



链滴

mysql 索引优化注意这两项

作者: [selfjt](#)

原文链接: <https://ld246.com/article/1573037979364>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



先看以下例子假设在员工表(emp)的(ENAME, JOB, MGR)三个字段上建了一个索引，例如索引名叫IDX_1。三个字段分别为员工姓名、工作和所属经理号。然后，写如下一个查询语句，并不断进行查询条件和次序的排列组合，例如：

```
Select * from emp where ENAME='a' and JOB='b' and MGR=3;
Select * from emp where JOB='b' and MGR=3 and ENAME='a';
Select * from emp where JOB='b' and ENAME='a' and MGR=3;
Select * from emp where JOB='b' and MGR=3;
Select * from emp where ENAME='a' and MGR=3;
Select * from emp where ENAME='a';
Select * from emp where JOB='b';
Select * from emp where MGR=3;
```

在各种条件组合情况下，刚才建的索引(IDX_1)是用还是不用？也就是说对emp表的访问是全表扫描还是按索引(IDX_1)访问？

答案是只要有ENAME=' a' 条件，就能用上索引(IDX_1)，而不是全表扫描。创建复合索引时一定要考虑到索引的前缀性否则会由于没有前缀列在检索条件中导致的全表扫描。

索引的前缀性指的是**必须用到索引的第一个字段**。

索引的可选择性，指的是**不重复的索引值（基数）和表记录数的比值**。可选择性是索引筛选能力的一指标。当可选择性越大，索引价值也就越大。

如一张订单表order记录为10万条，表中user_id列的不重复值为10000，order_date列不重复值为100，则创建在user_id上创建索引的查询效率要比在order_date上创建索引的查询效率高。这是因为，段值越多，可选性越强，按照索引查询后需要定位的记录越少，查询效率越高。

数据库最常用的索引为B树索引(不同的数据库实现稍有不同，例如：oracle创建的是B*树，mysql是B树)，不同数据库可能还有自己特有的索引如：oracle的位图索引，mysql的hash索引等等。这里我们讨论常用的B树索引。下面给出几条创建B树索引时设计单字段索引和复合索引的建议

1. 分析SQL语句中的约束条件字段，如果约束条件字段比较固定，则优先考虑创建针对多字段 **复合索引**。例如同时涉及到多个字段的条件，则可以考虑建立一个复合索引。
2. 如果单个字段是主键或 **唯一字段**，或者**可选性非常高的字段**，尽管约束条件字段比较固定，**也不要建成复合索引，可建成单字段索引**，降低复合索引开销。
3. 在复合索引设计中，需首先考虑复合索引第一个设计原则：**复合索引的前缀性**。即SQL语句中，有复合索引的第一个字段作为约束条件，该复合索引才会启用。
4. 在复合索引设计中，**其次应考虑复合索引的可选性**。即按可选性高低，进行复合索引字段的**排序**。
5. 如果条件涉及的字段不固定，组合比较灵活，则分别为 **不同的列建立单字段索引**。
6. 如果是多表连接SQL语句，注意是否可以在被驱动表（drived table）的连接字段与该表的其它约束条件字段上，创建**复合索引**。
7. 通过多种SQL分析工具，分析执行计划并以量化形式评估效果。

1.在索引上使用表达式或者计算

```
select user_name from user where age -30 = 0  
select user_name from user where age = 30
```

虽然age列上创建了索引，但是第一条语句依然是会按照全表扫描来执行的

2.存在隐式数据类型转换(查询时候注意数据类型)

```
select user_name from user where age = '30'
```

age类型为int类型，但是在查询时就会被存在隐式类型转换,这种情况下,age上的索引类型是不起作用
特别注意

3.数据可选性不高

例如一个用户表中1000w条数据,给gender(性别)建立索引,因为性别只有男女,这也不会走索引,原因是据索引查找出的结果集依然很大，查询效率还不如全表扫描的效率，这是数据库就会执行全表扫描。

4.忽略的索引的前缀性

执行语句时，忽略了索引的前置性，则执行语句时是不会走索引的。使用is null或者is not null，null并没有被定义，所以索引会被抑制。

- 1.使用占位符进行查询（因为mysql查询时进行hash缓存，使用占位符这样缓存的命中率比较高。）
- 2.执行计划，在不同的场景我们根据条件选择不同最优索引查询策略

□ □ □ □ □ □ □ □ □ □ □