explain显示了mysql如何使用索引来处理select语句以及连接表。可以帮助选择更好的索引和写出更优化的查询语句。

虽然这篇文章我写的很长，但看起来真的不会困啊，真的都是干货啊！！！！

先解析一条sql语句，看出现什么内容

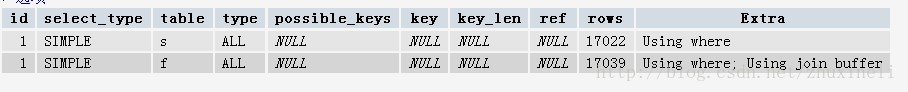
EXPLAIN SELECT s.uid,s.username,s.name,f.email,f.mobile,f.phone,f.postalcode,f.address

FROM uchome\_space AS s,uchome\_spacefield AS f

WHERE 1

AND s.groupid=0

AND s.uid=f.uid

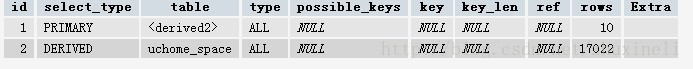


**1. id**

SELECT识别符。这是SELECT查询序列号。这个不重要,查询序号即为sql语句执行的顺序，看下面这条sql

EXPLAINSELECT \*FROM (SELECT\* FROMuchome\_space LIMIT 10)AS s

它的执行结果为



可以看到这时的id变化了

**2.select\_type**

select类型，它有以下几种值

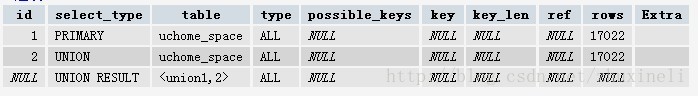
2.1 simple 它表示简单的select,没有union和子查询

2.2 primary 最外面的select,在有子查询的语句中，最外面的select查询就是primary,上图中就是这样

2.3 union union语句的第二个或者说是后面那一个.现执行一条语句，explain

select  \*  from uchome\_space limit 10 union select \* from uchome\_space limit 10,10

会有如下结果



第二条语句使用了union

2.4 dependent union    UNION中的第二个或后面的SELECT语句，取决于外面的查询

2.5 union result        UNION的结果,如上面所示

还有几个参数，这里就不说了，不重要

**3 table**

输出的行所用的表，这个参数显而易见，容易理解

**4 type**

**连接类型。**有多个参数，先从最佳类型到最差类型介绍 **重要且困难**

4.1 system

表仅有一行，这是const类型的特列，平时不会出现，这个也可以忽略不计

4.2 const

表最多有一个匹配行，const用于比较primary key 或者unique索引。因为只匹配一行数据，所以很快

记住一定是用到primary key 或者unique，并且只检索出两条数据的 情况下才会是const,看下面这条语句

explain SELECT \* FROM `asj\_admin\_log` limit 1,结果是

IMG_259

虽然只搜索一条数据，但是因为没有用到指定的索引，所以不会使用const.继续看下面这个

explain SELECT \* FROM `asj\_admin\_log` where log\_id = 111

IMG_260

log\_id是主键，所以使用了const。所以说可以理解为const是最优化的

4.3 eq\_ref

对于eq\_ref的解释，mysql手册是这样说的:"对于每个来自于前面的表的行组合，从该表中读取一行。这可能是最好的联接类型，除了const类型。它用在一个索引的所有部分被联接使用并且索引是UNIQUE或PRIMARY KEY"。eq\_ref可以用于使用=比较带索引的列。看下面的语句

explain select \* from uchome\_spacefield,uchome\_space where uchome\_spacefield.uid = uchome\_space.uid

得到的结果是下图所示。很明显，mysql使用eq\_ref联接来处理uchome\_space表。

IMG_261

目前的疑问：

       4.3.1 为什么是只有uchome\_space一个表用到了eq\_ref,并且sql语句如果变成

       explain select \* from uchome\_space,uchome\_spacefield where uchome\_space.uid = uchome\_spacefield.uid

       结果还是一样，需要说明的是uid在这两个表中都是primary

4.4 ref 对于每个来自于前面的表的行组合，所有有匹配索引值的行将从这张表中读取。如果联接只使用键的最左边的前缀，或如果键不是UNIQUE或PRIMARY KEY（换句话说，如果联接不能基于关键字选择单个行的话），则使用ref。如果使用的键仅仅匹配少量行，该联接类型是不错的。

看下面这条语句 explain select \* from uchome\_space where uchome\_space.friendnum = 0，得到结果如下，这条语句能搜出1w条数据

IMG_262

4.5 ref\_or\_null 该联接类型如同ref，但是添加了MySQL可以专门搜索包含NULL值的行。在解决子查询中经常使用该联接类型的优化。

上面这五种情况都是很理想的索引使用情况

4.6 index\_merge 该联接类型表示使用了索引合并优化方法。在这种情况下，key列包含了使用的索引的清单，key\_len包含了使用的索引的最长的关键元素。

4.7 unique\_subquery

4.8 index\_subquery

4.9 range 给定范围内的检索，使用一个索引来检查行。看下面两条语句

explain select \* from uchome\_space where uid in (1,2)

explain select \* from uchome\_space where groupid in (1,2)

uid有索引，groupid没有索引，结果是第一条语句的联接类型是range,第二个是ALL.以为是一定范围所以说像 between也可以这种联接,很明显

explain select \* from uchome\_space where friendnum = 17

这样的语句是不会使用range的，它会使用更好的联接类型就是上面介绍的ref

4.10 index     该联接类型与ALL相同，除了只有索引树被扫描。这通常比ALL快，因为索引文件通常比数据文件小。（也就是说虽然all和Index都是读全表，但index是从索引中读取的，而all是从硬盘中读的）

当查询只使用作为单索引一部分的列时，MySQL可以使用该联接类型。

4.11  ALL  对于每个来自于先前的表的行组合，进行完整的表扫描。如果表是第一个没标记const的表，这通常不好，并且通常在它情况下***很***差。通常可以增加更多的索引而不要使用ALL，使得行能基于前面的表中的常数值或列值被检索出。

**5 possible\_keys** 提示使用哪个索引会在该表中找到行，不太重要

**6 keys** MYSQL使用的索引，简单且重要

**7 key\_len** MYSQL使用的索引长度

**8 ref** ref列显示使用哪个列或常数与key一起从表中选择行。

**9 rows** 显示MYSQL执行查询的行数，简单且重要，数值越大越不好，说明没有用好索引

**10 Extra**  该列包含MySQL解决查询的详细信息。

10.1 Distinct MySQL发现第1个匹配行后，停止为当前的行组合搜索更多的行。一直没见过这个值

10.2 Not exists

10.3 range checked for each record

没有找到合适的索引

10.4 using filesort

MYSQL手册是这么解释的“MySQL需要额外的一次传递，以找出如何按排序顺序检索行。通过根据联接类型浏览所有行并为所有匹配WHERE子句的行保存排序关键字和行的指针来完成排序。然后关键字被排序，并按排序顺序检索行。”目前不太明白

10.5 using index 只使用索引树中的信息而不需要进一步搜索读取实际的行来检索表中的信息。这个比较容易理解，就是说明是否使用了索引

explain select \* from ucspace\_uchome where uid = 1的extra为using index（uid建有索引）

explain select count(\*) from uchome\_space where groupid=1 的extra为using where(groupid未建立索引)

10.6 using temporary

为了解决查询，MySQL需要创建一个临时表来容纳结果。典型情况如查询包含可以按不同情况列出列的GROUP BY和ORDER BY子句时。

出现using temporary就说明语句需要优化了，举个例子来说

EXPLAIN SELECT ads.id FROM **ads**, city WHERE   city.city\_id = 8005   AND ads.status = 'online'   AND city.ads\_id=ads.id **ORDER BY ads.id desc**

**id  select\_type  table   type    possible\_keys   key      key\_len  ref                     rows  filtered  Extra**

**------  -----------  ------  ------  --------------  -------  -------  --------------------  ------  --------  -------------------------------**

**1  SIMPLE** **city   ref     ads\_id,city\_id  city\_id  4        const                   2838    100.00 Using temporary; Using filesort**

**1  SIMPLE       ads     eq\_ref  PRIMARY         PRIMARY  4        city.ads\_id       1    100.00  Using where**

**这条语句会使用using temporary,而下面这条语句则不会**

EXPLAIN SELECT ads.id FROM **ads**, city WHERE   city.city\_id = 8005   AND ads.status = 'online'   AND city.ads\_id=ads.id **ORDER BYcity.ads\_id desc**