

以MySQL单表查询来举例，看看执行计划包含哪些内容（1）？

今天咱们就以MySQL单表查询来举例，看看执行计划到底包含哪些内容

其实只要大家跟着专栏一步一步的学习下来，会很轻松的看懂执行计划，但是如果你之前对什么数据页，索引，索引使用规则，这些东西学的不扎实，那你可能会觉得现在看着吃力，很痛苦，如果你觉得痛苦，那就回过头去看看之前的内容，学扎实一些。

今天和下次要讲解的执行计划包含的内容，说白了，全是之前讲过的，只不过我们把之前讲过的一些东西跟MySQL的执行计划中的一些概念匹配起来，这样衔接上之后，你就知道MySQL的执行计划里出现一些专业术语和名词的时候，对应的是底层的什么行为。

我们之前都知道，假设你写一个`select * from table where id=x`，或者`select * from table where name=x`的语句，直接就可以通过聚簇索引或者二级索引+聚簇索引回源，轻松查到你要的数据，这种根据索引直接可以快速查找数据的过程，在执行计划里称之为`const`，意思就是性能超高的常量级的。

所以你以后在执行计划里看到`const`的时候，就知道他就是直接通过索引定位到数据，速度极快，这就是`const`的意思。

但是这里有一个要点，你的二级索引必须是唯一索引，才是属于`const`方式的，也就是说你必须建立`unique key`唯一索引，保证一个二级索引的每一个值都是唯一的，才可以。

那么如果你是一个普通的二级索引呢？就是个普通的KEY索引，这个时候如果你写一个`select * from table where name=x`的语句，`name`是个普通二级索引，不是唯一索引，那么此时这种查询速度也是很慢的，他在执行计划里叫做`ref`。

如果你是包含多个列的普通索引的话，那么必须是从索引最左侧开始连续多个列都是等值比较才可以是属于`ref`方式，就是类似于`select * from table where name=x and age=x and xx=xx`，然后索引可能是一个`KEY(name,age,xx)`。

然后一个例外，就是如果你用`name IS NULL`这种语法的话，即使`name`是主键或者唯一索引，还是只能走`ref`方式。但是如果你是针对一个二级索引同时比较了一个值还有限定了`IS NULL`，类似于`select * from table where name=x and name IS NULL`，那么此时在执行计划里就叫做`ref_or_null`

说白了，就是在二级索引里搜你要的值以及是`NULL`的值，然后再回源去聚簇索引里查罢了，因为同时有索引等值比较和`NULL`值查询，就叫做`ref_or_null`了，其实也没啥。

那这个ref就说完了，到这里大家先停顿一下，稍微来点深度思考，我们换个角度看，假设你以后在分析执行计划的时候看到了const，那是什么？对，肯定是通过主键或者唯一索引的访问，速度超高。

如果你看到了ref是什么意思？对，就是用了普通的索引，或者用主键/唯一索引搞了一个IS NULL/IS NOT NULL。

所以说，我们别急着去看后续的内容，先思考一下，以后你在执行计划里看到const和ref，是不是立马就知道他们底层都是基于什么方式来查询的，然后如果反映到之前画的很多图里，是如何查询那些索引的。

不管怎么说，只要你看到const或者ref，那恭喜你，说明起码这部分执行速度是很快的！而且相信大家结合之前的内容思考一下，立马就知道那部分查询是怎么做的，底层是通过哪些索引怎么查询的，这个之前都讲过了，下一次我们继续看执行计划里可能有的其他部分。

End