：首先向被参照关系中插入相应的元组，其主码值等于参照关系插入元组的外码值，然后向参照关系插入元组。

修改关系中主码的问题。

不允许修改主码。

允许修改主码。

**4）触发器**

触发器（trigger）是在关系数据库管理系统中应用得比较多的一种完整性保护措施。触发器的功能一般比完整性约束要强得多，一般而言在完整性约束功能中，当系统检查出数据中有违反完整

性约束条件时，则仅给出必要提示以通知用户，仅此而已。而触发器的功能则不仅仅起提示作用，它还会引起系统内自动进行某些操作以消除违反完整性约束条件所引起的负面影响。

所谓触发器，其抽象的含义即是一个事件的发生必然触发（或导致）另外一些事件的发生，其中前面的事件称为触发事件，后面的事件称为结果事件。触发事件一般即为完整性约束条件的否定。而结果事件即为一组操作用以消除触发事件所引起的不良影响。在目前数据库中事件一般表示为数据的插入、修改、删除等操作。

触发器除了有完整性保护功能外，还有安全性保护功能。



上一节　本书简介　下一节



## 数据仓库与数据挖掘

### 5.7 数据仓库与数据挖掘

本节将介绍数据仓库与数据挖掘。



上一节　本书简介　下一节



## 数据仓库的概念

### 5.7.1 数据仓库的概念

目前，数据仓库一词尚没有一个统一的定义，著名的数据仓库专家W.H.Inmon在其著作《Buiding the Data Warehouse》中给予了如下描述：数据仓库（Data Warehouse）是一个面向主题的、集成的、相对稳定的且随时间变化的数据集合，用于支持管理决策。

**1.面向主题**

操作型数据库的数据组织面向事务处理任务（面向应用），各个业务系统之间各自分离，而数据仓库中的数据是按照一定的主题域进行组织的。主题是一个抽象的概念，是指用户使用数据仓库进行决策时所关心的重点方面，一个主题通常与多个操作型信息系统相关。例如，一个保险公司所进行的事务处理（应用问题）可能包括汽车保险、人寿保险、健康保险和意外保险等，而公司的主要主题范围可能是顾客、保险单、保险费和索赔等。

**2.集成的**

在数据仓库的所有特性中，这是最重要的。面向事务处理的操作型数据库通常与某些特定的应用相关，数据库之间相互独立，并且往往是异构的。而数据仓库中的数据是在对原有分散的数据库数据抽取、清理的基础上经过系统加工、汇总和整理得到的，必须消除源数据中的不一致性，以保证数据仓库内的信息是关于整个企业的一致的全局信息。

**3.相对稳定的**

操作型数据库中的数据通常实时更新，数据根据需要及时发生变化。数据仓库的数据主要供企业决策分析之用，所涉及的数据操作主要是数据查询，一旦某个数据进入数据仓库以后，一般情况下将被长期保留，也就是数据仓库中一般有大量的查询操作，但修改和删除操作很少，通常只需要定期的加载、刷新。

**4.随时间变化**

操作型数据库主要关心当前某一个时间段内的数据，而数据仓库中的数据通常包含历史信息，系统记录了企业从过去某一时点（如开始应用数据仓库的时点）到目前的各个阶段的信息，通过这些信息，可以对企业的发展历程和未来趋势做出定量分析和预测。

数据仓库反映历史变化的属性主要表现在以下方面。

数据仓库中的数据时间期限要远远长于传统操作型数据系统中的数据时间期限，传统操作型数据系统中的数据时间期限可能为数十天或数个月，数据仓库中的数据时间期限往往为数年甚至几十年。

传统操作型数据系统中的数据含有"当前值"的数据，这些数据在访问时是有效的，当然数据的当前值也能被更新，但数据仓库中的数据仅仅是一系列某一时刻（可能是传统操作型数据系统）生成的复杂的快照。

传统操作型数据系统中可能包含也可能不包含时间元素，如年、月、日、时、分、秒等，而数据仓库中一定会包含时间元素。

数据仓库虽然是从传统数据库系统发展而来，但是两者还是存在着诸多差异，如：从数据存储的内容看，数据库只存放当前值，而数据仓库则存放历史值；数据库数据的目标是面向业务操作人员的，为业务处理人员提供数据处理的支持，而数据仓库则是面向中高层管理人员的，为其提供决策支持等。表5-14详细说明了数据仓库与传统数据库的区别。

**表5-14 数据仓库与传统数据库的比较**

| 比 较 项 目 | 传统数据库 | 数 据 仓 库 |
|---|---|---|
| 数据内容 | 当前值 | 历史的、归档的、归纳的、计算的数据（处理过的） |
| 数据目标 | 面向业务操作程序、重复操作 | 面向主体域，分析应用 |
| 数据特性 | 动态变化、更新 | 静态、不能直接更新，只能定时添加、更新 |
| 数据结构 | 高度结构化、复杂，适合操作计算 | 简单、适合分析 |
| 使用频率 | 高 | 低 |
| 数据访问量 | 每个事务一般只访问少量记录 | 每个事务一般访问大量记录 |
| 对响应时间的要求 | 计时单位小，如秒 | 计时单位相对较大，除了秒，还有分钟、小时 |



上一节 本书简介 下一节



## 数据仓库的结构

### 5.7.2 数据仓库的结构

**1.数据仓库的概念结构**

从数据仓库的概念结构看，一般来说，数据仓库系统要包含数据源、数据准备区、数据仓库数

据库、数据集市/知识挖掘库，以及各种管理工具和应用工具，如图5-3所示。数据仓库建立之后，首先要从数据源中抽取相关的数据到数据准备区，在数据准备区中经过净化处理后再加载到数据仓库数据库，最后根据用户的需求将数据导入数据集市和知识挖掘库中。当用户使用数据仓库时，可以利用包括OAP在内的多种数据仓库应用工具向数据集市/知识挖掘库或数据仓库进行决策查询分析或知识挖掘。数据仓库的创建、应用可以利用各种数据仓库管理工具辅助完成。

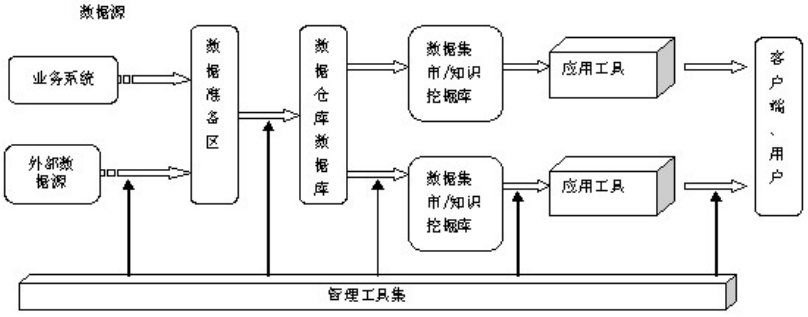


**图5-3 数据仓库的概念结构**

**2.数据仓库的参考框架**

数据仓库的参考框架由数据仓库基本功能层、数据仓库管理层和数据仓库环境支持层组成，如图5-4所示。

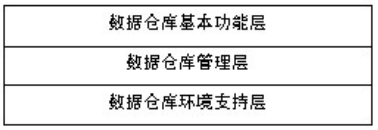


**图5-4 数据仓库的框架结构**

1）数据仓库基本功能层

数据仓库的基本功能层部分包含数据源、数据准备区、数据仓库结构、数据集市或知识挖掘库，以及存取和使用部分。本层的功能是从数据源抽取数据，对所抽取的数据进行筛选、清理，将处理过的数据导入或者说加载到数据仓库中，根据用户的需求设立数据集市，完成数据仓库的复杂查询、决策分析和知识的挖掘等。

2）数据仓库管理层

数据仓库的正常运行除了需要数据仓库功能层提供的基本功能外，还需要对这些基本功能进行管理与支持的结构框架。数据仓库管理层由数据仓库的数据管理和数据仓库的元数据管理组成。

数据仓库的数据管理层包含数据抽取、新数据需求与查询管理，数据加载、存储、刷新和更新系统，安全性与用户授权管理系统，以及数据归档、恢复及净化系统四部分。

3）数据仓库的环境支持层

数据仓库的环境支持层由数据仓库数据传输层和数据仓库基础层组成。数据仓库中不同结构之间的数据传输需要数据仓库的传输层来完成。

数据仓库的传输层包含数据传输和传送网络、客户/服务器代理和中间件、复制系统，以及数据传输层的安全保障系统。

**3.数据仓库的体系结构**

大众观点的数据仓库的体系结构如图5-5所示。

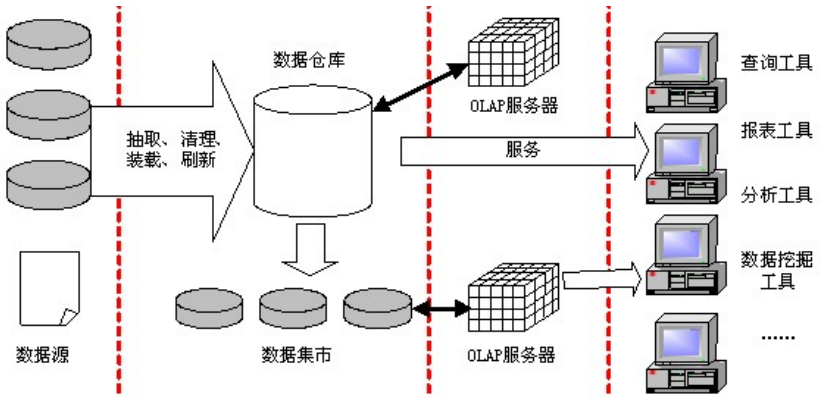


**图5-5 数据仓库体系结构**

1）数据源

数据源是数据仓库系统的基础，是整个系统的数据源泉。通常包括企业内部信息和外部信息。内部信息包括存放于RDBMS中的各种业务处理数据和各类文档数据。外部信息包括各类法律法规、市场信息和竞争对手的信息等。

2）数据的存储与管理

它是整个数据仓库系统的核心。数据仓库的真正关键是数据的存储和管理。数据仓库的组织管理方式决定了它有别于传统数据库，同时也决定了其对外部数据的表现形式。要决定采用什么产品和技术来建立数据仓库的核心，则需要从数据仓库的技术特点着手分析。针对现有各业务系统的数据，进行抽取、清理，并有效集成，按照主题进行组织。数据仓库按照数据的覆盖范围可以分为企业级数据仓库和部门级数据仓库（通常称为数据集市）。

3）OAP服务器

对分析需要的数据进行有效集成，按多维模型予以组织，以便进行多角度、多层次的分析，并发现趋势。其具体实现可以分为：ROAP、MOAP和HOAP.ROAP基本数据和聚合数据均存放在RDBMS之中；MOAP基本数据和聚合数据均存放于多维数据库中；HOAP基本数据存放于RDBMS之中，聚合数据存放于多维数据库中。

4）前端工具

主要包括各种报表工具、查询工具、数据分析工具、数据挖掘工具，以及各种基于数据仓库或数据集市的应用开发工具。其中数据分析工具主要针对OAP服务器，报表工具、数据挖掘工具主要针对数据仓库。



上一节　本书简介　下一节



## 数据挖掘技术概述

### 5.7.3 数据挖掘技术概述

随着数据库技术的迅速发展及数据库管理系统的广泛应用，人们积累的数据越来越多。激增的数据背后隐藏着许多重要的信息，人们希望能够对其进行更高层次的分析，以便更好地利用这些数据。目前的数据库系统可以高效地实现数据的录入、查询、统计等功能，但无法发现数据中存在的

关系和规则，无法根据现有的数据预测未来的发展趋势。缺乏挖掘数据背后隐藏的知识的手段，导致了"数据爆炸但知识贫乏"的现象。

数据挖掘（Data Mining）技术是人们长期对数据库技术进行研究和开发的结果。起初各种商业数据是存储在计算机的数据库中的，然后发展到可对数据库进行查询和访问，进而发展到对数据库的即时遍历。数据挖掘使数据库技术进入了一个更高级的阶段，它不仅能对过去的数据进行查询和遍历，并且能够找出过去数据之间的潜在联系，从而促进信息的传递。现在数据挖掘技术在商业应用中已经可以马上投入使用，因为对这种技术进行支持的三种基础技术已经发展成熟，它们是海量数据搜集、强大的多处理器计算机和数据挖掘算法。

从技术上来看，数据挖掘就是从大量的、不完全的、有噪声的、模糊的、随机的实际应用数据中，提取隐含在其中的、人们事先不知道的但又是潜在有用的信息和知识的过程。这个定义包括好几层含义：数据源必须是真实的、大量的、含噪声的；发现的是用户感兴趣的知识；发现的知识要可接受、可理解、可运用；并不要求发现放之四海而皆准的知识，仅支持特定的发现问题。

还有很多和这一术语相近似的术语，如从数据库中发现知识（KDD）、数据分析、数据融合（Data Fusion）及决策支持等。

何为知识？从广义上理解，数据、信息也是知识的表现形式，但是人们更把概念、规则、模式、规律和约束等看做知识。原始数据可以是结构化的，如关系数据库中的数据；也可以是半结构化的，如文本、图形和图像数据；甚至是分布在网络上的异构型数据。发现知识的方法可以是数学的，也可以是非数学的；可以是演绎的，也可以是归纳的。发现的知识可以被用于信息管理，查询优化，决策支持和过程控制等，还可以用于数据自身的维护。因此，数据挖掘是一门交叉学科，它把人们对数据的应用从低层次的简单查询，提升到从数据中挖掘知识，提供决策支持。在这种需求牵引下，汇聚了不同领域的研究者，尤其是数据库技术、人工智能技术、数理统计、可视化技术、并行计算等方面的学者和工程技术人员，投身到数据挖掘这一新兴的研究领域，形成新的技术热点。

从商业角度来看，数据挖掘是一种新的商业信息处理技术，其主要特点是对商业数据库中的大量业务数据进行抽取、转换、分析和其他模型化处理，从中提取辅助商业决策的关键性数据。

简而言之，数据挖掘其实是一类深层次的数据分析方法。数据分析本身已经有很多年的历史，只不过在过去数据收集和分析的目的是用于科学研究，另外，由于当时计算能力的限制，对大数据量进行分析的复杂数据分析方法受到很大限制。现在，由于各行业业务自动化的实现，商业领域产生了大量的业务数据，这些数据不再是为了分析的目的而收集的，而是由于纯机会的（Opportunistic）商业运作而产生的。分析这些数据也不再是单纯为了研究的需要，更主要的是为商业决策提供真正有价值的信息，进而获得利润。但所有企业面临的一个共同问题是：企业数据量非常大，而其中真正有价值的信息却很少，因此从大量的数据中经过深层分析，获得有利于商业运作、提高竞争力的信息，就像从矿石中淘金一样，数据挖掘也因此而得名。

因此，数据挖掘可以描述为：按企业既定业务目标，对大量的企业数据进行探索和分析，揭示隐藏的、未知的或验证已知的规律性，并进一步将其模型化的先进有效的方法。

数据挖掘与传统的数据分析（如查询、报表、联机应用分析）的本质区别是数据挖掘是在没有明确假设的前提下去挖掘信息、发现知识。数据挖掘所得到的信息应具有先知、有效和可实用3个特征。

先前未知的信息是指该信息是预先未曾预料到的，即数据挖掘是要发现那些不能靠直觉发现的

信息或知识，甚至是违背直觉的信息或知识，挖掘出的信息越是出乎意料，就可能越有价值。在商业应用中最典型的例子就是一家连锁店通过数据挖掘发现了小孩尿布和啤酒之间有着惊人的联系。

特别要指出的是，数据挖掘技术从一开始就是面向应用的。它不仅是面向特定数据库的简单检索查询调用，而且要对这些数据进行微观、中观乃至宏观的统计、分析、综合和推理，以指导实际问题的求解，企图发现事件间的相互关联，甚至利用已有的数据对未来的活动进行预测。例如，加拿大BC省电话公司要求加拿大SimonFraser大学KDD研究组，根据其拥有十多年的客户数据，总结、分析并提出新的电话收费和管理办法，制定既有利于公司又有利于客户的优惠政策。这样一来，就把人们对数据的应用，从低层次的末端查询操作，提高到为各级经营决策者提供决策支持。这种需求驱动力比数据库查询更为强大。





第 5 章：数据库系统　作者：希赛教育软考学院　来源：希赛网　2014年01月26日

## 数据挖掘的功能

### 5.7.4 数据挖掘的功能

数据挖掘通过预测未来趋势及行为，做出前摄的、基于知识的决策。数据挖掘的目标是从数据库中发现隐含的、有意义的知识，主要有以下五类功能：

**1.自动预测趋势和行为**

数据挖掘自动在大型数据库中寻找预测性信息，以往需要进行大量手工分析的问题如今可以迅速直接由数据本身得出结论。一个典型的例子是市场预测问题，数据挖掘使用过去有关促销的数据来寻找未来投资中回报最大的用户，其他可预测的问题包括预报破产及认定对指定事件最可能做出反应的群体。

**2.关联分析**

数据关联是数据库中存在的一类重要的可被发现的知识。若两个或多个变量的取值之间存在某种规律性，就称为关联。关联可分为简单关联、时序关联、因果关联。关联分析的目的是找出数据库中隐藏的关联网。有时并不知道数据库中数据的关联函数，即使知道也是不确定的，因此关联分析生成的规则带有可信度。

**3.聚类**

数据库中的记录可被划分为一系列有意义的子集，即聚类。聚类增强了人们对客观现实的认识，是概念描述和偏差分析的先决条件。聚类技术主要包括传统的模式识别方法和数学分类学。20世纪80年代初，Mchalski提出了概念聚类技术及其要点，即在划分对象时不仅考虑对象之间的距离，还要求划分出的类具有某种内涵描述，从而避免了传统技术的某些片面性。

**4.概念描述**

概念描述就是对某类对象的内涵进行描述，并概括这类对象的有关特征。概念描述分为特征性描述和区别性描述，前者描述某类对象的共同特征，后者描述不同类对象之间的区别。生成一个类的特征性描述只涉及该类对象中所有对象的共性。生成区别性描述的方法很多，如决策树方法、遗传算法等。

**5.偏差检测**

数据库中的数据常有一些异常记录，从数据库中检测这些偏差很有意义。偏差包括很多潜在的知识，如分类中的反常实例、不满足规则的特例、观测结果与模型预测值的偏差、量值随时间的变化等。偏差检测的基本方法是，寻找观测结果与参照值之间有意义的差别。



上一节　本书简介　下一节



## 数据挖掘常用技术

**5.7.5 数据挖掘常用技术**

常用的数据挖掘技术包括关联分析、序列分析、分类、预测、聚类分析，以及时间序列分析等。

**1.关联分析**

关联分析主要用于发现不同事件之间的关联性，即一个事件发生的同时，另一个事件也经常发生。关联分析的重点在于快速发现那些有实用价值的关联发生的事件。其主要依据是事件发生的概率和条件概率应该符合一定的统计意义。

对于结构化的数据，以客户的购买习惯数据为例，利用关联分析，可以发现客户的关联购买需要。例如，一个开设储蓄账户的客户很可能同时进行债券交易和股票交易，购买纸尿裤的男顾客经常同时购买啤酒等。利用这种知识可以采取积极的营销策略，扩展客户购买的产品范围，吸引更多的客户。通过调整商品的布局便于顾客买到经常同时购买的商品，或者通过降低一种商品的价格来促进另一种商品的销售等。

对于非结构化的数据，以空间数据为例，利用关联分析，可以发现地理位置的关联性。例如，85%的靠近高速公路的大城镇与水相邻，或者发现通常与高尔夫球场相邻的对象等。

**2.序列分析**

序列分析技术主要用于发现一定时间间隔内接连发生的事件。这些事件构成一个序列，发现的序列应该具有普遍意义，其依据除了统计上的概率之外，还要加上时间的约束。

**3.分类分析**

分类分析通过分析具有类别的样本的特点，得到决定样本属于各种类别的规则或方法。利用这些规则和方法对未知类别的样本分类时应该具有一定的准确度。其主要方法有基于统计学的贝叶斯方法、神经网络方法、决策树方法等。

利用分类技术，可以根据顾客的消费水平和基本特征对顾客进行分类，找出对商家有较大利益贡献的重要客户的特征，通过对其进行个性化服务，提高他们的忠诚度。

利用分类技术，可以将大量的半结构化的文本数据，如Web页面、电子邮件等进行分类。可以将图片进行分类，例如，根据已有图片的特点和类别，可以判定一幅图片属于何种类型的规则。对于空间数据，也可以进行分类分析，例如，可以根据房屋的地理位置决定房屋的档次。

**4.聚类分析**

聚类分析是根据物以类聚的原理，将本身没有类别的样本聚集成不同的组，并且对每一个这样

的组进行描述的过程。其主要依据是聚到同一个组中的样本应该彼此相似，而属于不同组的样本应该足够不相似。

仍以客户关系管理为例，利用聚类技术，根据客户的个人特征及消费数据，可以将客户群体进行细分。例如，可以得到这样的一个消费群体：女性占91%,全部无子女、年龄在31~40岁占70%,高消费级别的占64%,买过针织品的占91%,买过厨房用品的占89%,买过园艺用品的占79%.针对不同的客户群，可以实施不同的营销和服务方式，从而提高客户的满意度。

对于空间数据，根据地理位置及障碍物的存在情况可以自动进行区域划分。例如，根据分布在不同地理位置的ATM机的情况将居民进行区域划分，根据这一信息，可以有效地进行ATM机的设置规划，避免浪费，同时也避免失掉每一个商机。

对于文本数据，利用聚类技术可以根据文档的内容自动划分类别，从而便于文本的检索。

**5.预测**

预测与分类类似，但预测是根据样本的已知特征估算某个连续类型的变量的取值过程，而分类则只是用于判别样本所属的离散类别而已。预测常用的技术是回归分析。

**6.时间序列分析**

时间序列分析的是随时间而变化的事件序列，目的是预测未来发展趋势，或者寻找相似发展模式或者是发现周期性发展规律。





第 5 章：数据库系统　作者：希赛教育软考学院　来源：希赛网　2014年01月26日

## 数据挖掘的流程

**5.7.6 数据挖掘的流程**

数据挖掘是指一个完整的过程，该过程从大型数据库中挖掘先前未知的、有效的、可实用的信息，并利用这些信息做出决策或丰富知识。

数据挖掘环境示意图如图5-6所示。


**图5-6 数据挖掘环境框图**

数据挖掘的流程大致如下。

**1.问题定义**

在开始数据挖掘之前最先的也是最重要的要求就是熟悉背景知识，弄清用户的需求。缺少了背景知识，就不能明确定义要解决的问题，就不能为挖掘准备优质的数据，也很难正确地解释得到的结果。要想充分发挥数据挖掘的价值，必须对目标有一个清晰明确的定义，即决定到底想干什么。

**2.建立数据挖掘库**

要进行数据挖掘必须收集要挖掘的数据资源。一般建议把要挖掘的数据都收集到一个数据库中，而不是采用原有的数据库或数据仓库。这是因为大部分情况下需要修改要挖掘的数据，而且还会遇到采用外部数据的情况；另外，数据挖掘还要对数据进行各种纷繁复杂的统计分析，而数据仓

库可能不支持这些数据结构。

**3.分析数据**

分析数据就是通常所进行的对数据深入调查的过程。从数据集中找出规律和趋势，用聚类分析区分类别，最终要达到的目的就是搞清楚多因素相互影响的、十分复杂的关系，发现因素之间的相关性。

**4.调整数据**

通过上述步骤的操作，对数据的状态和趋势有了进一步的了解，这时要尽可能对问题解决的要求能进一步明确化，进一步量化。针对问题的需求对数据进行增删，按照对整个数据挖掘过程的新认识组合或生成一个新的变量，以体现对状态的有效描述。

**5.模型化**

在问题进一步明确，数据结构和内容进一步调整的基础上，就可以建立形成知识的模型。这一步是数据挖掘的核心环节，一般运用神经网络、决策树、数理统计、时间序列分析等方法来建立模型。

**6.评价和解释**

上面得到的模式模型，有可能是没有实际意义或没有实用价值的，也有可能是其不能准确反映数据的真实意义，甚至在某些情况下是与事实相反的，因此需要评估，确定哪些是有效的、有用的模式。评估的一种办法是直接使用原先建立的挖掘数据库中的数据来进行检验，另一种办法是另找一批数据并对其进行检验，再一种办法就是在实际运行的环境中取出新鲜数据进行检验。

数据挖掘过程的分步实现，不同的步骤需要不同专长的人员，他们大体可以分为三类。

业务分析人员：要求精通业务，能够解释业务对象，并根据各业务对象确定出用于数据定义和挖掘算法的业务需求。

数据分析人员：精通数据分析技术，并对统计学有较熟练的掌握，有能力把业务需求转化为数据挖掘的各步操作，并为每步操作选择合适的技术。

数据管理人员：精通数据管理技术，并从数据库或数据仓库中收集数据。

由此可见，数据挖掘是一个多种专家合作的过程，也是一个在资金上和技术上高投入的过程。这一过程要反复进行，在反复过程中，不断地趋近事物的本质，不断地优选问题的解决方案。



上一节　本书简介　下一节



作者：希赛教育软考学院　来源：希赛网　2014年01月26日

## 分布式数据库

**5.8　分布式数据库**

**1.分布式数据库系统的定义与特点**

分布式数据库是由一组数据组成的，这组数据分布在计算机网络的不同计算机上，网络中的每个结点具有独立处理的能力（称为场地自治），它可以执行局部应用，同时，每个结点也能通过网络通信子系统执行全局应用。

分布式数据库系统是在集中式数据库系统技术的基础上发展起来的，具有如下特点：

数据独立性：在分布式数据库系统中，数据独立性这一特性更加重要，并具有更多的内容。除了数据的逻辑独立性与物理独立性外，还有数据分布独立性，亦称"分布透明性".

集中与自治共享结合的控制结构：各局部的DBMS可以独立地管理局部数据库，具有自治的功能。同时，系统又设有集中控制机制，协调各局部DBMS的工作，执行全局应用。

适当增加数据冗余度：在不同的场地存储同一数据的多个副本，这样可以提高系统的可靠性、可用性，同时也能提高系统性能。

全局的一致性、可串行性和可恢复性。

分布式数据库系统的目标，主要包括技术和组织两方面的目标。

适应部门分布的组织结构，降低费用。

提高系统的可靠性和可用性。

充分利用数据库资源，提高现有集中式数据库的利用率。

逐步扩展处理能力和系统规模。

**2.分布式数据存储**

分布式数据存储可以从数据分配和数据分片两个角度考察。

数据分配是指数据在计算机网络各场地上的分配策略，包括以下内容。

集中式：所有数据均安排在同一个场地上。

分割式：所有数据只有一份，分别被安置在若干个场地。

全复制式：数据在每个场地重复存储。

混合式：数据库分成若干可相交的子集，每一子集安置在一个或多个场地上，但是每一场地未必保存全部数据。

对于上述分配策略，有四个评估因素。

（1）存储代价；（2）可靠性；（3）检索代价；（4）更新代价。

存储代价和可靠性是一对矛盾的因素；检索代价和更新代价也是一对矛盾的因素。

数据分片是指数据存放单位不是全部关系，而是关系的一个片段，也就是关系的一部分，包括以下内容。

水平分片：按一定的条件把全局关系的所有元组划分成若干不相交的子集，每个子集为关系的一个片段。

垂直分片：把一个全局关系的属性集分成若干子集，并在这些子集上做投影运算，每个投影为垂直分片。

混合型分片：将水平分片与垂直分片方式综合使用，则为混合型分片。

数据分片应遵循的准则如下：

完备性条件：必须把全局关系的所有数据映射到各个片段中，绝不允许发生属于全局关系的某个数据不属于任何一个片段。

重构条件：划分所采用的方法必须确保能够由各个片段重建全局关系。

不相交条件：要求一个全局关系被划分后得到的各个数据片段互相不重叠。

**3.分布式数据库系统的体系结构**

分布式DBS的体系结构分为4级：全局外模式、全局概念模式、分片模式和分布模式。

全局外模式：它们是全局应用的用户视图，是全局概念模式的子集。

全局概念模式：全局概念模式定义了分布式数据库中所有数据的逻辑结构。

分片模式：分片模式定义片段及定义全局关系与片段之间的映像。这种映像是一对多的，即每个片段来自一个全局关系，而一个全局关系可分成多个片段。

分布模式：片段是全局关系的逻辑部分，一个片段在物理上可以分配到网络的不同结点上。分布模式根据数据分配策略的选择定义片段的存放场地。

分布式DBS的分层体系结构有3个特征。

数据分片和数据分配概念的分离，形成了"数据分布独立性"概念。

数据冗余的显式控制。

局部DBMS的独立性。

**4.分布透明性**

分布透明性指用户不必关心数据的逻辑分片，不必关心数据物理位置分配的细节， 也不必关心各个场地上数据库数据模型。分布透明性可归入物理独立性的范围。

分布透明性包括3个层次：分片透明性、位置透明性和局部数据模型透明性。

**5.分布式数据库管理系统的功能及组成**

主要功能有：

接受用户请求，并判定把它送到哪里，或必须访问哪些计算机才能满足该请求。

访问网络数据字典，或者至少了解如何请求和使用其中的信息。

如果目标数据存储于系统的多个计算机上，就必须进行分布式处理。

通信接口功能，在用户、局部DBMS和其他计算机的DBMS之间进行协调。

在一个异构型分布式处理环境中，还需提供数据和进程移植的支持。这里的异构型是指各个场地的硬件、软件之间存在一定差别。

D-DBMS由4个部分组成：

DBMS.局部场地上的数据库管理系统其功能是建立和管理局部数据库，提供场地自治能力、执行局部应用及全局查询的子查询。

GDBMS.全局数据库管理系统主要功能是提供分布透明性，协调全局事务的执行，协调各局部DBMS以完成全局应用，保证数据库的全局一致性，执行并发控制，实现更新同步，提供全局恢复功能。

全局数据字典存放全局概念模式、分片模式、分布模式的定义，以及各模式之间映像的定义，存放有关用户存取权限的定义，以保证全局用户的合法权限和数据库的安全性，存放数据完整性约束条件的定义，其功能与集中式数据库的数据字典类似。

通信管理在分布数据库各场地之间传送消息和数据，完成通信功能。

**6.分布式数据库系统要解决的问题**

在集中式系统中，主要目标是减少对磁盘的访问次数。对于分布式系统，压倒一切的性能目标是使通过网络传送信息的次数和数据量最小。



上一节　　本书简介　　下一节



## 例题分析

**5.9 例题分析**

**例题1（2011年5月试题51~54）**

某医院数据库的部分关系模式为：科室（科室号，科室名，负责人，电话）、病患（病历号，姓名，住址，联系电话）和职工（职工号，职工姓名，科室号，住址，联系电话）。假设每个科室有一位负责人和一部电话，每个科室有若干名职工，一名职工只属于一个科室；一个医生可以为多个病患看病；一个病患可以由多个医生多次诊治。科室与职工的所属联系类型为 （51） ,病患与医生的就诊联系类型为 （52） .对于就诊联系最合理的设计是 （53） ,就诊关系的主键是 （54） .

（51）A.1:1 B.1:n C.n:1 D.n:m

（52）A.1:1 B.1:n C.n:1 D.n:m

（53）A.就诊（病历号，职工号，就诊情况）

B.就诊（病历号，职工姓名，就诊情况）

C.就诊（病历号，职工号，就诊时间，就诊情况）

D.就诊（病历号，职工姓名，就诊时间，就诊情况）

（54）A.病历号，职工号 B.病历号，职工号，就诊时间

C.病历号，职工姓名 D.病历号，职工姓名，就诊时间

**例题分析：**

本题主要考查关系模式的基础知识。

在本题中，题目告诉我们每个科室有一位负责人和若干名职工，而一名职工只属于一个科室，那么很容易我们就能知道科室与职工的所属联系类型为1:n.

另外，题目告诉我们一个医生可以为多个病患看病，一个病患可以由多个医生多次诊治，所以病患与医生的就诊联系类型为多对多。

根据题目意思，就诊应该是病患与医生之间的联系，他们之间的联系是多对多，因此其联系要转换为独立的关系模式时，应该包含病患和医生关系模式的主键及自身的一些属性，如就诊时间，就诊情况。而病患的主键是病历号，而职工关系模式的主键为职工号，因此就诊关系模式为就诊（病历号，职工号，就诊时间，就诊情况），而该关系模式的主键是（病历号，职工号，就诊时间），因为这才能唯一标识一条记录。至于主键为什么不是（病历号，职工号），是因为存在同一个病人多次看同一个医生的情况，所以（病历号，职工号）不能唯一标识一条记录。

例题答案：（51）B（52）D （53）C （54）B

例题2（2011年5月试题55~56）

给定关系模式R<U,F>,U= {A,B,C},F={AB→C,C→B}.关系R （55） ,且分别有 （56） .

（55）A.只有1个候选关键字AC B.只有1个候选关键字AB

C.有2个候选关键字AC和BC D.有2个候选关键字AC和AB

（56）A.1个非主属性和2个主属性 B.2个非主属性和1个主属性

C.0个非主属性和3个主属性 D.3个非主属性和O个主属性

**例题分析：**

本题主要考查函数依赖的基础知识。

关系中的某一属性或属性组的值能唯一的标识一个元组，而其任何真子集都不能再标识，则称该属性组为候选码。

但这里大家要注意，如果一个关系有多个不同的主码时，那么这些主码组合在一起就是候选码，也就是说一个关系的主码只能选一个，而候选码可以有多个，这就好选总统一样，候选人可以多个，但最终的总统只能有一个，当然也有些地方的候选人就只有一个，候选码也一样，也有可能只有一个，在这种情况下，候选码就是主码。

主属性和非主属性是互补的，一个关系模式中的属性不是主属性就是非主属性。组成候选码的属性就是主属性，其它的就是非主属性，所以要判断关系模式中的属性是主属性还是非主属性，首先要求解出其候选码。

在本题中，从题目给出的函数依赖关系我们可以看出，AB能推导出C,即能推导出所有的属性；而C能推导出B,同样，AC也能推导出所有的属性，因此AB与AC都是该关系的候选码。所以该关系中的属性都是主属性，没有非主属性。

**例题答案：（55）D（56）C**





第 6 章：多媒体技术及其应用　作者：希赛教育软考学院　来源：希赛网　2014年01月26日

## 多媒体技术基本概念

**第6章　多媒体技术及其应用**

根据考试大纲的规定，本章要求考生掌握以下知识：

多媒体系统基础知识；

简单图形的绘制，图像文件的处理方法；

音频和视频信息的应用；

多媒体应用开发过程。

但从历年考试试题来看，主要集中在音频、视频、图形和图像等方面。考生在复习时应掌握基本概念，熟悉有关的多媒体文件容量、量化方面的计算。

在多媒体应用开发过程方面，考生要注意多媒体应用系统具有以下特点：

增强了计算机的友好性；

涉及技术领域广、技术层次高；

多媒体技术的标准化；

多媒体技术的集成化和工具化。

多媒体应用系统开发组需要应用系统组长、多媒体设计师、音频专家、视频专家、写作专家、多媒体程序员等人员，其具体开发过程与非多媒体项目是类似的，请读者参考本书有关软件工程知识的章节。

**6.1　多媒体技术基本概念**

多媒体主要是指文字、声音和图像等多种表达信息的形式和媒体，它强调多媒体信息的综合和集成处理。多媒体技术依赖于计算机的数字化和交互处理能力，它的关键是信息压缩技术和光盘存储技术。

**1.亮度、色调和饱和度**

视觉上的彩色可用亮度、色调和饱和度来描述，任意一种彩色光都是这3个特征的综合效果。

亮度是光作用于人眼时所引起的明亮程度的感觉，它与被观察物体的发光强度有关；由于其强度不同，看起来可能亮一些或暗一些。对于同一物体照射的光越强，反射光也越强，感觉越亮，对于不同物体在相同照射情况下，反射性越强者看起来越亮。显然，如果彩色光的强度降至使人看不清了，在亮度等级上它应与黑色对应；同样如果其强度变得很大，那么亮度等级应与白色对应。此外，亮度感还与人类视觉系统的视敏功能有关，即使强度相同，颜色不同的光进入视觉系统，也可能会产生不同的亮度。

色调是当人眼看到一种或多种波长的光时所产生的彩色感觉，它反映颜色的种类，是决定颜色的基本特性。如红色、绿色等都是指色调。不透明物体的色调是指该物体在日光照射下，所反射的各光谱成分作用于人眼的综合效果；透明物体的色调则是透过该物体的光谱综合作用的效果。

饱和度是指颜色的纯度，即掺入白光的程度，或者说是指颜色的深浅程度。对于同一色调的彩色光，饱和度越深，颜色越鲜明，或者说越纯。例如，当红色加进白光之后冲淡为粉红色，其基本色调还是红色，但饱和度降低；换句话说，淡色的饱和度比深色要低一些。饱和度还和亮度有关，因为若在饱和的彩色光中增加白光的成分，由于增加了光能，因而变得更亮了，但是它的饱和度却降低了。如果在某色调的彩色光中掺入别的彩色光，会引起色调的变化，掺入白光时仅引起饱和度的变化。

**2.三原色原理**

三原色原理是色度学中最基本的原理，是指自然界常见的各种颜色光，都可由红（R）、绿（G）、蓝（B）3种颜色按不同比例相配制而成；同样绝大多数颜色光也可以分解成红、绿、蓝三种色光。当然三原色的选择并不是唯一的，也可以选择其他3种颜色为三原色，但是，3种颜色必须是相互独立的，即任何一种颜色都不能由其他两种颜色合成。由于人眼对红、绿、蓝3种色光最敏感，因此，由这3种颜色相配制所得的彩色范围也最广，所以一般都选用这3种颜色作为基色。

**3.彩色空间**

RGB彩色空间：在多媒体计算机技术中，用得最多的是RGB彩色空间表示。因为计算机的彩色监视器的输入需要R、G、B 3个彩色分量，通过3个分量的不同比例，在显示屏幕上可以合成所需要的任意颜色，所以不管多媒体系统采用什么形式的彩色空间表示，最后的输出一定要转换成RGB彩色空间表示。

YUV彩色空间：在现代彩色电视系统中，通常采用三管彩色摄像机或彩色CCD摄像机，把摄得的彩色图像信号经分色棱镜分成R0、G0、B0 3个分量的信号；分别经放大和校正得到三基色，再经过矩阵变换电路得到亮度信号Y、色差信号R–Y和B–Y,最后发送端将Y、R–Y和B–Y 3个信号进行编码，用同一信道发送出去，这就是我们常用的YUV彩色空间。

YUV彩色空间与RGB彩色空间的换算关系如下：

Y = 0.3×R+0.59×G+0.11×B;

U = （B–Y）×0.493;

V = （R–Y）×0.877.

其他彩色空间表示：彩色空间表示还有很多种如CIE（国际照明委员会）制定的CIE XYZ、CIE AB彩色空间和CCIR（Consutative Committee Internationa Radio）制定的CCIR601-2YCC彩色空间，以及HIS（Hue,Saturation,Intensity）等。