



链滴

MYSQL 之修改篇

作者: [vcjmhg](#)

原文链接: <https://ld246.com/article/1569939740124>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

对一个数据库系统来说最常见的无非就是增删改查操作，对于查询操作，我们在前边已经详细的叙述SELECT的使用，下边我们将分三个专题对增删改操作进行说明。

3.1 INSERT

INSERT语句主要用于向数据库表中插入一条或者多条记录。

INSERT语句的基本语法：

```
INSERT INTO <表名> (字段1, 字段2, ...) VALUES (值1, 值2, ...);
```

例如向students表中插入一条记录：

```
insert into students (class_id,name,gender,score) values(2,'张芳','f',82);
/*显示插入结果*/
select * from students;
```

运行结果为：

```
mysql> insert into students (class_id,name,gender,score) values(2,'张芳','F',82);
Query OK, 1 row affected (0.01 sec)

mysql> select * from students;
+----+-----+-----+-----+-----+
| id | class_id | name  | gender | score |
+----+-----+-----+-----+-----+
| 1  | 1        | 小明  | M      | 90    |
| 2  | 1        | 小红  | F      | 95    |
| 3  | 1        | 小军  | M      | 88    |
| 4  | 1        | 小米  | F      | 73    |
| 5  | 2        | 小白  | F      | 81    |
| 6  | 2        | 小兵  | M      | 55    |
| 7  | 2        | 小林  | M      | 85    |
| 8  | 3        | 小新  | F      | 91    |
| 9  | 3        | 小王  | M      | 89    |
| 10 | 3        | 小丽  | F      | 85    |
| 11 | NULL    | 李平  | M      | 95    |
| 12 | 2        | 张芳  | F      | 82    |
+----+-----+-----+-----+-----+
```

注意：

1. 我们并没有列出 id 字段，也没有列出id字段对应的值，这是因为id字段是一个自增主键，它的值以由数据库自己推算出来。此外，如果一个字段有默认值，那么在INSERT语句中也可以不出现。要注意，字段顺序不必和数据库表的字段顺序一致，但值的顺序必须和字段顺序一致。也就是说，可以写INSERT INTO students (score, gender, name, class_id) ...，但是对应的VALUES就得变成(80, 'M', '牛', 2)。

2.如果插入的字段值的顺序与数据库的字段顺序一致时我们并不需要列出属性值(但默认字段要用default来进行填充不可不写)即可以写成

```
insert into students values(default,2,'张芳','f',82);
```

另一方面我们还可以一次性添加多条记录，只需要在VALUES子句中指定多个记录值，每个记录是由(...)含的一组值：

```
INSERT INTO students (class_id, name, gender, score) VALUES
(1, '大宝', 'M', 87),
```

```
(2, '二宝', 'M', 81);
/*显示插入之后的结果*/
SELECT * FROM students;
```

运行结果:

```
mysql> INSERT INTO students (class_id, name, gender, score) VALUES
-> (1, '大宝', 'M', 87),
-> (2, '二宝', 'M', 81);
Query OK, 2 rows affected (0.00 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> /*显示插入之后的结果*/
mysql> SELECT * FROM students;
+----+-----+-----+-----+-----+
| id | class_id | name  | gender | score |
+----+-----+-----+-----+-----+
| 1  |         | 小明  | M      | 90    |
| 2  |         | 小红  | F      | 95    |
| 3  |         | 小军  | M      | 88    |
| 4  |         | 小米  | F      | 73    |
| 5  |         | 小白  | F      | 81    |
| 6  |         | 小兵  | M      | 55    |
| 7  |         | 小林  | M      | 85    |
| 8  |         | 小新  | F      | 91    |
| 9  |         | 小王  | M      | 89    |
| 10 |         | 小丽  | F      | 85    |
| 11 |        | 李平  | M      | 95    |
| 12 |         | 张芳  | F      | 82    |
| 14 |         | 大宝  | M      | 87    |
| 15 |         | 二宝  | M      | 81    |
+----+-----+-----+-----+-----+
14 rows in set (0.00 sec)
```

3.2 UPDATE

当我们需要修改数据库表中的记录时我们便需要使用UPDATE语句来进行实现,使用UPDATE,我们就以一次更新表中的一条或多条记录。。

UPDATE语句的基本语法为:

```
UPDATE <表名> SET 字段1=值1, 字段2=值2, ... WHERE ...;
```

例如, 我们想更新students表id=1的记录的name和score这两个字段, 先写出UPDATE students SET name='大牛', score=66, 然后在WHERE子句中写出需要更新的行的筛选条件id=1:

```
UPDATE students SET name='大牛', score=66 WHERE id=1;
SELECT * FROM students WHERE id=1;
```

运行结果:

```
mysql> UPDATE students SET name='大牛', score=66 WHERE id=1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> SELECT * FROM students WHERE id=1;
+----+-----+-----+-----+-----+
| id | class_id | name  | gender | score |
+----+-----+-----+-----+-----+
| 1  |         | 大牛  | M      | 66    |
+----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

注意到UPDATE语句的WHERE条件和SELECT语句的WHERE条件其实是一样的，因此完全可以一次新多条记录：

```
UPDATE students SET name='小牛', score=77 WHERE id>=5 AND id<=7;
SELECT * FROM students;
```

执行结果为：

```
mysql> UPDATE students SET name='小牛', score=77 WHERE id>=5 AND id<=7;
Query OK, 3 rows affected (0.00 sec)
Rows matched: 3  Changed: 3  Warnings: 0

mysql> SELECT * FROM students;
+----+-----+-----+-----+-----+
| id | class_id | name  | gender | score |
+----+-----+-----+-----+-----+
| 1  |         | 大牛  | M      | 66    |
| 2  |         | 小红  | F      | 95    |
| 3  |         | 小军  | M      | 88    |
| 4  |         | 小米  | F      | 73    |
| 5  |         | 小牛  | F      | 77    |
| 6  |         | 小牛  | M      | 77    |
| 7  |         | 小牛  | M      | 77    |
| 8  |         | 小新  | F      | 91    |
| 9  |         | 小王  | M      | 89    |
| 10 |         | 小丽  | F      | 85    |
| 11 |        | 李平  | M      | 95    |
| 12 |         | 张芳  | F      | 82    |
| 14 |         | 大宝  | M      | 87    |
| 15 |         | 二宝  | M      | 81    |
+----+-----+-----+-----+-----+
14 rows in set (0.00 sec)
```

注意：

UPDATE语句在执行的过程可以没有where条件进行限制

```
UPDATE students SET score=60;
```

但是要注意此时所有表的记录都会被更新，因此在通过UPDATE语句进行更新时一定要通过WHERE句筛选出自己所需要更改的字段。

3.3 DELETE

如果我们需要删除数据库中的记录我们需要使用DELETE语句。

DELETE语句的基本语法是：

```
DELETE FROM <表名> WHERE ...;
```

比如我们想删除students表中id=1的记录，就需要这么写：

```
DELETE FROM students WHERE id=1;
SELECT * FROM students;
```

执行结果：

```
mysql> DELETE FROM students WHERE id=1;
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM students;
+----+-----+-----+-----+-----+
| id | class_id | name  | gender | score |
+----+-----+-----+-----+-----+
| 2  | 1       | 小红  | F      | 95    |
| 3  | 1       | 小军  | M      | 88    |
| 4  | 1       | 小米  | F      | 73    |
| 5  | 2       | 小牛  | F      | 77    |
| 6  | 2       | 小牛  | M      | 77    |
| 7  | 2       | 小牛  | M      | 77    |
| 8  | 3       | 小新  | F      | 91    |
| 9  | 3       | 小王  | M      | 89    |
| 10 | 3       | 小丽  | F      | 85    |
| 11 | NULL    | 李平  | M      | 95    |
| 12 | 2       | 张芳  | F      | 82    |
| 14 | 1       | 大宝  | M      | 87    |
| 15 | 2       | 二宝  | M      | 81    |
+----+-----+-----+-----+-----+
13 rows in set (0.00 sec)
```

注意到DELETE语句的WHERE条件也是用来筛选需要删除的行，因此和UPDATE类似，DELETE语句可以一次删除多条记录：

```
DELETE FROM students WHERE id>=5 AND id<=7;
```

```
SELECT * FROM students;
```

执行结果：

```
mysql> DELETE FROM students WHERE id>=5 AND id<=7;
Query OK, 3 rows affected (0.00 sec)

mysql>
mysql> SELECT * FROM students;
+----+-----+-----+-----+-----+
| id | class_id | name  | gender | score |
+----+-----+-----+-----+-----+
| 2  | 1       | 小红  | F      | 95    |
| 3  | 1       | 小军  | M      | 88    |
| 4  | 1       | 小米  | F      | 73    |
| 8  | 3       | 小新  | F      | 91    |
| 9  | 3       | 小王  | M      | 89    |
| 10 | 3       | 小丽  | F      | 85    |
| 11 | NULL    | 李平  | M      | 95    |
| 12 | 2       | 张芳  | F      | 82    |
| 14 | 1       | 大宝  | M      | 87    |
| 15 | 2       | 二宝  | M      | 81    |
+----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

最后，要特别小心的是，和UPDATE类似，不带WHERE条件的DELETE语句会删除整个表的数据：

```
DELETE FROM students;
```

这时，整个表的所有记录都会被删除。所以，在执行DELETE语句时也要非常小心，最好先用SELECT句来测试WHERE条件是否筛选出了期望的记录集，然后再用DELETE删除。