**NSD RDBMS1 DAY02**

1 表结构

1.1 字段约束条件

作用：限制给字段赋值

限制如下：

NULL 字段是否允许赋空null值 not null

key 键值（mysql服务有多种键 功能个不相同）

Default 默认值，不给字段赋值时 使用默认值赋值

Extra 额外设置

练习10分钟到 10：33

create table db1.t9 (

class char(7) not null default "nsd2005" ,

name char(10) not null, sex enum("boy","girl") default "boy");

desc db1.t9;

mysql> insert into db1.t9(name)values("jerry");

mysql> insert into db1.t9 values("nsd2006","tom","girl");

mysql> select \* from db1.t9;

mysql> insert into db1.t9 values(null,"lucy",null);

ERROR 1048 (23000): Column 'class' cannot be null

mysql>

mysql> insert into db1.t9 values("nsd2004","lucy",null);

Query OK, 1 row affected (0.00 sec)

mysql> select \* from db1.t9;

mysql> insert into db1.t9 values(null,null,null);

ERROR 1048 (23000): Column 'class' cannot be null

mysql>

mysql> insert into db1.t9 values("",null,null);

ERROR 1048 (23000): Column 'name' cannot be null

mysql>

mysql> insert into db1.t9 values("","null",null);

Query OK, 1 row affected (0.00 sec)

mysql> select \* from db1.t9;

姓名 xxx

性别 男

年龄 29

职业 it工程师

房子 无

车子

存款0.00 无

1.2 修改表结构

命令格式：

mysql> alter table 库.表 执行动作；

动作包括：

添加新字段 add 字段名 类型 [约束条件] first;

add 字段名 类型 [约束条件] after 列名；

add 字段名 类型 [约束条件]；

mysql> alter table db1.t2 add class char(7) default "nsd2005";

mysql> alter table db1.t2 add stu\_num int first;

mysql> alter table db1.t2 add sex enum("boy","girl") not null default "boy" after name;

mysql> alter table db1.t2 add qq varchar(11) , add tel char(11) ;

mysql> select \* from db1.t2;

删除已有字段 drop 列表名

drop 列表，drop 列名;

mysql> alter table db1.t2 drop tel;

mysql> alter table db1.t2 drop stu\_num , drop qq ;

mysql> select \* from db1.t2;

练习+订餐+休息 到11：15 上课

修改字段类型 modify 字段名 类型 约束条件；

mysql> alter table db1.t2 modify sex enum("man","woman","no") not null default "no";

ERROR 1265 (01000): Data truncated for column 'sex' at row 1

mysql> alter table db1.t2 modify sex enum("boy","woman","no") not null default "no";

mysql> alter table db1.t2 modify class char(7) default "nsd2005" after name;

mysql> select \* from db1.t2;

修改字段名 change 原字段名 新字段名 类型 约束条件；

只修改字段名，字段类型和约束条件原样不变

mysql> alter table db1 .t2 change email mail char(30);

即改名又该字段类型，指定新的字段名和类型

mysql> alter table db1.t2 change name user varchar(15) not null ;

mysql> desc db1.t2;

修改表名 alter table 原表名 rename 新表名；

mysql> use db1;

mysql> alter table t2 rename 学生信息表;

mysql> show tables;

2 MySQL键值(key) 给字段加标志 ，有各自的功能和赋值的限制方式

支持如下键值 ：

普通索引 index

唯一索引 unique

主键 primary key

外键 foreign key

全文索引 fulltext

14:00 上课

2.1 普通索引 index 的使用

索引介绍？

作用： 给字段的值排序

使用规则？ (限制的方式)

创建 给字段加 index

建表时创建索引 index(字段名)；

create table db1.t9(name char(10), class char(7),age int , sex enum("m","w") , index(name),index(class));

在已有表里添加索引

mysql> create index 索引名 on 库名.表名(字段名);

mysql> create index xxx on db1.t4(name);

查看索引

查看标志 desc 表名；

desc db1.t4

查看具体信息 show index from 库.表；

mysql> show index from db1.t4;

删除 删除字段的标志

mysql> drop index 索引名 on 库.表;

mysql> drop index xxx on db1.t4;

2.2 主键primary key 的使用 作用限制字段赋值

使用规则？(限制的方式)

创建主键

建表时创建 priamry key

create table 库.表(字段列表，primary key(字段名) );

create table 库.表(字段名 类型 约束条件 primary key , 字段名 类型 约束条件);

mysql> create table t11 (stu\_num int primary key , name char(10) , class char(7) );

mysql> create table t12 ( stu\_num int , name char(10) , class char(7), primary key(stu\_num) );

在已有表里添加主键

alter table 库.表 add primary key(字段名) ;

mysql> alter table db1.学生信息表 add primary key(class);

删除 删除字段的标志

alter table 库.表 drop primary key;

mysql> alter table db1.学生信息表 drop primary key ;

验证主键的限制

mysql> insert into db1.t11 values (3 , "a" , "05");

mysql> insert into db1.t11 values (3 , "b" , "07");

ERROR 1062 (23000): Duplicate entry '3' for key 'PRIMARY'

mysql>

mysql> insert into db1.t11 values (null, "b" , "07");

ERROR 1048 (23000): Column 'stu\_num' cannot be null

mysql>

复合主键（多列一起做主键）

插入表记录时，符合主键字段的值不同时重复即可

create table db1.t13( ipaddr char(15) , serport smallint , stat enum("allow","deny") , primary key(ipaddr,serport) );

desc db1.t13;

insert into db1.t13 values ("1.1.1.1",22,"allow");

mysql> insert into db1.t13 values ("1.1.1.1",22,"deny");

ERROR 1062 (23000): Duplicate entry '1.1.1.1-22' for key 'PRIMARY'

mysql>

mysql> insert into db1.t13 values ("1.1.1.1",80,"deny");

Query OK, 1 row affected (0.00 sec)

mysql> insert into db1.t13 values ("1.1.1.1",80,"allow");

ERROR 1062 (23000): Duplicate entry '1.1.1.1-80' for key 'PRIMARY'

mysql>

mysql> insert into db1.t13 values ("2.1.1.1",80,"allow");

Query OK, 1 row affected (0.01 sec)

mysql> select \* from db1.t13;

+---------+---------+-------+

| ipaddr | serport | stat |

+---------+---------+-------+

| 1.1.1.1 | 22 | allow |

| 1.1.1.1 | 80 | deny |

| 2.1.1.1 | 80 | allow |

+---------+---------+-------+

3 rows in set (0.00 sec)

mysql>

主键通常与auto\_increment 连用

自增长 （通过自己+1的方式赋值）

create table db1.t14(

id int primary key auto\_increment,

name char(10),

age int,

class char(7),

tel char(11)

);

desc db1.t14;

mysql> insert into db1.t14(name,age,class,tel) values("bob",19,"05","110");

mysql> insert into db1.t14(name,age,class,tel) values("jery",19,"05","110");

mysql> insert into db1.t14(name,age,class,tel) values("lucuy",19,"05","120");

mysql> insert into db1.t14 values(7,"jim",23,"05","110");

mysql> insert into db1.t14(name,age,class,tel) values("alices",21,"05","110");

select \* from db1.t14;

2.3 外键foreign key 的使用 （难点）

外键：给外键字段赋值时，字段的值在另一个表字段值的范围内选择

使用规则？

创建外键的命令格式：

create table 库.表（字段名列表 ，foreign key(字段名) references 表名（字段名） on update cascade on delete cascade ）engine=innodb;

下次课讲

创建 外键

验证 外键

删除 外键

创建 外键

首先创建被参考的表 yg\_tab

create table db1.yg\_tab( yg\_id int primary key auto\_increment , name char(10) )engine=innodb;

insert into db1.yg\_tab(name)values("bob");

insert into db1.yg\_tab(name)values("tom");

insert into db1.yg\_tab(name)values("jim");

select \* from db1.yg\_tab;

创建有外键的工资表gz\_tab

create table db1.gz\_tab( gz\_id int , gz float(7,2) , foreign key(gz\_id) references yg\_tab(yg\_id) on update cascade on delete cascade )engine=innodb;

mysql> show create table db1.gz\_tab \G #查看建表的命令

验证 外键

赋值约束

mysql> insert into gz\_tab values (4,50000);

ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`db1`.`gz\_tab`, CONSTRAINT `gz\_tab\_ibfk\_1` FOREIGN KEY (`gz\_id`) REFERENCES `yg\_tab` (`yg\_id`) ON DELETE CASCADE ON UPDATE CASCADE)

mysql>

mysql> insert into gz\_tab values (1,50000);

Query OK, 1 row affected (0.01 sec)

mysql> insert into gz\_tab values (2,40000);

Query OK, 1 row affected (0.00 sec)

mysql> insert into gz\_tab values (3,20000);

Query OK, 1 row affected (0.00 sec)

mysql> select \* from gz\_tab;

同步更新

mysql> update yg\_tab set yg\_id=8 where yg\_id=2 ;

mysql> select \* from gz\_tab;

同步删除

mysql> delete from yg\_tab where yg\_id=3;

mysql> select \* from gz\_tab;

弥补外键的不足

弥补外键的不足

弥补外键的不足（考考各位对外键的理解）

mysql> delete from gz\_tab;

mysql> alter table gz\_tab add primary key(gz\_id);

MySQL> desc gz\_tab;

mysql> select \* from gz\_tab;

mysql> insert into gz\_tab values(1, 50000);

mysql> insert into gz\_tab values(8, 50000);

mysql> insert into gz\_tab values(8, 50000);

mysql> insert into gz\_tab values(5, 50000);

mysql> insert into gz\_tab values(1, 50000);

mysql> insert into gz\_tab values(null, 60000);

mysql> select \* from gz\_tab; 只能给已有的员工发1次工资

删除 外键

mysql> alter table db1.gz\_tab drop foreign key gz\_tab\_ibfk\_1 ;

在已有表里添加外键

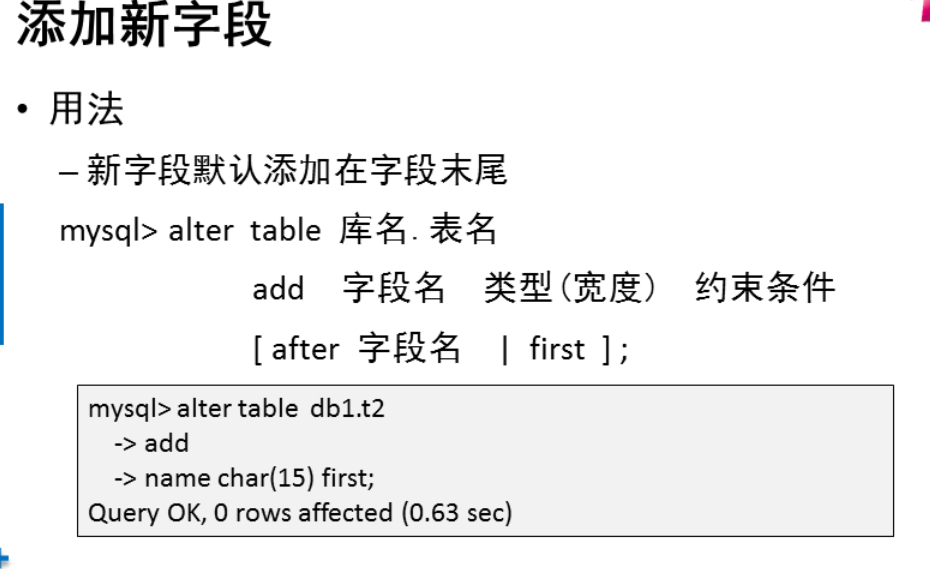
mysql> alter table db1.gz\_tab add foreign key(gz\_id) references yg\_tab(yg\_id) on update cascade on delete cascade ;

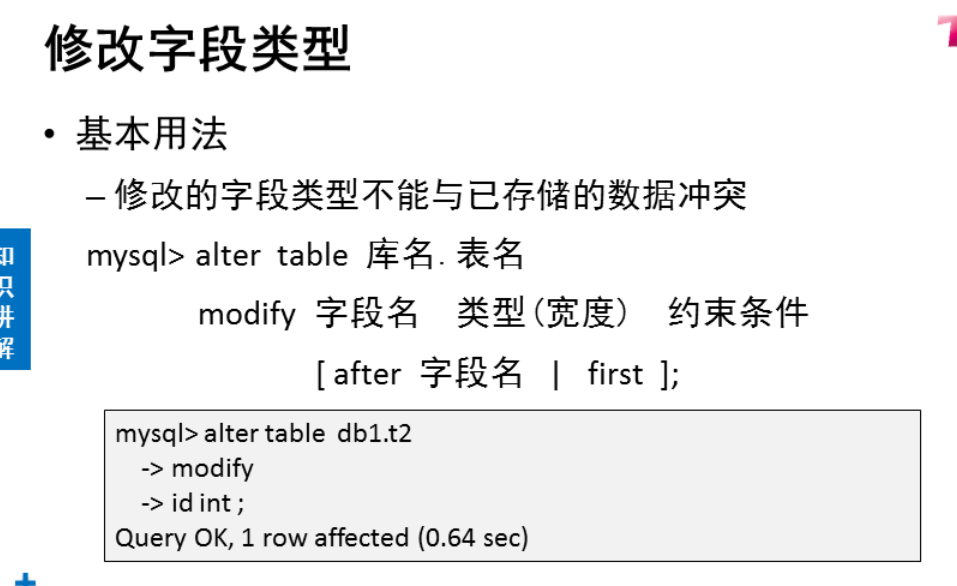
1. [案例1：约束条件](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/RDBMS1/DAY02/CASE/01/index.html#case1)
2. [案例2：修改表结构](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/RDBMS1/DAY02/CASE/01/index.html#case2)
3. [案例3：index 普通索引](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/RDBMS1/DAY02/CASE/01/index.html#case3)
4. [案例4：primary key 主键](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/RDBMS1/DAY02/CASE/01/index.html#case4)
5. [案例5：foreign key 外键](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/RDBMS1/DAY02/CASE/01/index.html#case5)

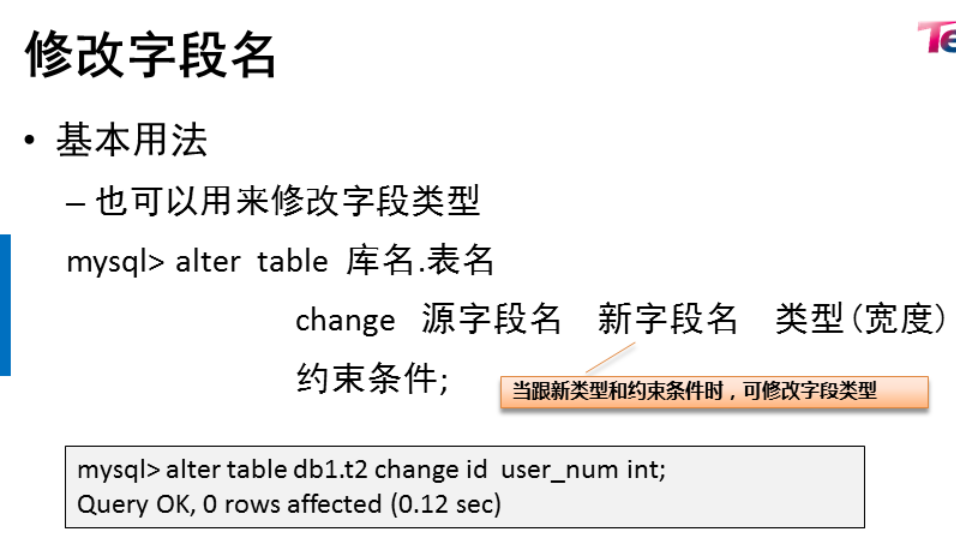


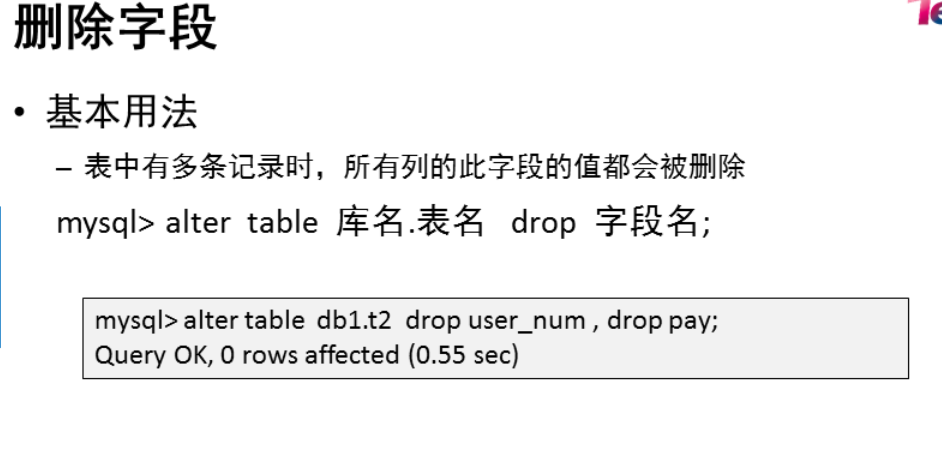


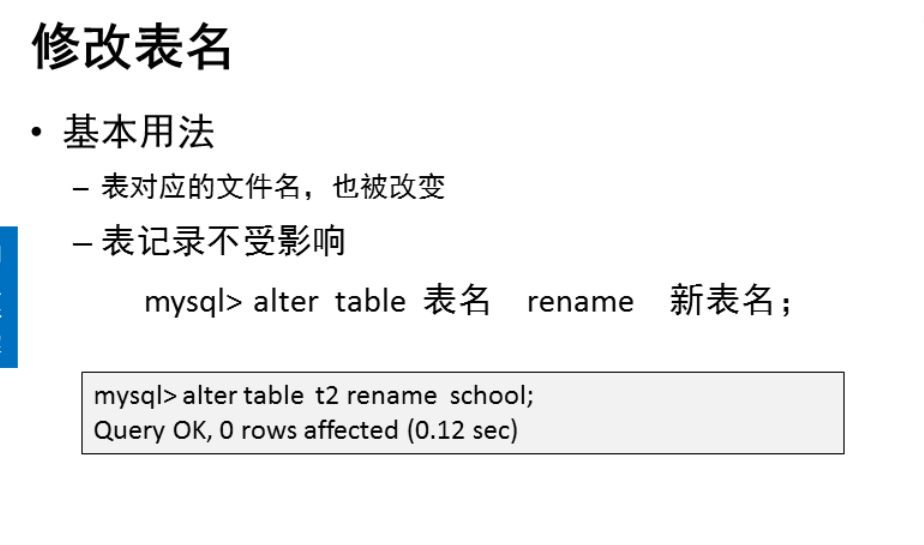




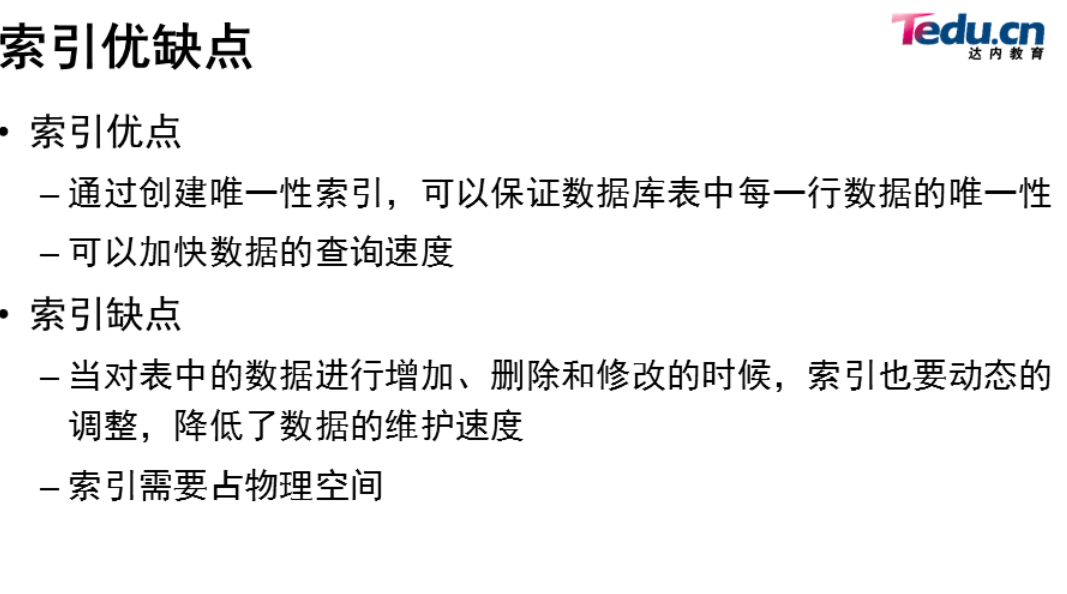


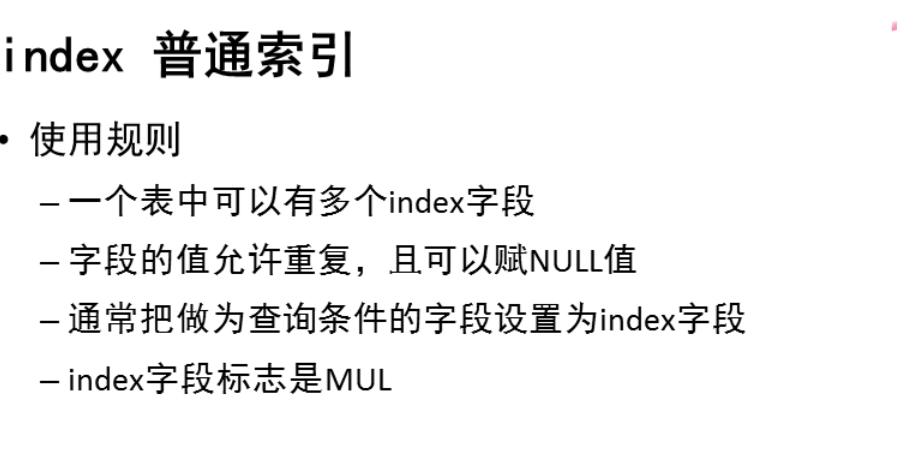


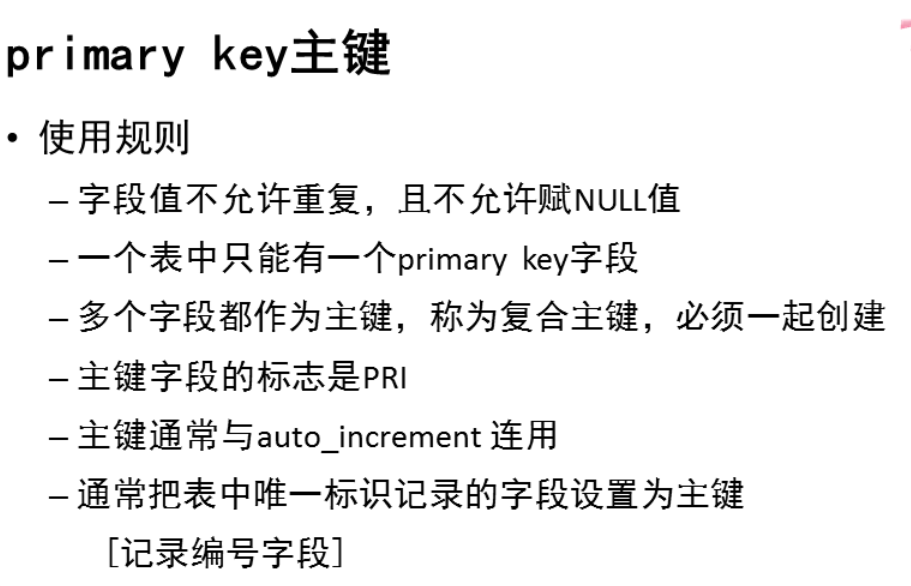


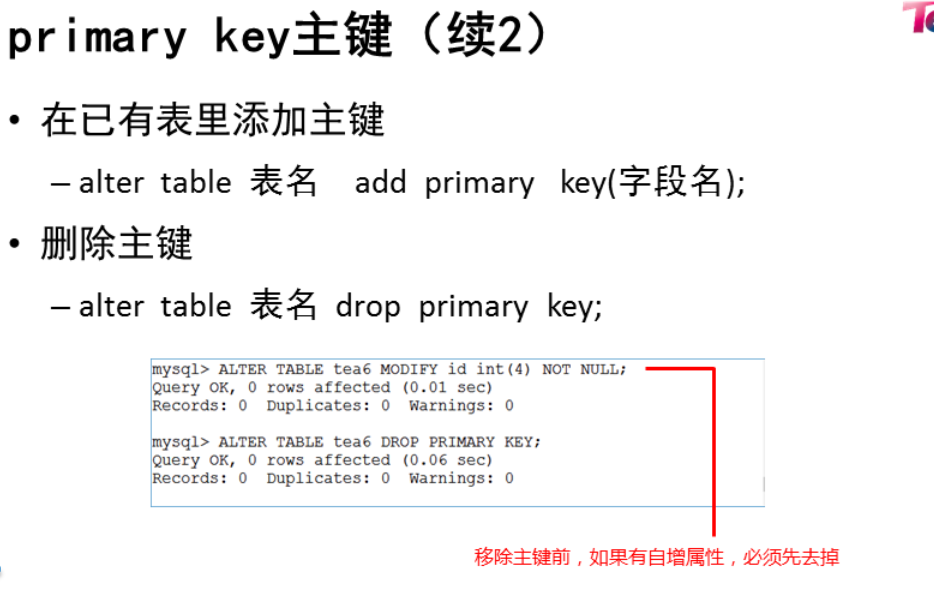


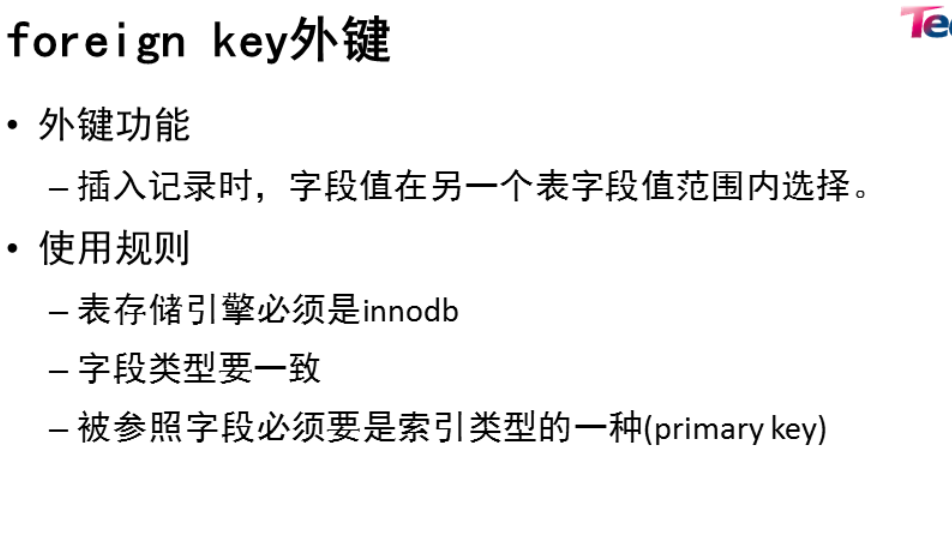












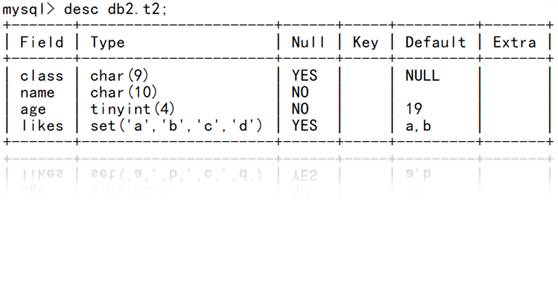


**1 案例1：约束条件**

**1.1 问题**

具体要求如下：

* 如图-1所示设置约束条件



图－1

**1.2 步骤**

实现此案例需要按照如下步骤进行。

步骤一：设置约束

1）在db2库里创建t2表时设置字段约束条件

1. mysql>
2. mysql> create database db2; //建库
3. Query OK, 1 row affected (0.00 sec)
4. mysql> use db2; //切换库
5. Database changed
6. mysql> create table t2 ( //建表
7. -> class char(9),
8. -> name char(10) not null ,
9. -> age tinyint not null default 19 ,
10. -> likes set("a","b","c","d") default "a,b"
11. -> );
12. Query OK, 0 rows affected (0.05 sec)
13. mysql> desc t2; //查看表结构
14. +-------+----------------------+------+-----+---------+-------+
15. | Field | Type | Null | Key | Default | Extra |
16. +-------+----------------------+------+-----+---------+-------+
17. | class | char(9) | YES | | NULL | |
18. | name | char(10) | NO | | NULL | |
19. | age | tinyint(4) | NO | | 19 | |
20. | likes | set('a','b','c','d') | YES | | a,b | |
21. +-------+----------------------+------+-----+---------+-------+
22. 4 rows in set (0.00 sec)
23. mysql>
24. mysql> insert into t2 values (null,"bob",29,"c,d");
25. Query OK, 1 row affected (0.05 sec)
26. mysql> insert into t2(class,name) values ("nsd1902","tom");
27. Query OK, 1 row affected (0.05 sec)
28. mysql> insert into t2 values (null,null,null,null);
29. ERROR 1048 (23000): Column 'name' cannot be null //不允许赋null值
30. MariaDB [db2]>
31. MariaDB [db2]> select \* from db2.t1; //查看记录
32. +---------+------+-----+-------+
33. | class | name | age | likes |
34. +---------+------+-----+-------+
35. | NULL | bob | 29 | c,d |
36. | nsd1902 | tom | 19 | a,b |
37. +---------+------+-----+-------+
38. 2 rows in set (0.00 sec)
39. mysql>

**2 案例2：修改表结构**

**2.1 问题**

具体要求如下：

* 添加字段
* 修改字段名
* 修改字段类型
* 删除字段
* 修改表名

**2.2 步骤**

实现此案例需要按照如下步骤进行。

步骤一：添加字段

1）在studb库下创建tea6表

1. Mysql> create database studb;
2. mysql> CREATE TABLE studb.tea6(
3. -> id int(4) ,
4. -> name varchar(4) NOT NULL,
5. -> age int(2) NOT NULL
6. -> );
7. Query OK, 0 rows affected (0.34 sec)
8. mysql>

2）为tea6表添加一个address字段

1. mysql> DESC tea6;
2. +-------+------------+------+-----+---------+-------+
3. | Field | Type | Null | Key | Default | Extra |
4. +-------+------------+------+-----+---------+-------+
5. | id | int(4) | YES | | NULL | |
6. | name | varchar(4) | NO | | NULL | |
7. | age | int(2) | NO | | NULL | |
8. +-------+------------+------+-----+---------+-------+
9. 3 rows in set (0.00 sec)

添加address字段：

1. mysql> ALTER TABLE tea6 ADD address varchar(48);
2. Query OK, 0 rows affected (0.84 sec)
3. Records: 0 Duplicates: 0 Warnings: 0

添加后（默认作为最后一个字段）：

1. mysql> DESC tea6;
2. +---------+-------------+------+-----+---------+-------+
3. | Field | Type | Null | Key | Default | Extra |
4. +---------+-------------+------+-----+---------+-------+
5. | id | int(4) | YES | | NULL | |
6. | name | varchar(4) | NO | | NULL | |
7. | age | int(2) | NO | | NULL | |
8. | address | varchar(48) | YES | | NULL | |
9. +---------+-------------+------+-----+---------+-------+
10. 4 rows in set (0.00 sec)

3）在tea6表的age列之后添加一个gender字段

添加操作：

1. mysql> ALTER TABLE tea6 ADD gender enum('boy','girl') AFTER age;
2. Query OK, 0 rows affected (0.59 sec)
3. Records: 0 Duplicates: 0 Warnings: 0

确认添加结果：

1. mysql> DESC tea6;
2. +---------+--------------------+------+-----+---------+-------+
3. | Field | Type | Null | Key | Default | Extra |
4. +---------+--------------------+------+-----+---------+-------+
5. | id | int(4) |YES | | NULL | |
6. | name | varchar(4) | NO | | NULL | |
7. | age | int(2) | NO | | NULL | |
8. | gender | enum('boy','girl') | YES | | NULL | |
9. | address | varchar(48) | YES | | NULL | |
10. +---------+--------------------+------+-----+---------+-------+
11. 5 rows in set (0.00 sec)

步骤二：修改字段名和字段类型

将tea6表的gender字段改名为sex，并添加非空约束

修改操作：

1. mysql> ALTER TABLE tea6 CHANGE gender
2. -> sex enum('boy','girl') NOT NULL;
3. Query OK, 0 rows affected (0.08 sec)
4. Records: 0 Duplicates: 0 Warnings: 0

确认修改结果：

1. mysql> DESC tea6;
2. +---------+--------------------+------+-----+---------+-------+
3. | Field | Type | Null | Key | Default | Extra |
4. +---------+--------------------+------+-----+---------+-------+
5. | id | int(4) | YES | | NULL | |
6. | name | varchar(4) | NO | | NULL | |
7. | age | int(2) | NO | | NULL | |
8. | sex | enum('boy','girl') | NO | | NULL | |
9. | address | varchar(48) | YES | | NULL | |
10. +---------+--------------------+------+-----+---------+-------+
11. 5 rows in set (0.00 sec)

步骤三：删除字段

删除tea6表中名为sex的字段：

1. mysql> ALTER TABLE tea6 DROP sex; //删除操作
2. Query OK, 0 rows affected (0.52 sec)
3. Records: 0 Duplicates: 0 Warnings: 0
4. mysql> DESC tea6;                                         //确认删除结果
5. +---------+-------------+------+-----+---------+-------+
6. | Field | Type | Null | Key | Default | Extra |
7. +---------+-------------+------+-----+---------+-------+
8. | id | int(4) | YES | | NULL | |
9. | name | varchar(4) | NO | | NULL | |
10. | age | int(2) | NO | | NULL | |
11. | address | varchar(48) | YES | | NULL | |
12. +---------+-------------+------+-----+---------+-------+
13. 4 rows in set (0.00 sec)

**3 案例3：index 普通索引**

**3.1 问题**

具体要求如下：

* 在已有表里添加index字段
* 建表时，添加index字段
* 查看表索引
* 删除表索引

**3.2 步骤**

实现此案例需要按照如下步骤进行。

步骤一：索引的创建与删除

1）创建表的时候指定INDEX索引字段

创建库home：

1. mysql> create database home;
2. Query OK, 1 row affected (0.00 sec)

允许有多个INDEX索引字段。比如，以下操作在home库中创建了tea4表，将其中的id、name作为索引字段：

1. mysql> USE home;
2. Database changed
3. mysql> CREATE TABLE tea4(
4. -> id char(6) NOT NULL,
5. -> name varchar(6) NOT NULL,
6. -> age int(3) NOT NULL,
7. -> gender ENUM('boy','girl') DEFAULT 'boy',
8. -> INDEX(id),INDEX(name)
9. -> );
10. Query OK, 0 rows affected (0.59 sec)

查看新建tea4表的字段结构，可以发现两个非空索引字段的KEY标志为MUL：

1. mysql> DESC tea4;
2. +--------+--------------------+------+-----+---------+-------+
3. | Field | Type | Null | Key | Default | Extra |
4. +--------+--------------------+------+-----+---------+-------+
5. | id | char(6) | NO | MUL | NULL | |
6. | name | varchar(6) | NO | MUL | NULL | |
7. | age | int(3) | NO | | NULL | |
8. | gender | enum('boy','girl') | YES | | boy | |
9. +--------+--------------------+------+-----+---------+-------+
10. 4 rows in set (0.00 sec)

2）删除现有表的某个INDEX索引字段

比如，删除tea4表中名称为named的INDEX索引字段：

1. mysql> drop INDEX name ON tea4; //删除name字段的索引
2. Query OK, 0 rows affected (0.18 sec)
3. Records: 0 Duplicates: 0 Warnings: 0
4. mysql> DESC tea4; //确认删除结果
5. +--------+--------------------+------+-----+---------+-------+
6. | Field | Type | Null | Key | Default | Extra |
7. +--------+--------------------+------+-----+---------+-------+
8. | id | char(6) | NO | MUL | NULL | |
9. | name | varchar(6) | NO | | NULL | |
10. | age | int(3) | NO | | NULL | |
11. | gender | enum('boy','girl') | YES | | boy | |
12. +--------+--------------------+------+-----+---------+-------+
13. 4 rows in set (0.00 sec)

3）在已有的某个表中设置INDEX索引字段

比如，针对tea4表的age字段建立索引，名称为 nianling：

1. mysql> CREATE INDEX nianling ON tea4(age);     //针对指定字段创建索引
2. Query OK, 0 rows affected (0.62 sec)
3. Records: 0 Duplicates: 0 Warnings: 0
4. mysql> DESC tea4;                                 //确认创建结果
5. +--------+--------------------+------+-----+---------+-------+
6. | Field | Type | Null | Key | Default | Extra |
7. +--------+--------------------+------+-----+---------+-------+
8. | id | char(6) | NO | MUL | NULL | |
9. | name | varchar(6) | NO | | NULL | |
10. | age | int(3) | NO | MUL | NULL | |
11. | gender | enum('boy','girl') | YES | | boy | |
12. +--------+--------------------+------+-----+---------+-------+
13. 4 rows in set (0.00 sec)

4）查看指定表的索引信息

使用SHOW INDEX 指令：

1. mysql> SHOW INDEX FROM tea4\G
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
3. Table: tea4
4. Non\_unique: 1
5. Key\_name: id
6. Seq\_in\_index: 1
7. Column\_name: id
8. Collation: A
9. Cardinality: 0
10. Sub\_part: NULL
11. Packed: NULL
12. Null:
13. Index\_type: BTREE //使用B树算法
14. Comment:
15. Index\_comment:
16. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 2. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
17. Table: tea4
18. Non\_unique: 1
19. Key\_name: nianling //索引名称
20. Seq\_in\_index: 1
21. Column\_name: age //字段名称
22. Collation: A
23. Cardinality: 0
24. Sub\_part: NULL
25. Packed: NULL
26. Null:
27. Index\_type: BTREE
28. Comment:
29. Index\_comment:
30. 2 rows in set (0.00 sec)
31. Query OK, 0 rows affected (0.30 sec)
32. Mysql>
33. Query OK, 0 rows affected (0.47 sec)
34. Records: 0 Duplicates: 0 Warnings: 0

**4 案例4：primary key 主键**

**4.1 问题**

具体要求如下：

* 建表时，创建主键
* 在已有表里添加主键
* 建表时创建复合主键
* 删除主键
* 设置字段值自增长

**4.2 步骤**

实现此案例需要按照如下步骤进行。

步骤一：练习主键的使用

1）建表时设置PRIMARY KEY主键索引

1. Mysql> create database db2;
2. mysql> CREATE TABLE db2.biao01(
3. -> id int(4) PRIMARY KEY,                     //直接在字段定义时约束
4. -> name varchar(8)
5. -> );
6. Query OK, 0 rows affected (0.19 sec)

或者：

1. mysql> CREATE TABLE db2.biao02(
2. -> id int(4),
3. -> name varchar(8),
4. -> PRIMARY KEY(id)                             //所有字段定义完，最后指定
5. -> );
6. Query OK, 0 rows affected (0.17 sec)

在建表的时候，如果主键字段为int类型，还可以为其设置AUTO\_INCREMENT自增属性，这样当添加新的表记录时，此字段的值会自动从1开始逐个增加，无需手动指定。比如，新建一个tea6表，将id列作为自增的主键字段：

1. mysql> CREATE TABLE db2.tea6(
2. -> id int(4) AUTO\_INCREMENT,
3. -> name varchar(4) NOT NULL,
4. -> age int(2) NOT NULL,
5. -> PRIMARY KEY(id)
6. -> );
7. Query OK, 0 rows affected (0.29 sec)

2）删除现有表的PRIMARY KEY主键索引

如果要移除某个表的PRIMARY KEY约束，需要通过ALTER TABLE指令修改。比如，以下操作将清除biao01表的主键索引。

清除前（主键为id）：

1. mysql> DESC db2.biao01;
2. +-------+------------+------+-----+---------+-------+
3. | Field | Type | Null | Key | Default | Extra |
4. +-------+------------+------+-----+---------+-------+
5. | id | int(4) | NO | PRI | NULL | |
6. | name | varchar(8) | YES | | NULL | |
7. +-------+------------+------+-----+---------+-------+
8. 2 rows in set (0.00 sec)

清除操作：

1. mysql> ALTER TABLE db2.biao01 DROP PRIMARY KEY;
2. Query OK, 0 rows affected (0.49 sec)
3. Records: 0 Duplicates: 0 Warnings: 0

清除后（无主键）：

1. mysql> DESC db2.biao01;
2. +-------+------------+------+-----+---------+-------+
3. | Field | Type | Null | Key | Default | Extra |
4. +-------+------------+------+-----+---------+-------+
5. | id | int(4) | NO | | NULL | |
6. | name | varchar(8) | YES | | NULL | |
7. +-------+------------+------+-----+---------+-------+
8. 2 rows in set (0.00 sec)

当尝试删除tea6表的主键时，会出现异常：

1. mysql> ALTER TABLE tea6 DROP PRIMARY KEY;
2. ERROR 1075 (42000): Incorrect table definition; there can be only one auto column and it must be defined as a key

这是因为tea6表的主键字段id具有AUTO\_INCREMNET自增属性，提示这种字段必须作为主键存在，因此若要清除此主键必须先清除自增属性——修改id列的字段定义：

1. mysql> ALTER TABLE tea6 MODIFY id int(4) NOT NULL;
2. Query OK, 0 rows affected (0.75 sec)
3. Records: 0 Duplicates: 0 Warnings: 0

然后再清除主键属性就OK了：

1. mysql> ALTER TABLE tea6 DROP PRIMARY KEY;                 //清除主键
2. Query OK, 0 rows affected (0.39 sec)
3. Records: 0 Duplicates: 0 Warnings: 0
4. mysql> desc tea6; //确认清除结果
5. +-------+------------+------+-----+---------+-------+
6. | Field | Type | Null | Key | Default | Extra |
7. +-------+------------+------+-----+---------+-------+
8. | id | int(4) | NO | | NULL | |
9. | name | varchar(4) | NO | | NULL | |
10. | age | int(2) | NO | | NULL | |
11. +-------+------------+------+-----+---------+-------+
12. 3 rows in set (0.01 sec)

3）为现有表添加PRIMARY KEY主键索引

重新为tea6表指定主键字段，仍然使用id列：

1. mysql> ALTER TABLE tea6 ADD PRIMARY KEY(id);             //设置主键字段
2. Query OK, 0 rows affected (0.35 sec)
3. Records: 0 Duplicates: 0 Warnings: 0
4. mysql> DESC tea6;                                         //确认设置结果
5. +-------+------------+------+-----+---------+-------+
6. | Field | Type | Null | Key | Default | Extra |
7. +-------+------------+------+-----+---------+-------+
8. | id | int(4) | NO | PRI | NULL | |
9. | name | varchar(4) | NO | | NULL | |
10. | age | int(2) | NO | | NULL | |
11. +-------+------------+------+-----+---------+-------+
12. 3 rows in set (0.00 sec)

4）建表时创建复合主键

1. mysql>
2. mysql> create table db2.t6(
3. -> class char(7),
4. -> name char(15),
5. -> pay enum("yes","no") default "no",
6. -> primary key(class,name,pay) //指定多个字段一起做主键
7. -> );
8. Query OK, 0 rows affected (0.04 sec)
9. mysql> desc db2.t6;
10. +-------+------------------+------+-----+---------+-------+
11. | Field | Type | Null | Key | Default | Extra |
12. +-------+------------------+------+-----+---------+-------+
13. | class | char(7) | NO | PRI | | |
14. | name | char(15) | NO | PRI | | |
15. | pay | enum('yes','no') | NO | PRI | no | |
16. +-------+------------------+------+-----+---------+-------+
17. 3 rows in set (0.01 sec)
18. mysql>

**5 案例5：foreign key 外键**

**5.1 问题**

具体要求如下：

* 创建员工表yg 如表-1所示
* 创建工资表gz如表-2所示 ,并设置外键实现同步更新与同步删除
* 测试外键
* 删除外键

表-1 员工表yg的数据

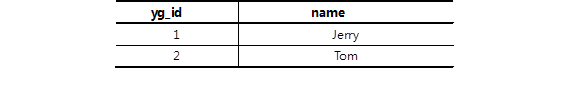
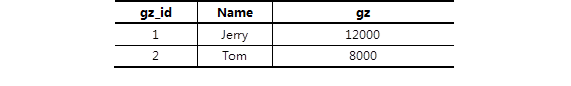


表-2 工资表gz的数据



步骤一：创建外键

1）创建yg表，用来记录员工工号、姓名，其中yg\_id列作为主键，并设置自增属性

1. mysql> CREATE TABLE yg(
2. -> yg\_id int primary key AUTO\_INCREMENT,
3. -> name char(16)
4. -> )engine=innodb;
5. Query OK, 0 rows affected (0.15 sec)
6. Mysql>

2）创建gz表，用来记录员工的工资信息

其中gz\_id需要参考员工工号，即gz表的gz\_id字段设为外键，将yg表的yg\_id字段作为参考键：

1. mysql> CREATE TABLE gz(
2. -> gz\_id int,
3. -> name char(16) ,
4. -> gz float(7,2) ,
5. -> FOREIGN KEY(gz\_id) REFERENCES yg(yg\_id) //创建外键
6. -> ON UPDATE CASCADE ON DELETE CASCADE //同步更新、同步删除
7. -> )engine=innodb;
8. Query OK, 0 rows affected (0.23 sec)
9. Mysql>

3）为yg表添加2条员工信息记录

因yg\_id有AUTO\_INCREMENT属性，会自动填充，所以只要为name列赋值就可以了。

插入表记录可使用INSERT指令，这里先执行下列操作，具体在下一章学习：

1. mysql> INSERT INTO yg(name) VALUES('Jerry'),('Tom');
2. Query OK, 2 rows affected (0.16 sec)
3. Records: 2 Duplicates: 0 Warnings: 0

确认yg表的数据记录：

1. mysql> SELECT \* FROM yg;
2. +-------+-------+
3. | yg\_id | name |
4. +-------+-------+
5. | 1 | Jerry |
6. | 2 | Tom |
7. +-------+-------+
8. 2 rows in set (0.00 sec)

4）为gz表添加2条工资信息记录

同上，数据参考图-2，插入相应的工资记录（gz\_id字段未指定默认值，也未设置自增属性，所以需要手动赋值）：

1. mysql> INSERT INTO gz(gz\_id,name,gz)
2. -> VALUES(1,'Jerry',12000),(2,'Tom',8000)
3. -> ;
4. Query OK, 2 rows affected (0.06 sec)
5. Records: 2 Duplicates: 0 Warnings: 0

确认gz表的数据记录：

1. mysql> SELECT \* FROM gz;
2. +-------+-------+----------+
3. | gz\_id | name | gz |
4. +-------+-------+----------+
5. | 1 | Jerry | 12000.00 |
6. | 2 | Tom | 8000.00 |
7. +-------+-------+----------+
8. 2 rows in set (0.05 sec)

5）验证表记录的UPDATE更新联动

将yg表中Jerry用户的yg\_id修改为1234：

1. mysql> update yg SET yg\_id=1234 WHERE name='Jerry';
2. Query OK, 1 row affected (0.05 sec)
3. Rows matched: 1 Changed: 1 Warnings: 0

确认修改结果：

1. mysql> SELECT \* FROM yg;
2. +-------+-------+
3. | yg\_id | name |
4. +-------+-------+
5. | 2 | Tom |
6. | 1234 | Jerry |
7. +-------+-------+
8. 2 rows in set (0.00 sec)

同时也会发现，gz表中Jerry用户的gz\_id也跟着变了：

1. mysql> SELECT \* FROM gz;
2. +-------+-------+----------+
3. | gz\_id | name | gz |
4. +-------+-------+----------+
5. | 1234 | Jerry | 12000.00 |
6. | 2 | Tom | 8000.00 |
7. +-------+-------+----------+
8. 2 rows in set (0.00 sec)

6）验证表记录的DELETE删除联动

删除yg表中用户Jerry的记录：

1. mysql> DELETE FROM yg WHERE name='Jerry';
2. Query OK, 1 row affected (0.05 sec)

确认删除结果：

1. mysql> SELECT \* FROM yg;
2. +-------+------+
3. | yg\_id | name |
4. +-------+------+
5. | 2 | Tom |
6. +-------+------+
7. 1 row in set (0.00 sec)

查看gz表中的变化（Jerry的记录也没了）：

1. mysql> SELECT \* FROM gz;
2. +-------+------+---------+
3. | gz\_id | name | gz |
4. +-------+------+---------+
5. | 2 | Tom | 8000.00 |
6. +-------+------+---------+
7. 1 row in set (0.00 sec)

7）删除指定表的外键约束

先通过SHOW指令获取表格的外键约束名称：

1. mysql> SHOW CREATE TABLE gz\G
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
3. Table: gz
4. Create Table: CREATE TABLE `gz` (
5. `gz\_id` int(4) NOT NULL,
6. `name` char(16) NOT NULL,
7. `gz` float(7,2) NOT NULL DEFAULT '0.00',
8. KEY `name` (`name`),
9. KEY `gz\_id` (`gz\_id`),
10. CONSTRAINT `gz\_ibfk\_1` FOREIGN KEY (`gz\_id`) REFERENCES `yg` (`yg\_id`) ON DELETE CASCADE ON UPDATE CASCADE
11. ) ENGINE=InnoDB DEFAULT CHARSET=utf8
12. 1 row in set (0.00 sec)

其中gz\_ibfk\_1即删除外键约束时要用到的名称。

删除操作：

1. mysql> ALTER TABLE gz DROP FOREIGN KEY gz\_ibfk\_1;
2. Query OK, 0 rows affected (0.01 sec)
3. Records: 0 Duplicates: 0 Warnings: 0

确认删除结果：

1. mysql> SHOW CREATE TABLE gz\G
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
3. Table: gz
4. Create Table: CREATE TABLE `gz` (
5. `gz\_id` int(4) NOT NULL,
6. `name` char(16) NOT NULL,
7. `gz` float(7,2) NOT NULL DEFAULT '0.00',
8. KEY `name` (`name`),
9. KEY `gz\_id` (`gz\_id`)
10. ) ENGINE=InnoDB DEFAULT CHARSET=utf8
11. 1 row in set (0.00 sec)