# 第二节 关系数据库模型



# 知识点



#### 基本概念

#### 关系、记录

一个表格就是-个关系。表中的 每一行称为一个 记录,也称为 "元组"。

也可由多个属性组合

#### 域

数据项的取值 范围。

#### 候选码

关系中唯一标 识一个记录的 属性或属性组。

#### 主码

一个关系中候 选码可能有多 个,选定一个 作为主码。

记录

#### 学生基本信息表

学号 性别 学院 出生年月 姓名 20140123 1992, 7 男 计算机学院 李明 关系 王芳 20140125 女 外文学院 1990, 8 候选码可由 主码 单个属性构成 域:{1-12}

关系数 据 库 的 概 念

# 关系数据库的概念

#### 关系的性质

表中的关键字属性必须有唯一的名字。

行的顺序 不是影响因 素。 任意两个 记录不能完 全相同。

表格不能 嵌套。 每一列中 分量是同一 类型的数据

学生表

性 出生年月 别 姓名 学号 学院 专业 年 月 可以 男 20140123 李明 计算机学院 软件工程 1991 6 只要主 李明 20140124 男 计算机学院 软件工程 1991 6 键不同 王芳 20140125 7 女 外文学院 英国文学 1990 可以调 7 陈红 20140125 男 数学院 应用数学 1992 换顺序 • • • 不能相同 来自同一个域

不允许 嵌套

#### 关系的完整性

#### 实体完整性

所有主码对应的属性值不能为空

#### 参照完整性

定义外码与主码之间的引用规则,即外码必须是另一个表的主码的有效值

#### 用户自定义 完整性

针对某一具体 关系数据库的 约束条件

#### 学生表

| 姓名   | 学号           | 学院   |  |
|------|--------------|------|--|
| 李明   | 201401<br>23 | 地理学院 |  |
| 王芳   | 201401<br>25 | 计算机  |  |
| •••  | ··!          |      |  |
| 不能为空 |              |      |  |

#### 课程表

| 课程号           | 课程<br>名 | 学分 |
|---------------|---------|----|
| 1             | C#      | 3  |
| 2             | GIS     | 4  |
| ," <b>₹</b> - |         |    |

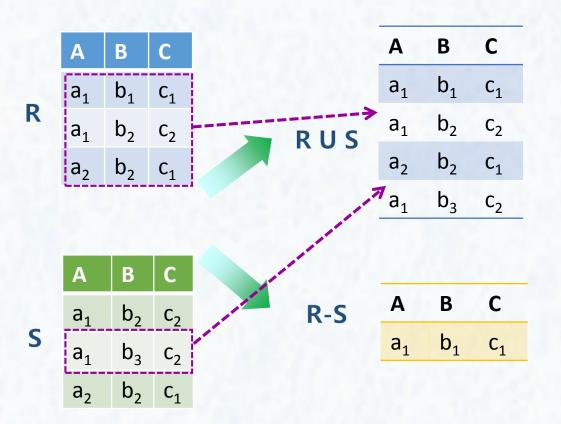
主码

#### 成绩表

| 4 4          | 4 49-9 4-9-4 |      |  |  |  |
|--------------|--------------|------|--|--|--|
| 学号           | 课程<br>号      | 成绩   |  |  |  |
| 20140<br>123 | 1            | 80   |  |  |  |
| 20140<br>125 | 2            | 95   |  |  |  |
| , .,, ´´     |              | ,    |  |  |  |
| 外码           |              | 0-10 |  |  |  |

#### 传统的关系运算

关系代数是一种**抽象的查询语言**,用关系的运算来表达查询,是作为研究 关系数据语言的数学工具。



#### 并(Union)

由属于R<mark>或</mark>属于 S的记录组成。

#### 差(Difference)

由属于R<mark>而</mark>不属于S 的所有记录组成。

#### 传统的关系运算

列:m=3 行:k1=3

 $\begin{array}{c|cccc} A & B & C \\ \hline a_1 & b_1 & c_1 \\ a_1 & b_2 & c_2 \\ \hline a_2 & b_2 & c_1 \\ \hline \end{array}$ 

RNS

| Α              | В              | С     |
|----------------|----------------|-------|
| $a_1$          | b <sub>2</sub> | $C_2$ |
| a <sub>2</sub> | b <sub>2</sub> | $C_1$ |

交 (Intersection)

由既属于R又属于 S的记录组成

两个关系都有n个属性,相应的属性取自同一个域。

行: k2=3A B C  $a_1 b_2 c_2$   $a_1 b_3 c_2$   $a_2 b_2 c_1$ 

列:n=3

**RB** RC SA SB SC RA  $b_1$  $C_1$  $a_1$  $b_2$  $C_2$  $b_1$  $b_3$  $C_1$  $a_1$ R×S  $C_2$  $b_1$  $C_1$  $a_2$  $b_2$  $C_1$  $b_2$  $b_2$  $a_1$  $C_2$  $a_1$  $C_2$ 

笛卡尔积

列: (m+n)列 行: k1×k2个元组

> m+n=6 k1\*k2=9

#### 专门的关系运算

#### 选择 (Selection)

选择又称为限制,在关系R中选择满足给定条件的诸元组构成新的关系:

$$\sigma_F(R) = \{ t \mid t \in R \land F(t) = '真' \}$$

F:选择条件,是一个逻辑表达式。

#### 学生表

| 姓名 | 学号       | 所在的系 | 性别 |
|----|----------|------|----|
| 李明 | 20140123 | 测绘工程 | 男  |
| 王芳 | 20140125 | GIS  | 女  |
| 陈强 | 20140126 | GIS  | 男  |
| 张立 | 20140127 | 测绘工程 | 男  |

## 查询信息系(GIS系)全体学生

 $\sigma$ 所在的系 = 'GIS' (学生表)

| 姓名 | 学号       | 所在的系 | 性别 |
|----|----------|------|----|
| 王芳 | 20140125 | GIS  | 女  |
| 陈强 | 20140126 | GIS  | 男  |

● 选择是从行的角度进行的运算

#### 专门的关系运算

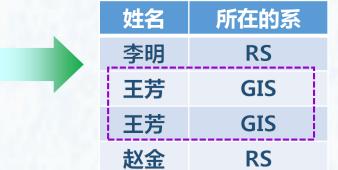
#### 投影 (Projection)

从关系中选择出<mark>若干属性列组成新的关系</mark>  $\Pi_{\Lambda}(R) = \{ t[A] \mid t \in R, A : R$ 中的属性列  $\}$ 

#### 学生表

| 姓名 | 学号       | 所在的<br>系 | 性别 |
|----|----------|----------|----|
| 李明 | 20140123 | RS       | 男  |
| 王芳 | 20140125 | GIS      | 女  |
| 王芳 | 20140126 | GIS      | 男  |
| 赵金 | 20140127 | RS       | 男  |
|    |          |          |    |

#### 查询学生的姓名和所在系



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<mark>还可能</mark>取消某些元组(避免重复行)

#### 专门的关系运算

#### 等值连接 (Join)

从关系R与关系S的广义笛卡尔积中选取R、S属性值相等的那些元组。

$$\mathbf{R} \overset{\triangleright}{\triangleright} \mathbf{S} = \{ \widehat{\mathbf{t}_r \, \mathbf{t}_s} | \ \mathbf{t}_r \in \mathbf{R} \wedge \mathbf{t}_s \in \mathbf{S} \wedge \mathbf{t}_r[\mathbf{A}] = \mathbf{t}_s[\mathbf{B}] \ \}$$

|       | R                     |    |
|-------|-----------------------|----|
| Α     | В                     | С  |
| $a_1$ | $b_1$                 | 5  |
| $a_1$ | <b>b</b> <sub>2</sub> | 6  |
| $a_2$ | $b_3$                 | 8  |
| $a_2$ | $\mathbf{b}_4$        | 12 |

| В                     | E  |  |  |
|-----------------------|----|--|--|
| $b_1$                 | 3  |  |  |
| <b>b</b> <sub>2</sub> | 7  |  |  |
| <b>b</b> <sub>3</sub> | 10 |  |  |
| $\mathbf{b}_3$        | 2  |  |  |
| <b>b</b> <sub>5</sub> | 6  |  |  |

一般的连接操作是从行的角度进行运算。但还需要取消重复列。

| R |    |    | S |
|---|----|----|---|
| R | B: | =S | B |

| Α     | R.B                   | C | S.B                   | Е  |
|-------|-----------------------|---|-----------------------|----|
| $a_1$ | $b_1$                 | 5 | $b_1$                 | 3  |
| $a_1$ | <b>b</b> <sub>2</sub> | 6 | <b>b</b> <sub>2</sub> | 7  |
| $a_2$ | <b>b</b> <sub>3</sub> | 8 | <b>b</b> <sub>3</sub> | 10 |
| $a_2$ | $b_3$                 | 8 | $b_3$                 | 2  |
|       |                       |   |                       |    |

重复列

#### 数据库查找程序

通常,查找一个数据库记录是一个<mark>连续的过程。计算机从头开始</mark> 进行查询,直到<mark>找到需要的项目为止。</mark>

#### 查找ID= 050792的地块信息

| 依次对    |               | 地块ID   | 地块类型 | 调查人 | 调查时间    |
|--------|---------------|--------|------|-----|---------|
|        | <del>-7</del> | 010682 | 耕地   | 陈强  | 2012.10 |
| 对      | +             | 010683 | 建筑用地 | 李菲  | 2012.10 |
| 比      | +             | 010684 | 林地   | 陈强  | 2012.10 |
| ,<br>杏 |               |        | •••  | ••• | •••     |
| 查询     |               | 050792 | 建筑用地 | 王强  | 2012.11 |
|        | 1             | 050793 | 建筑用地 | 高磊  | 2013.10 |
|        |               |        |      |     |         |

方法可行,如果 记录较多,需要很长 的时间才能找到信息, 效率相当低。

查找成功

#### 二分法查找程序

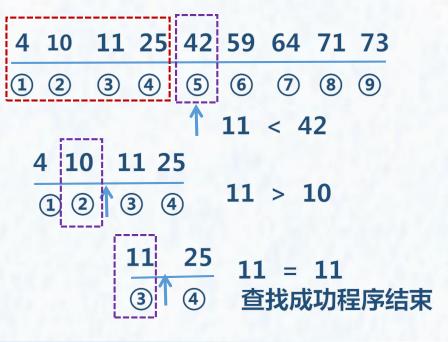
如果已经对数据进行了排序,数据查找效率就可得到提高。

二分法查找程序是计算机快速查找数据的典型例子。

#### 算法步骤:

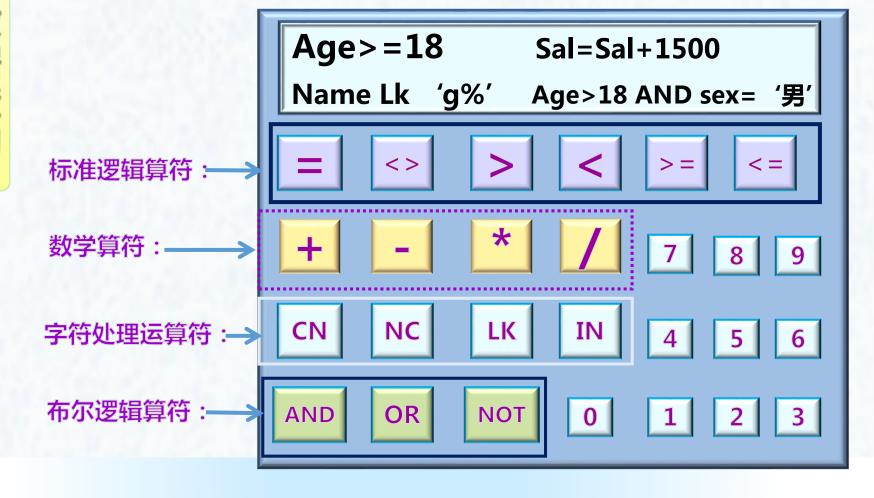
- 把资料组分为两部分
- 记录在前半部分还是 在后半部分?
- 放弃不需要的一半;
- 重复上面的操作。

#### 举例: 查找数字11



#### 主要的算符

● 数据库查询中所用到的算符包括:



#### 结构化查询语言

为实现数据库查询而专门设计的一种标准化语言。应用广泛,可以进行独立的、分散的数据库连接。



# SQL(Structural Query Language)

查询语言

广州市人口? ———— 1600 广州市人均GDP?—— 4900

#### 全国各城市人口统计表

| 城市<br>代码 | 城市<br>名称 | 城市<br>人口数 |
|----------|----------|-----------|
| 14536    | 北京       | 1300      |
| 14661    | 广州       | 1600      |
| 14663    | 上海       | 城市        |
|          |          | 代码        |

全国各城市 GDP总量

| 城市<br>代码 | 城市<br>名称 | GDP<br>总量 |
|----------|----------|-----------|
| 14536    | 北京       | 5600      |
| 14661    | 上海       | 6300      |
| 14663    | 广州       | 4900      |
| •••      | •••      | •••       |

#### SQL数据库查询的句法

SQL查询的基础句法是:

SELECT {数据字段}

FROM {表格}

WHERE <字段 > Expression <值 >

#### · 土地信息表

| 土地    | 面积  | 所有者     | 是否<br>登记 |
|-------|-----|---------|----------|
| 14536 | 350 | S.Engel | 是        |
| 14661 | 399 | G.Leeds | 否        |
| 14663 | 399 | S.Engel | 是        |

#### 例如:

SELECT 土地代码, 面积, 所有者

FROM 土地信息表

WHERE 面积 > 360



#### 查询结果

| 土地代码  | 面积  | 所有者            |
|-------|-----|----------------|
| 14661 | 399 | <b>G.Leeds</b> |
| 14663 | 399 | S.Engel        |

### SQL连接查询示例

选择所有满足下列要求的土地:价值大于25000,其上有多于或等于5个房产:

SELECT 编号, 价值, 房产个数

FROM 土地表, 位置表

查询条件

联接条件

WHERE (价值 > 25000 AND 房产个数 >=5) (

AND (土地表.编号 = 位置表.编号) -

表 位置表

#### 土地表

| 编号    | 拥有者        | 价值    | 用途           |
|-------|------------|-------|--------------|
| 34651 | D.Smith    | 25000 | Residential  |
| 34652 | C.Lawrence | 22500 | Industrial   |
| 34653 | D.Lewsley  | 32600 | Agricultural |
| 34654 | S.Bishop   | 24300 | Residential  |
| 34655 | C.Lawrence | 88500 | Commercial   |

| 编号    | 地址          | 面积   | 房产个数 |
|-------|-------------|------|------|
| 34651 | George St.  | 1540 | 12   |
| 34652 | Adam St.    | 2200 | 03   |
| 34653 | Henry St.   | 9500 | 01   |
| 34654 | Park Ave.   | 0950 | 03   |
| 34655 | Station Rd. | 1450 | 10   |

输出的 结果

| 编号    | 价值    | 房产个数 |
|-------|-------|------|
| 34651 | 25000 | 12   |
| 34655 | 88500 | 10   |



# 谢谢大家!

