103 透彻研究通过explain命令得到的SQL执行计划（4）

之前我们已经初步的对SQL执行计划有了一个了解了，现在开始，我们就来更加细致的探索一下执行计划的方方面面，把各种SQL语句的执行计划可能长什么样，都给大家分析出来，首先我们都知道，SQL执行计划里有一个id的概念。

这个id是什么意思呢？简单来说，有一个SELECT子句就会对应一个id，如果有多个SELECT那么就会对应多个id。但是往往有时候一个SELECT字句涉及到了多个表，所以会对应多条执行计划，此时可能多条执行计划的id是一样的。

接着我们来看看这个select_type，select_type之前我们似乎看到过几种，有什么SIMPLE的，还有primary和subquery的，那么这些select_type都是什么意思？除此之外，还有哪几种select_type呢？

首先要告诉大家的是，一般如果单表查询或者是多表连接查询，其实他们的select_type都是SIMPLE，这个之前大家也都看到过了，意思就是简单的查询罢了。

然后如果是union语句的话，就类似于select * from t1 union select * from t2，那么会对应两条执行计划，第一条执行计划是针对t1表的，select_type是PRIMARY，第二条执行计划是针对t2表的，select_type是UNION，这就是在出现union语句的时候，他们就不一样了。

我们之前给大家讲过，在使用union语句的时候，会有第三条执行计划，这个第三条执行计划意思是针对两个查询的结果依托一个临时表进行去重，这个第三条执行计划的select_type就是union_result。

另外，之前我们还看到过，如果是在SQL里有子查询，类似于select * from t1 where x1 in (select x1 ffrom t2) or x3='xxx'，此时其实会有两条执行计划，第一条执行计划的select_type是PRIMARY，第二条执行计划的select_type是SUBQUERY，这个我们之前也看到过了。

那么现在我们来看一个稍微复杂一点的SQL语句：

EXPLAIN SELECT * FROM t1 WHERE x1 IN (SELECT x1 FROM t2 WHERE x1 = 'xxx' UNION SELECT x1 FROM t1 WHERE x1 = 'xxx');

这个SQL语句就稍微有点复杂了，因为他有一个外层查询，还有一个内层子查询，子查询里还有两个SELECT语句进行union操作，那么我们来看看他的执行计划会是什么样的呢？

□+----+-------------------+------------+------------+------+---------------+----------+---------+-------+------+----------+-----------------------+

| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |

+----+-------------------+------------+------------+------+---------------+----------+---------+-------+------+----------+-----------------------+

| 1 | PRIMARY | t1 | NULL | ALL | NULL | NULL | NULL | NULL | 3467 | 100.00 | Using where |

| 2 | DEPENDENT SUBQUERY | t2 | NULL | ref | index_x1 | index_x1 | 899 | const | 59 | 100.00 | Using where; Using index |

| 3 | DEPENDENT UNION | t1 | NULL | ref | index_x1 | index_x1 | 899 | const | 45 | 100.00 | Using where; Using index |

| NULL | UNION RESULT | <union2,3> | NULL | ALL | NULL | NULL | NULL | NULL | NULL | Using temporary |

+----+-------------------+------------+------------+------+---------------+----------+---------+-------+------+----------+-----------------------+

第一个执行计划一看就是针对t1表查询的那个外层循环，select_type就是PRIMARY，因为这里涉及到了子查询，所以外层查询的select_type一定是PRIMARY了。

然后第二个执行计划是子查询里针对t2表的那个查询语句，他的select_type是DEPENDENT SUBQUERY，第三个执行计划是子查询里针对t1表的另外一个查询语句，select_type是DEPENDENT UNION，因为第三个执行计划是在执行union后的查询，第四个执行计划的select_type是UNION RESULT，因为在执行子查询里两个结果集的合并以及去重。

现在再来看一个更加复杂一点的SQL语句：

EXPLAIN SELECT * FROM (SELECT x1, count(*) as cnt FROM t1 GROUP BY x1) AS _t1 where cnt > 10;

这个SQL可有点麻烦了，他是FROM子句后跟了一个子查询，在子查询里是根据x1字段进行分组然后进行count聚合操作，也就是统计出来x1这个字段每个值的个数，然后在外层则是针对这个内层查询的结果集进行查询通过where条件来进行过滤，看看他的执行计划：

+----+-------------+------------+------------+-------+---------------+----------+---------+------+------+----------+-------------+

| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |

+----+-------------+------------+------------+-------+---------------+----------+---------+------+------+----------+-------------+

| 1 | PRIMARY | | NULL | ALL | NULL | NULL | NULL | NULL | 3468 | 33.33 | Using where |

| 2 | DERIVED | t1 | NULL | index | index_x1 | index_x1 | 899 | NULL | 3568 | 100.00 | Using index |

+----+-------------+------------+------------+-------+---------------+----------+---------+------+------+----------+-------------+

上面的执行计划里，我们其实应该先看第二条执行计划，他说的是子查询里的那个语句的执行计划，他的select_type是derived，意思就是说，针对子查询执行后的结果集会物化为一个内部临时表，然后外层查询是针对这个临时的物化表执行的。

大家可以看到，他这里执行分组聚合的时候，是使用的index_x1这个索引来进行的，type是index，意思就是直接扫描偶了index_x1这个索引树的所有叶子节点，把x1相同值的个数都统计出来就可以了。

然后外层查询是第一个执行计划，select_type是PRIMARY，针对的table是，就是一个子查询结果集物化形成的临时表，他是直接针对这个物化临时表进行了全表扫描根据where条件进行筛选的。

好，今天的执行计划就讲解到这里了，下次我们继续讲解。

End