

```
mysql> explain select bid from book where typeid in(2,3) and authorid=1 order by typeid desc ;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | book | NULL | range | idx_tab | idx_tab | 4 | NULL | 3 | 25.00 | Using where; Using index |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.00 sec)

mysql> explain select bid from book where typeid in(2,3) order by typeid desc ;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | book | NULL | range | idx_tab | idx_tab | 4 | NULL | 3 | 100.00 | Using where; Using index |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.00 sec)

mysql> explain select bid from book where authorid=1 order by typeid desc ;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | book | NULL | index | NULL | idx_tab | 12 | NULL | 4 | 25.00 | Using where; Using index |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.00 sec)
```

mysql中 执行计划的extra字段---- using where , using index 和 using where & using index 整理

原创 jeffrey11223 最后发布于2018-01-19 11:43:23 阅读数 7339 收藏 展开

先上表结构:

```
CREATE TABLE table (
xxx varchar(20) NOT NULL,
yyy varchar(20) NOT NULL,
zzz datetime NOT NULL,
aaa varchar(10) NOT NULL,
PRIMARY KEY (xxx, yyy, zzz),
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

先看一下表的索引，这是一个基于主键的联合聚集索引  
执行 show index from table

```
Table Non_unique Key_name Seq_in_index Column_name Collation
Cardinality Sub_part Packed Null Index_type Comment Index_comment
table 0 PRIMARY 1 xxx A 1012 NULL NULL BTREE
table 0 PRIMARY 2 yyy A 1012 NULL NULL BTREE
table 0 PRIMARY 3 zzz A 11134 NULL NULL BTREE
```

-----  
-----  
  
`explain select xxx from table`  
`explain select yyy from table`  
`explain select zzz from table`

这三个的执行计划，extra字段是using index，意思就是索引覆盖，查询的内容可以直接在索引中拿到。

`explain select aaa from table where xxx=' something'`

查询内容不在索引内，where条件为索引最左列，extra是using where，type是ref，表明虽然用到了索引，但是没有索引覆盖，产生了回表。

`explain select yyy from table where xxx=' something'`

查询内容在索引内，where条件为索引最左列，extra是using where using index，type是ref，表明用到了索引，索引也覆盖了，using where代表发生了过滤，网上有不少文章误把using where和是否读取索引进行关联，是不正确的，请参考此链接：<https://segmentfault.com/q/1010000003094577/a-1020000003697411>

`explain select xxx from table where zzz=' 2018-01-19 00:00:00'`

查询内容在索引内，where条件为索引第三列，这次不是最左列了，extra依旧是using where using index，但是type变成了index，这表明发生了索引扫描，因为where条件不是最左列的缘故，其性能肯定是差于ref的。

`explain select xxx from table where yyy= 'some' and zzz=' 2018-01-19 00:00:00'`

结果同上一条，因为where条件没有最左列

```
explain select xxx from table where xxx=' something' and yyy=
'some' and zzz=' 2018-01-19 00:00:00'
```

查询内容在索引内，where条件为联合索引的全部列，并且最左列在第一个，extra是using index，type是const，代表引擎根据主键直接取出了唯一的数据，因为是唯一的数据，所以没有using where。可以想像这句sql速度应该是相当快的，联合索引这样用可以发挥最大的功效。

```
explain select aaa from table where xxx=' something' and yyy=
'some' and zzz=' 2018-01-19 00:00:00'
```

此句和上一句的区别就是查询内容不在索引内，type还是const，但是extra变成了null，理由很简单，因为没有索引覆盖，回表拿数据了。

```
explain select aaa from table where zzz=' 2018-01-19 00:00:00'
explain select aaa from table where yyy=' some'
explain select xxx from table where aaa= 'any'
```

这三个的extra都为using where，type是all，代表是全表扫描，既然是全表扫描，那就肯定没using index什么事了，using where说明在全表扫描后发生了过滤。

前两个走全表扫描的原因应该是查询内容不在索引范围内，且where条件没有最左列，所以引擎选择了走全表扫描。

最后一个应该是where条件不是索引，所以走全表扫描。

```
explain select aaa from table where xxx=' something'
```

查询内容不在索引内，where条件是索引最左列，extra为using where，type是ref，因为where的索引列是有序的，所以走了ref，又因为查询内容不在索引内，所以没有using index，因为产生了过滤，所以有using where。

以上对extra字段的using where，using index 和 using where & using index，还有null的情况进行了整理，并对type为all，ref和index的原因进行了说明，自己在整理的同时，也加深了印象和理解。

---

版权声明：本文为CSDN博主「jeffrey11223」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/jeffrey11223/article/details/79100761>