

目录

- 1. MySQL 键值概述: ..... 1
  - 1.1 索引介绍 ..... 1
  - 1.2 索引的优缺点 ..... 2
  - 1.3 普通索引 index..... 2
    - 1.3.1 创建索引| ..... 2
    - 1.3.2 查看索引| ..... 3
    - 1.3.3 删除索引| ..... 3
- 2. MySQL 主键 primary key: ..... 4
  - 2.1 创建主键 ..... 4
    - 2.1.1 主键的作用：限制字段赋值 ..... 4
    - 2.1.2 主键的使用规则 ..... 4
    - 2.1.3 创建主键 ..... 4
  - 2.2.4 删除主键 ..... 5

## 1. MySQL 键值概述:

根据数据存储要求，选择键值

- ① **index**                  普通索引
- ② **unique**                唯一索引
- ③ **fulltext**               全文索引
- ④ **primary key**          主键
- ⑤ **foreign key**          外键

index、primary key、foreign key    #生产环境一定会用到的键值类型

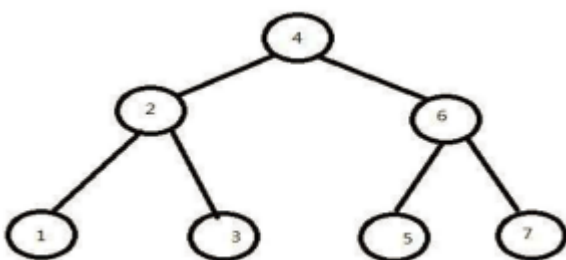
### 1.1 索引介绍

类似于书的目录

对表中字段值进行排序

索引算法: Btree、B+tree、hash

Btree 算法(二叉树):



#1》查找数字 5 时，先用数字 5 和数字 4 对比；

#2》当数字 5 大于数字 4，则直接从数字 4 的右分支进行查找；

#3》接下来用要查找的数字 5 和数字 6 对比；

#4》当数字 5 小于数字 6，则直接从数字 6 的左分支进行查找；

#5》按照以上的方式继续比对查找，直到查找到数据为止；

## 1.2 索引的优缺点

### 索引优点

- ✓ 可以加快数据的查询速度

### 索引缺点

- ✓ 当对表中的数据进行增加、删除和修改的时候，索引也要动态的调整，降低了数据的维护速度
- ✓ 索引需要占物理空间

生产环境下，对数据查的请求远远高于对数据写的请求；

## 1.3 普通索引 index

### 使用规则

#### 具体要求如下：

- ✓ 一个表中可以有多个index字段
- ✓ 字段的值允许重复，且可以赋NULL值
- ✓ index字段 标志是 MUL
- ✓ 通常把做为查询条件的字段设置为index字段

### 1.3.1 创建索引

建表的时候创建索引：index(字段名), index(字段名)....

创建 t1 表时，将 name 字段和 class 字段设置为索引

```
mysql> create table t1(name char(10), class char(9), sex enum("m", "w"), index(name),  
index(class));  
mysql> desc t1;      #约束条件Key变为MUL(索引的标志)
```

在已有的表里创建索引: `create index 索引名 on 表名(字段名);`

```
mysql> create table t2 ( name char(16), pay float(5,2) );
mysql> desc t2;
```

在已有表 t2 表中为字段创建索引 xxx(索引名称可以随便定义)

```
mysql> create index xxx on t2(name);
mysql> desc t2;
```

### 1.3.2 查看索引

语法格式: `show index from 表名 \G;`

```
mysql> show index from t1\G;
***** 1. row *****
      Table: t9                                #表名
    Non_unique: 1
      Key_name: name                            #索引名
    Seq_in_index: 1
    Column_name: name                          #字段名
    ...
```

### 1.3.3 删除索引

语法格式: `drop index 索引名 on 表名;`

删除 t1 表中的索引 name

```
mysql> drop index name on t1;
mysql> desc t1;
mysql> show index from t1\G;
```

## 2. MySQL 主键 primary key:

### 2.1 创建主键

#### 2.1.1 主键的作用：限制字段赋值

#### 2.1.2 主键的使用规则

具体要求如下：

- ✓ 字段值不允许重复，且不允许赋NULL值
- ✓ 一个表中只能有一个primary key字段
- ✓ 多个字段都作为主键，称为复合主键，必须一起创建。
- ✓ 主键字段的标志是PRI
- ✓ 主键通常与 auto\_increment 连用
- ✓ 通常把表中唯一标识记录的字段设置为主键 [记录编号字段]

创建表时，表中存在类似身份证号，编号等时，将表中的该字段设置为主键，让其不能重复，可以自动增长。

#### 2.1.3 创建主键

建表时创建主键，命令：primary key(字段名)

创建 t3 表，字段有：姓名(name)，年龄(age)，将 name 字段设置为主键 primary key

```
mysql> create table t3(name char(10) primary key, age int);
mysql> desc t3;          #查看t3表的表结构，key的值为PRI，则代表该字段为主键
在表t3中插入数据，主键所在的字段，数据不能重复，不允许有空值
mysql> insert into t3 values("bob",29);          #成功
mysql> insert into t3 values("bob",39);          #失败，主键字段的数据重复
mysql> insert into t3 values("jim",19);          #成功
mysql> insert into t3 values(null,29);          #失败，主键所在的字段值不能为NULL值
mysql> insert into t3 values("null",39);          #成功，加引号代表的是字符串
```

```
mysql> insert into t3 values("",59);      #"" 指没有内容，不代表null
mysql> select * from t3;
```

在已有表里创建主键

语法格式: alter table 表名 add primary key(字段名列表);

将表中的字段设置为主键时，则表中该字段的值不能为空，也不能重复，否则添加失败; 表中没有数据时，添加成功

```
mysql> select * from t1;
mysql> desc t1;      #查看t1表的表结构，原先没有主键，允许数据重复
mysql> alter table t1 add primary key(name);    #将t1表中的字段name设置为主键
mysql> desc t1;
```

#### 2.2.4 删除主键

语法格式: alter table 表名 drop primary key;

删除 t1 表的主键

```
mysql> alter table t1 drop primary key;
mysql> desc t1;      #主键消失，但是name字段的约束条件不许为空，可以重复插入数据
mysql> insert into t1 values('bob','NSD2001','m');    #成功
mysql> insert into t1 values('bob','NSD2002','m');    #成功
mysql> insert into t1 values(null,'NSD2002','m');    #失败
```