

# 一、插入数据

## 1. 插入一条数据

### (1) 基本写法

使用"insert into"和 "values"关键字:

**insert into 表名 (关键字1, 关键字2, ...关键字n) values (值1, 值2, 值3,... 值n);**

**两个括号的key和value必须一一对应!**

```
mysql> insert into teacher (id, name, phone, address) values(1, 'Frank',  
'188888888',  
'Shanghai');  
Query OK, 1 row affected (0.00 sec)
```

### (2) 有关两个括号内的顺序问题

字段(field)的顺序是可以改变的, 但是value要与之对应:

```
mysql> insert into teacher (phone, address, id, name) values  
( '1899999999', 'Beijing',  
2, 'Jeff');  
Query OK, 1 row affected (0.00 sec)
```

### (3) 进(tou)阶(lan)写法

当然还有一种偷懒的写法:

**insert into 表名 values(值1, 值2, 值3,... 值n)**

**但是, 这种写法必须严格按照字段的顺序去写!**

```
mysql> insert into teacher values(3, 'Tom', '100000000', 'Nanjing');  
Query OK, 1 row affected (0.00 sec)
```

### (4) 赋值问题

另外还有一些其他的写法, 让我们先看一下这张表的信息:

```
mysql> desc teacher;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type          | Null | Key | Default | Extra          |  
+-----+-----+-----+-----+-----+-----+  
| id    | int(11)       | NO   | PRI | NULL    | auto_increment |  
| name  | varchar(30)   | NO   |     | NULL    |                |  
| phone | varchar(20)   | YES  |     | NULL    |                |  
| address | varchar(100) | YES  |     | 暂时未知 |                |  
+-----+-----+-----+-----+-----+-----+
```

在创建时，我们设置了id为"auto\_increment"，因此其值可以自增，所以我们可以给id赋值为NULL(这和id在Null一栏为NO并不冲突，因为有自增设置)，便有了下面的写法：

```
mysql> insert into teacher values(NULL, 'Tom', NULL, NULL);
Query OK, 1 row affected (0.00 sec)

mysql> insert into teacher values(NULL, 'Tom', NULL, NULL);
Query OK, 1 row affected (0.00 sec)
```

对于name，我们有设置其值不能为NULL，因此在给name赋值为NULL时会报错：

```
mysql> insert into teacher values(NULL, NULL, NULL, NULL);
ERROR 1048 (23000): Column 'name' cannot be null
```

对于address，我们对其设置了默认值(default)为"暂时未知"，所以在为其赋值为"DEAFULT"(大写装比，小写易读，都正确)时，会默认为"暂时未知"：

```
mysql> insert into teacher values(NULL, 'Jerry', NULL, DEFAULT);
Query OK, 1 row affected (0.00 sec)
```

还有一种写法，如果不想把values()中的内容写全的话，可以有如下写法，还是需要注意，一一对应！

```
mysql> insert into teacher (name, phone, address) values('Jerry', NULL, default);
Query OK, 1 row affected (0.00 sec)
```

以上的所有运行结果课在第二个point中查看。

## 2.查看表中插入的数据

使用"select from"关键字：

先介绍最基本的操作：

**select \* from 表名;**

```
mysql> select * from teacher;
+----+-----+-----+-----+
| id | name  | phone    | address |
+----+-----+-----+-----+
| 1  | Frank | 188888888 | Shanghai |
| 2  | Jeff  | 1899999999 | Beijing  |
| 3  | Tom   | 100000000  | Nanjing  |
| 4  | Tom   | NULL      | NULL     |
| 5  | Tom   | NULL      | NULL     |
| 6  | Jerry | NULL      | 暂时未知 |
| 7  | Jerry | NULL      | 暂时未知 |
+----+-----+-----+-----+
7 rows in set (0.00 sec)
```

这种写法其实不是特别规范，\*代表全部的意思，之所以不规范是因为使用\*时在某些场景下性能损失较大。

