

# 一. INSERT INTO

INSERT INTO是一种用来在**表的末尾**插入**行**的语句。（不能插入列，记住！）

1. 插入完整的行——这个有两种方法。

①不给出列名的形式：INSERT INTO *tablename*

VALUES('...', '...');

很显然，这种做法必须要遵从原来表中列的顺序。

**技巧：**在之前可以先SELECT \* FROM *tablename*, 这样就可以看出列的顺序。

注意NULL值的使用——如果不想给出值，可以使用NULL（假设允许的话）

——如果想使用默认值（比如自动增量的列），也可以使用NULL

②给出列名的形式：INSERT INTO *tablename(columnname)* VALUES(...)

这样的好处是，即使表的结构改变，我们的命令仍然能正确地工作。（**推荐**）

2. 插入多个行：其实就是使用多条INSERT INTO 语句，每条INSERT INTO语句用;结尾。

3. 插入检索的数据：其实就是在VALUES后面加上SELECT语句。

（使用INSERT INTO的技巧——INSERT INTO的操作可能很耗时，而且可能会降低待处理的SELECT语句的性能，在必要的时候，我们可以在INSERT后面加上**LOW\_PRIORITY**，以降低INSERT语句的优先级）

INSERT INTO 格式：mysql> INSERT INTO 409\_student (

```
->     stu_id,
->     stu_sex,
->     stu_gpa)
-> VALUES(
->     80181,
->     'male',
->     4.0);
```

# 二. UPDATE

UPDATE语句专门是用来**修改表中的数据**的。

1. 使用格式：UPDATE *tablename* SET ... = ... WHERE ... = ...;

2. 更新多个列的时候，只要在每个“列=值”对后面添加逗号（,）即可。

3. 关于NULL值：有的时候，要删除特定单元格上的数据，就可以将他设为NULL

4. IGNORE: 在使用UPDATE修改多个列的时候，如果有一行出现错误，那么整个UPDATE操作将被取消。为了防止这种事情的发生，可以使用IGNORE关键字：

UPDATE IGNORE customers

## 三. DELETE

顾名思义，DELETE语句是专门用来删除相应的行的。

1. DELETE的使用更加简单了！DELETE FROM customers WHERE cust\_id = ...;
2. 如果想从表中删除所有的行，不要使用DELETE语句，使用TRUNCATE TABLE语句，后者是专门用来删除表的。所以从性能上看，后者的效率更高。

## 四. 一些原则

在mysql下，有一些奇怪的规则——比如，**没有撤销按钮**（undo），而在使用UPDATE和DELETE的时候，不适用WHERE子句，很可能对表的**每一行**都进行操作。所以我们制定了一些规则：

1. 绝对不要使用不带WHERE子句的DELETE和UPDATE子句。
2. 在使用WHERE之前，应该先用SELECT语句过滤一下，已查看是否是相应的语句。
3. 确保每个表都有primary key,然后在WHERE子句中使用它。