

# 插入数据

---

insert可以完成：

- 插入完整的行
- 行的一部分
- 多行
- 插入某些查询的结果

## 插入完整的行

```
insert into table_name
values(
    col0,
    col1,
    col2,
    ...,
    NULL      -- 必须要进行赋值, 如果不想给出一个值, 必须赋值为NULL
);
-- 插入一条新的记录, 值为 (col0,...)
```

以上的插入顺序默认和表中的列的顺序一致。但也可以指定

```
insert into table_name(c1,c2,c3)
values (...);    -- 将会按照c1,c2,c3字段的顺序插入
```

其中, 未显示指明的字段 (列) , 将会默认插入NULL值或其他默认值。

## 插入多行

很简单：

```
insert into table_name(c1,c2,c3)
values (...);
insert into table_name(c2,c3,c1)    -- 这里的插入顺序和上一个不同
values (...);
...
```

如果每天insert的插入顺序相同, 可以合并上面的几个insert,

```
insert into table_name(c1,c2,c3)
values (...),
values (...),
...;
```

## 插入select返回的值

insert select 语句。

很显然, select返回的字段如果是主键, 必须要保证唯一性。

```
insert into table_name(c1,c2)
select c1,c2 from some_table;    -- 这里的c1,c2和上面的c1,c2没关系
```

insert和select中没必要使用相同的字段名称, 插入按照table\_name(c1,c2)的顺序插入即可。

---

## UPDATE, DELETE

---

### 更新数据

```
update customers          -- 指定表名
set cust_email='...'      -- 指定某个字段进行update
where cust_id=10005;       -- 更新cust_id为10005的cust_email字段的记录值
```

如果没有where过滤条件, 将会作用于整个cust\_email字段, 也就是所有行的cust\_email字段值都会被update。

更新多个字段:

```
update customers
set cust_name='..',
    cust_email='..'
where cust_id=2005;    -- 不可缺少
```

为了删除某个字段的值, 可以将其设置为NULL, 即

```
update customers set cust_email=NULL where cust_id=20005;
```

### 删除数据

delete可以删除:

- 表中特定的行
- 表中所有的行 (不加where过滤条件时将会发生), 删除所有的行, 不是删除表, 只是表中无任何记录

用法如下:

```
delete from customers where cust_id=10006;
```

delete删除整行, 不是某个字段。如果删除某个字段, 使用update, 将其设置为NULL。