

# 数据库原理与应用教程—SQL Server

## 习题解答

### 第一章 数据库系统概述

#### 1. 数据库的发展历史分哪几个阶段？各有什么特点？

答：数据库技术经历了人工管理阶段、文件系统阶段和数据库系统三个阶段。

##### 1) 人工管理阶段

这个时期数据管理的特点是：

数据由计算或处理它的程序自行携带，数据和应用程序一一对应，应用程序依赖于数据的物理组织，因此数据的独立性差，数据不能被长期保存，数据的冗余度大等给数据的维护带来许多问题。

##### 2) 文件系统阶段

在此阶段，数据以文件的形式进行组织，并能长期保留在外存储器上，用户能对数据文件进行查询、修改、插入和删除等操作。程序与数据有了一定的独立性，程序和数据分开存储，然而依旧存在数据的冗余度大及数据的不一致性等缺点。

##### 3) 数据库系统阶段

数据库系统的特点如下：

- (1) 数据结构化
- (2) 较高的数据共享性
- (3) 较高的数据独立性
- (4) 数据由 DBMS 统一管理和控制

#### 2. 简述数据、数据库、数据库管理系统、数据库应用系统的概念。

答：数据是描述事物的符号记录，是信息的载体，是信息的具体表现形式。

数据库就是存放数据的仓库，是将数据按一定的数据模型组织、描述和存储，能够自动进行查询和修改的数据集合。

数据库管理系统是数据库系统的核心，是为数据库的建立、使用和维护而配置的软件。它建立在操作系统的基础上，位于用户与操作系统之间的一层数据管理软件，它为用户或应用程序提供访问数据库的方法，包括数据库的创建、查询、更新及各种数据控制等。凡使用数据库技术管理其数据的系统都称为数据库应用系统。

#### 3. 简述数据库管理系统的功能。

答：数据库管理系统是数据库系统的核心软件，一般说来，其功能主要包括以下 5 个方面。

- (1) 数据定义和操纵功能
- (2) 数据库运行控制功能
- (3) 数据库的组织、存储和管理

(4) 建立和维护数据库

(5) 数据通信接口

4. 简述数据库的三级模式和两级映像。

答：为了保障数据与程序之间的独立性，使用户能以简单的逻辑结构操作数据而无需考虑数据的物理结构，简化了应用程序的编制和程序员的负担，增强系统的可靠性。通常 DBMS 将数据库的体系结构分为三级模式：外模式、模式和内模式。

模式也称概念模式或逻辑模式，是对数据库中全部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。

外模式也称子模式或用户模式，它是对数据库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。

内模式也称存储模式或物理模式，是对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。

三级模式结构之间差别往往很大，为了实现这 3 个抽象级别的联系和转换，DBMS 在三级模式结构之间提供了两级映像：外模式/模式映像，模式/内模式映像。

5. 简述数据库的逻辑独立性和物理独立性。

答：当模式改变时(如增加新的关系、新的属性、改变属性的数据类型等)，由数据库管理员对各个外模式/模式映像作相应改变，可以使外模式保持不变。应用程序是依据数据的外模式编写的，因而应用程序不必修改，保证了数据与程序的逻辑独立性，简称逻辑数据独立性。

当数据库的存储结构改变了(如选用了另一种存储结构)，由数据库管理员对模式/内模式映像作相应改变，可以保证模式保持小变，因而应用程序也不必改变。保证了数据与程序的物理独立性，简称物理数据独立性。

6. 简述几种数据库新技术的特点。

答：分布式数据库系统的主要特点是：

(1) 数据是分布的。

(2) 数据是逻辑相关的。

(3) 结点的自治性。

面向对象数据库系统强调在数据库框架中发展类型、数据抽象、继承和持久性。它的基本设计思想是，一方面把面向对象语言向数据库方向扩展，使应用程序能够存取并处理对象，另一方面扩展数据库系统，使其具有面向对象的特征，提供一种综合的语义数据建模概念集，以便对现实世界中复杂应用的实体和联系建模。

多媒体数据库系统 (Multi-media Database System, MDBS) 是数据库技术与多媒体技术相结合的产物。多媒体数据库不是对现有的数据进行界面上的包装，而是从多媒体数据与信息本身的特性出发，考虑将其引入到数据库中之后而带来的有关问题。

数据仓库 (Data Warehouse) 是一个面向主题的、集成的、相对稳定的、反映历史变化的数据集合，用于支持管理决策。

## 第二章 数据模型

### 1. 信息有哪三种世界，它们各有什么特点，它们之间有什么联系？

答：现实世界、信息世界和机器世界

现实世界就是存在于人脑之外的客观世界，客观事物及其相互联系就处于现实世界中。

信息世界就是现实世界在人们头脑中的反映，又称概念世界。客观事物在信息世界中称为实体，反映事物间联系的是实体模型或概念模型。

数据世界就是信息世界中的信息数据化后对应的产物。现实世界中的客观事物及其联系，在数据世界中以数据模型描述。

计算机信息处理的对象是现实生活中的客观事物，在对客观事物实施处理的过程中，首先要经历了解、熟悉的过程，从观测中抽象出大量描述客观事物的信息，再对这些信息整理、分类和规范，进而将规范化的信息数据化，最终由数据库系统存储、处理。

### 2. 什么是概念模型，什么是数据模型？

答：概念模型是现实世界的抽象反映，它表示实体类型及实体间的联系，是独立于计算机系统的模型，是现实世界到机器世界的一个中间层次。

数据模型是对客观事物及联系的数据描述，是概念模型的数据化，即数据模型提供表示和组织数据的方法。

### 3. 什么是实体、属性、码、联系？

答：客观存在并可以相互区分的事物叫实体。

属性是实体所具有的某些特性，通过属性对实体进行描述。

一个实体往往有多个属性，这些属性之间是有关系的，它们构成该实体的属性集合。如果其中有一个属性或属性集能够唯一标识整个属性集合，则称该属性或属性集为该实体的码。

现实世界的事物之间是有联系的，即各实体型之间是有联系的。就两个实体型的联系来说，主要有以下 3 种情况：一对一联系(1:1)、一对多联系(1:M)和多对多联系(M:N)。

### 4. 实体的联系有哪三种？

答：一对一联系(1:1)、一对多联系(1:M)和多对多联系(M:N)。

### 5. 分析层次模型、网状模型和关系模型的特点。

答：层次模型用树形结构来表示各类实体以及实体间的联系。每个结点表示一个记录类型，结点之间的连线表示记录类型间的联系，这种联系只能是父子联系。

层次模型存在如下特点：

(1) 只有一个结点没有双亲结点，称为根结点。

(2) 根结点以外的其他结点有且只有一个双亲结点。

网状数据模型是一种比层次模型更具普遍性的结构，它去掉了层次模型的两个限

制，允许多个结点没有双亲结点，也允许一个结点有多个双亲结点。因此，网状模型可以方便地表示各种类型的联系。网状模型是一种较为通用的模型，从图论的观点看，它是一个不加以任何条件的无向图。

用二维表格结构表示实体以及实体之间的联系的数据模型称为关系模型。关系模型在用户看来是一个二维表格。

6. 解释关系模型的基本概念：关系、元组、属性、域、关系模式、候选关键字、主键、外键、主属性。

答：关系：一个关系就是一张二维表。

元组：二维表中的行称为元组，每一行是一个元组。

属性：二维表的列称为属性，每一列有一个属性名，属性值是属性的具体值。

域：是属性的取值范围。

关系模式：对关系的信息结构及语义限制的描述称为关系模式，用关系名和包含的属性名的集合表示。

候选关键字：如果在一个关系中，存在多个属性(或属性组合)都能用来唯一标识该关系中的元组，这些属性(或属性组合)都称为该关系的候选关键字或候选码，候选码可以有多个。

主键：在一个关系的若干候选关键字中，被指定作为关键字的候选关键字称为该关的主键或主码。

主属性：在一个关系中，包含在任何候选关键字中的各个属性称为主属性。

外键：一个关系的某个属性(或属性组合)不是该关系的主键或只是主键的一部分，却是另一个关系的主码，则称这样的属性为该关系的外键或外码。

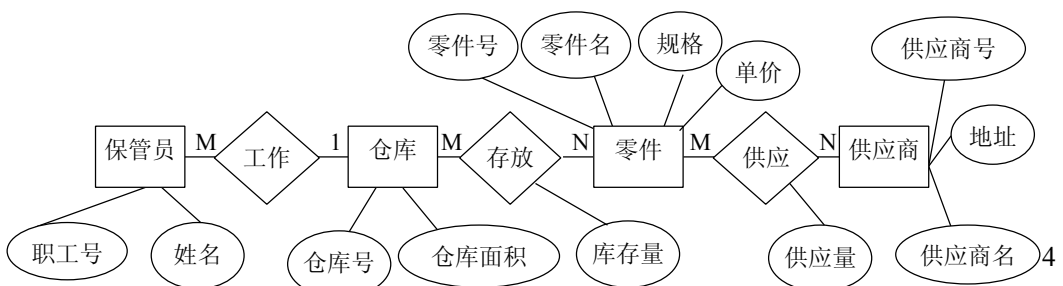
7. 设某工厂数据库中有四个实体集。一是“仓库”实体集，属性有仓库号、仓库面积等；二是“零件”实体集，属性有零件号、零件名、规格、单价等；三是“供应商”实体集，属性有供应商号、供应商名、地址等；四是“保管员”实体集，属性有职工号、姓名等。

设仓库与零件之间有“存放”联系，每个仓库可存放多种零件，每种零件可存放于若干仓库中，每个仓库存放每种零件要记录库存量；供应商与零件之间有“供应”联系，一个供应商可供应多种零件，每种零件也可由多个供应商提供，每个供应商每提供一种零件要记录供应量；仓库与保管员之间有“工作”联系，一个仓库可以有多名保管员，但一名保管员只能在一个仓库工作。

(1)试为该工厂的数据库设计一个 ER 模型，要求标注联系类型，可省略实体属性。

(2)根据转换规则，将 ER 模型转换成关系模型，要求标明每个关系模式的主键和外键。

答：(1) ER 模型



(2) 关系模式:

仓库 (仓库号, 仓库面积)

零件 (零件号, 零件名, 规格, 单价)

供应商 (供应商号, 供应商名, 地址)

保管员 (职工号, 姓名, 仓库号)

库存 (仓库号, 零件号, 库存量)

供应 (供应商号, 零件号, 供应量)

8. 某网上订书系统, 涉及如下信息:

(1) 客户: 客户号、姓名、地址、联系电话。

(2) 图书: 书号、书名、出版社、单价。

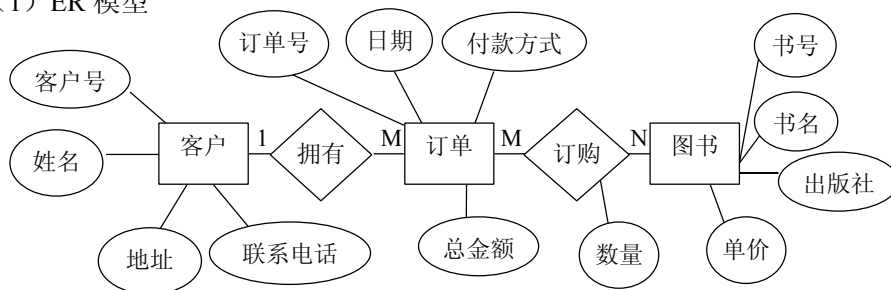
(3) 订单: 订单号、日期、付款方式、总金额。

其中: 一份订单可订购多种图书, 每种图书可订购多本; 一位客户可有多份订单, 一份订单仅对应一位客户。

(1) 根据以上叙述, 建立 ER 模型, 要求标注联系类型 (可省略实体的属性)。

(2) 根据转换规则, 将 ER 模型转换成关系模型, 要求标注每个关系模型的主键和外键 (如果存在)。

答: (1) ER 模型



(2) 关系模式:

客户 (客户号, 姓名, 地址, 联系电话, 订单号)

图书 (书号, 书名, 出版社, 单价)

订单 (订单号, 日期, 付款方式, 总金额)

订购 (订单号, 书号, 数量)

## 第三章 关系数据库

1. 关系数据模型由哪三个要素组成。

答：关系数据模型由关系数据结构、关系操作集合和关系完整性约束三部分组成。

## 2. 简述关系的性质。

答：(1) 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。

(2) 在同一个关系中，不同的列的数据可以是同一种数据类型，但各属性的名称都必须是互不相同。

(3) 同一个关系中，任意两个元组都不能完全相同。

(4) 在一个关系中，列的次序无关紧要。即列的排列顺序是不分先后的。

(5) 在一个关系中，元组的位置无关紧要。即排行不分先后，可以任意交换两行的位置。

(6) 关系中的每个属性必须是单值，即不可再分，这就要求关系的结构不能嵌套。这是关系应满足的最基本的条件。

## 3. 简述关系的完整性。

答：关系模型允许定义三类完整性约束：实体完整性、参照完整性和用户自定义的完整性约束。

实体完整性规则要求关系中元组在组成主码的属性上不能有空值。

参照完整性规则：若属性（或属性组）F 是基本关系 R 的外码，它与基本关系 S 的主码 Ks 相对应（基本关系 R 和 S 可能是相同的系统），则对于 R 中每个元组在 F 上的值必须为：或者取空值（F 的每个属性值均为空值）；或者等于 S 中某个元组的主码值。

用户定义的完整性规则由用户根据实际情况对数据库中数据的内容进行的规定，也称为域完整性规则。

## 4. 传统的集合运算和专门的关系运算都有哪些。

答：(1) 传统的集合操作：并、差、交、笛卡儿积。

(2) 专门的关系操作：投影（对关系进行垂直分割）、选择（水平分割）、连接（关系的结合）、除法（笛卡儿积的逆运算）等。

## 5. 根据给定的关系模式进行查询。

设有学生-课程关系数据库，它由三个关系组成，它们的模式是：学生 S(学号 S#,姓名 SN,所在系 SD,年龄 SA)、课程 C(课程号 C#,课程名 CN,先修课号 PC#)、SC(学号 S#,课程号 C#,成绩 G)。请用关系代数分别写出下列查询：

(1)检索学生的所有情况。

$\Pi_{S\#, SN, SD, SA}(S)$

(2)检索学生年龄大于等于 20 岁的学生姓名。

$\Pi_{SN}(\sigma_{SA \geq 20}(S))$

(3)检索先修课号为 C2 的课程号。

$\Pi_{C\#}(\sigma_{PC\#='C2'}(C))$

(4)检索课程号 C1 的成绩为 A 的所有学生姓名。

$\Pi_{SN}(\sigma_{C\#='C1' \wedge G='A'}(S \bowtie SC))$

(5)检索学号为 S1 的学生选修的所有课程名及先修课号。

$$\prod_{CN, PC\#}(\sigma_{S\#='S1'}(C \bowtie SC))$$

(6)检索年龄为 23 岁的学生所选修的课程名。

$$\prod_{CN}(\sigma_{SA=23}(C \bowtie SC \bowtie S))$$

## 第四章 关系规范化基础

1. 解释下列术语的含义：函数依赖、平凡函数依赖、非平凡函数依赖、部分函数依赖、完全函数依赖、传递函数依赖、范式。

答：函数依赖：指在关系 R 中，X、Y 为 R 的两个属性或属性组，如果对于 R 的所有关系 r 都存在：对于 X 的每一个具体值，Y 都只有一个具体值与之对应，则称属性 Y 函数依赖于属性 X。记作  $X \rightarrow Y$ 。当 Y 不函数依赖于 X 时，记作： $X \nrightarrow Y$ 。当  $X \rightarrow Y$  且  $Y \rightarrow X$  时，则记作： $X \leftrightarrow Y$ 。

平凡函数依赖：设关系模式 R(U)，U 是 R 上的属性集，X、Y  $\subseteq$  U；如果  $X \rightarrow Y$ ，且  $Y \subseteq X$ ，则称  $X \rightarrow Y$  为平凡的函数依赖。

非平凡函数依赖、如果  $X \rightarrow Y$ ，且 Y 不是 X 的子集，则称  $X \rightarrow Y$  为非平凡的函数依赖。

完全函数依赖：设关系模式 R(U)，U 是 R 上的属性集，X、Y  $\subseteq$  U；如果  $X \rightarrow Y$ ，并且对于 X 的任何一个真子集 Z， $Z \rightarrow Y$  都不成立，则称 Y 完全函数依赖于 X；

部分函数依赖：如果  $X \rightarrow Y$ ，但对于 X 的某一个真子集 Z，有  $Z \rightarrow Y$  成立，则称 Y 部分函数依赖于 X。

传递函数依赖：设关系模式 R(U)，X  $\subseteq$  U，Y  $\subseteq$  U，Z  $\subseteq$  U；如果  $X \rightarrow Y$ ， $Y \rightarrow Z$  成立，但  $Y \rightarrow X$  不成立，且 Z-X、Z-Y 和 Y-X 均不空，则称  $X \rightarrow Z$  为传递函数依赖。

范式：范式（Normal Form）是符合某一种级别的关系模式的集合，是衡量关系模式规范化程度的标准，达到的关系才是规范化的。

2. 简述非规范化的关系中存在哪些问题。

答：①数据冗余 ②更新异常 ③插入异常 ④删除异常

3. 简述关系模式规范化的目的。

答：关系模式规范化的目的是解决关系模式中存在的冗余、插入和删除异常以及更新异常等问题。其基本思想是消除数据依赖中的不合适部分，使各关系模式达到某种程度的分离，使一个关系描述一个概念、一个实体或实体间的一种联系。因此，规范化的实质是概念的单一化。

4. 要建立关于系、学生、班级、研究会等信息的一个关系数据库。规定：一个系有若干专业、每个专业每年只招一个班，每个班有若干学生，一个系的学生住在同一个宿舍区。每个学生可参加若干研究会，每个研究会有若干学生。学生参加某研究会，有一个入会年份。

描述学生的属性有：学号、姓名、出生年月、系名、班号、宿舍区。

描述班级的属性有：班号、专业名、系名、人数、入校年份。

描述系的属性有：系号、系名、系办公室地点、人数。

描述研究会的属性有：研究会名、成立年份、地点、人数。

试给出上述数据库的关系模式；写出每个关系的基本的函数依赖集；指出是否存在传递函数依赖，指出各关系的主码和外码。

答：关系模式：

学生（学号，姓名，出生年月，系号，班号，宿舍区）

班级（班号，专业名，系号，人数，入校年份）

系（系号，系名，系办公室地点，人数）

研究会（研究会名，成立年份，地点，人数）

参加研究会（学号，研究会名，入会年份）

学生关系的基本函数依赖集：学号→（姓名，出生年月，系号，班号），系名→宿舍区

班级关系的基本函数依赖集：班号→（专业名，系号，人数，入校年份）

系关系的基本函数依赖集：系名→（系号，系办公室地点，人数）

研究会关系的基本函数依赖集：研究会名→（成立年份，地点，人数）

参加研究会关系的基本函数依赖集：（学号，研究会名）→入会年份

5. 设有关系模式 R(运动员编号，姓名，性别，班级，班主任，项目号，项目名，成绩)

如果规定：每名运动员只能代表一个班级参加比赛，每个班级只能有一个班主任；每名运动员可参加多个项目，每个比赛项目也可由多名运动员参加；每个项目只能有一个项目名；每名运动员参加一个项目只能有一个成绩。根据上述语义，回答下列问题：

(1)写出关系模式 R 的主关键字；

(2)分析 R 最高属于第几范式，说明理由。

(3)若 R 不是 3NF，将其分解为 3NF。

答：（1）关系模式 R 的主关键字是{运动员编号，项目号}。

（2）R 最高属于第一范式。

因为存在着姓名、性别、班级和项目名对主关键字{运动员编号，项目号}的部分函数依赖，没有达到 2NF。

（3）首先分解为 2NF：R1（运动员编号、姓名、性别、班级、班主任），R2（项目号、项目名），R3（运动员编号、项目号、成绩）

因为 R1 存在班主任对运动员编号的传递函数依赖，所以没有达到 3NF，再分解为 3NF：R1 分解为 R1（运动员编号、姓名、性别、班级）和 R4（班级、班主任）

6. 设有关系模式：R（职工号，日期，日营业额，部门名，部门经理）

如果规定：每个职工每天只有一个营业额，每个职工只在一个部门工作，每个部门只有一个经理。

(1)根据上述规定，写出模式 R 主关键字。

(2)分析 R 最高属于第几范式，说明理由。

(3)若 R 不是 3NF，将其分解为 3NF。

答：（1）关系模式 R 的主关键字是（职工号，日期）。



(2) R 最高属于第一范式。

因为存在着部门名对主关键字（职工号，日期）的部分函数依赖，没有达到 2NF。

(3) 首先分解为 2NF: R1（职工号，部门名，部门经理），R2（职工号，日期，营业额）

因为 R1 存在部门经理对职工号的传递函数依赖，所以没有达到 3NF，再分解为 3NF: R1 分解为 R1（职工号，部门名）和 R3（部门名，部门经理）

## 第五章 数据库设计

1. 简述数据库设计过程。

答：数据库设计可分为以下六个阶段：需求分析阶段、概念结构设计阶段、逻辑结构设计阶段、数据库物理设计阶段、数据库实施阶段以及数据库运行和维护阶段。

2. 简述数据库设计过程的各个阶段上的设计任务。

答：需求分析的任务是通过详细调查现实世界要处理的对象（组织、部门、企业等），充分了解原系统(手工系统或计算机系统)工作概况，明确用户的各种需求，然后在此基础上确定新系统的功能。

概念设计阶段要做的工作不是直接将需求分析得到的数据存储格式转换为 DBMS 能处理的数据库模式，而是将需求分析得到的用户需求抽象为反映用户观点的概念模型。

逻辑设计的主要目标是产生一个 DBMS 可处理的数据模型和数据库模式。该模型必须满足数据库的存取、一致性及运行等各方面的用户需求。

将逻辑设计中产生的数据库逻辑模型结合指定的 DBMS，设计出最适合应用环境的物理结构的过程，称为数据库的物理结构设计。

根据逻辑和物理设计的结果，在计算机上建立起实际的数据库结构，并装入数据，进行试运行和评价的过程，叫做数据库的实施（或实现）。

维护工作包括以下内容：数据库的转储和恢复；数据库的安全性和完整性控制；数据库性能的监督、分析和改造；数据库的重组织和重构造。

3. 简述数据库设计的概念。

答：数据库设计是对于给定的应用环境，在关系数据库理论的指导下，构造最优的数据库模式，在数据库管理系统上建立数据库及其应用系统，使之能有效地存储数据，满足用户的各种需求的过程。

4. 简述数据库概念结构设计的方法和设计步骤。

答：方法：(1) 自顶向下法 (2) 自底向上法 (3) 逐步扩张法 (4) 混合策略

在概念结构设计时，可以分为两步。进行数据抽象，设计局部 E-R 模型，即设计用户视图；集成各局部 E-R 模型，形成全局 E-R 模型，即视图的集成。

5. 什么是数据库的逻辑结构设计？试述其设计步骤。

答：逻辑设计的主要目标是产生一个 DBMS 可处理的数据模型和数据库模式。该模型必须满足数据库的存取、一致性及运行等各方面的用户需求。

其逻辑结构设计阶段一般要分为三步进行：将 E-R 图转化为关系数据模型，关系模式的优化，设计用户外模式。

6. 简述把 E-R 图转换为关系模型的转换规则。

答：一个实体转化为一个关系模式。实体的属性就是该关系模式的属性。实体的主码就是该关系模式的主码。

两实体间 1:1 联系可以转换为一个独立的关系模式，也可以与任意一端对应的关系模式合并。

两实体间 1:N 联系可以转换为一个独立的关系模式，也可以与 N 端对应的关系模式合并。

同一实体间的 1:N 联系可在这个实体所对应的关系模式中多设一个属性，用来作为与该实体相联系的另一个实体的主码。

两实体间 M:N 联系，必须为联系产生一个新的关系模式。该关系模式中至少包含被它所联系的双方实体的主码，若联系中有属性，也要并入该关系模式中。

同一实体间 M:N 联系，必须为联系产生一个新的关系模式。该关系模式中至少包含被它所联系的双方实体的主码，若联系有属性，也要并入该关系模式中。由于这个“联系”只涉及一个实体，所以加入的实体的主码不能相同。

两个以上实体集之间 M:N 的联系，必须为联系产生一个新的关系模式，该关系模式中至少包含被它所联系的所有实体的主码，若联系有属性，也要并入该关系模式中。

7. 将图 1 学生信息数据库系统的 E-R 图转换为关系模型。

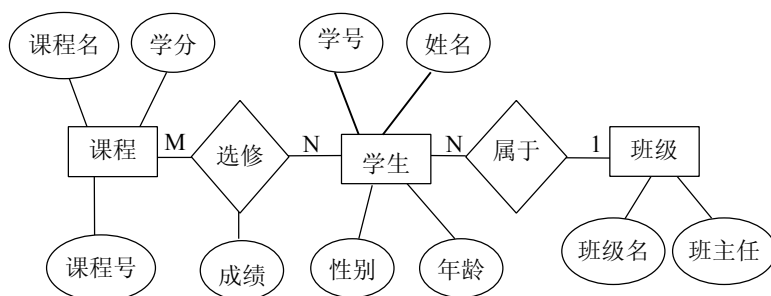


图 1 习题 7 图

答：课程（课程号，课程名，学分）

学生（学号，姓名，性别，年龄，班级名）

班级（班级名，班主任）

选修（学号，课程号，成绩）

8. 一个图书馆理系统中有如下信息：

图书：书号、书名、数量、位置

借书人：借书证号、姓名、单位

出版社：出版社名、邮编、地址、电话、E-mail

其中约定：任何人都可以借多种书，任何一种书可以被多个人借，借书和还书时，要登记相应的借书日期和还书日期；一个出版社可以出版多种书籍，同一本书仅为一个出版社所出版，出版社名具有唯一性。

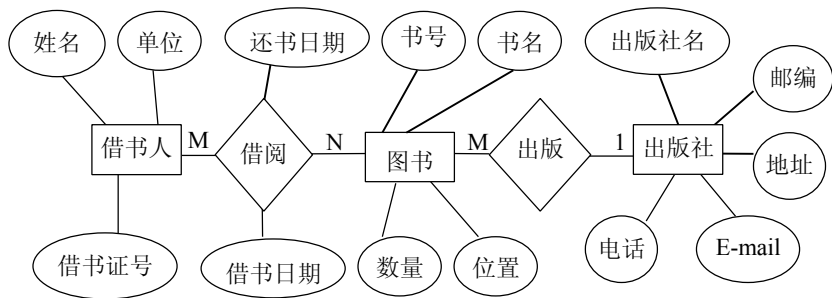
根据以上情况，完成如下设计：

(1) 设计系统的 E-R 图；

将 E-R 图转换为关系模式；

(3) 指出转换后的每个关系模式的主码。

答：(1) E-R 图



(2) 和 (3) 关系模式

图书 (书号, 书名, 数量, 位置, 出版社名)

借书人 (借书证号, 姓名, 单位)

出版社 (出版社名, 邮编, 地址, 电话, E-mail)

借阅 (书号, 借书证号, 借书日期, 还书日期)

## 第六章 SQL Server 2005 基础

1. 安装 Microsoft SQL Server 2005 系统操作。

答：略。

2. SQL Server 2005 主要提供了哪些服务？如何启动、暂停或停止 SQL Server 服务？

答：服务器管理、对象资源管理、SQL Server 配置管理等。

启动、暂停或停止 SQL Server 服务：首先在 SQL Server 配置管理器的窗口左边单击“SQL Server 2005 服务”，此时在窗口右边会看到已安装的所有服务，可以选中某个服务，然后单击窗口上部工具栏中的相应按钮，或右击某个服务名称，在弹出的快捷菜单中选择相应的菜单选项来启动或停止服务。

3. SQL Server 外围应用配置器的主要作用是什么？

答：SQL Server 外围应用配置器 (SQL Server Surface Area Configuration) 是 SQL Server 2005 提供的一个新的管理工具。通过 SQL Server 外围应用配置器，可以在统一集中的界面下设置各种 SQL Server 服务实例对外沟通的渠道，降低可能的危险。

4. 简述 SQL Server Management Studio 的使用。

答：略。

5. 简述对象资源管理器的功能。

答：SQL Server Management Studio 的对象资源管理器组件是一种集成工具，可以查看和管理所有服务器类型的对象。

用户可以通过该组件操作数据库，包括新建、修改、删除数据库、表、视图等数据库对象，新建查询、设置关系图、设置系统安全、数据库复制、数据备份、恢复等操作，是 SQL Server Management Studio 中最常用、也是最重要的一个组件，类似于 SQL Server 2000 中的企业管理器。

## 第七章 Transact-SQL 语言基础

1. 简述 T-SQL 语言的特点。

答：（1）综合统一

（2）两种使用方式，统一的语法结构

（3）高度非过程化

（4）类似于人的思维习惯，容易理解和掌握

2. 什么是标识符？T-SQL 中有几种标识符？它们的区别是什么？

答：标识符是用来标识事物的符号，其作用类似于给事物起的名称。标识符分为两类：常规标识符和分隔标识符。

符合所有标识符格式规则的标识符为常规标识符，可以使用分隔符，也可以不使用分隔符。不符合标识符格式规则的标识符必须使用分隔符。

3. 如何表示数据库对象名？

答：[server\_name].[database\_name].[schema\_name] | database\_name. [schema\_name] | schema\_name.object\_name

4. 什么是局部变量？什么是全局变量？如何表示它们？

答：全局变量由系统提供且预先声明，通过在名称前加两个“@”符号区别于局部变量。用户只能使用全局变量，不能对它们进行修改。全局变量的作用范围是整个 SQL Server 系统，任何程序都可以随时调用它们。

T-SQL 语言中的变量是可以保存单个特定类型的数据值的对象，也称为局部变量，只在定义它们的批处理或过程中可见。

5. 以下变量名中，哪些是合法的变量名，哪些是不合法的变量名？

A1, 1a, @x, @@y, &变量 1, @姓名, 姓名, #m, ##n, @@@abc##, @my\_name

答：合法的变量名：@x, @@y, @姓名, @my\_name

不合法的变量名：A1, 1a, &变量 1, 姓名, #m, ##n, @@@abc##

6. SQL Server 2005 所使用的运算符类别有哪些？

答：SQL Server 2005 系统中，可以使用的运算符可以分为算术运算符、逻辑运算符、赋值运算符、字符串串联运算符、按位运算符、一元运算符及比较运算符等。

7. 利用 T-SQL 语句计算下列表达式，并给出运算结果。

(1)  $9-3*5/2+6\%4$  (2)  $5\&2|4$  (3) '你们'+ '好' (4)  $\sim 10$

答：（1）4 （2）4 （3）你们好 （4）-11

8. 给出以下 T-SQL 语句的运行结果。

DECLARE @d SMALLDATETIME

```
SET @d='2007-1-26'
```

```
SELECT @d+10,@d-10
```

答：2007-02-05，2007-01-16

9. 什么是批处理？使用批处理有何限制？批处理的结束符是什么？

答：批处理是包含一个或多个 T-SQL 语句的集合，从应用程序一次性地发送到 SQL Server 2005 进行执行，因此可以节省系统开销。SQL Server 将批处理的语句编译为一个可执行单元，称为执行计划，批处理的结束符为“GO”。

10. 注释有几类，它们分别是什么？

答：在 T-SQL 中可使用两类注释符：

(1) ANSI 标准的注释符“--”用于单行注释；

(2) 与 C 语言相同的程序注释符号，即“/\*.....\*/”，“/\*”用于程序注释开头，“\*/”用  
语程序注释结尾，可以在程序中多行文字标示为注释。

## 第八章 数据库和表

1. 简述数据库的两种存储结构。

答：数据库的物理存储结构指的是保存数据库各种逻辑对象的物理文件是如何在磁盘上存储的，数据库在磁盘上是以文件为单位存储的，SQL Server 2005 将数据库映射为一组操作系统文件。

逻辑存储结构：数据库是存储数据的容器，即数据库是一个存放数据的表和支持这些数据的存储、检索、安全性和完整性的逻辑成分所组成的集合。

2. 数据库由哪几种类型的文件组成？其扩展名分别是什么？

答：(1) 主数据文件：主数据文件是数据库的起点，指向数据库中的其他文件。主数据文件的推荐文件扩展名是 .mdf。

(2) 辅助数据文件：除主数据文件以外的所有其他数据文件都是辅助数据文件。辅助数据文件的推荐文件扩展名是 .ndf。

(3) 事务日志文件：日志文件包含着用于恢复数据库的所有日志信息。日志文件的推荐文件扩展名是 .ldf。

3. 简述 SQL Server 2005 中文件组的作用和分类。

答：为了便于管理和分配数据而将文件组织在一起，通常可以为一个磁盘驱动器创建一个文件组（File Group），将多个数据库文件集合起来形成一个整体。

SQL Server 中的数据库文件组分为主文件组（Primary File Group）和用户定义文件组（user\_defined Group）。

4. 使用 SQL Server Management Studio 创建名为“仓库库存”的数据库，并设置数据库主文件名为仓库库存\_data，大小为 10MB，日志文件名为仓库库存\_log，大小为 2MB。

答：略。

5. 删除习题 4 创建的数据库，使用 T-SQL 语句再次创建该数据库，主文件和日志文件的文件名同上，要求：仓库库存\_data 最大尺寸为无限大，增长速度为 20%，日志文件

初始大小为 2MB，最大尺寸为 5MB，增长速度为 1MB。

答：CREATE DATABASE 仓库库存

```
(NAME = '仓库库存_data',  
FILENAME = 'F:\仓库库存_data.MDF',  
SIZE = 10MB,  
FILEGROWTH = 20%)  
LOG ON  
(NAME = '仓库库存_log',  
FILENAME = 'F:\仓库库存_log.LDF',  
SIZE = 2MB,  
MAXSIZE = 5MB,  
FILEGROWTH = 1MB)
```

6. 请分别使用 SQL Server Management Studio 和 T-SQL 语句创建数据库 Student，要创建的数据库的要求如下所示：数据库名称为 Student，包含 3 个 20MB 的数据库文件，2 个 10MB 的日志文件，创建使用一个自定义文件组，主文件为第一个文件，主文件的后缀名为.mdf，次要文件的后缀名为.ndf；要明确地定义日志文件，日志文件的后缀名为.ldf；自定义文件组包含后两个数据文件，所有的文件都放在目录“C:\DATA”中。

答：略。

7. 在数据库“仓库库存”中完成下列操作。

(1)创建“商品”表，表结构如表 1：

表 1 商品表

列名	数据类型	长度	是否允许为空值	说明
商品编号	Char	6	NOT NULL	主键
商品名称	Varchar	20	NOT NULL,	
单价	Float			
生产商	Varchar	30		

(2)创建“仓库”表，表结构如表 2：

表 2 仓库表

列名	数据类型	长度	是否允许为空值	说明
仓库编号	Char	3	NOT NUL	主键
仓库地址	Varchar	20	NOT NULL	

(3)创建“库存情况”表，表结构如表 3：

表 3 库存情况表

列名	数据类型	长度	是否允许为空值	说明
----	------	----	---------	----

仓库编号	Char	3	NOT NULL	主键
商品编号	Char	6	NOT NUL	主键
数量	int			

答：(1) USE 仓库库存

GO

CREATE TABLE 商品

(商品编号 char(6) NOT NULL PRIMARY KEY,

商品名称 char(20) NOT NULL,

单价 Float,

生产商 Varchar (30)

)

(2), (3) 略。

8. 建立“商品”表、“仓库”表和“库存情况”表三表之间的关系图。

答：略。

9. 分别给“商品”表、“仓库”表和“库存情况”表添加数据。

答：略。

## 第九章 数据库查询

针对“教学库”中的三个表，完成 1~5 题。

1. 试用 SQL 语句在学生表中加入“年龄”字段，要求：年龄为 tinyint 类型，可以为空。然后为所有学生输入年龄数据。

2. 试用 SQL 的查询语句实现下列查询：

(1) 统计有学生选修的课程门数。

答：SELECT COUNT(DISTINCT 课程号) FROM 选课

(2) 求选修 C004 课程的学生的平均年龄。

答：SELECT AVG(年龄) FROM 学生,选课

WHERE 学生.学生号=选课.学生号 and 课程号='C004'

(3) 求学分为 3 的每门课程的学生平均成绩。

答：SELECT 课程.课程号,AVG(成绩) FROM 课程,选课

WHERE 课程.课程号=选课.课程号and 学分=3

GROUP BY 课程.课程号

(4) 统计每门课程的学生选修人数，超过 3 人的课程才统计。要求输出课程号和选修人数，查询结果按人数降序排列，若人数相同，按课程号升序排列。

答：SELECT 课程号,COUNT(\*) FROM 选课

GROUP BY 课程号

HAVING COUNT(\*)>3

ORDER BY COUNT(\*) DESC, 课程号

(5) 检索学号比王非同学大, 而年龄比他小的学生姓名。

答: SELECT 姓名 FROM 学生

```
WHERE 学生号>(SELECT 学生号 FROM 学生
                WHERE 姓名='王非')
and 年龄<(SELECT 年龄 FROM 学生
           WHERE 姓名='王非')
```

(6) 检索姓名以王打头的所有学生的姓名和年龄。

答: SELECT 姓名,年龄 FROM 学生

```
WHERE 姓名 LIKE '王%'
```

(7) 在选课表中检索成绩为空值的学生学号和课程号。

答: SELECT 学生号,课程号 FROM 选课

```
WHERE 成绩 IS NULL
```

(8) 求年龄大于女同学平均年龄的男学生姓名和年龄。

答: SELECT 姓名,年龄 FROM 学生

```
WHERE 性别='男'
and 年龄 >(SELECT AVG(年龄) FROM 学生
            WHERE 性别='女')
```

(9) 求年龄大于所有女同学年龄的男学生姓名和年龄。

答: SELECT 姓名,年龄 FROM 学生

```
WHERE 性别='男'
and 年龄 > all (SELECT 年龄 FROM 学生
                 WHERE 性别='女')
```

(10) 检索所有比王华年龄大的学生姓名、年龄和性别。

答: SELECT 姓名,年龄,性别 FROM 学生

```
WHERE 年龄 > (SELECT 年龄 FROM 学生
              WHERE 姓名='王华')
```

(11) 检索选修课程 C002 的学生中成绩最高的学生的学号。

答: SELECT 学生号 FROM 选课

```
WHERE 课程号='C002' and 成绩=(SELECT MAX(成绩) FROM 选课
                                WHERE 课程号='C002')
```

(12) 检索学生姓名及其所选修课程的课程号和成绩。

答: SELECT 姓名, 课程号, 成绩 FROM 学生,选课

```
WHERE 学生.学生号=选课.学生号
```

(13) 检索选修 4 门以上课程的学生总成绩(不统计不及格的课程), 并要求按总成绩的降序排列出来。

答: SELECT 学生号,SUM(成绩) FROM 选课

```
WHERE 成绩>=60
GROUP BY 学生号
```



```
HAVING COUNT(*)>=4
ORDER BY SUM(成绩) DESC
```

3. 利用控制流语句，查询学号为 0101001 的学生的各科成绩，如果没有这个学生的成绩，就显示“此学生无成绩”。

答：IF EXISTS ( SELECT \* FROM 选课 WHERE 学号='0101001')

```
    SELECT 课程号,成绩 FROM 选课
    WHERE 学号='0101001'
ELSE
    PRINT '此学生无成绩'
```

4. 用函数实现：求某个专业选修了某门课的学生人数。

答：CREATE FUNCTION renshu(@p char(10),@cn char(4)) RETURNS float

```
AS
BEGIN
    DECLARE @cout float
    SELECT @cou=( SELECT count(*) FROM 学生,选课
                  WHERE 学生.学生号=选课.学生号 and 课程号=@cn
                  and 专业=@p)
    RETURN @cou
END
```

5. 用函数实现：查询某个专业所有学生所选的每门课的平均成绩。

答：CREATE FUNCTION average (@p char(10)) RETURNS float

```
AS
BEGIN
    DECLARE @aver float
    SELECT @aver=( SELECT 课程号,avg(成绩) FROM 学生,选课
                  WHERE 学生.学生号=选课.学生号 and 专业=@p
                  GROUP BY 课程号)
    RETURN @aver
END
```

6. 针对“仓库库存”中的“商品”表，查询商品的价格等级，商品号、商品名和价格等级（单价 1000 元以内为“低价商品”，1000~3000 元为“中等价位商品”，3000 元以上为“高价商品”）。

答：SELECT 商品号, 商品名,

```
CASE
    WHEN 单价<1000 then '低价商品'
    WHEN 单价<3000 then '中等价位商品'
    WHEN 单价>=3000 then '高价商品'
END AS 价格等级
```

## 第十章 视图和索引

1. 引入视图的主要目的是什么？

答：数据库的基本表是按照数据库设计人员的观点设计的，并不一定符合用户的需求。SQL Server 2005 可以根据用户需求重新定义表的数据结构，这种数据结构就是视图。视图是关系数据库系统提供给用户以多种角度观察数据库中数据的重要机制，其结构和数据是建立在对表的查询基础上的。

2. 当删除视图时所对应的数据表会删除吗？

答：不会。

3. 简述视图的优点。

答：使用视图有很多优点，主要表现在：

(1) 为用户集中数据，简化用户的数据查询和处理。

(2) 保证数据的逻辑独立性。

(3) 重新定制数据，使得数据便于共享；合并分割数据，有利于数据输出到应用程序中。

(4) 数据保密。

4. 可更新视图必须满足哪些条件？

答：(1)任何修改（包括 UPDATE、INSERT 和 DELETE 语句）都只能引用一个基本表的列。

(2)视图中被修改的列必须直接引用表列中的基础数据。不能通过任何其他方式对这些列进行派生，如通过聚合函数、计算（如表达式计算）、集合运算等。

(3)被修改的列不受 GROUP BY、HAVING、DISTINCT 或 TOP 子句的影响。

5. 创建索引的必要性和作用是什么？

答：数据库的索引就类似于书籍的目录，如果想快速查找而不是逐页查找指定的内容，可以通过目录中章节的页号找到其对应的内容。类似地，索引通过记录表中的关键值指向表中的记录，这样数据库引擎就不用扫描整个表而定位到相关的记录。相反，如果没有索引，则会导致 SQL Server 搜索表中的所有记录，以获取匹配结果。

索引的优点包括：

(1) 大大加快数据的检索速度，这是创建索引的最主要的原因。

(2) 创建唯一性索引，保证表中每一行数据的唯一性。

(3) 加速表和表之间的连接。

(4) 在使用分组和排序子句进行数据检索时，同样可以显著减少查询中分组和排序的时间。

(5) 查询优化器可以提高系统的性能，但它是依靠索引起作用的。

6. 聚集索引和非聚集索引有何异同？

答：聚集索引会对表和视图进行物理排序，所以这种索引对查询非常有效，在表和视图

中只能有一个聚集索引。非聚集索引不会对表和视图进行物理排序。如果表中不存在聚集索引，则表是未排序的。

7. 在 SQL Server Management Studio 中创建一个仓库库存信息视图，要求包含仓库库存数据库中三个表的所有列。

答：略。

8. 利用 T-SQL 语句创建一个查询每个学生的平均成绩的视图，要求包含学生的学生号和姓名。

答：CREATE VIEW 学生\_平均成绩

AS

SELECT 学生.学生号,姓名,avg(成绩) AS 平均成绩

FROM 学生,选课

WHERE 学生.学生号=选课.学生号

GROUP BY 学生.学生号,姓名

9. 在 SQL Server Management Studio 中按照选课表的成绩列升序创建一个普通索引（非唯一、非聚集）。

答：略。

10. 利用 T-SQL 语句按照商品表的单价列降序创建一个普通索引。

答：CREATE INDEX index\_商品单价 ON 商品(单价 DESC)

## 第十一章 存储过程、触发器和游标

1. 简述存储过程和触发器的优点。

答：存储过程最主要的特色是当写完一个存储过程后即被翻译成可执行码存储在系统表内，当作是数据库的对象之一，一般用户只要执行存储过程，并且提供存储过程所需的参数就可以得到所要的结果而不必再去编辑 T-SQL 命令。

由于在触发器中可以包含复杂的处理逻辑，因此，应该将触发器用来保持低级的数据的完整性，而不是返回大量的查询结果。

使用触发器主要可以实现以下操作：

- (1) 强制比 CHECK 约束更复杂的数据的完整性
- (2) 使用自定义的错误提示信息
- (3) 实现数据库中多张表的级联修改
- (4) 比较数据库修改前后数据的状态
- (5) 调用更多的存储过程
- (6) 维护非规范化数据

2. 简述游标的概念及类型。

答：游标是处理数据的一种方法，它允许应用程序对查询语句 SELECT 返回的结果集中每一行进行相同或不同的操作，而不是一次对整个结果集进行同一种操作。为了查看或者处理结果集中的数据，游标提供了在结果集中一次以行或者多行前进或向后浏览数据的能力，我们可以把游标当作一个指针，它可以指定结果中的任何位置，然后允许用

户对指定位置的数据进行处理。

SQL Server 支持三种类型的游标：T-SQL 游标，API 服务器游标和客户游标。

由于 API 游标和 T-SQL 游标使用在服务器端，所以被称为服务器游标，也被称为后台游标，而客户端游标被称为前台游标。服务器游标包含以下四种：静态游标、动态游标、只进游标、键集驱动游标。

3. 简述 SQL Server2005 中存储过程和触发器的分类。

答：存储过程分为系统存储过程、本地存储过程、临时存储过程、远程存储过程和扩展存储过程。

触发器：DML 触发器是当数据库服务器中发生数据操作语言(DML)事件时会自动执行的存储过程。DDL 触发器是在响应数据定义语言（DDL）语句时触发，一般用于数据库中执行管理任务。

4. 创建存储过程，从课程表中返回指定的课程的信息。该存储过程对传递的参数进行模式匹配，如果没有提供参数，则返回所有课程的信息。

答：CREATE PROCEDURE Pcourse

    @name varchar(20) = '%'

AS

SELECT \* FROM 课程

WHERE 课程名 LIKE @name

5. 创建存储过程，计算指定学生（姓名）的总成绩，存储过程中使用一个输入参数（姓名）和一个输出参数(总成绩)。

答：CREATE PROCEDURE Sname (@S\_n varchar(20), @sum1 int OUTPUT

AS

SELECT @sum1= sum(成绩) FROM 选课,学生

WHERE 姓名=@S\_n and 学生.学生号=选课.学生号

6. 为 dept 表创建一个实现级联删除的触发器，当执行删除时，激活该触发器同时删除 gongcheng 表中相应记录(leader 列)。

答：CREATE TRIGGER d\_tr ON dept

FOR delete

AS

delete from gongcheng where leader=(SELECT leader FROM deleted)

7. 在教学库中建一个学生党费表，属性（学生号，姓名，党费），学生号是主键，也是外键（参考学生表的学生号）；创建一个触发器，保证只能在每年的 6 月和 12 月交党费，如果在其它时间录入则显示提示信息。

答：CREATE TABLE 学生党费表

(学生号 CHAR (7) primary key foreign key references 学生(学生号),

姓名 char(6),

党费 int)

```

CREATE TRIGGER trg_学生党费表
on 学生党费表 for insert
AS
    if not(datepart(mm,getdate())='06' or datepart(mm,getdate())='12')
BEGIN
    print'对不起,只能在每年的 6 月和 12 月交党费'
    rollback
END

```

8. 利用 T-SQL 扩展方式声明一个游标，查询学生表中所有男生的信息，并读取数据。  
 要求：(1)读取最后一条记录。(2)读取第一条记录。(3)读取第 5 条记录。(4)读取当前记录指针位置后第 3 条记录。

答：略。

## 第十二章 事务与并发控制

1. 什么是事务？如果要提交或取消一个事务，使用什么语句？

答：事务处理是数据库的主要工作，事务由一系列的数据操作组成，是数据库应用程序的基本逻辑单元，用来保证数据的一致性。

提交或取消一个事务：COMMIT TRANSACTION 或 ROLLBACK TRANSACTION 语句。

2. 事务分为哪几类？

答：根据系统的设置，SQL Server 2005 将事务分为两种类型：系统提供的事务和用户定义的事务。

根据运行模式的不同，SQL Server 2005 将事务分为 4 种类型：显示事务、隐式事务、自动提交事务和批处理级事务。

3. 简述事务回滚机制。

答：(1)如果不指定回滚的事务名称或保存点，则 ROLLBACK TRANSACTION 命令会将事务回滚到事务的起点。

(2)在嵌套事务时，该语句将所有内层事务回滚到最远的 BEGIN TRANSACTION 语句，transaction\_name 也只能是来自最远的 BEGIN TRANSACTION 语句的名称。

(3)在执行 COMMIT TRANSACTION 语句后不能回滚事务。

(4)如果在触发器中发出 ROLLBACK TRANSACTION 命令，将回滚对当前事务中所做的所有数据修改，包括触发器所做的修改。

(5)事务在执行过程中出现任何错误，SQL Server 都将自动回滚事务。

4. 简述锁机制，锁分为哪几类。

答：锁是防止其他事务访问指定的资源、实现并发控制的一种手段，是多个用户能够同时操纵同一个数据库中的数据而不发生数据不一致现象的重要保障。

锁分为(1) 共享锁 (Shared Lock) (2) 排它锁 (Exclusive Lock) (3) 更新锁 (Update

Lock)(4) 意向锁(Intent Lock)(5) 模式锁(Schema Lock)(6) 大容量更新锁(Bulk Update Lock)

5. 分析各类锁之间的兼容性。

在一个事务已经对某个对象锁定的情况下，另一个事务请求对同一个对象的锁定，此时就会出现锁定兼容性问题。当两种锁定方式兼容时，可以同意对该对象的第二个锁定请求。如果请求的锁定方式与已挂起的锁定方式不兼容，那么就不能同意第二个锁定请求。相反，请求要等到第一个事务释放其锁定，并且释放所有其他现有的不兼容锁定为止。详细内容见书表 12-1。

6. 简述死锁及其解决办法。

答：产生死锁的情况一般包括以下两种：

第一种情况，当两个事务分别锁定了两个单独的对象，这时每一个事务都要求在另外一个事务锁定的对象上获得一个锁，因此每一个事务都必须等待另外一个事务释放占有的锁，这时，就发生了死锁。这种死锁是最典型的死锁形式。

第二种情况，当在一个数据库中时，有若干个长时间运行的事务执行并行的操作，当查询分析器处理一种非常复杂的查询例如连接查询时，那么由于不能控制处理的顺序，有可能发生死锁现象。

在数据库中解决死锁常用的方法有：

(1)要求每个事务一次就将要使用的数据全部加锁，否则就不能继续执行。

(2)允许死锁发生，系统来用某些方式诊断当前系统中是否有死锁发生。

7. 创建一个事务，将所有女生的考试成绩都加 5 分，并提交。

答：

```
BEGIN TRANSACTION
USE 教学库
UPDATE 选课
SET 成绩=成绩+5
WHERE 学生号 in (SELECT 学生号 FROM 学生 WHERE 性别='女')
COMMIT TRANSACTION
```

8. 创建一个事务，向商品表中添加一条记录，设置保存点；再将商品编号为“ds-001”的单价改为“2000”。

答：

```
BEGIN TRANSACTION
USE 仓库库存
INSERT INTO 商品(商品编号,商品名称,单价,生产商)
VALUES('bx-159','冰箱',2500,'安徽美菱')
SAVE TRAN savepoint
UPDATE 商品 SET 单价=2000 WHERE 商品编号='ds-001'
COMMIT
```

## 第十三章 数据库的备份与还原

1. 简述数据库备份和还原的基本概念？

答：数据库备份就是在某种介质上(磁盘、磁带等)创建完整数据库(或者其中一部分)的副本，并将所有的数据项都复制到备份集，以便在数据库遭到破坏时能够恢复数据库。

还原数据库是一个装载数据库的备份，然后应用事务日志重建的过程。

2. 数据库备份有哪几种类型？

答：在 SQL Server 系统中，有 4 种备份类型，分别如下所示：

- ①完整数据库备份。
- ②差异数据库备份。
- ③事务日志备份。
- ④数据库文件或文件组备份。

3. 简述数据库的恢复模式。

答：SQL Server 2005 系统提供了 3 种数据库的恢复模式：

- ①完整恢复模式。
- ②简单恢复模式。
- ③大容量日志恢复模式。

4. 在 SQL Server Management Studio 中使用界面进行备份和还原数据库的操作过程。

答：略。

5. 使用 T-SQL 语句分别实现数据库的备份和还原操作。

答：略。

6. 了解数据库的分离和附加的作用及操作方法。

答：SQL Server 2005 允许分离数据库的数据和事务日志文件，然后将其重新附加到同一台或另一台服务器上。分离数据库将从 SQL Server 删除数据库，但是保证在组成该数据库的数据和事务日志文件中的数据库完好无损。然后这些数据和事务日志文件可以用来将数据库附加到任何 SQL Server 实例上，这使数据库的使用状态与它分离时的状态完全相同。

## 第十三章 数据库的安全管理

### 1. 简答题

(1)简述 SQL Server 的安全性机制。

答：SQL Server 2005 的安全性管理机制可分为三个等级：操作系统级、SQL Server 级和数据库级。在用户使用客户计算机通过网络实现 SQL Server 服务器的访问时，用户首先要获得计算机操作系统的使用权。SQL Server 的服务器级安全性建立在控制服务器登录账号和口令的基础上。SQL Server 采用了标准 SQL Server 登录和集成 Windows NT 登录两种方式。在用户通过 SQL Server 服务器的安全性检验以后，将直接面对不同的数据库入口这是用户将接受的第三次安全性检验。

(2)简述服务器登录账户和数据库用户账号的创建方法。

答：略。

(3)SQL Server 两种身份验证的优点分别是什么？

答：Windows 验证模式有以下主要优点：

数据库管理员的工作可以集中在管理数据库上，而不是管理用户账户。对用户账户的管理可以交给 Windows 去完成。

Windows 有着更强的用户账户管理工具。可以设置账户锁定、密码期限等。

Windows 的组策略支持多个用户同时被授权访问 SQL Server。

SQL Server 是从 RPC 协议连接中自动登录 Windows 用户账户信息的。

混合验证模式具有如下优点：

创建了 Windows 之上的另外一个安全层次。

支持更大范围的用户，例如非 Windows 客户、Novell 网络等。

一个应用程序可利用单个的 SQL Server 登录或口令。

(4)简述角色的概念及其分类。

答：角色是一种 SQL Server 安全账户，是 SQL Server 内部的管理单元，是管理权限时可以视为单个单元的其他安全账户的集合。

角色被分成 3 类：固定服务器角色、固定数据库角色和用户自定义的数据库角色。

(5)简述什么是固定服务器角色？什么是固定数据库角色。

答：固定服务器角色是在服务器级别定义的，所以存在于数据库外面，是属于数据库服务器的。在 SQL Server 安装时就创建了服务器级别上应用的大量预定义的角色，每个角色对应着相应的管理权限。在 SQL Server 2005 安装时，数据库级别上也有一些预定义的角色，在创建每个数据库时都会添加这些角色到新创建的数据库中，每个角色对应着相应的权限。

(6)试述用户自定义数据库角色及应用程序角色的建立方法。

答：略。

(7)简述进行权限设置时，授予、拒绝或撤销三者的关系。

答：授予：用户被授予特定数据库对象的某种操作权限，该用户就可以对数据在此权限范围内进行操作；撤销：用户不再具有特定数据库对象的某种操作权限，但用户可以通过其组或角色成员身份继承此种权限；拒绝：禁止用户对特定数据库对象的某种操作权限，防止用户通过其组或角色成员身份继承此种权限。

## 2. 操作题

(1)创建一个登录，并对其进行修改和删除。

(2)对 SQL Server 进行下列操作：更改 SQL Server 服务登录。

(3)利用系统过程创建 3 个新的 SQL Server 登录：al、a2、a3，相对应的密码是 al、a2、a3，默认的数据库为仓库库存数据库。

(4)根据(10)中的 3 个登录，创建 3 个数据库用户 al、a2、a3。练习使用 GRANT、REVOKE 和 DENY 语句设置数据库用户的权限。

(5)用这 3 个登录名登录 SQL Server，进行相应权限范围内的数据库操作。



答：略。

## 第十五章 基于 C#.NET 的数据库系统开发

1. 简述C#语言的特点。

答：(1)语法简洁 (2)完全面向对象 (3)与Web紧密结合 (4)目标软件的安全性 (5)版本管理技术 (6)灵活性与兼容性

2. 简述ADO.NET对象模型的五个主要的组件。

答：ADO.NET 对象模型有五个主要的组件，分别是 Connection 对象、Command 对象、DataReader 对象、DataSet 对象以及 DataAdapter 对象。

Connection对象表示与数据源之间的连接，用它来建立或断开与数据库的连接。

Command对象主要可以用来对数据库发出一些指令，例如可以对数据库下达查询、插入、修改、删除等数据指令，以及呼叫存在数据库中的预存程序等。

DataReader用于以最快的速度检索并检查查询所返回的行。

DataSet对象可视为暂存区。可以把数据库中查到的信息保存起来，甚至可以显示整个数据库。

DataAdapter对象充当数据库和ADO.NET对象模型中非连接对象之间的桥梁。

3. 简述ADO.NET访问数据库的两种方法。

答：利用 Connection、Command 和 DataReader 对象访问数据库，只能从数据库读取数据，不能添加、修改和删除记录。如果只想进行查询，这种方式效率更高一些。

利用 Connection、Command、DataAdapter 和 DataSet 对象，这种方式比较灵活，不仅可以对数据库进行查询操作，还可以进行增加、删除和修改等操作。

4. 使用C#语言，通过ASP.NET技术，创建一个“仓库库存管理系统”的数据库应用系统。包括进货录入、产品库存查询、产品库存维护、出货等功能。数据库的设计过程和程序开发全部由读者自行完成。

答：略。