

# mysql explain详解

1)、id列数字越大越先执行，如果说数字一样大，那么就从上往下依次执行，id列为null的就表这是这是一个结果集，不需要使用它来进行查询。

2)、select\_type列常见的有：

- A: simple: 表示不需要union操作或者不包含子查询的简单select查询。有连接查询时，外层的查询为simple，且只有一个
- B: primary: 一个需要union操作或者含有子查询的select，位于最外层的单位查询的select\_type即为primary。且只有一个
- C: union: union连接的两个select查询，第一个查询是dervied派生表，除了第一个表外，第二个以后的表select\_type都是union
- D: dependent union: 与union一样，出现在union 或union all语句中，但是这个查询要受到外部查询的影响
- E: union result: 包含union的结果集，在union和union all语句中,因为它不需要参与查询，所以id字段为null
- F: subquery: 除了from字句中包含的子查询外，其他地方出现的子查询都可能是subquery
- G: dependent subquery: 与dependent union类似，表示这个subquery的查询要受到外部表查询的影响
- H: derived: from字句中出现的子查询，也叫做派生表，其他数据库中可能叫做内联视图或嵌套select

3)、table

显示的查询表名，如果查询使用了别名，那么这里显示的是别名，如果不涉及对数据表的操作，那么这显示为null，如果显示为尖括号括起来的<derived N>就表示这个是临时表，后边的N就是执行计划中的id，表示结果来自于这个查询产生。如果是尖括号括起来的<union M,N>，与<derived N>类似，也是一个临时表，表示这个结果来自于union查询的id为M,N的结果集。

4)、type

依次从好到差：

system, const, eq\_ref, ref, fulltext, ref\_or\_null, unique\_subquery, index\_subquery, range, index\_merge, index, ALL，除了all之外，其他的type都可以使用到索引，除了index\_merge之外，其他的type只可以用到一个索引

- A: system: 表中只有一行数据或者是空表，且只能用于myisam和memory表。如果是Innodb引擎表，type列在这个情况通常都是all或者index
- B: const: 使用唯一索引或者主键，返回记录一定是1行记录的等值where条件时，通常type是const。其他数据库也叫做唯一索引扫描
- C: eq\_ref: 出现在要连接过个表的查询计划中，驱动表只返回一行数据，且这行数据是第二个表的主键或者唯一索引，且必须为not null，唯一索引和主键是多列时，只有所有的列都用作比较时才会出现eq\_ref
- D: ref: 不像eq\_ref那样要求连接顺序，也没有主键和唯一索引的要求，只要使用相等条件检索时就可能出现，常见与辅助索引的等值查找。或者多列主键、唯一索引中，使用第一个列之外的列作为等值查找也会出现，总之，返回数据不唯一的等值查找就可能出现。
- E: fulltext: 全文索引检索，要注意，全文索引的优先级很高，若全文索引和普通索引同时存在时，mysql不管代价，优先选择使用全文索引
- F: ref\_or\_null: 与ref方法类似，只是增加了null值的比较。实际用的不多。
- G: unique\_subquery: 用于where中的in形式子查询，子查询返回不重复值唯一值
- H: index\_subquery: 用于in形式子查询使用到了辅助索引或者in常数列表，子查询可能返回重复值，可以使用索引将子查询去重。
- I: range: 索引范围扫描，常见于使用>,<,is null,between ,in ,like等运算符的查询中。
- J: index\_merge: 表示查询使用了两个以上的索引，最后取交集或者并集，常见and，or的条件使用了不同的索引，官方排序这个在ref\_or\_null之后，但是实际上由于要读取所个索引，性能可能大部分时间都不如range
- K: index: 索引全表扫描，把索引从头到尾扫一遍，常见于使用索引列就可以处理不需要读取数据文件的查询、可以使用索引排序或者分组的查询。
- L: all: 这个就是全表扫描数据文件，然后再在server层进行过滤返回符合要求的记录。

5)、possible\_keys

查询可能使用到的索引都会在这里列出来

6)、key

查询真正使用到的索引，select\_type为index\_merge时，这里可能出现两个以上的索引，其他的select\_type这里只会出现一个。

7)、key\_len

用于处理查询的索引长度，如果是单列索引，那就整个索引长度算进去，如果是多列索引，那么查询不一定都能使用到所有的列，具体使用到了多少个列的索引，这里就会计算进去，没有使用到的列，这里不会计算进去。留意下这个列的值，算一下你的多列索引总长度就知道有没有使用到所有的列了。要注意，mysql的ICP特性使用到的索引不会计入其中。另外，key\_len只计算where条件用到的索引长度，而排序和分组就算用到了索引，也不会计算到key\_len中。

8)、ref

如果是使用的常数等值查询，这里会显示const，如果是连接查询，被驱动表的执行计划这里会显示驱动表的关联字段，如果是条件使用了表达式或者函数，或者条件列发生了内部隐式转换，这里可能显示为func

## 9)、rows

这里是执行计划中估算的扫描行数，不是精确值

## 10)、extra

这个列可以显示的信息非常多，有几十种，常用的有

A: distinct: 在select部分使用了distinct关键字

B: no tables used: 不带from字句的查询或者From dual查询

C: 使用not in()形式子查询或not exists运算符的连接查询，这种叫做反连接。即，一般连接查询是先查询内表，再查询外表，反连接就是先查询外表，再查询内表。

D: using filesort: 排序时无法使用到索引时，就会出现这个。常见于order by和group by语句中

E: using index: 查询时不需要回表查询，直接通过索引就可以获取查询的数据。

F: using join buffer (block nested loop), using join buffer (batched key access): 5.6.x之后的版本优化关联查询的BNL, BKA特性。主要是减少内表的循环数量以及比较顺序地扫描查询。

G: using sort\_union, using\_union, using intersect, using sort\_intersection:

using intersect: 表示使用and的各个索引的条件时，该信息表示是从处理结果获取交集

using union: 表示使用or连接各个使用索引的条件时，该信息表示从处理结果获取并集

using sort\_union和using sort\_intersection: 与前面两个对应的类似，只是他们是出现在用and和or查询信息量大时，先查询主键，然后进行排序合并后，才能读取记录并返回。

H: using temporary: 表示使用了临时表存储中间结果。临时表可以是内存临时表和磁盘临时表，执行计划中看不出来，需要查看status变量，used\_tmp\_table, used\_tmp\_disk\_table才能看出来。

I: using where: 表示存储引擎返回的记录并不是所有的都满足查询条件，需要在server层进行过滤。查询条件中分为限制条件和检查条件，5.6之前，存储引擎只能根据限制条件扫描数据并返回，然后server层根据检查条件进行过滤再返回真正符合查询的数据。5.6.x之后支持ICP特性，可以把检查条件也下推到存储引擎层，不符合检查条件和限制条件的数据，直接不读取，这样就大大减少了存储引擎扫描的记录数量。extra列显示using index condition

J: firstmatch(tb\_name): 5.6.x开始引入的优化子查询的新特性之一，常见于where字句含有in()类型的子查询。如果内表的数据量比较大，就可能出现这个

K: loosescan(m..n): 5.6.x之后引入的优化子查询的新特性之一，在in()类型的子查询中，子查询返回的可能有重复记录时，就可能出现这个

除了这些之外，还有很多查询数据字典库，执行计划过程中就发现不可能存在结果的一些提示信息

## 11)、filtered

使用explain extended时会出现这个列，5.7之后的版本默认就有这个字段，不需要使用explain extended了。这个字段表示存储引擎返回的数据在server层过滤后，剩下多少满足查询的记录数量的比例，注意是百分比，不是具体记录数。

参考：<http://www.cnblogs.com/xiaoboluo768/p/5400990.html>

<http://blog.csdn.net/u010061060/article/details/52473244>

<http://www.cnblogs.com/hustcat/archive/2009/10/28/1591648.html>