第二章　关系数据库

关系模型、关系代数

本章主要学习内容

- **关系模型**
  - 关系数据结构
  - 关系操作
  - 关系的完整性约束
- **关系代数**
  - 集合计算
  - 关系运算
- **关系演算**
  - 元组关系演算语言ALPHA
  - 域关系演算语言QBE

一、关系数据结构及形式化定义

- 关系：现实世界的实体以及实体间的各种联系**均用关系**来表示
- 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
- 关系操作的对象和结果都是集合，关系模型建立在**集合代数**的基础上；

1、域

- **域（Domain）**：一组具有相同数据类型的值的集合，也称为值域，用 D 表示。
- 域中所包含的值的个数称为域的**基数**，用 m 表示。
- 关系中用域表示属性的取值范围。例如：
  - D1={李力，王平，刘伟}　　m1=3
  - D2={男，女}　　m2=2
  - D3={47,28,30}　　m3=3
  - 其中，D1，D2，D3为域名，分别表示教师关系中姓名、性别、年龄的集合。
- 域的取值无排列次序，如D2={男，女}={女，男}

2、笛卡尔积

- **1）笛卡尔积（Cartesian Product）**

  给定一组域$D_1, D_2, \ldots D_n$(域可相同），它们的笛卡儿积为：

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

  - 所有域的所有取值组成一个集合，其中每一个元素（$d_1, d_2, \ldots, d_n$)叫做一个n元组，简称元组。
  - 元组中的每个值d_i叫做一个分量。元组的每个分量（di）是按序排列的。如：（1，2，3）≠（2，3，1）≠（1，3，2）；
  - 元组不能重复，元组之间是无序的。

2）实例

例1：D1={a1,a2}

D2={b1,b2,b3}

则：

D1XD2={(a1,b1),(a1,b2),(a1,b3),(a2,b1),(a2,b2),(a2,b3)}

- 其中a1、b1、b2等是分量
- (a1,b1),(a1,b2)等是元组
- 该笛卡尔积的基数为2x3=6
- 元组的个数为6

例2： D1=导师集合SUPERVISOR=张清玫，刘逸

D2=专业集合SPECIALITY=计算机专业，信息专业

D3=研究生集合POSTGRADUATE=李勇，刘晨，王敏

则：D1XD2XD3=

{（张清玫，计算机专业，李勇），
（张清玫，计算机专业，刘晨），
（张清玫，计算机专业，王敏），
（张清玫，信息专业，李勇），
（张清玫，信息专业，刘晨），
（张清玫，信息专业，王敏），
（刘逸，计算机专业，李勇），
（刘逸，计算机专业，刘晨），
（刘逸，计算机专业，王敏），
（刘逸，信息专业，李勇），
（刘逸，信息专业，刘晨），
（刘逸，信息专业，王敏）}

该笛卡尔积的基数为2x2x3=12

即元组的个数为12

3）笛卡尔积的二维表表示

- 笛卡尔积也可以用二维表表示，其中表的框架由域构成，表的任意一行就是一个元组，每一列数据来自同一域。

例1：D1=学生的集合{甲，乙，丙}

D2=性别的集合{男，女}

D3=班级的集合{01，02}

共2×2×3=12个元组，用二维表可表示为：

D1	D2	D3
甲	男	01
甲	男	02
甲	女	01
甲	女	02
乙	男	01
乙	男	02
乙	女	01
乙	女	02
丙	男	01
丙	男	02
丙	女	01
丙	女	02

3、关系

- **关系（Relation）**
  - $D_1 \times D_2 \times \ldots \times D_n$的子集叫作在域$D_1$，$D_2$，…，$D_n$上的关系。
  - 表示为R（D_1，D_2，…，D_n）
  - R：关系名
  - n：关系的目或度（Degree）

几点说明

- 1）单元关系与二元关系
  - n：关系的目或度（Degree）
  - 当n=1时，称该关系为单元关系（Unary relation）或一元关系。
  - 当n=2时，称该关系为二元关系（Binary relation）
  - …
  - 当n=n时，称为n元关系。

- 2）在数学上，关系是笛卡尔积的任意子集，按照笛卡儿积的定义，关系可以是一个无限集合。
- 但在实际应用中**关系是笛卡尔积中所取的有意义的子集**。例如在表中选取一个子集构成如下关系，显然不符合实际情况。

姓名	性别
李力	男
李力	女

- **属性（Attribute）**
  - 关系中不同列可以对应相同的域
  - 为了加以区分，必须对每列起一个名字，称为属性名
  - *n*目关系必有*n*个属性
- **码（Key）**

  （1）候选码（Candidate key）
  - 若关系中的某一属性组的值能唯一地标识一个元组，则称该属性组为候选码。
  - 最简单的情况：候选码只包含一个属性

（2）全码（All-key）

- 最极端的情况：关系模式的所有属性组是关系模式的候选码，称为全码。

（3）主码（Primary key）

- 若一个关系有多个候选码，则选定其中一个为主码。

（4）主属性（Prime attribute）

- 候选码的诸属性称为主属性

（5）非主属性（Non-key attribute）

- 不包含在任何侯选码中的属性称为非主属性或非码属性。

基本关系的6条性质

① 列是同质的

② 不同的列可出自同一个域

③ 列的顺序无所谓，列的次序可以任意交换

④ 任意两个元组的候选码不能相同

⑤ 行的顺序无所谓，行的次序可以任意交换

⑥ 分量必须取原子值

SUPERVISOR	SPECIALITY	POSTGRADUATE	
		PG1	PG2
张清玫	信息专业	李勇	刘晨
刘逸	信息专业	王敏	

小表

4、关系模式

- 关系模式是对关系的描述
- 关系模式是型，关系是值
- 关系模式：静态的、稳定的
- 关系
  - 关系模式在某一时刻的状态或内容
  - 动态的、随时间不断变化的
- 关系模式和关系往往统称为关系，通过上下文加以区别

- **关系模式可以形式化地表示为：**

  R（U，D，DOM，F）

  R　关系名

  U　组成该关系的属性名集合

  D　属性组U中属性所来自的域

  DOM　属性向域的映象集合

  F　　属性间的数据依赖关系集合

注：域名及属性向域的映象常常直接说明为属性的类型、长度

- 关系模式通常可以简记为

  $R(U)$ **或** $R(A_1, A_2, \ldots, A_n)$

  - R：关系名
  - A_1，A_2，…，A_n ：属性名

5、关系数据库

- 关系数据库
  - 在一个给定的应用领域中，所有关系的集合构成一个关系数据库。
- 关系数据库的**型**：关系数据库模式，对数据库的描述。是稳定的。
- 关系数据库的**值**：关系模式在某一时刻对应的关系的集合。通常就称为关系数据库。

二、关系操作

- 常用的关系操作
  - **查询**：查询是关系操作中最主要的部分，包括选择、投影、连接、除、并、交、差、笛卡尔积，其中选择、投影、并、差、笛卡尔积是5种基本操作。
  - **数据更新**：插入、删除、修改
- 关系操作的特点
  - **集合操作方式**：操作的对象和结果都是集合，一次一集合的方式

关系数据语言的分类

- 关系代数语言
  - 用对关系的运算来表达查询要求，代表：ISBL
- 关系演算语言：用谓词来表达查询要求
  - 元组关系演算语言，谓词变元的基本对象是元组变量，代表：APLHA, QUEL
  - 域关系演算语言，谓词变元的基本对象是域变量，代表：QBE
- 具有关系代数和关系演算双重特点的语言
  - 代表：SQL，集查询、DDL、DML、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。

三、关系的完整性

- 为了维护数据库中数据与现实世界的一致性，对关系数据库的插入、删除和修改操作必须有一定的约束条件，这就是关系模型的三类完整性：
  - 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**：

  关系模型必须满足的完整性约束条件，称为关系的两个**不变性**，应该由关系系统自动支持。

- **用户定义的完整性**：

  应用领域需要遵循的约束条件，体现了具体领域中的语义约束。

1、实体完整性

规则2.1 实体完整性规则（Entity Integrity），若属性A是基本关系R的主属性，则属性A不能取空值。

如：学生（**学号**，姓名，性别，专业号，年龄）

课程（**课程号**，课程名，学分）

其中主码**学号，课程号**不可取空值。

实体完整性规则的说明

(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。

(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。

(3) 关系模型中以主码作为唯一性标识。

(4) 主码中的属性即主属性不能取空值。

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）点相矛盾，因此这个规则称为实体完整性

2、参照完整性

- 在关系模型中实体及实体间的联系都是用**关系**来描述的，因此可能存在着**关系与关系间的引用**。

例1：学生实体、专业实体

学生（**学号**，姓名，性别，**专业号**，年龄）

专业（**专业号**，专业名）

- 学生关系引用了专业关系的主码“专业号”。
- 学生关系中的“专业号”值必须是确实存在的专业的专业号 ，即专业关系中有该专业的记录。

例2：学生、课程、学生与课程之间的多对多联系

学生（学号，姓名，性别，专业号，年龄）

课程（课程号，课程名，学分）

选修（学号，课程号，成绩）

例3：学生实体及其内部的一对多联系

学生（学号，姓名，性别，专业号，年龄，班长）

学号	姓名	性别	专业号	年龄	班长
801	张三	女	**01**	**19**	**802**
802	李四	男	**01**	**20**	
803	王五	男	**01**	**20**	**802**
804	赵六	女	**02**	**20**	**805**
805	钱七	男	**02**	**19**	

- “学号”是主码，“班长”是外码，它引用了本关系的“学号”
- “班长” 必须是确实存在的学生的学号

- **外码（Foreign Key）**

设F是基本关系R的一个或一组属性，但不是关系R的码。如果F与基本关系S的主码Ks相对应，则称F是基本关系R的外码。

基本关系R称为参照关系（Referencing Relation）

基本关系S称为被参照关系（Referenced Relation）或目标关系（Target Relation）

规则2.2 参照完整性规则

若属性（或属性组）F是基本关系R的外码，它与基本关系S的主码K_S相对应（基本关系R和S不一定是不同的关系），则对于R中每个元组在F上的值必须为：

- 或者取空值（F 的每个属性值均为空值）
- 或者等于S 中某个元组的主码值

例1：

学生关系中每个元组的**“专业号”**属性只取两类值：

（1）**空值**，表示尚未给该学生分配专业

（2）非空值，这时该值必须**是专业关系中某个元组的“专业号”值**，该学生不可能分配一个不存在的专业。

例2：

选修（**学号**，**课程号**，成绩）

“学号”和“课程号”可能的取值 ：

（1）选修关系中的主属性，不能取空值

（2）只能取相应被参照关系中已经存在的主码值

例3：

学生（**学号**，姓名，性别，专业号，年龄，**班长**）

“班长”属性值可以取两类值：

（1）空值，表示该学生所在班级尚未选出班长

（2）非空值，该值必须是本关系中某个元组的学号值

3、用户定义的完整性

- 针对某一具体关系数据库的约束条件，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，而不要由应用程序承担这一功能。

例：　1）选修关系中成绩的取值范围为0~100之间

　　2）某个属性（如：课程名）必须取唯一值

- 最后，关系模型的优点：

1. 结构简单，具有高度的简明性和高度性
2. 可直接处理多对多关系
3. 一次处理一个元组集合
4. 数据独立性很高
5. 坚实的数学理论基础

四、关系代数

- 关系数据库的数据操纵语言（DML）的语句分成查询语句和更新语句两大类。查询语句用于描述用户的各种检索要求；更新语句用于描述用户进行插入、删除、修改等操作。
- 根据关系查询语言其理论基础的不同分成两大类：
  - 关系代数语言：查询操作是以**集合操作**为基础的演算
  - 关系演算语言：查询操作是以**谓词演算**为基础的演算

关系代数

- 关系代数是一种抽象的查询语言，属于关系操作的一种
- 通过对关系的运算来表达查询操作
- 运算的对象及结果均为关系
- 运算
  - 集合运算、关系运算、比较运算、逻辑运算

关系代数的运算符

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

- 由于关系定义为元数相同的元组的集合，因此把关系看成集合，集合代数中的运算（并、差、交、笛卡尔积）就可以引入到关系运算中来。还有一些运算是针对关系数据库环境专门设计的，譬如对关系进行垂直分割（投影）、水平分割（选择）、关系的结合（连接）等。

分别为

1. 传统的集合运算
2. 专门的关系运算

1、传统的集合运算

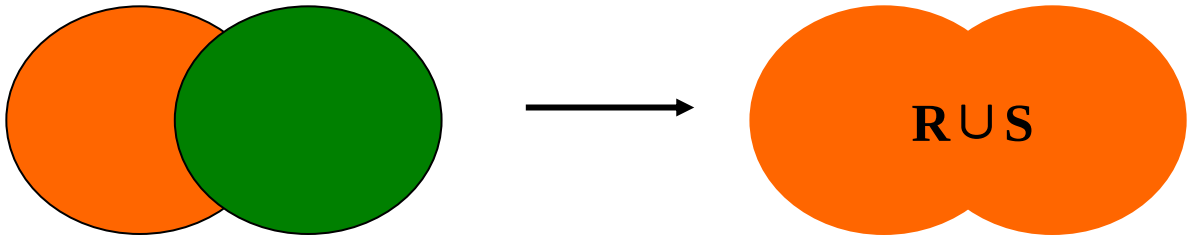
- **1）并（Union）**

  R和S

  - 具有相同的目n（即两个关系都有n个属性）
  - 相应的属性取自同一个域

  $R \cup S$

  - 仍为n目关系，由属于R或属于S的元组组成
  - 可表示为：$R \cup S = \{ t \mid t \in R \vee t \in S \}$



R

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1

S

A	B	C
a_1	b_2	c_2
a_1	b_3	c_2
a_2	b_2	c_1

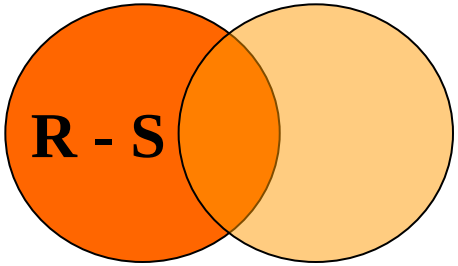
则R∪S为：

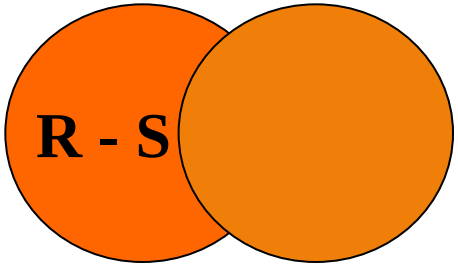
$R \cup S$

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1
a_1	b_3	c_2

2）差（Except ）

- R和S
  - 具有相同的目n
  - 相应的属性取自同一个域
- R-S
  - 仍为n目关系，由属于R而不属于S的所有元组组成
  - 可表示为： $R-S=\{\ t \mid t\in R\wedge t\notin S\ \}$





R

A	*B*	*C*
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1

S

A	*B*	*C*
a_1	b_2	c_2
a_1	b_3	c_2
a_2	b_2	c_1

则R-S为：

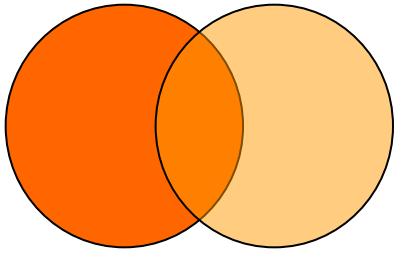
R-S

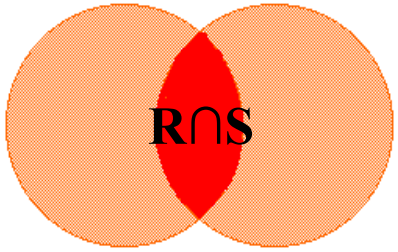
A	*B*	*C*
a_1	b_1	c_1

3）交（Intersect ）

- R和S
  - 具有相同的目n
  - 相应的属性取自同一个域
- $R \cap S$
  - 仍为n目关系，由既属于R又属于S的元组组成
  - 可表示为：$R \cap S = \{ t | t \in R \wedge t \in S \}$

$$R \cap S = R - (R - S) = \ ?$$





R

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1

S

A	B	C
a_1	b_2	c_2
a_1	b_3	c_2
a_2	b_2	c_1

则R∩S为：

$R \cap S$

A	B	C
a_1	b_2	c_2
a_2	b_2	c_1

4）笛卡尔积（Cartesian Product）

- 关系R、S的笛卡尔积是两个关系的元组的集合所组成的新关系。
- R×S：
  - **属性**是R和S的组合（n+m个列，有重复）
  - **元组**是R和S所有元组的可能组合（$K_1 \times k_2$个元组）
  - 是R、S的无条件连接，使任意两个关系的信息能组合在一起
  - 记作$R \times S = \{t_r\, t_s \mid t_r \in R \wedge t_s \in S\}$

R

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1

S

A	B	C
a_1	b_2	c_2
a_1	b_3	c_2
a_2	b_2	c_1

$R \times S$

A	B	C	A	B	C
a_1	b_1	c_1	a_1	b_2	c_2
a_1	b_1	c_1	a_1	b_3	c_2
a_1	b_1	c_1	a_2	b_2	c_1
a_1	b_2	c_2	a_1	b_2	c_2
a_1	b_2	c_2	a_1	b_3	c_2
a_1	b_2	c_2	a_2	b_2	c_1
a_2	b_2	c_1	a_1	b_2	c_2
a_2	b_2	c_1	a_1	b_3	c_2
a_2	b_2	c_1	a_2	b_2	c_1

2、专门的关系运算

- 选择、投影、连接、除法
- 符号介绍
  - 1）R，$t \in R$，$t[A_i]$
  - 2）A，t[A]， A
  - 3）$t_r \frown t_s$
  - 4）象集Z_x

1） R，t∈R，t[A_i]

设关系模式为R(A_1，A_2，…，A_n)

它的一个关系设为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

2）A，t[A]，$\bar{A}$

若**A**=$\{A_{i1}, A_{i2}, \ldots, A_{ik}\}$，其中$A_{i1}$，$A_{i2}$，…，$A_{ik}$是$A_1$，$A_2$，…，$A_n$中的一部分，则$A$称为属性列或属性组。

t[A]=$(t[A_{i1}], t[A_{i2}], \ldots, t[A_{ik}])$表示元组$t$在属性列$A$上诸分量的集合。

$\bar{A}$则表示$\{A_1, A_2, \ldots, A_n\}$中去掉$\{A_{i1}, A_{i2}, \ldots, A_{ik}\}$后剩余的属性组。

3）$\widehat{t_r}\,t_s$

R为n目关系，S为m目关系。

$t_r \in R$，$t_s \in S$，$\widehat{t_r}\,t_s$称为元组的连接。

$\widehat{t_r}\,t_s$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。

4）象集Z_x

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**象集**（Images Set）为：

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

它表示R中属性组X上值为x的诸元组在Z上分量的集合

R

x_1	Z_1
x_1	Z_2
x_1	Z_3
x_2	Z_2
x_2	Z_3
x_3	Z_1
x_3	Z_3

- x_1在R中的象集 $Z_{x1}=\{Z1, Z2, Z3\}$，
- x_2在R中的象集 $Z_{x2}=\{Z2, Z3\}$，
- x_3在R中的象集 $Z_{x3}=\{Z1, Z3\}$

1）选择（Selection）

- 从关系R中选择符合条件的元组构成新的关系
- $\sigma_F(R) = \{t|t \in R \wedge F(t) = \text{'真'}\}$
- **σ**为选取运算符
- σF(R),表示从R中选择满足条件(F表示选择条件)的元组
- 选取运算实际上是从关系R中选取使逻辑表达式为真的元组，是从行的角度进行的运算，**即对行的运算。**

选择运算练习

R

A	B	C
3	6	7
2	5	7
7	2	3
4	4	3

$\sigma_{A<5}(R)$

A	B	C
3	6	7
2	5	7
4	4	3

$\sigma_{A<5 \wedge C=7}(R)$

A	B	C
3	6	7
2	5	7

例1：查询计算机系的全体学生

$\sigma_{Sdept = 'IS'}(Student)$ 或 $\sigma_{5 ='IS'}(Student)$

例2：查询年龄小于20岁的男同学

$\sigma_{(Sage < 20) \wedge (SSex=’男’)}(Student)$

或$\sigma_{(4 < 20) \wedge (3=’男’)}(Student)$

注意：

- 对于$\sigma_{5 = ‘IS’}$，其中5为DEPT的属性序号，表示从Student中挑选第5个分量值等于IS的元组所构成的关系。
- 字符型数据的值应该使用单引号括起来，例如：’IS’，‘计算机’，‘男’。

2）投影（Projection）

- 对R的垂直分割，从关系R中选择若干属性组成新的关系
- $\pi_{A_1,A_2,\ldots,A_n}(R)$,表示从R中选择属性集$A_1,A_2,\ldots,A_n$组成新的关系
- **对列的运算**
- 投影运算的结果中,也要去除可能的重复元组

- 例1：查询学生关系中有哪些系

  $\Pi_{Sdept}(Student)$ 或 $\Pi_5(Student)$
- 例2：查询学生关系中学生的姓名和所在系

  $\Pi_{Sname，Sdept}(Student)$
- 例3：查询男同学所在的系

  $\Pi_{Sdept}(\sigma_{SEX=’男’}（Student）)$
- 例4：查询学号为95001的学生的课程号及成绩

  $\Pi_{Cno，Grade}(\sigma_{Sno=’95001’}（SC）)$

其中π_5(Student）表示关系Student只取第5列，组成新的关系。

3）连接（Join）

- 从R×S的笛卡尔积结果集中选取在指定的属性集上满足θ条件的元组，组成新的关系。
- $R \underset{A\theta B}{\bowtie} S = \{ \quad | \; t_r \widehat{t_r} t_s \in R \wedge t_s \in S \wedge t_r[A]\theta t_s[B] \}$
- 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

- $R \underset{A\theta B}{\bowtie} S$ S是在R和S的笛卡尔积中挑选第a个分量和S中第b个分量满足θ运算的元组。
- 两类常用连接运算
  - 等值连接（θ为“＝”的连接运算）
  - 自然连接（一种特殊的等值连接）

不等连接

R

A	B	C
a_1	b_1	5
a_1	b_2	6
a_2	b_3	8
a_2	b_4	12

S

B	E
b_1	3
b_2	7
b_3	10
b_3	2
b_5	2

则 $R \underset{C<E}{\bowtie} S$ 结果为

A	$R.B$	C	$S.B$	E
a_1	b_1	5	b_2	7
a_1	b_1	5	b_3	10
a_1	b_2	6	b_2	7
a_1	b_2	6	b_3	10
a_2	b_3	8	b_3	10

等值连接

R

A	*B*	*C*
a_1	b_1	5
a_1	b_2	6
a_2	b_3	8
a_2	b_4	12

S

B	*E*
b_1	3
b_2	7
b_3	10
b_3	2
b_5	2

R ⋈ S

R.B=S.B

A	*R.B*	*C*	*S.B*	*E*
a_1	b_1	5	b_1	3
a_1	b_2	6	b_2	7
a_2	b_3	8	b_3	10
a_2	b_3	8	b_3	2

自然连接

R

A	*B*	*C*
a_1	b_1	5
a_1	b_2	6
a_2	b_3	8
a_2	b_4	12

S

B	*E*
b_1	3
b_2	7
b_3	10
b_3	2
b_5	2

R ⋈ S

A	*B*	*C*	*E*
a_1	b_1	5	3
a_1	b_2	6	7
a_2	b_3	8	10
a_2	b_3	8	2

等值连接与自然连接的区别

- 1. **等值连接中不要求相等属性值的属性名相同，而自然连接要求相等属性值的属性名必须相同。**即两关系只有在同名属性才能进行自然连接。如上例R中的C列和S中的D列可进行等值连接，但因为属性名不同，不能进行自然连接。
- 2. **等值连接不将重复属性去掉，而自然连接去掉重复属性。**也可以说，自然连接是去掉重复列的等值连接。如上例R中的B列和S中的B列进行等值连接时，结果有两个重复的属性列B，而进行自然连接时，结果只有一个属性列B。

- 外连接
  - 如果把舍弃的元组也保存在结果关系中，而在其他属性上填空值(Null)，这种连接就叫做外连接（OUTER JOIN）。
- 左外连接
  - 如果**只把左边关系*R*中要舍弃的元组保留**就叫做左外连接(LEFT OUTER JOIN或LEFT JOIN)
- 右外连接
  - 如果**只把右边关系S中要舍弃的元组保留**就叫做右外连接(RIGHT OUTER JOIN或RIGHT JOIN)。

A	B	C	E
a_1	b_1	5	3
a_1	b_2	6	7
a_2	b_3	8	10
a_2	b_3	8	2
a_2	b_4	12	NULL
NULL	b_5	NULL	2

(a) 外连接

A	B	C	E
a_1	b_1	5	3
a_1	b_2	6	7
a_2	b_3	8	10
a_2	b_3	8	2
a_2	b_4	12	NULL

(b) 左外连接

A	B	C	E
a_1	b_1	5	3
a_1	b_2	6	7
a_2	b_3	8	10
a_2	b_3	8	2
NULL	b_5	NULL	2

(c) 右外连接

R

A	B	C
a1	b1	2
a1	b2	4
a2	b3	6
a2	b4	8

S

B	D
b1	5
b2	6
b4	7
b5	8

外连接

A	B	C	D
a1	b1	2	5
a1	b2	4	6
a2	b3	6	null
a2	b4	8	7
null	b5	null	8

左外连接

A	B	C	D
a1	b1	2	5
a1	b2	4	6
a2	b3	6	null
a2	b4	8	7

右外连接

A	B	C	D
a1	b1	2	5
a1	b2	4	6
a2	b4	8	7
null	b5	null	8

3、关系代数的应用实例

例：数据库中有三个关系

S(S#,SNAME,AGE,SEX)　　C(C#,CNAME,TEACHER)

SC(S#,C#,GRADE)

1. 查询讲授数据库课程的教师
2. 检索学习课程号为C2的学生的学号与成绩
3. 检索学习课程号为C2的学生的学号与姓名
4. 检索选修课程名为Maths的学生学号与姓名
5. 检索选修课程号为C2或C4的学生学号
6. 检索不学C2课的学生姓名与年龄
7. 至少选修了C1和C2课程的学生学号
8. 至少选修了两门以上课程的学生学号