

97 MySQL是如何基于各种规则去优化执行计划的？（上）

之前我们已经给大家讲解了单表查询语句和多表关联语句具体的执行原理，同时也给大家讲了在生成具体执行计划的时候，是如何根据成本计算去选择最优执行计划的，因为每个查询执行的时候实际都可能有多种执行计划可供选择，必须要选择成本最低的那种。

接着我们来给大家讲解一下MySQL在执行一些相对较为复杂的SQL语句的时候是如何对查询进行重写来优化具体的执行计划的，因为他有时候可能会觉得你写的SQL一点都不好，直接按你的SQL生成的执行计划效率还是不够高，需要自动帮你改改。

这里有很多很多的规则，可能比较琐碎，但还是需要让大家了解一下MySQL可能会改写我们的SQL语句这个事，所以请大家耐着性子看下去，大概理解一下就行了

首先呢，要是MySQL觉得你的SQL里有很多括号，那么无关紧要的括号他会给你删除了，其次比如你有类似于 $i = 5 \text{ and } j > i$ 这样的SQL，就会改写为 $i = 5 \text{ and } j > 5$ ，做一个常量替换。

还有比如 $x = y \text{ and } y = k \text{ and } k = 3$ 这样的SQL，都会给你优化成 $x = 3 \text{ and } y = 3 \text{ and } k = 3$ ，本质也是做个常量替换。或者是类似于什么 $b = b \text{ and } a = a$ 这种一看就是乱写的SQL，一看就是没意义的，就直接给你删了。

大家可能觉得很琐碎，其实不是的，这些SQL的改写，你会发现，他本质都是在优化SQL语句的清晰语义，方便后续在索引和数据页里进行查找。

还有一些是比较有意思的改写，比如下面的SQL语句：

```
select * from t1 join t2 on t1.x1=t2.x1 and t1.id=1
```

这个SQL明显是针对t1表的id主键进行了查询，同时还要跟t2表进行关联，其实这个SQL语句就可能在执行前就先查询t1表的id=1的数据，然后直接做一个替换，把SQL替换为：

```
select t1表中id=1的那行数据的各个字段的常量值, t2.* from t1 join t2 on t1表里x1字段的常量值=t2.x1
```

上面的SQL就是直接把t1相关的字段都替换成了提前查出来的id=1那行数据的字段常量值了。

今天就先给大家讲一点开胃菜，也就是大家能知道一下其实你写的SQL语句真正执行时，可能是会对SQL进行各种改动的，接下来我们还会继续分析。

End