

# 复习

## 前端

1. 前后端分离偏多
2. 后面两个大项目(SOA,微服务) 都不会写前端
3. 工作里面如果要参与写一些前端, 再复习
  - HTML : 表格 表单
  - JS

### ■ 函数

```
function 函数名(参数列表){
```

```
}
```

1. 没有重载
2. 参数列表不需要写类型
3. 不管有没有返回值都是这个格式
4. 调用: 函数名()

### ■ 事件

在标签里面添加属性

```
<input type="button" onclick="函数名()"/>
```

派发事件

```
标签对象.onxxx=function(){
```

```
}
```

### ■ bom

```
location.href="跳转资源";
```

### ◦ JQ

#### ■ 对象转换

```
js转成jq对象  $(js对象)
```

```
jq转js对象  jq对象[0]或者jq对象.get(0)
```

#### ■ 事件

```
jq对象.事件类型(function(){});
```

eg:

```
btn.click(function(){
```

```
});
```

## ■ 操作dom

```
html() 获得标签里面的内容,把标签一起获得到
html(...) 向标签里面设置标签内容 如果有标签,解析标签
eg:
<a>
    <span>a</span>
</a>
a标签对象.html() <span>a</span>

val() 获得标签里面的value属性值
val(...)设置标签里面的value属性值

a.append("标签"); 向a标签内部插入内容
a.after(c)
a.before(c)
```

## ■ 操作标签的属性

```
attr("标签的属性")/prop("标签的属性") 获得属性值
attr("标签属性",属性值)/prop("标签属性",属性值) 设置属性值
```

## ■ ajax

```
$.get("请求的路径",{key:value},function(result){},"json");
$.post("请求的路径",{key:value},function(result){},"json");
```

- Vue 能把传智健康里面写出来 工作就不担心

# MYSQL

## 1.对数据的CRUD

### • 新增

```
insert into 表 values(值,值,值...);

insert into 表(列,列,列) values(值,值,值);
```

### • 更新

```
update 表 set 列=值, 列=值 where 条件
```

### • 删除

```
delete from 表 where 条件
```

工作里面一般很少使用物理删除,使用逻辑删除

- 1.物理删除就是delete 真删除了
- 2.逻辑删除就是更新,改状态

### • 查询

```
select ... from... where 条件
select ... from... where 条件 order by 列 desc/asc
select ... from... where 条件 order by 列 desc/asc limit a,b

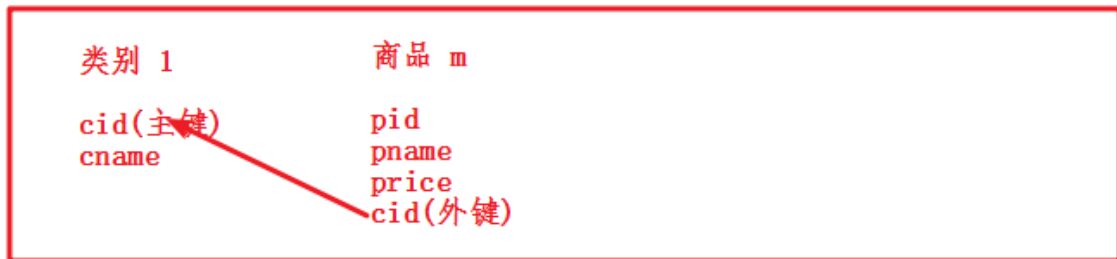
select ... from... where 条件 group by xxx 分组的目的一般是为了统计
```

## 2.多表

### 2.1多表关系

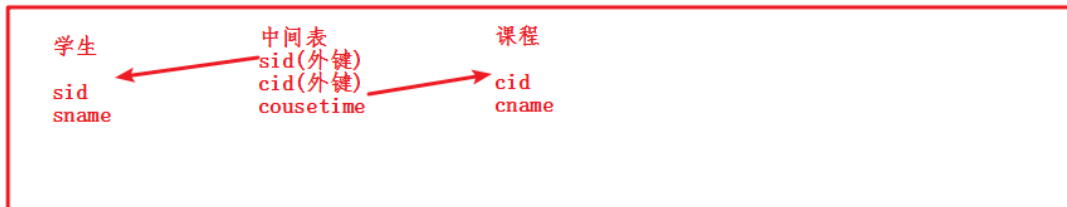
- 1对多

在多方创建一个列作为外键指向一方的主键



- 多对多

新建一张第三方表，这个表里面至少包含两个列作为外键，分别指向各自的主键



- 1对1

先当做1对多，在外键列添加唯一

直接把主键对应关联

做项目的时候: 一般不设置外键, 把外键列创建出来, 心里明白它的指向。

### 2.2多表查询

- 内连接 查询的是公共部分

```
select xxx from A ,B where A.主键=B.外键;

select xxx from A inner join B on A.主键=B.外键 where 其它条件;
```

- 外连接

- 左外 以左边表为主表, 查询左边表的所有, 通过连接条件匹配出右边表的数据, 如果满足 就显示; 没有通过null代替

```
select xxx from A left outer join B on A.主键=B.外键 where 其它条件;
```

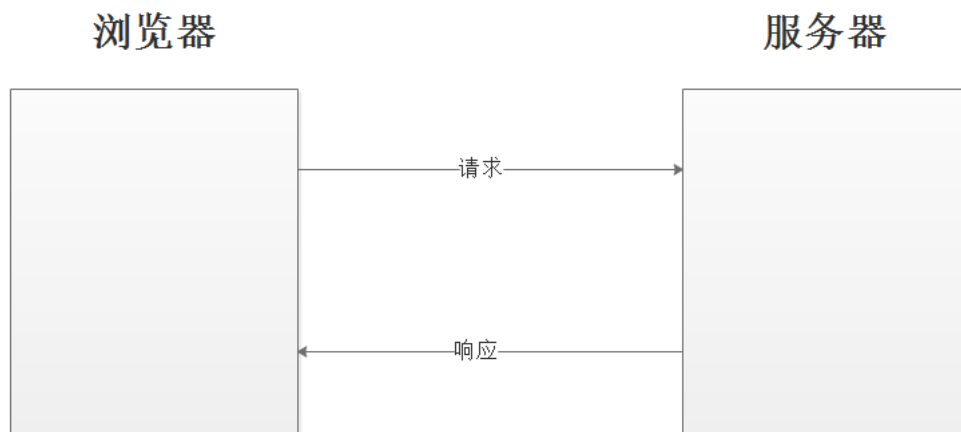
- 子查询

互联网项目里面很少会多表查询,查询单表,查询多次比较多.

传统项目多表查询会用

## web

### 0.请求响应机制



- 1次请求,1次响应
- 先请求,再响应
- 基于HTTP协议的

## 1.Servlet

### 1.1生命周期【面试】

- 服务器启动的时候, 调用init()方法进行初始化, 只调用一次
- 任何一次请求 都会调用service()方法处理这个请求
- 服务器正常关闭, 会调用destory()方法进行销毁

Servlet是单例多线程的.

## 2.Filter过滤器

- 创建一个类实现Filter
- 配置(不是SpringBoot项目在web.xml配置, 是SpringBoot不是,在代码配置)

## 3.Cookie,Session

### 3.1概念

会话技术, 保存用户的各自数据

### 3.2Cookie

浏览器端的技术, 数据保存在浏览器上面的.

先由服务器创建, 通过响应头写给浏览器

浏览器保存, 通过请求头带给服务器

### 3.3Session

session是服务器端的技术, 数据保存在服务器端. web里面通常用session保存用户的登录状态

工作实际项目里面: 很少使用session保存用户的登录状态.

后台生成token, 响应给客户端, 客户端保存, 下次携带过来.

## 4.web层

- 获得请求参数
- 调用业务
- 响应

## JSON

### 1.定义

就是一种容易生成和解析的数据格式. 独立于语音的.

### 2.JSON的格式

- 对象
  - key的类型是字符串
  - value是任意的合法数据类型
  - 多个之间用, 隔开

```
{key:value}
```

- 数组

```
[value,value]
```

- 嵌套

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
```

```
</head>
<body>

</body>
<script>
    /* var p = {"name": "zs", "age": 18};
    var pp = [{"name": "zs", "age": 18}, {"name": "ls", "age": 19}];*/
    var pppppp = {
        "p1": {"name": "zs", "age": 18}, "p23": [{"name": "ls", "age": 18},
{"name": "ww", "age": 20}],
        "p456": {"p4": {"name": "z1", "age": 21}, "p56": [{"name": "tq", "age":
22}, {"name": "wb", "age": 23}]}
    };

    //获得第四个人的名字 z1
    console.log(pppppp.p456.p4.name);

    //获得第三个人的age 20
    console.log(pppppp.p23[1].age);

    //获得第六个人的名字 wb
    console.log(pppppp.p456.p56[1].name);

</script>
</html>
```