

获取最新事业编微信：9062749【课程+时政+密卷+题库】



华图网校  
V.HUATU.COM

乐享品质 高效学习

# 数据库系统

刘佩贤

华图网校

## 目录

数据库系统	2
第 1 章 数据模型与数据库系统结构	2
1. 数据和数据模型	3
2. 概念层数据模型	3
3. 组织层数据模型	4
4. 数据库系统结构	4
第 2 章 关系数据库	5
1. 关系数据模型	6
2. 关系模型的基本术语与形式化定义	6
3. 完整性约束	8
4. 关系代数	8
第 3 章 关系数据库标准语言 SQL	14
1. 基本表的定义、删除及修改	15
2. 索引的建立与删除	17
3. SQL 的数据查询	18
4. 数据更新	21
5. 视图	21
第 4 章 数据库安全性	23
第 5 章 数据库完整性	24
第 6 章 关系数据理论	24
第 7 章 数据库设计	26
第 8 章 并发控制	28
1. 事务基本概念	29
2. 并发控制概述	29
3. 并发控制措施	31
4. 封锁协议	31
5. 死锁	32

6. 两阶段锁 .....	32
第 9 章 数据库恢复技术 .....	33
1. 恢复的基本概念 .....	33
2. 数据库故障的种类 .....	34
3. 恢复方法 .....	34

## 数据库系统

### 第 1 章 数据模型与数据库系统结构

主要内容：

1. 数据和数据模型



第 2 页

最新事业编综合类教育类医疗类 获取方式微信：9062749

2. 概念层数据模型
3. 组织层数据模型
4. 数据库系统结构

## 1. 数据和数据模型

数据：描述事物的符号记录称为数据。

将从数据中获得的有意义的内容称为信息。

例如数据：（张三，9912101，男，1981，计算机系，应用软件）



数据模型：它是对现实世界数据特征的抽象。

一般将对数据的静态特征和动态特征的描述称为数据模型三要素：数据的基本结构、数据的约束条件（静态特征）和定义在数据上的操作（动态特征）。

数据模型根据模型应用的不同目的，可以将模型分为两大类：

概念层数据模型（概念模型），从数据的语义视角来抽取模型，是按用户的观点来对数据和信息进行建模。

组织层数据模型（组织模型）。从数据的组织层次来描述数据。

## 2. 概念层数据模型

用于对信息世界建模，是现实世界到信息世界的第一层抽象，是数据库设计人员进行数据库设计的工具。

是面向用户、面向现实世界的数据模型，与具体的 DBMS（数据库管理系统）无关。

常用概念模型：实体-联系模型（E-R 模型）

- 实体-联系模型

①实体：具有公共性质的可相互区分的现实世界对象的集合。例如：学生、课程。

在 E-R 图中用矩形框表示具体的实体，把实体名写在框内。

实体中每个具体的记录值（一行数据），称为实体的一个实例。

②属性：描述实体或者联系的性质或特征的数据项。用圆角矩形或椭圆框表示。

能够唯一标识实体的一个属性或最小的一组属性称为实体的标识属性（码）。例如：“学号”就是学生实体的码。



③联系：联系是数据之间的关联集合。实体之间的联系用菱形框表示，框内写上联系名，并用连线与有关的实体相连。



联系的种类：一对一联系(1:1)：部门和经理

一对多联系(1:n)：部门和职工

多对多联系(m:n)：学生和课程

### 3.组织层数据模型

组织层数据模型是从数据的组织形式的角度来描述信息。

分类：层次模型、网状模型、关系模型

①层次数据模型

是数据库管理系统中最早出现的数据模型。用树形结构表示实体和实体之间的联系。

②网状数据模型

用图形结构表示实体和实体之间的联系。

③关系数据模型

关系数据模型源于数学，它把数据看成是二维表中的元素，而这个二维表在关系数据库中就称为关系。

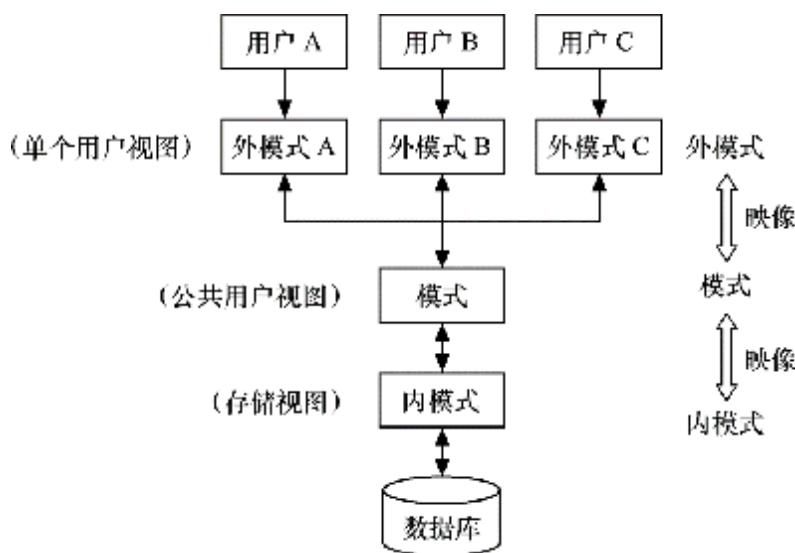
在关系数据模型中，实体本身以及实体和实体之间的联系都用关系来表示。

### 4.数据库系统结构

数据模型（组织层数据模型）是描述数据的组织形式，模式是用给定的数据模型对具体数据的描述，是数据库中全体数据的逻辑结构和特征的描述，关系模式实际上对应的是关系表的表头。

- 从数据库管理角度看，数据库通常采用三级模式结构。这是数据库管理系统

内部的系统结构。



- 外模式：最接近用户，也就是用户看到的数据视图。是关系的，它们的内容来自概念模式。
- 概念模式：是关系的，在该层可见的实体是关系的表和关系的操作符。
- 内模式：不是关系的，它是数据的物理存储方式。
- 为了能够在内部实现这三个抽象层次的联系和转换，数据库管理系统在三级模式之间提供了两层映象：
  - Ø 外模式/模式映象：是数据逻辑独立性的关键。
  - Ø 模式/内模式映象：是数据物理独立性的关键。

#### 习题解析

1. 要保证数据库的数据独立性，需要修改的是（ ）。
  - 模式与外模式
  - 模式与内模式
  - 三层之间的两种映射
  - 三层模式
2. 描述数据库全体数据的全局逻辑结构和特性的是（ ）。
  - 模式
  - 内模式
  - 外模式
  - 模式和内模式
3. 实体之间的联系有（ ）、（ ）、（ ）三种。

## 第 2 章 关系数据库

主要内容：

1. 关系数据模型
2. 基本术语与形式化定义

3. 完整性约束
4. 关系代数

## 1.关系数据模型

关系模型由三部分组成（关系模型三要素）：数据结构、操作集合、完整性约束。

①数据结构：用二维表（关系）来组织数据。表是逻辑结构而不是物理结构。

②数据操作

- 传统的关系运算：并、交、差、广义笛卡尔乘积；
- 专门的关系运算：选择、投影、连接、除；
- 有关的数据操作：查询、插入、删除、更改。

③数据完整性约束

数据完整性是一种语义概念，包括：

- 与现实世界中应用需求的数据的相容性和正确性；
- 数据库内数据之间的相容性和正确性。

## 2.关系模型的基本术语与形式化定义

关系：关系就是二维表，二维表的名字就是关系的名字。

属性：二维表中的每个列就称为一个属性（或叫字段），每个属性有一个名字——属性名，某一列的值——属性值，列的个数——关系的元数。

值域：二维表中属性的取值范围称为值域。

元组：二维表中的一行数据称为一个元组（记录值）。

分量：元组中的每一个属性值称为元组的一个分量。

关系模式：二维表的结构称为关系模式。

$R (A_1, A_2, \dots, A_n)$ ，关系名为  $R$ ，属性分别为  $A_1, A_2, \dots, A_n$ 。

关系数据库：对应于一个关系模型的所有关系的集合称为关系数据库。

候选键：如果一个属性或属性集的值能够唯一标识一个关系的元组而又不包含多余的属性，则称该属性或属性集为候选键。

主键（主码、主关键字）：当一个关系中有多个候选键时，从中选择一个作为主键。每个关系只能有一个主键。主键可以由一个属性组成，也可以由多个属性共同组成。

例：

学生（学号，姓名，性别，年龄，所在系）



选课（学号，课程号，成绩）

主属性：包含在任一候选键中的属性称为主属性。

非主属性：不包含在任一候选键中的属性称为非主属性。

术语对比

关系术语	一般的表格术语
关系名	表名
关系模式	表头（表所含列的描述）
关系	（一张）二维表
元组	记录或行
属性	列
分量	一条记录中某个列的值

- 形式化定义

- 定义笛卡尔积：设  $D_1, D_2, \dots, D_n$  为任意集合，定义笛卡尔积  $D_1, D_2, \dots, D_n$  为： $D_1 \times D_2 \times \dots \times D_n = \{(d_1, d_2, \dots, d_n) \mid d_i \in D_i, i=1, 2, \dots, n\}$
- 其中每一个元素  $(d_1, d_2, \dots, d_n)$  称为一个  $n$  元组，简称元组。
- 元组中每一个  $d_i$  称为是一个分量。

例：笛卡尔乘积实际就是二维表



- 对关系的限定

Ø 关系中的每个分量都是不可再分的最小属性

- Ø 表中列的数据类型是固定的。
- Ø 不同列的数据可以取自相同的值域。
- Ø 关系表中列的顺序不重要。
- Ø 关系表行的顺序也不重要。
- Ø 同一个关系中的元组不能重复。

### 3.完整性约束

数据完整性是指数据库中存储的数据是有意义的或正确的，和现实世界相符。

- ① 实体完整性
- ② 参照完整性（引用完整性）
- ③ 用户定义的完整性
- ④ 实体完整性

指关系数据库中所有的表都必须有主键，而且表中不允许存在如下记录：

- Ø 无主键值的记录。
  - Ø 主键值相同的记录
- ② 参照完整性

也称引用完整性。现实世界中的实体之间往往存在着某种联系，在关系模型中，实体以及实体之间的联系都是用关系来表示的，这样就自然存在着关系与关系之间的引用。

学生(学号, 姓名, 班号, 性别)

班(班号, 所属专业, 人数)

- 设 F 是关系 R 的一个或一组属性，如果 F 与关系 S 的主键相对应，则称 F 是关系 R 的外键 (Foreign Key)，并称关系 R 为参照关系，关系 S 为被参照关系。

- ③ 用户定义的完整性

也称为域完整性或语义完整性。是针对某一具体应用领域定义的数据约束条件。

### 4.关系代数

关系代数是关系操作语言的一种传统表示方式，是一种抽象的查询语言。

是一种纯理论语言，它定义了一些操作，运用这些操作可以从一个或多个关系中得到

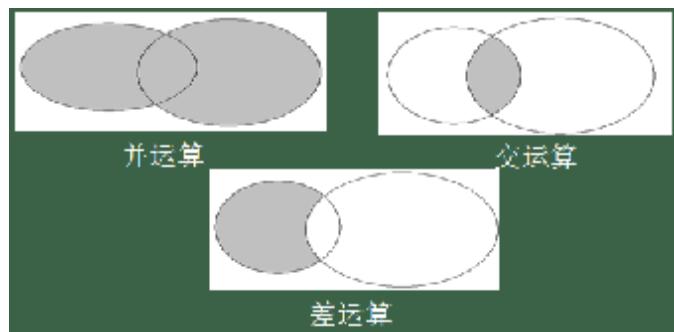


另一个关系，而不改变源关系。

- 关系代数中常用的运算及运算符：

- 集合运算符： $\cup$ （并）， $-$ （差）， $\cap$ （交）， $\times$ （笛卡儿积）；
- 关系运算符： $\Pi$ （投影）， $\sigma$ （选择）， $\bowtie$ （连接）；
- 算术比较符： $>$ （大于）， $\geq$ （大于等于）， $<$ （小于）， $\leq$ （小于等于）， $\neq$ （不等于）
- 逻辑运算符： $\neg$ （非）， $\wedge$ （与）， $\vee$ （或）

- 集合运算： $\cup$ （并）， $-$ （差）， $\cap$ （交）， $\times$ （笛卡儿积）；



**A**

学号	课程名称
050001	高等数学
050001	英语
050001	C语言
050001	数据库原理
050002	高等数学
050002	英语

**B**

学号	课程名称
050001	英语
050001	文学
050001	计算机基础
050001	管理学
050002	英语
050002	文学

**A $\cap$ B**

学号	课程名称
050001	英语
050002	英语

含义：学生在A地和B地进行选课，学生在两个地方同时选修了哪些课程.

**A-B**

学号	课程名称
050001	高等数学
050001	C语言
050001	数据库原理
050002	高等数学

含义：学生在A地和B地进行选课，学生仅在A地选的课程.

**A $\cup$ B**

学号	课程名称
050001	高等数学
050001	英语
050001	C语言
050001	数据库原理
050002	高等数学
050002	英语
050002	文学
050001	文学
050001	计算机基础
050001	管理学
050002	英语
050002	文学

含义：学生在A地和B地共选修了哪些课程.

含义：学生在A地和B地进行选课，学生在两个地方同时选修了哪些课程.

- 关系运算符：

- 投影运算 ( $\Pi$ ) ——Project

对单个关系的运算，从一个关系中选择所需要的属性，并重新排列组成一个新的关系。是列方向的运算。

例如：求学生的学号、姓名和所学专业

学生信息表						学生名单		
学号	姓名	性别	出生年月	所学专业	入学日期	学号	姓名	所学专业
050001	李红	女	1983.01.03	信息管理	2005.09.01	050001	李红	信息管理
050002	王名	男	1984.12.12	信息管理	2005.09.01	050002	王名	信息管理
050003	刘丽	女	1982.07.08	信息管理	2005.09.01	050003	刘丽	信息管理
050005	张力	男	1981.09.12	信息管理	2005.09.01	050005	张力	信息管理
050005	武晓	男	1982.10.22	信息管理	2005.09.01	050005	武晓	信息管理
050006	黄平	男	1983.06.06	信息管理	2005.09.01	050006	黄平	信息管理

② 选择运算 ( $\sigma$ ) ——Select

- 对单个关系的运算，从一个关系中选择出满足条件的元组构成新的关系，这个新关系是原关系的子集。是行方向的运算。
- 例如：选择开课学期是“一上”的课程

课程信息表

课程号	课程名称	考试方式	学分	学时数	周学时	上机时数	开课学期
060100	高等数学	考试	4	80	4	0	一上
050101	英语	考试	4	80	4	0	一上
050102	政治理论	考查	3	60	3	0	一下
050103	法律基础	考查	2	40	2	0	一下
050101	数据库原理	考试	4	80	4	32	三上
050101	数据结构	考试	4	80	4	30	三上

“一上”学期课程信息

课程号	课程名称	考试方式	学分	学时数	周学时	上机时数	开课学期
060100	高等数学	考试	4	80	4	0	一上
050101	英语	考试	4	80	4	0	一上

## ③ 影与选择的混合运算

R

学号	姓名	性别	出生年月	所学专业	入学日期
050001	李红	女	1983.01.03	信息管理	2005.09.01
050002	王名	男	1984.12.12	信息管理	2005.09.01
050003	刘丽	女	1982.07.08	信息管理	2005.09.01
050005	张力	男	1981.09.12	信息管理	2005.09.01
050005	武晓	男	1982.10.22	计算机	2005.09.01
050006	黄萍	女	1983.06.06	计算机	2005.09.01

1、信息管理专业的女生？

2、信息管理专业女生的学号和姓名？



• 投影与选择的混合运算:

$\sigma_{\text{所学专业} = '信息管理' \wedge \text{性别} = '女'} (R)$

信息专业的女生

$\Pi_{\text{学号, 姓名}} (\sigma_{\text{所学专业} = '信息管理' \wedge \text{性别} = '女'} (R))$

信息专业女生的学号和姓名

**R**

学号	姓名	性别	出生年月	所学专业	入学日期
050001	李红	女	1983.01.03	信息管理	2005.09.01
050002	王名	女	1982.07.08	信息管理	2005.09.01
050003	张力	男	1984.12.12	信息管理	2005.09.01
050005	武晓	男	1981.09.12	信息管理	2005.09.01
050006	黄萍	女	1982.10.22	计算机	2005.09.01
050007	陈华	女	1983.06.06	计算机	2005.09.01

- 投影与选择的混合运算:

④等值连接: 等值连接是在两个关系的笛卡尔积上进行的选择运算, 记做:  $R \bowtie_{A=B} S$

其中: A是关系R中的属性, B是关系S中的属性;  
 $\Theta \in \{=, \geq, <, \leq, \neq\}$

$$R \bowtie_{A=B} S = \sigma_{A=B} (R \times S)$$

- ⑤自然连接: 要求两个关系中进行比较的分量必须是相同的属性组, 并且在结果中去掉重复的属性列。

**R**

学号	姓名	性别	所学专业
050001	李红	女	信息管理
050002	王名	男	信息管理

**S**

学号	姓名	课程名称	成绩
050001	李红	C语言	89
050001	李红	数据库	82
050001	李红	计算机	78
050002	王名	C语言	82
050002	王名	数据库	77
050002	王名	计算机	92

第一步: 连接



$$R \bowtie S$$

学号	姓名	性别	所学专业	学号	姓名	课程名称	成绩
050001	李红	女	信息管理	050001	李红	C语言	89
050001	李红	女	信息管理	050001	李红	数据库	82
050001	李红	女	信息管理	050001	李红	计算机	78
050001	李红	女	信息管理	050002	王名	C语言	82
050001	李红	女	信息管理	050002	王名	数据库	77
050001	李红	女	信息管理	050002	王名	计算机	92
050002	王名	男	信息管理	050001	李红	C语言	89
050002	王名	男	信息管理	050001	李红	数据库	82
050002	王名	男	信息管理	050001	李红	计算机	78
050002	王名	男	信息管理	050002	王名	C语言	82
050002	王名	男	信息管理	050002	王名	数据库	77
050002	王名	男	信息管理	050002	王名	计算机	92

第二步：条件选择

$$R \bowtie S$$

条件：学号=学号

学号	姓名	性别	所学专业	学号	姓名	课程名称	成绩
050001	李红	女	信息管理	050001	李红	C语言	89
050001	李红	女	信息管理	050001	李红	数据库	82
050001	李红	女	信息管理	050001	李红	计算机	78
050001	李红	女	信息管理	050002	王名	C语言	82
050001	李红	女	信息管理	050002	王名	数据库	77
050002	王名	男	信息管理	050001	李红	C语言	89
050002	王名	男	信息管理	050001	李红	数据库	82
050002	王名	男	信息管理	050001	李红	计算机	78
050002	王名	男	信息管理	050002	王名	C语言	82
050002	王名	男	信息管理	050002	王名	数据库	77
050002	王名	男	信息管理	050002	王名	计算机	92

第三步：去掉重复属性

$$R \bowtie S$$

条件：学号=S.学号

学号	姓名	性别	所学专业	学号	姓名	课程名称	成绩
050001	李红	女	信息管理	050001	李红	C语言	89
050001	李红	女	信息管理	050001	李红	数据库	82
050001	李红	女	信息管理	050001	李红	计算机	78
050002	王名	男	信息管理	050002	王名	C语言	82
050002	王名	男	信息管理	050002	王名	数据库	77
050002	王名	男	信息管理	050002	王名	计算机	92

学号	姓名	性别	所学专业
050001	李红	女	信息管理
050002	王名	男	信息管理

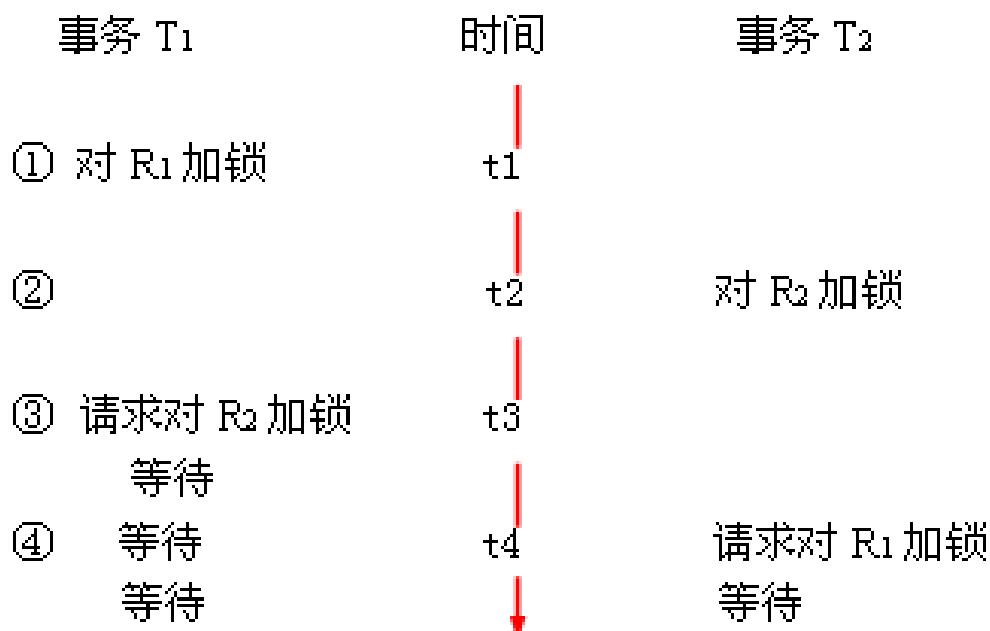
  

学号	姓名	课程名称	成绩
050001	李红	C语言	89
050001	李红	数据库	82
050001	李红	计算机	78
050002	王名	C语言	82
050002	王名	数据库	77
050002	王名	计算机	92



## 习题解析

1. 关系数据模型的三要素是（ ）。
  - A. 关系数据结构、关系操作集合和关系规范化理论
  - B. 关系数据结构、关系规范理论和关系完整性约束
  - C. 关系规范理论、关系操作集合和关系完整性约束
  - D. 关系数据结构、关系操作集合和关系完整性约束
2. 在数据库设计中用关系模型表示实体和实体之间的联系。关系模型的结构是（ ）。
  - A. 层次结构
  - B. 二维表结构
  - C. 网状结构
  - D. 封装结构
3. 如果把学生看成实体，某个学生的姓名叫“张三”则张三应看成是（ ）。
  - A. 记录型
  - B. 记录值
  - C. 属性型
  - D. 属性值
4. 设有关系 SC (SNO, CNO, GRADE)，主码是 (SNO, CNO) 遵照实体完整性规则（ ）。
  - A. 只有 SNO 不能取空值
  - B. 只有 CNO 不能取空值
  - C. 只有 GRADE 不能取空值
  - D. SNO 与 CNO 都不能取空值
5. 设关系 R 和 S 的元数分别是 r 和 s，且 R 有 n 个元组，S 有 m 个元组。执行关系 R 和 S 的笛卡儿积，记为 T=R×S，则（ ）。
  - A. T 的元数是 r+s，且有 n+m 个元组
  - B. T 的元数是 r+s，且有 n×m 个元组
  - C. T 的元数是 r×s，且有 n+m 个元组
  - D. T 的元数是 r×s，且有 n×m 个元组
6. 对表进行水平方向的分割用的运算是（ ）。
  - A. 交
  - B. 投影
  - C. 选择
  - D. 连接
7. 对表进行垂直方向的分割用的是（ ）。
  - A. 交
  - B. 投影
  - C. 选择
  - D. 连接

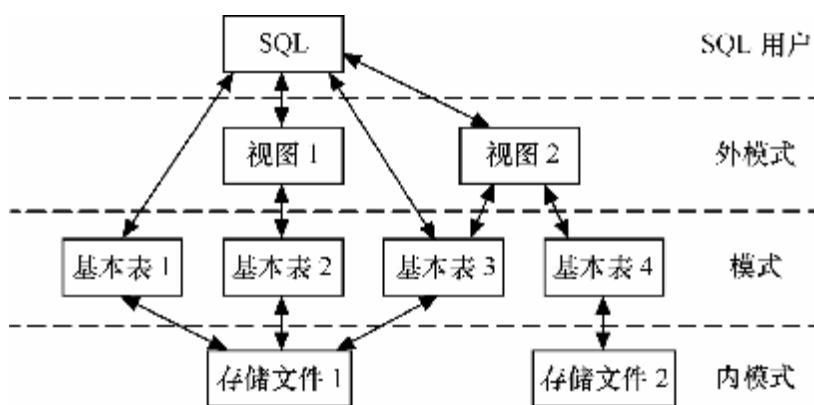


## 第 3 章 关系数据库标准语言 SQL

主要内容：

1. 基本表的定义、删除及修改
2. 索引的建立与删除
3. SQL 的数据查询
4. 数据更新
5. 视图

SQL 对关系数据库模式的支持



- SQL 语言的命令动词

SQL 功 能	动 词
数 据 定 义	CREATE, DROP, ALTER
数 据 查 询	SELECT
数 据 操 纵	INSERT, UPDATE DELETE
数 据 控 制	GRANT, REVOKE

- SQL 的数据类型

Ø 数值型: Int, Smallint, Bit, float 等

Ø 字符串型: Char(n), Varchar(n), Text, Binary(n),

image 等

Ø 日期时间型: DateTime, Smalldatetime 等。

Ø 货币型: Money, Smallmoney 等

## 1. 基本表的定义、删除及修改

①表的定义（默认情况下，SQL 不区分大小写）

```
CREATE TABLE <表名> (
  <列名> <数据类型> [列级完整性约束定义]
  {, <列名> <数据类型> [列级完整性约束定义] ... }
  [, 表级完整性约束定义 ] )
```

- 在列级完整性约束定义处可定义的约束

NOT NULL: 限制列取值非空。

DEFAULT: 给定列的默认值。

UNIQUE: 限制列取值不重。

CHECK: 限制列的取值范围。

PRIMARY KEY: 指定本列为主键。

FOREIGN KEY: 定义本列为引用其他表的外键。

例: CREATE TABLE SC (

```

Sno    CHAR(7) NOT NULL,
Cno    CHAR(6) NOT NULL,
Grade  INT,
PRIMARY KEY (Sno, Cno),
FOREIGN KEY (Sno) REFERENCES Student(Sno),
FOREIGN KEY (Cno) REFERENCES Course(Cno)
)

```

### ②修改表结构

ALTER TABLE 语句可以对表添加列、删除列、修改列的定义、定义主键、外键，也可以添加和删除约束。

ALTER TABLE <表名>

```

[ ALTER COLUMN <列名> <新数据类型>]
| [ ADD <列名> <数据类型> [约束]]
| [ DROP COLUMN <列名> ]
| [ ADD [constraint <约束名>] 约束定义]
| [ DROP [constraint] <约束名>]

```

例: 在 S 表中增加一个入学时间, 为日期型

ALTER TABLE S ADD SCOME DATE;

例: 删除对 SN 列的唯一约束

ALTER TABLE S DROP UNIQUE(SN);

### ③基本表的删除

DROP TABLE <表名>;

### 习题解析

1. SQL 语言集数据查询、数据操纵、数据定义和数据控制功能于一体，语句 CREATE、DROP、ALTER 实现哪类功能（ ）。

- A. 数据查询      B. 数据操纵
- C. 数据定义      D. 数据控制

2. 下列 SQL 语句中修改表结构的是( )。 A. ALTER      B. CREATE      C. INSERT



D. UPDATE

3. 在 SQL 中基本表的撤消从数据库中删除表可以用( )。

- A. DROP SCHEMA 命令    B. DROP TABLE 命令  
C. DROP VIEW 命令    D. DROP INDEX 命令

## 2. 索引的建立与删除

建立索引是加快查询速度的有效手段，用户可以根据需要为表中的单列（或多列）建立索引，也可以在一个表上建立一个或多个索引，以提供多种存取路径，从而加快查询速度。



### ①索引的建立

```
CREATE [UNIQUE] [CLUSTER] INDEX <索引名> ON <表名>(<列名>[<次序>][,<列名>[<次序>]]...);
```

### ②索引的删除

```
DROP INDEX <索引名>;
```

### 习题解析

1. 在 SQL 中，建立索引用( )。

- A. CREATE SCHEMA 命令    B. CREATE TABLE 命令  
C. CREATE VIEW 命令    D. CREATE INDEX 命令

2. 基于“学生—选课—课程”数据库中的三个关系是 S(S#, SNAME, SEX, AGE), SC(S#, C#, GRADE),

C(C#, CNAME, TEACHER)为了提高查询学生成绩的查询速度，对关系 SC 创建唯一索引，应该创建在哪一个(组)属性上( )。

- A. S#    B. C#    C. GRADE    D. (S#, C#)





### 3.SQL 的数据查询

①SELECT 语句的一般格式

SELECT <目标列名序列>

FROM <数据源>

[ WHERE <条件表达式> ]

[ GROUP BY <列名 1> [ HAVING <条件表达式> ] ]

[ ORDER BY <列名 2> [ ASC|DESC ] ];

② 表查询

查询仅涉及一个表或一个视图，是一种最简单的查询操作。WHERE 子句常用的查询条件：

查询条件	谓词
比较	=, >, <, >=, <=, <> (或!=), NOT+上述比较运算符
确定范围	BETWEEN AND, NOT BETWEEN AND
确定集合	IN, NOT IN
字符匹配	LIKE, NOT LIKE
空值	IS NULL, IS NOT NULL
逻辑运算	AND, OR, NOT

对查询结果排序：

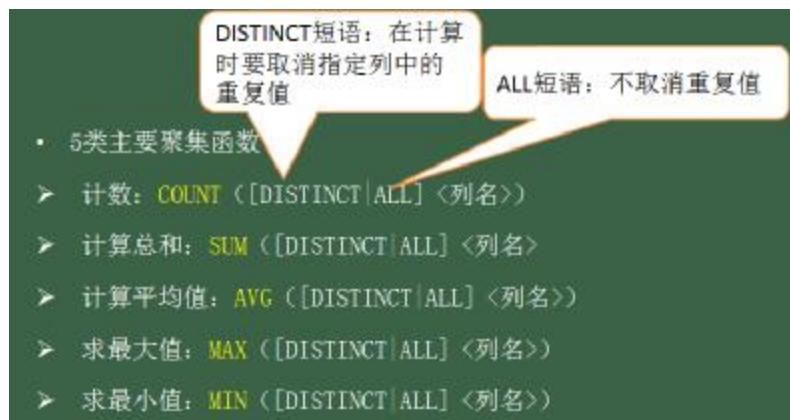
使用 ORDER BY 子句，语法格式为：

ORDER BY <列名> [ASC | DESC] [, …]

Ø 可以按一个或多个属性列排序

Ø 升序：ASC；降序：DESC；缺省值为升序





DISTINCT短语：在计算时要取消指定列中的重复值  
 ALL短语：不取消重复值

- 5类主要聚集函数
- 计数: COUNT ([DISTINCT|ALL] <列名>)
- 计算总和: SUM ([DISTINCT|ALL] <列名>)
- 计算平均值: AVG ([DISTINCT|ALL] <列名>)
- 求最大值: MAX ([DISTINCT|ALL] <列名>)
- 求最小值: MIN ([DISTINCT|ALL] <列名>)

对查询结果分组

### Ø GROUP BY 子句

含义：用 GROUP BY 子句将查询的结果按指定的列进行分组，即将指定列值相同的元组分为同一个组。

### Ø HAVING 短语

含义：按一定的条件对分组进行筛选，即把符合指定条件的分组进行输出

例：关系结构如 S(S#, SNAME, SEX, AGE), SC(S#, C#, GRADE), C(C#, CNAME, TEACHER)查询至少有 3 门以上课程成绩在 90 分以上的学号。

```
SELECT S#
  FROM SC
 WHERE GRADE>=90
 GROUP BY S# HAVING COUNT(*)>3;
```

### ③连接查询

若一个查询涉及两个或两个以上的表，称为连接查询。

用来连接两个表的条件称为连接条件或连接谓词。

连接条件一般格式：

### Ø [<表名 1>.]<列名 1> <比较运算符> [<表名 2>.]<列名 2>

比较运算符：=、>、<、>=、<=、!=

### Ø [<表名 1>.]<列名 1> BETWEEN [<表名 2>.]<列名 2> AND [<表名 2>.]<列名 3>

例：S(S#, SNAME, SEX, AGE), SC(S#, C#, GRADE), C(C#, CNAME, TEACHER)求选课情况，要求输出学号、姓名、课程名与成绩。

```
SELECT S.S#, SNAME, CNAME, GRADE
```

```
  FROM S, C, SC
```



WHERE S. S#=SC. S# AND C. C#=SC. C#;

#### ④嵌套查询

一个 SELECT-FROM-WHERE 语句称为一个查询块。

将一个查询块嵌套在另一个查询块的 WHERE 子句或 HAVING 短语的条件中的查询称为嵌套查询。

#### 习题解析

1. 在 SQL 语句的 SELECT 语句中, 用于对结果元组进行排序的是哪一个子句 ( )。

- A. GROUP BY                    B. HAVING  
C. ORDER BY                    D. WHERE

2. 在用 SELECT 语句进行数据库查询时, 可使用( )子句给出查询选择的条件。

3. 在 SQL 语言的 SELECT 语句中, 实现投影操作的是哪个子句 ( )。

- A. SELECT                    B. FROM                    C. WHERE                    D. GROUP BY

4. 用 GROUP BY 语句对数据分组时, 在分组中用来进行条件选择的语句是( )。

- A. SELECT                    B. WHERE                    C. HAVING                    D. ORDER BY

第 5~6 题基于学生-课程数据库中的三个基本表:

Ø 学生信息表: S (sno, sname, sex, age, dept) 主码为 sno

Ø 课程信息表: C (cno, cname, teacher) 主码为 cno

Ø 学生选课信息表: SC (sno, cno, grade) 码为(sno, cno)

5. “从学生选课信息表中找出无成绩的元组” 的 SQL 语句是 ( )。

- A. SELECT\*FROM SC WHERE GRADE=NULL  
B. SELECT\*FROM SC WHERE GRADE IS ''  
C. SELECT\*FROM SC WHERE GRADE IS NULL  
D. SELECT\*FROM SC WHERE GRADE=''

6. 在“查询选修课程号为 4, 且成绩在 80 分以上的所有学生的学号和姓名”的 SQL 语句中, 将使用的表是 ( )。

- A. 学生信息表 S  
B. 学生信息表 S 和课程信息表 C  
C. 学生信息表 S、课程信息表 C 和学生选课信息表 SC  
D. 学生信息表 S 和学生选课信息表 SC

## 4.数据更新

### ①插入数据

插入单个元组：在指定表中插入一条新元组。

```
INSERT INTO <表名> [<属性列 1>[, <属性列 2>…)]VALUES (<常量 1> [, <常量 2>] … )
```

例：新增一个学生信息，新增一条选课记录

```
INSERT INTO S VALUES('990021', '陈冬', '男', 18);
```

```
INSERT INTO SC (S#, C#) VALUES ('9807121', 'C175');
```

将子查询结果插入指定表中

```
INSERT INTO <表名> [<属性列 1> [, <属性列 2>… ]]
```

子查询；

②修改数据：修改指定表中满足 WHERE 子句条件的元组。

```
UPDATE <表名>
```

```
SET <列名>=<表达式>[, <列名>=<表达式>]…
```

```
[WHERE <条件>];
```

例：将所有学生的年龄增加 1 岁

```
UPDATE S SET AGE=AGE+1;
```

### ③ 删除数据

删除指定表中满足 WHERE 子句条件的元组，若不带 WHERE 子句，将删除表中所有元组，使表变为空表。

```
DELETE FROM <表名>
```

```
[WHERE <条件>];
```

例：删除学号为 ‘95019’ 的学生。

```
DELETE FROM SC WHERE S# = '95019';
```

```
DELETE FROM S WHERE S# = '95019';
```

例：删除选课但无成绩的学生的选课信息

```
DELETE FROM SC WHERE GRADE IS NULL;
```

## 5.视图

视图的特点：





- Ø 虚表，是从一个或几个基本表（或视图）导出的表
- Ø 只存放视图的定义，不会出现数据冗余
- Ø 基本表中的数据发生变化，从视图中查询出的数据也随之改变

①新建视图

```
CREATE VIEW <视图名> [(<列名>[, <列名>]…)]
```

```
AS <SELECT 子查询 >
```

②删除视图

```
DROP VIEW <视图名>;
```

例：建立计算机系学生视图

```
CREATE VIEW CS_S
```

```
AS (SELECT S#, SN, SEX, SA FROM S WHERE SD= 'CS' )
```

- 视图的作用
  - Ø 能够简化用户的操作
  - Ø 使用户能以多种角度看待同一数据
  - Ø 对重构数据库提供了一定程度的逻辑独立性
  - Ø 视图能够对机密数据提供安全保护

习题解析

1. SQL 语言中的“视图（VIEW）”对应于数据库三级模式结构的（ ）。

- A. 外模式 B. 模式 C. 内模式 D. 概念模式

2. 在 SQL 中建立视图用（ ）。

- A. CREATE SCHEMA 命令 B. CREATE TABLE 命令  
C. CREATE VIEW 命令 D. CREATE INDEX 命令

3. 在 SQL 的删除语句 DELETE 中，如果省略（ ）子句，则会删除表中所有的记录。

4. 视图机制具有很多优点，下面的（ ）不是它的优点。

- A. 视图对数据库的重构提供了一定程度的逻辑独立性  
B. 简化了用户观点  
C. 方便了不同的用户以同样的方式看待同一数据  
D. 对机密数据提供了自动安全防护功能



## 第4章 数据库安全性

数据库的安全性是指保护数据库,防止被不合法的使用所造成的数据泄露、更改或破坏。

DBMS 自主存取控制机制(授权机制)是数据库安全的一个重要保证,它确保具有数据库使用权限的用户访问数据库并进行权限范围内的操作。

DBMS 自主存取控制机制由以下几部分构成:

(1) 定义用户权限:在 SQL 中是由 GRANT 和 REVOKE 语句实现。

(2) 合法权限检查

①授权(GRANT)

```
GRANT <权限 1>[, <权限 2> ... ]  
ON [<对象类型>] <对象名> [, [<对象类型>] <对象名>... ]  
TO <用户 1> [, <用户 2> ... ]
```

例: 把查询 Student 表和修改姓名 (Sname) 列的权限授予用户 user1。

```
GRANT UPDATE(Sname), SELECT ON Student  
TO user1;
```

②收回权限(REVOKE)

```
REVOKE <权限 1>[, <权限 2> ... ]  
ON [<对象类型>] <对象名> [, [<对象类型>] <对象名>... ]  
FROM <用户 1> [, <用户 2> ... ];
```

例: 把用户 user1 修改姓名的权限收回。

```
REVOKE UPDATE(Sname) ON Student  
FROM user1;
```

习题解析

1. SQL 语言的 REVOKE 语句实现下列哪一种数据控制功能 ( )。

- A. 可靠性控制
- B. 并发性控制
- C. 安全性控制
- D. 完整性控制

2. 安全性控制的防范对象是 ( ), 防止他们对数据库数据的存取。

- A. 不合语义的数据
- B. 非法用户
- C. 不正确的数据
- D. 不符合约束数据



## 第5章 数据库完整性

- 数据库的完整性是指数据的正确性和相容性。它是防止不合语义的数据进入数据库，是确保数据库中数据一致、正确及符合企业规则的一种思想和手段。
- 完整性约束条件: 加在数据库数据之上的语义约束条件
- 完整性检查: DBMS 中检查数据是否满足完整性条件的机制称为完整性检查。
- DBMS 的完整性控制机制具有以下功能。
  - (1) 定义完整性约束条件。
  - (2) 完整性检查。
  - (3) 违约处理。
    - SQL 提供了以下几种完整性约束:
      - (1) PRIMARY KEY (主键) 约束
      - (2) FOREIGN KEY (外键) 约束
      - (3) UNIQUE (唯一) 约束
      - (4) CHECK (检查) 约束
      - (5) NOT NULL (非空) 约束
      - (6) DEFAULT (默认值) 约束
    - 删约束:

ALTER TABLE <表名>

DROP CONSTRAINT <约束名>;

- 触发器可以实现比 FOREIGN KEY、CHECK 约束更加复杂的完整性，具有更精细和更强大的数据控制能力。

### 习题解析

下述 SQL 语言中的权限，哪一个允许用户定义新关系时，引用其他关系的主码作为外码（ ）。

- A. INSERT      B. DELETE  
C. REFERENCES    D. SELECT

## 第6章 关系数据理论

主要内容:

1. 函数依赖



## 2. 关系规范化

### 1. 函数依赖

函数依赖：设一个关系模式  $R(U)$ ， $R$  是关系的名称， $U$  是关系  $R$  的所有属性的集合， $X$  和  $Y$  为属性集  $U$  上的子集。若对于任一元组在  $X$  上的每一个值，都有  $Y$  上的一个唯一值与之对应，则称  $X$  函数决定  $Y$ ，或称  $Y$  函数依赖于  $X$ ，记作  $X \rightarrow Y$ 。 $X$  称为这个函数依赖的决定因子。 $Y$  不函数依赖于  $X$  记作： $X \not\rightarrow Y$

#### 部分函数依赖

若  $X \rightarrow Y$ ，同时  $X$  的一个真子集  $X'$  且也能够函数决定  $Y$ ，即存在  $X' \rightarrow Y$ ，则称  $X$  部分函数决定  $Y$ ，或  $Y$  部分函数依赖于  $X$ ，记作： $X \xrightarrow{P} Y$ 。

例：设有关系模式  $SC(Sno, Sname, Cno, Grade)$ ，其中各属性分别为：学号，姓名，课程号，成绩，主码为  $(Sno, Cno)$ 。则  $(Sno, Cno) \rightarrow Sname$

#### 完全函数依赖

若不存在一个真子集  $X'$ ，使得  $X'$  也能够函数决定  $Y$ ，则称  $X$  完全函数决定  $Y$ ，或者  $Y$  完全函数依赖于  $X$ 。记作： $X \xrightarrow{F} Y$ 。

上例中  $(Sno, Cno) \rightarrow Grade$

传递函数依赖：若  $X \rightarrow Y$ ,  $Y \rightarrow Z$ , 且  $Y \rightarrow Z$ ，则存在  $X \xrightarrow{t} Z$ ，称  $Z$  传递函数依赖于  $X$ 。

## 2. 关系规范化

候选码：设  $K$  为  $R(U)$  中的属性或属性组，若  $K \xrightarrow{F} U$ ，则  $K$  为  $R$  的候选码。可以一个或者多个。

- 第一范式(1NF)：如果一个关系  $R$  的所有属性都是不可再分的基本数据项，则称关系  $R$  属于第一范式，记作  $R \in 1NF$ 。
- 第二范式(2NF)：若关系模式  $R \in 1NF$ ，并且每一个非主属性都完全函数依赖于  $R$  的主码，则  $R \in 2NF$ 。
- 第三范式(3NF)：如果关系  $R \in 2NF$ ，且每一个非主属性都不传递依赖于主码，则  $R$  属于第三范式，记作  $R \in 3NF$ 。
- BCNF 范式：如果关系  $R \in 1NF$ ，且  $R$  中每一个决定因子都是候选键，则  $R$  属于 BC 范式，记作  $R \in BCNF$ 。

### 习题解析

- 规范化理论是关系数据库进行逻辑设计的理论依据，根据这个理论，关系数据库中的关系必须满足：每一个属性都是（ ）。

- A. 长度不变的 B. 不可分解的  
C. 互相关联的 D. 互不相关的
2. 关系模式中，满足 2NF 的模式（ ）。  
A. 可能是 1NF B. 必定是 1NF  
C. 必定是 3NF D. 必定是 BCNF
3. 根据关系数据库规范化理论，关系数据库中的关系要满足第一范式，部门（部门号，部门名，部门成员，部门总经理）关系中，因哪个属性而使它不满足第一范式（ ）。  
A. 部门总经理 B. 部门成员 C. 部门名 D. 部门号
4. 关系的规范化中，各个范式之间的关系是（ ）。  
A.  $1NF \in 2NF \in 3NF$  B.  $3NF \in 2NF \in 1NF$   
C.  $1NF=2NF=3NF$  D.  $1NF \in 2NF \in BCNF \in 3NF$
5. 设有关系模式 R (S, D, M)，其函数依赖集：F={S→D, D→M}，则关系模式 R 的规范化程度最高达到（ ）。  
A. 1NF B. 2NF C. 3NF D. BCNF

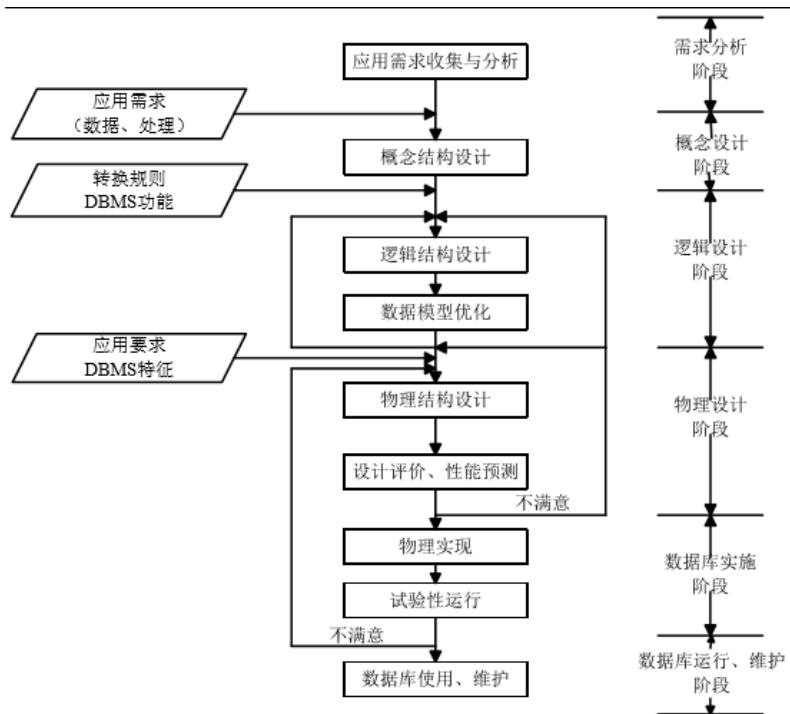
## 第 7 章 数据库设计

- 数据库设计概述

数据库设计的广义定义是指基于数据库的应用系统或管理信息系统的应用设计和数据库结构设计两部分内容。

数据库设计的狭义定义则专指数据库模式或结构的设计。

- 通过分析、比较与综合各种常用的数据库规范设计方法，将数据库设计分为以下 6 个阶段。需求分析阶段、概念结构设计阶段、逻辑结构设计阶段、物理结构设计阶段、数据库实施阶段、数据库运行与维护阶段。



## 1. 数据库需求分析

需求分析阶段的主要任务是对现实世界要处理的对象（公司，部门，企业）进行详细调查，在了解现行系统的概况、确定新系统功能的过程中，收集支持系统目标的基础数据及其处理方法。

需求调查内容：信息需求、处理需求、安全性与完整性要求

2. 概念设计阶段：分析数据间内在的语义关联，在此基础上建立一个数据的抽象模型——概念数据模型。概念数据模型是根据用户需求设计出来的，它不依赖于任何的数据库管理系统。

概念数据模型设计的描述最常用的工具是 E-R 图（实体-联系图）。

3. 逻辑设计阶段：首先将 E-R 图转换成具体的数据库管理系统所支持的数据模型，如关系模型，形成数据库逻辑模式；然后根据用户处理的要求、安全性的考虑，在基本表的基础上再建立必要的视图（View），形成数据的外模式。

## 4. 物理设计阶段：

根据数据库管理系统的特性和处理的需要，进行物理存储安排，建立索引，形成数据库内模式。

## 5. 数据库实施

Ø 定义数据库结构：用某一具体的数据库管理系统提供的数据定义语言严格描述数据库结构。

Ø 组织数据录入数据库：数据库结构建立完成后，便可以将原始数据载入到数



据库中。

- Ø 编写和调试应用程序：编写程序时可用一些模拟数据进行程序调试，待程序编写完成方可正式输入数据。
- Ø 数据库试运行：在这个阶段，最好常对数据库中的数据进行备份的操作，因为，调试期间系统不稳定，容易破坏已存在的数据信息。

6. 数据库的运行和维护，包括：

- Ø 数据库的备份和恢复
- Ø 数据库的安全性和完整性控制
- Ø 监视、分析、调整数据库性能
- Ø 数据库的重组

习题解析

1. 数据库设计中，用 E-R 图来描述信息结构但不涉及信息在计算机中的表示，这是数据库设计的（ ）。  
A. 需求分析阶段    B. 逻辑设计阶段  
C. 概念设计阶段    D. 物理设计阶段
2. 在数据库设计中，将 E-R 图转换成关系数据模型的过程属于（ ）。  
A. 需求分析阶段    B. 逻辑设计阶段  
C. 概念设计阶段    D. 物理设计阶段
3. 数据库设计可划分为六个阶段，每个阶段都有自己的设计内容，“为哪些关系，在哪些属性上、键什么样的索引”这一设计内容应该属于（ ）设计阶段。  
A. 概念设计    B. 逻辑设计  
C. 物理设计    D. 全局设计

## 第 8 章 并发控制

主要内容：

1. 事务基本概念    2. 并发控制概述
3. 并发控制措施    4. 封锁协议
5. 死锁    6. 两阶段锁



## 1. 事务基本概念

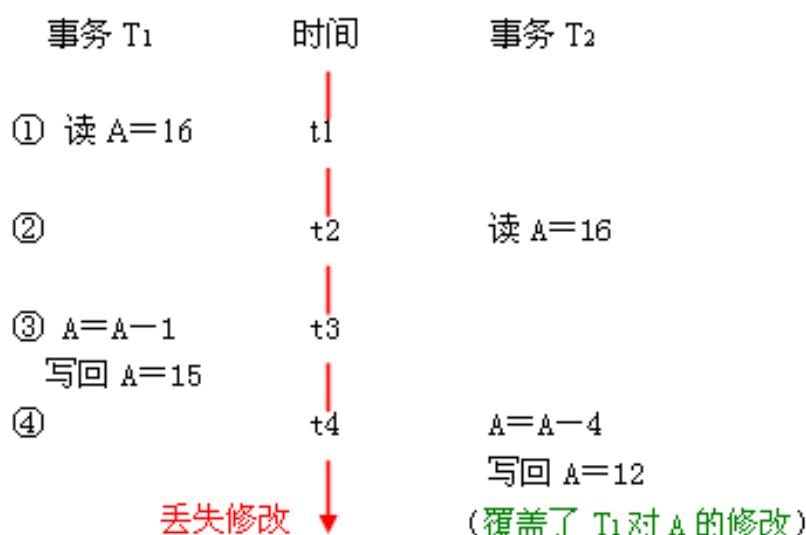
事务是用户定义的数据操作系列，这些操作作为一个完整的工作单元，一个事务内的所有语句被作为一个整体，要么全部执行，要么全部不执行。

- 事务的特性 (ACID 特性)
  - Ø 原子性 (Atomicity): 指事务是数据库的逻辑工作单位，事务中的操作要么都做，要么都不做。
  - Ø 一致性 (Consistency): 指事务执行的结果必须是使数据库从一个一致性状态变到另一个一致性状态。
  - Ø 隔离性 (Isolation): 指数据库中一个事务的执行不能被其它事务干扰。
  - Ø 持久性 (Durability): 指事务一旦提交，其对数据库数据的改变就是永久的。
- 当开始一个事务后，系统必须连续地执行全部后续的 SQL 语句，直到出现下列四个事件之一：
  - Ø 到达了 COMMIT 语句。自动结束一个事务并表明成功地完成了事务。
  - Ø 到达了 ROLLBACK 语句。表明没有成功地完成事务。数据库被回滚到之前的一个一致性状态。
  - Ø 成功地到达了程序的结束。
  - Ø 程序被异常终止了。
- 事务日志：为支持事务处理，DBMS 对数据库所做的每个更改操作都维护一个事务记录，并保存到事务日志中。
- DBMS 用事务日志来跟踪所有影响数据库值的操作，以使 DBMS 能够从由事务引起的失败中恢复数据库。

## 2. 并发控制概述

并发操作所带来的数据不一致情况大致为四种：

- ① 失修改

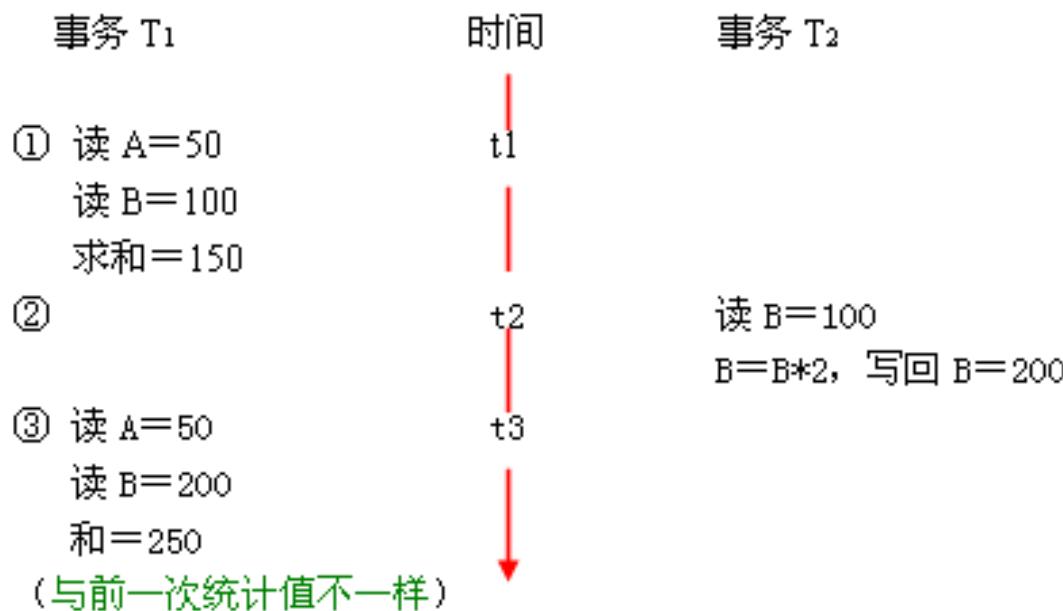


② “脏”数据



③ 可重复读





④产生“幽灵”数据：属于不可重复读的范畴。

指当事务 T1 按一定条件从数据库中读取了某些数据记录后，事务 T2 删除了其中的部分记录，或者在其中添加了部分记录，当 T1 再次按相同条件读取数据时，发现其中少了（对删除）或多了（对插入）一些记录。

### 3. 并发控制措施

进行并发控制的主要方式是使用封锁机制，即加锁。

锁有两种基本类型：

①共享锁 (Share Locks, 也称 S 锁或读锁)。若事务 T 给数据项 A 加了 S 锁，则事务 T 可以读 A，但不能修改 A，其他事务可以再给 A 加 S 锁，但不能加 X 锁，直到 T 释放了 A 上的 S 锁为止。

②排它锁 (Exclusive Locks, 也称为 X 锁或写锁)。若事务 T 给数据项 A 加了 X 锁，则允许 T 读取和修改 A，但不允许其他事务再给 A 加任何类型的锁和进行任何操作。

## 4. 封锁协议

①一级封锁协议：对事务 T 要修改的数据加 X 锁，直到事务结束（包括正常结束和非正常结束）时才释放。一级封锁协议可以防止丢失修改，并保证事务 T 是可恢复的。

在一级封锁协议中，如果事务 T 只是读数据而不修改数据，则不需要加锁。所以不能保证可重复读和不读“脏”数据。

②二级封锁协议：一级封锁协议加上对事务 T 对要读取的数据加 S 锁，读完后即释放 S 锁。

除了可以防止丢失修改外，还可以防止读“脏”数据。

但不能保证可重复读数据

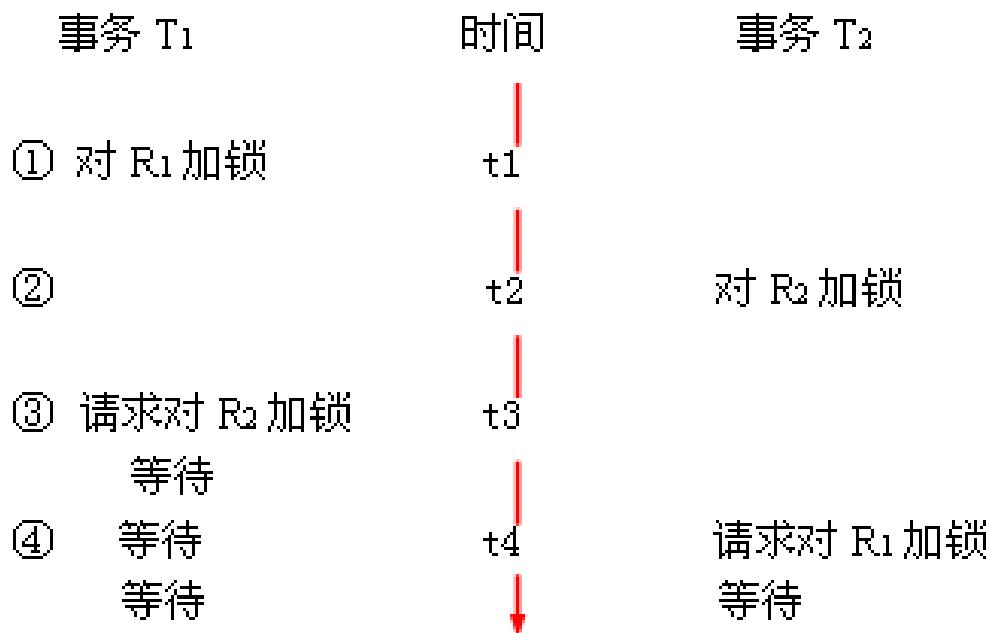


③三级封锁协议：一级封锁协议加上事务 T 对要读取的数据加 S 锁，并直到事务结束才释放。

除了可以防止丢失修改和不读“脏”数据之外，还进一步防止了不可重复读。

## 5.死锁：

两个事务相互等待对方先释放资源



- 预防死锁
  - Ø 一次封锁法：每个事务一次将所有要使用的数据全部加锁。
  - Ø 顺序封锁法：预先对数据对象规定一个封锁顺序，所有事务都按这个顺序封锁

## 6.两阶段锁

- Ø 两阶段锁是控制并发处理的一个方法或一个协议，也称为两段锁协议。
- Ø 在对任何数据进行读、写操作之前，首先要封锁该数据。
- Ø 在释放一个封锁之后，事务不再申请和获得任何其它封锁。
- Ø 事务遵守两段锁协议是可串行化调度的充分条件，而不是必要条件。

例：事务 1 的封锁序列：

Lock A ... Lock B ... XLock C ... Unlock C ... Unlock B ... Unlock A;

事务 2 的封锁序列：

SIlock A ... Unlock A ... SIlock B ... XIlockC ... Unlock C ... Unlock B;

事务 1 遵守两段锁协议，而事务 2 不遵守两段协议。

#### 习题解析

1. 解决并发操作带来的数据不一致问题普遍采用（ ）技术。  
A. 封锁      B. 存取控制      C. 恢复      D. 协商
2. 下列不属于并发操作带来的问题是（ ）。  
A. 丢失修改      B. 不可重复读  
C. 死锁      D. 脏读
3. 如果事务 T 获得了数据项 Q 上的排他锁，则 T 对 Q ( )。  
A. 只能读不能写      B. 只能写不能读  
C. 既可读又可写      D. 不能读也不能写
4. 数据库中的封锁机制是（ ）的主要方法。  
A. 完整性      B. 安全性  
C. 并发控制      D. 恢复
5. 如果有两个事务，同时对数据库中同一数据进行操作，不会引起冲突的操作是（ ）。  
A. 一个是 DELETE，一个是 SELECT  
B. 一个是 SELECT，一个是 DELETE  
C. 两个都是 UPDATE  
D. 两个都是 SELECT

## 第 9 章 数据库恢复技术

主要内容：

1. 恢复的基本概念
2. 数据库故障的种类
3. 恢复方法

### 1. 恢复的基本概念

数据库恢复是指当数据库发生故障时，将数据库恢复到正确（一致性）状态的过程。  
数据库恢复是基于事务的原子性特性。

恢复机制有两个关键问题：如何建立备份数据；如何利用备份数据进行恢复。





数据库恢复采用的基本技术：数据转储（也称为数据库备份）。转储就是定期地将整个数据库复制到辅助存储设备上，比如磁带、磁盘。如果想恢复到故障发生时的状态，则必须利用转储之后的事务日志。

## 2. 数据库故障的种类

① 事务内部的故障

- Ø 可预期的：这类故障可通过事务程序本身发现。
- Ø 非预期性的：这类故障不能由应用程序来处理。

恢复策略：反向扫描日志文件并执行相应操作的逆操作。

② 系统故障：是指造成系统停止运转、系统要重启的故障。

恢复策略：恢复子系统在系统重新启动时必须：

- Ø 撤销所有未完成的事务
- Ø 重做所有已提交的事务

③ 其它故障：介质故障或由计算机病毒引起的故障或破坏，均归为其他故障。

- Ø 介质故障指外存故障，如磁盘损坏等。
- Ø 计算机病毒的破坏性很大，而且极易传播，它也可以对数据库造成毁灭性的破坏。

恢复策略：重装数据库管理系统，利用备份或镜像恢复数据库。

## 3. 恢复方法：

- (1) 利用备份技术
- (2) 利用事务日志
- (3) 利用镜像技术

习题解析

1. 若系统在运行过程中，由于某种硬件故障，使存储在外存上的数据部分损失或全部损失，这种情况称为（ ）。

- A. 介质故障
- B. 运行故障
- C. 系统故障
- D. 事务故障

2. 事务日志用于保存（ ）。

- A. 程序运行过程
- B. 程序的执行结果
- C. 对数据的更新操作
- D. 数据操作



3. 数据库恢复的基础是利用转储的冗余数据。这些转储的冗余数据包括（ ）。
- A. 数据字典、应用程序、审计档案、数据库后备副本
  - B. 数据字典、应用程序、审计档案、日志文件
  - C. 日志文件、数据库后备副本
  - D. 数据字典、应用程序、数据库后备副本



## ■ 华图网校介绍

华图网校（V.HUATU.COM）于2007年3月由华图教育投资创立，是华图教育旗下的远程教育高端品牌。她专注于公职培训，目前拥有遍及全国各地500万注册用户，已成为公职类考生学习提高的专业门户网站。

华图网校是教育部中国远程教育理事单位。她拥有全球最尖端高清录播互动技术和国际领先的网络课程设计思想，融汇华图教育十余年公职辅导模块教学法，凭借强大师资力量与教学资源、利用教育与互联网的完美结合，真正为考生带来“乐享品质”的学习体验，通过“高效学习”成就品质人生。

华图网校课程丰富多元，涵盖公务员、事业单位、招警、法院、检察院、军转干、选调生、村官、政法干警、三支一扶、乡镇公务员、党政公选等热门考试、晋升及选拔。同时，华图网校坚持以人为本的原则，不断吸引清华、北大等高端人才加入经营管理，优化课程学习平台，提升用户体验，探索网络教育新技术和教学思想，力争为考生提供高效、个性、互动、智能的高品质课程和服务。

华图网校将秉承“以教育推动社会进步”的使命，加快网站国际化进程，打造全球一流的网络学习平台。

我们的使命：以教育推动社会进步

我们的愿景：德聚最优秀人才，仁就基业长青的教育机构

我们的价值观：诚信为根、质量为本、知难而进、开拓创新。

- 咨询电话：400-678-1009
- 听课网址：[v.huatu.com](http://v.huatu.com)（华图网校）