进销存实训课程

主讲：汤燕

* 课程回顾（打印功能）
* 数据库四种连接查询方式
* 饼状图和环形图
* 什么是闭包

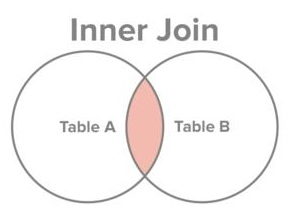
先git提交版本。

# 数据库四种连接

图解SQL连接：内连接、左外连接、右外连接、全外连接。

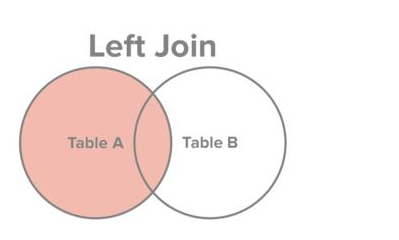
内连接（inner join）

满足条件的记录才会出现在结果集中。



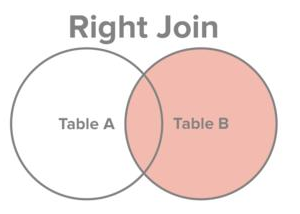
左外连接（left outer join，outer可省略）

左表全部出现在结果集中，若右表无对应记录，则相应字段为NULL



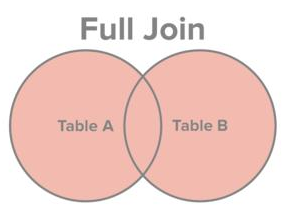
右外连接（right outer join，outer可省略）

右表全部出现在结果集中，若左表无对应记录，则相应字段为NULL



全外连接（full outer join，outer可省略）

全外连接=左外连接+右外连接



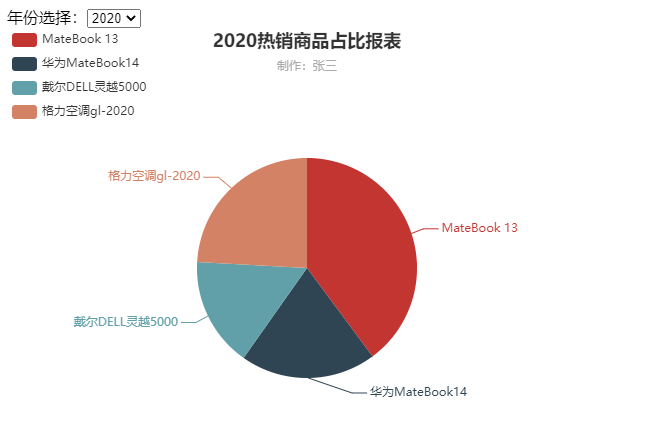
# 创建视图

从销售订单明细表中看不出每个商品销售的时间。可以从抬头表中把销售时间查询出来。

select tbsaleorderitems.\* ,tbsaleorderhead.ctime FROM tbsaleorderitems left JOIN tbsaleorderhead on tbsaleorderitems.orderid=tbsaleorderhead.orderid

|  |
| --- |
|  |

# 饼状图报表



## 前端jsp代码

|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">  <html>  <head>  <meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>  <title>Insert title here</title>  </head>  <body>  <div>  年份选择：<select id=*"yearselect"*>  <option value=*"2020"*>2020</option>  <option value=*"2021"*>2021</option>  <option value=*"2022"*>2022</option>  <option value=*"2023"*>2023</option>  </select>  <div id=*"main"* style="width: *600px*;height:*400px*;"></div>  </div>  <script src=*"../js/jquery-1.11.0.js"*></script>  <script src=*"../js/echarts.min.js"*></script>  <script>  **var** myChart = echarts.init(document.getElementById('main'));  // 指定图表的配置项和数据  **var** option = {  title: {  text: '2020热销商品占比报表',  subtext: '制作：张三',  left: 'center'  },  tooltip: {  trigger: 'item',  formatter: '{a} <br/>{b} : {c} ({d}%)'  },  legend: {  orient: 'vertical',  left: 'left',  data: []  },  series: [  {  name: '占比',  type: 'pie',  radius: '55%',  center: ['50%', '60%'],  data: [],  emphasis: {  itemStyle: {  shadowBlur: 10,  shadowOffsetX: 0,  shadowColor: 'rgba(0, 0, 0, 0.5)'  }  }  }  ]  };  myChart.setOption(option);  **function** getajaxdata(objyear) {  $.ajax({  type: "post",  url: "../ajax",  data: { rnum:12, cyear: objyear },  timeout: 5000,  dataType: "json",  async: **true**,//默认设置为true，所有请求均为异步请求  //cache：true,//默认为true（当dataType为script时，默认为false）设置为false将不会从浏览器缓存中加载请求信息。  success: **function** (data) {  **var** optionhasvalue = {  title: {  text: '2020热销商品占比报表',  subtext: '制作：张三',  left: 'center'  },  tooltip: {  trigger: 'item',  formatter: '{a} <br/>{b} : {c} ({d}%)'  },  legend: {  orient: 'vertical',  left: 'left',  data:data.data1  },  series: [  {  name: '每月占比',  type: 'pie',  radius: '55%',  center: ['50%', '60%'],  data:data.data2,  emphasis: {  itemStyle: {  shadowBlur: 10,  shadowOffsetX: 0,  shadowColor: 'rgba(0, 0, 0, 0.5)'  }  }  }  ]  };  myChart.setOption(optionhasvalue);  }  });  }  getajaxdata(2020);  $("#yearselect").change(**function** () {  **var** v = $(**this**).val();  getajaxdata(v);  });  </script>  </body>  </html> |

## 后端ajax响应函数

|  |
| --- |
| **protected** **void** getSaleBili(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  String StrSql="select proname,sum(price) as sumprice,sum(procount) as sumcount from v\_saleitems where DATE\_FORMAT(ctime,'%Y') =? GROUP BY proname";  String cyear=request.getParameter("cyear");  List<Object> params= **new** ArrayList<Object>();  params.add(cyear);  DBHelper db=**new** DBHelper();  List<Map<String, Object>> reslist = **null**;  String html="";  //{"data1":["MateBook 13","戴尔DELL灵越5000"],"data2":[{"MateBook 13":100},{"戴尔DELL灵越5000":200}]}  **try** {  reslist=db.executeQuery(StrSql, params);  String html\_1="[";  String html\_2="[";  **int** i=1;  **for** (Map<String, Object> m : reslist) {  **if**(i==reslist.size())  {  html\_1+="\""+m.get("proname")+"\"";  html\_2+="{\"name\":\""+m.get("proname")+"\",\"value\":"+m.get("sumprice")+"}";  }  **else**  {  html\_1+="\""+m.get("proname")+"\",";  html\_2+="{\"name\":\""+m.get("proname")+"\",\"value\":"+m.get("sumprice")+"},";  }  i++;  }  html\_1+="]";  html\_2+="]";  html="{\"data1\":"+html\_1+",\"data2\":"+html\_2+"}";  } **catch** (SQLException e) {  e.printStackTrace();  }  response.setCharacterEncoding("utf-8");  response.setContentType("text/json;charset=utf-8");  response.getWriter().write(html);    } |

## 完成作业

比如：退换货功能做了，我们就可以做一个在一段时间内退换货比例的报表。

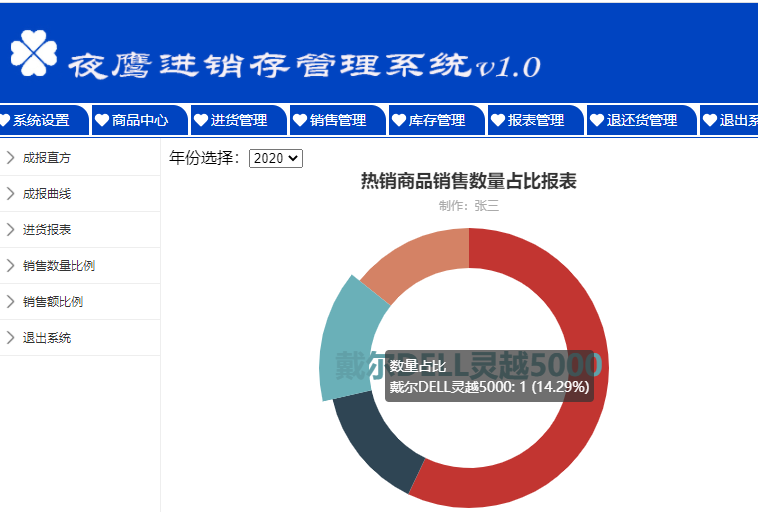
以饼状图来展示。

# 环形图报表

## 完成作业

比如：退换货功能做了，我们就可以做一个在一段时间内退换货比例的报表。以环形图来展示。

## 前端代码



|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">  <html>  <head>  <meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>  <title>Insert title here</title>  </head>  <body>  <div>  年份选择：<select id=*"yearselect"*>  <option value=*"2020"*>2020</option>  <option value=*"2021"*>2021</option>  <option value=*"2022"*>2022</option>  <option value=*"2023"*>2023</option>  </select>  <div id=*"main"* style="width: *600px*;height:*400px*;"></div>  </div>  <script src=*"../js/jquery-1.11.0.js"*></script>  <script src=*"../js/echarts.min.js"*></script>  <script>  **var** myChart = echarts.init(document.getElementById('main'));  // 指定图表的配置项和数据  // 指定图表的配置项和数据  **var** option = {  title: {  text: '热销商品销售数量占比报表',  subtext: '制作：张三',  left: 'center'  },  tooltip: {  trigger: 'item',  formatter: '{a} <br/>{b}: {c} ({d}%)'  },  legend: {  orient: 'vertical',  left:20,  data: ['7月', '8月', '9月']  },  series: [  {  name: '数量占比',  type: 'pie',  radius: ['50%', '70%'],  center: ['50%', '50%'],  avoidLabelOverlap: **false**,  label: {  show: **false**,  position: 'center'  },  emphasis: {  label: {  show: **true**,  fontSize: '30',  fontWeight: 'bold'  }  },  labelLine: {  show: **false**  },  data: [    ]    }  ]  };  myChart.setOption(option);  **function** getajaxdata(objyear) {  $.ajax({  type: "post",  url: "../ajax",  data: { rnum:13, cyear: objyear },  timeout: 5000,  dataType: "json",  async: **true**,//默认设置为true，所有请求均为异步请求  //cache：true,//默认为true（当dataType为script时，默认为false）设置为false将不会从浏览器缓存中加载请求信息。  success: **function** (data) {  **var** optionhasvalue = {  title: {  text: '热销商品销售数量占比报表',  subtext: '制作：张三',  left: 'center'  },  tooltip: {  trigger: 'item',  formatter: '{a} <br/>{b}: {c} ({d}%)'  },  legend: {  orient: 'vertical',  left:20,  data: ['7月', '8月', '9月']  },  series: [  {  name: '数量占比',  type: 'pie',  radius: ['50%', '70%'],  center: ['50%', '50%'],  avoidLabelOverlap: **false**,  label: {  show: **false**,  position: 'center'  },  emphasis: {  label: {  show: **true**,  fontSize: '30',  fontWeight: 'bold'  }  },  labelLine: {  show: **false**  },  data: data.data2  }  ]  };  myChart.setOption(optionhasvalue);  }  });  }  getajaxdata(2020);  $("#yearselect").change(**function** () {  **var** v = $(**this**).val();  getajaxdata(v);  });  </script>  </body>  </html> |

## 后端ajax代码

|  |
| --- |
| **protected** **void** getSaleBiliCount(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  String StrSql="select proname,sum(price) as sumprice,sum(procount) as sumcount from v\_saleitems where DATE\_FORMAT(ctime,'%Y') =? GROUP BY proname";  String cyear=request.getParameter("cyear");  List<Object> params= **new** ArrayList<Object>();  params.add(cyear);  DBHelper db=**new** DBHelper();  List<Map<String, Object>> reslist = **null**;  String html="";  //{"data1":["MateBook 13","戴尔DELL灵越5000"],"data2":[{"MateBook 13":100},{"戴尔DELL灵越5000":200}]}  **try** {  reslist=db.executeQuery(StrSql, params);  String html\_1="[";  String html\_2="[";  **int** i=1;  **for** (Map<String, Object> m : reslist) {  **if**(i==reslist.size())  {  html\_1+="\""+m.get("proname")+"\"";  html\_2+="{\"name\":\""+m.get("proname")+"\",\"value\":"+m.get("sumcount")+"}";  }  **else**  {  html\_1+="\""+m.get("proname")+"\",";  html\_2+="{\"name\":\""+m.get("proname")+"\",\"value\":"+m.get("sumcount")+"},";  }  i++;  }  html\_1+="]";  html\_2+="]";  html="{\"data1\":"+html\_1+",\"data2\":"+html\_2+"}";  } **catch** (SQLException e) {  e.printStackTrace();  }  response.setCharacterEncoding("utf-8");  response.setContentType("text/json;charset=utf-8");  response.getWriter().write(html);    } |

## 增加配置

|  |
| --- |
| **case** "12":getSaleBili(request,response);**break**;  **case** "13":getSaleBiliCount(request,response);**break**; |

# 什么是闭包

闭包（closure）是javascript的一大难点，也是它的特色。很多高级应用都要依靠闭包来实现。

**1、变量作用域**

要理解闭包，首先要理解javascript的特殊的变量作用域。

变量的作用域无非就两种：全局变量和局部变量。

javascript语言的特别之处就在于：函数内部可以直接读取全局变量，但是在函数外部无法读取函数内部的局部变量。

注意点：在函数内部声明变量的时候，一定要使用var命令。如果不用的话，你实际上声明的是一个全局变量！

**2、如何从外部读取函数内部的局部变量？**

出于种种原因，我们有时候需要获取到函数内部的局部变量。但是，上面已经说过了，正常情况下，这是办不到的！只有通过变通的方法才能实现。

那就是在函数内部，再定义一个函数。

function f1(){

　　　　var n=999;

　　　　function f2(){  
　　　　　　alert(n); // 999  
　　　　}

　　}

在上面的代码中，函数f2就被包括在函数f1内部，这时f1内部的所有局部变量，对f2都是可见的。但是反过来就不行，f2内部的局部变量，对f1就是不可见的。

这就是Javascript语言特有的"链式作用域"结构（chain scope），

子对象会一级一级地向上寻找所有父对象的变量。所以，父对象的所有变量，对子对象都是可见的，反之则不成立。

既然f2可以读取f1中的局部变量，那么只要把f2作为返回值，我们不就可以在f1外部读取它的内部变量了吗！

**3、闭包的概念**

上面代码中的f2函数，就是闭包。

各种专业文献的闭包定义都非常抽象，我的理解是: 闭包就是能够读取其他函数内部变量的函数。

由于在javascript中，只有函数内部的子函数才能读取局部变量，所以说，闭包可以简单理解成“定义在一个函数内部的函数“。

所以，在本质上，闭包是将函数内部和函数外部连接起来的桥梁。

**4、闭包的用途**

闭包可以用在许多地方。它的最大用处有两个，一个是前面提到的可以读取函数内部的变量，另一个就是让这些变量的值始终保持在内存中，不会在f1调用后被自动清除。

为什么会这样呢？原因就在于f1是f2的父函数，而f2被赋给了一个全局变量，这导致f2始终在内存中，而f2的存在依赖于f1，因此f1也始终在内存中，不会在调用结束后，被垃圾回收机制（garbage collection）回收。

这段代码中另一个值得注意的地方，就是"nAdd=function(){n+=1}"这一行，首先在nAdd前面没有使用var关键字，因此nAdd是一个全局变量，而不是局部变量。其次，nAdd的值是一个匿名函数（anonymous function），而这个匿名函数本身也是一个闭包，所以nAdd相当于是一个setter，可以在函数外部对函数内部的局部变量进行操作。

**5、使用闭包的注意点**

（1）由于闭包会使得函数中的变量都被保存在内存中，内存消耗很大，所以不能滥用闭包，否则会造成网页的性能问题，在IE中可能导致内存泄露。解决方法是，在退出函数之前，将不使用的局部变量全部删除。

（2）闭包会在父函数外部，改变父函数内部变量的值。所以，如果你把父函数当作对象（object）使用，把闭包当作它的公用方法（Public Method），把内部变量当作它的私有属性（private value），这时一定要小心，不要随便改变父函数内部变量的值。

# 立即执行函数

|  |
| --- |
|  |

延迟执行函数

定时器函数

# 登录通知功能（略）