

MySQL创建索引的三种方式

1. 新建表中添加索引

#方法一：创建表时

```
CREATE TABLE 表名 (  
    字段名1 数据类型 [完整性约束条件...],  
    字段名2 数据类型 [完整性约束条件...],  
    [UNIQUE | FULLTEXT | SPATIAL ] INDEX | KEY  
    [索引名] (字段名[(长度)] [ASC | DESC])  
);
```

① 普通索引

```
create table t_dept(  
    no int not null primary key,  
    name varchar(20) null,  
    sex varchar(2) null,  
    info varchar(20) null,  
    index index_no(no)  
)
```

② 唯一索引

```
create table t_dept(  
    no int not null primary key,  
    name varchar(20) null,  
    sex varchar(2) null,  
    info varchar(20) null,  
    unique index index_no(no)  
)
```

③ 全文索引

```
create table t_dept(  
    no int not null primary key,  
    name varchar(20) null,  
    sex varchar(2) null,  
    info varchar(20) null,  
    fulltext index index_no(no)  
)
```

④ 多列索引

```
create table t_dept(  
    no int not null primary key,  
    name varchar(20) null,  
    sex varchar(2) null,  
    info varchar(20) null,
```

```
key index_no_name(no,name)
)
```

2. 在已建表中添加索引

#方法二：CREATE在已存在的表上创建索引

```
CREATE [UNIQUE | FULLTEXT | SPATIAL] INDEX 索引名
      ON 表名 (字段名[(长度)] [ASC |DESC]);
```

① 普通索引

```
create index index_name
      on t_dept(name);
```

② 唯一索引

```
create unique index index_name
      on t_dept(name);
```

③ 全文索引

```
create fulltext index index_name
      on t_dept(name);
```

④ 组合索引

```
create index index_name_no
      on t_dept(name,no)
```

3. 以修改表的方式添加索引

方法三：ALTER TABLE在已存在的表上创建索引

```
ALTER TABLE 表名 ADD [UNIQUE | FULLTEXT | SPATIAL] INDEX
      索引名 (字段名[(长度)] [ASC |DESC]);
```

#删除索引：DROP INDEX 索引名 ON 表名字;

① 普通索引

```
alter table t_dept
      add index index_name(name);
```

② 唯一索引

```
alter table t_dept
      add unique index index_name(name);
```

③ 全文索引

```
alter table t_dept
      add fulltext index_name(name);
```

④ 多列索引

```
alter table t_dept  
    add index index_name_no(name,no);
```

例子:

#方式一

```
create table t1(  
    id int,  
    name char,  
    age int,  
    sex enum('male','female'),  
    unique key uni_id(id),  
    index ix_name(name)  
);  
  
create table t1(  
    id int,  
    name char,  
    age int,  
    sex enum('male','female'),  
    unique key uni_id(id),  
    index(name)  
);
```

#方式二

```
create index ix_age on t1(age);
```

#方式三

```
alter table t1 add index ix_sex(sex);  
alter table t1 add index(sex);
```

#查看

```
mysql> show create table t1;  
| t1 | CREATE TABLE `t1` (  
  `id` int(11) DEFAULT NULL,  
  `name` char(1) DEFAULT NULL,  
  `age` int(11) DEFAULT NULL,  
  `sex` enum('male','female') DEFAULT NULL,  
  UNIQUE KEY `uni_id` (`id`),  
  KEY `ix_name` (`name`),  
  KEY `ix_age` (`age`),  
  KEY `ix_sex` (`sex`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

MySQL索引的类型

1、普通索引

是最基本的索引，它没有任何限制。它有以下几种创建方式：

直接创建索引

```
CREATE INDEX index_name ON table(column(length))
```

修改表结构的方式添加索引

```
ALTER TABLE table_name ADD INDEX index_name ON (column(length))
```

创建表的时候同时创建索引

```
CREATE TABLE `table` (  
  `id` int(11) NOT NULL AUTO_INCREMENT ,  
  `title` char(255) CHARACTER NOT NULL ,  
  `content` text CHARACTER NULL ,  
  `time` int(10) NULL DEFAULT NULL ,  
  PRIMARY KEY (`id`),  
  INDEX index_name (title(length))  
)
```

删除索引

```
DROP INDEX index_name ON table
```

2、唯一索引

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，则列值的组合必须唯一。它有以下几种创建方式：

创建唯一索引

```
CREATE UNIQUE INDEX indexName ON table(column(length))
```

修改表结构 UNIQUE (INDEX|KEY)可有可无

```
ALTER TABLE table_name ADD UNIQUE indexName ON (column(length))
```

创建表的时候直接指定

```
CREATE TABLE `table` (  
  `id` int(11) NOT NULL AUTO_INCREMENT ,  
  `title` char(255) CHARACTER NOT NULL ,  
  `content` text CHARACTER NULL ,  
  `time` int(10) NULL DEFAULT NULL ,
```

```
    UNIQUE indexName (title(length))  
);
```

3、主键索引

是一种特殊的唯一索引，一个表只能有一个主键，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 `table` (  
    `id` int(11) NOT NULL AUTO_INCREMENT ,  
    `title` char(255) NOT NULL ,  
    PRIMARY KEY (`id`)  
);
```

4、组合索引

指多个字段上创建的索引，只有在查询条件中使用了创建索引时的第一个字段，索引才会被使用。使用组合索引时遵循最左前缀匹配原则

```
ALTER TABLE `table` ADD INDEX name_city_age (name,city,age);
```

最左前缀匹配原则：顾名思义，就是最左优先，在创建多列索引时，要根据业务需求，where子句中

使用最频繁的一列(where或join中经常使用的列)放在最左边。mysql会一直向右匹配直到遇到范围查询(>、<、between、like)

就停止匹配，比如a=1 and b=2 and c>3 and d=4如果建立(a,b,c,d)顺序的索引，d是用不到索引的，如果建立(a,b,d,c)的索引则都可以用到，a,b,d的顺序可以任意调整。

=和in可以乱序，比如a=1 and b=2 and c=3建立(a,b,c)索引可以任意顺序，mysql的查询优化器会帮你优化成索引可以识别的形式

5、全文索引

主要用来查找文本中的关键字，而不是直接与索引中的值相比较。fulltext索引跟其它索引大不相同，它更像是一个搜索引擎，而不是简单的where语句的参数匹配。fulltext索引配合match against操作使用，而不是一般的where语句加like。它可以在create table，alter table，create index使用，不过目前只有char、varchar，text 列上可以创建全文索引。值得一提的是，在数据量较大时候，现将数据放入一个没有全局索引的表中，然后再用CREATE index创建fulltext索引，要比先为一张表建立fulltext然后再将数据写入的速度快很多。

创建表的适合添加全文索引

```
CREATE TABLE `table` (  
    `id` int(11) NOT NULL AUTO_INCREMENT ,  
    `title` char(255) CHARACTER NOT NULL ,
```

```
`content` text CHARACTER NULL ,  
`time` int(10) NULL DEFAULT NULL ,  
PRIMARY KEY (`id`),  
FULLTEXT KEY (content)  
);
```

修改表结构添加全文索引

```
ALTER TABLE article ADD FULLTEXT index_content(content)
```

直接创建索引

```
CREATE FULLTEXT INDEX index_content ON article(content)
```

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delete。因为更新表时，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多种组合索引，索引文件的增长会很快。索引只是提高效率的一个因素，如果有大数据量的表，就需要花时间研究建立最优秀的索引，或优化查询语句。

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，那么这一列对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10个或20个字符内，多数值是唯一的，

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且可以节省磁盘空间和I/O操作。

3、索引列排序

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中的列是不会使用索引的。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的排序，如果需要最好给这些列创建复合索引。

4、like语句操作

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

5、不要在列上进行运算，这将导致索引失效而进行全表扫描，例如
SELECT * FROM table_name WHERE YEAR(column_name)<2017;

6、不使用not in和<>操作