TTS 11.0 COOKBOOK

(NSD RDBMS1 DAY02)

版本编号 11.0

2019-06

达内 IT 培训集团

NSD RDBMS1 DAY02

1. 案例 1：约束条件

- **问题**

  - 具体要求如下：
  - 如图-1 所示设置约束条件

```
mysql> desc db2.t2;
+--------+----------------------+------+-----+---------+-------+
| Field  | Type                 | Null | Key | Default | Extra |
+--------+----------------------+------+-----+---------+-------+
| class  | char(9)              | YES  |     | NULL    |       |
| name   | char(10)             | NO   |     |         |       |
| age    | tinyint(4)           | NO   |     | 19      |       |
| likes  | set('a','b','c','d') | YES  |     | a,b     |       |
+--------+----------------------+------+-----+---------+-------+
```

图-1

- **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

```
mysql>

mysql> create database db2; //建库
Query OK, 1 row affected (0.00 sec)

mysql> use  db2; //切换库
Database changed
mysql> create table  t2 (  //建表
    -> class   char(9),
    -> name    char(10) not null  ,
    -> age     tinyint  not null default  19 ,
    -> likes   set("a","b","c","d")  default  "a,b"
    -> );
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> desc t2; //查看表结构
+-------+----------------------+------+-----+---------+-------+
| Field | Type                 | Null | Key | Default | Extra |
+-------+----------------------+------+-----+---------+-------+
| class | char(9)              | YES  |     | NULL    |       |
| name  | char(10)             | NO   |     | NULL    |       |
| age   | tinyint(4)           | NO   |     | 19      |       |
| likes | set('a','b','c','d') | YES  |     | a,b     |       |
+-------+----------------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql>

mysql> insert into  t2  values (null,"bob",29,"c,d");
Query OK, 1 row affected (0.05 sec)

mysql> insert into  t2(class,name) values ("nsd1902","tom");
Query OK, 1 row affected (0.05 sec)

mysql> insert into  t2  values (null,null,null,null);
ERROR 1048 (23000): Column 'name' cannot be null  //不允许赋 null 值
MariaDB [db2]>

MariaDB [db2]> select  * from db2.t1;  //查看记录
+---------+------+-----+-------+
| class   | name | age | likes |
+---------+------+-----+-------+
| NULL    | bob  |  29 | c,d   |
| nsd1902 | tom  |  19 | a,b   |
+---------+------+-----+-------+
2 rows in set (0.00 sec)

mysql>
```

2. 案例 2：修改表结构

- **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

- **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

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

2）为 tea6 表添加一个 address 字段

添加前：

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

添加 address 字段：

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

添加后（默认作为最后一个字段）：

```
mysql> DESC tea6;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| id      | int(4)      | YES  |     | NULL    |       |
| name    | varchar(4)  | NO   |     | NULL    |       |
| age     | int(2)      | NO   |     | NULL    |       |
| address | varchar(48) | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
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：

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

确认添加结果：

```
mysql> DESC tea6;
+---------+--------------------+------+-----+---------+-------+
| Field   | Type               | Null | Key | Default | Extra |
+---------+--------------------+------+-----+---------+-------+
| id      | int(4)             |YES   |     | NULL    |       |
| name    | varchar(4)         | NO   |     | NULL    |       |
| age     | int(2)             | NO   |     | NULL    |       |
| gender  | enum('boy','girl') | YES  |     | NULL    |       |
| address | varchar(48)        | YES  |     | NULL    |       |
+---------+--------------------+------+-----+---------+-------+
```

```
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：

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

确认修改结果：

```
mysql> DESC tea6;
+---------+--------------------+------+-----+---------+-------+
| Field   | Type               | Null | Key | Default | Extra |
+---------+--------------------+------+-----+---------+-------+
| id      | int(4)             | YES  |     | NULL    |       |
| name    | varchar(4)         | NO   |     | NULL    |       |
| age     | int(2)             | NO   |     | NULL    |       |
| sex     | enum('boy','girl') | NO   |     | NULL    |       |
| address | varchar(48)        | YES  |     | NULL    |       |
+---------+--------------------+------+-----+---------+-------+
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：

```
mysql> ALTER TABLE tea6 DROP sex;                                    //删除操作
Query OK, 0 rows affected (0.52 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESC tea6;                                                    //确认删除结果
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| id      | int(4)      | YES  |     | NULL    |       |
| name    | varchar(4)  | NO   |     | NULL    |       |
| age     | int(2)      | NO   |     | NULL    |       |
| address | varchar(48) | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

3. 案例 3：index 普通索引

- **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

- **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：

```
mysql> create database home;
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：

```
mysql> USE home;
Database changed
mysql> CREATE TABLE tea4(
    -> id char(6) NOT NULL,
    -> name varchar(6) NOT NULL,
    -> age int(3) NOT NULL,
    -> gender ENUM('boy','girl') DEFAULT 'boy',
    -> INDEX(id),INDEX(name)
    -> );
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：

```
mysql> DESC tea4;
+--------+--------------------+------+-----+---------+-------+
| Field  | Type               | Null | Key | Default | Extra |
+--------+--------------------+------+-----+---------+-------+
| id     | char(6)            | NO   | MUL | NULL    |       |
| name   | varchar(6)         | NO   | MUL | NULL    |       |
| age    | int(3)             | NO   |     | NULL    |       |
| gender | enum('boy','girl') | YES  |     | boy     |       |
+--------+--------------------+------+-----+---------+-------+
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：

```
mysql> drop INDEX name ON tea4;                    //删除 name 字段的索引
Query OK, 0 rows affected (0.18 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESC tea4;                                  //确认删除结果
+--------+--------------------+------+-----+---------+-------+
| Field  | Type               | Null | Key | Default | Extra |
+--------+--------------------+------+-----+---------+-------+
| id     | char(6)            | NO   | MUL | NULL    |       |
| name   | varchar(6)         | NO   |     | NULL    |       |
| age    | int(3)             | NO   |     | NULL    |       |
| gender | enum('boy','girl') | YES  |     | boy     |       |
+--------+--------------------+------+-----+---------+-------+
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：

```
mysql> CREATE INDEX nianling ON tea4(age);      //针对指定字段创建索引
Query OK, 0 rows affected (0.62 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESC tea4;                                    //确认创建结果
+--------+--------------------+------+-----+---------+-------+
| Field  | Type               | Null | Key | Default | Extra |
+--------+--------------------+------+-----+---------+-------+
| id     | char(6)            | NO   | MUL | NULL    |       |
| name   | varchar(6)         | NO   |     | NULL    |       |
| age    | int(3)             | NO   | MUL | NULL    |       |
| gender | enum('boy','girl') | YES  |     | boy     |       |
+--------+--------------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

4）查看指定表的索引信息

使用 SHOW INDEX 指令：

```
mysql> SHOW INDEX FROM tea4\G
*************************** 1. row ***************************
       Table: tea4
   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                          //使用 B 树算法
      Comment:
Index_comment:
*************************** 2. row ***************************
       Table: tea4
   Non_unique: 1
     Key_name: nianling                       //索引名称
 Seq_in_index: 1
  Column_name: age                            //字段名称
    Collation: A
  Cardinality: 0
     Sub_part: NULL
       Packed: NULL
         Null:
   Index_type: BTREE
      Comment:
Index_comment:
2 rows in set (0.00 sec)
Query OK, 0 rows affected (0.30 sec)
Mysql>
Query OK, 0 rows affected (0.47 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

4. 案例 4：primary key 主键

- **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

- **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

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

或者：

```
mysql> CREATE TABLE db2.biao02(
    -> id int(4),
    -> name varchar(8),
    -> PRIMARY KEY(id)                           //所有字段定义完，最后指定
    -> );
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：

```
mysql> CREATE TABLE db2.tea6(
    -> id int(4) AUTO_INCREMENT,
    -> name varchar(4) NOT NULL,
    -> age int(2) NOT NULL,
    -> PRIMARY KEY(id)
    -> );
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）：

```
mysql> DESC db2.biao01;
+-------+------------+------+-----+---------+-------+
| Field | Type       | Null | Key | Default | Extra |
+-------+------------+------+-----+---------+-------+
| id    | int(4)     | NO   | PRI | NULL    |       |
```

```
| name  | varchar(8) | YES  |     | NULL    |       |
+-------+------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

清除操作：

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

清除后（无主键）：

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

当尝试删除 tea6 表的主键时，会出现异常：

```
mysql> ALTER TABLE tea6 DROP PRIMARY KEY;
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：

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

然后再清除主键属性就 OK 了：

```
mysql> ALTER TABLE tea6 DROP PRIMARY KEY;                //清除主键
Query OK, 0 rows affected (0.39 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc tea6;                                        //确认清除结果
+-------+------------+------+-----+---------+-------+
| Field | Type       | Null | Key | Default | Extra |
+-------+------------+------+-----+---------+-------+
| id    | int(4)     | NO   |     | NULL    |       |
| name  | varchar(4) | NO   |     | NULL    |       |
| age   | int(2)     | NO   |     | NULL    |       |
+-------+------------+------+-----+---------+-------+
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：

```
mysql> ALTER TABLE tea6 ADD PRIMARY KEY(id);             //设置主键字段
Query OK, 0 rows affected (0.35 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESC tea6;                                            //确认设置结果
+-------+------------+------+-----+---------+-------+
| Field | Type       | Null | Key | Default | Extra |
```

```
+-------+------------+------+-----+---------+-------+
| id    | int(4)     | NO   | PRI | NULL    |       |
| name  | varchar(4) | NO   |     | NULL    |       |
| age   | int(2)     | NO   |     | NULL    |       |
+-------+------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

4）建表时创建复合主键

```
mysql>
mysql> create table  db2.t6(
    -> class char(7),
    -> name  char(15),
    -> pay   enum("yes","no") default "no",
    -> primary key(class,name,pay)  //指定多个字段一起做主键
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> desc db2.t6;
+-------+-----------------+------+-----+---------+-------+
| Field | Type            | Null | Key | Default | Extra |
+-------+-----------------+------+-----+---------+-------+
| class | char(7)         | NO   | PRI |         |       |
| name  | char(15)        | NO   | PRI |         |       |
| pay   | enum('yes','no') | NO   | PRI | no      |       |
+-------+-----------------+------+-----+---------+-------+
3 rows in set (0.01 sec)

mysql>
```

5. 案例 5：foreign key 外键

- **问题**

具体要求如下：

- 创建员工表 yg
- 创建工资表 gz ,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

yg_id	name
1	Jerry
2	Tom

表-2 工资表 gz 的数据

gz_id	Name	gz
1	Jerry	12000
2	Tom	8000

步骤二：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

```
mysql> CREATE TABLE yg(
    -> yg_id int primary key AUTO_INCREMENT,
    -> name char(16)
    -> )engine=innodb;
Query OK, 0 rows affected (0.15 sec)
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：

```
mysql> CREATE TABLE gz(
    -> gz_id  int,
    -> name char(16) ,
    -> gz float(7,2) ,
    -> FOREIGN KEY(gz_id) REFERENCES yg(yg_id)  //创建外键
    -> ON UPDATE CASCADE ON DELETE CASCADE //同步更新、同步删除
    -> )engine=innodb;
Query OK, 0 rows affected (0.23 sec)
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：

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

确认 yg 表的数据记录：

```
mysql> SELECT * FROM yg;
+-------+-------+
| yg_id | name  |
+-------+-------+
|     1 | Jerry |
|     2 | Tom   |
+-------+-------+
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）：

```
mysql> INSERT INTO gz(gz_id,name,gz)
    -> VALUES(1,'Jerry',12000),(2,'Tom',8000)
    -> ;
Query OK, 2 rows affected (0.06 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

确认 gz 表的数据记录：

```
mysql> SELECT * FROM gz;
+-------+-------+----------+
| gz_id | name  | gz       |
+-------+-------+----------+
|     1 | Jerry | 12000.00 |
|     2 | Tom   |  8000.00 |
+-------+-------+----------+
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：

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

确认修改结果：

```
mysql> SELECT * FROM yg;
+-------+-------+
| yg_id | name  |
+-------+-------+
|     2 | Tom   |
|  1234 | Jerry |
+-------+-------+
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：

```
mysql> SELECT * FROM gz;
+-------+-------+----------+
| gz_id | name  | gz       |
+-------+-------+----------+
|  1234 | Jerry | 12000.00 |
|     2 | Tom   |  8000.00 |
+-------+-------+----------+
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：

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

确认删除结果：

```
mysql> SELECT * FROM yg;
+-------+------+
| yg_id | name |
+-------+------+
|     2 | Tom  |
+-------+------+
1 row in set (0.00 sec)
```

查看 gz 表中的变化（Jerry 的记录也没了）：

```
mysql> SELECT * FROM gz;
+-------+------+---------+
| gz_id | name | gz      |
+-------+------+---------+
|     2 | Tom  | 8000.00 |
+-------+------+---------+
```

```
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：

```
mysql> SHOW CREATE TABLE gz\G
*************************** 1. row ***************************
       Table: gz
Create Table: CREATE TABLE `gz` (
  `gz_id` int(4) NOT NULL,
  `name` char(16) NOT NULL,
  `gz` float(7,2) NOT NULL DEFAULT '0.00',
  KEY `name` (`name`),
  KEY `gz_id` (`gz_id`),
  CONSTRAINT `gz_ibfk_1` FOREIGN KEY (`gz_id`) REFERENCES `yg` (`yg_id`) ON DELETE
CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8
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：

```
mysql> ALTER TABLE gz DROP FOREIGN KEY gz_ibfk_1;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

确认删除结果：

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