8. 扩展关系代数

Bag（可重集）语义：一个关系实际上是一个 Bag（可重集，multiset），可以包含多于一个相同的元组，没有特定的顺序。

- 关系代数中定义的选择、投影以及连接同样适用于 Bag；
- **选用 Bag 语义的原因**：高效的实现（不需要删除重复元组）

Bag 运算

- 并集：一个元素在并集中出现的次数等于其在两个集合中出现次数之和；
- 交集：一个元素在交集中出现的次数等于其在两个集合中出现次数的最小值；
- 对称差：如果 t 在 R 中出现了 n 次，在 S 中出现了 m 次，那么在 $R-S$ 中出现了 $\max(n-m,0)$ 次。
- 对于投影、选择、乘积以及连接，Bag 的运算与 Set 运算相同，不需要删除重复。

重复消除：$\delta(R)$ 返回一个 R 中删除重复元素后的关系，可以将可重集转化为集合。

聚合操作 (aggregation operation)：从很多值中综合出一个值的操作称为聚合操作，如 `SUM`，`AVG`，`MIN`，`MAX` 与 `COUNT` 等等。

分组运算 (grouping operation)：$\gamma_L(R)$ 表示将表的元组根据一个或多个属性将元组进行分组，再对每一组施加聚合操作；其中 L 要么是分组元素（根据其将元素进行分组），要么是 $\theta(A)$，其中 θ 是聚合操作子且 A 是聚合属性。

- 可以用 $\theta(A)\to AttName$ 来给聚合操作后的属性命名。
- 例：$\gamma_{sno,AVG(grade)}(SC)$ 表示对每个学号计算平均分。
- $\gamma_L(R)$ 通过如下方式计算：
  - 将 R 中元组根据 L 中的分组属性进行分组，如果 L 中没有分组属性，那么 R 单独分为一组；
  - 对于每组，根据聚合操作进行计算即可。

扩展投影 (extended projection)：投影 $\pi_L(R)$ 中，表 L 可以有如下类型的元素：

- R 中的属性；
- 一个表达式 $x\to y$ 进行属性更名；
- 一个表达式 $E\to y$，对于每一个元组，计算出表达式 E 的值得到新的一列，并命名为 y。

排序操作 (sorting operation)：排序操作 $\tau_L(R)$ 按照 L 中属性对 R 中元组进行排序。排序后的结果有 `list` 的类型。

- 将元素进行排序可以使得我们快速查找元素，并且提高数据库的效率。
- 排序后的 `list` 类型进行连接时可能会丢失有序性，但是有序性在投影和选择时可以保持。

外连接 (outerjoin) $R\overset{\circ}{\bowtie}S$ 除普通连接包含的元组，保留 R 与 S 中的 dangling tuples；对 dangling tuples，连接得到的其他属性为空。这个操作也被称为（自然）外连接，也记作 $\overset{\circ}{\bowtie}_F$。

- 左-外连接 (left-outerjoin) $\overset{\circ}{\bowtie}_L$；右-外连接 (right-outerjoin) $\overset{\circ}{\bowtie}_R$：只加入左边或右边的 dangling tuples。

- θ-外连接 (theta-outerjoin) $\overset{\circ}{\bowtie}_\theta$：带条件 θ 的外连接。

8.5. 基于关系逻辑的 Datalog 语言

RDB v.s. Datalog

RDB	Datalog
关系 $R()$	谓词 $R()$
属性 X	参数 x
关系实例 $R(X)$	关系原子 $R(x)$
元组的集合	Bool 值函数
元组 $t \in R$	$R(t)$ 为 TRUE

算术原子 (arithmetic atoms)：两个算术表达式的比较 $exp_1\ \theta\ exp_2$，可以看成谓词 $\theta(exp_1, exp_2)$。

Datalog 规则：head $\leftarrow$ body，其中

- head 是一个关系原子；
- body 是子目标的 AND 表达式，其中
  - 子目标是原子或原子的 NOT 表达式；
  - 原子可以是关系原子或算术原子；
  - 参数可以是常量或变量。
- $\leftarrow$ 表示 if，在 Datalog 中用 `:-` 表示。

查询：是一系列规则的组合。

- 如果规则的 head 仅有一个关系，那么这个关系的值即为查询的答案；
- 如果规则的 head 包含多个关系，则其中一个是询问的答案；其余关系用来协助定义答案，一般用 `Answer` 命名询问结果关系。

安全性条件：每个在规则中出现过的变量必须在某些非否定（没有前缀 NOT）的关系子目标中出现，这样保证了结果关系的有限。下面是一些违反安全性条件的例子：

- $S(x) \leftarrow R(y)$，因为 x 不在子目标中；
- $S(x) \leftarrow \text{NOT } R(x)$，因为 x 不在非否定子目标中；
- $S(x) \leftarrow R(y) \text{ AND } x < y$，因为 x 不在关系子目标中出现。

外延性谓词与内涵性谓词

- 外延性谓词 (EDB predicate)：存储在数据库中的关系；
- 内涵性谓词 (IDB predicate)：按照规则计算出来的关系；
- 外延性谓词不能出现在规则的 head，只能出现在规则的 body 中；内涵性谓词可以出现在规则的 head、body 中，一般用来构造复杂的外延性关系的函数；
- 内涵型谓词有时也用做中间结果的暂时存储。

应用于 Bag 的 Datalog 规则：与应用于 Set 的规则类似。

关系模型与 Datalog 语言

- 交集$R \cap S$：$Answer(x) \leftarrow R(x)\ \text{AND}\ S(x)$
- 并集 $R \cup S$：$Answer(x) \leftarrow R(x)$，$Asnwer(x) \leftarrow R(x)$；
- 对称差 $R - S$：$Answer(x) \leftarrow R(x)\ \text{AND NOT}\ S(x)$；
- 投影 $\pi_A(R)$：$Answer(a) \leftarrow R(a, b)$；
- 选择
  - $\sigma_C(R)$：$Answer(x) \leftarrow R(x)\ \text{AND}\ C$；
  - $\sigma_{C_1\ \text{AND}\ C_2}(R)$：$Answer(x) \leftarrow R(x)\ \text{AND}\ C_1\ \text{AND}\ C_2$；
  - $\sigma_{C_1\ \text{OR}\ C_2}(R)$：$Answer(x) \leftarrow R(x)\ \text{AND}\ C_1$，$Answer(x) \leftarrow R(x)\ \text{AND}\ C_2$；
  - 更复杂的规则：分解为析取范式。
- 乘积 $R \times S$：$Answer(x, y) \leftarrow R(x)\ \text{AND}\ S(y)$；
- 自然连接 $R \bowtie S$：$Answer(x, y, z) \leftarrow R(x, y)\ \text{AND}\ S(y, z)$；
- θ-连接 $R \bowtie_\theta S$：自然连接后加选择即可。

Datalog 语言中的递归：Datalog 的规则可以描述递归，直接在 head 与 body 均使用内涵性谓词即可。

> 【例】给出关系 $Edge(X, Y)$，构造关系 $Path(X, Y)$ 表示 X 与 Y 之间存在路径连接，即
>
> $Path(X, Y) \leftarrow Edge(X, Y)$；
>
> $Path(X, Y) \leftarrow Edge(X, Z)\ \text{AND}\ Path(Z, Y)$。

Datalog 与关系代数模型 RA

- 没有递归的 Datalog 语言即为基本的 RA 模型；
- Datalog 不支持扩展 RA 中的聚合操作和分组操作；
- Datalog 目前不支持对 Bag 操作的操作子；
- 包含递归的 Datalog 比 RA 更强大；
- 他们都不具有完全的表达能力（图灵完备性）。