

mysql 索引(mysql index) - luketty的专栏 - 博客频道

分类：

Mysql (6)

版权声明：本文为博主原创文章，未经博主允许不得转载。

以下是查阅《mysql_administrators_bible》的摘录和总结

索引 [MySQL](#) index

索引是一种数据结构，可以是B-tree, R-tree, 或者 hash 结构。其中R-tree 常用于查询比较接近的数据；B-trees适合用于查找某范围内的数据，可以很快的从当前数据找到下条数据；hash结构则适用于随机访问的场合，查找每条数据的时间几乎相同。显然，若要查找某个时间段的数据，用B-tree结构要比hash结构快好多。

优化查询的有效方法是为经常查询的字段建立索引，如无索引查询数据时，会遍历整张表(多么恐怖啊);若有了索引查找会容易很多。当进行 UPDATE, DELETE, 以及 INSERT 操作时，mysql会自动更新索引信息。

1.创建和删除索引(Creating and dropping indexes)创建：

1. mysql> USE sakila;
2. Database changed
3. mysql> CREATE INDEX idx_actor_first_name ON actor (first_name);
4. 或者
5. mysql> ALTER TABLE actor ADD INDEX idx_actor_first_name (first_name);

PS:当用 create index 创建索引时,必须指定索引的名字，否则mysql会报错；

用 ALTER TABLE 创建索引时，可以不指定索引名字，若不指定mysql会自动生成索引名字

建立索引时，若不想用存储引擎的默认索引类型，可以指定索引的类型:

3.索引冗余

如果两个或者多个索引包含了相同的索引信息，那么就存在索引冗余。不同类型的索引有不同的索引冗余判断:

(1) SPATIAL 索引

SPATIAL 只能是一个简单索引，不能说复合索引；存在冗余的情况是同一个字段有两个索

引。

(2) FULLTEXT 索引

若一个FULLTEXT索引是另一个索引的子集(不考虑字段顺序),则存在冗余。

例如: 表中包含如下两个索引

- (field1)
- (field1, field2)

(3) HASH 索引

若一个索引在不考虑字段顺序的情况下,索引字段相同,则存在冗余。

例如: 表中包含如下两个索引

- (field1, field2, field3)
- (field1, field3, field2)

(4) BTREE 索引

若一个索引是另一个索引的子集(考虑字段顺序),则存在冗余。

例如: 表中包含如下两个索引

- (field2)
- (field2, field1)PS: 若两个或多个索引有相同的索引字段和字段顺序,但是有不同的索引类型,这样的索引是不冗余的。因为每种索引类型有自己的索引值。

顶