

## 2-8数据库函数封装

2-8数据库函数封装 .....	1
1. 课堂案例 .....	2
课堂讲解 .....	2
一、上节回顾 .....	2
数据库的操作 .....	2
二、学习目标 .....	2
把数据的操作封装成函数后面方便后期使用 .....	2
三、教学过程描述 .....	2
1、回顾函数 .....	2
2、封装mysql查找 .....	2
3、封装mysql插入 .....	3
4、封装mysql更新 .....	4
5、封装mysql删除 .....	4
四、小结 .....	4
2. 课堂练习 .....	4
3. 课后练习 .....	5
4. 资料扩展 .....	5

## 1. 课堂案例

### 课堂讲解

#### 一、上节回顾

##### 数据库的操作

添加

修改

查看

删除

#### 二、学习目标

把数据的操作封装成函数后面方便后期使用

#### 三、教学过程描述

##### 1、回顾函数

```
function 命名 ()  
{  
    php执行的代码  
}
```

为什么要封装函数？

方便我们开发的时候重复使用

##### 2、封装mysql查找

###### 2-1、单条查询

1、回顾函数的概念

2、查询的mysql写法返回的数组只是一维数组而不是二维数组

3、用函数封装好

```
function dbSelectOne($table, $fields = '*', $where = [], $order = 'asc')
{
    $arr = [];
    foreach ($where as $key => $val) {
        $arr[] = "$key = '$val'";
    }
    $where = count($arr) > 0 ? join(' and ', $arr) : 1;
    $sql = "Select " . "$fields from $table where $where order by id $order";
    $rs = sqlQuery($sql);
    return mysqli_fetch_assoc($rs);
}
```

## 2-2、多条查询

- 1、回顾函数的概念
- 2、查询的mysql写法返回的二维数组
- 3、用函数封装好

```
function dbSelect($table, $fields = '*', $where = [], $offset = 0, $count = 10, $order = 'asc')
{
    $arr = [];
    foreach ($where as $key => $val) {
        $arr[] = "$key = '$val'";
    }
    $where = count($arr) > 0 ? join(' and ', $arr) : 1;
    $sql = "Select " . "$fields from $table where $where order by id $order limit $offset,$count";
    $rs = sqlQuery($sql);
    $put = [];
    while ($row = mysqli_fetch_assoc($rs)) {
        $put[] = $row;
    }
    return $put;
}
```

## 2-3、统计

- 1、回顾函数的概念
- 2、统计条数的写法
- 3、用函数封装好

```
function dbCount($table, $where = [])
{
    $arr = [];
    foreach ($where as $key => $val) {
        $arr[] = "$key = '$val'";
    }
    $where = count($arr) > 0 ? join(' and ', $arr) : 1;
    $sql = "select " . "count(*) as count from $table where $where";
    $rs = sqlQuery($sql); //装资源
    $count = mysqli_fetch_assoc($rs); //返回当前指针下的一条记录
    return $count['count'];
}
```

## 3、封装mysql插入

- 1、回顾函数的概念

## 2、mysql插入的写法

### 3、用函数封装好

```
function dbInsert($table, $data)
{
    global $con;
    $keyArr = $varArr = [];
    foreach ($data as $key => $val) {
        $keyArr[] = "$key";
        $varArr[] = "'$val'";
    }
    $sql = "insert " . "into $table (" . join(', ', $keyArr) . ") values(" . join(', ', $varArr) . ")";
    sqlQuery($sql);
    return mysqli_insert_id($con); //返回失败回到id
}
```

## 4、封装mysql更新

### 1、回顾函数的概念

### 2、mysql更新的写法

### 3、用函数封装好

```
function dbUpdate($table, $data, $where)
{
    $varArr = [];
    foreach ($data as $key => $val) {
        $varArr[] = "$key = '$val'";
    }
    $arr = [];
    foreach ($where as $key => $val) {
        $arr[] = "$key = '$val'";
    }
    $where = count($arr) > 0 ? join('and', $arr) : 1;
    $sql = "update " . " $table set " . join(', ', $varArr) . " where $where";
    return sqlQuery($sql); //返回失败回到id
}
```

## 5、封装mysql删除

### 1、回顾函数的概念

### 2、mysql删除的写法

### 3、用函数封装好

## 四、小结

如何把mysql的各种操作封装成函数

如何返回函数的结果

如何调用封装好的函数

## 2. 课堂练习

### 1、封装数据库函数

### 2、处理错误分析

### 3. 课后练习

- 1、封装sum, min, max, and, or函数
- 2、把以前的练习用PHP规范修改命名
- 3、用封装函数操作上节课创建用户表信息

### 4. 资料扩展