

第二章 关系数据库

- 关系模型概述
 - 关系模型
 - 关系数据库
 - 单一的数据结构：关系
 - 实体和实体之间的联系都用关系描述
 - 有算法把ER模型的实体和联系转化为关系模型的关系
 - 关系操作集合
 - 关系操作的特点
 - 一次一集合的操作方式 set-at-a-time
 - 操作的对象和结果都是关系
 - 抽象查询语言
 - 关系代数
 - ISBL
 - 元组关系演算语言
 - ALPHA
 - QUEL
 - 域关系演算语言
 - QBE
 - SQL
 - 是介于关系代数和关系演算之间的标准查询语言
 - 关系完整性约束
 - 关系模型必须满足
 - 实体完整性
 - 参照完整性
 - 用户定义完整性（语义约束）
 - 有DBMS实现
- 关系数据结构及形式化定义
 - 关系 Relation
 - 域
 - 域是具有相同数据类型的值的集合
 - 笛卡尔积
 - 笛卡尔积

- 给定一组域 D_1, D_2, \dots, D_n , 这些域中可以有相同的域; D_1, D_2, \dots, D_n 的笛卡尔积为:
 - $D_1 \times D_2 \times \dots \times D_n = \{(d_1, d_2, \dots, d_n) \mid d_i \in D_i, i=1, 2, \dots, n\}$
 - 集合论中笛卡儿积不满足交换率, 即笛卡儿积的元组有序。
- n元组 (元组)
 - 每一个元素 (d_1, d_2, \dots, d_n)
- 分量
 - 元素中的每个值 d_i
- 基数 cardinal number
 - 是一个域允许的不同取值个数
 - 若 $D_i (i=1, 2, \dots, n)$ 为有限集, 其基数为 $|D_i|$, 则 $D_1 \times D_2 \times \dots \times D_n$ 的基数为: $|D_1| \times |D_2| \times \dots \times |D_n|$
- 关系
 - $D_1 \times D_2 \times \dots \times D_n$ 的任意子集叫做在域 D_1, D_2, \dots, D_n 上的关系
 - 记做: $R(D_1, D_2, \dots, D_n)$
 - R 为关系名, n 是关系的目或度(degree, arity)
 - 任意子集包括空集, 空关系
 - 关系通过给关系的列附加属性名的方式取消笛卡儿积元组的有序性。
 - 关系可以是无限集
 - 通常我们在关系数据模型中限定关系为有限关系
 - 候选码 candidate key
 - 若关系中的某一属性组的值能唯一地标识一个元组, 而其子集不能, 则称该属性组为该关系的一个候选码(Candidate Key)
 - 主属性
 - 候选码中的属性
 - 非主属性
 - 不包含在任何候选码中的属性
 - 主码 primary key
 - 一个关系可能有多个候选码, 则选定其中一个作为主码(Primary Key)
 - 全码 all-key
 - 关系模式的所有属性组是这个关系模式的唯一候选码
 - 六个性质 (三列两行一分量)
 - 列是同质的
 - 不同列可出自同一个域, 每一列为一个属性, 不同属性(列)给不同属性名
 - 列的顺序可任意交换

- 任意两个元组 **不能完全相同**
 - 行的顺序可任意交换
 - 每一分量是不可分的数据项
 - 范式 normal form, NF
 - 规范化的关系
- 关系模式 relation schema
 - 是对关系的描述, 是type的描述
 - 关系模式是静态的、稳定的, 而关系是动态的、随时间变化的, 两者是型与值的关系
 - $R(U, D, \text{DOM}, F)$
 - R为关系名
 - U为组成该关系地属性名集合
 - D为属性组U中属性所来自的域
 - dom为属性向域的映象集合,
 - $\text{DOM}(\text{SUPERVISOR}) = \text{PERSON}$
 - F为属性间数据的依赖关系集(第六章讨论)
 - 通常简记为R(U)或 $R(A_1, A_2, \dots, A_n)$
 - 需描述元组集合的结构
 - 由哪些属性构成
 - 这些属性来自哪些域
 - 以及属性与域之间的映象关系
 - 关系模式应刻划出**完整性约束条件**
- 关系数据库
 - 关系数据库
 - 所有实体和实体间联系的关系的集合构成一个关系数据库
 - 型: 关系数据库模式, 是对关系数据库的描述。
 - 值: 一般就称为关系数据库
- 关系操作
 - 基本关系操作
 - 查询 query
 - 选择 select**
 - 投影 project**
 - 连接 join
 - 除 divide
 - 并 union**
 - 差 except**

- 交 intersection
 - 广义笛卡尔积
 - etc.
- 更新 update
 - 插入
 - 删除
 - 修改
-
- 特点是集合操作方式
 - 一次一集合 set at a time
 - 对象与结果都是集合
- 关系数据语言分类
 - 共同特点
 - 语言具有完备的表达能力
 - 功能强大
 - 是非过程化的集合操作语言
 - 能嵌入高级语言中使用
 - 高度非过程化的语言
 - 关系代数语言 algebra
 - 如ISBL
 - 是用对关系的运算来表达查询要求的
 - 关系演算语言 calculus
 - 是用谓词来表达查询要求的
 - 元组关系演算语言
 - 谓词变元基本对象是元组变量
 - 如ALPHA, QUEL
 - 域关系演算语言
 - 谓词变元基本对象是域变量
 - 如QBE
 - 具有关系代数和关系演算双重特点的语言
 - SQL
 - Structured Query language
 - 具有数据定义和数据控制功能
 - 集查询、DDL（数据定义语言）、DML（数据操纵语言）和DCL（数据控制语言）于一体的关系数据语言
 - 关系数据库的**标准语言**

- **关系的完整性**

- 完整性规则

- 是对关系的某种约束条件

- 三类完整性

- **实体完整性 entity integrity (必)**

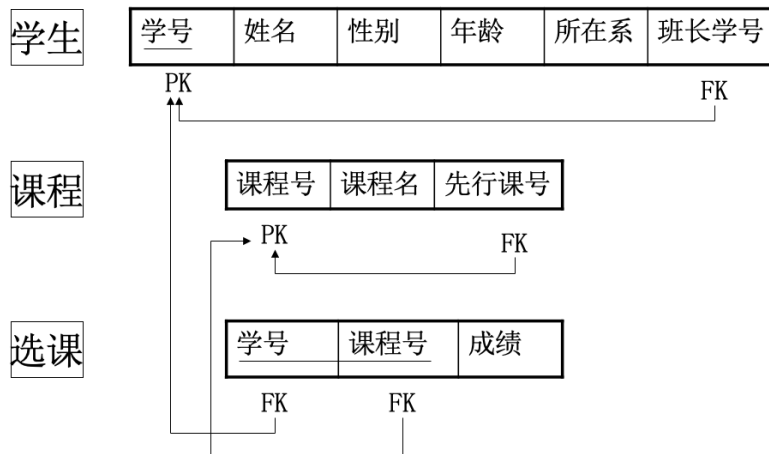
- 实体完整性规则

- **每一关系必有一主码,构成主码的各属性值均不能取空值**
 - 实体完整性可以引申一下, 主码 (Primary key)和没有被选为主码的候选码(Unique)都不能取重复值
 - 注意⚠
 - ~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。(名字怎么来的)
 - 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。
 - 相应地, 关系模型中以主码作为唯一标识。
 - 主码中的属性, 即主属性, 不能取空值。所谓空值就是“不知道”或“不确定”的值

- **参照完整性 referential integrity (必)**

- 与实体完整体被称为两个不变性

- 参照完整性规则



- 是定义外码与主码之间的引用规则
 - 关系与关系之间的引用。
 - 引用的时候, 必须取基本表中已经存在的值
 - 外码 foreign key
 - 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Ks相对应, 则称F是基本关系R的外码
 - 参照关系 referencing relation
 - 基本关系R
 - 被参照关系 referenced relation (目标关系 target relation)

- 基本关系S
- 关系R和S不一定是不同的关系。
- 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**Ks**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
 - 用户定义的完整性
 - 是针对**某一具体**关系数据库的约束条件，它反映某一具体应用所涉及的数据必须满足的语义要求
- 关系代数(要求：会算，会写)
 - 运算对象&运算结果
 - 都是关系
 - 运算符 (4)
 - 集合运算符
 - 传统的集合运算
 - 二目运算 (4)
 - 并 \cup
 - **R, S具有相同的目n，且相应属性取自同一个域**
 - $R \cup S = \{t | t \in R \vee t \in S\}$
 - 差 $-$
 - **R, S具有相同的目n，且相应属性取自同一个域**
 - $R - S = \{t | t \in R \wedge t \text{ 不属于 } S\}$
 - 交 \cap
 - **R, S具有相同的目n，且相应属性取自同一个域**
 - $R \cap S = \{t | t \in R \wedge t \in S\}$
 - 基本运算表达
 - $R \cap S = R - (R - S)$
 - $R \cap S = S - (S - R)$
 - $R \cap S = (R - (R - S)) \cup (S - (S - R))$
 - 笛卡尔积 \times
 - 元组连接
 - R为n目关系，S为m目关系。tr ts 称为元组的连接 (Concatenation)。它是一个(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

$\overbrace{tr \ ts}$

- 两个分别为n目和m目的关系R和S的广义笛卡尔积是一个(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的一个元组。记作：

$$R \times S = \{ \overbrace{tr \ ts} \mid tr \in R \wedge ts \in S \}$$

- 若R有k1个元组，S有k2个元组，则关系R和关系S的广义笛卡尔积有k1×k2个元组
- 专门的关系运算符

- **选择 select**

- 又称为限制 (Restriction)
- 在关系R中选择满足给定条件的诸元组，记做：

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

- F：表示选择条件，它是一个逻辑表达式，取逻辑值‘真’或‘假’
- 逻辑表达式F的基本形式

$$X1 \ \theta \ Y1 \ [\ \phi \ X2 \ \theta \ Y2 \]$$

- θ 表示比较运算符
- ϕ 表示逻辑运算符
- **投影 project**
 - 有关两个记号
 - 元组属性分量记号
 - 设关系模式为R(A1,A2,...,An)。它的一个关系设为R。t∈R表示t是R的一个元组。t[Ai]则表示元组t中相应于属性Ai的一个分量
 - 元组属性列分量记号
 - 若A={Ai1, Ai2,...,Aik}，其中Ai1,Ai2,...,Aik是A1,A2,...,An中的一部分，则A称为属性列或域列。
 - 非A则表示{A1,A2,...,An}中去掉{Ai1,Ai2,...,Aik}后剩余的属性组。
 - t[A]=(t[Ai1],t[Ai2],...,t[Aik])表示元组t在属性列A上诸分量的集合
 - 投影
 - 从R中选择出若干属性列组成新的关系

- 记作： $\Pi_A(R) = \{t[A] \mid t \in R\}$
 - A为R中的属性列
- 从关系中消除某些属性
- 取消了某些属性列后，就可能出现重复行，应取消这些完全相同的行

- 连接 join

- 也称为 θ 连接
- 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
- 记作：

$$R \bowtie_{A \theta B} S = \{tr \underset{A}{ts} \mid tr \in R \wedge ts \in S \wedge tr[A] \theta ts[B]\}$$

- A和B分别为R和S上度数相等且可比的属性组
- θ 是比较运算符
- **连接运算**从R和S的笛卡尔积 $R \times S$ 中选取（R关系）在A属性组上的值与（S关系）在B属性组上值满足比较关系 θ 的元组
- 类型
 - **等值连接（equijoin）**
 - θ 为“=”的连接运算称为等值连接（equijoin）
 - 它是从关系R与S的笛卡尔积中选取A、B属性值相等的那些元组。
 - **自然连接（Naturaljoin）**
 - 是一种**特殊的等值连接**，它要求两个关系中进行比较的分量必须是相同的属性组，并且要在结果中把重复的属性去掉
 - 自然连接还**需要取消了重复列**，所以是同时从行和列的角度进行运算。
- 外连接
- 半连接
- 赋值运算
- 连接的计算
 - 根据连接的类型（ θ 连接、自然连接），确定结果的属性。
 - 计算过程类似二重循环
- 连接的表达：

$$R \underset{F}{\bowtie} S = \sigma_F(R \times S)$$

- 除 divide

- 解释

- Given relations $r(R)$ and $s(S)$, such that S 属于 R , $r \div s$ is the largest relation $t(R-S)$ such that 商是满足逆乘不溢出的最大关系

$$t \times s \subseteq r$$

- 象集

- 给定一个关系 $R(X,Y)$, X 和 Y 为属性组。定义, 当 $t[X]=x$ 时, x 在 R 中的**象集 (Image Set)** 为: $Y_x = \{ t[Y] \mid t \in R, t[X] = x \}$
 - 它表示 R 中属性组 X 上值为 x 的诸元组在 Y 上分量的集合

- 计算过程

补充例5: 象集的计算。设 $R(A, B)$ 如下

A	B

1	aa
1	ab
2	st
1	mm

求 B_1

计算过程

- 考察 r 的每一个元组 t
- 如果 $t[A]=1$, 则把 $t[B]$ 放入结果。
- 得如下:

B

aa
ab
mm

- \div 的定义

- 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：

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

- Y_x 为x在R中的象集， $x = \text{tr}[X]$
- 除运算有“包含”的语义(书上例子)
- 除的表达

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

- 算术比较符
- 逻辑运算符
- 关系演算（了解，自学）

- ALPHA

- 格式：
 - 表达式： 操作条件
 - eg: GET W
(Student.Sno, Student.Sage): Student.Sdept='IS' ^ Student.Sage < 20
- 语句
 - GET
 - DOWN
 - RANGE
 - PUT
 - HOLD
 - UPDATE
 - 修改
 - 插入
 - 删除
 - DELETE
 - DROP