

# 第二章 关系模型与关系代数

陶煜波

计算机学院CAD&CG国家重点实验室

# 第二章 关系模型与关系代数

---

- 2.1 关系模型
  - 2.1.1 关系数据结构
  - 2.1.2 关系操作
  - 2.1.3 关系完整性
- 2.2 关系代数

## 2.1 关系模型

---

- 关系模型是一种数据模型，它和一般的数据模型一样，由如下三部分组成：
  - 关系数据结构
  - 关系操作集合
  - 关系完整性约束

## 2.1.1 关系数据结构

---

- 单一的数据结构 —— 关系
  - 在关系模型中，无论是实体还是实体间的各种联系均用关系来表示
- 数据的逻辑结构 —— 二维表
  - 从用户角度，关系模型中数据的逻辑结构是一张二维表
- 关系模型采用关系（**Relation**）作为数据结构，直观地讲，关系就是简单的表（**Table**）。一个表一般由表名、表头和数据三部分构成

## 2.1.1 关系数据结构

- 定义2.1 域（Domain）是一组具有相同数据类型的值的集合
- 例如，整数、实数都是域。域可以被理解为程序设计语言中的数据类型，如C语言中的int，float等
- 定义2.2 给定一组域 $D_1, D_2, \dots, D_n$ ， $D_1, D_2, \dots, D_n$ 的笛卡尔积(Cartesian Product)为：

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

- 其中，每一个元素 $(d_1, d_2, \dots, d_n)$ 叫作一个 $n$ 元组
- (N-Tuple)或简称元组(Tuple)。元素中的每一个值 $d_i$ 叫作一个分量(Component)

## 2.1.1 关系数据结构

---

- 属性(Attribute): 关系中的一列即为一个属性
- 域: 属性的取值范围称为该属性的域
- 码: 可以唯一确定一个元组的最小属性集合称为候选码 (Candidate Key), 或简称为码(Key)
- 分量: 元组中的一个属性值
- 关系模式(Relation Schema): 对关系的描述, 一般表示为: 关系名 (属性1, 属性2, ..., 属性n)
- 关系实例( Relation Instance): 关系的内容

## 2.1.1 关系数据结构

- 一个关系由关系名、关系模式和关系实例组成，分别对应于表名、表头和表中的数据。关系名和关系模式是相对变化的，关系实例会随时间而发生变化

关系模式

关系实例

学生 ( 学号,	姓名,	性别,	类(专业),	成绩,	学分)
3050644001	徐绮阳	女	地理信息系统		
3050644002	应晓航	女	地理信息系统		
3050644003	包黎莉	女	地理信息系统		
3050644004	俞春娜	女	地理信息系统		
3050644006	司徒笑微	女	地理信息系统		
3050644007	罗军	男	地理信息系统		
3050644008	楼阳波	男	地理信息系统		
3050644009	李欣	男	地理信息系统		
3050644010	陈林开	男	地理信息系统		

## 2.1.1 关系数据结构

---

- 关系可以有三种类型：
  - **基本关系**（通常又称为基本表或基表），基本表是实际存在的表，是实际存储数据的逻辑表示
  - **查询表**，是查询结果对应的表
  - **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



## 2.1.2 关系操作

---

- 关系操作的特点

- 集合操作方式，即操作的对象和结果都是集合
- 非关系数据模型的数据操作方式：一次一记录

- 常用的关系操作

- 查询
  - 选择、投影、连接、除、并、交、差
- 数据更新
  - 插入、删除、修改
- 查询的表达能力是其中最主要的部分

## 2.1.2 关系操作

---

- 关系数据语言的种类
  - 关系代数语言：用对关系的运算来表达查询要求
  - 关系演算语言：用谓词来表达查询要求
  - 介于关系代数和关系演算之间的语言，即结构化查询语言 SQL

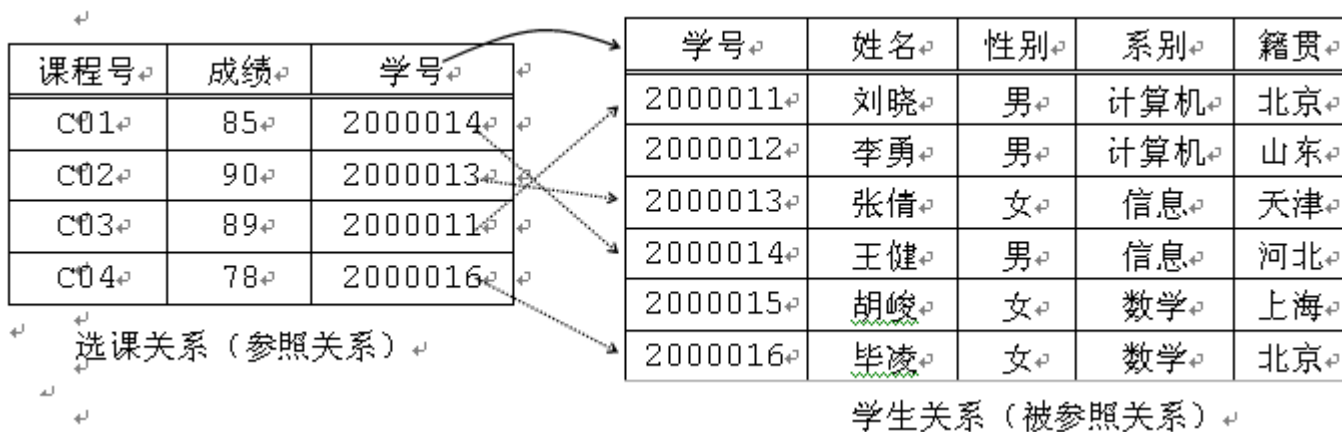
## 2.1.3 关系的完整性

---

- 关系模型的完整性规则是对关系的某种约束条件
- 关系模型中三类完整性约束：
  - 实体完整性
  - 参照完整性
  - 用户定义的完整性
- 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，被称作是关系的两个不变性，应该由关系系统自动支持

## 2.1.3 关系的完整性

- 实体完整性规则（Entity Integrity）：
  - 若属性A是基本关系R的主属性，则任何一个元组在属性A上不能取空值（Null）
- 参照完整性
  - 关系间的引用
  - 外码
  - 参照完整性规则



## 2.1.3 关系的完整性

---

- 用户定义的完整性
- 举例：课程(课程号，课程名，学分)
  - “课程号” 属性必须取唯一值
  - 非主属性“课程名”不能取空值
  - “学分”属性只能取值{1, 2, 3, 4}

## 2.1 关系模型小结

---

- Schema = 数据库中关系的结构描述
- Instance = 实际存储内容
- Database = sets of named relations (tables)
- Each relation has a set of named attributes (or columns)
- Each tuple (row) has a value for each attribute
- Each attribute has a Type (Domain)

## 2.1 关系模型小结

---

- NULL
  - Special value for “unknown” or “undefined”
- Key
  - Attribute whose value is unique in each tuple
  - Set of attributes whose combined values are unique
- Relationships are implicit: no pointers
- Creating relations (tables) in SQL
  - Create Table ...

# 第二章 关系模型与关系代数

---

- 2.1 关系模型
- 2.2 关系代数
  - 2.2.1 概述
  - 2.2.2 传统的集合运算
  - 2.2.3 专门的关系运算



## 2.2.1 概述

---

- 关系代数
  - 一种抽象的查询语言
  - 用对关系的运算来表达查询
- 关系代数运算的三个要素
  - 运算对象：关系
  - 运算结果：关系
  - 运算符：四类

## 2.2.1 概述

---

- 集合运算符
  - 将关系看成元组的集合
  - 运算是从关系的“水平”方向即行的角度来进行
- 专门的关系运算符
  - 不仅涉及行而且涉及列
- 算术比较符
  - 辅助专门的关系运算符进行操作
- 逻辑运算符
  - 辅助专门的关系运算符进行操作

# 2.2.1 概述

- 关系代数运算符

运算符		含义	运算符		含义
集合运算符	$\cup$	并	比较运算符	$>$	大于
	$-$	差		$\geq$	大于等于
	$\cap$	交		$<$	小于
	$\times$	笛积		$\leq$	小于等于
		广义笛积		$=$	等于
		卡积		$\neq$	不等于

## 2.2.1 概述

- 关系代数运算符

运算符	含义		运算符	含义	
专门的关系运算符	$\sigma$	选择	逻辑运算符	$\neg$	非
	$\pi$	投影		$\wedge$	与
	$\bowtie$	连接		$\vee$	或
	$\div$	除			

## 2.2.2 传统的集合运算

---

- 并
- 交
- 差
- 广义笛卡尔积

## 2.2.2 传统的集合运算

*R*

<i>A</i>	<i>B</i>	<i>C</i>
<i>a1</i>	<i>b1</i>	<i>c1</i>
<i>a1</i>	<i>b2</i>	<i>c2</i>
<i>a2</i>	<i>b2</i>	<i>c1</i>

*S*

<i>A</i>	<i>B</i>	<i>C</i>
<i>a1</i>	<i>b2</i>	<i>c2</i>
<i>a1</i>	<i>b3</i>	<i>c2</i>
<i>a2</i>	<i>b2</i>	<i>c1</i>

*RUS*

<i>A</i>	<i>B</i>	<i>C</i>
<i>a1</i>	<i>b1</i>	<i>c1</i>
<i>a1</i>	<i>b2</i>	<i>c2</i>
<i>a1</i>	<i>b3</i>	<i>c2</i>
<i>a2</i>	<i>b2</i>	<i>c1</i>

## 2.2.2 传统的集合运算

$R$

$A$	$B$	$C$
$a1$	$b1$	$c1$
$a1$	$b2$	$c2$
$a2$	$b2$	$c1$

$S$

$A$	$B$	$C$
$a1$	$b2$	$c2$
$a1$	$b3$	$c2$
$a2$	$b2$	$c1$

$R \cap S$

$A$	$B$	$C$
$a1$	$b2$	$c2$
$a2$	$b2$	$c1$

## 2.2.2 传统的集合运算

*R*

<i>A</i>	<i>B</i>	<i>C</i>
<i>a1</i>	<i>b1</i>	<i>c1</i>
<i>a1</i>	<i>b2</i>	<i>c2</i>
<i>a2</i>	<i>b2</i>	<i>c1</i>

*S*

<i>A</i>	<i>B</i>	<i>C</i>
<i>a1</i>	<i>b2</i>	<i>c2</i>
<i>a1</i>	<i>b3</i>	<i>c2</i>
<i>a2</i>	<i>b2</i>	<i>c1</i>

*R-S*

<i>A</i>	<i>B</i>	<i>C</i>
<i>a1</i>	<i>b1</i>	<i>c1</i>



# 2.2.2 传统的集合运算

*R*

<i>A</i>	<i>B</i>	<i>C</i>
<i>a1</i>	<i>b1</i>	<i>c1</i>
<i>a1</i>	<i>b2</i>	<i>c2</i>
<i>a2</i>	<i>b2</i>	<i>c1</i>

*S*

<i>A</i>	<i>B</i>	<i>C</i>
<i>a1</i>	<i>b2</i>	<i>c2</i>
<i>a1</i>	<i>b3</i>	<i>c2</i>
<i>a2</i>	<i>b2</i>	<i>c1</i>

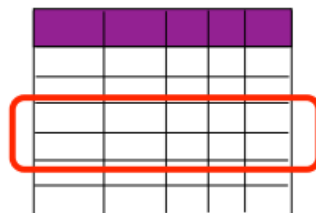
$R \times S$

<i>A</i>	<i>B</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>C</i>
<i>a1</i>	<i>b1</i>	<i>c1</i>	<i>a1</i>	<i>b2</i>	<i>c2</i>
<i>a1</i>	<i>b1</i>	<i>c1</i>	<i>a1</i>	<i>b3</i>	<i>c2</i>
<i>a1</i>	<i>b1</i>	<i>c1</i>	<i>a2</i>	<i>b2</i>	<i>c1</i>
<i>a1</i>	<i>b2</i>	<i>c2</i>	<i>a1</i>	<i>b2</i>	<i>c2</i>
<i>a1</i>	<i>b2</i>	<i>c2</i>	<i>a1</i>	<i>b3</i>	<i>c2</i>
<i>a1</i>	<i>b2</i>	<i>c2</i>	<i>a2</i>	<i>b2</i>	<i>c1</i>
<i>a2</i>	<i>b2</i>	<i>c1</i>	<i>a1</i>	<i>b2</i>	<i>c2</i>
<i>a2</i>	<i>b2</i>	<i>c1</i>	<i>a1</i>	<i>b3</i>	<i>c2</i>
<i>a2</i>	<i>b2</i>	<i>c1</i>	<i>a2</i>	<i>b2</i>	<i>c1</i>

## 2.2.3 专门的关系运算

- An Algebra of Tables

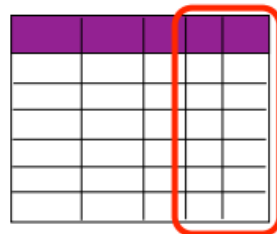
- 选择
- 投影
- 连接
- 除







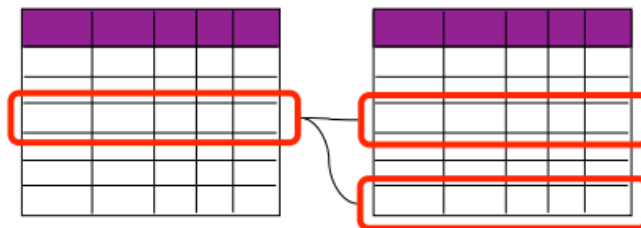
select







project











join

## 2.2.3 专门的关系运算

- 选择针对单个关系中的数据进行操作，属于一元运算符，是指从关系中选择若干行
- 例如，可以使用选择操作符（ $\sigma$ ）从学生关系S中选出那些年龄大于19岁的学生。该查询可以用如下的表达式来表示：

$$\sigma_{\text{Sage} > 19}(S)$$

查询表达式中的下标“Sage>19”给出的是选择条件，只有符合该条件的元组才可以被返回到结果中

## 2.2.3 专门的关系运算

S

学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	系别 Sdept
2000011	刘晓	男	20	计算机
2000012	李勇	男	19	计算机
2000013	张倩	女	20	信息
2000015	胡峻	女	19	数学
2000016	毕凌	女	19	数学

$\sigma_{Sage>19}(S)$

学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	系别 Sdept
2000011	刘晓	男	20	计算机
2000013	张倩	女	20	信息

## 2.2.3 专门的关系运算

- 投影操作 ( $\pi$ ) 是指从关系中选择若干列
- 例如, 可以使用投影操作符 ( $\pi$ ) 从学生关系 **S** 中找出所有学生的名字和年龄。该查询可以用如下的表达式来表示:

$$\pi_{\text{Sname, Sage}}(\text{S})$$

查询表达式中的下标 “**Sname, Sage**” 指出需要返回的列的名字, 其他的列在结果关系中被去掉了

## 2.2.3 专门的关系运算

S

学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	系别 Sdept
2000011	刘晓	男	20	计算机
2000012	李勇	男	19	计算机
2000013	张倩	女	20	信息
2000015	胡峻	女	19	数学
2000016	毕凌	女	19	数学

$\pi_{\text{Sname, Sage}}(S)$

姓名 Sname	年龄 Sage
刘晓	20
李勇	19
张倩	20
胡峻	19
毕凌	19

思考：  $\pi_{\text{Sage}}(S)$ 的结果是？

## 2.2.3 专门的关系运算

- 思考：选择与投影满足交换律？

$$\sigma_c(\pi_{A1, \dots, An}(S)) = ? \pi_{A1, \dots, An}(\sigma_c S)$$

- $\sigma_{\text{Sage} < 20}(\pi_{\text{Sno}, \text{Sname}}(S))$

- $\pi_{\text{Sno}, \text{Sname}}(\sigma_{\text{Sage} < 20}(S))$

S

学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	系别 Sdept
2000011	刘晓	男	20	计算机
2000012	李勇	男	19	计算机
2000013	张倩	女	20	信息
2000015	胡峻	女	19	数学
2000016	毕凌	女	19	数学

## 2.2.3 专门的关系运算

- Theta-Join最有用的操作之一
- 连接运算的含义
  - 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

$$R \bowtie_{\text{条件}} S$$

- 其中条件的一般形式是 $A\theta B$ ， $\theta = \{ =, >, >=, <, <=, <> \}$ ， $A$ 是关系 $R$ 中的属性或者是一个常数， $B$ 是关系 $S$ 中的属性或者是一个常数， $A$ 和 $B$ 必须是同一个定义域（相同的类型）。还可以用逻辑运算符和上面的一般形式构成更复杂的条件
- $\pi_{\text{name}}(\text{Hospitals} \bowtie_{\text{distance}(\text{location}, \text{location}) < 5} \text{Schools})$



## 2.2.3 专门的关系运算

---

- 两类常用连接运算

- (1) 等值连接 (equijoin)

- 什么是等值连接

- 当条件 $A\theta B$ 中的 $\theta$ 为“=”的连接运算称为等值连接

- 等值连接的含义

- 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

## 2.2.3 专门的关系运算

- 两类常用连接运算
  - (1) 等值连接 (equijoin)
  - (2) 自然连接 (Natural join)
    - 什么是自然连接
      - 自然连接是一种特殊的等值连接
        - » 它要求关系 $R$ 中的属性 $A$ 和关系 $S$ 中的属性 $B$ 名字相同
        - » 在结果中把重复的属性列去掉
    - 自然连接的含义
      - $R$ 和 $S$ 具有相同的属性组 $B$
- 一般的连接操作是从行的角度进行运算。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

## 2.2.3 专门的关系运算

- 等值连接举例

S

学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	系别 Sdept
2000011	刘晓	男	20	计算机
2000012	李勇	男	19	计算机
2000013	张倩	女	20	信息
2000015	胡峻	女	19	数学
2000016	毕凌	女	19	数学

SC

学号 Sno	课程号 Cno	成绩 Grade
2000012	C01	98
2000015	C02	80
2000015	C03	76
2000017	C04	90

学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	系别 Sdept	学号 Sno	课程号 Cno	成绩 Grade
2000012	李勇	男	19	计算机	2000012	C01	98
2000015	胡峻	女	19	数学	2000015	C02	80
2000015	胡峻	女	19	数学	2000015	C03	76

$S \bowtie SC$   
 $S \Join SC = SC \Join S$

## 2.2.3 专门的关系运算

- 自然连接举例

S

学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	系别 Sdept
2000011	刘晓	男	20	计算机
2000012	李勇	男	19	计算机
2000013	张倩	女	20	信息
2000015	胡峻	女	19	数学
2000016	毕凌	女	19	数学

SC

学号 Sno	课程号 Cno	成绩 Grade
2000012	C01	98
2000015	C02	80
2000015	C03	76
2000017	C04	90

学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	系别 Sdept	课程号 Cno	成绩 Grade
2000012	李勇	男	19	计算机	C01	98
2000015	胡峻	女	19	数学	C02	80
2000015	胡峻	女	19	数学	C03	76

$S \bowtie SC$

## 2.2.3 专门的关系运算

- 下图中的关系A表示某个学生选修了某门课程，B表示可选的选修课列表。 $A \div B$ 表示选修了 $B_i$ 中所有课程的学生名单

A	学号 Sno	课程号 Cno
	s1	C1
	s1	C2
	s1	C3
	s1	C4
	s2	C1
	s2	C2
	s3	C2
	s4	C2
	s4	C4

B1	课程号 Cno
	C2

B2	课程号 Cno
	C2
	C4

B3	课程号 Cno
	C1
	C2
	C4

$A \div B1$	学号 Sno
	s1
	s2
	s3
$A \div B2$	学号 Sno
	s1
	s4
$A \div B3$	学号 Sno
	s1

## 2.2.3 专门的关系运算

- 除运算可以这样来理解：假设有两个关系A和B，其中A只有两个属性X和Y，B只有一个属性Y(和关系A中的Y属性具有相同的域),对于A中的每一个x(属性X的取值)所对应的y（属性Y的取值）集合，看它是否包含B中所有的y值，如果包含，则x属于 $A \div B$ 的结果集

思考：除与笛卡尔积的关系？

# 重命名

---

- $\rho_{R(A_1, \dots, A_n)}(S)$
- $\rho_R(S)$
- $\rho_{A_1, \dots, A_n}(S)$
- 用途
  - To unify schemas for set operators
$$\rho_{c(\text{name})}(\pi_{\text{Sname}}(\text{Student})) \cup \rho_{c(\text{name})}(\pi_{\text{Cname}}(\text{Course}))$$
  - For disambiguation in “self-joins”
$$\rho_{c1(\text{no1}, \text{n1}, \text{p1}, \text{c})}(\text{Course}) \bowtie \rho_{c2(\text{no2}, \text{n2}, \text{p2}, \text{c})}(\text{Course})$$

## 2.2 关系代数小结

- 核心操作

- 关系
- 选择
- 投影
- 笛卡尔积
- 并
- 差

- 扩展操作

- 连接 (natural, equi-join, theta join, semi-join)
- 交
- 除

- 辅助操作

- 重命名

思考1：扩展操作能否用核心操作实现？

思考2：SQL中的操作都能在关系代数中找到对应符号？

思考3：关系代数是否有不足？



# 第二章 关系模型与关系代数

---

- 2.1 关系模型
  - 2.1.1 关系数据结构
  - 2.1.2 关系操作
  - 2.1.3 关系完整性
- 2.2 关系代数
  - 2.2.1 概述
  - 2.2.2 传统的集合运算
  - 2.2.3 专门的关系运算