链滴

MySQL 索引类型总结和使用技巧以及注意事项

作者：huihui
原文链接：https://ld246.com/article/1504162931548
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)

转自 脚本之家

MySQL索引类型包括：

一、普通索引

这是最基本的索引，它没有任何限制。它有以下几种创建方式：

1.创建索引

CREATE INDEX indexName ON mytable(username(length));

如果是CHAR，VARCHAR类型，length可以小于字段实际长度；如果是BLOB和TEXT类型，必须指定ength，下同。

2.修改表结构

ALTER mytable ADD INDEX [indexName] ON (username(length)) -- 创建表的时候直接指定

CREATE TABLE mytable(ID INT NOT NULL, username VARCHAR(16) NOT NULL, INDEX [indexName] (username(length)));

-- 删除索引的语法：

DROP INDEX [indexName] ON mytable;

二、唯一索引

它与前面的普通索引类似，不同的就是：索引列的值必须唯一，但允许有空值。如果是组合索引，则值的组合必须唯一。它有以下几种创建方式：

CREATE UNIQUE INDEX indexName ON mytable(username(length))

-- 修改表结构

ALTER mytable ADD UNIQUE [indexName] ON (username(length))

-- 创建表的时候直接指定

CREATE TABLE mytable(ID INT NOT NULL, username VARCHAR(16) NOT NULL, UNIQUE [ndexName] (username(length)));

三、主键索引

它是一种特殊的唯一索引，不允许有空值。一般是在建表的时候同时创建主键索引：

CREATE TABLE mytable(ID INT NOT NULL, username VARCHAR(16) NOT NULL, PRIMARY KEY(ID));

当然也可以用 ALTER 命令。记住：一个表只能有一个主键。

四、组合索引

为了形象地对比单列索引和组合索引，为表添加多个字段：

CREATE TABLE mytable(ID INT NOT NULL, username VARCHAR(16) NOT NULL, city VARHAR(50) NOT NULL, age INT NOT NULL);

为了进一步榨取MySQL的效率，就要考虑建立组合索引。就是将 name, city, age建到一个索引里：

ALTER TABLE mytable ADD INDEX name_city_age (name(10),city,age);[code]

建表时，usernname长度为 16，这里用 10。这是因为一般情况下名字的长度不会超过10，这样会速索引查询速度，还会减少索引文件的大小，提高INSERT的更新速度。

如果分别在 usernname，city，age上建立单列索引，让该表有3个单列索引，查询时和上述的组合引效率也会大不一样，远远低于我们的组合索引。虽然此时有了三个索引，但MySQL只能用到其中那个它认为似乎是最有效率的单列索引。

建立这样的组合索引，其实是相当于分别建立了下面三组组合索引：

usernname,city,age　usernname,city　usernname 为什么没有 city，age这样的组合索引呢？这因为MySQL组合索引“最左前缀”的结果。简单的理解就是只从最左面的开始组合。并不是只要包这三列的查询都会用到该组合索引，下面的几个SQL就会用到这个组合索引：

[code]

SELECT * FROM mytable WHREE username="admin" AND city="郑州"　SELECT * FROM mytab e WHREE username="admin"

而下面几个则不会用到：

SELECT * FROM mytable WHREE age=20 AND city="郑州"　SELECT * FROM mytable WHREE ci y="郑州"

五、建立索引的时机

到这里我们已经学会了建立索引，那么我们需要在什么情况下建立索引呢？一般来说，在WHERE和JON中出现的列需要建立索引，但也不完全如此，因为MySQL只对<，<=，=，>，>=，BETWEEN，I，以及某些时候的LIKE才会使用索引。例如：

SELECT t.Name　FROM mytable t LEFT JOIN mytable m　ON t.Name=m.username WHERE m. ge=20 AND m.city='郑州'

此时就需要对city和age建立索引，由于mytable表的userame也出现在了JOIN子句中，也有对它建索引的必要。

刚才提到只有某些时候的LIKE才需建立索引。因为在以通配符%和_开头作查询时，MySQL不会使用引。例如下句会使用索引：

SELECT * FROM mytable WHERE username like'admin%'

而下句就不会使用：

SELECT * FROM mytable WHREEt Name like'%admin'

因此，在使用LIKE时应注意以上的区别。

六、索引的不足之处

上面都在说使用索引的好处，但过多的使用索引将会造成滥用。因此索引也会有它的缺点：

1.虽然索引大大提高了查询速度，同时却会降低更新表的速度，如对表进行INSERT、UPDATE和DELE E。因为更新表时，MySQL不仅要保存数据，还要保存一下索引文件。

2.建立索引会占用磁盘空间的索引文件。一般情况这个问题不太严重，但如果你在一个大表上创建了种组合索引，索引文件的会膨胀很快。

索引只是提高效率的一个因素，如果你的MySQL有大数据量的表，就需要花时间研究建立最优秀的引，或优化查询语句。

七、使用索引的注意事项

使用索引时，有以下一些技巧和注意事项：

1.索引不会包含有NULL值的列

只要列中包含有NULL值都将不会被包含在索引中，复合索引中只要有一列含有NULL值，那么这一列于此复合索引就是无效的。所以我们在数据库设计时不要让字段的默认值为NULL。

2.使用短索引

对串列进行索引，如果可能应该指定一个前缀长度。例如，如果有一个CHAR(255)的列，如果在前1个或20个字符内，多数值是惟一的，那么就不要对整个列进行索引。短索引不仅可以提高查询速度而可以节省磁盘空间和I/O操作。

3.索引列排序

MySQL查询只使用一个索引，因此如果where子句中已经使用了索引的话，那么order by中的列是会使用索引的。因此数据库默认排序可以符合要求的情况下不要使用排序操作；尽量不要包含多个列排序，如果需要最好给这些列创建复合索引。

4.like语句操作

一般情况下不鼓励使用like操作，如果非使用不可，如何使用也是一个问题。like “%aaa%” 不会使索引而like “aaa%”可以使用索引。

5.不要在列上进行运算

select * from users where YEAR(adddate)<2007;

将在每个行上进行运算，这将导致索引失效而进行全表扫描，因此我们可以改成:

select * from users where adddate< ‘2007-01-01';

6.不使用NOT IN和<>操作

以上，就对其中MySQL索引类型进行了介绍。希望对大家有所帮助。