

NSD RDBMS1 DAY02

1. [案例1：约束条件](#)
2. [案例2：修改表结构](#)
3. [案例3：index 普通索引](#)
4. [案例4：primary key 主键](#)
5. [案例5：foreign key 外键](#)

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

```
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

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

01. mysql>
- 02.
03. mysql> create database db2; //建库
04. Query OK, 1 row affected (0.00 sec)
- 05.
06. mysql> use db2; //切换库

[Top](#)

```

07. Database changed
08. mysql> create table t2 ( //建表
09.     -> class char(9),
10.     -> name char(10) not null ,
11.     -> age tinyint not null default 19 ,
12.     -> likes set("a","b","c","d") default "a,b"
13.     -> );
14. Query OK, 0 rows affected (0.05 sec)
15.
16.
17.
18.
19.
20. mysql> desc t2; //查看表结构
21. +-----+-----+-----+-----+
22. | Field | Type          | Null | Key | Default | Extra |
23. +-----+-----+-----+-----+
24. | class | char(9)       | YES  |     | NULL    |      |
25. | name  | char(10)      | NO   |     | NULL    |      |
26. | age   | tinyint(4)    | NO   |     | 19      |      |
27. | likes | set('a','b','c','d') | YES  |     | a,b     |      |
28. +-----+-----+-----+-----+
29. 4 rows in set (0.00 sec)
30.
31. mysql>
32.
33. mysql> insert into t2 values (null,"bob",29,"c,d");
34. Query OK, 1 row affected (0.05 sec)
35.
36. mysql> insert into t2(class,name) values ("nsd1902","tom");
37. Query OK, 1 row affected (0.05 sec)
38.
39. mysql> insert into t2 values (null,null,null,null);
40. ERROR 1048 (23000): Column 'name' cannot be null //不允许赋null值
41. MariaDB [db2]>
42.
43. MariaDB [db2]> select * from db2.t1; //查看记录
44. +-----+-----+-----+
45. | class | name | age | likes |
46. +-----+-----+-----+
47. | NULL  | bob  | 29  | c,d   |

```

[Top](#)

```

48.      | nsd1902 | tom | 19 | a,b |
49.      +-----+-----+-----+
50.      2 rows in set (0.00 sec)
51.
52.      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

```

01.      Mysql> create database studb;
02.      mysql> CREATE TABLE studb.tea6(
03.      -> id int(4) ,
04.      -> name varchar(4) NOT NULL,
05.      -> age int(2) NOT NULL
06.      -> );
07.      Query OK, 0 rows affected (0.34 sec)
08.      mysql>

```

2) 为tea6表添加一个address字段

```

01.      mysql> DESC tea6;
02.      +-----+-----+-----+-----+
03.      | Field | Type      | Null | Key | Default | Extra |
04.      +-----+-----+-----+-----+
05.      | id    | int(4)    | YES  |     | NULL    |       |
06.      | name  | varchar(4)| NO   |     | NULL    |       |
07.      | age   | int(2)    | NO   |     | NULL    |       |

```

[Top](#)

08. +-----+-----+-----+-----+
09. 3 rows in set (0.00 sec)

添加address字段：

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

添加后（默认作为最后一个字段）：

01. mysql> DESC tea6;
02. +-----+-----+-----+-----+
03. | Field | Type | Null | Key | Default | Extra |
04. +-----+-----+-----+-----+
05. | id | int(4) | YES | | NULL | |
06. | name | varchar(4) | NO | | NULL | |
07. | age | int(2) | NO | | NULL | |
08. | address | varchar(48) | YES | | NULL | |
09. +-----+-----+-----+-----+
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：

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

确认添加结果：

01. mysql> DESC tea6;
02. +-----+-----+-----+-----+
03. | Field | Type | Null | Key | Default | Extra |
04. +-----+-----+-----+-----+
05. | id | int(4) | YES | | NULL | |

[Top](#)

```

06.  | name  | varchar(4)      | NO  |  | NULL  |  |
07.  | age   | int(2)          | NO  |  | NULL  |  |
08.  | gender | enum('boy','girl') | YES |  | NULL  |  |
09.  | 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：

```

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

```

确认修改结果：

```

01.  mysql> DESC tea6;
02.  +-----+-----+-----+-----+
03.  | Field | Type          | Null | Key | Default | Extra |
04.  +-----+-----+-----+-----+
05.  | id    | int(4)        | YES  |     | NULL    |       |
06.  | name  | varchar(4)    | NO   |     | NULL    |       |
07.  | age   | int(2)        | NO   |     | NULL    |       |
08.  | sex   | enum('boy','girl') | NO   |     | NULL    |       |
09.  | 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：

```

01.  mysql> ALTER TABLE tea6 DROP sex;           //删除操作
02.  Query OK, 0 rows affected (0.52 sec)
03.  Records: 0 Duplicates: 0 Warnings: 0
04.

```

[Top](#)

```

05.  mysql> DESC tea6;                                     //确认删除结果
06.  +-----+-----+-----+-----+-----+
07.  | Field | Type      | Null | Key | Default | Extra |
08.  +-----+-----+-----+-----+-----+
09.  | id    | int(4)    | YES  |     | NULL    |      |
10.  | name  | varchar(4)| NO   |     | NULL    |      |
11.  | age   | int(2)    | NO   |     | NULL    |      |
12.  | address | varchar(48)| YES  |     | NULL    |      |
13.  +-----+-----+-----+-----+-----+
14.  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：

```

01.  mysql> create database home;
02.  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：

```

01.  mysql> USE home;
02.  Database changed
03.  mysql> CREATE TABLE tea4(
04.      -> id char(6) NOT NULL,
05.      -> name varchar(6) NOT NULL,
06.      -> age int(3) NOT NULL,
07.      -> gender ENUM('boy','girl') DEFAULT 'boy',

```

[Top](#)

08. -> INDEX(id),INDEX(name)
09. ->);
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：

01. mysql> DESC tea4;
02. +-----+-----+-----+-----+-----+
03. | Field | Type | Null | Key | Default | Extra |
04. +-----+-----+-----+-----+-----+
05. | id | char(6) | NO | MUL | NULL | |
06. | name | varchar(6) | NO | MUL | NULL | |
07. | age | int(3) | NO | | NULL | |
08. | gender | enum('boy','girl') | YES | | boy | |
09. +-----+-----+-----+-----+-----+
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：

01. mysql> drop INDEX name ON tea4; //删除name字段的索引
02. Query OK, 0 rows affected (0.18 sec)
03. Records: 0 Duplicates: 0 Warnings: 0
- 04.
05. mysql> DESC tea4; //确认删除结果
06. +-----+-----+-----+-----+-----+
07. | Field | Type | Null | Key | Default | Extra |
08. +-----+-----+-----+-----+-----+
09. | id | char(6) | NO | MUL | NULL | |
10. | name | varchar(6) | NO | | NULL | |
11. | age | int(3) | NO | | NULL | |
12. | gender | enum('boy','girl') | YES | | boy | |
13. +-----+-----+-----+-----+-----+
14. 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：

[Top](#)

```

01.  mysql> CREATE INDEX nianling ON tea4(age);    //针对指定字段创建索引
02.  Query OK, 0 rows affected (0.62 sec)
03.  Records: 0 Duplicates: 0 Warnings: 0
04.
05.  mysql> DESC tea4;                            //确认创建结果
06.  +-----+-----+-----+-----+-----+
07.  | Field | Type          | Null | Key | Default | Extra |
08.  +-----+-----+-----+-----+-----+
09.  | id    | char(6)       | NO   | MUL | NULL    |      |
10.  | name  | varchar(6)    | NO   |     | NULL    |      |
11.  | age   | int(3)        | NO   | MUL | NULL    |      |
12.  | gender | enum('boy','girl') | YES  |     | boy     |      |
13.  +-----+-----+-----+-----+-----+
14.  4 rows in set (0.00 sec)

```

4) 查看指定表的索引信息

使用SHOW INDEX 指令：

```

01.  mysql> SHOW INDEX FROM tea4\G
02.  ***** 1. row *****
03.      Table: tea4
04.      Non_unique: 1
05.      Key_name: id
06.      Seq_in_index: 1
07.      Column_name: id
08.      Collation: A
09.      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

```

[Top](#)


```
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

```
01.      Mysql> create database db2;
02.      mysql> CREATE TABLE db2.biao01(
03.          -> id int(4) PRIMARY KEY,           //直接在字段定义时约束
04.          -> name varchar(8)
05.          -> );
06.      Query OK, 0 rows affected (0.19 sec)
```

或者：

```
01.      mysql> CREATE TABLE db2.biao02(
02.          -> id int(4),
```

[Top](#)

```

03.      -> name varchar(8),
04.      -> PRIMARY KEY(id)           //所有字段定义完，最后指定
05.      -> );
06.  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：

```

01.  mysql> CREATE TABLE db2.tea6(
02.      -> id int(4) AUTO_INCREMENT,
03.      -> name varchar(4) NOT NULL,
04.      -> age int(2) NOT NULL,
05.      -> PRIMARY KEY(id)
06.      -> );
07.  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）：

```

01.  mysql> DESC db2.biao01;
02.  +-----+-----+-----+-----+
03.  | Field | Type   | Null | Key | Default | Extra |
04.  +-----+-----+-----+-----+
05.  | id    | int(4) | NO   | PRI | NULL    |       |
06.  | name  | varchar(8) | YES |     | NULL    |       |
07.  +-----+-----+-----+-----+
08.  2 rows in set (0.00 sec)

```

清除操作：

```

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

```

[Top](#)

清除后（无主键）：

```

01.  mysql> DESC db2.biao01;
02.  +-----+-----+-----+-----+-----+
03.  | Field | Type      | Null | Key | Default | Extra |
04.  +-----+-----+-----+-----+-----+
05.  | id    | int(4)    | NO   |     | NULL    |       |
06.  | name  | varchar(8)| YES  |     | NULL    |       |
07.  +-----+-----+-----+-----+-----+
08.  2 rows in set (0.00 sec)

```

当尝试删除tea6表的主键时，会出现异常：

```

01.  mysql> ALTER TABLE tea6 DROP PRIMARY KEY;
02.  ERROR 1075 (42000): Incorrect table definition; there can be only one auto column an

```



这是因为tea6表的主键字段id具有AUTO_INCREMENT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：

```

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

```

然后再清除主键属性就OK了：

```

01.  mysql> ALTER TABLE tea6 DROP PRIMARY KEY;           //清除主键
02.  Query OK, 0 rows affected (0.39 sec)
03.  Records: 0 Duplicates: 0 Warnings: 0
04.
05.  mysql> desc tea6;                                     //确认清除结果
06.  +-----+-----+-----+-----+-----+
07.  | Field | Type      | Null | Key | Default | Extra |
08.  +-----+-----+-----+-----+-----+
09.  | id    | int(4)    | NO   |     | NULL    |       |
10.  | name  | varchar(4)| NO   |     | NULL    |       |
11.  | age   | int(2)    | NO   |     | NULL    |       |
12.  +-----+-----+-----+-----+-----+
13.  3 rows in set (0.01 sec)

```

[Top](#)

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：

```
01. mysql> ALTER TABLE tea6 ADD PRIMARY KEY(id);           //设置主键字段
02. Query OK, 0 rows affected (0.35 sec)
03. Records: 0 Duplicates: 0 Warnings: 0
04.
05. mysql> DESC tea6;                                       //确认设置结果
06. +-----+-----+-----+-----+-----+
07. | Field | Type      | Null | Key | Default | Extra |
08. +-----+-----+-----+-----+-----+
09. | id    | int(4)    | NO   | PRI | NULL    |      |
10. | name  | varchar(4)| NO   |     | NULL    |      |
11. | age   | int(2)    | NO   |     | NULL    |      |
12. +-----+-----+-----+-----+-----+
13. 3 rows in set (0.00 sec)
```

4) 建表时创建复合主键

```
01. mysql>
02. mysql> create table db2.t6(
03.     -> class char(7),
04.     -> name char(15),
05.     -> pay enum("yes","no") default "no",
06.     -> primary key(class,name,pay) //指定多个字段一起做主键
07.     -> );
08. Query OK, 0 rows affected (0.04 sec)
09.
10. mysql> desc db2.t6;
11. +-----+-----+-----+-----+-----+
12. | Field | Type      | Null | Key | Default | Extra |
13. +-----+-----+-----+-----+-----+
14. | class | char(7)   | NO   | PRI |         |      |
15. | name  | char(15)  | NO   | PRI |         |      |
16. | pay   | enum('yes','no') | NO   | PRI | no      |      |
17. +-----+-----+-----+-----+-----+
18. 3 rows in set (0.01 sec)
19.
```

[Top](#)

20. 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

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

```
01. mysql> CREATE TABLE yg(  
02.     -> yg_id int primary key AUTO_INCREMENT,  
03.     -> name char(16)  
04.     -> )engine=innodb;  
05. Query OK, 0 rows affected (0.15 sec)  
06. 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：

```
01. mysql> CREATE TABLE gz(  
02.     -> gz_id int,  
03.     -> name char(16),  
04.     -> gz float(7,2),  
05.     -> FOREIGN KEY(gz_id) REFERENCES yg(yg_id) //创建外键
```

[Top](#)

06. -> ON UPDATE CASCADE ON DELETE CASCADE //同步更新、同步删除
07. ->)engine=innodb;
08. Query OK, 0 rows affected (0.23 sec)
09. 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：

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

确认yg表的数据记录：

01. mysql> SELECT * FROM yg;
02. +-----+-----+
03. | yg_id | name |
04. +-----+-----+
05. | 1 | Jerry |
06. | 2 | Tom |
07. +-----+-----+
08. 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）：

01. mysql> INSERT INTO gz(gz_id,name,gz)
02. -> VALUES(1,'Jerry',12000),(2,'Tom',8000)
03. -> ;
04. Query OK, 2 rows affected (0.06 sec)
05. Records: 2 Duplicates: 0 Warnings: 0

确认gz表的数据记录：

[Top](#)

```

01.  mysql> SELECT * FROM gz;
02.  +-----+-----+-----+
03.  | gz_id | name | gz      |
04.  +-----+-----+-----+
05.  | 1 | Jerry | 12000.00 |
06.  | 2 | Tom   | 8000.00  |
07.  +-----+-----+-----+
08.  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：

```

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

```

确认修改结果：

```

01.  mysql> SELECT * FROM yg;
02.  +-----+-----+
03.  | yg_id | name |
04.  +-----+-----+
05.  | 2 | Tom |
06.  | 1234 | Jerry |
07.  +-----+-----+
08.  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：

```

01.  mysql> SELECT * FROM gz;
02.  +-----+-----+-----+
03.  | gz_id | name | gz      |
04.  +-----+-----+-----+
05.  | 1234 | Jerry | 12000.00 |
06.  | 2 | Tom   | 8000.00  |
07.  +-----+-----+-----+

```

[Top](#)

08. 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：

01. mysql> DELETE FROM yg WHERE name='Jerry';

02. Query OK, 1 row affected (0.05 sec)

确认删除结果：

01. mysql> SELECT * FROM yg;

02. +-----+-----+

03. | yg_id | name |

04. +-----+-----+

05. | 2 | Tom |

06. +-----+-----+

07. 1 row in set (0.00 sec)

查看gz表中的变化（Jerry的记录也没了）：

01. mysql> SELECT * FROM gz;

02. +-----+-----+-----+

03. | gz_id | name | gz |

04. +-----+-----+-----+

05. | 2 | Tom | 8000.00 |

06. +-----+-----+-----+

07. 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：

01. mysql> SHOW CREATE TABLE gz\G

02. ***** 1. row *****

03. Table: gz

04. Create Table: CREATE TABLE `gz` (

05. `gz_id` int(4) NOT NULL,

[Top](#)


```

06.     `name` char(16) NOT NULL,
07.     `gz` float(7,2) NOT NULL DEFAULT '0.00',
08.     KEY `name` (`name`),
09.     KEY `gz_id` (`gz_id`),
10.     CONSTRAINT `gz_ibfk_1` FOREIGN KEY (`gz_id`) REFERENCES `yg` (`yg_id`) ON DELETE
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：

```

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

```

确认删除结果：

```

01. mysql> SHOW CREATE TABLE gz\G
02. ***** 1. row *****
03.      Table: gz
04.      Create Table: CREATE TABLE `gz` (
05.        `gz_id` int(4) NOT NULL,
06.        `name` char(16) NOT NULL,
07.        `gz` float(7,2) NOT NULL DEFAULT '0.00',
08.        KEY `name` (`name`),
09.        KEY `gz_id` (`gz_id`)
10.      ) ENGINE=InnoDB DEFAULT CHARSET=utf8
11. 1 row in set (0.00 sec)

```

[Top](#)