

关于关系代数运算与 SQL 语句的等价实现

一、传统的集合运算与SQL语句实现

传统的集合运算包括四种运算：并（ \cup ）、交（ \cap ）、差（ $-$ ）、广义笛卡尔（ \times ）。

并运算 (Union)

对应的 SQL 语句：

```
Select * from R Union Select *from S;
```

交运算(Intersect)

对应的SQL语句：

```
Select * from R Intersect Select *from S;
```

差运算(Except)

对应的SQL语句：

```
Select * from R Except Select *from S;
```

注意：对于并、交与差运算，都要求关系R和关系S具有相同的目数，且相应的属性取自同一个域。

笛卡尔积 $R \times S$ 运算 (Extended Cartesian Product)

对应的SQL语句：

```
Select * from R ,S;
```

二、专门的关系运算

专门的关系运算包括四种运算即选择（ σ ）、投影（ Π ）、连接（ \Join ）和除法（ \div ），是关系数据库数据维护、查询、统计等操作的基础。

选择(Selection)

设有关系R，在关系R中求取满足给定条件F的元组组成新的关系的运算称为选择。记作 $\sigma_F(R)$ 。其中F是一个条件表达式，其值为“真”或“假”。 $\sigma_F(R)$ 是从关系R中选取使条件表达式F为真的元组。

对应的SQL语句：

```
SELECT R. 学号, R. 课程名, R. 分数 from R WHERE 分数 >85
```

投影 (Projection)

设有关系R，在关系R中求指定的若干个属性列组成新的关系的运算称作投影，记作 $\Pi_A(R)$ ，其中A为欲选取的属性列列名的列表。

对应的SQL语句：

若关系运算式为： $\Pi_{\text{品名, 数量}}(R)$

```
SELECT 品名, 数量 FROM R;
```

连接 (Join)

从两个分别为 n , m 目的关系 R 和 S 的广义笛卡尔积中选取满足给定条件 F 的元组组成新的关系称为 R 和 S 的连接, 记作 $R \bowtie F S$ ($F=A \theta B$)。其中 A 和 B 分别为 R 和 S 上度数相等且可比的属性列, θ 是算术比较符 ($>$, \geq , $<$, \leq , $=$, \neq)。即连接运算就是从两个关系的笛卡尔积中选取满足一定连接条件的元组的集合, 连接的结果是一个 $(k_1 + k_2)$ 元的关系。

对应的SQL语句:

```
Select * from R, S where R.A = S.C;
```

除 (Division)

给定关系 $R(x, y)$ 与 $S(y)$ 其中 x, y 为属性集 (也可单属性), R 中的 y 和 S 中的 y 是同名的属性 (集) 也可以有不同的属性名, 但必须出自相同的域集。在求解 $R \div S$ 时, 对 R 按 x 的值的分组, 然后检查每一组, 如某一组中的 y 包含 S 中全部的 y , 则取该组中的 x 的值作为关系 P 中的一个元组, 否则不取。 $R \div S$ 的商等于关系 P 。

对应于SQL语言:

```
SELECT DISTINCT A.Name FROM R AS A WHERE NOT EXISTS  
(SELECT * FROM S AS B WHERE NOT EXISTS (SELECT * FROM R AS C where C.Name=A.Name  
and B.City=C.City))  
GROUP BY A.Name
```