



数据库系统概论

An Introduction to Database System

第二章 关系数据库(续)

An Introduction to Database System



- 2.1 关系模型概述**
- 2.2 关系数据结构**
- 2.3 关系的完整性**
- 2.4 关系代数**
- 2.5 关系演算**
- 2.6 小结**

2.4 关系代数



- ❖ 概述
- ❖ 传统的集合运算
- ❖ 专门的关系运算



- ❖ 关系代数是一种抽象的查询语言，它用对关系的运算来表达查询的。
- ❖ 包括：运算对象、运算符、运算结果。

表2.4 关系代数运算符

| 运算符 | | 含义 | 运算符 | | 含义 |
|-------|----------|---------------------|--------|--------|------|
| 集合运算符 | \cup | 并 差 交 笛卡尔积 | 比较运算符 | $>$ | 大于 |
| | $-$ | | | \geq | 大于等于 |
| | \cap | | | $<$ | 小于 |
| | \times | | | \leq | 小于等于 |
| | | | | $=$ | 等于 |
| | | | \neq | 不等于 | |



表2.4 关系代数运算符（续）

| 运算符 | 含义 | | 运算符 | 含义 | |
|----------|-----------|----|-------|----------|---|
| 专门的关系运算符 | σ | 选择 | 逻辑运算符 | \neg | 非 |
| | π | 投影 | | \wedge | 与 |
| | \bowtie | 连接 | | \vee | 或 |
| | \div | 除 | | | |

2.4 关系代数



- ❖ 概述
- ❖ 传统的集合运算
- ❖ 专门的关系运算

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

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

并(续)



| <i>R</i> | | |
|----------|----------|----------|
| <i>A</i> | <i>B</i> | <i>C</i> |
| a_1 | b_1 | c_1 |
| a_1 | b_2 | c_2 |
| a_2 | b_2 | c_1 |

| <i>S</i> | | |
|----------|----------|----------|
| <i>A</i> | <i>B</i> | <i>C</i> |
| a_1 | b_2 | c_2 |
| a_1 | b_3 | c_2 |
| a_2 | b_2 | c_1 |

| <i>RUS</i> | | |
|------------|----------|----------|
| <i>A</i> | <i>B</i> | <i>C</i> |
| 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 (Difference)



❖ 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

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

差(续)



| <i>R</i> | | |
|----------|----------|----------|
| <i>A</i> | <i>B</i> | <i>C</i> |
| a_1 | b_1 | c_1 |
| a_1 | b_2 | c_2 |
| a_2 | b_2 | c_1 |

| <i>S</i> | | |
|----------|----------|----------|
| <i>A</i> | <i>B</i> | <i>C</i> |
| a_1 | b_2 | c_2 |
| a_1 | b_3 | c_2 |
| a_2 | b_2 | c_1 |

| <i>R-S</i> | | |
|------------|----------|----------|
| <i>A</i> | <i>B</i> | <i>C</i> |
| a_1 | b_1 | c_1 |

3. 交 (Intersection)



❖ 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

$$R \cap S = \{ t \mid t \in R \wedge t \in S \}$$

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

交 (续)



| 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 |

| $R \cap S$ | | |
|------------|-------|-------|
| A | B | C |
| a_1 | b_2 | c_2 |
| a_2 | b_2 | c_1 |

4. 笛卡尔积 (Cartesian Product)



- ❖ 严格地讲应该是广义的笛卡尔积 (Extended Cartesian Product)
- ❖ R : n 目关系, k_1 个元组
- ❖ S : m 目关系, k_2 个元组
- ❖ $R \times S$
 - 列: ($n+m$) 列元组的集合
 - ⑩ 元组的前 n 列是关系 R 的一个元组
 - ⑩ 后 m 列是关系 S 的一个元组
 - 行: $k_1 \times k_2$ 个元组
 - ⑩ $R \times S = \{t_r t_s \mid t_r \in R \wedge t_s \in S\}$

(续)



| <i>R</i> | | |
|----------|----------|----------|
| <i>A</i> | <i>B</i> | <i>C</i> |
| a_1 | b_1 | c_1 |
| a_1 | b_2 | c_2 |
| a_2 | b_2 | c_1 |

| <i>S</i> | | |
|----------|----------|----------|
| <i>A</i> | <i>B</i> | <i>C</i> |
| a_1 | b_2 | c_2 |
| a_1 | b_3 | c_2 |
| a_2 | b_2 | c_1 |

| <i>RXS</i> | | | | | |
|------------|------------|------------|------------|------------|------------|
| <i>R.A</i> | <i>R.B</i> | <i>R.C</i> | <i>S.A</i> | <i>S.B</i> | <i>S.C</i> |
| 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 |

2.4 关系代数



- ❖ 概述
- ❖ 传统的集合运算
- ❖ 专门的关系运算

2.4.2 专门的关系运算



先引入几个记号

(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。



()

为 m 目关系。

t_s 称为元组的连接。

$t_r t_s$ 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 |

象集举例

❖ x_1 在 R 中的象集

$$Z_{x_1} = \{Z_1, Z_2, Z_3\},$$

❖ x_2 在 R 中的象集

$$Z_{x_2} = \{Z_2, Z_3\},$$

❖ x_3 在 R 中的象集

$$Z_{x_3} = \{Z_1, Z_3\}$$

专门的关系运算(续)



- ❖ 选择
- ❖ 投影
- ❖ 连接
- ❖ 除

专门的关系运算(续)



4) 学生-课程数据库:
学生关系Student、课程关系Course和选修关系SC

Student

| 学号 Sno | 姓名 Sname | 性别 Ssex | 年龄 Sage | 所在系 Sdept |
|-----------|-------------|------------|------------|--------------|
| 200215121 | 李勇 | 男 | 20 | CS |
| 200215122 | 刘晨 | 女 | 19 | IS |
| 200215123 | 王敏 | 女 | 18 | MA |
| 200215125 | 张立 | 男 | 19 | IS |

(a)

An Introduction to Database System

专门的关系运算(续)



Course

| 课程号 Cno | 课程名 Cname | 先行课 Cpno | 学分 Ccredit |
|------------|--------------|-------------|---------------|
| 1 | 数据库 | 5 | 4 |
| 2 | 数学 | | 2 |
| 3 | 信息系统 | 1 | 4 |
| 4 | 操作系统 | 6 | 3 |
| 5 | 数据结构 | 7 | 4 |
| 6 | 数据处理 | | 2 |
| 7 | PASCAL语言 | 6 | 4 |

(b)

An Introduction to Database System

专门的关系运算(续)



SC

| 学号 Sno | 课程号 Cno | 成绩 Grade |
|-----------|------------|-------------|
| 200215121 | 1 | 92 |
| 200215121 | 2 | 85 |
| 200215121 | 3 | 88 |
| 200215122 | 2 | 90 |
| 200215122 | 3 | 80 |

(c)

1. 选择 (Selection)



❖ 1) 选择又称为限制 (Restriction)

❖ 2) 选择运算符的含义

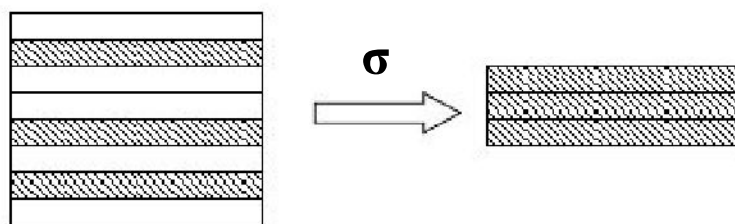
- 在关系 R 中选择满足给定条件的诸元组

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

- F : 选择条件, 是一个逻辑表达式, 基本形式为:

$$X_1 \theta Y_1$$

- ❖ 3) 选择运算是从关系 R 中选取使逻辑表达式 F 为真的元组，是从行的角度进行的运算



选择（续）



[例1] 查询信息系（IS系）全体学生

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

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

结果:

| Sno | Sname | Ssex | Sage | Sdept |
|-----------|-------|------|------|-------|
| 200215122 | 刘晨 | 女 | 19 | IS |
| 200215125 | 张立 | 男 | 19 | IS |

选择（续）



[例2] 查询年龄小于20岁的学生

$\sigma_{\text{Sage} < 20}(\text{Student})$

结果：

| Sno | Sname | Ssex | Sage | Sdept |
|-----------|-------|------|------|-------|
| 200215122 | 刘晨 | 女 | 19 | IS |
| 200215123 | 王敏 | 女 | 18 | MA |
| 200215125 | 张立 | 男 | 19 | IS |

2. 投影 (Projection)



❖ 1) 投影运算符的含义

- 从 R 中选择出若干属性列组成新的关系

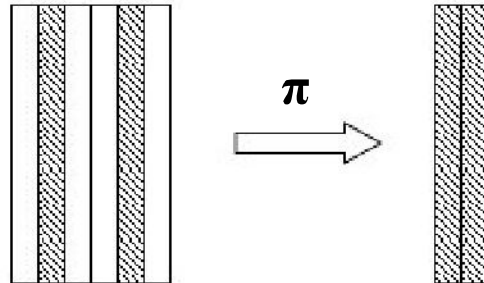
$$\pi_A(R) = \{ t[A] \mid t \in R \}$$

A : R 中的属性列

2. 投影 (Projection)



❖ 2) 投影操作主要是从列的角度进行运算



- 但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）



❖ [例3] 查询学生的姓名和所在系

即求**Student**关系上学生姓名和所在系两个属性上的投影

$\pi_{\text{Sname, Sdept}}(\text{Student})$

结果:

投影（续）



| Sname | Sdept |
|-------|-------|
| 李勇 | CS |
| 刘晨 | IS |
| 王敏 | MA |
| 张立 | IS |



[例4] 查询学生关系Student中都有哪些系

$\pi_{\text{Sdept}}(\text{Student})$

结果:

| Sdept |
|-------|
| CS |
| IS |
| MA |

3. 连接 (Join)



❖ 1) 连接也称为 θ 连接

❖ 2) 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

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

➤ **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**×**S**中选取 (**R**关系) 在**A**属性组上的值与 (**S**关系) 在**B**属性组上值满足比较关系 θ 的元组

$$R \underset{A\theta B}{\bowtie} S = \sigma_{r[A] \theta s[B]} (R \times S)$$



❖ 3) 两类常用连接运算

■ 等值连接 (equijoin)

➤ 什么是等值连接

θ 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，即等值连接为：

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



■ 自然连接 (Natural join)

⑩ 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

⑩ 自然连接的含义

R 和 S 具有相同的属性组 B

$$R \bowtie S = \{ t_r \in R \wedge t_s \in S \wedge t_r[B] = t_s[B] \}$$



| R | | | S | |
|---|---|---|---|---|
| A | B | C | D | E |
| 1 | 2 | 3 | 3 | 1 |
| 4 | 5 | 6 | 6 | 2 |
| 7 | 8 | 9 | | |

$R \bowtie S$
 $B < D$

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

$$R \bowtie_{A \theta B} S = \sigma_{r[A] \theta s[B]} (R \times S)$$



■ 等值连接

| R | | S | | |
|----------|---|----------|----|---|
| A | B | C | D | E |
| α | 1 | α | 10 | a |
| β | 2 | β | 10 | a |
| | | β | 20 | b |
| | | γ | 10 | b |

$$R \bowtie_{A=C} S$$

$R \times S$

| 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 |

$\sigma_{A=C}(R \times S)$

| A | B | C | D | E |
|----------|---|----------|----|---|
| α | 1 | α | 10 | a |
| β | 2 | β | 20 | a |
| β | 2 | β | 20 | b |

$$R \bowtie_{A=C} S$$

$$= \sigma_{A=C}(R \times S)$$

■ 1. $R \times S$

■ 2. $\sigma_{A=C}(R \times S)$

$$R \bowtie S$$



自然连接运算：例

R和S有公共属性B, D

结果中只有一个B, D

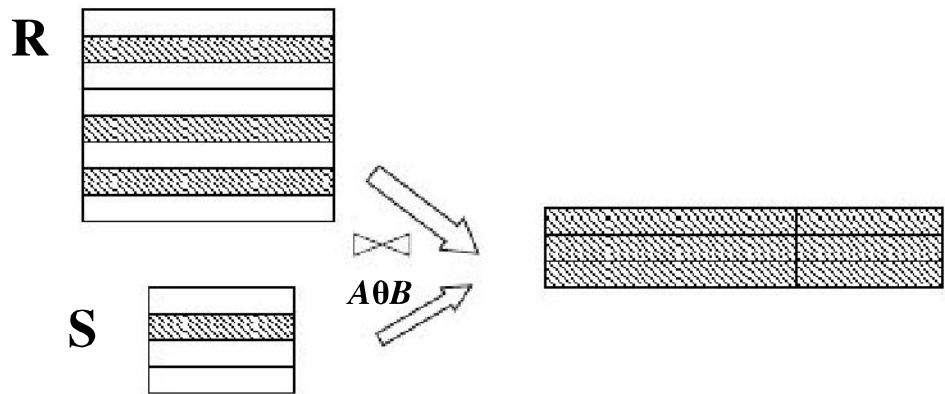
| R | | | |
|----------|---|----------|---|
| A | B | C | D |
| α | 1 | α | a |
| β | 2 | γ | a |
| γ | 4 | β | b |
| α | 1 | γ | a |
| δ | 2 | β | b |

| S | | |
|---|---|------------|
| B | D | E |
| 1 | a | α |
| 3 | a | β |
| 1 | a | γ |
| 2 | b | δ |
| 3 | b | ϵ |

| R \bowtie S | | | | |
|---------------|---|----------|---|----------|
| A | B | C | D | E |
| α | 1 | α | a | α |
| α | 1 | α | a | γ |
| α | 1 | γ | a | α |
| α | 1 | γ | a | γ |
| δ | 2 | β | b | δ |

- 当R与S无公共属性时， $R \bowtie S = R \times S$
- 当R与S全是公共属性时， $R \bowtie S = R \cap S$

❖ 4) 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。

连接(续)



❖ [例5]关系 R 和关系 S 如下所示:

| 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 |

连接(续)



一般连接 $R \bowtie_{C < E} S$ 的结果如下:

$$R \bowtie_{C < E} S$$

| <i>A</i> | <i>R.B</i> | <i>C</i> | <i>S.B</i> | <i>E</i> |
|----------|------------|----------|------------|----------|
| 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 |

连接(续)



等值连接 $R \bowtie_{R.B=S.B} S$ 的结果如下:

| 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 |

连接(续)



自然连接 $R \bowtie S$ 的结果如下:

| 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 |



❖ 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**)。(悬浮元组)

❖ 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**)。

连接(续)



下图是例5中关系*R*和关系*S*的外连接

| <i>A</i> | <i>B</i> | <i>C</i> | <i>E</i> |
|----------|----------|----------|----------|
| 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

连接(续)



图(b)是例5中关系 R 和关系 S 的左外连接,图(c)是右外连接

| 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

4. 除 (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:

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

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



■ 除运算过程有三个步骤:

- (1) 求出 $\Pi_Y(S)$
- (2) 考察 R 的元组在 X 上的每个值 x , 求出像集 Y_x
- (3) 判断 $Y_x \supseteq \Pi_Y(S)$? 是, 则 x 为 $R \div S$ 关系的一个元组, 否则不是。

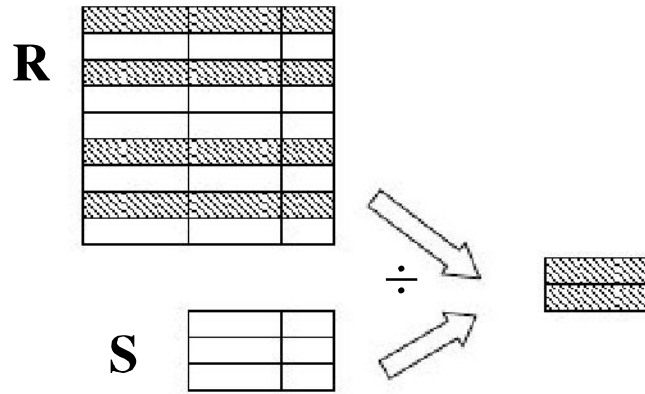
■ 可以用其他运算表示除运算:

$$\begin{aligned} & R(X,Y) \div S(Y,Z) \\ &= \Pi_X(R) - \Pi_X(\Pi_X(R) \times \Pi_Y(S) - R) \end{aligned}$$

除(续)



❖ 2) 除操作是同时从行和列角度进行运算



除(续)



[例6] 设关系 R 、 S 分别为下图的(a)和(b)， $R \div S$ 的结果为图(c)

| R | | |
|-------|-------|-------|
| A | B | C |
| a_1 | b_1 | c_2 |
| a_2 | b_3 | c_7 |
| a_3 | b_4 | c_6 |
| a_1 | b_2 | c_3 |
| a_4 | b_6 | c_6 |
| a_2 | b_2 | c_3 |
| a_1 | b_2 | c_1 |

(a)

| S | | |
|-------|-------|-------|
| B | C | D |
| b_1 | c_2 | d_1 |
| b_2 | c_1 | d_1 |
| b_2 | c_3 | d_2 |

(b)

| $R \div S$ |
|------------|
| A |
| a_1 |

(c)



- ❖ 在关系R中，A可以取四个值{a1, a2, a3, a4}
 - a_1 的象集为 $\{(b_1, c_2), (b_2, c_3), (b_2, c_1)\}$
 - a_2 的象集为 $\{(b_3, c_7), (b_2, c_3)\}$
 - a_3 的象集为 $\{(b_4, c_6)\}$
 - a_4 的象集为 $\{(b_6, c_6)\}$
- ❖ S在(B, C)上的投影为
 $\{(b_1, c_2), (b_2, c_1), (b_2, c_3)\}$
- ❖ 只有 a_1 的象集包含了S在(B, C)属性组上的投影
所以 $R \div S = \{a_1\}$

5. 综合举例



以学生-课程数据库为例 (P56)

[例7] 查询至少选修1号课程和3号课程的学生号码

首先建立一个临时关系 K :

| <u>Cno</u> |
|------------|
| 1 |
| 3 |

然后求: $\Pi_{Sno,Cno}(SC) \div K$

综合举例(续)



❖ 例 7续 $\pi_{Sno,Cno}(SC)$

200215121象集{1, 2, 3}

200215122象集{2, 3}

$K=\{1, 3\}$

于是:

$\pi_{Sno,Cno}(SC) \div K=\{200215121\}$

| Sno | Cno |
|-----------|-----|
| 200215121 | 1 |
| 200215121 | 2 |
| 200215121 | 3 |
| 200215122 | 2 |
| 200215122 | 3 |



[例 8] 查询选修了2号课程的学生的学号。

$$\begin{aligned} & \pi_{Sno} (\sigma_{Cno='2'} (SC)) \\ = & \{ 200215121, 200215122 \} \end{aligned}$$



R (X, Y) ÷ **S (Y)** = **R ÷ S**

| A | B | C | D |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 7 | 8 | 5 | 6 |
| 7 | 8 | 3 | 4 |
| 1 | 2 | 5 | 6 |
| 1 | 2 | 4 | 2 |

①

| C | D |
|---|---|
| 3 | 4 |
| 5 | 6 |

综合举例(续)



[例9] 查询至少选修了一门其直接先行课为5号课程的学生姓名

或 $\pi_{Sname}(\sigma_{Cpno='5'}(\text{Course} \bowtie \sim \bowtie \sim \text{Ident}))$

或 $\pi_{Sname}(\sigma_{Cpno='5'}(\text{Course}) \bowtie \bowtie_{Sno, Sname}(\text{Student}))$

$\pi_{Sname}(\pi_{Sno}(\sigma_{Cpno='5'}(\text{Course})) \bowtie_{Sno, Sname} \pi_{Sno, Sname}(\text{Student}))$

综合举例(续)



[例10] 查询选修了全部课程的学生号码和姓名。

$$\pi_{Sno, Cno} (SC) \div \pi_{Cno} (Course) \bowtie \pi_{Sno, Sname} (Student)$$



关系代数运算

- 关系代数运算

并、差、交、笛卡尔积、投影、选择、连接、除

- 基本运算

并、差、笛卡尔积、投影、选择

- 交、连接、除

可以用5种基本运算来表达

引进它们并不增加语言的能力，但可以简化表达

$$R \bowtie_{A \theta B} S = \sigma_{r[A] \theta s[B]} (R \times S)$$

- 可以用其他运算表示除运算：

$$\begin{aligned} R(X,Y) \div S(Y,Z) \\ = \Pi_X(R) - \Pi_X(\Pi_X(R) \times \Pi_Y(S) - R) \end{aligned}$$



❖ 关系代数表达式

- 关系代数运算经有限次复合后形成的式子

❖ 典型关系代数语言

▪ ISBL (Information System Base Language)

- 由IBM United Kingdom研究中心研制
- 用于PRTV (Peterlee Relational Test Vehicle) 实验系统



2.1 关系模型概述

2.2 关系数据结构

2.3 关系的完整性

2.4 关系代数

2.5 关系演算

2.6 小结

2.5 关系演算



❖ 关系演算

以数理逻辑中的谓词演算为基础

❖ 按谓词变元不同 进行分类

1. 元组关系演算:

以元组变量作为谓词变元的基本对象

元组关系演算语言 ALPHA

2. 域关系演算:

以域变量作为谓词变元的基本对象

域关系演算语言 QBE

2.5.1 元组关系演算语言ALPHA



❖ 由E.F.Codd提出

INGRES所用的QUEL语言是参照ALPHA语言研制的

❖ 语句

检索语句

■ GET

更新语句

■ PUT, HOLD, UPDATE, DELETE, DROP

一、检索操作



❖ 语句格式:

GET 工作空间名 [(定额)] (表达式1)
[: 操作条件][DOWN/UP 表达式2]

定额: 规定检索的元组个数

⑩格式: 数字

表达式1: 指定语句的操作对象

⑩格式:

关系名| 关系名. 属性名| 元组变量. 属性名| 集函数
[, ...]

操作条件: 将操作结果限定在满足条件的元组中

⑩格式: 逻辑表达式

表达式2: 指定排序方式

⑩格式: 关系名. 属性名| 元组变量. 属性名
[, ...]

一、检索操作



(1) 简单检索

GET 工作空间名 (表达式1)

[例1] 查询所有被选修的课程号码。

GET W (SC.Cno)

[例2] 查询所有学生的数据。

GET W (Student)

(2) 限定的检索



格式

GET 工作空间名 (表达式1) : 操作条件

[例3]查询信息系(IS)中年龄小于20岁的学生的学号和年龄

GET W (Student.Sno, Student.Sage):

Student.Sdept='IS' ^ Student.Sage<20

(3) 带排序的检索



格式

```
GET 工作空间名 (表达式1) [操作条件]  
    DOWN/UP _____
```

[例4]查询计算机科学系(CS)学生的学号、年龄，结果按年龄降序排序

```
GET W (Student.Sno, Student.Sage):  
    Student.Sdept='CS' DOWN Student.Sage
```


(4) 带定额的检索



格式

```
GET 工作空间名 (定额) (表达式1)  
[: 操作条件] [DOWN/UP 表达式2]
```

[例5] 取出一个信息系学生的学号。

```
GET W (1) (Student.Sno):  
Student.Sdept='IS'
```

[例6] 查询信息系年龄最大的三个学生的学号及其年龄，结果按年龄降序排序。

```
GET W (3) (Student.Sno, Student.Sage):  
Student.Sdept='IS' DOWN Student.Sage
```

(5) 用元组变量的检索



- ❖ 元组变量的含义
 - 表示可以在某一关系范围内变化（也称为范围变量Range Variable）
- ❖ 元组变量的用途
 - ① 简化关系名：设一个较短名字的元组变量来代替较长的关系名。
 - ② 操作条件中使用量词时必须用元组变量。
- ❖ 定义元组变量
 - 格式：RANGE 关系名 变量名
 - 一个关系可以设多个元组变量

(6) 用存在量词的检索



❖ 操作条件中使用量词时必须用元组变量

[例8] 查询选修2号课程的学生名字。

RANGE SC X

GET W (Student.Sname):

$\exists X(X.Sno=Student.Sno \wedge X.Cno='2')$

[例9] 查询选修了这样课程的学生学号，其直接先行课是6号课程。

RANGE Course CX

GET W (SC.Sno):

$\exists CX(CX.Cno=SC.Cno \wedge CX.Pcno='6')$

用存在量词的检索(续)



[例10]查询至少选修一门其先行课为6号课程的学生名字

RANGE Course **CX**

SC **SCX**

GET W (Student.Sname): \exists **SCX** (SCX.Sno=Student.Sno \wedge
 \exists **CX** (CX.Cno=SCX.Cno \wedge CX.Pcno='6'))

前束范式形式:

GET W (Student.Sname):

\exists SCX \exists CX (SCX.Sno=Student.Sno \wedge
CX.Cno=SCX.Cno \wedge CX.Pcno='6')

(7) 带有多个关系的表达式的检索



[例11] 查询成绩为90分以上的学生名字与课程名字。

RANGE SC SCX

GET W(Student.Sname, Course.Cname):

\exists SCX (SCX.Grade \geq 90 \wedge

SCX.Sno=Student.Sno \wedge

Course.Cno=SCX.Cno)

(8) 用全称量词的检索



[例12] 查询不选1号课程的学生名字

RANGE SC SCX

GET W (Student.Sname):

\forall SCX (SCX.Sno \neq Student.Sno \vee SCX.Cno \neq '1')

用存在量词表示:

RANGE SC SCX

GET W (Student.Sname):

$\neg \exists$ SCX (SCX.Sno=Student.Sno \wedge SCX.Cno='1')

(9) 用两种量词的检索



[例13] 查询选修了全部课程的学生姓名。

RANGE Course CX

SC SCX

GET W (Student.Sname):

$\forall CX \exists SCX (SCX.Sno=Student.Sno \wedge$
 $SCX.Cno=CX.Cno)$

(10) 用蕴涵 (Implication) 的检索



[例14] 查询最少选修了200215122学生所选课程的学生学号

RANGE Couse CX
SC SCX
SC SCY

GET W (Student.Sno): $\forall CX(\exists SCX$
(SCX.Sno='200215122' \wedge SCX.Cno=CX.Cno)
 $\Rightarrow \exists SCY(SCY.Sno=Student.Sno \wedge$
SCY.Cno= CX.Cno))

(11) 聚集函数



常用聚集函数 (Aggregation function) 或内部函数 (Build-in function)

| 函数名 | 功能 |
|-------|-------|
| COUNT | 对元组计数 |
| TOTAL | 求总和 |
| MAX | 求最大值 |
| MIN | 求最小值 |
| AVG | 求平均值 |

关系演算中的聚集函数

An Introduction to Database System



[例15] 查询学生所在系的数目。

```
GET W ( COUNT(Student.Sdept) )
```

COUNT函数在计数时会自动排除重复值。

[例16] 查询信息系学生的平均年龄

```
GET W (AVG(Student.Sage):
```

```
Student.Sdept='IS' )
```

二、更新操作



- (1) 修改操作
- (2) 插入操作
- (3) 删除操作

(1) 修改操作步骤



① 用HOLD语句将要修改的元组从数据库中读到工作空间中

HOLD 工作空间名 (表达式1) [: 操作条件]

HOLD语句是带上并发控制的GET语句

② 用宿主语言修改工作空间中元组的属性

③ 用UPDATE语句将修改后的元组送回数据库中

UPDATE 工作空间名

修改操作(续)



[例17] 把200215121学生从计算机科学系转到信息系。

```
HOLD W (Student.Sno, Student.Sdept):
```

```
    Student.Sno='200215121'
```

(从Student关系中读出95007学生的数据)

```
MOVE 'IS' TO W.Sdept
```

(用宿主语言进行修改)

```
UPDATE W
```

(把修改后的元组送回Student关系)

(2) 插入操作



步骤

- ① 用宿主语言在工作空间中建立新元组
- ② 用PUT语句把该元组存入指定关系中

PUT 工作空间名 (关系名)

PUT语句只对一个关系操作，关系演算中的聚集函数



[例18] 学校新开设了一门2学分的课程“计算机组织与结构”，其课程号为8，直接先行课为6号课程。插入该课程元组

```
MOVE '8' TO W.Cno
```

```
MOVE '计算机组织与结构' TO W.Cname
```

```
MOVE '6' TO W.Cpno
```

```
MOVE '2' TO W.Ccredit
```

```
PUT W (Course)
```

(3) 删除操作



步骤

- ① 用**HOLD**语句把要删除的元组从数据库中读到工作空间中
- ② 用**DELETE**语句删除该元组

DELETE 工作空间名

删除操作(续)



[例19] 200215125学生因故退学，删除该学生元组

```
HOLD W (Student): Student.Sno='200215125'
```

```
DELETE W
```

删除操作(续)



[例20] 将学号200215121改为200215126

```
HOLD W (Student): Student.Sno='200215121'
```

```
DELETE W
```

```
MOVE '200215126' TO W.Sno
```

```
MOVE '李勇' TO W.Sname
```

```
MOVE '男' TO W.Ssex
```

```
MOVE '20' TO W.Sage
```

```
MOVE 'CS' TO W.Sdept
```

```
PUT W (Student)
```



[例21] 删除全部学生

```
HOLD W (Student)
```

```
DELETE W
```

为保证参照完整性，删除Student中元组时相应地要删除SC中的元组

```
HOLD W (SC)
```

```
DELETE W
```



❖ 检索操作 GET

GET 工作空间名 [(定额)] (表达式1)
[: 操作条件] [DOWN/UP 表达式2]

❖ 插入操作

- 建立新元组--PUT

❖ 修改操作

- HOLD--修改--UPDATE

❖ 删除操作

- HOLD--DELETE

2.5 关系演算



❖ 2.5.1 元组关系演算语言ALPHA

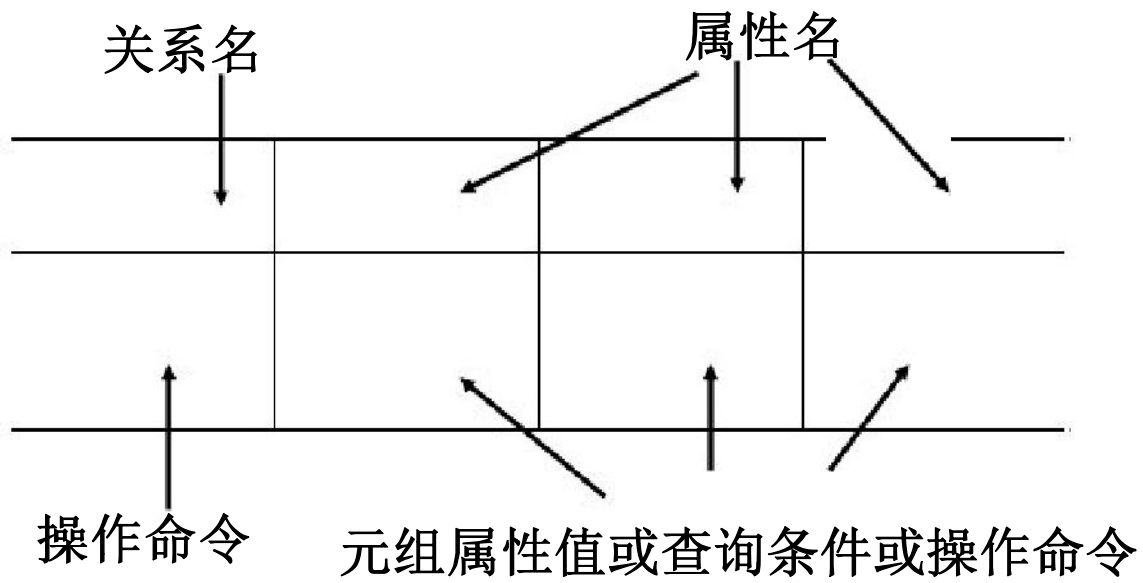
❖ 2.5.2 域关系演算语言QBE

2.5.2 域关系演算语言QBE



- ❖ 一种典型的域关系演算语言
 - 由M.M.Zloof提出
 - 以元组变量的分量即域变量作为谓词变元的基本对象
- ❖ QBE: Query By Example
 - 基于屏幕表格的查询语言
 - 查询要求: 以填写表格的方式构造查询
 - 用示例元素(域变量)来表示查询结果可能的情况
 - 查询结果: 以表格形式显示

QBE操作框架



一、检索操作



1. 简单查询

[例1]求信息系全体学生的姓名

操作步骤为：

- (1) 用户提出要求；
- (2) 屏幕显示空白表格；

| | | | | | |
|--|--|--|--|--|--|
| | | | | | |
| | | | | | |

简单查询（续）



(3) 用户在最左边一栏输入要查询的关系名Student;

| | | | | | |
|---------|--|--|--|--|--|
| Student | | | | | |
| | | | | | |

(4) 系统显示该关系的属性名

| Student | Sno | Sname | Ssex | Sage | Sdept |
|---------|-----|-------|------|------|-------|
| | | | | | |

简单查询（续）



(5) 用户在上面构造查询要求

| Student | Sno | Sname | Ssex | Sage | Sdept |
|---------|-----|-------|------|------|-------|
| | | P.李勇 | | | IS |

■ 李勇是示例元素，即域变量

(6) 屏幕显示查询结果

| Student | Sno | Sname | Ssex | Sage | Sdept |
|---------|-----|----------|------|------|-------|
| | | 李勇 张立 | | | IS |

构造查询的几个要素



- ❖ **示例元素** 即域变量 一定要加下划线

示例元素是这个域中可能的一个值，它不必是查询结果中的元素

- ❖ **打印操作符P.** 实际上是显示

- ❖ **查询条件**

可使用比较运算符 $>$, \geq , $<$, \leq , $=$ 和 \neq

其中 $=$ 可以省略

简单查询（续）



[例2] 查询全体学生的全部数据

| Student | Sno | Sname | Ssex | Sage | Sdept |
|---------|--------------------|-------------|------------|-------------|-------------|
| | <u>P.200215121</u> | <u>P.李勇</u> | <u>P.男</u> | <u>P.20</u> | <u>P.CS</u> |

简单查询（续）



显示全部数据也可以简单地把P.操作符作用在关系名上。

| Student | Sno | Sname | Ssex | Sage | Sdept |
|---------|-----|-------|------|------|-------|
| P. | | | | | |

2. 条件查询



[例3] 求年龄大于19岁的学生的学号

| Student | Sno | Sname | Ssex | Sage | Sdept |
|---------|--------------------|-------|------|------|-------|
| | <u>P.200215121</u> | | | >19 | |

条件查询（与条件）



[例4] 求计算机科学系年龄大于**19**岁的学生的学号。

方法(1): 把两个条件写在同一行上

| Student | Sno | Sname | Ssex | Sage | Sdept |
|---------|--------------------|-------|------|------|-------|
| | <u>P.200215121</u> | | | >19 | CS |

条件查询（与条件）



方法(2): 把两个条件写在不同行上, 但使用相同的示例元素值

| Student | Sno | Sname | Ssex | Sage | Sdept |
|---------|------------------------------------------|-------|------|------|-------|
| | <u>P.200215121</u> <u>P.200215121</u> | | | >19 | CS |

条件查询（与条件）



[例5] 查询既选修了1号课程又选修了2号课程的学生学号。

| Sc | Sno | Cno | Grade |
|----|--------------------|-----|-------|
| | <u>P.200215121</u> | 1 | |
| | <u>P.200215121</u> | 2 | |

条件查询（或条件）



[例6]查询计算机科学系或者年龄大于19岁的学生的学号。

| Student | Sno | Sname | Ssex | Sage | Sdept |
|---------|------------------------------------------|-------|------|------|-------|
| | <u>P.200215121</u> <u>P.200215122</u> | | | >19 | CS |

条件查询（多表连接）



[例7] 查询选修1号课程的学生姓名。

| Student | Sno | Sname | Ssex | Sage | Sdept |
|---------|-----------|-------|------|------|-------|
| | 200215121 | P.李勇 | | | |
| | | | | | |
| | | | | | |

注意：示例元素Sno是连接属性，其值在两个表中要相同。

条件查询（非条件）



[例8] 查询未选修1号课程的学生姓名

| Student | Sno | Sname | Ssex | Sage | Sdept |
|---------|-----------|-------|------|------|-------|
| | 200215121 | P.李勇 | | | |
| | | | | | |
| | | | | | |

思

选修1号课程的情况为假

条件查询（续）



[例9] 查询有两个人以上选修的课程号。

| Sc | Sno | Cno | Grade |
|----|----------------------------------------|------------------------|-------|
| | <u>200215121</u> - <u>200215121</u> | <u>P.1</u> <u>1</u> | |

思路：查询这样的课程1，它不仅被200215121选修而且也被另一个学生（-200215121）选修了

3. 聚集函数



常用聚集函数:

| 函数名 | 功能 |
|-----|-------|
| CNT | 对元组计数 |
| SUM | 求总和 |
| AVG | 求平均值 |
| MAX | 求最大值 |
| MIN | 求最小值 |

QBE中的聚集函数

聚集函数（续）



[例10] 查询信息系学生的平均年龄。

| Student | Sno | Sname | Ssex | Sage | Sdept |
|---------|-----|-------|------|-----------|-------|
| | | | | P.AVG.ALL | IS |

4.对查询结果排序



❖ 升序排序:

- 对查询结果按某个属性值的升序排序，只需在相应列中填入“**AO.**”

❖ 降序排序:

- 按降序排序则填“**DO.**”

❖ 多列排序:

- 如果按多列排序，用“**AO(i).**”或“**DO(i).**”表示，其中*i*为排序的优先级，*i*值越小，优先级越高

对查询结果排序（续）



[例11] 查全体男生的姓名，要求查询结果按所在系升序排序，对相同系的学生按年龄降序排序。

| Student | Sno | Sname | Ssex | Sage | Sdept |
|---------|-----|-------|------|----------|----------|
| | | P.李勇 | 男 | DO (2) . | AO (1) . |

二、更新操作



1.修改操作

[例12] 把200215121学生的年龄改为18岁。

方法(1)：将操作符“U.”放在值上

| Student | Sno | Sname | Ssex | Sage | Sdept |
|---------|-----------|-------|------|------|-------|
| | 200215121 | | | U.18 | |

修改操作(续)



方法(2): 将操作符“U.”放在关系上

| Student | Sno | Sname | Ssex | Sage | Sdept |
|---------|-----------|-------|------|------|-------|
| U. | 200215121 | | | 18 | |

码200215121标明要修改的元组。

“U.”标明所在的行是修改后的新值。

由于主码是不能修改的，所以系统不会混淆要修改的属性。

修改操作(续)



[例13] 把200215121学生的年龄增加1岁

| Student | Sno | Sname | Ssex | Sage | Sdept |
|---------|------------------------|-------|------|--------------------------|-------|
| U. | 200215121 200215121 | | | <u>17</u> <u>17+1</u> | |

操作涉及表达式，必须将操作符“U.”放在关系上

修改操作(续)



[例14] 将计算机系所有学生的年龄都增加1岁

| Student | Sno | Sname | Ssex | Sage | Sdept |
|---------|--------------------------------------|-------|------|--------------------------|-------|
| U. | <u>200215122</u> <u>200215122</u> | | | <u>18</u> <u>18+1</u> | CS |

2.插入操作



[例15] 把信息系女生200215701，姓名张三，年龄17岁存入数据库中。

| Student | Sno | Sname | Ssex | Sage | Sdept |
|---------|-----------|-------|------|------|-------|
| 1. | 200215701 | 张三 | 女 | 17 | IS |

3. 删除操作



[例17] 删除学生200215089

| Student | Sno | Sname | Ssex | Sage | Sdept |
|---------|-----------|-------|------|------|-------|
| D. | 200215089 | | | | |

为保证参照完整性，删除200215089学生前，先删除200215089学生选修的全部课程

| Sc | Sno | Cno | Grade |
|----|-----------|-----|-------|
| D. | 200215089 | | |



- 2.1 关系模型概述
- 2.2 关系数据结构
- 2.3 关系的完整性
- 2.4 关系代数
- 2.5 关系演算
- 2.6 小结

2.6 小结



- ❖ 关系数据库系统是目前使用最广泛的数据库系统
- ❖ 关系数据库系统与非关系数据库系统的区别：
 - 关系系统只有“表”这一种数据结构；
 - 非关系数据库系统还有其他数据结构，以及对这些数据结构的操作



❖ 关系数据结构

- 关系
 - ⑩ 域
 - ⑩ 笛卡尔积
 - ⑩ 关系
 - ∞ 关系，属性，元组
 - ∞ 候选码，主码，主属性
 - ∞ 基本关系的性质
- 关系模式
- 关系数据库



❖ 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



❖ 关系的完整性约束

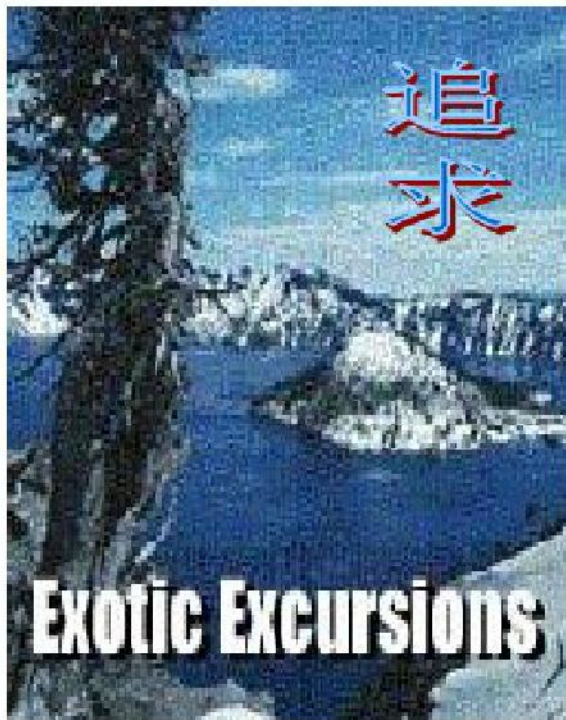
- 实体完整性
- 参照完整性
 - 外码
- 用户定义的完整性



❖ 关系数据语言

- 关系代数语言
- 关系演算语言
 - 元组关系演算语言 ALPHA
 - 域关系演算语言 QBE

下课了。。。。



休息一会儿。。。。



An Introduction to Database System