

今天我们就一步一步的来讲解不同的SQL语句的执行计划长什么样子，先来看第一条SQL语句，特别的简单，就是：

```
explain select * from t1
```

就这么一个简单的SQL语句，那么假设他这个里面有大概几千条数据，此时执行计划看起来是什么样的？

```
+----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered
| Extra |
+----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE   | t1   | NULL    | ALL  | NULL      | NULL | NULL    | NULL | 3457 | 100.00 | NULL
|
+----+-----+-----+-----+-----+-----+-----+-----+
```

一起来分析一下上面的执行计划吧，学习当然得从最简单的地方开始一步一步得来，相信每个人都能成为分析SQL执行计划的高手。

首先呢，id是1，这个不用管他了，select_type是SIMPLE，这个先不说他什么意思，你要知道顾名思义，这个表的查询类型是很普通的、而且简单的就可以了。

table是t1，这还用说么？表名就是t1，所以意思就是这里要访问t1这个表。type是all，这就是我们之前提到的多种访问方式之一了，all就是全表扫描，这没办法，你完全没加任何where条件，那当然只能是全表扫描了！

而且如果大家记得我们之前讲解的底层访问方式，就会知道，这里直接会扫描表的聚簇索引的叶子节点，按顺序扫描过去拿到表里全部数据。

rows是3457，这说明全表扫描会扫描这个表的3457条数据，说明这个表里就有3457条数据，此时你全表扫描会全部扫描出来。filtered是100%，这个也很简单了，你没有任何where过滤条件，所以直接筛选出来的数据就是表里数据的100%占比。

怎么样，有没有觉得稍微对执行计划有点感觉了，似乎也没那么难是吧？因为有了之前内容的大量铺垫和积累，大家对SQL语句的底层执行原理本身已经有了一定的理解了，所以看执行计划就会很简单的。

接着再来看一个SQL语句的执行计划：

```
explain select * from t1 join t2
```

这是一个典型的多表关联语句，之前我们说过，这种关联语句，实际上会选择一个表先查询出来数据，接着遍历每一条数据去另外一个表里查询可以关联在一起的数据，然后关联起来，此时他的执行计划大概长下面这样子：

```
+----+-----+-----+-----+-----+-----+-----+-----+
-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered
| Extra |
+----+-----+-----+-----+-----+-----+-----+-----+
-----+
| 1 | SIMPLE   | t1  | NULL   | ALL  | NULL    | NULL | NULL  | NULL | 3457 | 100.00 | NULL |
| 1 | SIMPLE   | t2  | NULL   | ALL  | NULL    | NULL | NULL  | NULL | 4568 | 100.00 | Using
join buffer (Block Nested Loop) |
+----+-----+-----+-----+-----+-----+-----+-----+
```

这个执行计划就比较有意思了，因为是一个多表关联的执行计划。首先很明显，他的执行计划分为了两条，也就是会访问两个表，先看他如何访问第一个表的，针对第一个表就是t1，明显是先用ALL方式全表扫描他了，而且扫出了3457条数据。

接着对第二个表的访问，也就是t2表，同样是全表扫描，因为他这种多表关联方式，基本上是笛卡尔积的效果，t1表的每条数据都会去t2表全表扫描所有4568条数据，跟t2表的每一条数据都会做一个关联，而且extra里说了是Nested Loop，也就是嵌套循环的访问方式，跟我们之前讲解的关联语句的执行原理都是匹配的。

另外大家会发现上面两条执行计划的id都是1，是一样的，实际上一般来说，在执行计划里，一个SELECT会对应一个id，因为这两条执行计划对应的是一个SELECT语句，所以他们俩的id都是1，是一样。

如果你要是有一个子查询，有另外一个SELECT，那么另外一个SELECT子查询对应的执行计划的id就可能是2了。

好，那么今天我们讲解了一下单表查询和多表关联的执行计划长什么样子，接下来我们会讲解子查询之类的语句的执行计划，其实讲解执行计划的本质，就是用各种不同的SQL语句来给大家讲解他们的执行计划什么样子，大家看多了自然就知道了。

End