

索引使用场景分析

哪些情况需要创建索引

1. 主键自动建立唯一索引
2. 频繁作为查询条件的字段应该创建索引
3. 多表关联查询中，关联字段应该创建索引 on 两边都要创建索引
4. 查询中排序的字段，应该创建索引

哪些情况不需要创建索引

表记录太少

经常进行增删改

操作的表频繁更新的字段

where条件里使用频率不高的字段

为什么使用组合索引

为了节省mysql索引存储空间以及提升搜索性能，可建立组合索引（能使用组合索引就不使用单列索引）

索引失效

查看执行计划

介绍

MySQL 提供了一个 **EXPLAIN** 命令, 它可以对 **SELECT** 语句的执行计划进行分析, 并输出 SELECT 执行的详细信息, 以供开发人员针对性优化。

使用explain这个命令来查看一个这些SQL语句的执行计划，查看该SQL语句有没有使用上了索引，有没有做全表扫描，这都可以通过explain命令来查看。

可以通过explain命令深入了解MySQL的基于开销的优化器，还可以获得很多可能被优化器考虑到的访问策略的细节，以及当运行SQL语句时哪种策略预计会被优化器采用。

EXPLAIN 命令用法十分简单, 在 SELECT 语句前加上 explain 就可以了, 例如:

```
mysql> explain select * from user;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | type | possible_keys | key | key_len | ref | rows | Extra |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1  | SIMPLE     | user  | ALL  | NULL          | NULL | NULL    | NULL | 7    | NULL  |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

参数说明

explain出来的信息有10列，分别是

id、select_type、table、type、possible_keys、key、key_len、ref、rows、Extra

select_type (重要)

查询类型，主要用于区别**普通查询**、**联合查询(union、union all)**、**子查询**等复杂查询。

type (重要)

- 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

- 注意事项：

最少要索引使用到range级别。

索引失效分析

1. 全值匹配我最爱
2. 最佳左前缀法则
3. 不在索引列上做任何操作（计算、函数、(自动or手动)类型转换），会导致索引失效而转向全表扫描
4. 存储引擎不能使用索引中范围条件右边的列
5. 尽量使用覆盖索引(只访问索引的查询(索引列和查询列一致))，减少select *
6. mysql 在使用不等于(!= 或者<>)的时候无法使用索引会导致全表扫描
7. is null ,is not null 也无法使用索引
8. like以通配符开头('%abc...')mysql索引失效会变成全表扫描的操作
9. 字符串不加单引号索引失效
10. 少用or,用它来连接时会索引失效

<http://blog.csdn.net/wuseyukui>

MySQL锁介绍

- 按照锁的粒度来说，MySQL主要包含三种类型（级别）的锁定机制：

- 全局锁：锁的是整个database。由MySQL的SQL layer层实现的
- 表级锁：锁的是某个table。由MySQL的SQL layer层实现的
- 行级锁：锁的是某行数据，也可能锁定行之间的间隙。由某些存储引擎实现，比如InnoDB。

- 按照锁的功能来说分为：**共享读锁**和**排他写锁**。
- 按照锁的实现方式分为：**悲观锁**和**乐观锁**（使用某一版本列或者唯一列进行逻辑控制）
- 表级锁和行级锁的区别：

表级锁：开销小，加锁快；不会出现死锁；锁定粒度大，发生锁冲突的概率最高，并发度最低；

行级锁：开销大，加锁慢；会出现死锁；锁定粒度最小，发生锁冲突的概率最低，并发度也最高；

MySQL表级锁

表级锁介绍

由MySQL SQL layer层实现

- MySQL的表级锁有两种：

一种是表锁。

一种是元数据锁 (meta data lock, MDL)。

- MySQL 实现的表级锁定的争用状态变量：

```
mysql> show status like 'table%';
```

- table_locks_immediate: 产生表级锁定的次数；
- table_locks_waited: 出现表级锁定争用而发生等待的次数；

表锁介绍

- 表锁有两种表现形式：

表共享读锁 (Table Read Lock)

表独占写锁 (Table Write Lock)

元数据锁介绍

MDL不需要显式使用，在访问一个表的时候会被自动加上。MDL的作用是，保证读写的正确性。你可以想象一下，如果一个查询正在遍历一个表中的数据，而执行期间另一个线程对这个表结构做变更，删了一列，那么查询线程拿到的结果跟表结构对不上，肯定是不行的。