

第2章 关系数据库

2.1 关系数据库简介

2.2 基本概念

2.3 关系代数

2.4 关系演算

关系数据库简介

- 采用关系模型作为数据组织方式的系统称为关系数据库系统(RDBS)。
- 1970年E. F. Codd提出关系数据模型 “A Relational Model of Data for Large Shared Data Banks”
《Communication of the ACM》1970
- 目前数据库厂商推出的DBMS几乎都支持关系模型。

关系模型的基本特点

- 关系模型具有严格的数学理论基础。
- 关系模型的数据结构相对单一，在关系模型中，实体及实体间的联系均用“关系”来表示，这也是关系模型能够取代其它数据模型的原因之一。
- 关系在用户的观点中即是一张“二维表”。

关系模型的基本特点

- 关系模型通常提供查询、增加、删除、修改等数据操作，其中查询主要用于处理数据的各种检索需求，是数据操作的主要研究对象。
- 关系查询操作按照其理论基础的不同，一般分为关系代数和关系演算两类，前者以“集合论”为基础，后者以“一阶谓词逻辑”为基础。

关系模型的数据结构

- 在用户观点下，关系模型中数据的逻辑结构是“二维表”，由“行”和“列”组成。

The diagram illustrates the components of a 2D table structure. A central table is shown with columns for '学号' (Student ID), '姓名' (Name), '年龄' (Age), '性别' (Gender), '系' (Department), and '年级' (Grade). Annotations include: '主码' (Primary Key) pointing to the '学号' column; '域' (Domain) pointing to the '年龄' column with a list of integers '1,2,3,4,5'; '属性名' (Attribute Name) pointing to the '年级' column; '元组(行)' (Tuple/Row) pointing to the rows of the table; and '属性列' (Attribute Column) pointing to the columns of the table. A bracket on the left side of the table is labeled '关系' (Relation).

| 学号 | 姓名 | 年龄 | 性别 | 系 | 年级 |
|-------|-----|-----|-----|-----|-----|
| 82021 | 张平 | 18 | 男 | 计算机 | 2 |
| 82022 | 王英 | 17 | 女 | 计算机 | 1 |
| ... | ... | ... | ... | ... | ... |
| 82030 | 李强 | 17 | 男 | 物理 | 1 |

关系模型的数据结构

- 关系 (Relation)
 - 一个关系对应通常说的一张表 (Table)
- 元组 (Tuple)
 - 表中的一行 (Row) 即为一个元组
- 属性 (Attribute)
 - 表中的一列即为一个属性，给每一个属性起一个名称即属性名

关系模型的数据结构

- 域(Domain)
 - 属性的取值范围
- 分量
 - 元组中的一个属性值
- 码(Key)
 - 表中的某个属性组，它可以唯一确定一个元组
- 关系模式
 - 对关系的描述，相当于“表头”部分
 - 表示形式：关系名(属性1, 属性2, ..., 属性n)
 - 例：学生(学号, 姓名, 年龄, 性别, 系, 年级)

关系模型的数据结构

- ⑩ 关系必须是规范化的，满足一定的规范条件
- ⑩ 最基本的规范条件：关系的每一个分量必须是一个不可分的数据项，不允许“表中套表”
- ⑩ 例：图中工资和扣除是可分的数据项，因此不符合关系规范化要求

| 职工号 | 姓名 | 职 称 | 工 资 | | | 扣 除 | | 实 发 |
|-------|-----|-----|------|------|-----|-----|-----|------|
| | | | 基 本 | 津 贴 | 职 务 | 房 租 | 水 电 | |
| 86051 | 陈 平 | 讲 师 | 1305 | 1200 | 50 | 160 | 112 | 2283 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

图 一个工资表(表中套表)实例

关系模型的数据结构

术语对比

| 关系术语 | 一般表格的术语 |
|-------|---------------|
| 关系名 | 表名 |
| 关系模式 | 表头(表格的描述) |
| 关系 | (一张)二维表 |
| 元组 | 记录或行 |
| 属性 | 列 |
| 属性名 | 列名 |
| 属性值 | 列值 |
| 分量 | 一条记录中的一个列值 |
| 非规范关系 | 表中有表(大表中嵌有小表) |

关系型数据库发展历程

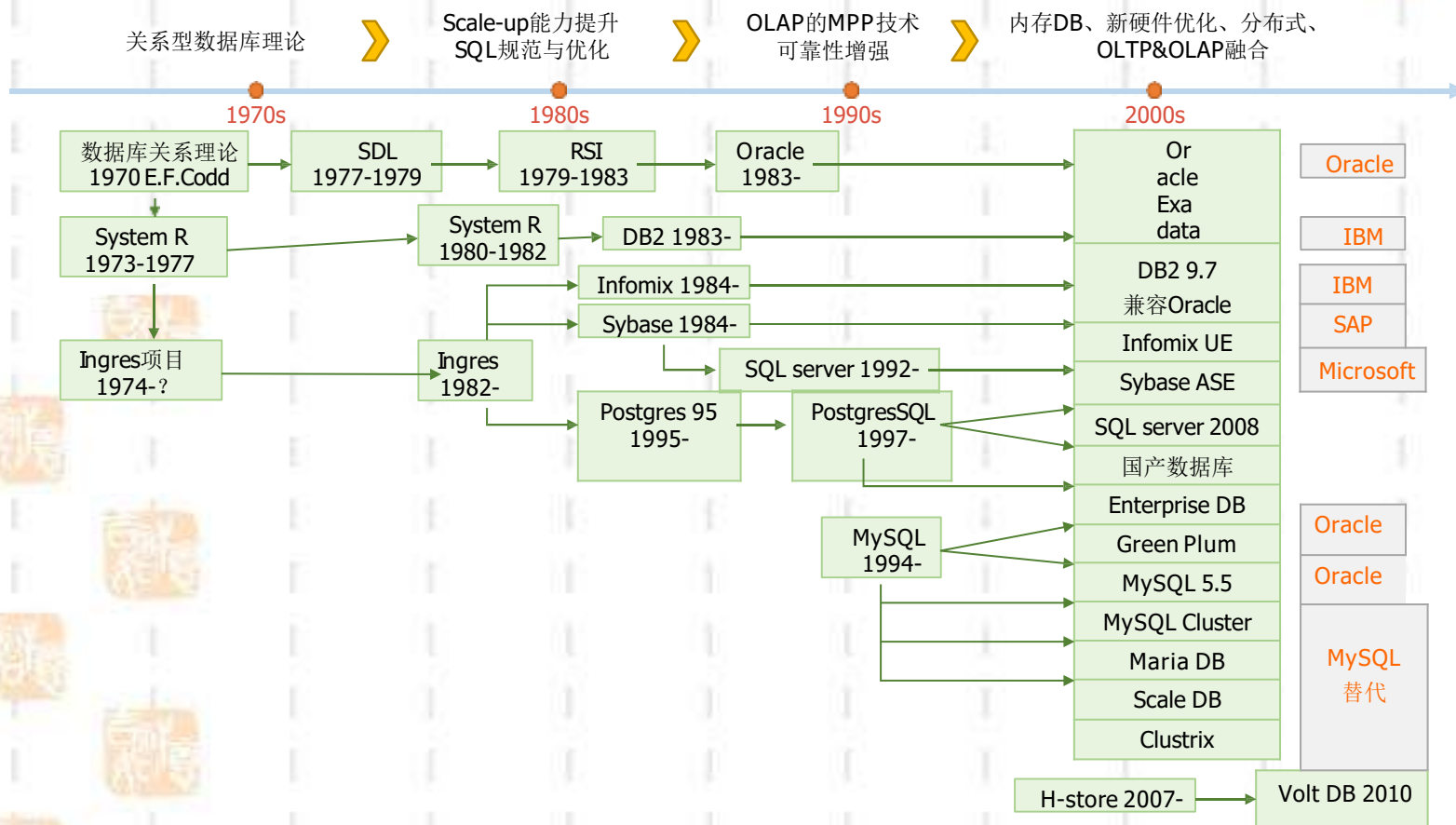
■ 早期代表系统

- **System R**: 由**IBM**研制
- **INGRES**: 由加州**Berkeley**分校研制

■ 目前主流的关系数据库管理系统软件产品包括:

- **Oracle, IBM DB2 UDB, Informix, Sybase, MS SQL Server, Access, Dbase, Foxbase, Foxpro, MySQL**

关系型数据库发展历程



第2章 关系数据库

2.1 关系数据库简介

2.2 基本概念

2.3 关系代数

2.4 关系演算

关系的定义

- **定义2-1:** 域(domain)是一组值的集合, 同一个域中的所有值均应具有相同的数据类型。
- **定义2-2:** 域 D_1, D_2, \dots, D_n 上的笛卡儿积是一个集合:
$$D_1 \times D_2 \times \dots \times D_n = \{(d_1, d_2, \dots, d_n) \mid d_i \in D_i, 1 \leq i \leq n\}$$
其中允许 $D_i = D_j$ 且 $i \neq j$, 我们将该集合中的每一个元素 (d_1, d_2, \dots, d_n) 称为一个元组(tuple), 元组中的每一个值 d_i 称为一个分量(component), 有 n 个分量的元组称为 n 元组。

我们用 $|D_i|$ 表示域 D_i 中值的个数, 即 D_i 的基数, 设 $|D_i| = m_i$, 则 $D_1 \times D_2 \times \dots \times D_n$ 的基数 M 为: $M = \prod m_i, 1 \leq i \leq n$ 。

关系的定义

例2-1：给出如下的5个域：

$D_1 = \{02055002, 02055016, 02055039, 02055057, 02055065\}$

$D_2 = \{\text{计算机01}, \text{计算机02}, \text{计算机03}\}$

$D_3 = \{\text{辛欣}, \text{李楠}, \text{吴晓涛}, \text{陈思宇}, \text{雷鸣}\}$

$D_4 = \{\text{男}, \text{女}\}$

$D_5 = \{17-231, 18-506, 18-510, 17-236, 18-508\}$

则这五个域的笛卡儿积 $D_1 \times D_2 \times D_3 \times D_4 \times D_5$ 将会有 $M = 5 \times 3 \times 5 \times 2 \times 5 = 750$ 个元组，图2-1所示的即是该笛卡儿积集合的一个子集。

| S# | CLASS | NAME | SEX | DORM |
|----------|-------|------|-----|--------|
| 02055002 | 计算机01 | 辛欣 | 女 | 17-231 |
| 02055016 | 计算机01 | 李楠 | 男 | 18-506 |
| 02055039 | 计算机02 | 吴晓涛 | 男 | 18-510 |
| 02055057 | 计算机02 | 陈思宇 | 女 | 17-236 |
| 02055065 | 计算机03 | 雷鸣 | 男 | 18-508 |

关系的定义

- **定义2-3:** $D_1 \times D_2 \times \dots \times D_n$ 上的任意一个子集均是定义在域 D_1, D_2, \dots, D_n 上的一个关系(relation), 记为 R 。
- 由 n 个域构成的关系通常称为 n 元关系, 关系中的每个元素即是这个关系的元组。
- 由于笛卡儿积集合即是一个二维表, 所以关系自然也可以表示为二维表。
- 表的每行是一个元组, 每列对应一个域。
- 由于无限关系无法在数据库中表示, 因此关系模型中的关系也特指有限的元组集合。

关系的定义

- 由于域可以相同，为了加以区分，必须为每列起一个名字，称为属性(Attribute)。
- **定义2-4：** 为关系的每个列所起的名字称为关系的属性(attribute)，一般可表示为A，B，C...。
- n目关系必有n个属性，表的任意一列对应一个属性，属性的名称称为“属性(列)名”，具体元组的属性取值称为“属性(列)值”。

关系的定义

- **定义2-5:** 如果一个关系中的某个属性或属性集能够唯一地确定一个元组，则称该属性（集）是这个关系上的超键（Super key, SK）；如果将超键中的任一属性去掉后剩余的属性集不能唯一标识一个元组，则称该属性集是关系上的候选键（Candidate key, CK）；若一个关系中有多个候选键，则可从中选择一个作为关系的主键（Primary key, PK）。
- 一般情况下，如不加特别说明，键即指主键。

关系的定义

- **定义2-6:** 如果关系R中的某个属性集是另外一个关系S的候选键，那么称该属性集为关系R上的（来自S的）外键（foreign key, FK）。
- 具有一定联系的不同关系间依靠外键建立彼此的关联，这是关系模型数据结构的重要组成部分。

关系模式

- **定义2.7:** 关系模式 (relation schema) 是对关系的型的描述, 可以表示为:

$$R(U, D, \text{DOM}, I, F)$$

- 其中, U 是 R 的属性集合 $\{A_1, A_2, \dots, A_n\}$, D 是属性的取值范围, 即域的集合 $\{D_1, D_2, \dots, D_m\}$, DOM 是 U 到 D 的映射集合, I 是完整性约束规则集, F 是函数依赖集合
- 习惯上将关系简记为 $R(A_1 / D_1, A_2 / D_2, \dots, A_n / D_n)$ 或者 $R(A_1, A_2, \dots, A_n)$

关系的特性

➤列是同质：即每一列中的分量来自同一域，是同一类型的数据

Student

| S# | Sname | Ssex | Sage | D# | Sclass |
|----------|-------|------|------|-----|--------|
| 98030101 | 张三 | 男 | 20 | 03 | 980301 |
| 98030102 | 张四 | 女 | 20 | 计算机 | 980301 |
| 98030103 | 张五 | 男 | 19 | 03 | 980301 |
| 98040201 | 王三 | 01 | 20 | 04 | 980402 |
| 98040202 | 王四 | 男 | 21 | 控制 | 980402 |
| 98040203 | 王五 | 02 | 19 | 04 | 980402 |

出现列不同质的现象

关系的特性

- 不同的列可来自同一个域，称其中的每一列为一个属性，不同的属性要给予不同的属性名。
- 关系模式 $R(A_1:D_1, A_2:D_2, \dots, A_n:D_n)$ 中， $A_i (i = 1, \dots, n)$ 必须是不同的，而 $D_i (i = 1, \dots, n)$ 可以是相同的
- 例，我们定义一个域为 **Person** = 所有男人、女人和儿童的集合 = {李基，张鹏，王芳，刘玉，李健，张睿，张峰}，则下述“家庭”关系的三个列将来自同一个域 **Person**，因此需要不同的属性名“丈夫”“妻子”“子女”以示区分。

家庭

| 丈夫 | 妻子 | 子女 |
|----|----|----|
| 李基 | 王方 | 李健 |
| 张鹏 | 刘玉 | 张睿 |
| 张鹏 | 刘玉 | 张峰 |

关系的特性

- 列位置互换性：区分哪一列是靠列名
- 行位置互换性：区分哪一行是靠某一或某几列的值(关键字/键字/码字)
- 关系是以内容(名字或值)来区分的，而不是属性在关系的位置来区分
- 如下面两个关系是完全相同的关系

家庭

| 丈夫 | 妻子 | 子女 |
|----|----|----|
| 李基 | 王方 | 李键 |
| 张鹏 | 刘玉 | 张睿 |
| 张鹏 | 刘玉 | 张峰 |

家庭

| 丈夫 | 子女 | 妻子 |
|----|----|----|
| 李基 | 李键 | 王方 |
| 张鹏 | 张峰 | 刘玉 |
| 张鹏 | 张睿 | 刘玉 |

关系的特性

- 理论上，**关系**的任意两个元组不能完全相同。（集合的要求：集合内不能有相同的两个元素）；现实应用中，**表(Table)**可能并不完全遵守此特性。
- 元组相同是指两个元组的每个分量都相同。

Student

| S# | Sname | Ssex | Sage | D# | Sclass |
|----------|-------|------|------|----|--------|
| 98030101 | 张三 | 男 | 20 | 03 | 980301 |
| 98030102 | 张四 | 女 | 20 | 03 | 980301 |
| 98030103 | 张五 | 男 | 19 | 03 | 980301 |
| 98040201 | 王三 | 男 | 20 | 04 | 980402 |
| 98040202 | 王四 | 男 | 21 | 04 | 980402 |
| 98040203 | 王五 | 女 | 19 | 04 | 980402 |
| 98030103 | 张五 | 男 | 19 | 03 | 980301 |

在同一个关系中存在相同的元组，去掉其中的重复元组

关系的特性

➤属性不可再分特性: 又被称为关系第一范式1NF

Students

| sid | lname | fname | class | telephone |
|-----|-------|--------|-------|-----------|
| 1 | Jones | Allan | 2 | 555-1234 |
| 2 | Smith | John | 3 | 555-4321 |
| 3 | Brown | Harry | 2 | 555-1122 |
| 5 | White | Edward | 3 | 555-3344 |

符合第一范式(Table)

Students

| sid | name | | class | telephone | enrollment | |
|-----|-------|--------|-------|-----------|------------|-------|
| | lname | fname | | | cno | major |
| 1 | Jones | Allan | 2 | 555-1234 | 101 | No |
| | | | | | 108 | Yes |
| 2 | Smith | John | 3 | 555-4321 | 105 | No |
| 3 | Brown | Harry | 2 | 555-1122 | 101 | Yes |
| | | | | | 108 | No |
| 4 | White | Edward | 3 | 555-3344 | 102 | No |
| | | | | | 105 | No |

Head: structured type

不符合第一范式
(Not Table)

Value: structured value
collection of values

关系的特性

➤属性不可再分特性(续)

家庭

| 丈夫 | 妻子 | 子女 |
|----|----|----|
| 李基 | 王方 | 李键 |
| 张鹏 | 刘玉 | 张睿 |
| 张鹏 | 刘玉 | 张峰 |

符合第一范式(Table)

家庭

| 丈夫 | 妻子 | 子女 | |
|----|----|-----|-----|
| | | 第一个 | 第二个 |
| 李基 | 王芳 | 李健 | |
| 张鹏 | 刘玉 | 张睿 | 张峰 |

不符合第一范式
(Not Table)

关系模式与关系

- 关系可看作是关系模式在某一时刻的状态或内容，即，关系模式是型，关系是值。
- 关系是“动态的”，关系模式是“静态的”。

关系完整性

- 关系模型的完整性(integrity)规则是为了保证数据的正确性、有效性和相容性而在关系上施加的约束条件。
- 关系模型三类完整性：
 - 实体完整性
 - 参照完整性
 - 用户自定义完整性

实体完整性

- ❑ 关系的主码中的属性值不能为空值；
- ❑ 空值：不知道或无意义的值；
- ❑ 意义：关系中的元组对应到现实世界相互之间可区分的一个个个体，这些个体是通过主码来唯一标识的；若主码为空，则出现不可标识的个体，这是不容许的。

主码

| Student | | | | | |
|----------|-------|------|------|----|--------|
| S# | Sname | Ssex | Sage | D# | Sclass |
| 98030101 | 张三 | 男 | 20 | 03 | 980301 |
| 98030102 | 张四 | 女 | 20 | 03 | 980301 |
| X | 张五 | 男 | 19 | 03 | 980301 |
| 98040201 | 王三 | 男 | 20 | 04 | 980402 |
| 98040202 | 王四 | 男 | 21 | 04 | 980402 |
| X | 王五 | 女 | 19 | 04 | 980402 |

实体完整性

➤ 空值的含义

- ❑ 空值**NULL**：不知道、不存在或无意义的值；
- ❑ 在进行关系操作时，有时关系中的某属性值在当前是填不上的，比如档案中有“生日不详”、“下落不明”、“日程尚待公布”等，这时就需要空值来代表这种情况。关系模型中用‘?’表征。
- ❑ 数据库中有了空值，会影响许多方面，如影响聚集函数运算的正确性，不能参与算术、比较或逻辑运算等。
- ❑ 例如：“**3 + ?**”结果是多少呢？ “**3 * ?**”结果是多少呢？ “**? and (A=A)**”结果又是多少呢？
- ❑ 再例如，一个班有**30**名同学，如所有同学都有成绩，则可求出平均成绩；如果有一个同学没有成绩，怎样参与平均成绩的计算呢，是当作**0**，还是当作**100**呢？还是不考虑他呢？
- ❑ 有空值的时候是需要特殊处理的，要特别注意。

参照完整性

- ❑ 如果关系R1的外码Fk与关系R2的主码Pk相对应，则R1中的每一个元组的Fk值或者等于R2中某个元组的Pk值，或者为空值
- ❑ 意义：如果关系R1的某个元组t1参照了关系R2的某个元组t2，则t2必须存在。
- ❑ 例如关系Student在D#上的取值有两种可能：
 - ✓ 空值，表示该学生尚未分到任何系中
 - ✓ 若非空值，则必须是Dept关系中某个元组的D#值，表示该学生不可能分到一个不存在的系中

R1----Student
外码Fk----D#

Student

| S# | Sname | Ssex | Sage | D# | Sclass |
|----------|-------|------|------|----|--------|
| 98030101 | 张三 | 男 | 20 | 03 | 980301 |
| 98030102 | 张四 | 女 | 20 | 03 | 980301 |
| 98030103 | 张五 | 男 | 19 | 03 | 980301 |
| 98040201 | 王三 | 男 | 20 | 04 | 980402 |
| 98040202 | 王四 | 男 | 21 | 04 | 980402 |
| 98040203 | 王五 | 女 | 19 | 04 | 980402 |
| 98050206 | 孙六 | 男 | 21 | 05 | 980502 |
| 98060207 | 吴七 | 女 | 20 | | 980602 |

R2----Dept
主码Pk----D#

Dept

| D# | Dname | Dean |
|----|-------|------|
| 01 | 机电 | 李三 |
| 02 | 能源 | 李四 |
| 03 | 计算机 | 李五 |
| 04 | 自动控制 | 李六 |

用户自定义完整性

- 用户针对具体的应用环境定义的完整性约束条件。
- 如S#要求是10位整数，其中前四位为年度，当前年度与他们的差必须在4以内。
- 再如：

要求名字在5个
汉字字符之内

| Student | | | | | |
|----------|-------------|------|------|----|--------|
| S# | Sname | Ssex | Sage | D# | Sclass |
| 98030101 | 张三 | 男 | 20 | 03 | 980301 |
| 98030102 | 张四 | 女 | 20 | 03 | 980301 |
| 98030103 | 张五 | 男 | 19 | 03 | 980301 |
| 98040201 | 王三 | 男 | 20 | 04 | 980402 |
| 98040202 | 王四 | 男 | 21 | 04 | 980402 |
| 98040203 | 王五 | 女 | 19 | 04 | 980402 |
| 98040204 | 詹姆斯. 卡梅隆 | 男 | 20 | 04 | 980402 |
| 98040206 | 王六 | 01 | 21 | 04 | 980402 |
| 98040207 | 王七 | 女 | 200 | 04 | 980402 |

年龄在[12, 35]
之间

性别只能是
“男”或
“女”

颠覆完整性
的示例

DBMS对关系完整性的支持

➤ 实体完整性和参照完整性由**DBMS**系统自动支持。

➤ **DBMS**系统通常提供了如下机制：

(1) 它使用户可以自行定义有关的完整性约束条件。

(2) 当有更新操作发生时，**DBMS**将自动按照完整性约束条件检验更新操作的正确性，即是否符合用户自定义的完整性。

关系模型

- 关系模型(Relation Model)由一组相互联系的关系组成，是用二维表结构来表示实体与实体间的联系的模型
- 关系模型由关系数据结构、关系操作集合、关系完整性约束三部分组成，如图所示：

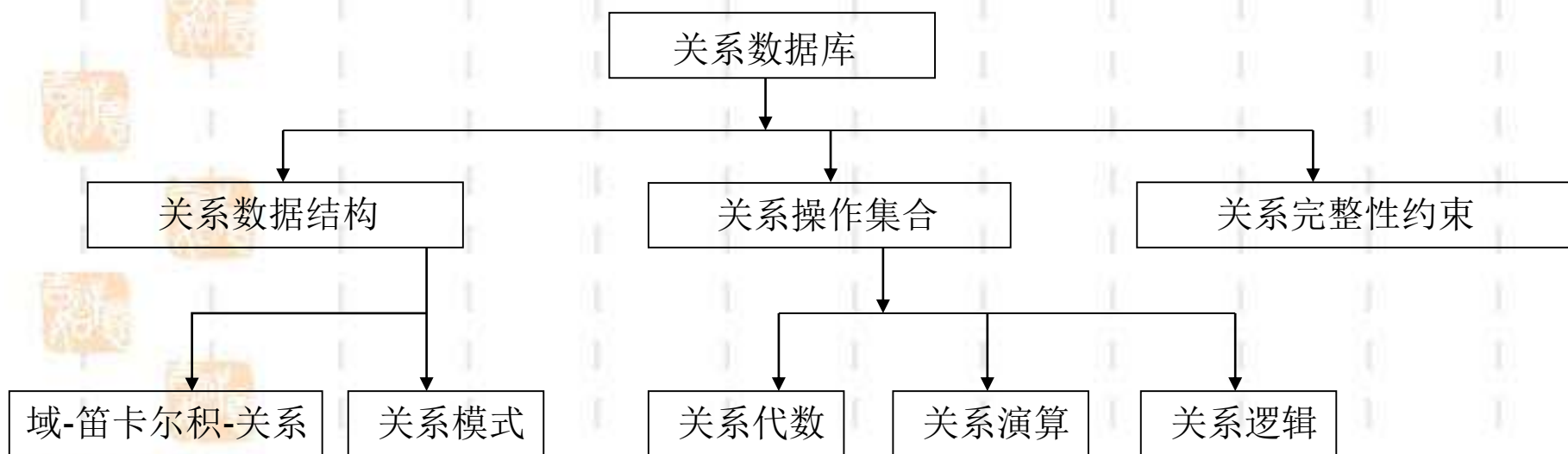


图 关系模型的组成部分

关系模型

- 关系操作集合是指关系模型提供的一组完备的高级关系运算，这些关系运算支持对数据库的各种操作。关系运算通常分成关系代数和关系演算。
- 为简化分析，有时候也使用简化的关系模型，如图所示：

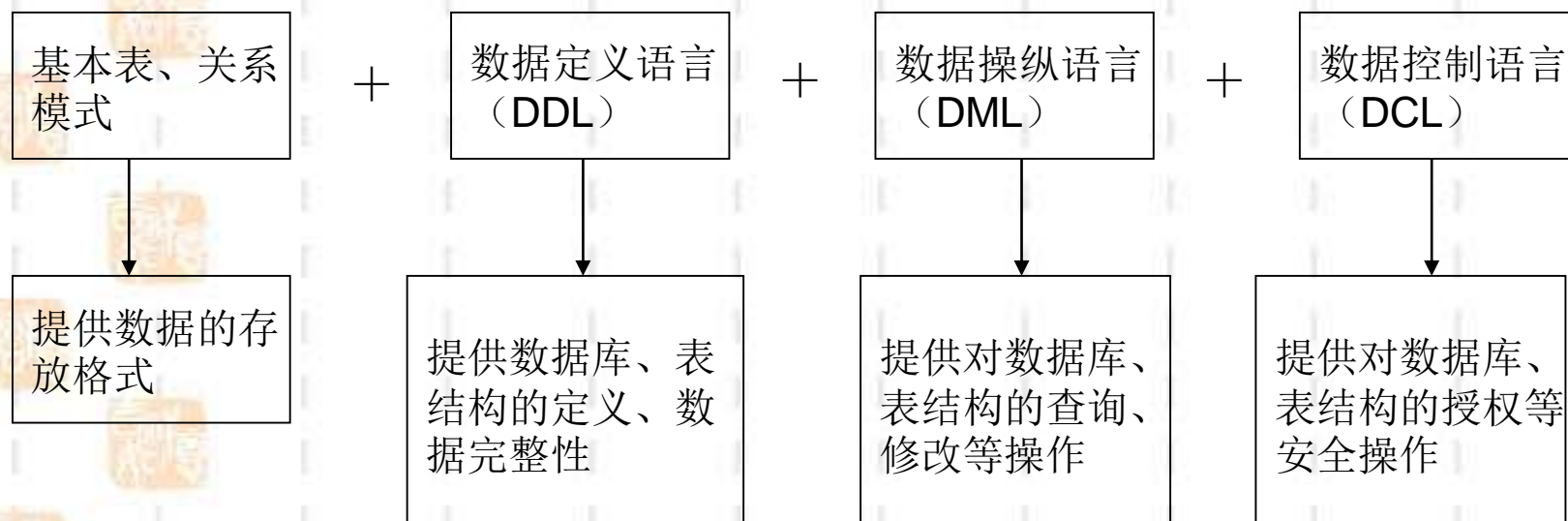


图 简化的关系模型

关系模型的优点

- 单一的数据结构形式，具有高度的简明性和精确性。
- 逻辑结构和相应的操作完全独立于数据存储方式，具有高度的数据独立性。
- 关系模型使数据库的研究建立在比较坚实的数学基础上。
- 关系数据库语言与一阶谓词逻辑的固有内在联系，为以关系数据库为基础的推理系统和知识库系统的研究提供了方便。

关系数据语言

➤ 抽象的查询语言

✓ 关系代数

- 用对关系的运算来表达查询，需要指明所用操作

✓ 关系演算

- 用谓词来表达查询，只需描述所需信息的特性

✓ 元组关系演算

- 谓词变元的基本对象是元组变量

✓ 域关系演算

- 谓词变元的基本对象是域变量

关系数据语言

➤ 具体系统中的实际语言

✓ SQL

- 介于关系代数和关系演算之间，由IBM公司在研制System R时提出

✓ QUEL

- 基于Codd提出的元组关系演算语言ALPHA，在INGRES上实现

✓ QBE

- 基于域关系演算，由IBM公司研制

第2章 关系数据库

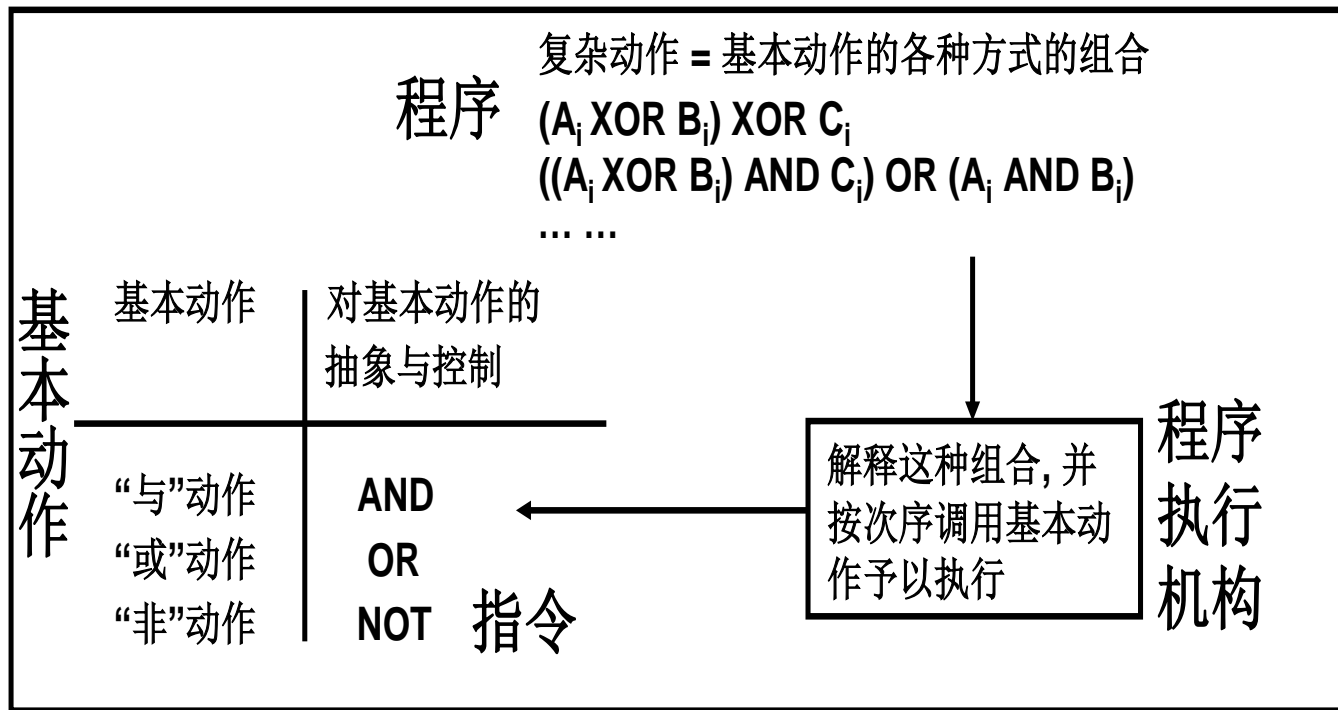
2.1 关系数据库简介

2.2 基本概念

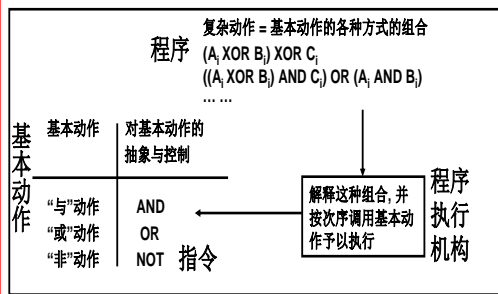
2.3 关系代数

2.4 关系演算

为什么要研究关系代数



为什么要研究关系代数



程序

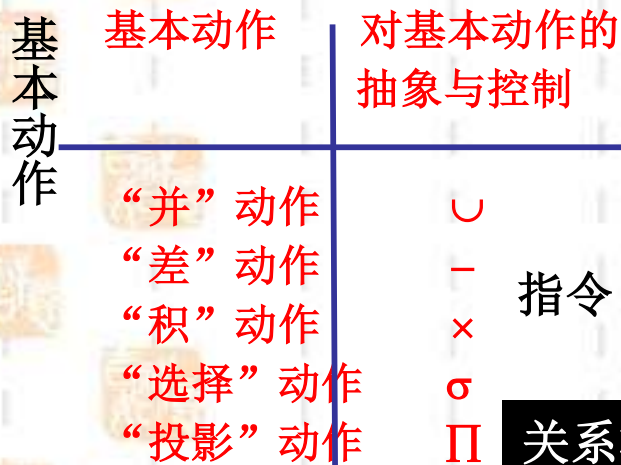
SQL 语言

```
Select Sname From Student, SC
Where Student.S# = SC.S# and SC.C# = '001'
Order By Score DESC;
```

$\Pi_{\text{Sname}}(\sigma_{\text{student.s\#}=\text{sc.s\#}}(\text{Student} \times \text{SC}))$

复杂动作 = 基本动作的各种方式的组合

关系模型基本运算的各种组合



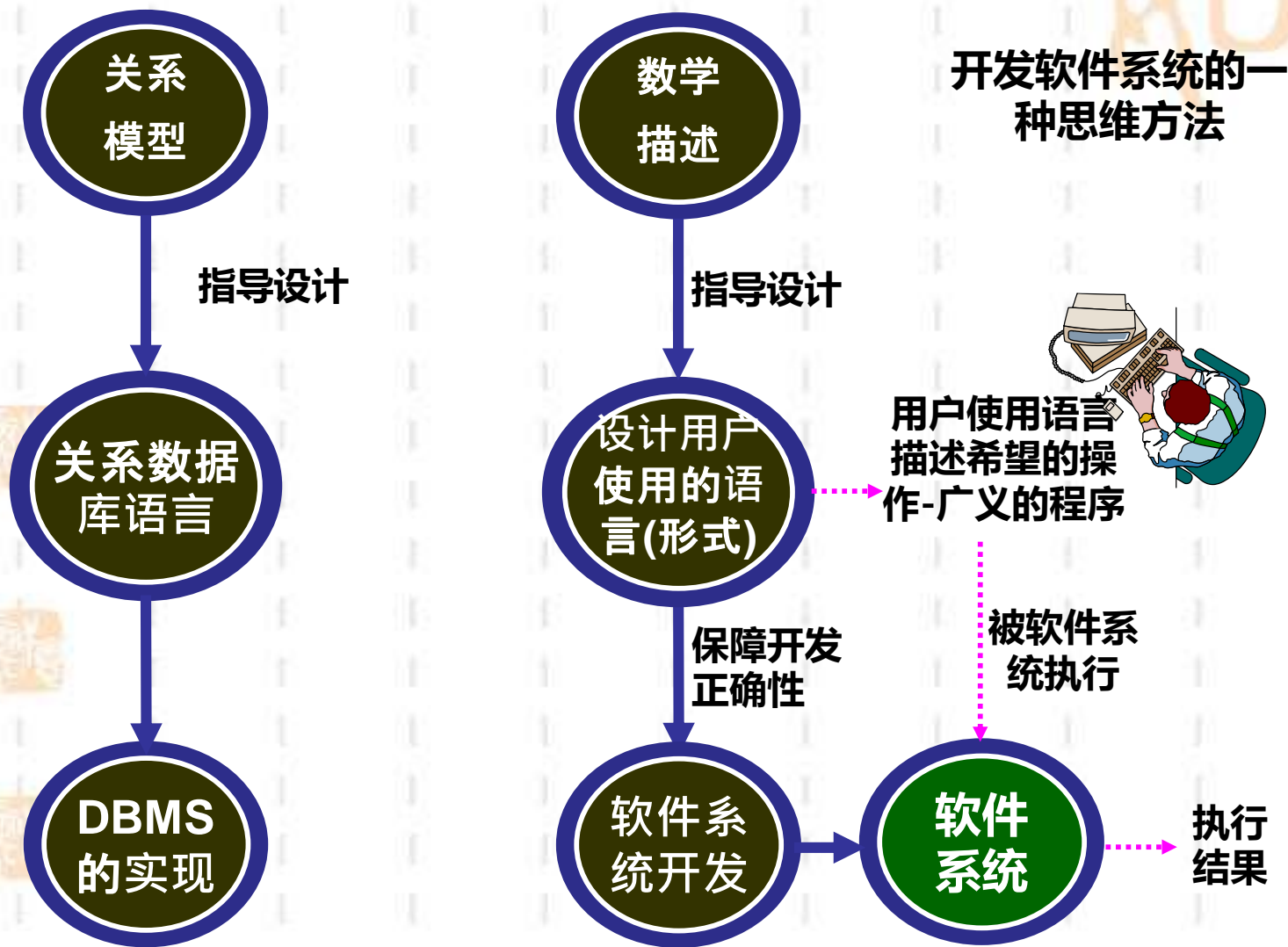
关系模型基本运算

解释这种组合, 并按次序调用基本动作予以执行

程序执行机构

数据库管理系统

关系模型与关系数据语言



关系代数：组成

■ 关系代数中运算对象是关系，运算结果也是关系，运算符包括四类：

- (1) 集合运算符： \cup (并)， $-$ (差)， \cap (交)， \times (广义笛卡尔积)
- (2) 专门的关系运算符： σ (选择)， π (投影)， \bowtie (连接)， \div (除)
- (3) 算术比较符： $>$ (大于)， \geq (大于等于)， $<$ (小于)， \leq (小于等于)， $=$ (等于)， \neq (不等于)
- (4) 逻辑运算符： \wedge (与)， \vee (或)， \neg (非)

关系代数：记号

- 给定关系模式 $R(A_1, A_2, \dots, A_n)$ ，设 R 是它的一个具体的关系， $t \in R$ 是关系的一个元组
- 分量：
设 $t \in R$ ，则 $t[A_i]$ 表示元组 t 中相应于属性 A_i 的一个分量
- 属性列：
 $A_i = \{A_{i1}, A_{i2}, \dots, A_{ik}\} \subseteq \{A_1, A_2, \dots, A_n\}$ ，称 A 为属性列
 \overline{A} 表示 $\{A_1, A_2, \dots, A_n\}$ 中去掉 A 后剩余的属性组
 $t[A_i] = (t[A_{i1}], t[A_{i2}], \dots, t[A_{ik}])$

选择

- 定义

在关系R中选择满足给定条件的元组（从行的角度）

$$\sigma_C(R) = \{t \mid t \in R, C(t) = \text{'真'}\}$$

C是选择的条件， $\forall t \in R$ ， $C(t)$ 要么为真，要么为假

C的形式：由逻辑运算符连接算术表达式而成

逻辑表达式： \wedge ， \vee ， \neg

算术表达式： $X \theta Y$

X，Y是属性名、常量、或简单函数

θ 是比较算符， $\theta \in \{ > , \geq , < , \leq , = , \neq \}$

选择运算：例

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 |

选择运算：例

- 查询年龄不小于20的男生

$$\sigma_{\text{年龄} \geq 20 \wedge \text{SEX} = \text{'男'}}(S)$$

投影

- 定义

- 从关系R中取若干列组成新的关系（从列的角度）

$$\Pi_A(R) = \{ t[A] \mid t \in R \}, A \subseteq R$$

- 投影的结果中要去掉相同的行

R

| A | B | C |
|---|---|---|
| a | b | c |
| d | e | f |
| c | b | c |

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

| B | C |
|---|---|
| b | c |
| e | f |

投影：例

- 列出所有学生的姓名和年龄：

$\Pi_{SN, AGE}(S)$

- 查询001号学生所选修的课程号：

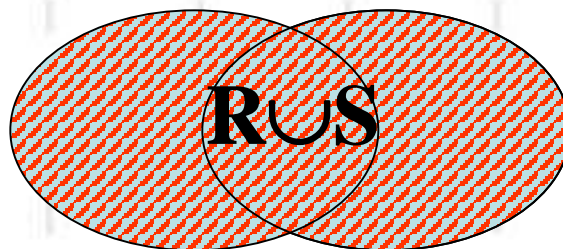
$\Pi_{C\#}(\sigma_{S\#='001'}(SC))$

并

- 定义

- 所有至少出现在两个关系中之一的元组集合

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



- 两个关系R和S若进行并运算，则它们必须是相容的：
 - 关系R和S必须是同元的，即它们的属性数目必须相同
 - 对 $\forall i$ ，R的第i个属性的域必须和S的第i个属性的域相同

并：例

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 |

RUS

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

并：例

- 查询选修了001号或002号课程的学生号
方案1:

$$\Pi_{S\#}(\sigma_{C\# = '001' \vee C\# = '002'}(SC))$$

方案2:

$$\Pi_{S\#}(\sigma_{C\# = '001'}(SC)) \cup \Pi_{S\#}(\sigma_{C\# = '002'}(SC))$$

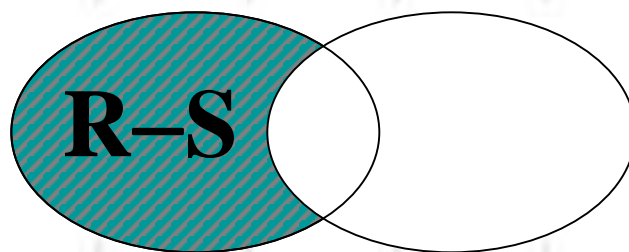
- 问题：哪个方案执行效率更高？

差

- 定义

- 所有出现在一个关系而不在另一关系中的元组集合

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



- R和S必须是相容的

差：例

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 |

差：例

- 查询选修了001号而没有选002号课程的学生号：
- 方案1：

$$\Pi_{S\#}(\sigma_{C\# = '001' \wedge C\# \neq '002'}(SC))$$

- 方案2：

$$\Pi_{S\#}(\sigma_{C\# = '001'}(SC)) - \Pi_{S\#}(\sigma_{C\# = '002'}(SC))$$

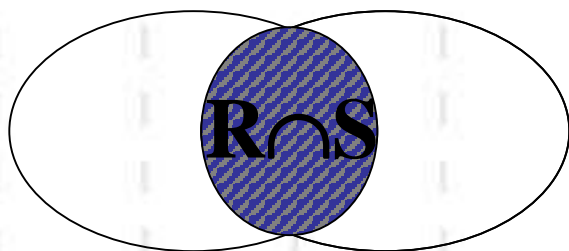
- 注意：方案1是错误滴！

交

- 定义

- 所有同时出现在两个关系中的元组集合

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



- 交运算可以通过差运算重写

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

交:例

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 |

$R \cap S$

| A | B | C |
|---|---|---|
| 7 | 2 | 3 |

交:例

- 查询同时选修了001号和002号课程的学生号
- 方案1:

$$\Pi_{S\#}(\sigma_{C\# = '001' \wedge C\# = '002'}(SC))$$

- 方案2:

$$\Pi_{S\#}(\sigma_{C\# = '001'}(SC)) \cap \Pi_{S\#}(\sigma_{C\# = '002'}(SC))$$

- 注意: 方案1是错误滴!

广义笛卡尔积

- 元组的连串 (Concatenation)

- 若 $r = (r_1, \dots, r_m)$, $s = (s_1, \dots, s_n)$, 则定义 r 与 s 的连串为:

$$\widehat{rs} = (r_1, \dots, r_m, s_1, \dots, s_n)$$

- 定义

- 两个关系 R, S , 其度分别为 m, n , 则它们的笛卡尔积是所有这样的元组集合: 元组的前 m 个分量是 R 中的一个元组, 后 n 个分量是 S 中的一个元组

$$R \times S = \{ \widehat{rs} \mid r \in R \wedge s \in S \}$$

- $R \times S$ 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

广义笛卡尔积:例

r

| A | B |
|----------|-----|
| α | 1 |
| β | 2 |

s

| C | D | E |
|----------|-----|-----|
| α | 10 | a |
| β | 10 | a |
| β | 20 | b |
| γ | 10 | b |

$r \times s$

| 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 |

广义笛卡尔积:例

- $\sigma_{A=C}(r \times s)$

- $r \times s$

| A | B | C | D | E |
|----------|---|----------|----|---|
| α | 1 | α | 10 | a |
| α | 1 | β | 19 | 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 |

广义笛卡尔积:例

- 查询数学成绩比王红同学高的学生

$\Pi_{S.姓名}(\sigma_{R.成绩 < S.成绩 \wedge R.课程=数学 \wedge S.课程=数学 \wedge R.姓名=王红}(R \times \rho_S(R)))$

R

| 姓名 | 课程 | 成绩 |
|----|----|----|
| 张军 | 物理 | 93 |
| 王红 | 数学 | 86 |
| 张军 | 数学 | 89 |

| R.姓名 | R.课程 | R.成绩 | S.姓名 | S.课程 | S.成绩 |
|------|------|------|------|------|------|
| 王红 | 数学 | 86 | 张军 | 物理 | 93 |
| 王红 | 数学 | 86 | 王红 | 数学 | 86 |
| 王红 | 数学 | 86 | 张军 | 数学 | 89 |

θ 连接

- 定义

- 从两个关系的广义笛卡儿积中选取给定属性间满足一定条件的元组

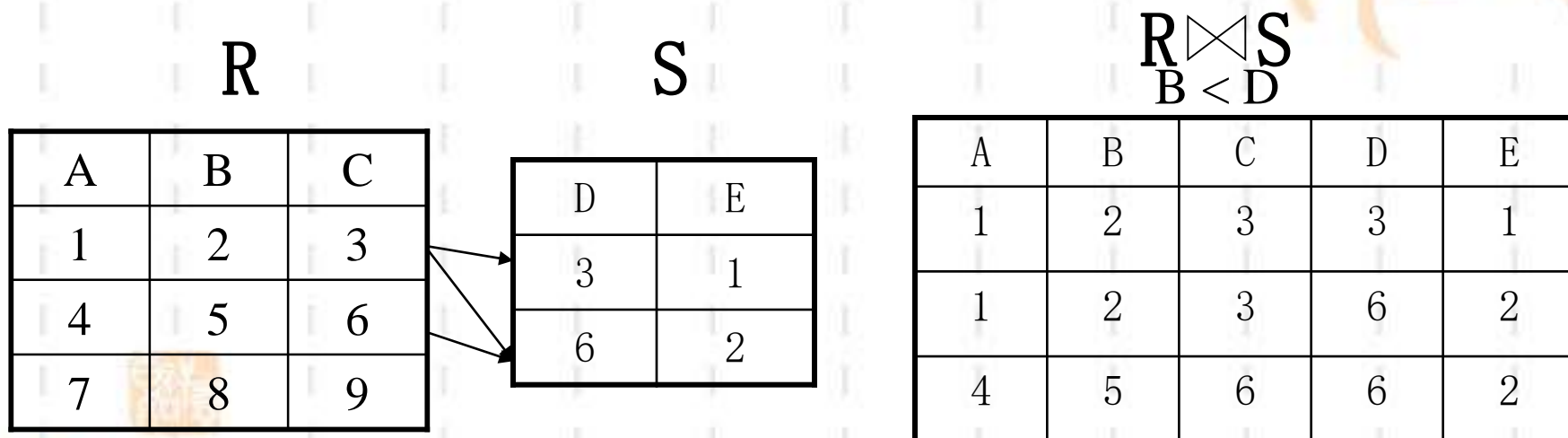
$$R \bowtie_{A \theta B} S = \{ \overbrace{rs} \mid r \in R \wedge s \in S \wedge r[A] \theta s[B] \}$$

A, B为R和S上度数相等且可比的属性列

θ 为算术比较符，为等号时称为等值连接

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

θ连接:例



- 查询数学成绩比王红同学高的学生。

$$\Pi_{S.姓名}((\sigma_{\text{课程=数学} \wedge \text{姓名=王红}}(R)) \bowtie_{R.成绩 < S.成绩} (\sigma_{\text{课程=数学}} \rho_S(R)))$$

自然连接

- 定义

- 从两个关系的广义笛卡儿积中选取在相同属性列B上取值相等的元组，并去掉重复的行

$$R \bowtie S = \{ \pi_{\bar{B}}[r \bowtie s] \mid r \in R \wedge s \in S \wedge r[B] = s[B] \}$$

- 自然连接与等值连接的不同

- 自然连接中相等的分量必须是相同的属性组，并且要在结果中去掉重复的属性，而等值连接则不必。

- 当R与S无相同属性时， $R \bowtie S = R \times S$

自然连接: 例

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 |

自然连接: 例

R

| A | B | C |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

S

| C | D |
|---|---|
| 3 | 1 |
| 6 | 2 |

$R \bowtie S$

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

- 示例

查询001号学生所在系的名称

$$\Pi_{DN} (\sigma_{S\# = 001} (S) \bowtie DEPT)$$

外连接

- 例：列出所有教师的信息，包括姓名、工资、所教授的课程

$\Pi_{P\#, PN, SAL, C\#, CN} ((PROF) \bowtie PC \bowtie C)$

| P# | PN | SAL |
|-----|----|-----|
| P01 | 赵明 | 800 |
| P02 | 钱广 | 700 |
| P03 | 孙立 | 600 |
| P04 | 李三 | 500 |



| C# | P# |
|-----|-----|
| C01 | P01 |
| C02 | P02 |
| C02 | P04 |



| C# | CN |
|-----|----|
| C01 | 物理 |
| C02 | 数学 |
| C03 | 化学 |

| P# | PN | SAL | C# | CN |
|-----|----|-----|-----|----|
| P01 | 赵明 | 800 | C01 | 物理 |
| P02 | 钱广 | 700 | C02 | 数学 |
| P04 | 李三 | 500 | C02 | 数学 |

问题：有关P03号职工的姓名和工资信息没有显示出来

外连接

- 为避免自然连接时因失配而发生的信息丢失，可以假定往参与连接的一方表中附加一个取值全为空值的行，它和参与连接的另一方表中的任何一个未匹配上的元组都能匹配，称之为外连接

外连接 = 自然连接 + 失配的元组

- 外连接的形式：左外连接、右外连接、全外连接

$\bowtie\leftarrow$ 左外连接 = 自然连接 + 左侧表中失配的元组

$\rightarrow\bowtie$ 右外连接 = 自然连接 + 右侧表中失配的元组

$\bowtie\cup$ 全外连接 = 自然连接 + 两侧表中失配的元组

左外连接

| P# | PN | SAL |
|-----|----|-----|
| P01 | 赵明 | 800 |
| P02 | 钱广 | 700 |
| P03 | 孙立 | 600 |
| P04 | 李三 | 500 |



| C# | P# |
|-----|-----|
| C01 | P01 |
| C02 | P02 |
| C02 | P04 |



| C# | CN |
|-----|----|
| C01 | 物理 |
| C02 | 数学 |
| C03 | 化学 |

==

| P# | PN | SAL | C# | CN |
|-----|----|-----|------|------|
| P01 | 赵明 | 800 | C01 | 物理 |
| P02 | 钱广 | 700 | C02 | 数学 |
| P04 | 李三 | 500 | C02 | 数学 |
| P03 | 孙立 | 600 | null | null |

所有老师的信息

右外连接

| P# | PN | SAL |
|-----|----|-----|
| P01 | 赵明 | 800 |
| P02 | 钱广 | 700 |
| P03 | 孙立 | 600 |
| P04 | 李三 | 500 |



| C# | P# |
|-----|-----|
| C01 | P01 |
| C02 | P02 |
| C02 | P04 |



| C# | CN |
|-----|----|
| C01 | 物理 |
| C02 | 数学 |
| C03 | 化学 |

| P# | PN | SAL | C# | CN |
|------|------|------|-----|----|
| P01 | 赵明 | 800 | C01 | 物理 |
| P02 | 钱广 | 700 | C02 | 数学 |
| P04 | 李三 | 500 | C02 | 数学 |
| null | null | null | C03 | 化学 |

所有课程的信息

全外连接

| P# | PN | SAL |
|-----|----|-----|
| P01 | 赵明 | 800 |
| P02 | 钱广 | 700 |
| P03 | 孙立 | 600 |
| P04 | 李三 | 500 |



| C# | P# |
|-----|-----|
| C01 | P01 |
| C02 | P02 |
| C02 | P04 |



| C# | CN |
|-----|----|
| C01 | 物理 |
| C02 | 数学 |
| C03 | 化学 |

| P# | PN | SAL | C# | CN |
|------|------|------|------|------|
| P01 | 赵明 | 800 | C01 | 物理 |
| P02 | 钱广 | 700 | C02 | 数学 |
| P04 | 李三 | 500 | C02 | 数学 |
| P03 | 孙立 | 600 | null | null |
| null | null | null | C03 | 化学 |

所有老师和课程的信息

外连接

| A | B | C |
|----|----|----|
| a1 | b1 | c1 |
| a2 | b2 | c2 |
| a3 | b3 | c3 |

R

| B | C | D |
|----|----|----|
| b1 | c1 | d1 |
| b3 | c3 | d3 |
| b4 | c4 | d4 |

S

| A | B | C | D |
|----|----|----|------|
| a1 | b1 | c1 | d1 |
| a2 | b2 | c2 | null |
| a3 | b3 | c3 | d3 |

$R * \triangleright \triangleleft S$

| A | B | C | D |
|------|----|----|----|
| a1 | b1 | c1 | d1 |
| a3 | b3 | c3 | d3 |
| null | b4 | c4 | d4 |

$R \triangleright \triangleleft * S$

| A | B | C | D |
|------|----|----|------|
| a1 | b1 | c1 | d1 |
| a2 | b2 | c2 | null |
| a3 | b3 | c3 | d3 |
| null | b4 | c4 | d4 |

$R * \triangleright \triangleleft * S$

半联接

- R和S的半联接是将其自然联接的结果在R的属性集上投影，可以表示为：

$$R \bowtie S = \pi_R (R \bowtie S), \quad S \bowtie R = \pi_S (R \bowtie S)$$

- 半联接不具有交换率

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

半联接:例

| A | B | C |
|----|----|----|
| a1 | b1 | c1 |
| a2 | b1 | c1 |
| a3 | b2 | c3 |
| a4 | b4 | c4 |

R

| B | C | D |
|----|----|----|
| b1 | c1 | d1 |
| b1 | c1 | d2 |
| b2 | c2 | d2 |
| b4 | c4 | d3 |

S

| A | B | C |
|----|----|----|
| a1 | b1 | c1 |
| a2 | b1 | c1 |
| a4 | b4 | c4 |

R \bowtie **S**

| B | C | D |
|----|----|----|
| b1 | c1 | d1 |
| b1 | c1 | d2 |
| b4 | c4 | d3 |

S \bowtie **R**

除 (Division)

- 象集 (Image Set)

- 关系 $R(X, Z)$, X, Z 是属性组, x 是 X 上的取值, 定义 x 在 R 中的象集为

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

- 从 R 中选出在 X 上取值为 x 的元组, 去掉 X 上的分量, 只留 Z 上的分量

| X | Z |
|----|----|
| 姓名 | 课程 |
| 张军 | 物理 |
| 王红 | 数学 |
| 张军 | 数学 |

x =张军

| Z_x |
|-------|
| 课程 |
| 数学 |
| 物理 |

张军同学所选修的全部课程

除 (Divison)

如何得到选修了全部课程的学生？

- 方法：逐个考虑选课关系SC中的元组r，求r在姓名SN上的分量x，再求x在选课关系中的象集课程 C_x ，若 C_x 包含了所有的课程C，则x是满足条件的一个元组

选修全部课程的学生

x同学所选修的全部课程

全部课程

$$\{ x \mid x=r[SN] \wedge r \in SC \wedge C_x \supseteq C \}$$

除 (Divison)

- 定义:

- $R(X, Y) \div S(Y) = \{x \mid x=r[R] \wedge r \in R \wedge Y_x \supseteq S\}$

- $R(X, Y) \div S(Y) = \{t \mid t \in \Pi_X(R) \wedge \forall s \in S (ts \in R)\}$

- $R \div S = \Pi_X(R) - \Pi_X(\Pi_X(R) \times \Pi_Y(S) - R)$

- 注意上述3个定义是相互等价的

除 (Divison)

- $R \div S$ 的计算过程
- (1) $T = \Pi_X(R)$
- (2) $W = (T \times S) - R$
- (3) $V = \Pi_X(W)$
- (4) $R \div S = T - V$

除 (Division)

- 例: 计算 $R \div S$

| A | B | C | D |
|---|---|---|---|
| a | b | c | d |
| a | b | e | f |
| a | b | d | e |
| b | c | e | f |
| e | d | c | d |
| e | d | e | f |

关系R

| C | D |
|---|---|
| c | d |
| e | f |

关系S

除 (Divison)

- (1) $T = \Pi_x(R)$

| A | B |
|---|---|
| a | b |
| a | b |
| a | b |
| b | c |
| e | d |
| e | d |

=

| A | B |
|---|---|
| a | b |
| b | c |
| e | d |

关系T

除 (Divison)

$$(2) W = (T \times S) - R$$

| A | B | C | D |
|---|---|---|---|
| a | b | c | d |
| a | b | e | f |
| b | c | c | d |
| b | c | e | f |
| e | d | c | d |
| e | d | e | f |

| A | B | C | D |
|---|---|---|---|
| a | b | c | d |
| a | b | e | f |
| a | b | d | e |
| b | c | e | f |
| c | d | e | d |
| c | d | e | f |

关系W

除 (Divison)

$$(3) V = \Pi_X(W)$$

| A | B |
|---|---|
| b | c |

关系V

除 (Divison)

$$(4) R \div S = T - V$$

| A | B |
|---|---|
| a | b |
| b | c |
| e | d |

关系T

-

| A | B |
|---|---|
| b | c |

关系V

=

| A | B |
|---|---|
| a | b |
| e | d |

除 (Divison)

吉祥如意

| A | B | C | D |
|---|---|---|---|
| a | b | c | d |
| a | b | e | f |
| a | b | d | e |
| b | c | e | f |
| e | d | c | d |
| e | d | e | f |

| A | B |
|---|---|
| a | b |
| e | d |

| C | D |
|---|---|
| c | d |
| e | f |

除 (Divison)

- 示例： 查询同时选修了001和002号课程的学生号

方案1:

$$\Pi_{S\#, C\#} (SC) \div \sigma_{C\# = '001' \vee C\# = '002'} (C)$$

方案2:

$$\Pi_{S\#} (SC \div \sigma_{C\# = '001' \vee C\# = '002'} (C))$$

哪一个正确?

除 (Divison)

所有学生选修全部课程

没有选修全部课程的学生

选修了全部课程的学生

| 课程 |
|----|
| 数学 |
| 物理 |

×

| 姓名 |
|----|
| 张军 |
| 王红 |

=

| 姓名 | 课程 |
|----|----|
| 张军 | 物理 |
| 王红 | 数学 |
| 张军 | 数学 |
| 王红 | 物理 |

| 姓名 | 课程 |
|----|----|
| 张军 | 物理 |
| 王红 | 数学 |
| 张军 | 数学 |
| 王红 | 物理 |

—

| 姓名 | 课程 |
|----|----|
| 张军 | 物理 |
| 王红 | 数学 |
| 张军 | 数学 |

=

| 姓名 |
|----|
| 王红 |

| 姓名 |
|----|
| 王红 |
| 张军 |

—

| 姓名 |
|----|
| 王红 |

=

| 姓名 |
|----|
| 张军 |

除 (Divison)

| A | B | C | D |
|----|----|----|----|
| a1 | b1 | c1 | d1 |
| a1 | b1 | c2 | d2 |
| a1 | b1 | c3 | d3 |
| a2 | b2 | c2 | d2 |
| a3 | b3 | c1 | d1 |
| a3 | b3 | c3 | d3 |

R

| C | D |
|----|----|
| c1 | d1 |
| c3 | d3 |

S

| A | B |
|----|----|
| a1 | b1 |
| a3 | b3 |

$R \div S$

关系代数查询实例

- 关系数据库模式:

Student (S#, Class, Sname, Sex, Age)

Course (C#, Cname, Teacher)

SC (S#, C#, Grade)

关系代数查询实例

■例2-1：查询王老师开设课程的编号及课程名：

$\pi_{C\#, Cname}(\sigma_{Teacher='王'}(Course))$

■例2-2：查询至少选修了一门王老师所开设课程的学生学号：

$\pi_{S\#}(SC \bowtie \pi_{C\#}(\sigma_{Teacher='王'}(Course)))$

关系代数查询实例

- 例2-3：查询仅选修了C01号课程的学生号

选修C01号课程的学生—仅选C01号课程之外的学生

$$\Pi_{S\#}(\sigma_{C\# = 'C01'}(SC)) - \Pi_{S\#}(SC - (\sigma_{C\# = 'C01'}(SC)))$$

| S# | C# | SCORE |
|-----|-----|-------|
| P01 | C01 | 96 |
| P02 | C01 | 90 |
| P03 | C02 | 88 |
| P01 | C03 | 92 |

| S# | C# | SCORE |
|-----|-----|-------|
| P01 | C01 | 96 |
| P02 | C01 | 90 |

| S# | C# | SCORE |
|-----|-----|-------|
| P03 | C02 | 88 |
| P01 | C03 | 92 |

| S# |
|-----|
| P01 |
| P02 |

| S# |
|-----|
| P01 |
| P03 |

| S# |
|-----|
| P02 |

关系代数查询实例

✓例2-4：查询李同学未选课程的编号：

$$\pi_{C\#}(\text{Course}) - \pi_{C\#}(\sigma_{Sname='李'}(\text{Student} \bowtie \text{SC}))$$

✓例2-5：查询至少选修了两门课程的学生学号：

$$\pi_1(\sigma_{1=4 \wedge 2 \neq 5}(\text{SC} \times \text{SC}))$$

✓例2-6：查询全部学生均选修了的课程编号及名称：

$$\pi_{C\#, Cname}(\text{Course} \bowtie (\pi_{S\#, C\#}(\text{SC}) \div \pi_{S\#}(\text{Student})))$$

第2章 关系数据库

2.1 关系数据库简介

2.2 基本概念

2.3 关系代数

2.4 关系演算

关系演算

- 关系演算

以谓词演算为基础表达关系运算。

- 按谓词变元分类

1. 元组关系演算：

以元组变量作为谓词变元的基本对象

2. 域关系演算：

以域变量作为谓词变元的基本对象

元组关系演算

- 在元组关系演算(Tuple Relational Calculus)中，元组关系演算表达式简称为元组表达式，其一般形式为

$$\{ t \mid P(t) \}$$

- 其中：
 - t 是元组变量，表示一个元数固定的元组
 - P 是公式，在数理逻辑中也称为谓词，也就是计算机语言中的条件表达式
- $\{ t \mid P(t) \}$ 表示满足公式 P 的所有元组 t 的集合

元组关系演算

- 在元组表达式中，公式由原子公式组成

- 原子公式有以下三种形式：

- ① $R(t)$ ， t 是元组变量， R 是关系， $R(t)$ 表示 $t \in R$ ，即 t 是 R 的一个元组；
- ② $s[i] \theta t[j]$ ， s 和 t 都是元组变量， $s[i]$ 表示元组 s 的第 i 个分量， $t[j]$ 表示 t 的第 j 个分量， θ 是算术比较运算符。
 $s[i] \theta t[j]$ 表示 s 的第 i 个分量和 t 的第 j 个分量满足 θ 比较关系；
- ③ $s[i] \theta c$ 或 $c \theta s[i]$ ， c 是常量， $s[i] \theta c$ 表示元组 s 的第 i 个分量和常量 c 满足 θ 这样的比较关系。

元组关系演算2022-3-6

- 在定义关系演算操作时，要用到“自由”（Free）和“约束”（Bound）变量概念。
- 在一个公式中，如果元组变量未用存在量词 \exists 或全称量词 \forall 符号定义，那么称为自由元组变量，否则称为约束元组变量。

元组关系演算

■ 在原子公式的基础上可以通过以下递归定义构成元组关系演算公式：

- ① 每个原子是一个公式。其中的元组变量是自由变量；
- ② 如果 P_1 和 P_2 是公式，那么 $\neg P_1$ 、 $P_1 \vee P_2$ 、 $P_1 \wedge P_2$ 和 $P_1 \Rightarrow P_2$ 也都是公式
- ③ 如果 P_1 是公式，那么 $(\exists s) (P_1)$ 和 $(\forall s) (P_1)$ 也都是公式；
- ④ 公式中各种运算符的优先级从高到低依次为： θ ， \exists 和 \forall ， \neg ， \wedge 和 \vee ， \Rightarrow ；
- ⑤ 在公式外还可以加括号，以改变上述优先顺序；
- ⑥ 公式只能由上述5种形式构成，除此之外构成的都不是公式。

元组关系演算

- 在元组关系演算公式中，有下列三个等价的转换规则：

① $P_1 \wedge P_2$ 等价于 $\neg (\neg P_1 \vee \neg P_2)$

$P_1 \vee P_2$ 等价于 $\neg (\neg P_1 \wedge \neg P_2)$

② $(\forall s) (P_1(s))$ 等价于 $\neg (\exists s) (\neg P_1(s))$

$(\exists s) (P_1(s))$ 等价于 $\neg (\forall s) (\neg P_1(s))$

③ $P_1 \Rightarrow P_2$ 等价于 $\neg P_1 \vee P_2$

| p | q | $\neg p$ | $p \Rightarrow q$ | $\neg p \vee q$ |
|---|---|----------|-------------------|-----------------|
| T | T | F | T | T |
| T | F | F | F | F |
| F | T | T | T | T |
| F | F | T | T | T |

元组关系演算

R

| A | B | C |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

S

| A | B | C |
|---|---|---|
| 1 | 2 | 3 |
| 3 | 4 | 6 |
| 5 | 6 | 9 |

| A | B | C |
|---|---|---|
| 3 | 4 | 6 |
| 5 | 6 | 9 |

$\{ t \mid S(t) \wedge t[A] > 2 \}$

| A | B | C |
|---|---|---|
| 4 | 5 | 6 |
| 7 | 8 | 9 |

$\{ t \mid R(t) \wedge \neg S(t) \}$

元组关系演算

| A | B | C |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

R

| D | E |
|---|---|
| 3 | 4 |
| 7 | 6 |

S

| A | B | C |
|---|---|---|
| 4 | 5 | 6 |
| 7 | 8 | 9 |

| A | B | E |
|---|---|---|
| 1 | 2 | 4 |
| 1 | 2 | 6 |
| 4 | 5 | 6 |

$$E1 = \{t \mid \exists(u)(R(t) \wedge S(u) \wedge t[3] > u[1])\}$$

$$E2 = \{t \mid (\exists u)(\exists v)(R(u) \wedge S(v) \wedge u[2] < v[1] \wedge t[1] = u[1] \wedge t[2] = u[2] \wedge t[3] = v[2])\}$$

安全表达式

- 在关系代数中，运算是封闭的，不会出现无限关系。
- 而在关系演算中，却有一些特殊的情况需要考虑，例如表达式 $\{t \mid \neg R(t)\}$ ，如果对计算的范围不加限制，那么该表达式的结果将会是一个无限关系。
- 对于带有量词的公式同样具有类似的问题，验证 $(\forall t) (P(t))$ 为假和验证 $(\exists t) (P(t))$ 为真都需要对讨论范围内所有可能的元组进行验证，如果这个范围是无限的，那么就会出现无穷验证的问题。
- 无限关系和无穷验证在计算机中都是不允许出现的，我们将不产生无限关系和无穷验证的表达式称为安全表达式，为了保证运算的安全性而采取的限制措施称为安全约束。

安全表达式

- 为定义安全约束，首先引入公式P域的概念，用 $dom(P)$ 表示。
- $dom(P) =$ 显式出现在 P 中的值 + 在 P 中出现的关系的元组中出现的值（不必是最小集）。
- 如： $dom(t \mid \neg(t \in R))$ 是 R 中出现的所有值的集合。

安全表达式

■ 如果 $\{t \mid P(t)\}$ 满足下面的三个条件，则 $\{t \mid P(t)\}$ 是安全的：

- ① 如果 $P(t)$ 为真，则 t 的每个分量在 $\text{Dom}(P)$ 中。
- ② 对于 P 中每个形如 $(\exists u)(Q(u))$ 的子表达式，如果 u 使 $Q(u)$ 为真，则 u 的每个分量在 $\text{Dom}(P)$ 中。
- ③ 对于 P 中每个形如 $(\forall u)(Q(u))$ 的子表达式，如果 u 使 $Q(u)$ 为假，则 u 的每个分量在 $\text{Dom}(P)$ 中。

元组关系演算

R

| A | B |
|----|----|
| A1 | B1 |
| A1 | B2 |
| A2 | B3 |

$\{ t \mid \neg(t \in R) \}$

| A | B |
|----|----|
| A1 | B3 |
| A2 | B1 |
| A2 | B2 |

$$\text{dom}(\neg(t \in R)) = \{\{A1, A2\}, \{B1, B2, B3\}\}$$

元组关系演算

例2-7：查询“离散数学”课程的编号及任课教师：

$$R = \{t \mid (\exists u) (Course(u) \wedge u[2] = '离散数学' \wedge t[1] = u[1] \wedge t[2] = u[3])\}$$

例2-8：查询选修了‘数据结构’的学生学号及姓名：

$$R = \{t \mid (\exists u) (\exists v) (\exists w) \\ (Student(u) \wedge SC(v) \wedge Course(w) \wedge u[1] = v[1] \wedge v[2] = w[1] \wedge w[2] = '数据结构' \wedge t[1] = u[1] \wedge t[2] = u[3])\}$$

元组关系演算

- 例2-9: 查询选修了全部课程的学生号

$$\{t \mid \forall u \in C \left(\exists s \in SC \left(s[CN0] = u[CN0] \wedge t[SNO] = s[SNO] \right) \right)\}$$

- 例2-10: 查询选修了张军同学所选修的全部课程的学生姓名
 \forall 课程, 张军选之 \Rightarrow 所求同学选之

$$\{t \mid \forall u \in C (\exists s \in SC \wedge \exists w \in S (s[CN0] = u[CN0] \wedge w[SNO] = s[SNO] \wedge w[SNAME] = \text{张军}) \Rightarrow$$

$$\exists s^1 \in SC \wedge \exists w^1 \in S (s^1[CN0] = u[CN0] \wedge w^1[SNO] = s^1[SNO] \wedge w^1[SNAME] = t[SNAME]) \}$$

元组关系演算

$R(A, B)$ 是二元关系

$$\{t \mid t \in R \wedge \exists u \in R (t[A] \neq u[A] \vee t[B] \neq u[B])\}$$

求**R**中的那些元组，其中在**R**中存在与之不同的元组，
因此当**R**中元组个数为**0**或**1**时，结果为空；当**R**中元组
个数大于**1**时，表达式的结果是**R**本身



$$\{t \mid t \in R \wedge \exists u \in R (t[A] \neq u[A] \wedge t[B] \neq u[B])\}$$

all or none

域关系演算

- 域关系演算表达式的一般形式：
- $\{x_1 \cdots x_k \mid P(x_1, \cdots, x_k)\}$ ，其中 x_1, \cdots, x_k 都是域变量， $P(x_1, \cdots, x_k)$ 是域关系演算公式，一般由原子公式构成

域关系演算

■类似元组演算，域演算原子公式定义如下：

- ① $R(x_1, \dots, x_k)$ ，其中 x_1, \dots, x_k 都是域变量或常量， R 是 k 元关系， $R(x_1, \dots, x_k)$ 表示由 x_1, \dots, x_k 构成的元组满足关系 R ；
- ② $x \theta y$ ，其中 x 和 y 都是域变量， θ 是算术比较运算符， $x \theta y$ 表示 x 和 y 满足 θ 比较关系；
- ③ $x \theta c$ 或 $c \theta x$ ，其中 x 是域变量， c 是常量， $x \theta c$ 表示变量 x 与常量 c 满足 θ 比较关系， $x \theta c$ 和 $c \theta x$ 是等价的。

域关系演算

■在原子公式的基础上域演算公式递归定义如下：

- ① 原子公式也是公式；
- ② 如果 P_1 , P_2 是域演算公式, 则 $P_1 \wedge P_2$, $P_1 \vee P_2$, $\neg P_1$ 也是域演算公式；
- ③ 如果 P 是域演算公式, 则 $\exists x (P)$ 也是域演算公式；
- ④ 如果 P 是域演算公式, 则 $\forall x (P)$ 也是域演算公式；
- ⑤ 域演算公式中各类算符优先级与元组演算相同；
- ⑥ 有限次使用上述规则得到的即是域演算公式, 除此之外的构成形式不是域演算公式。

域关系演算

| A | B | C |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

R

| D | E |
|---|---|
| 3 | 4 |
| 7 | 6 |

S

| A | B | C |
|---|---|---|
| 4 | 5 | 6 |

| B | C | D |
|---|---|---|
| 5 | 6 | 3 |
| 8 | 9 | 3 |
| 8 | 9 | 7 |

$$R1 = \{xyz \mid (R(xyz) \wedge z=6 \wedge y < 10)\}$$

$$R2 = \{xyz \mid (\exists u)(\exists v)(R(uxy) \wedge S(zv) \wedge u \geq v)\}$$

关系代数到关系演算的转换

- 前面已经提及关系代数与关系演算是等价的，实际上Codd在他的论文中就给出了一个规约算法，通过该算法任意一个演算表达式都可以规约为语法上等价的代数表达式。
- 相应的，由于关系代数中存在最小完备集，因此我们只要能够将最小完备集中的运算用关系演算表示出来，那么其它的运算同样也能够用关系代数表示出来。
- 下面给出五种基本运算的元组关系演算等价表示。

关系代数到关系演算的转换

1) \cup

$$- R \cup S = \{t \mid R(t) \vee S(t)\}$$

2) $-$

$$- R - S = \{t \mid R(t) \wedge \neg S(t)\}$$

3) \times

$$- R^{(m)} \times S^{(n)} = \{t^{(m+n)} \mid (\exists u^{(m)}) (\exists v^{(n)}) (R(u) \wedge S(v) \wedge t[1] = u[1] \wedge \cdots \wedge t[m] = u[m] \wedge t[m+1] = v[1] \wedge \cdots \wedge t[m+n] = v[n])\}, \text{ 其中 } t^{(m+n)} \text{ 表示 } t \text{ 是 } m+n \text{ 元关系的元组。}$$

4) σ

$$- \sigma_F(R) = \{t \mid R(t) \wedge F'\}, \text{ 其中 } F' \text{ 是选择条件 } F \text{ 在元组关系演算中的等价表示。}$$

5) π

$$- \pi_{i_1, i_2, \dots, i_k}(R) = \{t^{(k)} \mid (\exists u) (R(u) \wedge t[1] = u[i_1] \wedge \cdots \wedge t[k] = u[i_k])\}$$

元组表达式到域表达式的转换

- 我们可以很容易地把元组表达式转换成域表达式，转换规则如下：
 - ①对于k元的元组变量 t ，可引入k个域变量 $x_1 \cdots x_k$ ，在公式中 t 用 $x_1 \cdots x_k$ 替换，元组分量 $t[i]$ 用 x_i 替换。
 - ②对于每个量词 $(\exists u)$ 或 $(\forall u)$ ，若 u 是m元的元组变量，则引入m个新的域变量 $u_1 \cdots u_m$ 。
 - ③在量词的辖域内， u 用 $u_1 \cdots u_m$ 替换， $u[i]$ 用 u_i 替换， $(\exists u)$ 用 $(\exists u_1) \cdots (\exists u_m)$ 替换， $(\forall u)$ 用 $(\forall u_1) \cdots (\forall u_m)$ 替换。

关系运算的等价性

- 并、差、笛尔卡积、投影和选择是关系代数最基本的操作，并构成了关系代数运算的最小完备集。
- 已经证明，在这个基础上，关系代数、安全的元组关系演算、安全的域关系演算在关系的表达和操作能力上是完全等价的。

作业

➤ 2.3

➤ 2.5

➤ 2.6

➤ 2.7

➤ 2.8

吉祥如意

