

从 FTP 下载 : mysql-5.7.17.tar

重新克隆新的虚拟机:

eth0 网卡:192.168.4.50-192.168.4.57

主机名称:mysql50-mysql57

案例一：约束条件

1. 常用约束条件:

条件约束	功能描述
null	允许为空, 默认设置
not null	不允许为空
key	索引类型
default	设置默认值, 缺省为 NULL

```
mysql> desc school.tea;
```

Field	Type	Null	Key	Default	Extra
name	char(5)	YES		NULL	
gender	enum('male','female')	YES		NULL	
interest	set('book','movie','eat')	YES		NULL	

```
mysql> create table school.restrict(  
    name char(5) not null,  
    gender enum("male","female") not null default "male",  
    age int(3) not null default 21,  
    interest set("book","movie","eat")  
);
```

```
mysql> desc school.restrict(
```

Field	Type	Null	Key	Default	Extra
name	char(5)	NO		NULL	
gender	enum('male','female')	YES		male	
age	int(3)	NO		21	
interest	set('book','movie','eat')	YES		NULL	

```
mysql> insert into school.restrict values ("tom","male",18,"book");
```

```
mysql> insert into school.restrict(name) values ("tom"); #仅插入姓名可以成功
```

```
Query OK, 1 row affected (0.06 sec)
```

```
mysql> select * from school.restrict;
```

```
+-----+-----+-----+-----+  
| name | gender | age | interest |  
+-----+-----+-----+-----+  
| tom  | male   | 21 | NULL     |  
+-----+-----+-----+-----+  
1 row in set (0.00 sec)
```

```
mysql> insert into school.restrict(age) values (11); #name 不可以为空
```

```
ERROR 1364 (HY000): Field 'name' doesn't have a default value
```

案例二：修改表结构

基本用法：

alter table 表名 执行动作；

执行动作	功能描述
add	添加字段
modify	修改字段类型
change	修改自定名称
drop	删除字段
rename	修改表名称

1. add 添加字段

alter table 表名 **add** 字段名称 类型（宽度） 约束条件；

```
mysql> alter table school.restrict
add
email varchar(30);
```

Query OK, 0 rows affected (0.33 sec)

Records: 0 Duplicates: 0 Warnings: 0

#默认添加的字段在表的最后。所有已经添加的数据，email 列都是 NULL。

通过 after 字段名可以将新添加的字段放到某个字段后面，或者 first 直接放到第一列。

```
mysql> alter table school.restrict
add
phone varchar(30) not null after name;
```

#添加新字段 phone，放到 name 字段的后面。

```
mysql> alter table school.restrict
add
addr varchar(30) not null default 'beijing' first;
```

#添加新字段 addr，放到所有字段前面。

2. modify 修改字段

alter table 表名 **modify** 字段名称 类型（宽度） 约束条件；

```
mysql> alter table school.restrict
modify
addr varchar(50) default 'shanghai' after name;
```

#修改 addr 字段的类型长度为 50，修改默认值为'shanghai'，将 addr 调整到 name 字段的后面。

```
mysql> desc school.restrict;
```

Field	Type	Null	Key	Default	Extra
name	char(5)	NO		NULL	
addr	varchar(50)	YES		shanghai	
phone	varchar(30)	NO		NULL	
gender	enum('male','female')	NO		male	
age	int(3)	NO		21	
interest	set('book','movie','eat')	YES		NULL	
email	varchar(30)	YES		NULL	

3. chage 修改字段名称

alter table 表名 **change** 字段名称 类型 (宽度) 约束条件;

```
mysql> alter table school.restrict change name myname varchar(10);
```

#把数据表中的 name 字段重命名为 myname, 同时修改了数据类型的长度。

4. drop 修改字段名称

alter table 表名 **drop** 字段名称

```
mysql> alter table school.restrict drop interest;
```

#删除数据表中的 interest 字段。

5. rename 修改字段名称

alter table 表名 **rename** 新的数据表名称

```
mysql> alter table school.restrict rename school.rest;
```

```
mysql> use school;
```

```
mysql> show tables;
```

案例三：MySQL 键值

什么是索引：就是对数据表中的若干字段进行排序的方法，类似于对一本书做目录，有了目录就可以快速定位数据的具体位置。

索引的优点：

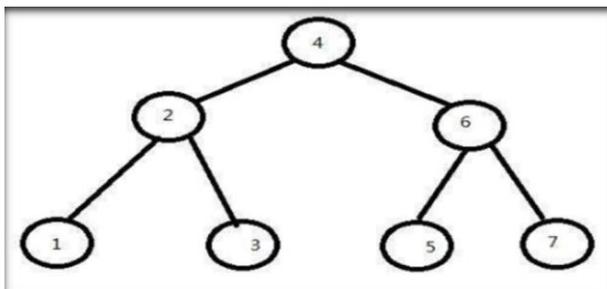
- 通过创建唯一性索引，可以保证数据库表中每一行数据的唯一性
- 可以加快数据的检索速度

索引的缺点：

- 当对表中的数据进行增加、删除和修改的时候，索引也要动态的维护，会降低数据库的写速度
- 索引需要占额外的物理空间

键值的类型

INDEX	普通索引
UNIQUE	唯一索引
FULLTEXT	全文索引
PRIMARY KEY	主键
FOREIGN KEY	外键



1. INDEX 普通索引

使用说明

- 一个表中可以有多个 INDEX 字段
- 字段的值允许有重复，切可以赋 NULL 值
- 经常把做查询条件的字段设置为 INDEX 字段
- INDEX 字段的 KEY 标志是 MUL

```
mysql> create table school.info(  
  id int(6) not null,  
  name varchar(5),  
  sex enum('male','female'),  
  age int(3) default 1,  
  index(id),index(name));
```

Query OK, 0 rows affected (0.13 sec)

```
mysql> desc school.info;
```

Field	Type	Null	Key	Default	Extra
id	int(6)	NO	MUL	NULL	
name	varchar(5)	YES	MUL	NULL	
sex	enum('male','female')	YES		NULL	
age	int(3)	YES		1	

在已有的数据表中创建或删除索引。

1.创建索引： create index 索引名称 on 数据表(字段名称)

```
mysql> create index age on school.info(age);
```

```
Query OK, 0 rows affected (0.18 sec)
```

```
mysql> create index nianling on school.info(age);
```

```
Query OK, 0 rows affected, 1 warning (0.09 sec)
```

提示：可以创建多个索引，索引与字段名称也可以不一样。

2. 删除索引： drop index 索引名称 on 数据表

```
mysql> drop index name on school.info;
```

查看索引信息：

```
mysql> show index from school.info\G;
```

```
***** 1. row ****
```

```
Table: info
Non_unique: 1
Key_name: id
Seq_in_index: 1
Column_name: id
Collation: A
Cardinality: 0
Sub_part: NULL
Packed: NULL
Null:
Index_type: BTREE
Comment:
Index_comment:
Visible: YES
Expression: NULL
```

... (部分内容省略) ...

2. primary key 主键索引

注意事项

- 一个表中只能有一个 primary key 字段
- 对应的字段值不允许有重复，且不允许赋 NULL 值
- 如果有多个字段都作为 PRIMARY KEY，称为复合主键，必须一起创建。
- 主键字段的 KEY 标志是 PRI
- 通常与 AUTO_INCREMENT 连用
- 经常把表中能够唯一标识记录的字段设置为主键字段[记录编号字段]

```
mysql> create table school.student(
```

```
    stu_id char(9),
```

```
    name char(5),
```

```
    primary key(stu_id));
```

```
Query OK, 0 rows affected (0.20 sec)
```

```
mysql> desc school.student;
```

Field	Type	Null	Key	Default	Extra
stu_id	char(9)	NO	PRI	NULL	
name	char(5)	YES		NULL	

提示：stu_id 字段自动被设置为不能为 Null，Key 下面有 Pri 标记（主键标记）。

```
mysql> insert into school.student values (null,'tom');
```

#写入 null 数据会报错。

```
ERROR 1048 (23000): Column 'stu_id' cannot be null
```

```
mysql> insert into school.student values ('NSD181001','卡哇伊');
```

```
Query OK, 1 row affected (0.04 sec)
```

```
mysql> insert into school.student values ('NSD181001','小帕克');
```

```
ERROR 1062 (23000): Duplicate entry 'NSD181001' for key 'PRIMARY'
```

#报错: 主键不能重复!

```
mysql> insert into school.student values ('NSD181002','诸葛波波');
```

```
Query OK, 1 row affected (0.03 sec)
```

创建主键索引的另一种方式:

```
mysql> create table school.student2(  
    id char(9) primary key,  
    name char(5));
```

```
Query OK, 0 rows affected (0.15 sec)
```

```
mysql> desc school.student2;
```

Field	Type	Null	Key	Default	Extra
id	char(9)	NO	PRI	NULL	
name	char(5)	YES		NULL	

对已经存在的数据表创建主键索引

```
mysql> create table school.student3(  
    id char(9),  
    name char(5));
```

```
Query OK, 0 rows affected (0.68 sec)
```

语法: alter table 数据表名称 add primary key (字段名称);

```
mysql> alter table school.student3 add primary key(id);
```

```
Query OK, 0 rows affected (0.34 sec)
```

提示: 在已经存在的表中创建主键索引, 一定要确保做为主键的字段数据中没有 null 的值。

如果确实已经有 null 值了, 可以 delete 删除数据或者 update 更新数据为非 null。

删除数据表中的主键索引

```
mysql> alter table school.student3 drop primary key;
```

```
Query OK, 0 rows affected (0.52 sec)
```

#提示:删除主键后, 对应的字段数据就可以出现重复的数据了。

复合主键索引 (多个字段做主键)

不做主键 (可能同一个人技能不同)

姓名	单位	技能
孙悟空	西游记	100
孙悟空	西游记	80

以姓名为主键 (不同书中的任务技能水平不同, 名称相同时无法写入, 因为姓名不能重复)

姓名	单位	技能
孙悟空	西游记	100
孙悟空	沉香救母	80

创建符合主键索引（姓名和单位，不能重复，但是单独的姓名或单位可以重复）

```
mysql> create table school.book(  
    姓名 char(20),  
    单位 char(20),  
    技能 int(100),  
    primary key(姓名,单位));
```

Query OK, 0 rows affected (0.17 sec)

```
mysql> insert into school.book values ('孙悟空','西游记',100);
```

Query OK, 1 row affected (0.03 sec)

```
mysql> insert into school.book values ('孙悟空','沉香救母',80);
```

Query OK, 1 row affected (0.02 sec)

#提示：单独的姓名重复是可以的

```
mysql> insert into school.book values ('孙悟空','沉香救母',100);
```

ERROR 1062 (23000): Duplicate entry '孙悟空-沉香救母' for key 'PRIMARY'

#提示：姓名和单位都重复会报错。

自动添加属性（可以自动将数据自动加 1）

```
mysql> create table school.demo(  
    id int(100) auto_increment primary key,  
    name char(10));
```

Query OK, 0 rows affected (0.16 sec)

```
mysql> insert into school.demo(name) values ('tom');
```

```
mysql> insert into school.demo(name) values ('jerry');
```

```
mysql> select * from school.demo;
```

```
+----+-----+  
| id | name |  
+----+-----+  
|  1 | tom  |  
|  2 | jerry|  
+----+-----+
```

提示：id 自动不写，默认也会自动加 1。

```
mysql> insert into school.demo values (10,'jerry');
```

```
mysql> insert into school.demo(name) values ('xyz');
```

```
mysql> select * from school.demo;
```

```
+----+-----+  
| id | name |  
+----+-----+  
|  1 | tom  |  
|  2 | jerry|  
| 10 | jerry|  
| 11 | xyz  |  
+----+-----+
```

提示：当同时有自增长和主键时，一定要通过 alter 将自增长删除后才可以删除主键，无法直接删除主键。

3. foreign key 外键

一个作者信息表

一个图书信息表

要求图书表中的图书作者必须是作者表中的作者。

什么是外键？

让当前表字段的值在另一个表中某个字段值的范围内选择。

使用外键的条件：

表存储引擎必须是 innodb（默认就是）

字段类型必须一致

被参照字段必须是索引类型中的一种（primary key）

创建外键的语法：

foreign key(表 A 的字段名称)

references 表 B(字段名称)

on update cascade #同步更新

on delete cascade #同步删除

创建数据库：

```
mysql> create database press character set utf8;
```

创建包含主键的数据表（作者信息表）：

```
mysql> create table press.author(  
    姓名 char(10) primary key,  
    地址 char(10));
```

往作者信息表中插入数据：

```
mysql> insert into press.author values  
    ('施耐庵','苏州'),  
    ('曹雪芹','辽宁'),  
    ('罗贯中','山西');
```

创建图书信息表并创建外键：

```
mysql> create table press.book(  
    书名 char(20),  
    作者 char(10),  
    foreign key(作者) references press.author(姓名)  
    on update cascade on delete cascade);
```

```
mysql> insert into press.book values ('红楼','无');
```

```
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails ('press`.`book`, CONSTRAINT  
`book_ibfk_1` FOREIGN KEY (`作者`) REFERENCES `author` (`姓名`) ON DELETE CASCADE ON UPDATE CASCADE)
```

#当作者不在 author 表中就报错

```
mysql> insert into press.book values ('红楼','曹雪芹');
```

```
Query OK, 1 row affected (0.21 sec)
```

```
mysql> delete from press.author where 姓名='曹雪芹';
```

```
Query OK, 1 row affected (0.03 sec)
```

#提示：当作者表中的作者被删除后，图书信息表中对应的数据也会被删除。

删除外键，需要先查看外键的名称:

```
mysql> show create table press.book\G;
```

```
***** 1. row *****
Table: book
Create Table: CREATE TABLE `book` (
  `书名` char(20) DEFAULT NULL,
  `作者` char(10) DEFAULT NULL,
  KEY `作者` (`作者`),
  CONSTRAINT `book_ibfk_1` FOREIGN KEY (`作者`) REFERENCES `author` (`姓名`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8
1 row in set (0.00 sec)
```

删除外键:

```
mysql> alter table press.book drop foreign key book_ibfk_1;
```

```
mysql> drop table press.book;
```

在现有的数据表中创建外键:

```
mysql> create table press.book(
  书名 char(20),
  作者 char(10));
```

```
mysql> alter table press.book
  add
  foreign key(作者) references press.author(姓名)
  on update cascade on delete cascade;
```