

Lecture 2: 关系模型

Database

Author: Forliage

Email: masterforliage@gmail.com

Date: June 9, 2025

College: 计算机科学与技术学院



浙江大学
ZHEJIANG UNIVERSITY

Abstract

本讲笔记系统介绍了关系模型的基本概念与运算，共分为五大部分：

1. 关系数据库的结构：阐明了关系（Relation）的定义及其由属性（Attribute）和元组（Tuple）构成的表格表示，介绍了关系模式（Schema）与关系实例（Instance）的区别，并详细说明了键（Key）与外键（Foreign Key）的概念及约束作用。
2. 基本关系代数操作：列举了六种原子运算——选择（Select）、投影（Project）、并集（Union）、差集（Set Difference）、笛卡尔积（Cartesian Product）和重命名（Rename），给出了每种运算的形式化定义及示例，帮助读者掌握如何在关系上进行基本查询。
3. 附加关系代数操作：在基本运算之上，进一步引入交集（Intersection）、自然连接（Natural Join）、 θ -连接（Theta-Join）、除法运算（Division）和赋值操作（Assignment），以简化常见复杂查询的表达，并说明了这些操作的逻辑含义与应用场景。
4. 扩展关系代数操作：介绍了广义投影（Generalized Projection），允许在投影列表中使用算术表达式，以及聚合函数（Aggregation Functions）和分组操作（Grouping），包括常见的 AVG、SUM、COUNT、MIN、MAX，并说明了其在统计分析查询中的用法。
5. 数据库的修改：说明了在关系代数框架下如何使用删除（DELETE）、插入（INSERT）和更新（UPDATE）三种操作修改数据库内容，分别给出了它们在关系代数中的表示方法。

通过本讲笔记的学习，读者将全面掌握关系模型的核心概念与操作，为后续深入理解查询处理、优化与事务管理奠定基础。

（该Abstract由ChatGPT-o4-mini-high生成）

Contents

1	关系数据库的结构	
1.1	基本结构	3
1.2	属性类型	3
1.3	关系的概念	3
1.4	关系模式	3
1.5	关系实例	3
1.6	关系的属性	3
1.7	键	4
1.8	外键	4
2	基本关系代数操作	
2.1	选择操作关系示例	4
2.2	选择操作形式化	5
2.3	投影操作示例	5
2.4	投影操作规范化	5
2.5	并集操作示例	5
2.6	并集操作形式化	6
2.7	差集运算示例	6
2.8	差集运算形式化	6
2.9	笛卡尔积操作示例	7
2.10	笛卡尔积操作形式化	7
2.11	重命名操作	7
3	附加关系代数操作	
3.1	集合交集操作形式化	8
3.2	自然连接操作的形式化	8
3.3	θ 连接操作的形式化	8
3.4	除法运算	8
3.5	赋值操作	9

4	扩展关系代数操作	
4.1	广义投影	9
4.2	聚合函数和操作	9
5	数据库的修改	
5.1	删除	10
5.2	插入	10
5.3	更新	10

1 关系数据库的结构

1.1 基本结构

形式上，给定集合 D_1, D_2, \dots, D_n ($D_i = \{a_{ij} | j=1, \dots, k\}$)，一个关系 r 是 (一组域 D_i 的笛卡尔积) $D_1 \times D_2 \times \dots \times D_n$ 的一个子集。

因此，一个关系是一个 n -元组的集合 $(a_{1j}, a_{2j}, \dots, a_{nj})$ ，其中每个 $a_{ij} \in D_i (i \in [1, n])$

1.2 属性类型

每个关系的属性都有一个名称。

每个属性的允许值集合称为该属性的域。

属性值（通常）要求是原子的，即不可分割的（第一范式）。例如，多值属性不是原子的；复合属性值不是原子的。

特殊值 `null` 是每个域的成员。`null` 值在许多操作的定义中会引起复杂性。

1.3 关系的概念

关系涉及两个概念：关系模式和关系实例。

关系模式描述了关系的结构；关系实例对应于某一时刻关系中数据的快照。

1.4 关系模式

假设 A_1, A_2, \dots, A_n 是属性。

正则表达： $R = (A_1, A_2, \dots, A_n)$ 是一个关系模式。

$r(R)$ 是关系模式 R 上的一个关系。

1.5 关系实例

关系的当前值（即关系实例）由一个表格指定。

一个 t 的元素 r 是一个元组，由表格中的一行表示。

设一个元组变量 t 为一个元组，则 $t[name]$ 表示 t 在 `name` 属性上的值。

1.6 关系的属性

元组的顺序无关紧要（即元组可以以任意方式存储）

关系中没有重复的元组

属性值是原子的。

1.7 键

令 $K \subset R$ 。

如果 K 的值足以唯一标识每个可能关系 $r(R)$ 的元组，那么 K 是 R 的超码。

如果 K 是最小超码，则 K 是候选码。

如果 K 是候选码且由用户明确定义，则 K 是主码（主码通常用下划线标记）。

1.8 外键

假设存在关系 r 和 s ： $r(\underline{A}, B, C), s(\underline{B}, D)$ ，我们可以说关系 r 中的属性 B 是引用 s 的外键，而 r 是参照关系， s 是被参照关系。

参照关系中外码的值必须在被参照关系中实际存在，或为 null。

主键和外键是综合约束。

2 基本关系代数操作

六个基本运算符： select, project, union, set difference, cartesian product, rename

操作符接受一个或两个关系作为输入，并返回一个新的关系作为结果。

2.1 选择操作关系示例

Relation

$$r =$$

A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

则

$$\sigma_{A=\beta \wedge D > 5}(r) =$$

A	B	C	D
β	β	23	10

请注意，在进行选择操作时，选择条件需要针对同一元组的属性值。

2.2 选择操作形式化

符号: $\sigma_p(r)$

p 被称为选择谓词

定义为: $\sigma_p(r) = \{t \in r \text{ and } p(t)\}$ 其中 p 是一个由 \wedge, \vee, \neg 连接的命题演算中的公式。

每个项是以下之一: $\langle \text{attribute} \rangle \text{op} \langle \text{attribute} \rangle$ or $\langle \text{constant} \rangle$ 其中op是以下之一: $=, \neq, >, \geq, <, \leq$

2.3 投影操作示例

$$r =$$

A	B	C
α	10	1
α	20	1
β	30	1
β	40	2

$$\Pi_{A,C}(r) =$$

A	C
α	1
α	1
β	1
β	2

 \Rightarrow

A	C
α	1
β	1
β	2

2.4 投影操作规范化

符号: $\Pi_{A_1, A_2, \dots, A_k}(r)$ 。其中 A_1, \dots, A_k 是属性名称, r 是关系名称。

结果定义为通过删除未列出的列而获得的 k 列的关系。

从结果中删除重复行, 因为关系是集合。

2.5 并集操作示例

$$r =$$

A	B
α	1
α	2
β	1

 $s =$

A	B
α	2
β	3

$$r \cup s =$$

A	B
α	1
α	2
β	1
β	3

2.6 并集操作形式化

符号: $r \cup s$

定义为: $r \cup s = \{t | t \in r \text{ or } t \in s\}$

为了使 $r \cup s$ 有效: r 和 s 必须具有相同的元数 (即, 相同的数量的属性); 属性域必须兼容。

2.7 差集运算示例

$$r = \begin{array}{|c|c|} \hline \text{A} & \text{B} \\ \hline \alpha & 1 \\ \hline \alpha & 2 \\ \hline \beta & 1 \\ \hline \end{array} \qquad s = \begin{array}{|c|c|} \hline \text{A} & \text{B} \\ \hline \alpha & 2 \\ \hline \beta & 3 \\ \hline \end{array}$$
$$r - s = \begin{array}{|c|c|} \hline \text{A} & \text{B} \\ \hline \alpha & 1 \\ \hline \beta & 1 \\ \hline \end{array}$$

2.8 差集运算形式化

表示法: $r - s$

定义为: $r - s = \{t | t \in r \text{ and } t \notin s\}$

差集必须在兼容关系之家进行: r 和 s 必须具有相同的元数, r 和 s 的属性域必须兼容。

2.9 笛卡尔积操作示例

$r =$	A	B	$s =$	C	D	E
	α	1		α	10	a
	β	2		β	10	a
				β	20	b
				γ	10	b

$r \times s =$	A	B	C	D	E
	α	1	α	10	a
	α	1	β	10	a
	α	1	β	20	b
	α	1	γ	10	b
	β	2	α	10	a
	β	2	β	10	a
	β	2	β	20	b
	β	2	γ	10	b

2.10 笛卡尔积操作形式化

表示法: $r \times s$

定义为: $r \times s = \{(t, q) | t \in r \text{ and } q \in s\}$

假设 $r(R)$ 和 $s(S)$ 是属性是互不相交的 (即 $R \cap S = \phi$)。如果 $r(R)$ 和 $s(S)$ 的属性不是互不相交的, 则必须对属性进行重命名。

2.11 重命名操作

允许我们为关系代数表达式的结果命名, 从而引用它们 (过程性)。

允许我们用多个名称引用一个关系。例如, $\rho_x(E)$ 在名称 X 下返回表达式 E 。如果关系代数表达式 E 的元数为 n , 则 $\rho_{X(A_1, A_2, \dots, A_n)}(E)$ (对关系 E 及其属性进行重命名) 返回表达式的结果 E

3 附加关系代数操作

四种基本符号: Set intersection, Natural Join, Division, Assignment

我们定义了额外的操作, 这些操作并不增加关系代数的能力, 但简化了常见查询。

3.1 集合交集操作形式化

符号: $r \cap s$

定义为: $r \cap s = \{t | t \in r \text{ and } t \in s\}$

假设: r 和 s 具有相同的元数; 属性兼容。

$$r \cap s = r - (r - s)$$

3.2 自然连接操作的形式化

符号: $r \bowtie s$

设 r 和 s 分别是模式 R 和 S 上的关系。那么, $r \bowtie s$ 是在模式 $R \cup S$ 上获得的关系, 考虑每对元组来自 r 的 t_r 和来自 s 的 t_s ; 如果 t_r 和 t_s 在 $R \cap S$ 的每个属性上具有相同的值, 则将元组 t 添加到结果中, 其中 t 在 r 上与 t_r 具有相同的值; t 在 s 上与 t_s 具有相同的值。

注意:

1. r, s 必须含有共同属性 (名称和域都对应相同)
2. 连接两个关系中同名属性值相等的元组
3. 结果属性是两个属性集的并集, 但消去重名属性

3.3 θ 连接操作的形式化

符号: $r \bowtie_{\theta} s$ 。其中 θ 是模式中属性的谓词。

$$\theta\text{连接: } r \bowtie_{\theta} s = \sigma_{\theta}(r \bowtie s)$$

3.4 除法运算

适合所有包含"for all"的语句。

实际上, 它确定了一个集合是否包含另一个集合。

符号: $r \div s$

设 r 和 s 分别是模式 R 和 S 上的关系, $R = (A_1, \dots, A_m, B_1, \dots, B_n)$ 和 $S = (B_1, \dots, B_n)$ 。那么, $r \div s$ 的结果是模式 $R - S = (A_1, \dots, A_m)$ 上的一个关系

$$r \div s = \{t | t \in \Pi_{R-S}(r) \wedge \forall u \in S (tu \in r)\}$$

注意 $\Pi_{R-S}(r)$ 包含了 $r \div s$ 的结果, 同时, 元组 t 的并集与 s 中的所有元组被 r 覆盖 (即, 商来自于 $\Pi_{R-S}(r)$, 并且其元组 t 与 s 所有元组的拼接被 r 覆盖)

属性/特性: 设 $q = r \div s$ 则 q 是满足 $q \times s \subset r$ 的最大关系。

基本代数运算的定义：设 $r(R)$ 和 $s(S)$ 为关系，且设 $S \subset R$ ，则

$$r \div s = \Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times s) - \Pi_{R-S,S}(r))$$

3.5 赋值操作

←提供了一种方便的方式来表达复杂查询。

将查询写成一个由以下组成的顺序程序：一系列赋值；后面跟着一个表达式，其值作为查询的结果显式。

赋值必须始终在临时关系变量中进行

在←右侧的结果被赋值给左侧的关系变量。

可以在后续表达式中使用变量。

4 扩展关系代数操作

4.1 广义投影

通过允许在投影列表中使用算术函数来扩展投影操作。

$$\Pi_{F_1, F_2, \dots, F_n}(E)$$

其中 E 是任何关系代数表达式， F_1, F_2, \dots, F_n 是涉及常量和 E 模式中属性的算术表达式。

给定一个关系credit-info(customer-name, limit, credit-balance)，找出每个人还能花多少钱：

$$\Pi_{\text{customer-name}, \text{limit-credit-balance}}(\text{credit-info})$$

4.2 聚合函数和操作

聚合函数接受一组值并返回一个单一值作为结果。

avg:平均值;min:最小值;max:最大值;sum:值的总和;count:值的数量。

$$G_1, G_2, \dots, G_n \mathcal{G}_{F_1(A_1), F_2(A_2), \dots, F_n(A_n)}(E)$$

其中 E 是任何关系代数表达式， G_1, G_2, \dots, G_n 是用于分组的属性列表(可以为空)，每个 F_i 是一个聚合函数，每个 A_i 是一个属性名称。

聚合结果没有名称。可以使用重命名操作作为其命名。为了方便，我们允许在聚合操作中进行重命名。

5 数据库的修改

可以使用以下操作修改数据库的内容：删除、插入、更新。

所有这些操作都使用赋值运算符表示。

5.1 删除

删除请求的表达方式与查询类似，不同之处在于选定的元组从数据库中被移除，而不是显式给用户。

它只能删除整个元组；不能删除某些特定属性上的值。

删除在关系代数中表示为： $r \leftarrow r - E$ 。其中 r 是一个关系， E 是一个关系代数查询。

5.2 插入

要将数据插入关系中，我们可以：指定要插入的元组；编写一个查询，其结果是一组要插入的元组。

在关系代数中，插入表示为： $r \leftarrow r \cup E$ 。其中 r 是一个关系， E 是一个关系代数查询。

单个元组的插入通过让 E 成为一个包含一个元组的常量关系来表示。

5.3 更新

一种不更改元组中所有值是情况下更改元组的某个值的机制。

使用广义投影算子来完成此任务： $r \leftarrow \Pi_{F_1, F_2, \dots, F_n}(r)$ 。其中每个 F_i 要么是 r 的 i 属性，如果 i 属性未更新；或者如果属性需要更新， F_i 是一个仅涉及常量和 r 属性的表达式，给出该属性的新值。