

第2章 关系数据库模型

【问题描述】关系数据库模型实际上就是关系运算理论，它是施加于关系上的一组高级运算，是关系数据库查询语言的理论基础。关系数据库之所以取得了巨大成功和广泛应用，就是因为它具有适合关系运算的集合运算、投影、选择、连接和商运算的数学基础，以及以这些运算为基础而建立起来的其它各种运算。

【辅导内容】给出本章的学习目标、学习方法、学习重点、学习要求、关联知识，以及相关概念的区别。然后，给出本章的习题解析、技能实训，以及知识拓展（从关系数据库到新一代数据库技术）。

【能力要求】通过学习引导，掌握本章的知识要点；通过习题解析，深入理解和掌握关系模型的基本理论知识（关系代数和关系运算）；通过技能实训，熟练掌握在 SQL Server 2008 环境中创建数据库的基本方法；通过知识拓展，了解关系数据库所存在的不足以及关系数据库技术与其它技术结合形成的数据库新技术及其发展趋势。通过本章学习，打下关系数据库的理论概念基础。

§ 2.1 学习引导

本章介绍了关系数据模型的组成、关系代数、关系演算和关系代数表达式的优化。关系数据模型是关系数据库的理论基础，更是关系数据库结构化查询 SQL 的理论基础。

2.1.1 学习导航

1. 学习目标

本章从关系数据库的基本概念——关系（表）和关系模式开始，逐步深入讨论关系模型的三要素（关系数据结构、关系操作和关系完整性约束）、关系代数、关系演算关系和代数表达式的优化。本章的学习目标：一是深入理解关系模型和关系运算的基本概念；二是熟练掌握关系完整性约束，以及基于外键的关系数据库模式导航图；三是熟练掌握关系代数的主要操作以及基于数据库模式导航图构造关系代数查询表达式。

2. 学习方法

本章学习的关键是通过大量实例做一些关系代数运算的习题，深刻理解并领会关系数据库模式在构造关系代数查询表达式中的作用，达到举一反三、融会贯通的学习目的。

3. 学习重点

关系数据模型的重点是关系模型的三要素（关系数据结构、关系操作和关系完整性约束）和关系代数的基本运算，难点是关系演算关系和代数表达式的优化。

4. 学习要求

本章主要介绍关系模型的三要素和关系代数的基本运算，涉及关系数据库的许多基本概念，例如什么是关系、关系模式、数据库模式、超键、候选键、主键和外键；关系模型的完整性约束有哪些；关系代数有哪些主要运算等。要求深刻理解关系模型的基本概念、关系完整性约束之外，还要熟练掌握关系代数的运算和关系演算关系法则以及代数表达式的优化方法。

5. 关联知识

关系代数、元组关系演算和域关系演算是评估实际系统中查询语言能力的标准和理论基础，与下一章所要介绍的结构化查询语言（Structured Query Language, SQL）构成一个完整的关系数据库查询语言体系。

关系演算是以数理逻辑的谓词演算为基础的，把谓词演算（Predicate Calculus）推广到关系运算中就构成了关系演算（Relational Calculus）。

关于数理逻辑的理论知识可参见本书作者李云峰，李婷编著的《计算机科学导论》第10章“离散结构”（中国水利水电出版社2014年出版）。

2.1.2 相关概念的区别

关系数据模型提供了一系列操作的定义，这些操作称为关系操作，关系数据操作的主要特点是操作对象和操作结果都是数据的集合。在本章学习过程中，应注意以下概念的区别。

1. 笛卡尔积操作、连接操作、等值连接操作、自然连接操作的区别

笛卡尔积操作是一种二元操作，它是将任意两个关系的元组组合在一起形成一个新的关系。若关系R有m个元组，关系S有n个元组，笛卡尔积操作的结果是得到 $m \times n$ 个元组。

连接操作也称为 θ 连接， $\theta \in \{=, <, >, \leq, \geq, \neq\}$ 。它是从两个关系的笛卡尔积中选取R关系在A属性组上的值与S关系在B属性组上的值满足比较关系 θ 的元组，记为 $R \bowtie S = \sigma_{A \theta B}(R \times S)$ 。

等值连接操作是指 θ 为“=”的连接操作，它是从笛卡尔积中分别选取A、B属性值相等的那些元组组成一个新的关系。等值连接不除去重复的属性。

自然连接操作是一种特殊的等值连接，它要求关系R与S中具有同名的属性，并且在结果中把重复的属性列除去。自然连接一定是等值连接，但等值连接不一定是自然连接。

2. 关系数据库的型与关系数据库的值的区别

关系数据库中的型也称为关系数据库模式，是关系数据库的描述，包括若干域的定义以及在这些域上定义的若干关系模式。

关系数据库的值则是这些关系模式在某一时刻对应的关系的集合，通常称为关系数据库。

3. 关系代数和关系演算的区别

我们可以用关系代数表示关系操作，也可以用谓词演算来表达关系操作，通常称为关系演算。用关系代数表示关系的运算，须标明关系操作的序列，因而以关系代数为基础的数据库语言是过程语言。用关系演算表达关系的操作，只要说明所要得到的结果，而不必标明操作的过程，因而以关系演算为基础的数据库语言是非过程语言。

4. 关系代数、元组关系演算、域演算的等价性

关系代数和关系演算所依据的理论基础是相同的，因此可以进行相互间的转换。在讨论元组关系演算时，实际上就研究了关系代数中5种基本运算与元组关系演算间的相互转换；在讨论域关系演算时，实际上也涉及了关系代数与域关系演算间的相互转换。因此，关系代数、元组关系演算、域演算3类关系运算是可以相互转换的，它们对于数据操作的表达能力是等价的。再结合安全性的考虑，经过进一步的分析，人们已经证明了如下重要结论：

- ① 每一个关系代数表达式都有一个等价的安全的元组演算表达式。
- ② 每一个安全的元组演算表达式都有一个等价的安全的域演算表达式。
- ③ 每一个安全的域演算表达式都有一个等价的关系代数表达式。

按照上述3个结论，即得到关系代数、元组关系演算和域演算的等价性。

(2) 不同的列可以出自同一个域, 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。

(3) 行或列位置顺序无关紧要, 即行或列的次序可以任意交换。

(4) 任意两个元组不能完全相同。

(5) 分量必须取原子值, 即每一个分量必须是不可再分的数据项。

一个关系对应一个二维表, 但一个二维表不一定都能成为一个关系, 如复式表格(数据项下还有子项)就不是一个关系, 只有符合一定要求的二维表才能表示关系。

[参考答案] C。

5. 设关系 R 和 S 的元数分别是 3 和 4, 关系 T 是 R 与 S 的笛卡尔积, 即 $T=R \times S$, 则关系 T 的元数是 ()。

- A. 7 B. 9 C. 12 D. 16

【解析】笛卡尔积的定义是设关系 R 和 S 的元数分别是 r 和 s, R 和 S 的笛卡尔积是一个 $(r+s)$ 元属性的集合, 每一个元组的前 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。所以关系 T 的属性元数是 $3+4=7$ 。

[参考答案] A。

6. 数据库的完整性是指数据库的正确性和相容性。下列叙述不是 DBMS 的完整性控制机制的是 ()。

- A. 提供定义完整性约束
B. 检查用户发出的操作请求是否违背了完整性约束条件
C. 系统提供一定的方式让用户标识自己的名字或身份, 用户进入系统时, 由系统核对用户提供的身份标识
D. 如果发现用户的操作请求使数据违背了完整性约束条件, 则采取一定的动作来保证数据的完整性

【解析】C 选项叙述的是数据库安全性控制的方法之一, 所以 C 不是完整性控制机制的内容。DBMS 的完整性控制机制包括如下 2 方面:

(1) 定义功能: 提供定义完整性约束。

(2) 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的条件。

如果发现用户的操作请求使数据违背了完整性约束条件, 则采取一定的动作来保证数据的完整性。由 DBMS 完整性控制机制可知, A、B 和 D 选项说法都是正确的。

[参考答案] C。

7. 对关系模型叙述错误的是 ()。

- A. 建立在严格的数学理论、集合论和谓词演算公式基础之上
B. 微机 DBMS 绝大部分采取关系数据模型
C. 用二维表表示关系模型是其一大特点
D. 不具有连接操作的 DBMS 也可以是关系数据库管理系统

【解析】关系模型采用二维表表示实体及实体间的联系, 实体间的联系是通过不同关系中的公共属性来实现的。若关系 DBMS 不提供连接操作, 将无法完成涉及多个表之间的查询操作。所以 A、B 和 C 选项的描述都是正确的。

[参考答案] D。

8. 有两个关系 R 和 S, 分别包含 15 个和 10 个元组, 则在 $R \cup S$, $R-S$, $R \cap S$ 中, 不可能出现

这里要求关系 R 和关系 S 具有相同的属性个数。对于选项 D, R-S 表示关系 R 和关系 S 的差运算, 由属于关系 R 但不属于关系 S 的元组组成的集合, 要求关系 R 和关系 S 具有相同的属性个数。综上所述可知, B、C 和 D 选项都是错误的。

[参考答案] A。

2.2.2 填空题

1. 在关系模型中, 实体和实体之间的联系是由单一的结构类型——关系表来表示的, 关系模型的物理表示为_____。

【解析】关系模式是对关系的描述, 例如对于一个二维表格, 表格的表头就是该表格所表示的关系的数据结构的描述。表的每一行表示一个元组, 表的每一列对应一个域。

[参考答案] 二维表格。

2. 数据库完整性的实现应该包括两个方面: 一是系统要提供定义完整性约束条件的功能; 二是提供_____的方法。

【解析】数据模型应该反映和规定本数据模型必须遵守的、基本的、通用的完整性约束条件, 它主要包括实体完整性规则和引用完整性规则, 应该由系统自动支持。此外, 数据模型还应该提供定义完整性约束条件的机制, 以反映具体应用所涉及的数据必须遵守的特定的语义约束条件, 它主要包括用户定义完整性规则, 不是由应用程序实现这部分功能。

[参考答案] 检查完整性约束条件。

3. 关系的数据操纵语言按照表达式查询方式可以分为两大类: 关系代数和_____。

【解析】关系代数是一种抽象的查询语言, 是关系数据操纵语言的一种传统表达方式, 它是用对关系的运算来表达查询的。关系演算以数理逻辑中的谓词演算公式为基础的。

按照谓词变元的不同, 关系演算可以分为元组关系演算和域关系演算。它也是关系数据操纵语言的一种表达方式。

[参考答案] 关系演算。

4. 在关系模型中, 若属性 A 是关系 R 的主关键字, 则在 R 的任何元组中, 属性 A 的取值都不允许为空, 这种约束称为_____。

【解析】关系数据库在操作时, 主要考虑 3 类完整性规则:

(1) 实体完整性规则: 这条规则要求关系中的元组的主键不能为空值。如果出现空值, 那么主键就起不到唯一标识元组的作用。

(2) 参照完整性规则: 这条规则要求“不允许引用不存在的实体(即元组、记录等)”。

(3) 用户定义完整性规则。这是针对某一具体数据的约束条件, 由应用环境决定; 它反映了某一具体应用所涉及的数据必须满足的语义要求。其中, 实体完整性规则是对关系中主属性(主键)的值的约束规则, 即主键值不允许为空。

[参考答案] 实体完整性。

5. 关系是一种规范化了的二维表格, 其行称为_____, 其列表示_____。

【解析】关系与二维表格、传统的数据文件既有相似之处, 又有区别。从严格意义上讲, 关系是一种规范化了的二维表格, 其行称为元组, 其列表示属性。

[参考答案] 元组, 属性。

6. 两个关系没有公共属性时, 其自然连接操作表现为_____操作。

【解析】自然连接是在两个关系的公共属性上进行的等值连接。但是当两个关系没有公共属性

时，自然连接操作就演变为笛卡尔积操作。

【参考答案】 笛卡尔积。

7. 在域关系演算中，域变量的变化范围是_____。

【解析】 在关系演算中，根据所用变量不同分为元组关系演算和域关系演算。域关系演算以域为变量。

【参考答案】 某个值域。

8. 根据关系模型的完整性规则，一个关系中的主键不能有_____个。

【解析】 实体完整性规则中指出一个关系中的主键值不能出现相同或为空值。

【参考答案】 两。

9. 设关系 R 有 K1 个元组，关系 S 有 K2 个元组，则关系 R 和 S 的自然连接后的结果关系的元组数目是_____个。

【解析】 两个关系的连接操作只包含那些满足连接条件的元组的组合。

【参考答案】 $\leq K1 \times K2$ 。

10. 设关系 R 有 K1 个元组，关系 S 有 K2 个元组。则关系 R 和 S 的笛卡尔积有_____个元组。

【解析】 两个关系的笛卡尔积包含两关系的所有元组的组合。

【参考答案】 $K1 \times K2$ 。

2.2.3 问答题

1. 关系模型由哪几部分组成？

【解析】 关系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. 笛卡尔积、等值连接、自然连接三者之间有什么区别？

【解析】 笛卡尔积是一个基本操作，而等值连接和自然连接是组合操作，它是两个关系中所有公共属性进行等值连接的结果。等值连接不把重复的属性去除，而自然连接是去除重复属性的等值连接，自然连接一定是等值连接，但等值连接不一定是自然连接。

3. 为什么关系中的元组没有先后顺序？

【解析】 由于关系定义为元组的集合，而集合中的元素是没有顺序的，因此关系中的元组对用户而言没有先后顺序，在关系中区分哪一个元组依据的是主键的值而不是行的顺序。

4. 关系代数运算与关系演算运算有什么区别？

【解析】 关系代数运算是以关系为运算对象，由并、差、笛卡尔积、投影、选择 5 个基本操作进行有限次的复合运算。关系演算运算是以元组或域为运算对象，由数理逻辑的谓词进行有限次的演算。

5. 关系查询语言根据其理论基础的不同分为哪两类？

【解析】 关系代数语言：查询操作是以集合操作为基础运算的 DML 语言。（非过程性弱）

关系演算语言：查询操作是以谓词演算为基础运算的 DML 语言。（非过程性强）

6. 关系演算有哪两种？

【解析】 关系演算可分为元组关系演算和域关系演算。前者以元组为变量，后者以属性（域）为变量。

7. 为什么对关系代数表达式进行优化？

【解析】 关系代数表达式由关系代数操作组合而成。操作中，笛卡尔积和连接操作费时最多。

如果直接按表达式书写的顺序执行，将花费很多时间，并生成大量的中间结果，致使效率较低。如果在执行前，由 DBMS 的查询子系统对关系代数表达式进行优化，尽可能先执行选择和投影操作，则进行笛卡尔积或连接操作时可以减少中间结果，并节省时间。

8. 简述查询优化的优化策略。

【解析】在关系数据库中，对于相同的查询请求存在着多种实现策略。在查询处理过程中，数据库系统必须能从查询的多个执行策略中选择合适的执行策略。优化策略的基本思想如下：

(1) 在关系代数表达式中尽可能早地执行选择操作。

(2) 把笛卡尔积和随后的选择操作合并成联接运算。

(3) 同时计算一连串的选择和投影操作，以免分开运算造成多次扫描文件，从而能节省操作时间。

(4) 如果在一个表达式中多次出现某个子表达式，应该将该子表达式预先计算出结果并保存起来，以免重复计算。

(5) 适当地对关系文件进行预处理。

(6) 在计算表达式之前应先估计一下怎么计算合适。

9. 关系数据语言有哪些类型和特点？

【解析】关系数据语言可以分为三类：

关系代数语言，例如 ISBL。

关系演算语言，分为元组关系演算语言，例如 ALPHA、QUEL 和域关系演算语言，例如 QBE。具有关系代数和关系演算双重特点的语言，例如 SQL。

关系数据语言的共同特点是具有完备的表达能力，是非过程化的集合操作语言、功能强，能够嵌入高级语言中使用。

10. 在参照完整性中，为什么外键属性的值也可以为空？什么情况下才可以为空？

【解析】在参照完整性中，外键属性的值可以为空，它表示该属性的值尚未确定。但前提条件是该外键属性不是其所在关系的主属性。例如在下面的“学生”表中，“专业号”是一个外键，不是学生表的主属性，可以为空。其语义是，该学生的专业尚未确定。

学生(学号，姓名，性别，专业号，年龄)；

专业(专业号，专业名)。

而在下面的“选修”表中，“课程号”虽然也是一个外键属性，但它又是“选修”表的主属性，所以不能为空。因为关系模型必须满足实体完整性。

课程(课程号，课程名，学分)；

选修(学号，课程号，成绩)。

2.2.4 应用题

1. 设有三个关系：S(Sno, Sname, Age, Sex)、SC(Sno, Cno, Grade)、C(Cno, Cname, Teacher)，试用关系代数表达式表示下列查询语句：

【解析】按照要求，对各问题给出查询表达式如下：

(1) 查询“张三”老师所授课程的课程号、课程名。

$$\Pi_{Cno, Cname}(\delta_{Teacher=张三}(C))$$

(2) 查询年龄大于 22 岁的男学生的学号与姓名。

$$\Pi_{Sno, Cname}(\delta_{Age>22 \wedge Sex=男}(S))$$

(3) 查询学号为 20121101 的学生所学课程的课程名与任课教师名。

$$\Pi_{Cname, Teacher}(\delta_{Sno=20121101}(SC \in C))$$

(4) 查询至少选修“李四”老师所授课程中一门课的女学生的姓名。

$$\Pi_{Sname}(\delta_{Sex='女' \wedge Teacher='李四'}(S \in SC \in C))$$

(5) 查询“钱学斌”同学不学的课程的课程号。

$$\Pi_{Cno}(C) - \Pi_{Cno}(\delta_{Sname='钱学斌'}(S \in SC))$$

(6) 查询全部学生都选修的课程的课程号与课程名。

$$\Pi_{Cno, Cname}((\delta_{Sno, Cno}(SC) \div \Pi_{Sno}(S)) \in C)$$

(7) 查询选修课程包含“张三”老师所授全部课程的学生的学号。

$$\Pi_{Sno, Cno}(SC) \div \Pi_{Cno}(\delta_{Teacher='张三'}(C))$$

2. 在教学管理数据库中，查询女同学选修课程的课程名和任课教师名，并且要求：

- (1) 写出该查询的关系代数表达式。
- (2) 画出该查询初始的关系代数表达式的语法树。
- (3) 使用优化算法，对语法树进行优化，并画出优化后的语法树。
- (4) 写出查询优化的关系代数表达式。

【解析】

(1) 该查询的关系代数表达式为：

$$\Pi_{Cname, Teacher}(\delta_{Sex='女' \wedge S.Sno=SC.Sno \wedge SC.Cno=C.Cno}(S \times SC \times C))$$

(2) 查询初始的关系代数表达式语法树如图 2-1 所示。

(3) 使用优化算法对语法树进行优化，优化后的语法树如图 2-2 所示。

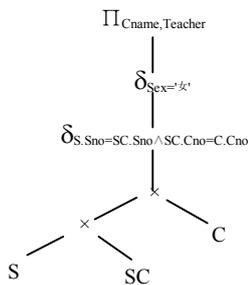


图 2-1 关系代数语法树

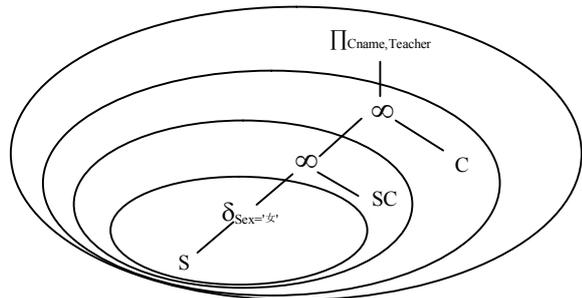


图 2-2 优化后的语法树

(4) 查询优化的关系代数表达式为：

$$\Pi_{Cname, Teacher}((\delta_{Sex='女'}(S) \in SC) \in C)$$

3. 设教学管理系统数据库中有两个关系模式：Student(Sno, Sname, Age, Sex); SC(Sno, Cno, Grade)。查询选修了课程 Cno='C110'的学生姓名，要求给出查询优化语法树。

【解析】根据题意，优化步骤如下：

(1) 查询转换成某种内部表示（如语法树）：关系代数语法树如图 2-3 所示。

(2) 代数优化：利用优化算法把关系代数语法树转换成标准（优化）形式。为此，利用等价规则 4 和规则 6，把选择 $\delta_{SC.Cno='C110'}$ 移到叶端形成标准（优化）图，如图 2-4 所示。

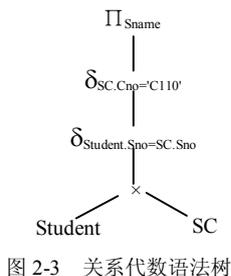


图 2-3 关系代数语法树

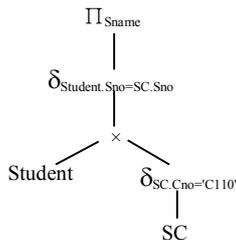


图 2-4 优化后关系代数语法树

§ 2.3 技能实训

本章介绍两项实训内容：创建数据表和管理数据表。数据库的操作大多是以数据表为对象的。

2.3.1 创建数据表

【实训背景】

数据表是 SQL Server 2008 中最基本的数据库对象，用于存储数据库中的所有用户数据。在建立了 SQL Server 数据库文件（空数据库）后，首先要在该数据库文件中创建的数据库对象就是数据表。在创建数据表前，先要确定表的结构、表的字段组成、每个字段的数据类型和字段属性等。标准数据表的创建既可以用关系数据库语言 SQL 提供的创建表语句实现，也可以用对象资源管理器来实现。

在定义基本表时，对于某些列和表要进行完整性约束条件定义，以实现实体完整性、参照完整性和用户定义完整性。通常完整性约束根据其作用的范围分为列级约束和表级约束。列级约束通常包含在列的定义中，对该列进行约束；表级约束通常被放在该表最后一列定义之后，对整个表进行约束。另外，有些约束既是列级约束，又是表级约束。

【实训目的】

- (1) 了解 SQL Server 2008 表的结构特点。
- (2) 了解 SQL Server 2008 的基本数据类型。
- (3) 掌握使用 SQL Server Management Studio 创建数据表的方法。
- (4) 掌握使用 T-SQL 语句创建数据表的方法。
- (5) 理解约束的概念。

【实训内容】

- (1) 利用 SQL Server Management Studio 创建、修改和删除数据表。
- (2) 利用 T-SQL 语句创建、修改和删除数据表。
- (3) 创建主键约束、缺省约束、Check 约束、唯一约束和外键约束。

【实训步骤】

1. 利用 SQL Server Management Studio 创建、修改和删除数据表

(1) 启动 SQL Server Management Studio: 执行“开始”→“所有程序”→Microsoft SQL Server 2008→SQL Server Management Studio 命令，显示“连接服务器”对话框，单击“连接”按钮，按默认方式连接服务器。

(2) 新建数据表: 在“对象资源管理器”窗口中，依次展开结点“数据库”→Sales，右击“表”，

在快捷菜单中选择“新建表”选项，显示“新建表”对话框。此时，可按照主教材表 9-7 学生信息表（StudentInfor）、表 9-8 课程信息表（CourseInfor）、表 9-13 学生修课信息表（StudCourse）的定义创建数据表。

（3）修改数据表：在“对象资源管理器”窗口中，依次展开结点“数据库”→Sales→“表”，在快捷菜单中选择“设计”选项，在查询窗口中选择并显示表的结构。此时，进行如下修改：

- 在学生信息表中，增加备注字段，字段名 memo，字段类型 nvarchar，字段长度 200，允许为空。
- 在数据表 StudentInfor 中，删除备注字段 memo。
- 在数据表 CourseInfor 中，对于字段 CourseName 设置 UNIQUE 约束。
- 在数据表 StudCourse 中，对于字段 ExamGrade 设置 CHECK 约束，其取值在 0~100 之间。
- 在数据表 StudCourse 中，对于字段 Scode 设置 FOREIGN KEY 约束，其取值参考数据表 StudentInfor 中 Scode 字段取值。

（4）删除数据表：在“对象资源管理器”窗口中，依次展开结点“数据库”→Sales→“表”，右击要删除的表名，例如修课信息表（StudCourse），在快捷菜单中选择“删除”选项，显示“删除对象”窗口。

2. 利用 T-SQL 语句创建、修改和删除数据表

在 SQL 编辑器中，启动 SQL Server Management Studio，然后进行语句操作。

（1）利用 T-SQL 的 CREATE TABLE 命令语句创建学生信息表（StudentInfor）、课程信息表（CourseInfor）、学生修课信息表（StudCourse）。

（2）利用 T-SQL 的 ALTER TABLE 命令语句修改数据表：

- 在学生信息表（StudentInfor）中，增加身份证号键字段，字段名 code，字段类型 char，字段长度 18，允许为空。
- 在学生信息表（StudentInfor）中，对于字段 code 设置 UNIQUE 约束。
- 在学生信息表（StudentInfor）中，删除身份证号键字段 code。
- 在学生修课信息表（StudCourse）中，对于字段 ExamGrade 设置 CHECK 约束，其取值在 0~100 之间。
- 在学生修课信息表（StudCourse）中，对于字段 Ccode 设置 FOREIGN KEY 约束，其取值参考课程信息表（CourseInfor）中 Ccode 字段取值。

（3）利用 T-SQL 的 DROP TABLE 命令语句删除学生信息表（StudentInfor）。

2.3.2 管理数据表

【实训背景】

在创建一个表结构后，此时还只是一张空表，表中并没有数据信息，用户可以使用插入操作录入数据，也可以对表中数据进行修改、删除等更新操作。

【实训目的】

- （1）掌握使用 SQL Server Management Studio 对数据表进行插入、修改和删除数据的操作。
- （2）掌握使用 T-SQL 语句对数据表进行插入、修改和删除数据的操作。

【实训内容】

- （1）利用 SQL Server Management Studio 向数据表中添加、修改和删除数据。
- （2）利用 T-SQL 语句向数据表中添加、修改和删除数据。

【实训步骤】**1. 利用 SQL Server Management Studio 实现数据插入、修改和删除**

(1) 插入数据: 启动 SQL Server Management Studio, 在“对象资源管理器”窗口中, 依次展开结点“数据库”→Sales→“表”, 向学生信息表 (StudentInfor)、课程信息表 (CourseInfor) 和学生修课信息表 (StudCourse) 中添加数据。数据内容可采用主教材图 1-10 中的学生关系、课程关系和学习关系中的数据信息。

(2) 修改数据: 在“对象资源管理器”窗口中, 对上述数据表中的数据进行如下修改:

- 在学生信息表中, 将表中的数据改为你同班同学的数据信息。
- 在课程信息表中, 将表中的数据改为你本学期开设课程的数据信息。
- 在修课信息表中, 将表中的数据改为你本人上学期各门课程考试成绩。

(3) 删除数据: 在“对象资源管理器”窗口中, 对上述数据表中的数据进行删除。删除修课信息表中成绩小于 60 分的记录, 将其置空。如果没有成绩小于 60 分的记录, 则删除你认为在教学管理中比较次要的数据信息。

2. 利用 T-SQL 语句实现数据添加、修改和删除

(1) 插入数据: 启动 SQL Server Management Studio, 在 SQL 编辑器中, 利用 T-SQL 语句 INSERT INTO 命令向学生信息表 (StudentInfor)、课程信息表 (CourseInfor) 和学生修课信息表 (StudCourse) 中添加数据。数据内容可采用主教材图 1-10 中的学生关系、课程关系和学习关系中的数据信息。

(2) 修改数据: 启动 SQL Server Management Studio, 在 SQL 编辑器中, 利用 T-SQL 语句 UPDATE 命令修改表数据:

- 在学生信息表中, 将表中的数据改为你同班同学的数据信息。
- 在课程信息表中, 将表中的数据改为你本学期开设课程的数据信息。
- 在修课信息表中, 将表中的数据改为你本人上学期各门课程考试成绩。

(3) 删除数据: 启动 SQL Server Management Studio, 在 SQL 编辑器中, 利用 T-SQL 语句 DELETE 命令删除修课信息表中成绩小于 60 分的记录, 将其置空。如果没有成绩小于 60 分的记录, 则删除你认为在教学管理中比较次要的数据信息。

§ 2.4 知识拓展——从关系数据库到新一代数据库技术

为了使读者对数据库技术有更加全面的了解, 这里简要介绍目前占有统治地位的关系数据库、关系数据库系统的局限性和新一代数据库技术。

2.4.1 关系数据库的概况

目前广泛使用的是基于关系数据模型 (Relational Data Model, RDM) 的关系数据库 (Relational Data Base, RDB)。关系数据模型通常简称为关系模型, 它是在层次模型和网状模型之后发展起来的一种逻辑数据模型, 具有严格的数据理论基础, 并且其表示形式更加符合现实世界中人们的常用形式。把关系技术最早引入数据库领域的是美国 IBM 公司 San Jose 研究所的埃德加·科德 (Edgar Frank Codd) 博士 (研究员)。1970 年 6 月他在美国计算机学会会刊《Communications of ACM》上发表了名为“大型共享数据库的数据关系模型” (A Relational Model of Data for Large Shared Databanks) 的论文, 把数学中一个称为关系代数的分支应用到存储大量数据问题中, 首次明确而清晰地提出了关系模型的概念, 从而奠定了关系数据库的理论基础, 开创了数据库的关系方法和关

系规范化理论的研究，并且从理论到实践都取得了辉煌成果。

在理论上，他提出了数据的关系表示与物理实现的独立，确立了完整的理论、数据依赖理论以及关系数据库的设计理论等；给出了关系模型的严格定义及逻辑数据库结构的规范化标准，并且在关系的数学定义基础上，提出了实现独立的数据库操纵的非过程化操纵语言，该语言集数据库的数据定义、数据操纵和数据控制于一体。使用方便灵活，不过分依赖于数据结构的细节。在实践上，开发了许多著名的关系数据库管理系统，关系数据库系统就是采用关系数据模型构建数据库的。

正是由于关系模型所具有的这些主要优点，已成为目前使用最为广泛的数据模型，使得基于关系模型的数据库管理系统成为当今实用系统的主流。

2.4.2 关系数据库的局限性

围绕数据库结构和模型的演变，数据库技术在发展过程中经历了网状数据库、层次数据库和关系数据库。网状数据库和层次数据库主要解决了数据的集中和共享问题，但在数据独立性和抽象级别上仍有很大欠缺。用户在这两种数据库上进行存取操作时，必须明确数据的存储结构，具体指明存取路径。为了弥补这些不足，人们开始将目光转向关系数据库管理系统。20世纪80年代以来，软件厂商推出的数据库管理系统几乎都支持关系模型，关系数据库在目前的数据库应用领域中占有绝对的统治地位。然而，当试图把关系数据库系统运用到现代新的应用领域时，便显现出传统关系数据库的一些局限和不足，主要体现以下4个方面。

1. 关系数据模型的表达能力有限

关系数据库采用的是高度结构化的表格结构的数据模型，是面向机器的语法模型，语义表达能力差。它们只能存储离散的数据和有限的数据之间的关系，难以表示客观存在的超文本、图形、图像、CAD图件、声音等多种复杂对象；缺乏对工程、地理、测绘等领域对象所拥有的许多复杂异形结构的抽象机制和非结构化数据的表达能力；不能有效地处理在许多事务处理中用到的多维数据，无法描述现实世界的复杂对象和揭示数据之间的深层次含义及内在联系，缺乏数据抽象。

2. 关系模型支持的数据类型有限

关系数据库管理系统只能理解、存储和处理诸如整数、浮点数、字符串、日期、货币等简单数据类型，不提供自定义数据类型机制和扩展自身数据类型集的能力。复杂的应用只能由用户编写程序，借助高级程序设计语言功能利用简单的数据类型进行描述和支持，这无疑加重了用户的负担，也不能保证数据的一致。特别是面对Internet飞速发展而涌现出来的大量的如图形、声音、大文本、时间序列和地理信息等这样的非结构化复杂数据类型，关系系统更显得力不从心。

3. 关系数据库所具备的功能有限

关系数据库系统存储和管理的对象是数据，对数据施行存储、管理、查询、排序、生成报表等简单操作，缺乏对知识的表达、管理和处理能力，不具备演绎和推理的功能。由于数据库中的数据反映的是客观世界中的静态和被动的客观事实，不能够在发现异常情况时主动响应和通过某些操作处理意外事件，因而不能满足MIS、DSS和AI等领域中的高层管理和决策需求，从而极大地限制了数据库技术的高级应用。

4. 关系数据库中的SQL能力有限

关系数据库的SQL是一种面向集合的非过程性语言，而作为主语言的通用程序语言（例如C语言）是面向过程的语言，所以这两种语言的类型不匹配。关系数据库的一条SQL查询(SELECT)语句通常是将含有多行的数据集（查询结果）返回给应用程序，但宿主语言（例如C语言）每次

一般只能表示和处理一个元组的数据，即 SQL 是在集合上操作，而宿主语言是在集合的成员上操作。这种表示和处理能力上的不匹配使得查询结果的输出和显示变得比较麻烦，所以才引入游标机制将对集合的操作转换成对单个元组的操作，以此弥补数据库操纵语言与宿主语言之间的不匹配，我们将其称为阻抗失配。

面对数据库应用领域的不断扩展和用户要求的多样化、复杂化，传统的数据库技术遇到了严峻的挑战。也正是关系数据库存在的上述这些局限，决定了新一代数据库技术的研究方向。

2.4.3 新一代数据库技术

随着计算机科学技术的飞速发展，数据库系统的应用已经从商业数据处理迅速扩展到诸如超大型数据检索、数据仓库、联机数据分析、数据挖掘等许多应用领域。这些应用领域的特点是数据量大、复杂度高、用户数目多，对数据库系统的处理能力提出了非常高的要求，这些应用需求也直接驱动了新一代高性能数据库的研究，各种新型数据库系统应运而生。

1. 数据库技术与应用领域相结合

为了适应数据库应用多元化的要求，在传统数据库技术的基础上，结合各个应用领域的特点，研究和开发出适合该应用领域的数据库技术，并在此基础上产生和发展了一系列支持特殊应用领域的新型数据库，例如数据仓库是信息领域近年来迅速发展起来的数据库技术，数据仓库的建立能充分利用已有的资源，把数据转换为信息，从中挖掘出知识，提炼出智慧，最终创造出效益；工程数据库系统的功能是用于存储、管理和使用面向工程设计所需要的工程数据；统计数据是来自于国民经济、军事、科学等各种应用领域的一类重要的信息资源，由于对统计数据操作的特殊要求，从而产生了统计学和数据库技术相结合的统计数据库系统等。数据库技术在特定领域的应用，为数据库技术的发展提供了源源不断的动力。

2. 数据库技术与多学科技术相结合

随着数据库技术应用领域的不断扩展，各种学科技术与数据库技术的有机结合和渗透，从而使数据库领域中的新内容、新应用、新技术层出不穷，数据库的许多概念、技术内容、应用领域，甚至某些原理都有了重大的发展和变化，形成和产生了一系列新型数据库系统。例如：

- 数据库技术与分布处理技术相结合，产生了分布式数据库系统；
- 数据库技术与并行处理技术相结合，产生了并行数据库系统；
- 数据库技术与人工智能相结合，产生了演绎数据库系统、知识库和主动数据库系统；
- 数据库技术与多媒体处理技术相结合，产生了多媒体数据库系统；
- 数据库技术与模糊技术相结合，产生了模糊数据库系统；
- 数据库技术与移动通信技术相结合，产生了移动数据库系统；
- 数据库技术与 Web 技术相结合，产生了 Web 数据库系统；
- 数据库技术与传感器网络相结合，产生了传感器网络数据库系统等。

3. 数据库技术与面向对象方法相结合

20 世纪 80 年代后期出现的面向对象设计方法对计算机的各个领域，包括程序设计、软件工程、信息系统设计、计算机硬件设计等都产生了深远影响。同时，也给面临机遇和挑战的数据库技术带来了机会和希望。数据库领域研究人员借鉴面向对象方法和技术，提出了面向对象模型和新一代数据库的发展方向，促进了数据库在一个新的技术平台上的继续发展。

在数据库技术的发展过程中，数据的组织模型经历了从层次模型、网状模型、关系模型到最新的面向对象模型的发展历程，数据模型的每一次变化都为数据的访问和操作带来新的特点和功能。

关系数据模型的产生使人们可以不再需要知道数据的物理组织方式就可以逻辑地访问数据。面向对象数据模型的产生突破了关系模型中数据必须是简单二维表的平面结构,使数据模型的表达能力更强,更能表达人们对数据的需求。

数据库技术的新发展除了表现在数据模型越来越复杂,数据模型包含的语义越来越多外,还呈现出多角度、全方位的发展态势。如分布式数据库、面向对象数据库、多媒体数据库、主动数据库、并行数据库、演绎数据库、模糊数据库、联邦数据库等,形成了共存于当今社会的数据库大家族,而且这些都是数据库技术重要的发展方向。

充分利用相关学科领域的技术成果,使数据库技术与多学科技术相互结合与相互渗透是当前数据库技术发展的重要特征。新一代数据库技术的发展,一方面立足于传统数据库已有的成果和技术,并在其基础上改进;另一方面,立足于新的应用需求和计算机技术的发展,研究全新的数据库系统。各种新型数据库的研究与发展概况,将在后面各章的知识拓展中予以介绍。