第 3 章　关系数据库基础

【知识要点】

- 关系模型的概念
- 关系模型的数据结构
- 关系代数的基础概念及应用
- 关系完整性约束

【简介】

本章将引领读者深入了解关系数据库的基础知识，首先概述理论基础，包括关系模型概念及其数据结构，然后介绍关系代数、完整性约束等核心要素，为读者构建一个稳固的关系数据库基础框架。

【场景】

在构建与优化外卖点餐系统的过程中，关系数据库基础知识成为不可或缺的基石。关系数据库以其强大的数据存储、查询优化和完整性保障机制，为外卖点餐系统提供了坚实的数据支撑。本章将聚焦关系数据库基础知识在外卖点餐系统中的关键作用，通过深入分析关系模型及其数据结构、关系代数、完整性约束等核心要素，揭示它们如何协同提升系统的数据处理能力、确保数据的一致性和完整性，并最终促进外卖点餐系统的智能化与高效化运行。

3.1　理论基础

3.1.1　关系模型概念

关系模型的数学基础是集合理论，在集合代数中对“关系”给出了严格的数学定义，并支持对关系进行各种操作。

关系模型作为数据结构模型的一种，在今天的数据库系统中具有重要的作用。它只有单一的数据结构——关系。所谓关系就是通常所说的二维表，二维表和关系在关系数据库中是同义词。现实世界中的实体和实体之间的各种联系均通过关系表示，并以二维表的形式组织数据，可通过 SQL 对关系中的数据进行操作。

关系操作是指对二维表进行操作所使用的方法。常用的关系操作包括选择（Select）、投影（Project）、连接（Join）、除（Divide）、并（Union）、交（Intersection）、差（Difference）等查询（Query）操作，以及插入（Insert）、删除（Delete）、修改（Update）等更新操作。其中，查询操作是最主要的部分。

关系操作的特点是采用集合操作方式，即操作的对象和结果都是集合。

3.1.2　关系模型数据结构

1．域（Domain）

定义 1.1　域是一组具有相同数据类型的值的集合。例如，整数、正数、负数、字符串、{男，女}、{苹果、橘子、香蕉、菠萝、西瓜} 等，都可以是域。在关系模型中，要求每个属性都属于某种基本类型，并将与关系中每个属性相对应的特定基本类型称为域。

2．笛卡儿积（Cartesian Product）

定义 1.2　给定一组域 $D_1, D_2, \cdots, D_n$（这些域可以相同），笛卡儿积定义为

$$D_1 \times D_2 \times D_3 \times \cdots \times D_n = \{(d_1, d_2, d_3, \cdots, d_n) \mid d_i \in D_i,\ i = 1, 2, \cdots, n\}$$

其中，每个元素（$d_1, d_2, d_3, \cdots, d_n$）称为一个 n 元组，简称元组；每个 d_i 称为一个分量。

笛卡儿积可以表示为一个二维表，表中的每一行对应一个元组，每一列对应一个域。

3．关系（Relation）

定义 1.3　笛卡儿积 $D_1 \times D_2 \times D_3 \times \cdots \times D_n$ 的一个子集叫作定义在域 $D_1, D_2, \cdots, D_n$ 上的关系，用 $R(D_1, D_2, \cdots, D_n)$表示，R 是关系名，n 是关系的目或度（Degree）。关系中的每个元素称为一个元组，通常用 t 表示。

关系是笛卡儿积中有一定意义的有限子集，所以关系也可以表示为二维表。n 目关系有 n 个属性（下面将介绍）。当 n=1 时，称该关系为单元关系；当 n=2 时，称该关系为二元关系。

以食品信息表（见表 3-1）为例，介绍关系模型中的一些术语。

表 3-1　食品信息表

食 品 编 号	食 品 名 称	食 品 介 绍	食 品 价 格	所属商家编号
1001	牛肉汉堡	100% 纯牛肉	58.50	1
1002	鸡肉沙拉	烤鸡肉	48.50	2
1003	小龙虾	麻辣口味	88.50	3
…	…	…	…	…

（1）关系：一个关系对应一张二维表。

（2）元组：表中的一行称为一个元组。

（3）属性：表中的一列称为一个属性，其名称为属性名，在表中具有唯一性。

（4）域：属性的取值范围称为域。

（5）分量：元组中一个属性的值称为该属性的分量。

（6）主键：表中可以唯一地确定一个元组的某个属性或属性组（不含多余属性）。

（7）关系模式：对关系的结构化描述，即描述关系中包含哪些属性。

表示形式：关系名（属性 1，属性 2，…，属性 n）。

例如，表 3-1 中的关系模式可描述为：食品信息（食品编号，食品名称，食品介绍，食品价格，所属商家编号）。

（8）关系数据库：若干关系的集合。

4．基本关系的性质

基本关系具有以下 6 条性质。

（1）列是同质的，即每一列中的分量都是同一类型的数据，来自同一个域。

（2）不同的列可以来自相同的域，称其中每一列为一个属性，不同的属性必须有不同的属性名。

（3）列的顺序无关紧要，即列的位置可以任意交换。

（4）关系中任意两个元组不能完全相同。

（5）元组的顺序无关紧要，即元组的位置可以任意交换。

（6）分量必须是原子值，即每一分量都不可再分。

关系模型要求关系必须是规范化的，即关系模式必须满足一定的规范化条件。最基本的一条是，关系中所有分量都必须是一个不可再分的数据项。

3.2　关系代数

关系代数是一种抽象的查询语言，是施加于关系的一组集合运算。每种运算以一个或多个关系作为运算对象，生成一个新的关系作为运算结果，即运算对象和运算结果都是关系。

本节以外卖点餐系统为例，介绍关系代数的一些具体应用，主要涉及以下 3 个数据表。

食品信息表（food）：食品编号（foodId），食品名称（foodName），食品介绍（foodExplain），食品价格（foodPrice），商家编号（businessId），如表 3-2 所示。

商家信息表（business）：商家编号（businessId），商家名称（businessName），商家地址（businessAddress），商家介绍（businessExplain），起送费（startPrice），配送费（deliveryPrice），如表 3-3 所示。

订单信息表（orders）：订单编号（orderId），订购日期（orderDate），订单总价（orderTotal），订单状态（orderState），用户编号（userId），送货地址编号（daId），商家编号（businessId），如表 3-4 所示。

表 3-2　食品信息表（food）

foodId	foodName	foodExplain	foodPrice	businessId
1001	牛肉汉堡	100%纯牛肉	58.50	1
1002	鸡肉沙拉	烤鸡肉	48.50	2
1003	小龙虾	麻辣口味	88.50	3

表 3-3　商家信息表（business）

businessId	businessName	businessAddress	businessExplain	startPrice	deliveryPrice
1	快乐快餐	蓝海路 1 号	快餐	20.00	5.00
2	健康轻食	蓝海路 2 号	轻食	30.00	6.00
3	辣味轩	蓝海路 3 号	川菜	25.00	5.50

表 3-4　订单信息表（orders）

orderId	orderDate	orderTotal	orderState	userId	daId	businessId
24331	2025.1.11	121.50	0	T122	131	1
24332	2025.1.12	150.50	0	T123	132	2
24333	2025.1.13	171.50	0	T124	134	3
24334	2025.1.14	124.50	1	T125	134	3

关系代数用到的运算符包括 4 类：集合运算符、关系运算符、比较运算符和逻辑运算符。其中，比较运算符和逻辑运算符通常与关系运算符一起使用。根据关系代数所支持的运算符的功能特点，运算主要分为传统的集合运算和专门的关系运算两类。

1．传统的集合运算

传统的集合运算包括并、差、交、笛卡儿积 4 种。这类运算是二目运算，将关系看成元组的集合，是在关系的水平方向上进行的，即以行的角度进行运算。

（1）并（Union）：设关系 R 和 S 是同一关系模式下的两个关系，则 R 和 S 的并是由属于 R 或属于 S 的所有元组组成的集合，即 $R \cup S = \{t \mid t \in R \vee t \in S\}$。

（2）差（Difference）：设关系 R 和 S 是同一关系模式下的两个关系，则 R 和 S 的差是由属于 R 但不属于 S 的元组组成的集合，即 $R - S = \{t \mid t \in R \wedge t \notin S\}$。

（3）交（Intersection）：设关系 R 和 S 是同一关系模式下的两个关系，则 R 和 S 的交是由既属于 R 又属于 S 的元组组成的集合，即 $R \cap S = \{t \mid t \in R \wedge t \in S\}$

（4）笛卡儿积（Cartesian Product）：设关系 R 和 S 的列数分别为 r 和 s，元组的个数分别为 m 和 n，则关系 R 和 S 的笛卡儿积是一个 $(r+s)$ 列的元组集合。每个元组的前 r 个分量是关系 R 的一个元组，后 s 个分量是关系 S 的一个元组，元组总数为 $m \times n$ 个，即 $R \times S = \{t \mid t = \langle t_r, t_s \rangle \wedge t_r \in R \wedge t_s \in S\}$。

【例 3-1】关系 R［见图 3-1（a）］和 S［见图 3-1（b）］具有相同的属性结构，传统的集合运算结果如下：图 3-1（c）为关系 R 与 S 的并；图 3-1（d）为关系 R 与 S 的差；图 3-1（e）为关系 R 与 S 的交；图 3-1（f）为关系 R 与 S 的笛卡儿积。

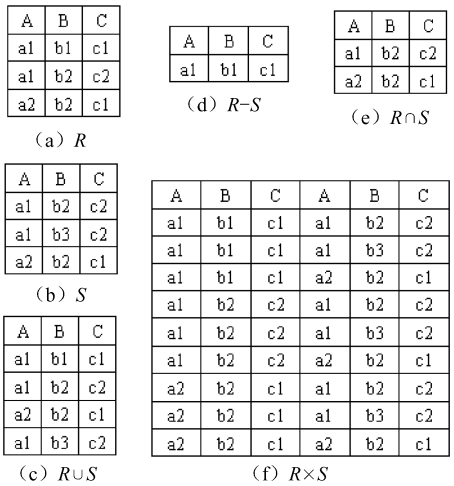

图 3-1　传统的集合运算举例

2．专门的关系运算

专门的关系运算包括选择、投影、连接、除等操作。这类运算不仅涉及二维表的行，还涉及二维表的列，主要用于实际的查询操作。

（1）选择（Select）：选择是一种一元关系操作，它作用在一个关系 R 上，按给定的选择条件 F 选出符合条件的元组，即 $\sigma_F(R)=\{t|t\in R\wedge F(t)=\text{true}\}$，简记为 $\sigma_F(R)$。其中，F 是一个逻辑表达式，取值为 true（真）或 false（假）。逻辑表达式 F 由逻辑运算符 ∧（与）、∨（或）、¬（非）连接算术表达式组成。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 是属性名、常量或简单函数。属性名也可以用它的序号来代替（如 1, 2, …）。

选择操作实际上是从关系 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。

【例 3-2】（1）查找食品名称为“牛肉汉堡”的食品信息。

（2）查找送货地址编号为“134”且已支付的订单信息。

解：（1）$\sigma_{\text{foodName}='\text{牛肉汉堡}'}(\text{food})$或 $\sigma_{2='\text{牛肉汉堡}'}(\text{food})$。

（2）$\sigma_{\text{daId}=134\wedge\text{orderState}=1}(\text{orders})$。

选择操作的结果如表 3-5 和表 3-6 所示。

表 3-5　【例 3-2】（1）的结果

foodId	foodName	foodExplain	foodPrice	businessId
1001	牛肉汉堡	100%纯牛肉	58.50	1

表 3-6　【例 3-2】（2）的结果

orderId	orderDate	orderTotal	orderState	userId	daId	businessId
24334	2025.1.14	124.50	1	T125	134	3

（2）投影（Project）：投影是一种一元关系操作，用于选取关系中的某些列。投影操作可以分两步。

① 选择指定的属性，形成一个可能含有重复行的表。

② 删除重复行，形成一个新的关系。

关系 R 在属性列 $A_{i_1},A_{i_2},\cdots,A_{i_m}$ 上的投影为 $\pi_{A_{i_1},A_{i_2},\cdots,A_{i_m}}(R)=\{r(A_{i_1},A_{i_2},\cdots,A_{i_m})|r\in R\}$，简记为 $\prod_{\langle\text{属性表}\rangle}$(<关系名>)。

【例 3-3】（1）从食品关系表中查找所有食品名称和价格。

（2）查询商家关系表中包含哪些商家，即查询该关系在“商家编号”属性上的投影。

解：（1）$\pi_{\text{foodName,foodPrice}}(\text{food})$或 $\pi_{2,4}(\text{food})$。

（2）$\pi_{\text{businessId}}(\text{business})$或 $\pi_1(\text{business})$。

投影操作的结果如表 3-7 和表 3-8 所示。

注意：投影操作过程中可能会出现内容完全相同的若干元组，但关系中不允许出现内容完全相同的元组，因此结果中只能保留其中一个元组。

表 3-7　【例 3-3】（1）的结果

foodName	foodPrice
牛肉汉堡	58.50
鸡肉沙拉	48.50
小龙虾	88.50

表 3-8　【例 3-3】（2）的结果

businessId
1
2
3

投影操作可与选择操作结合使用。例如，查找所有商家简介为“快餐”的商家名称，有

$$\pi_{\text{businessName}}(\sigma_{\text{businessExplain = '快餐'}}(\text{business}))$$

（3）连接（Join）：连接是一种二元关系操作，以符号 $\triangleright\triangleleft$ 表示。它的定义为

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

连接与笛卡儿积的区别在于，广义笛卡儿积包含两个关系所有元组的全部组合，而连接是从 R 和 S 的广义笛卡儿积 $R \times S$ 中，选择 R 在 A 属性组上的值与 S 在 B 属性组上的值满足比较条件 θ 的元组。

连接操作中有两种最为常用也最为重要的连接：一种是等值连接，另一种是自然连接。如果连接条件为两个关系中的两个属性间的相等比较，则称该连接为等值连接。若在等值连接的基础上，要求参与比较的属性组相同，并在结果中去除重复的属性列，则称该连接为自然连接。自然连接一般可以省略连接条件。

一般的连接操作是从行的角度进行的，而自然连接还需去除重复列，因此也涉及列的操作。自然连接中的连接属性通常是不同关系中语义相同的属性，一般是参照关系的外键和被参照关系的主键。

【例 3-4】取 business 中的部分属性列形成一个新关系 R，取 food 中的部分属性列形成一个新关系 S，如表 3-9 所示。计算下面的等值连接和自然连接。

（1）等值连接 $R \underset{R.\text{businessId}=S.\text{businessId}}{\triangleright\triangleleft} S$。

（2）自然连接 $R \triangleright\triangleleft S$。

表 3-9　关系 R 和 S

（a）R

businessId	businessName
1	快乐快餐
2	健康轻食
3	辣味轩

（b）S

businessId	foodName	foodPrice
1	牛肉汉堡	58.50
2	鸡肉沙拉	48.50
3	小龙虾	88.50

解：等值连接和自然连接的结果如表 3-10 和表 3-11 所示。

表 3-10　【例 3-4】（1）的结果

R.businessId	businessName	*S*.businessId	foodName	foodPrice
1	快乐快餐	1	牛肉汉堡	58.50
2	健康轻食	2	鸡肉沙拉	48.50
3	辣味轩	3	小龙虾	88.50

表 3-11　【例 3-4】（2）的结果

businessId	businessName	foodName	foodPrice
1	快乐快餐	牛肉汉堡	58.50
2	健康轻食	鸡肉沙拉	48.50
3	辣味轩	小龙虾	88.50

【例 3-5】查询用户在“辣味轩”所购食品的价格。

解：$\pi_{\text{businessId,foodPrice}}(\sigma_{\text{businessName='辣味轩'}}(\text{business}) \triangleright\triangleleft \text{food})$。

说明：该运算首先执行$\sigma_{\text{businessName='辣味轩'}}(\text{business})$，从商家关系表中选择商家名称为“辣味轩”的元组，然后通过“商家编号”属性将该元组与食品关系中的相应元组进行自然连接，最后投影出商家编号和食品价格。

（4）除（Divide）：在介绍除操作之前，需先了解像集（Image Set）的概念。

设存在关系$R(X,Z)$，其中X,Z是属性组。在R中，属性X取值为x的所有元组在Z上的分量集合，称为x在R中的像集，即

$$Z_x = \{t[Z] \mid t \in R \wedge t[X] = x\}$$

设存在关系$R(X,Y)$和$S(Y,Z)$，其中X,Y,Z为属性组，则R与S的除操作定义为，R中满足以下条件的元组在X属性上的投影——R中每个元组在X上取值x的像集Y_x包含S在Y上的投影集合，即

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

【例 3-6】关系R和S如表 3-12 中的（a）和（b）所示，$R \div S$的结果如表 3-12（c）所示。

表 3-12　关系 *R* 和 *S* 及除操作结果

（a）*R*

A	*B*	*C*
a1	b1	c2
a2	b3	c7
a3	b4	c6
a1	b2	c3
a4	b6	c6
a2	b2	c3
a1	b2	c1

（b）*S*

B	*C*	*D*
b1	c2	d1
b2	c1	d1
b2	c3	d2

（c）*R*÷*S*

A
a1

在关系 R 中，A 可以取 4 个值{a1,a2,a3,a4}。其中：

a1 的像集为{(b1,c2)，(b2,c3)，(b2,c1)}。

a2 的像集为{(b3,c7)，(b2,c3)}。

a3 的像集为{(b4,c6)}。

a4 的像集为{(b6,c6)}。

S 在(B,C)上的投影为{(b1,c2)，(b2,c1)，(b2,c3)}。

显然只有 a1 的像集包含 S 的投影，所以 $R \div S$={a1}。

【例 3-7】找出至少提供“牛肉汉堡”和“鸡肉沙拉”的商家。

构建一个临时关系 K，如表 3-13 所示。

表 3-13　临时关系 K

foodName
牛肉汉堡
鸡肉沙拉

解：$\pi_{\text{businessId,foodName}}(\text{food}) \div K$ 。

说明：首先执行 $\pi_{\text{businessId,foodName}}(\text{food})$，投影出食品信息表中所有的商家编号和食品名称，而 $\pi_{\text{businessId,foodName}}(\text{food}) \div K$ 则可选出至少含有“牛肉汉堡”和“鸡肉沙拉”的商家。

3.3　关系完整性约束

1．实体完整性——避免出现重复行

在数据正确输入的前提下，不出现重复行实际上反映了表中每一行的列值组合能够完整标识一条记录，而确保实体完整性的基本方法就是为表定义主键。实体完整性要求：基本关系中主键的所有属性都不能取空值。例如，在用户信息（用户编号，密码，用户名称，用户性别）中，“用户编号”为主键，则其值不能为空。

实体完整性约束针对基本关系，一个基本关系通常对应一个实体集。例如，用户关系对应用户集合。现实世界中的实体是可以区分的，具有唯一性标识，如用户的用户编号、订单的订单编号等。

2．参照完整性——表之间的数据一致性

如果订单和用户两个关系之间存在属性引用，即订单关系引用了用户关系的主键“用户编号”，那么订单表中的“用户编号”值必须是用户表中实际存在的用户编号，即用户关系中存在该用户的记录。这说明“用户编号”属性的取值需要参照用户关系中的主键值，这种约束就是参照完整性。

如果要删除被引用的对象，则必须先删除引用该对象的所有记录，或者将引用值设置为空（如果允许的话）。例如，在订单和用户关系中，删除某个用户元组前，应先删除引用该用户的订单记录。

3．用户定义完整性——数据的合理性和有效性

任何关系数据库系统都必须支持实体完整性和参照完整性。此外，根据具体应用环境的

需求，还需定义特定的约束条件，以反映数据必须满足的语义要求。例如，订单状态的取值只有 0（未支付）、1（已支付）；用户性别的取值只有 0（女）和 1（男）等。这类约束在关系模型中被称为用户定义完整性。关系模型应提供定义和验证这类约束的机制，以便系统用统一的方法管理，而不需要由应用程序来承担这一功能。

用户定义完整性约束还可以分为列级约束和行级约束。列级约束通常是定义属性取值的约束。例如，用户信息表中的“性别”属性取值只能为“男”或“女”。这类对属性值域的约束也称为域完整性规则（Domain Integrity Rule），即对关系中属性取值范围的限制，包括数据类型、精度、取值范围、是否允许空值等。行级约束则可能涉及一个以上的列。

3.4　章节结语

本章深入探讨了关系数据库的核心概念和原理，为理解数据库的构建和管理打下了坚实的基础。本章围绕关系模型的数学基础，逐步介绍了关系模型的数据结构、数据操作，以及关系完整性约束的重要性。通过学习关系代数和关系模型，读者不仅能够掌握关系数据库的基本操作，还能理解数据库模式的规范化方法，从而提升数据的一致性并减少冗余。

在本章的学习过程中，除掌握理论知识外，通过实践培养关系数据库应用技能也尤为重要。希望读者能将这些知识应用到实际的数据库设计和开发中，解决现实世界中的数据管理问题。随着技术的发展，数据库技术也在不断进步。期待读者保持学习热情，不断探索和掌握新的数据库技术。

在结束对本章的学习之前，建议读者回顾本章的重点内容，并尝试将其应用到实际项目中。下一章，我们将继续探索数据库原理与应用的基本知识，敬请期待。

作业及思考题

1．在关系代数中，5 种基本运算是（　　）。

A．$\cup -\times \sigma \pi$　　B．$\cup \cap - \sigma \pi$　　C．$\cup \cap \times \sigma \pi$　　D．$\cup \cap \wedge \sigma \pi$

2．关系代数运算是以（　　）为基础的运算。

A．关系运算　　B．谓词演算　　C．集合运算　　D．代数运算

3．在关系数据库中，实体与实体之间的联系用（　　）表示。

A．节点　　B．对象　　C．关系　　D．层次

4．以下关系代数运算中，（　　）不是基本运算。

A．选择　　B．广义笛卡儿积

C．投影　　D．连接

5．投影操作是指从关系中（　　）。

A．抽出特定记录　　B．抽出特定字段

C．建立相应的影像　　D．建立相应的图形

6．可以实现的专门关系运算包括（　　）。

A．排序、索引、统计　　B．选择、投影、连接

C．关联、更新、排序　　D．显示、打印、制表

7.（　　）操作将两个关系中的每一对元组组合成一个新元组。

8．简述选择操作是如何工作的。

9．描述投影操作的作用。

10．学生选课数据库的关系模式如下：

学生（学号，姓名，性别，年龄，所在系）；

课程（课程号，课程名，先行课）；

选课（学号，课程号，成绩）。

创建相应的二维表，用关系代数完成以下操作：

（1）求选修了课程号为“C2”的学生学号。

（2）求选修了课程号为“C2”的学生学号和姓名。

（3）求没有选修课程号为“C2”的学生学号。

（4）求既选修“C2”又选修“C3”的学生学号。

（5）求选修课程号为“C2”或“C3”的学生学号。

（6）求选修了全部课程的学生学号。

（7）学号为“98002”的学生所学过的所有课程可能也被其他学生选修，求这些学生的学号和姓名。