第 3 章　关系数据库

知识点：

- ✧ 关系模型的相关概念
- ✧ 关系及关系的性质
- ✧ 关系的完整性
- ✧ 关系代数运算

本章导读：

本章主要是围绕关系数据模型的三要素来展开介绍关系模型的基本概念、关系操作和完整性约束条件，并着重介绍了关系操作的数学基础——关系代数。通过本章的学习可以理解关系模型的基本概念、关系的性质、完整性约束条件和关系代数的基本用法，掌握使用关系代数进行关系操作。

关系数据库是以数学方法为基础的数据库，它以数学方法处理数据库中的数据。1970 年 E. F. Codd 在美国计算机学会会刊《Communication of the ACM》上发表的题为“A Relational Model of Data for Shared Data Banks”的论文，开创了数据库系统的新纪元。以后，他连续发表了多篇论文，奠定了关系数据库的理论基础。

目前，关系数据库系统的研究取得了辉煌的成就，涌现出许多性能良好的商品化关系数据库管理系统（简称 RDBMS），如著名的 SQL Server、DB2、Oracle、Sybase，Informix 等。数据库的应用领域正迅速扩大。

3.1　关系模型

关系数据库系统是支持关系模型的数据库系统，关系数据库的核心是关系模型。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。

3.1.1　关系数据结构

关系模型的数据结构非常单一。在关系模型中，现实世界的实体以及实体间的各种联系均用关系来表示。在用户看来，关系模型中数据的逻辑结构是一张张二维表。由于关系模型是建立在集合代数基础上的，因而一般从集合的角度对关系数据结构进行定义。

1. **关系的数学定义**

- 域（Domain）

定义 2.1　域是一组具有相同数据类型的值的集合。又称值域（用 D 表示），域中包含的值的个数称为域的基数（用 m 表示），关系中域表示属性的取值范围。

例如，自然数、整数、实数、长度小于 25 字节的字符串集合、{0，1}、{男，女}、大于等于 0 且小于等于 100 的正整数等，都可以是域。

- 笛卡尔积（Cartesian Product）

定义 2.2　给定一组域 D_1，D_2，…，D_n，则 D_1，D_2，…，D_n 的笛卡尔积为：

$$D_1 \times D_2 \times \ldots \times D_n = \{(d_1,d_2,\ldots,d_n) | d_i \in D_i, i=1,2,\ldots,n\}$$

其中：

- 每一个元素（d_1，d_2，…，d_n）叫作一个 n 元组（n-tuple），或简称元组（Tuple）。
- 元素中的每一个值 di 叫作一个分量（Component）。
- 若 Di(i=1,2,…,n)为有限集，其基数（Cardinal number）为 m_i(i =1,2,…,n)，则 $D_1 \times D_2 \times \cdots \times D_n$ 的基数 M 为：

$$M = \prod_{i=1}^{n} m_i$$

笛卡尔积可表示成一个二维表。表中的每行对应一个元组，表中的每列对应一个域。例如现有三个域：

D_1=学号={0762101，0762202}

D_2=姓名={张三，李星，赵强}

D_3=性别={男，女}

则 D_1，D_2，D_3 的笛卡尔积为：

$D_1 \times D_2 \times D_3$ = {

(0762101，张三，男)，(0762101，张三，女).，

(0762101，李星，男)，(0762101，李星，女)，

(0762101，赵强，男)，(0762101，赵强，女)，

(0762202，张三，男)，(0762202，张三，女)，

(0762202，李星，男)，(0762202，李星，女)，

(0762202，赵强，男)，(0762202，赵强，女)，

}

其中(0762202，张三，男)、(0762202，李星，女)等都是元组。张三、0762101、赵强、男等都是分量。

该笛卡尔积的基数为 2×2×3=12，也就是说，$D_1 \times D_2 \times D_3$ 一共有 2×2×3=12 个元组。这 12 个元组可列成一张二维表，如图 3-1 所示。

- 关系（Relation）

定义 2.3　$D_1 \times D_2 \times \cdots \times D_n$ 的子集叫作在域 D_1，D_2，…，D_n 上的关系，用 R (D_1, D_2…, D_n) 表示。其中，R 表示关系的名字，n 是关系的目或度（Degree）。关系中的每个元素是关系中的元组，通常用 t 表示。当 n=l 时，称为单元关系（Unary relation）。当 n=2 时，称为二元关系（Binary relation）。

由于关系是笛卡尔积的有限子集，所以关系也是一个二维表。看到关系，也要想到关系背后的笛卡尔积，所以，关系的每列对应一个域，关系的一行对应一个元组。读者可以在图 3-1 的笛卡尔积中取出一个子集来构造一个学生关系。由于一个学生只有一个学号、姓名和性

别，所以笛卡尔积中的许多元组是无实际意义的，从中取出有实际意义的元组来构造学生关系。该关系的名字为学生，属性名就取域名，即学号、姓名、性别。则这个关系表示为：学生（学号，姓名，性别），如表 3-1 所示。

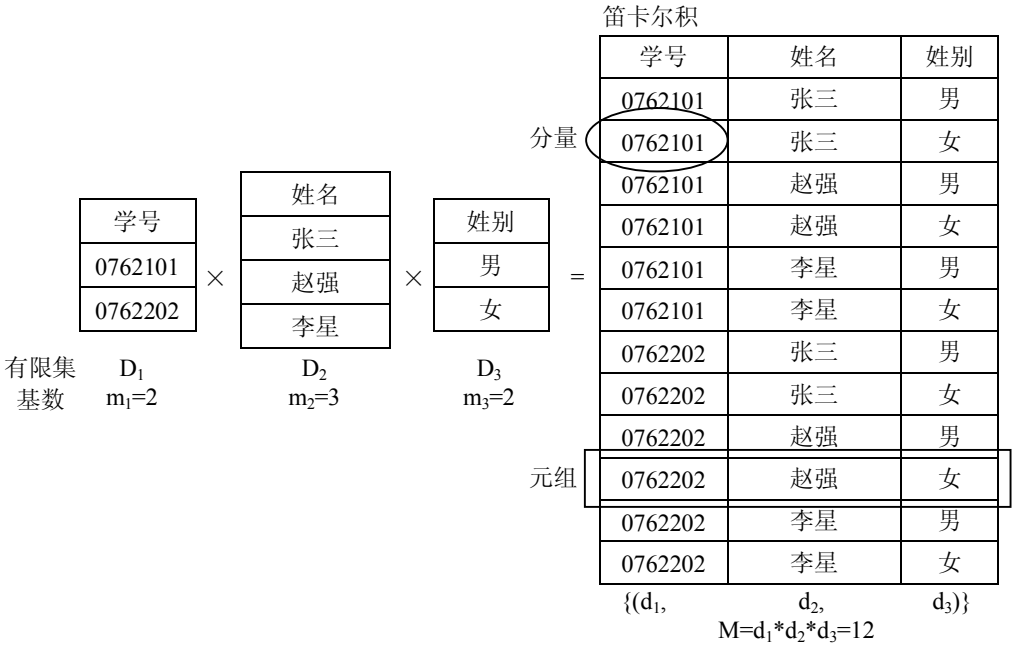


图 3-1 笛卡尔积 $D_1 \times D_2 \times D_3$

表 3-1 学生关系表

学号	姓名	性别
0762101	张三	男
0762202	李星	女
0762202	赵强	男

2. 关系中的常用术语

- 属性（Attribute）

关系中的列称为属性，属性用来表示实体的特征。属性有型和值之分，型是对数据特性的表示，它通过属性的名称、数据类型、数据宽度和值域等来描述；属性的值是指其具体取值。表的每列对应一个域。由于域可以相同，为了加以区分，同一关系中属性名（列名）不能相同。例如表 3-1 中，学号、姓名和性别各不相同。

- 元组（Tuple）

关系中的行称为元组，组成元组的元素称为分量。现实世界中的一个实体或实体间的一个联系在数据库中均用一个元组表示。例如表 3-1 中每一行都是一个元组，表示一个学生的信息。

- 候选键（Candidate Key）

关系中同时满足下列两个条件的属性或属性组称为候选键：

- 值可以唯一确定一个元组。
- 属性组中不包含有多余的属性。

例如表 3-1 中的“学号”候选键，如果“姓名”没有重名，“姓名”也是候选键。

- 主键（Primary Key）

关系中可能有多个候选键，为数据管理方便需选定一个作为主键。当然，如果关系中只有一个候选键，这个唯一的候选键就是主键。例如表 3-1 中的“学号”可以是主键。

- 主属性（Prime Attribute）和非主属性（Non-Key Attribute）

关系中候选键中的属性称为主属性，不包含在任何候选键中的属性称为非主属性。

3. 数据库中关系的类型

关系可以有三种类型：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：基本表是关系数据库中实际存在的表，它是实际存储数据的逻辑表示。
- 查询表：查询表是查询结果对应的表或查询中生成的临时表。
- 视图表：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。视图是为了查询方便、数据处理简便以及数据安全要求而设计的数据虚表。

4. 数据库中关系的性质

（1）列的同质性。每一列中的分量是同一类型的数据，来自同一个域。

（2）同一关系的列名具有不能重复性。关系中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。

（3）关系中列的顺序无关性。关系中列的顺序无关性说明关系中列的次序可以任意交换、重新组织，属性顺序不影响使用。

（4）关系中行的顺序无关性。关系中行的顺序无关性说明关系中行（元组）的顺序可以任意交换。

（5）关系中任意两个元组不能完全相同。

（6）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。

关系模型要求关系必须是规范化的，即要求关系模式必须满足一定的规范条件。这些规范条件中最基本的一条就是，关系的每一个分量必须是一个不可分的数据项。

5. 关系模式

关系的具体描述称为关系模式（Relation Schema）。一个关系模式通常从五个方面来描述关系，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 中属性所来自的域的集合，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。

例如，表 3-1 学生关系的关系模式是 R(U, D, Dom , F)，R 为学生，U 为{学号，姓名，性别}，D 为{学号，姓名，性别}，Dom 为{学号->学号，姓名->姓名，性别->性别}。

本章中关系模式仅涉及关系名、各属性名、域名、属性向域的映像四部分，属性间数据的依赖问题将在以后章节详细讨论。

关系模式可以简记为：R(U)或 R(A1,A2,…,An)。

其中 R 为关系名，A1，A2，…，An 为属性名。而域名及属性向域的映像常常直接说明为属性的类型、长度。

在关系数据库中，关系模式是型，关系是值。关系模式是静态的、稳定的，是关系的框架或结构。关系是按关系模式组织的数据表格，是关系模式在某一时刻的状态或内容。因为关系操作在不断更新着数据库中的数据，所以关系是动态的、随时间不断变化的。但在实际当中，人们常常把关系模式和关系都称为关系，这不难从上下文中加以区别。

6. 关系数据库

用关系模型来组织数据的数据库称为关系数据库。在一个给定的应用领域中，所有实体及实体之间的联系的关系描述的集合就构成一个关系数据库。在关系数据库中，实体以及实体间的联系都是用关系来表示的。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，是关系数据库的框架结构，是关系模式的集合。关系数据库的值是这些关系模式在某一时刻对应的关系的集合，所以关系数据库的型是数据库的结构，是静态的，关系数据库的值是数据库按结构装入数据后某一时刻的状态，是动态的。关系数据库模式与关系数据库通常统称为关系数据库。

3.1.2 关系操作

1. 关系操作的内容

关系模型提供了关系操作的能力，关系模型中常用的关系操作包括查询操作、更新操作和控制操作三大部分。

（1）查询操作：选择（Select）、投影（Project）、连接（Join）、除（Divide）、并（Union）、交（Intersection）、差（Difference）等。

（2）更新操作：增加（Insert）、删除（Delete）、修改（Update）等。

（3）控制操作：完整性控制、安全性控制、并发控制等。

查询操作是其中最主要的部分。

2. 关系操作的特点

（1）关系操作采用一次一集合的方式。

关系操作采用集合操作方式，即操作的对象和结果都是集合。这种操作方式也称为一次一集合（set-at-a-time）的方式。非关系操作采用记录操作，也称一次一记录（record-at-a-time）。

（2）关系语言是高度非过程化的语言。

关系语言是一种高度非过程化的语言，用户使用关系语言时，只需要指出做什么，而不需要指出怎么做，数据存取路径的选择、数据操作方法的选择和优化都由 DBMS 自动完成。

（3）关系语言操作一体化。

关系语言具有数据定义、查询、更新和控制一体化的特点。关系语言既可以作为宿主语言嵌入高级语言中，又可以作为独立语言交互使用。关系操作这一特点使得关系数据库语言学习容易，使用方便。

3. 关系操作语言的种类

关系操作语言可以分为三类，如图 3-2 所示。

（1）关系代数语言：用对关系的运算来表达查询要求的语言。

（2）关系演算语言：用谓词来表达查询要求的方式。关系演算又可按谓词变元的基本对

象是元组变量还是域变量分为元组关系演算和域关系演算。关系代数、元组关系演算和域关系演算三种语言在表达能力上是完全等价的。

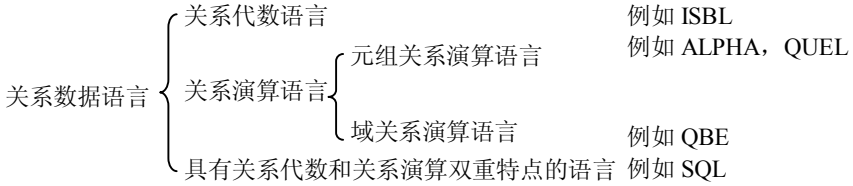


图 3-2　关系操作语言分类

（3）结构化查询语言（Structured Query Language，SQL）：不仅具有丰富的查询功能，而且具有数据定义和数据控制功能，是集查询（Query）、数据定义语言（DDL）、数据管理语言（DML）和数据控制语言（DCL）于一体的关系数据语言。它充分体现了关系数据语言的特点和优点，是关系数据库的标准语言和主流语言。

3.1.3　关系完整性

关系模型的完整性规则是对关系的某种约束条件。关系模型允许定义三类完整性约束：实体完整性、参照完整性和用户定义的完整性。其中实体完整性和参照完整性是关系模型必须满足的完整性约束条件，应该由关系系统自动支持。用户定义的完整性是应用领域需要遵循的约束条件，体现了具体应用领域中的语义约束。

1. **实体完整性**（Entity Integrity）

实体完整性规则：若属性 A 是基本关系 R 的主属性，则属性 A 不能取空值。

实体完整性规则规定基本关系的所有主属性都不能取空值，而不仅是主键整体不能取空值。例如学生选课关系“选修（学号，课程号，成绩）”中，“学号、课程号”为主键，则“学号”和“课程号”两个属性都不能取空值。

对于实体完整性规则说明如下：

（1）实体完整性规则是针对基本关系而言的。一个基本表通常对应现实世界的一个实体集。例如学生关系对应于学生的集合。

（2）现实世界中的实体是可区分的，即它们具有某种唯一性标识。相应地，关系模型中以主键作为唯一性标识。

（3）主键中的属性即主属性不能取空值。所谓空值就是“不知道”或“无意义”的值。如果主属性取空值，就说明存在某个不可标识的实体，即存在不可区分的实体，这与第（2）点矛盾。因此实体完整性能保证实体的唯一性和可区分性。

2. **参照完整性**（Referential Integrity）

- 外键和参照关系

关系模型中实体及实体间的联系都是用关系来描述的。这样就自然存在着关系与关系间的引用联系。先举个例子。

例 3.1　教师、课程、教师与课程之间的多对多联系可以用如下三个关系表示：

教师（教师号，姓名，性别，职称，系主任）

课程（课程号，课程名，学分，学时）

授课（教师号，课程号，时间）

这三个关系之间存在着属性的引用，即授课关系引用了教师关系的主键“教师号”和课程关系的主键“课程号”。授课关系中的“教师号”的值必须是确实存在的教师的教师号，即教师关系中有该教师的记录；授课关系中的“课程号”的值也必须是确实存在的课程的课程号，即课程关系中有该课程的记录。换句话说，授课关系中某些属性的取值需要参照其他关系的属性取值。不仅两个或两个以上的关系间可以存在引用关系，同一关系内部属性间也可能存在引用关系。在关系教师（教师号，姓名，性别，职称，系主任）中，“教师号”属性是主键，“系主任”属性表示该教师所在系的系主任的教师号，它引用了本关系中“教师号”属性，即“系主任”必须是确实存在的教师的教师号。

定义 2.4 设 F 是基本关系 R 的一个或一组属性，但不是关系 R 的键。如果 F 与基本关系 S 的主键 Ks 相对应，则称 F 是基本关系 R 的外键（Foreign Key），并称基本关系 R 为参照关系（Referencing Relation），基本关系 S 为被参照关系（Referenced Relation）或目标关系（Target Relation）。关系 R 和 S 不一定是不同的关系。

显然，目标关系 S 的主键 Ks 和参照关系的外键 F 必须定义在同一个（或一组）域上，但两者不一定同名。

在例 3.1 中，授课关系里教师号和课程号合在一起为主键，单独的教师号或课程号仅仅是主属性而不是关系的主键。由于教师号在教师关系中是主键，课程号在课程关系中是主键，因此，教师号和课程号是授课关系的外键，而授课关系为参照关系，教师关系和课程关系为被参照关系。授课表中教师号、课程号要分别参照教师表和课程表中教师号、课程号的值。在教师关系中，系主任不是主键，但它对应的教师号是主键，因此，系主任是教师关系的外键。教师表既是参照关系也是被参照关系。在表示关系的时候通常在主键下划横线，在外键下划波浪线。

例 3.2 在“职工管理数据库”中有“职工”和“部门”两个关系，其关系模式如下：

职工 (职工号，姓名，性别，职称，部门号)

部门 (部门号，名称，领导人号)

在职工表中，部门号不是主键，但部门表中部门号是主键，则职工表中的部门号为外键。对于职工表来说部门表为被参照表，职工表的“部门号”的值要参照部门表中“部门号”的取值，如图 3-3 所示。同理，在部门表中领导人号（实际上是领导人的职工号）不是主键，而在职工表中职工号为主键，则部门表中的领导人号为外键，职工表为被参照表。

- 参照完整性规则

参照完整性规则：若属性（或属性组）F 是基本关系 R 的外键，它与基本关系 S 的主键 Ks 相对应（基本关系 R 和 S 可能是相同的关系），则对于 R 中每个元组在 F 上的值必须为：

（1）或者取空值（F 的每个属性值均为空值），如图 3-3 中的“李东”部门号为空。

（2）或者等于 S 中某个元组的主键值，如图 3-3 中的“赵强”部门号为部门表中存在的值。

参照完整性规则就是定义外键与主键之间的引用规则。

如图 3-3 所示，职工表的李东的“部门号”只能取两类值：空值，表示尚未给该职工分配

部门；非空值，该值必须是部门关系中某个元组的“部门号”值。一个职工不可能分配到一个不存在的部门中，即被参照关系（部门）中一定存在一个元组，它的主键值等于该参照关系“职工”中的外键值。

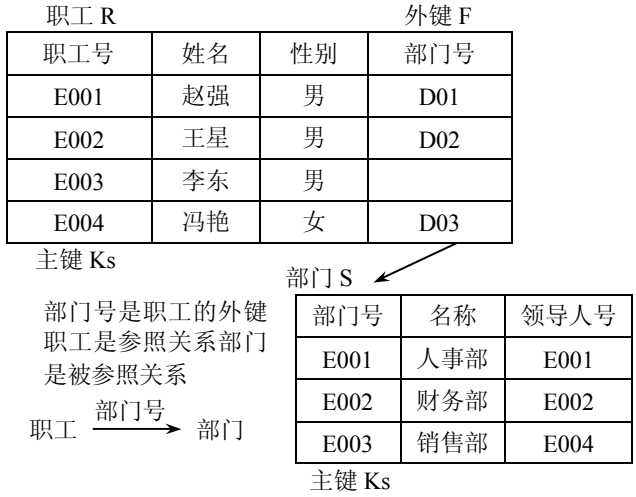


图 3-3　职工、部门参照关系

3. **用户定义的完整性**（User Defined Integrity）

实体完整性和参照完整性适用于任何关系数据库系统。此外，不同的关系数据库系统往往还根据应用环境的不同提供一些特殊的约束条件，用户定义的完整性就是针对某一具体应用的关系数据库的约束条件，它反映某一具体应用所涉及的数据必须满足的语义要求。关系模型应提供定义和检验这类完整性的机制，以便用统一的、系统的方法处理它们，而不是由应用程序承担这一功能。

关系数据库系统一般包括如下几种用户定义的完整性约束：空值（是否允许为空）、唯一性、取值范围、缺省值、属性间函数依赖关系。

例如，职工年龄不能大于 60 岁，职工性别只能是男或女等，都是针对具体关系提出的完整性条件。

3.2　关系代数

关系代数是一种抽象的查询语言，是用对关系的运算来表达关系操作。它是关系数据操纵语言的一种传统表达方式，

关系代数的运算对象是关系，运算结果亦为关系。关系代数用到的运算符包括四类：集合运算符、专门的关系运算符、算术比较符和逻辑运算符，如图 3-4 所示。

关系代数的运算分为两类：集合运算和专门的关系运算。集合运算将关系看成元组的集合，其运算是从关系的“水平”方向即行的角度来进行的。而专门的关系运算不仅涉及行，而且涉及列。比较运算符和逻辑运算符则是用来辅助专门的关系运算符进行操作的。

运算符		含义	运算符		含义
集号运算符	∪	并	比较运算符	＞	大于
	－	差		≥	大于等于
	∩	交		＜	小于
专门的关系运算符	×	广义笛卡尔积		≤	小于等于
	σ	投影		＝	等于
	∏	选择		≠	不等于
	⋈	连接	逻辑运算符	¬	非
	÷	除		∧	与
				∨	或

图 3-4　关系代数运算符

3.2.1　集合运算

集合运算是二目运算，包括并、差、交、广义笛卡尔积四种运算。

1. **并**（Union）

定义 2.5　设关系 R 和关系 S 具有相同的目 n（即两个关系都有 n 个属性），且相应的属性来自同一个域，则关系 R 与关系 S 的并由属于 R 或属于 S 的元组组成。其结果关系仍为 n 目关系。记作：

$$R \cup S = \{t \mid t \in R \vee t \in S\}$$

从定义可以看出 R 和 S 的并包含了 R 中的所有元组和 S 中所有元组，由于集合中没有重复值，所以 R 和 S 都有的元组在 R 和 S 的并中只出现一次。即 R 和 S 的并是 R 中的所有元组加上 S 中关系 R 没有的元组，如图 3-5（a）所示。

2. **差**（Difference）

定义 2.6　设关系 R 和关系 S 具有相同的目 n，且相应的属性来自同一个域，则关系 R 与关系 S 的差由属于 R 而不属于 S 的所有元组组成。其结果关系仍为 n 目关系。记作：

$$R - S = \{t \mid t \in R \wedge t \notin S\}$$

从定义可以看出 R 和 S 的差由属于 R 但不属于 S 的元组构成，即 R 和 S 的差的结果是 R 中的所有元组减去 R 和 S 都有的元组，所以 R-S 是 R 的一个子集。如图 3-5（b）所示。

3. **交**（Intersection）

定义 2.7　设关系 R 和关系 S 具有相同的目 n，且相应的属性取自同一个域，则关系 R 与关系 S 的交由既属于 R 又属于 S 的元组组成。其结果关系仍为 n 目关系。记作：

$$R \cap S = \{t \mid t \in R \wedge t \in S\}$$

从定义可以看出 R 和 S 的交由属于 R 并且属于 S 的元组构成，即 R 和 S 的交的结果是 R 和 S 都有的元组。如图 3-5（c）所示。关系的交操作可以用关系的差操作来表示，即 R∩S=R-(R-S)。

从交、并、差的定义可以看出，关系 R 和关系 S 两个关系要进行交、并、差的运算必须

满足两个条件：

- 关系 R 和关系 S 具有相同的目
- 相应的属性取自同一个域

即关系 R 和关系 S 必须具有相同的关系模式。交、并、差运算的结果与原关系的结构一致，改变的仅是关系中的数据（元组）。

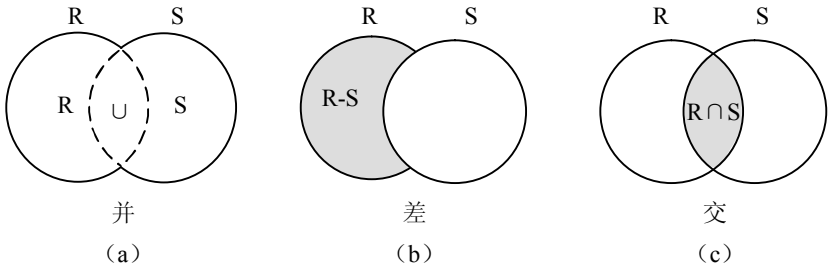


图 3-5　并、差、交操作示意

4. **广义笛卡尔积**（Extended Cartesian Product）

定义 2.8　两个分别为 n 目和 m 目的关系 R 和 S 的广义笛卡尔积是一个(n+m)列的元组的集合。元组的前 n 列是关系 R 的一个元组，后 m 列是关系 S 的一个元组。若 R 有 k_1 个元组，S 有 k_2 个元组，则关系 R 和关系 S 的广义笛卡尔积有 $k_1 \times k_2$ 个元组。记作：

$$R \times S = \{t_r t_s \mid t_r \in R \wedge t_s \in S\}$$

由于关系的运算对象和结果都是关系，所以在计算运算结果时应从结果的关系型和结构的元组两方面着手考虑。比如广义笛卡尔积结果的关系型是由前 n 列是 R 的的列（属性）后 m 列是关系 S 的列构成的新关系模式。广义笛卡尔积结果的元组是关系 R 中的所有元组与关系 S 中的所有元组进行元组横向合并所得到的元组的集合。

例如，给出 R 和 S 的数据，他们之间的并、交、差和笛卡尔积运算结果如图 3-6 所示。

R

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3

S

A	B	C
a1	b1	c1
a4	b4	c4

R∪S

A	B	C
a1	b1	c1
a1	b2	c2
a3	b3	c3
a4	b4	c4

R-S

A	B	C
a2	b2	c2
a3	b3	c3

R∩S

A	B	C
a1	b1	c1

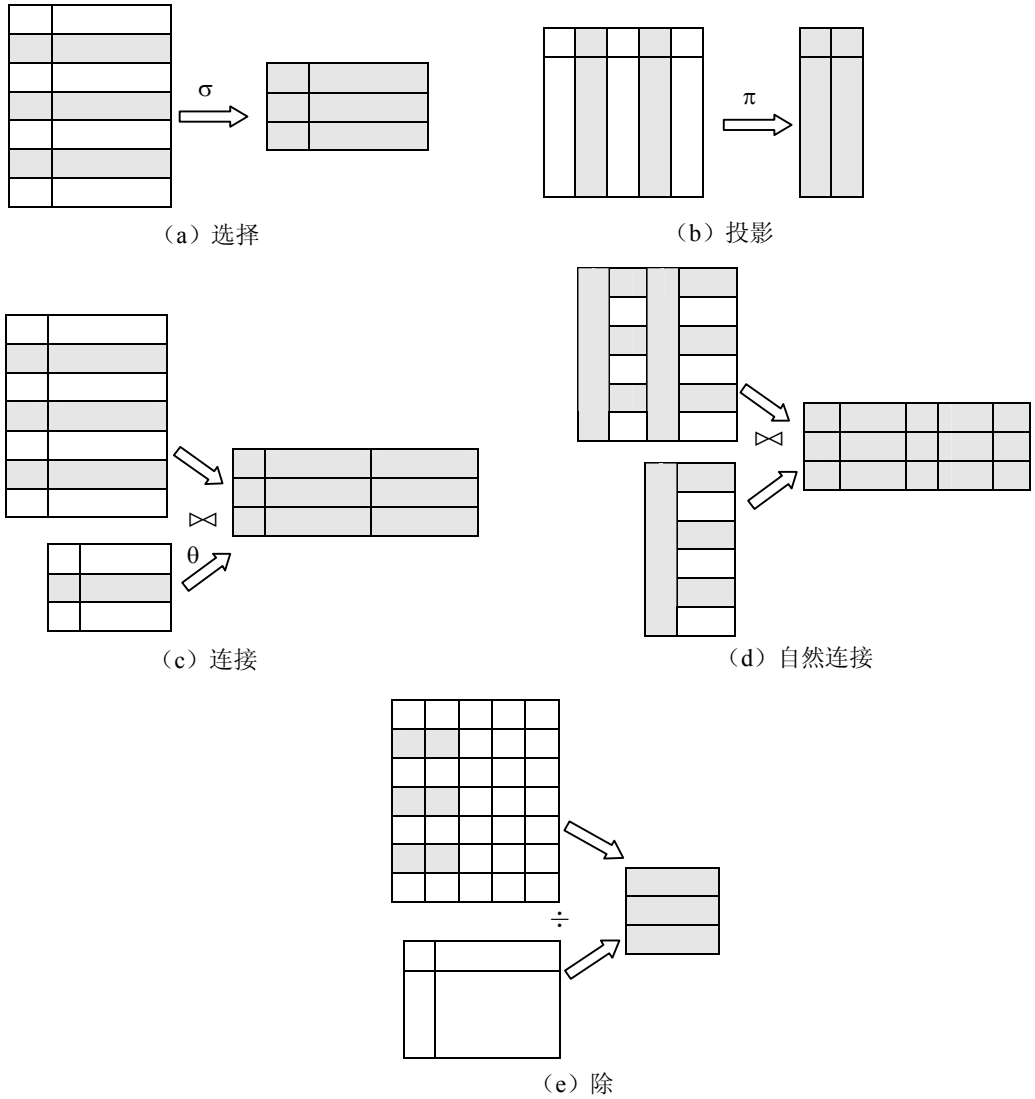
R×S

R.A	R.B	R.C	S.A	S.B	S.C
a1	b1	c1	a1	b1	c1
a1	b1	c1	a4	b4	c4
a2	b2	c2	a1	b1	c1
a2	b2	c2	a4	b4	c4
a3	b3	c3	a1	b1	c1
a3	b3	c3	a4	b4	c4

图 3-6　集合运算实例

3.2.2 专门的关系运算

集合运算只是从行的角度进行操作，而为了更灵活地实现数据库的多样查询操作，则需引入专门的关系运算。专门的关系运算包括选择、投影、连接、除等，如图 3-7 所示。



（a）选择　（b）投影

（c）连接　（d）自然连接

（e）除

图 3-7　专门关系运算示意图

为了叙述上的方便，先引入几个记号。

1. **常用记号**

- 分量

设关系模式为 $R(A_1, A_2, \ldots, A_n)$，它的一个关系为 R，$t \in R$ 表示 t 是 R 的一个元组。$t[A_i]$ 则表示元组 t 中相应于属性 A_i 的一个分量，分量是属性在元组上的取值。

- 域列

若 $A=\{A_{i1},A_{i2},\cdots,A_{ik}\}$，其中 $A_{i1},A_{i2},\cdots,A_{ik}$ 是 $A_1,A_2,\cdots,A_n$ 中的一部分，则 A 称为属性列或域列。$t[A]=(t[A_{i1}],t[A_{i2}],\cdots,t[A_{ik}])$表示元组 t 在属性列 A 上诸分量的集合。$\overline{A}$ 则表示$\{A_1,A_2,\cdots,A_n\}$中去掉$\{A_{i1},A_{i2},\cdots,A_{ik}\}$后剩余的属性组，它称为域列非。

- 元组的连接

R 为 n 目关系，S 为 m 目关系。$t_r \in R, t_s \in S, t_r t_s$ 称为元组的连接（Concatenation）。它是一个（n+m）列的元组，前 n 个分量为 R 中的一个 n 元组，后 m 个分量为 S 中的一个 m 元组。

- 像集。

给定一个关系 R(X,Z)，X 和 Z 为属性组，当 t[X]=x 时，x 在 R 中的像集（Images Set）为：

$$Z_x = \{t[z] \mid t \in R, t[X] = x\}$$

它表示 x 在 R 中的像集为 R 中属性组 X 上值为 x 的诸元组在 Z 上分量的集合。我们首先分析 x 在 R 中的像集 Z_X 的关系模式：由属性组 Z 构成的一个新关系，Z_X 关系的元组是：R 中属性组 X 上值为 x 的所有元组在 Z 上分量。像集示意图如图 3-8 所示。

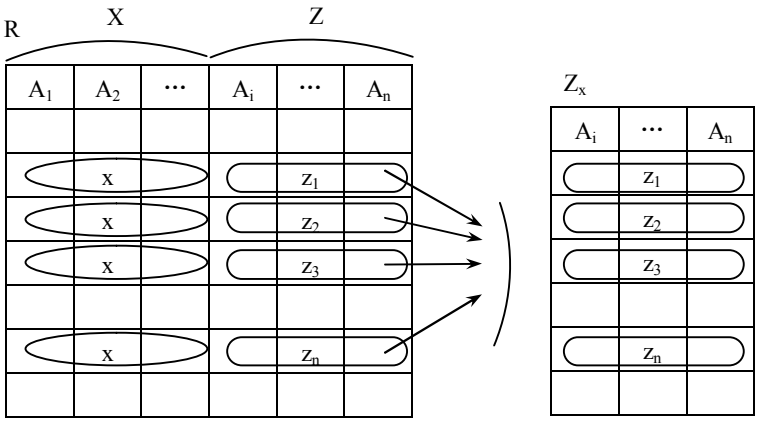


图 3-8　像集示意图

例如，如图 3-9 所示学生选课数据库中的选课关系中，设 X={Sno}，Z={Cno,Score}，令 X 的一个取值为 x'0762101'，则 x 在选课关系上的像集为 Z_X={(C01,70),(C02,80)}。

设有学生课程数据库，它包含学生、课程、选课和系四个关系，其关系模式为：

- 学生表　S(Sno,Sname,Sex,age,Dno)：分别表示学号、姓名、性别、年龄和所在系。
- 课程表　C(Cno,Cname,Cpno,CT)：分别表示课号、课名、先行课和学分。
- 选课表　SC(Sno,Cno,Score)：分别表示学号、课号和成绩。
- 系表　DEPT(Dno,Dname)：分别表示系号和系名。

每个表已有数据如图 3-9 所示。后面的关系运算我们都以此学生课程数据库的几个表为例。

2. **选择运算**（Selection）

定义：选择运算又称为限制（Restriction）。它是在关系 R 中选择满足给定条件的元组，记作：

$$\sigma_F(R) = \{t \mid t \in R \wedge F(t) = \text{'True'}\}$$

其中 F 表示选择条件，它是一个逻辑表达式，取逻辑值“真”或“假”。逻辑表达式 F 由逻辑运算符 ¬，∧，∨连接各算术表达式组成。算术表达式的基本形式为：

$$X_1\theta Y_1$$

其中θ表示比较运算符，它可以是>、≥、<、≤、=或≠，X_1、Y_1 为属性名，或为常量，或为简单函数；属性名也可以用它在关系中的序号来代替。

S

Sno	Sname	Sex	age	Dno
0762101	张三	男	19	D06
0762102	赵新	女	18	D06
0732201	周强	女	18	D03
0712102	张丰	男	19	D01

SC

Sno	Cno	Score
0762101	C01	70
0762101	C02	80
0762102	C01	90
0762102	C02	80
0762102	C03	70
0732201	C01	90
0732201	C02	80
0732201	C03	85
0712102	C01	85

C

Cno	Cname	Cpno	CT
C01	C++		4
C02	数据库	C01	3
C03	单片机	C01	4

DEPT

Dno	Dname
D01	管理系
D03	计算机系
D06	通信系

图 3-9　学生课程数据库

选择运算实际上是从关系 R 中选取使逻辑表达式 F 为真的元组。这是从行的角度进行的运算。关系的选择操作对应关系记录的选取操作，是关系查询的基本操作。选择运算的示意如图 3-7（a）所示，选择运算的结果关系的模式与被操作关系一致，结果关系的元组是被操作关系中满足条件 F 的那些元组。

例 3.3　用关系代数表示在学生课程数据库中查询年龄小于 19 岁的学生的操作。

$$\sigma_{age<19}(S) \text{ 或 } \sigma_{4<19}(S)$$

运算结果如图 3-10（a）所示。

例 3.4　用关系代数表示在学生课程数据库中查询全体男学生的操作。

$$\sigma_{Sex='男'}(S)$$

运算结果如图 3-10（b）所示。

3. **投影运算**（Projection）

关系 R 上的投影是从 R 中选择出若干属性列组成新的关系。记作：

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

其中 A 为 R 中的属性列。投影操作是从列的角度进行的运算。投影操作不仅取消了关系中的某些列，而且还可以取消某些元组，因为当取消某些属性后，就可能出现重复的元组，关系操作将自动取消这些相同的元组。投影运算的示意如图 3-7（b）所示，投影运算结果关系

只包含 A 这些属性列，包含元组 R 中所有元组在 A 这些属性列上的分量，如果有重复会自动去除。

Sno	Sname	Sex	age	Dno
0762102	赵新	女	18	D06
0732201	周强	女	18	D03

（a）例 3.3 运算结果

Sno	Sname	Sex	age	dept
0762101	张三	男	19	D06
0712102	张丰	男	19	D01

（b）例 3.4 运算结果

Sno	Sname
0762101	张三
0712102	张丰

（c）例 3.6 运算结果

Sno	Sname
0762101	张三
0762102	赵新
0732201	周强
0712102	张丰

（d）例 3.5 运算结果

图 3-10　关系运算结果

例 3.5　用关系代数表示在学生课程数据库中查询所有学生的学号和姓名。

$$\pi_{Sno,Sname}(S) \text{ 或 } \pi_{1,2}(S)$$

运算结果如图 3-10（d）所示。

选择和投影组合使用，能定位到关系中最小单元——任一分量值，从而完成对单一关系的任意查询操作。

例 3.6　用关系代数表示在学生课程数据库中查询年龄小于 20 岁的全体男学生的学号和姓名。

$$\pi_{Sno,Sname}(\sigma_{age<20 \wedge sex='男'}(S))$$

运算结果如图 3-10（c）所示。

例 3.7　给定一个关系 R(X,Z)，X 和 Z 为属性组，用选择和投影运算表示 x 在 R 中的像集 Z_X。

$$Z_X = \pi_Z(\sigma_{X=x}(R))$$

4. **连接运算**（Join）

连接也称为θ连接。它是从两个关系的笛卡尔积中选取属性间满足一定条件的元组。记作：

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

其中 A 和 B 分别为 R 和 S 上度数相等且可比的属性组。θ是比较运算符。连接运算的示意如图 3-7（c）所示。连接运算从 R 和 S 的笛卡尔积 R×S 中选取（R 关系）在 A 属性组上的值与（S 关系）在 B 属性组上值满足比较关系 θ 的元组。为此：

$$R \underset{A\theta B}{\bowtie} S = \sigma_{A\theta B}(R \times S)$$

连接运算中有两种最为重要也最为常用的连接，一种是等值连接（equal-join），另一种是自然连接（Natural join）。θ 为“=”的连接运算称为等值连接。它是从关系 R 与 S 的广义笛卡尔积中选取 A，B 属性值相等的那些元组，即等值连接为：

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

$$R \underset{A=B}{\triangleright\!\triangleleft} S = \sigma_{A=B}(R \times S)$$

或

自然连接（Natural join）是一种特殊的等值连接，它要求两个关系中进行比较的分量必须是相同的属性组，并且要在结果中把重复的属性去掉。自然连接的示意如图 3-7（d）所示，则自然连接可记作：

$$R \triangleright\!\triangleleft S = \{\widehat{t_r t_s} \mid t_r \in R \wedge t_s \in S \wedge t_r[A] = t_s[A]\}$$

或

$$R \triangleright\!\triangleleft S = \pi_{\bar{A}}(\sigma_{R.A=S.A}(R \times S))$$

一般的连接操作是从行的角度进行运算。但自然连接还需要取消重复列，所以是同时从行和列的角度进行运算。

关系的连接运算，实际上是在广义笛卡尔积的基础上再组合选择或投影操作复合而成的一种查询操作，在以后的查询中将广泛运用等值连接和自然连接。

例 3.8 以图 3-9 学生课程数据库中学生和系关系为例，学生与系之间的笛卡尔积、等值连接和自然连接的结果如图 3-11 所示。

S×DEPT

Sno	Sname	Sex	age	S.Dno	DEPT.Dno	Dname
0762101	张三	男	19	D06	D06	通信系
0762101	张三	男	19	D06	D03	计算机系
0762101	张三	男	19	D06	D01	管理系
0762102	赵新	女	18	D06	D06	通信系
0762102	赵新	女	18	D06	D03	计算机系
0762102	赵新	女	18	D06	D01	管理系
0732201	周强	女	18	D03	D03	计算机系
0732201	周强	女	18	D03	D01	管理系
0732201	周强	女	18	D03	D06	通信系
0712102	张丰	男	19	D01	D01	管理系
0712102	张丰	男	19	D01	D03	计算机系
0712102	张丰	男	19	D01	D01	管理系

$S \underset{S.Dno=DEPT.Dno}{\bowtie} DEPT$

Sno	Sname	Sex	age	S.Dno	DEPT.Dno	Dname
0762101	张三	男	19	D06	D06	通信系
0762102	赵新	女	18	D06	D06	通信系
0732201	周强	女	18	D03	D03	计算机系
0712102	张丰	男	19	D01	D01	管理系

S ⋈ DEPT

Sno	Sname	Sex	age	S.Dno	Dname
0762101	张三	男	19	D06	通信系
0762102	赵新	女	18	D06	通信系
0732201	周强	女	18	D03	计算机系
0712102	张丰	男	19	D01	管理系

图 3-11 关系间笛卡尔积、等值连接和自然连接运算结果比较

5. **除**（Division）

定义：给定关系 R(X,Y)和 S(Y,Z)，其中 X, Y, Z 为属性组。R 中的 Y 与 S 中的 Y 可以有不同的属性名，但必须出自相同的域集。R 与 S 的除运算得到一个新的关系 P(X)，P 是 R 中满足下列条件的元组在 X 属性列上的投影：元组在 X 上分量值 x 的像集 Y_x 包含 S 在 Y 上投影的集合。记作：

$$R \div S = \{t_r[X] \mid t_r \in R \wedge \pi_y(S) \subseteq Y_x\}$$

其中 Y_x 为 x 在 R 中的像集，$x = t_r[X]$。

除运算的示意如图 3-7（e）所示，除运算同时从行和列的角度进行运算，在进行除运算时，将被除关系 R 的属性分成两部分：与除关系相同的部分 Y 和剩余的部分 X。在被除关系中按 X 值分组，即相同 X 值的元组分为一组。除运算是求包括除关系中全部 Y 值的组，这些组的 X 将作为除运算结果的元组。除运算适合于包含“对于所有的/全部的”语句的查询操作。

关系除法求解步骤如下：

（1）将被除关系的属性分为像集属性（Y）和结果属性（X）两部分：与被除关系相同的属性属于像集属性，不同的属性属于结果属性。

（2）在除关系中，对被除关系相同的属性（像集属性）进行投影，得到除目标集。

（3）将被除关系按结果属性值进行分组。

（4）逐一考察每个组，如果它的像集属性值中包含除目标属性值，则对应的结果属性值应属于该除法结果集。

关系除操作是由关系代数基本操作复合而成的查询操作，可以用其他基本操作表示为：

$$R \div S = \pi_X(R) - \pi_X(\pi_X(R) \times \pi_Y(S) - R)$$

例 3.9　设关系 R，S，计算 R÷S 的结果。如图 3-12 所示。

在关系 R 中，像集属性为“Cno”，结果属性为“Sno”；“Sno”取四个值{0762101,0762102,0732201,0712102}，其中：0762101 的像集为{C01，C02}；0762102 的像集为{C01，C02，C03}；0732201 的像集为{C01，C02，C03}；0712102 的像集为{C01}，在 S 中除目标集为{C01，C02，C03}，显然只有 0762102，0732201 的像集包含 S 在“Cno”属性组上的投影，所以 R÷S={0762102，0732201 }。

R（SC）

Sno	Cno
0762101	C01
0762101	C02
0762102	C01
0762102	C02
0762102	C03
0732201	C01
0732201	C02
0732201	C03
0712102	C01

S（C）

Cno	Crame	Cpno	CT
C01	C++		4
C02	数据库	C01	3
C03	单片机	C01	4

R÷S

Sno
0762102
0732201

图 3-12　除运算举例

6. 用关系代数表示检索的例子

在关系代数中，关系代数运算经有限次复合后形成的式子称为关系表达式。对于关系数据库中数据的查询可以写出一个关系代数表达式。但关系表达式只是数学表达式不是实际的查询操作或语句。

下列举例不做特别说明都是针对图 3-9 学生课程数据库中的几个关系进行检索。

例 3.10 检索选修了课程号为“C02”的学生的学号和成绩。

$$\pi_{Sno,score}(\sigma_{Cno='C02'}(SC))$$

解题说明：本题只涉及关系 SC，需要选择和投影两种操作；但需要选择和投影时，应先进行选择操作。请读者思考下，为什么？

例 3.11 检索选修了课程号为“C02”的学生的学号和姓名。

$$\pi_{Sno,Sname}(\sigma_{Cno='C02'}(S \triangleright\!\triangleleft SC))$$

解题说明：由于学生的选课信息在关系 SC（选课）中而学生的姓名（Sname）在 S（学生）中，因此本题涉及两个关系 SC 和 S。

例 3.12 检索没有选修课程号为“C02”的学生的学号。

$$\pi_{Sno}(S) - \pi_{Sno}(\sigma_{Cno='C02'}(SC))$$

解题说明：特别注意，该题不能写成：$\pi_{Sno}(\sigma_{Cno\neq'C02'}(SC))$，想想为什么？

例 3.13 检索至少选修了课程号为“C02”和“C03”两门课程的学生的学号。

$$\pi_{Sno}(\sigma_{Cno='C02'}(SC)) \cap \pi_{Sno}(\sigma_{Cno='C03'}(SC))$$

解题说明：本题先分别求出选修了课程号为“C02”和“C03”的学生的学号，然后进行交运算，即得到选修了课程号为 C02 和 C03 两门课的学生学号，由于同一元组中课程号不可能既是“C02”又同时是“C03”所以本题不能写成：

$$\pi_{Sno}(\sigma_{Cno='C02' \wedge Cno='C03'}(SC))$$

例 3.14 检索选修了课程号为“C02”或“C03”的学生的学号。

$$\pi_{Sno}(\sigma_{Cno='C02' \vee Cno='C03'}(SC))$$

$$\text{或 } \pi_{Sno}(\sigma_{Cno='C02'}(SC)) \cup \pi_{Sno}(\sigma_{Cno='C03'}(SC))$$

解题说明：本题可以用选择条件中的或运算，也可以用并运算表示。

例 3.15 检索选修了全部课程学生的学号。

$$\pi_{Sno,Cno}(SC) \div C$$

解题说明：本题需先对选课关系进行投影操作，投影操作结果再除课程关系。

例 3.16 检索所学课程包含 0712102 所学课程的学生的学号。

$$\pi_{Sno,Cno}(SC) \div \pi_{Cno}(\sigma_{Sno='0712102'}(SC))$$

解题说明：本题是检索至少选修了 0712102 学生所选修过的所有课程的学生的学号，所以要先检索 0712102 所学的全部课程并投影到 Cno，然后用除运算得到检索结果。

3.3 疑难问题解答

问：怎样进行关系代数操作的表达？

答：

- 在进行操作表达前，应根据查询条件与要查询的信息等，首先确定查询涉及哪些表。
- 操作表达中要有动态操作变化的理念，即要有一步步动态操作关系，生成新关系，再操作新关系，如此反复，直到查询到所需信息的操作思路与方法。
- 关系代数的表达不是唯一的。

问：怎样理解除操作运算？

答：除操作同时从行和列的角度进行运算，在进行除运算时，将被除关系 R 的属性分成两部分：与除关系相同的部分 Y 和剩余的部分 X。在被除关系中按 X 值分组，即相同 X 值的元组分为一组。除法运算是求包括除关系中全部 Y 值的组，这些组的 X 将作为除结果的元组。除操作适合于包含“对于所有的/全部的”语句的查询操作。

关系除法求解步骤如下：

（1）将被除关系的属性分为像集属性（Y）和结果属性（X）两部分：与被除关系相同的属性属于像集属性，不同的属性属于结果属性。

（2）在除关系中，对被除关系相同的属性（像集属性）进行投影，得到除目标集。

（3）将被除关系按结果属性值进行分组。

（4）逐一考察每个组，如果它的像集属性值中包含除目标属性值，则对应的结果属性值应属于该除法结果集。

除运算是本章一个难点，读者可按上述步骤多做习题以加深对其的理解。

问：等值连接与自然连接的区别是什么？

答：等值连接是从关系 R 和 S 的广义笛卡尔积中选取 A 和 B 属性值相等的那些元组。自然连接是一种特殊的等值连接，它要求两个关系中进行比较的分量必须是相同的属性组，并且在结果中把重复的属性列去掉。

3.4　本章小结

本章系统讲解了关系数据库的重要概念，包括关系模型的数据结构、关系的完整性以及关系操作。本章还详细介绍了关系代数的基本概念，关系代数的各种运算操作。读者通过对本章知识的学习，应当：

- 掌握关系的数据结构、关系模式的概念，理解关系的特点。
- 掌握实体完整性、参照完整性和用户定义的完整性。
- 理解关系代数的基本概念、掌握关系运算的各种操作。
- 掌握用关系操作进行数据查询的方法。

3.5　实战检验

理论巩固

1．你是如何理解关系数据库中表和关系这两个概念的？它们之间有什么区别和联系？

2．试述关系模型的完整性规则。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？

3．定义并解释下列术语，说明它们之间的联系、区别以及各自的作用。

（1）主键、候选键、外键。

（2）笛卡尔积、关系、元组、属性、域。

（3）关系、关系模式、关系数据库。

4．对于学生选课关系，其关系模式为：

学生（学号，姓名，年龄，所在系）；

课程（课程名，课程号，先行课）；

选课（学号，课程号 成绩）。

用关系代数完成如下查询。

（1）检索选修了课程 C2 或 C4 的学生学号。

（2）检索选修了课程 C2 和 C4 的学生学号。

（3）检索不学 C2 课程的学生姓名与年龄

（4）检索学过数据库课程的学生的姓名和学号。

（5）检索年龄大于 23 岁的男学生的学号与姓名。

（6）检索学过数据库和数据结构的学生姓名和学号。

（7）检索没学过数据库课程的学生学号。

（8）检索学过数据库的先行课的学生学号。

（9）检索学习全部课程的学生姓名。

5．设有一个 SPJ 数据库，包括 S、P、J、SPJ 四个关系模式：

S（SNO，SNAME，STATUS，CITY）；

P（PNO，PNAME，COLOR，WEIGHT）；

J（JNO，JNANE，CITY）；

SPJ（SNO，PNO，JNO，QTY）。

其中：供应商表 S 由供应商代码（SNO）、供应商姓名（SNAME）、供应商状态（STATUS）、供应商所在城市（CITY）组成；零件表 P 由零件代码（PNO）、零件名（PNAME）、颜色（COLOR）、重量（WEIGHT）组成；工程项目表 J 由工程项目代码（JNO）、工程项目名（JNAME）、工程项目所在城市（CITY）组成；供应情况表 SPJ 由供应商代码（SNO）、零件代码（PNO）、工程项目代码（JNO）、供应数量（QTY）组成，表示某供应商供应某种零件给某工程项目的数量为 QTY。

试用关系代数完成如下查询：

（1）求供应工程 J1 零件的供应商号码 SNO。

（2）求供应工程 J1 零件 P1 的供应商号码 SNO。

（3）求供应工程 J1 零件为红色的供应商号码 SNO。

（4）求没有使用天津供应商生产的红色零件的工程号。

（5）求至少用了供应商 S1 所供应的全部零件的工程号。

（6．设属性 A 是关系 R 的主属性，则属性 A 不能取空值（NULL），这是（　　）。

A．实体完整性规则　　B．参照完整性规则

C．用户定义完整性规则　　D．域完整性规则

7．下面对于关系的叙述中，不正确的是（　　）。

A．关系中的每个属性是不可分解的　　B．在关系中元组的顺序是无关紧要的

C．任意的一个二维表都是一个关系　　D．每一个关系只有一种记录类型

8．设关系 R 和 S 的元组个数分别为 100 和 300，关系 T 是 R 与 S 的笛卡尔积，则 T 的元组个数是（　　）。

A．400　　B．10000　　C．30000　　D．90000

9．设关系 R 与关系 S 具有相同的目（或称度），且相对应的属性的值取自同一个域，则 R-（R-S）等于（　　）。

A．R∪S　　B．R∩S　　C．R×S　　D．R−S

上机实战

1．考察 SQL Server 2005 数据库管理系统中有哪几类表，它们有什么不同，各自有什么作用？

2．考察 SQL Server 2005 数据库管理系统中怎样支持用户定义的完整性。