2.1 关系数据结构及形式化定义

2.1.1 关系

- 单一的数据结构：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
  - 从用户的角度，关系模型中数据的逻辑结构是一张二维表
- 建立在集合代数的基础上

1. 域

域是一组具有相同数据类型的值的集合：`整数、实数、介于某个取值范围的整数、指定长度的字符串集合、{'男', '女'}、……`

2. 笛卡尔积

给定一组域D_1，D_2，…，D_n，允许其中某些域是相同的。

D_1，D_2，…，D_n的笛卡尔积为：

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

- 所有域的所有取值的一个组合
- 不能重复
- 元组
  - 笛卡尔积中每一个元素`d1, d2, ……, dn`叫作一个n元组，或简称元组

- 元组：（小明，计算机专业，张昀）

- **分量**
  - 笛卡尔积元素`d1, d2, ……, dn`中的每一个值di叫作一个分量
  - 分量：小明，计算机专业，郑云

- **基数**
  - 若D_i（*i = 1，2，……，n）为有限集，其基数为*m_i（*i = 1，2，……，n*），则$D_1 \times D_2 \times \ldots\ldots \times D_n$的基数M为：$M = \prod_{i}^{n} m_i$

- **笛卡尔积的表示方法**
  - 笛卡尔积可表示为一张二维表
  - 表中的每行对应一个元组，表中的每列对应一个域

3. 关系

1. **关系**
   a. $D_1 \times D_2 \times \ldots\ldots \times D_n$的子集叫作在域D1, D2, ……, Dn上的关系，表示为R（D1, D2, ……, Dn）
   b. R：关系名
   c. n：关系的目或度`Degree`
2. **元组**：关系中的每个元素是关系中的元组，通常用t表示
3. **单元关系与二元关系**
   a. 当n=1时，称该关系为单元关系或一元关系
   b. 当n=2时，称该关系为二元关系
4. **关系的表示**：关系也是一个二维表，表的每行对应一个元组，表的每列对应一个域
5. **属性**
   a. 关系中不同列可以对应相同的域
   b. 为了加以区分，必须对每列起一个名字，称为属性

c. n目关系必有n个属性

6. **码**
   a. 候选码：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；*简单的情况*：候选码只包含一个属性
   b. 全码：极端情况：关系模式的所有属性组是这个关系模式的候选码，称为全码
   c. 主码：若一个关系有多个候选码，则选定其中一个为主码
   d. 主属性
      i. 候选码的诸属性称为主属性
      ii. 不包含在任何候选码中的属性称为非主属性或非码属性
   e. 取出有实际意义的元组来构造关系

7. **三类关系**
   a. 基本关系（基本表或基表）：实际存在的表，是实际存储数据的逻辑表示
   b. 查询表：查询结果对应的表
   c. 视图表：由基本表或其他视图表导出的表，是虚表，不对应实际存储的数据

8. **基本关系的性质**
   a. 列是同质的
   b. 不同的列可出自同一个域
      i. 其中的每一个列称为一个属性
      ii. 不同的属性要给予不同的属性名
   c. 列的顺序无所谓，列的次序可以任意交换
   d. 任意两个元组的候选码不能相同
   e. 行的顺序无所谓，行的次序可以任意交换
   f. 分量必须取自原子值：**规范条件中的最基本的一条**

2.1.2 关系模式

1 什么是关系模式

关系模式是型

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

2 定义关系模式

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

关系模式通常简记为：R(U) / R(A1, A2, ……, An)

- R：关系名
- A：属性名
- 域名及属性向域的映象通常直接说明为属性的类型、长度

3 关系模式和关系

关系模式

- 对关系的描述

- 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

2.1.3 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

关系数据库的型与值

- 型：关系数据库模式，是对关系数据库的描述
- 值：关系模式在某一时刻对应的关系的集合，通常称为关系数据库

2.1.4 关系模型的存储结构

关系数据库的物理组织

- 有的关系数据库管理系统中一个表对应一个操作系统文件，将物理数据组织交给操作系统完成
- 有的关系数据库管理系统从操作系统那里申请若干个大的文件，自己划分文件空间，组织表、索引等存储结构，并进行存储管理

2.2 关系操作

2.2.1 基本的关系操作

常用关系操作

- 查询操作：
  - 选择、投影、连接、除、并、差、交、笛卡尔积
  - 五种基本操作：选择、投影、并、差、笛卡尔积
- 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

关系代数语言

- 用对关系的运算来表达查询要求

关系演算语言

- 元组关系演算语言：谓词变元的基本对象是元组变量
- 域关系演算语言：谓词变元的基本对象是域变量

具有关系代数和关系演算双重特点的语言：SQL

2.3 关系的完整性

实体完整性与参照完整性：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

2.3.1 实体完整性

实体完整性规则

- 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

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

2.3.2 参照完整性

1. 关系间的引用

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

2. 外码

- 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*
- 基本关系S称为*被参照关系*或*目标关系*
- 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

3. 参照完整性

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：

- 空值（F的每个属性值均为空值）
- S中某个元组的主码值

2.3.3 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，以便于使用统一的系统方法处理它们，而不需要由应用程序承担这一功能

2.4 关系代数

- 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
- 关系代数
  - 运算对象是关系
  - 运算结果也是关系
  - 关系代数的运算符有两类：集合运算符和专门的关系运算符
- 传统的集合运算是从关系的“水平”方向即行的角度进行的
- 专门的关系运算不仅涉及行而且涉及列
-

运　算　符		含　义
集合运算符	∪	并
	-	差
	∩	交
	×	笛卡尔积
专门的关系运算符	σ	选择
	π	投影
	⋈	连接
	÷	除

2.4.1 传统的集合运算

1. 并

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∪S：仍为n目关系，由属于R或属于S的元组组成，$R \cup S = \{t|t \in R \cup t \in S\}$

2. 差

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，$R-S = \{t|t \in R \cap t/\in S\}$

3. 交

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，由既属于R又属于S的元组组成：

- $R \cap S = \{t|t \in R \cap t \in S\}$
- $R \cap S = R-(R-S)$

4. 笛卡尔积

R：n目关系，k_1个元组

S：m目关系，k_2个元组

R * S：

- 列：n+m列元组的集合
  - 元组的前n列是关系R的一个元组
  - 后m列是关系S的一个元组
- 行：$k_1 * k_2$个元组
  - $R \times S = \{\widehat{t_r t_s} \mid t_r \in R \wedge t_s \in S\}$

2.4.2 专门的关系运算

几个记号

1. R，t∈R，$t[A_i]$

   设关系模式为$R(A_1, A_2, \ldots, A_n)$

   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$A=\{A_{i1}, A_{i2}, \ldots, A_{ik}\}$，其中$A_{i1}, A_{i2}, \ldots, A_{ik}$是$A_1, A_2, \ldots, A_n$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。

   $\overline{A}$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. （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. （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

1. 选择

- 选择又称为限制
- 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
    - $\sigma_F(R)=\{t|t\in R\wedge F(t)='真'\}$
    - 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>，≥，<，≤，=或<>
- 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计算

2. 投影

- 从R中选择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）

3. 连接

- 连接也成为∂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

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

  - A和B：分别为R和S上度数相等且可比的属性组
  - ∂：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属性组上的值满足比较关系∂的元组

- **两类常用连接运算**
  - 等值连接：从关系R和S的广义笛卡尔积中选取A、B属性值相等的那些元组
  - 自然连接：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
  - 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，所以是从行和列的角度进行运算

- **悬浮元组**
  - 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**
  - 把悬浮元组也保存在结果关系中，而在其他属性上填空值
  - 左外连接`LEFT OUTER JOIN`：只保留左边关系R中的悬浮元组
  - 右外连接`RIGHT OUTER JOIN`：只保留右边关系S中的悬浮元组

4. 除运算

- 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中满足条件的元组在X属性列上的投影
- $$R \div S=\{t_r[X] \mid t_r \in R \wedge \pi_Y(S) \subseteq Y_x\}$$
  Y_x：x在R中的象集，$x = t_r[X]$
- 除运算是同时从行和列的角度进行运算

2.6 小结

- 关系数据库是目前使用最广泛的数据库
- 关系数据库与非关系数据库区别
  - 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
- 2.1 关系数据结构及形式化定义[1]
- 2.1.2 关系模式[2]
- 2.1.3 关系数据库[3]
- 2.1.4 关系模型的存储结构[4]
- 2.2 关系操作[5]
  - 查询、数据更新
- 2.3 关系的完整性[6]
  - 实体完整性
  - 参照完整性：外码
  - 用户定义的完整性
- 2.4 关系代数[7]

---

1. ## 2.1 关系数据结构及形式化定义

2. 2.1.2 关系模式
3. 2.1.3 关系数据库
4. 2.1.4 关系模型的存储结构
5. 2.2 关系操作
6. 2.3 关系的完整性
7. 2.4 关系代数