关系数据模型

2.1 关系数据结构

2.1.1 二维表与关系数据结构

二维表有如下特点：
1.每个表具有表名
2.表由表头和若干行数据两部分组成
3.表由若干列，每列都有列名
4.同一列的值必取自同一个域
5.每一行的数据代表一个实体的信息
从用户的角度看，一个关系就是一个规范化的二维表,这里规范化的含义是：表中每列都是原子项，没有表中表

一个关系由关系名、关系模式（即关系的描述）和关系实例（即每一个实体的信息）组成。

术语：
关系：一个关系指一张二维表
元组：一个元组指关系表中的一行
属性：一个属性指二维表中的一列
码（键、关键字、关键码）：指表中唯一确定元组的属性或属性组合。
域：属性的取值范围
分量：元组的一个属性值
关系模式：对关系“型”的描述
关系：值

2.1.2 关系数据结构的形式化定义

关系数据结构的形式化定义：
域：域是一组具有相同数据类型的值的集合。
笛卡尔积；给定一组域 $D_1, D_2, \cdots, D_n$，$D_1, D_2, \cdots, D_n$ 的笛卡尔积为：$D_1 \times D_2 \times \cdots \times D_n = \{(d_1, d_2, \cdots, d_n) | d_i \in D_i, i = 1, 2, \cdots n\}$。
笛卡尔积中，每个分量 d_i 是按序排列的；元组中的每一个 d_i 叫做一个分量；笛卡尔积也是一个集合，笛卡尔积的基数 M（即元素 $(d_1, d_2, \cdots, d_n)$ 的个数），为所有域的基数的累计。
关系：笛卡尔积 $D_1 \times D_2 \times \cdots \times D_n$ 的任一子集称为域 $D_1, D_2, \cdots, D_n$ 上的关系。
关系可用 $R(D_1, D_2, \cdots, D_n)$ 的形式表示，其中 R 为关系名， n 是关系的度，也称目。
说明：

关系中元组的个数是关系的基数。
关系是一个二维表。
在数学上，关系是笛卡尔积的任意子集，但在数据库系统中，关系是看i大耳机中所取得有意义的子集，且数据库中关系满足交换律。

2.1.3 关系的性质

关系的性质：
1.列是同质的：每列中的分量必须是同一类型的数据。
2.不同的列可以出自同一个域，但不同的属性必须赋予不同的属性名。
3.列的顺序可以任一交换。
4.任意两个元组不能完全相同。
5.关系中的元组的顺序可以任意：可以任意交换两行的次序；
6.分量必须取原子值：要求每个分量都是不可再分的数据项

2.1.4 关系模式

关系模式是对关系“型”的描述
关系模式可形式化地表示为 $R(U, D, dom, F)$，其中 R 为关系名， U 为组成关系的属性名集合，D 为属性组 U 中属性所来自的域，dom 为属性与域之间的映像集合，F 为属性间依赖关系的集合。

2.1.5 关系数据库

在给定应用领域，所有实体及实体之间联系的关系集合构成一个关系数据库。

关系数据库也区分“型”和“值”。关系数据库的型即关系数据库模式。
关系数据库模式包括若干域的定义以及在这些域上定义的若干关系模式。
关系数据库的值是关系数据库模式中各关系模式在某一时刻对应的关系的集合。

空值表示信息的空缺，即表示未知的值或不存在的值，一般用 NULL 表示。

2.1.6 码

候选码（候选关键字、候选键）：能唯一标识关系中元组的一个属性或属性集
某一子集当且仅当满足唯一性和最小性时，为候选码。
唯一性：对关系 **R** 的任两个元组，其在属性集 **K** 上的值是不同的。
最小性：若删除当前子集对的任一属性，都不满足最小性。
全码：关系模式的所有属性组都是这个关系模式的候选码时称为全码。

主码：若一个关系有多个候选码，则从中选择一个作为主码。
通常，为了表示方便，在主码所包含的属性下方用下划线标出。
包含在主码中的各属性称为主属性。（还有非主属性，也叫非码属性）

外码：如果关系 **R1** 的属性或属性组 **K** 不是 **R1** 的主码，而是另一关系 **R2** 的主码，则称 **K** 为关系 **R1** 的外码，并称关系 **R1** 为参照关系，关系 **R2** 为被参照关系。
关系 **R1**、**R2** 不一定是不同的关系，主码和外码必须定义在同一个（或一组）域上，外码并不一定要域相应的主码同名，外码和相应的主码属于不同的关系时，往往取相同的名字，以便于实现。

2.2 关系操作

2.2.1 基本关系操作

1.查询
2.更新：插入、删除、修改

2.2.2 关系数据语言分类

早期的关系操作通常用代数方式或逻辑方式表示，分别成为管理代数和关系演算。
关系操作：关系代数和关系演算
关系演算：元组关系演算和域关系演算
关系数据库的标准语言是 SQL

2.2.3 关系代数

关系代数是一种抽象的查询语言，是用对关系的运算来查询的，其运算对象是关系，运算结果也是关系。

关系代数用到的运算符主要包括四类：集合运算符、专门的关系运算符、比较运算符、逻辑运算符

关系代数的运算可以分为两类：
1.传统的集合运算：其运算以元组作为集合中元素来进行。从行的角度进行运算。包括并、差、交、笛卡尔积。
2.专门的关系运算：其运行不仅涉及行，也涉及列。包括选择、投影、连接和除法。

传统的集合运算除了笛卡尔积之外，都要求参与运算的两个关系满足“相容性”条件：1.具有相同的目数（两个关系具有 n 个属性）。2.相应的属性取自同一个域。

1.并
2.差
3.交
4.广义笛卡尔积：广义笛卡尔积可以用于两个关系的连接操作。

专门的关系运算
引入概念：

1.设关系模式为 $R(A_1, A_2, \cdots, A_n)$，它的一个关系为 R，$t \in R$ 表示 t 是 R的一个元组，$t[A_i]$ 表述元组 t 中相对于属性 A_i 的一个分量。
2.若 A 是一个属性列，那么 $\overline{A}$ 是属性名的补集。
3.设 R 是 n 目关系，S 为 m 目关系，$t_R \in R, t_S \in S$，$\overline{t_R t_S}$ 称为元组的连接，且是一个 n + m 列的元组，前 n 个分量为 R 的一个 n 元组，后 m 个分量为 S 中的一个 m 元组。
4.给定一个关系 R(X, Z)，设 X 和 Z 为属性组，定义当 t[X] = x 时，x 在 R 中的像集为 $Z_x = \{t[Z]|t \in R, t[X] = x\}$。

4 个关系代数运算：
1.选择（限制）
是单目运算，是根据一定的条件在给定的关系 R 中选取若干元组，组成一个新关系。
记为：$\sigma_F = \{t|t \in R \wedge F(t) =' 真'\}$
是从行的角度对关系进行的操作。
2.投影
单目运算。从 R 中 选择若干属性列组成新的关系。是对关系在投影方向进行的运算，从左到右按指定的若干属性及顺序取出相应列，删去重复元组（避免重复行）。
记为：$\prod_A(R) = \{t[A]|t \in R\}$
3.连接
是二目运算，是从两个关系的笛卡尔积中选取满足连接条件的元组，组成新的关系。
记为：$R \infty S = \{\overline{t_R t_S}|t_R \in R \wedge t_S \in S \wedge t_R[X]\theta t_S[Y]为真\}$
当 θ 为 “=” 时，称为等值连接；当 θ 为 “<” 时，称为小于连接；当 θ 为 “>” 时，称为大于连接。

连接运算可以用选择运算和广义笛卡尔积运算来表示：$R \infty S = \sigma_{X=Y}(R \times S)$

在连接运算中，一种最常用的连接数自然连接。自然连接就是在等值连接的情况下，当连接属性 X 与 Y 具有相同的属性组时，把在连接结果中重复地属性列去掉。

等值连接与自然连接的区别：
1.等值连接中不要求相等属性值的属性名相同，而自然连接要求相等属性值得属性名必须相同，即两关系只有同名属性才能进行自然连接
2.等值连接不将重复属性去掉，自然连接去掉重复属性。

如果进行自然连接中把舍弃的元组也保存在结果中，而在其他属性上填空值，则称这种连接为外连接。
如果只把左边关系中舍弃的元组保存在结果中，则称这种连接为左外连接；相应的称为右外连接。

4.除法
是二目运算。在行和列上进行运算。
设有关系 R(X, Y) 与关系 S(Y, Z)，其中 X，Y，Z 为属性集合，R 中的 Y 与 S 中的 Y 可以有不同的属性名，但对应属性必须来自同一个域。关系 R 除以关系 S 所得的是一个新关系 P(X)，P 是 R 中满足下列条件的元组在 X 上的投影：元组在 X 上分量值 x 的像集 Y_X 包含 S 在 Y 上投影的集合。
记为：$R \div S = \{t_R[X]|t_R \in R \wedge \prod_Y(S) \subseteq Y_X\}$

2.3 关系完整性

2.3.1 实体完整性

实体完整性规则：指关系 **R** 的主属性不能取空值
根据实体完整性约束，一个关系中不允许存在两类元组：**1.**无主码值的元组。**2.**主码值相同的元组。

2.3.2 参照完整性

参照完整性指被参照关系的主码和参照关系的外码必须定义在同一个域上，并且参照关系的外码的取值只能是以下两种情形之一：**1.**或者取空值。**2.**或者取被参照关系的主码所取的值，

实体完整性和参照完整性关系模型必满足的完整性约束条件，由关系系统自动支持。

2.3.3 用户定义完整性

用户定义完整性规则就是针对数据的具体内容定义的数据约束条件，并提供检验机制。