

第一章：数据系统概述

➤ 数据库的四个基本概念

- 数据 (data)
- 数据库 (Database)
- 数据库管理系 (DBMS)
- 数据库系统 (DBS)

数据库系统(DBS)

- 数据库系统是由数据库、数据库管理系统（及其应用开发工具）、应用程序和数据库管理员（DBA）组成的存储、管理、处理和维护数据的系统。

第二章：关系数据库

2.1 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
- 关系的三类完整性
 - 实体完整性
 - 若属性（指一个或一组属性）A是基本关系的主属性，则A不能取空值
 - 参照完整性
 - 用户定义的完整性

➤ 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

■ 空值就是“不知道”或“不存在”或“无意义”的值

例：

选修 (学号, 课程号, 成绩)

“学号、课程号”为主码

“学号”和“课程号”两个属性都不能取空值

外键及参照完整性

➤ 在关系模型中实体及实体间的联系都是用关系来描述的，自然存在着关系与关系间的引用。

[例] 学生实体、专业实体

主码

学生 (学号, 姓名, 性别, 专业号, 年龄)

专业 (专业号, 专业名)

主码

例 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 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)

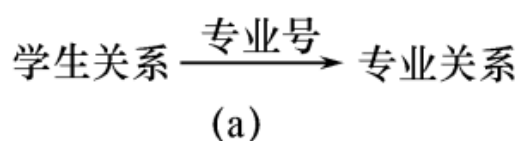
学号	姓名	性别	专业号	年龄	班长
801	张三	女	01	19	802
802	李四	男	01	20	
803	王五	男	01	20	802
804	赵六	女	02	20	805
805	钱七	男	02	19	

- “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
- 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 K_S 相对应, 则称 F 是 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)

[例] 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)



中学生关系的“专业号”与专业关系的主码“专业号”相对应, “专业号”属性是学生关系的外码, 专业关系是被参照关系, 学生关系为参照关系

[例] 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)

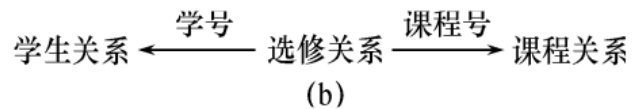
课程 (课程号, 课程名, 学分)

选修 (学号, 课程号, 成绩)

选修关系的“学号”与学生关系的主码“学号”相对应

选修关系的“课程号”与课程关系的主码“课程号”相对应

- “学号”和“课程号”是选修关系的外码
- 学生关系和课程关系均为被参照关系
- 选修关系为参照关系



[例] 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”
- 学生关系既是参照关系也是被参照关系

➤ 关系 R 和 S 不一定是不同的关系

➤ 目标关系 S 的主码 K_S 和参照关系的外码 F 必须定义在同一个（或一组）域上

➤ 外码并不一定要与相应的主码同名

当外码与相应的主码属于不同关系时，往往取相同的名字，以便于识别

➤ 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

用户定义的完整性 (user-defined integrity)

- 用户针对具体的应用环境定义的完整性约束条件，反映某一具体应用所涉及的数据必须满足的语义要求。
 - 如要求学号是8位整数，性别取值为“男”或“女”
 - 成绩取值范围为0 - 100
- 系统支持
 - 实体完整性和参照完整性(关系的两个不变性) 由系统自动支持
 - 系统应提供定义和检验用户定义的完整性机制 (DBMS中的实现)

2.3 关系模式

- 关系模式 (Relation Schema) 是型
- 关系是值
- 关系模式是对关系的描述
 - ▣ 元组集合的结构
 - 属性构成
 - 属性来自的域
 - 属性与域之间的映象关系
 - ▣ 完整性约束条件

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	U 中属性所来自的域
DOM	属性向域的映象集合
F	属性间数据的依赖关系的集合

2.3 关系操作

- 常用的关系操作
 - 查询操作：选择、投影、连接、除、并、差、交、笛卡尔积
 - 选择、投影、并、差、笛卡尔基是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

2.4 关系代数

- 用对关系的运算表达查询
- 运算的三大要素：运算对象、运算符、运算结果
 - 运算对象：集合
 - 运算符：
 - 运算结果：集合

运 算 符		含 义
集合 运算符	\cup	并
	$-$	差
	\cap	交
	\times	笛卡尔积
专门的 关系 运算符	σ	选择
	π	投影
		连接
	\div	除

并 (Union): 所有至少出现在两个关系中之一的元组集合

➤ 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

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

差(Difference): 所有出现在一个关系而不在另一关系中的元组集合

➤ 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

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

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

S	A	B	C
	3	4	5
	7	2	3

R-S

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

S-R

A	B	C
3	4	5

交 (Intersection): 所有同时出现在两个关系中的元组集合

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

R ∩ S		
A	B	C
7	2	3

S	A	B	C
	3	4	5
	7	2	3

广义笛卡尔积 (Extended Cartesian Product): 两个分别为n目和m目的关系R和S的笛卡儿积是一个(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中的一个元组。

$R \times S$ 的度为R与S的度之和, $R \times S$ 的元组个数为R和S的元组个数的乘积

r		s			r × s				
A	B	C	D	E	A	B	C	D	E
x	1	x	6	7	x	1	x	6	7
y	2	y	5	7	x	1	y	5	7
		z	2	3	x	2	z	2	3
					y	2	x	6	7
					y	2	y	5	7
					y	2	z	2	3

(1) $R, t \in R, t[A_i]$

设关系模式为 $R(A_1, A_2, \dots, A_n)$

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) $A, t[A], \overline{A}$

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

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

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

(3) 元组连接 $\widehat{t_r t_s}$

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:

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

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

$$Z_{x1} = \{z_1, z_2, z_3\},$$

➤ x_2 在 R 中的象集

$$Z_{x2} = \{z_2, z_3\},$$

➤ x_3 在 R 中的象集

$$Z_{x3} = \{z_1, z_3\}$$

➤ 选择又称为限制 (Restriction)

➤ 选择运算符的含义

■ 在关系 R 中选择满足给定条件的诸元组

$$\sigma_F(R) = \{t | t \in R \wedge F(t) = \text{'真'}\}$$

■ F : 选择条件, 是一个逻辑表达式, 取值为“真”或“假”

● 基本形式为: $X_1 \theta Y_1$

● θ 表示比较运算符, 它可以是 $>, \geq, <, \leq, =$ 或 $<>$

● X, Y 是属性名、常量、或简单函数

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

- 从 R 中选择出若干属性列组成新的关系

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

A : R 中的属性列

- 投影操作主要是从列的角度进行运算
- 投影之后不仅取消了原关系中的某些列，而且还可能取消某些元组（避免重复行）

R

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

$\pi_{B,C}(R)$

B	C
6	7
2	3
4	3

➤ 连接也称为θ连接

➤ 连接运算的含义

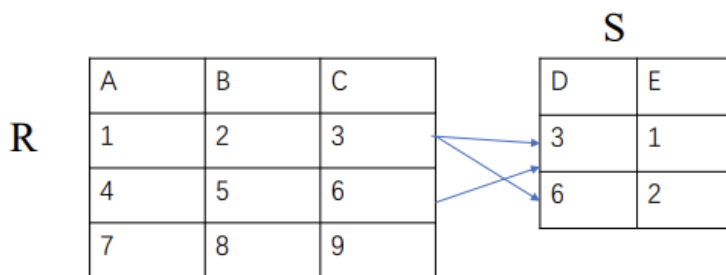
从两个关系的笛卡尔积中选取属性间满足一定条件的元组

$$R \bowtie_{A\theta B} S = \{ \widehat{t_r t_s} \mid t_r \in R \wedge t_s \in S \wedge t_r[A] \theta t_s[B] \}$$

● A 和 B : 分别为 R 和 S 上度数相等且可比的属性组

● θ : 比较运算符

■ 连接运算从 R 和 S 的广义笛卡尔积 $R \times S$ 中选取 R 关系在 A 属性组上的值与 S 关系在 B 属性组上的值满足比较关系 θ 的元组



$$R \bowtie_{B < D} S$$

A	B	C	D	E
1	2	3	3	1
1	2	3	6	2
4	5	6	6	2

➤ 等值连接: θ 为等号时称为等值连接

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

$$R \bowtie_{A=B} S = \{ \widehat{t_r t_s} \mid t_r \in R \wedge t_s \in S \wedge t_r[A] = t_s[B] \}$$

➤ 自然连接 (Natural join)

● 自然连接是一种特殊的等值连接

➤ 两个关系中进行比较的分量必须是相同的属性组

➤ 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

$$R \bowtie S = \{ \widehat{t_r t_s} [U-B] \mid t_r \in R \wedge t_s \in S \wedge t_r[B] = t_s[B] \}$$

R			S	
A	B	C	B	E
a1	b1	5	b1	3
a1	b2	6	b2	7
a2	b3	8	b3	10
a2	b4	12	b3	2
			b2	2

自然连接 $R \bowtie S$

A	B	C	E
a1	b1	5	3
a1	b2	6	7
a2	b3	8	10
a2	b3	8	2

➤ 悬浮元组 (Dangling tuple)

- 两个关系 R 和 S 在做自然连接时，关系 R 中某些元组有可能在 S 中不存在公共属性上值相等的元组，从而造成 R 中这些元组在操作时被舍弃了，这些被舍弃的元组称为悬浮元组。

➤ 外连接 (Outer Join)

- 如果把悬浮元组也保存在结果关系中，而在其他属性上填空值(Null)，就叫做外连接
- 左外连接(LEFT OUTER JOIN或LEFT JOIN)
 - 只保留左边关系 R 中的悬浮元组
- 右外连接(RIGHT OUTER JOIN或RIGHT JOIN)
 - 只保留右边关系 S 中的悬浮元组

R			S	
A	B	C	B	E
a1	b1	5	b1	3
a1	b2	6	b2	7
a2	b3	8	b3	10
a2	b4	12	b3	2
			b5	2

关系 R 和 S 的外连接

A	B	C	E
a1	b1	5	3
a1	b2	6	7
a2	b3	8	10
a2	b3	8	2
a2	b4	12	NULL
NULL	b5	NULL	2

R			S					
A	B	C	B	E				
a1	b1	5	b1	3				
a1	b2	6	b2	7				
a2	b3	8	b3	10				
a2	b4	12	b3	2				
			b5	2				

关系R和S的左外连接				关系R和S的右外连接			
A	B	C	E	A	B	C	E
a1	b1	5	3	a1	b1	5	3
a1	b2	6	7	a1	b2	6	7
a2	b3	8	10	a2	b3	8	10
a2	b3	8	2	a2	b3	8	2
a2	b4	12	NULL	NULL	b5	NULL	2

除运算 (Division)

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的象集 Y_x 包含S在Y上投影的集合, 记作:

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

$$Y_x: x \text{ 在 } R \text{ 中的象集, } x = t_r[X]$$

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

S

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

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

R ÷ S

A
a1

a1的象集为{(b1,c2), (b2,c3), (b2, c1)}

a2的象集为{(b3,c7), (b2,c3)}

a3的象集为{(b4,c6)}

a4的象集为{(b6,c6)}