

今天我们继续来讲解不同SQL语句的执行计划长什么样子，来一起看一个包含子查询的SQL语句的执行计划：

```
EXPLAIN SELECT * FROM t1 WHERE x1 IN (SELECT x1 FROM t2) OR x3 = 'xxxx';
```

这个SQL就稍微有一点点的复杂了，因为主SELECT语句的WHERE筛选条件是依赖于一个子查询的，而且除此之外还有一个自己的WHERE筛选条件，那么他的执行计划长什么样子呢？我们看看。

```
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key   | key_len | ref  | rows |
|    | filtered   |        |           |       |               |       |         |       |      |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY   | t1   | NULL    | ALL  | index_x3   | NULL  | NULL   | NULL | 3457 | 100.00 |
|    | Using where |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 2 | SUBQUERY  | t2   | NULL    | index | index_x1   | index_x1 | 507  | NULL | 4687 | 100.00 |
|    | Using index |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
```

这个执行计划值得我们好好分析一下，首先，第一条执行计划的id是1，第二条执行计划的id是2，这是为什么？因为这个SQL里有两个SELECT，主查询SELECT的执行计划的id就是1，子查询SELECT的执行计划的id就是2

其次，第一条执行计划里，select\_type是PRIMARY，不是SIMPLE了，说明第一个执行计划的查询类型是主查询的意思，对主查询而言，他有一个where条件是x3='xxx'，所以他的possible\_keys里包含了index\_x3，就是x3字段的索引，但是他的key实际是NULL，而且type是ALL，所以说他最后没选择用x3字段的索引，而是选择了全表扫描

这是为什么呢？其实很简单，可能他通过成本分析发现，使用x3字段的索引扫描xxx这个值，几乎就跟全表扫描差不多，可能x3这个字段的值几乎都是xxx，所以最后就选择还不如直接全表扫描呢。

接着第二条执行计划，他的select\_type是SUBQUERY，也就是子查询，子查询针对的是t2这个表，当然子查询本身就是一个全表查询，但是对主查询而言，会使用x1 in 这个筛选条件，他这里type是index，说明使用了扫描index\_x1这个x1字段的二级索引的方式，直接扫描x1字段的二级索引，来跟子查询的结果集做比对。

接着我们来看另外一个union的SQL语句：

```
EXPLAIN SELECT * FROM t1 UNION SELECT * FROM t2
```

这是一个典型的union语句，把两个表的查询结果合并起来，如果大家不理解union的意思，建议自己去网上查一下

那么他的执行计划是什么样的呢？

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	t1	NULL	ALL	NULL	NULL	NULL	NULL	3457	100.00	
2	UNION	t2	NULL	ALL	NULL	NULL	NULL	NULL	4687	100.00	NULL
											Using temporary
											Using temporary

这个执行计划的第一条和第二条很好理解对吧？两个SELECT字句对应两个id，就是分别从t1表和t2表里进行全表扫描罢了

接着第三条执行计划是什么呢？其实union字句默认的作用是把两个结果集合并起来还会进行去重，所以第三条执行计划干的是个去重的活儿。

所以上面他的table是<union 1,2>, 这就是一个临时表的表名，而且你看他的extra里，有一个using temporary, 也就是使用临时表的意思，他就是把结果集放到临时表里进行去重的，就这么个意思。当然，如果你用的是union all，那么就不会进行去重了。

好了，今天的讲解就先到这里，后续我们会继续用很大篇幅分析MySQL的执行计划，实际上这个执行计划可能是程序员最需要掌握的关于数据库的知识之一了。

**End**