

# 数据的基本操作

## 1、新增数据

方式1、全字段插入：

语法格式：

```
insert into 表名 values(值1, 值2, ... 最后一个值);
```

解释：

- 值的顺序必须与所有字段的顺序一致。
- 值的数据类型也必须与字段定义的数据类型一致。

举例（给表中插入一条完整的数据）：

```
insert into 表名 values(3, 'geektime', 28);
```

方式2、部分字段插入：

语法格式：

```
insert into 表名 (字段1, 字段2, 字段3) values(值1, 值2, 值3);
```

解释：字段的顺序可以随意，但值的顺序必须要与前面的字段顺序**一一对应**，数据类型也要一致。

举例（给表中的指定字段插入数据）：

```
insert into 表名 (id, name) values(4, 'xusong');
```

## 2、查询数据

查询数据的操作，占SQL日常操作的95%以上。

语法格式：

```
select xxx from 表名;
```

举例：

查询表中的所有数据：

```
select * from 表名;
```

查询表中 name、age 这两个字段的数据：

```
select name, age from 表名;
```

查询表中 id=2 的数据：

```
select * from 表名 where id = 2;
```

### 3、修改数据

语法格式：

```
update 表名 set (字段1 = 新值1, 字段2 = 新值2) [where 条件筛选];
```

解释：

- 我们通常是结合 where 条件语句来修改数据。
- **修改数据之前，要先保证表里面有数据。**如果这张表是空表，那么，执行这个命令后，等于没执行。

举例：

将表中，name 这个字段的值全部修改为 geektime：

```
update 表名 set name = 'geektime';
```

id = 3 的这条记录中，修改 name 和 age 这两个字段的值：

```
update 表名 set name = 'geekbang', age = '34' where id = 3;
```

### 4、删除数据

删除字段的操作不可逆，请谨慎操作。

语法格式：

```
delete from 表名 [where 条件];
```

解释：

- 执行删除操作之后，匹配到的**整条记录**，都会删除。
- **删除数据之前，要先保证表里面有数据。**如果这张表是空表，那么，执行这个命令后，等于没执行。

举例：

删除表中 id = 2 的记录：

```
delete from 表名 where id = 2;
```