# 《DBMS》实验报告五

**报告人：**冷友方；王大伟；罗传文；赵丹；杨婧如.

**学院：**信息学院

**日期：**2016年12月22日

## 1 查询计划的形式

以 SELECT 查询语句来说明我们查询计划的形式，我们支持的整个 SQL 里面共有五部分，其中：

(1)先将<FROMLIST>里的所有关系进行笛卡尔积运算，若只有一个关系，则省去这步操作；(2)然后对生成的临时关系根据<CONDLIST>里面的选择条件逐个做选择操作；

(3)然后就根据 ORDERBY 里面的属性进行相应的排序操作；

(4)然后根据<SELLIST>进行相应的投影操作；

(5)如果语句里面有 DISTINCT，则最后的时候再做一次去除重复操作。

以如下查询为例，对下面两个表的查询：

|  |
| --- |
| student (id, name, birth);  score (id, course, score);  SELECT name, course, score  FROM student, score  WHERE student.id = score.id AND course = ‘RDBMS’  ORDER BY student.id; |

其生成的关系代数树如图1所示。我们可以看到，先是将 student 和 score 两个表进行 cross 操作，然后进行两步选择，最后根据 id 排序后进行投影操作。

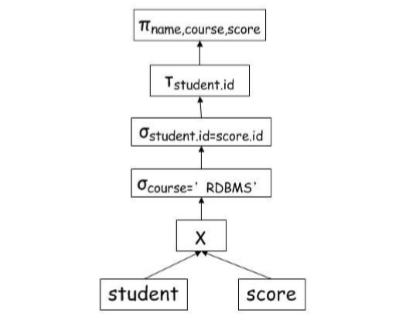


图1

可以看到，这是一个实际上并不太可行的查询计划，主要是考虑到 product(X)操作对时间和空间的需求太大。所以要对这个计划进行相应的优化，生成一个相对优化的逻辑查询计划。我们主要是应用选择下推的优化策略，对这个关系代数树进行了相应的优化。最后的形式如下图2所示。可以看出我们把一个查询分为相应的六个步骤：

（1）应用 WHERE 语句里面涉及单个表的条件，对这些表进行选择操作；

（2）对涉及两个表的条件，如果能进行等值连接，对它们进行等值连接；

（3）对这时候剩下的关系（如果这时候不是只剩一个关系）进行笛卡尔积；

（4）根据相应条件进行排序操作；

（5）投影；

（6）如果有 distinct，进行去除重复操作。

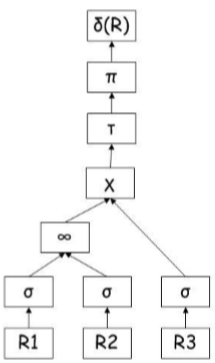


图2

## 2查询优化方法

在生成物理计划前，我们需要利用代数定律对逻辑计划进行优化，从而生成相对高效的物理计划。

实用优化规则有两个：1、选择下推；2、投影下推。(考虑到它的效果，未实现)

对于选择下推而言，我们的目的是使选择尽可能深的下推入表达式树中。如果一个选择条件是多个条件的 AND，则可以把该条件分解并分别将每个条件下推。这几乎是最有效的改进技术。每一次选择都很有可能减少结果集的数量，从而减少问题的规模，达到优化的目的。

我们的 SELECT 支持多个表上的多重条件查询，逻辑优化算法如下：以分解后 WHERE 子句中的各个查询条件分别进行选择下推；保存下推选择的结果，作为后续操作的输入，直至投影操作；选择下推时，优先考虑只涉及一个表上属性的条件，优先考虑等值条件。

具体说来，我们设计了两个队列，一个单表属性队列 sglist，一个双表属性队列 dulst，分别保存只涉及单表、涉及多表的查询条件。扫描 WHERE 子句中的所有条件，判断条件类型，分别插入到两个队列中，sglist 相对于 dulst 中的条件优先执行选择下推。在插入队列时，如果是等值条件，那么插入队列开头，选择下推操作被优先执行；其它条件插入队列的末尾，推后执行。

逻辑计划维护这两个队列，每次选择操作系统都自动生成一个临时表，保存选择的结果，以备后用；同时在队列中删除已经执行过选择下推的条件。

dulst 队列中的条件执行选择操作生成临时表之后，要更新 dulst 队列中剩余的条件，这个操作是由 update\_dul\_list()完成的。以新的临时模式取代原有条件中涉及的两个模式。此时，dulst 中的条件可能退化为只涉及单表属性的情况，需要将这些条件取出插入到 sglst 队列中去。

另外需要考虑的情况是，<FROMLIST>中可能出现<CONDLIST>中未涉及的模式，此时只能由卡氏积来表达了。在系统实现中由另外一个队列 tablst 来管理系统当前的模式。

在执行选择下推之前，将<FROMLIST>中的所有模式添加到 tablst 中；选择下推执行时动态维护这个队列，不断删除旧模式，添加新模式。在所有选择下推操作执行完毕之后，扫描tablst 队列，如果多余两个（含两个）模式，每次取出两个模式进行卡氏积操作，中间结果模式存入 tablst。

直至只剩一个模式，进行投影操作。

上面的例子经过逻辑优化后的关系代数树如下图3所示：

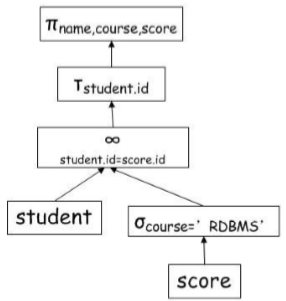


图3