国际物流云商系统第八天

一．回顾

1.什么是购销合同
2.分析出购销合同的表结构关系
3.设计表的三范式及反三范式
4.添加冗余字段，使得查询速度更快
5.分散计算原理
购销合同总金额的计算
6.实现购销合同的相关 CRUD 操作
7.实现货物的 CRUD 操作
8.实现附件的 CRUD 操作

二．细粒度权限控制

就是在粗粒度权限控制的基础上，加入对于不同用户显示的数据也不一样的控制。实质就是数据的控制。

1.数据库表的准备
createBy 字段：代表创建者的 id
createDept 字段:代表创建者所在部门 id
createTime 字段
updateBy 字段
updateTime 字段

补充 ContractAction 中的方法

```
/**
 * 实现新增
 */
@Action("contractAction_insert")
public String insert() throws Exception {
    //获得当前的登录用户信息
    User user = super.getCurUser();
    //设置细粒度权限的一些属性
    model.setCreateBy(user.getId());
    model.setCreateDept(user.getDept().getId());

    // 调用业务方法，实现保存
    contractService.saveOrUpdate(model);
    return "alist";
}
```

准备用户

OOP 员工 都属于船运部

AOP 员工 都属于船运部

SOA 部门经理 都属于船运部

给用户授予角色

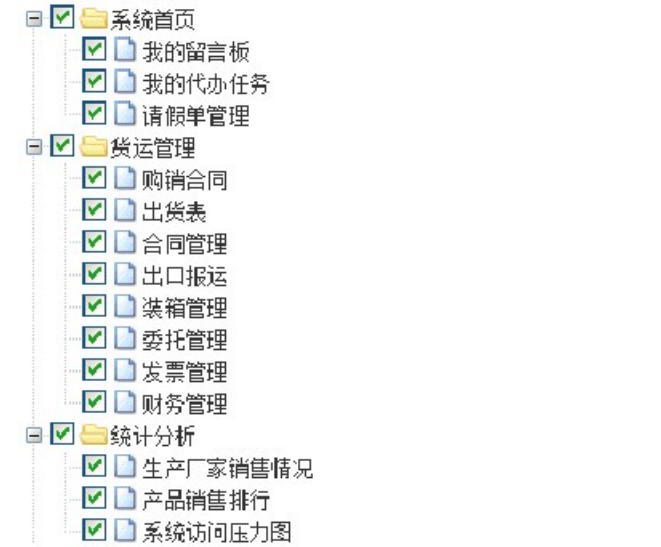
OOP 船运专员

AOP 船运专员

SOA 船运经理

给角色分配模块

船运经理



准备三条记录

7	OOP	OOP	0 / 0	OOP	OOP	OOP	2016-01-27	2015-12-28	2016-01-04	FOB	0.0	草稿	[货物]
8	AOP	AOP	0 / 0	AOP	AOP	AOP	2016-01-12	2016-01-25	2015-12-29	T/T	0.0	草稿	[货物]
9	SOA	SOA	0 / 0	SOA	SOA	SOA	2016-01-06	2016-01-26	2016-01-06	L/C	0.0	草稿	[货物]

从 jUserCreate.jsp 页面中，可以得出用户主要通过等级来决定可以看到什么数据

```
<input type="radio" name="userinfo.degree" value="0" class="input"/>超级管理员
                <input type="radio" name="userinfo.degree" value="1"
class="input"/>跨部门跨人员
                <input type="radio" name="userinfo.degree" value="2"
class="input"/>管理所有下属部门和人员
                <input type="radio" name="userinfo.degree" value="3"
class="input"/>管理本部门
                <input type="radio" name="userinfo.degree" value="4"
class="input"/>普通员工
```

修改 ContractAction.java 类中的 list()方法

```
public String list() throws Exception {
    final User user = super.getCurUser();
    final int degree = user.getUserinfo().getDegree();//等级
    //实现细粒度查询
    Specification<Contract> spec = new Specification<Contract>() {
        public Predicate toPredicate(Root<Contract> root, CriteriaQuery<?> query, CriteriaBuilder cb) {
            Predicate p = null;
            if(degree==4){
                //员工
                p = cb.equal(root.get("createBy").as(String.class), user.getId());
            }else if(degree == 3){
                //部门经理
                p = cb.equal(root.get("createDept").as(String.class), user.getDept().getId());
            }else if(degree==2){
                //管理本部门及下属部门
            }else if(degree==1){
                //管理跨部门跨人员
            }else if(degree==0){
                //总经理
                //什么也不写
            }
            return p;
        }
    };
    // 设置Pageable的取值
    Pageable pageable = new PageRequest(page.getPageNo() - 1, page.getPageSize());
    org.springframework.data.domain.Page<Contract> spage = contractService.findPage(spec, pageable);
```

改造后，通过 cgx 账号，AOP 账号，OOP 账号进行数据的测试，可以发现不同身份的用户，虽然都具有购销合同的查看功能，但是他们因为身份的等级信息的不同，最终导致看到的数据也是不一样的。

三．存在购销合同下的 **bug**

对于购销合同的货物数/附件数加载存在的问题？
存在的问题是：当加载一条购销合同记录时，会通过关联级别的数据加载，来加载购销合同下的货物列表，同时还要加载货物列表下的附件列表，所以造成加载速度变慢的问题。

解决思路：
在购销合同的表中加入两个冗余字段，将来在添加购销合同下的货物时就同时更新购销合同

表中的货物数量，在添加附件时，就更新购销合同表中的附件数量。

又产生了新的问题？
因为冗余字段的添加，导致程序员的开发和维护工作量添加。目的是保证数据的一致性。

作业：希望改造上面的问题，使用的是分散计算思想，在 Contract_c 表中添加两个冗余字段

【面试】
当系统开发完成后，项目刚上线一切正常，但随着数据量的增加，系统运行半年后，后期加载数据页面速度很慢，用户无法接受，请问你有什么好的优化方案。

四. POI 报表

1.POI 介绍

Apache 专门操作 excel api

2.主流操作 excel api

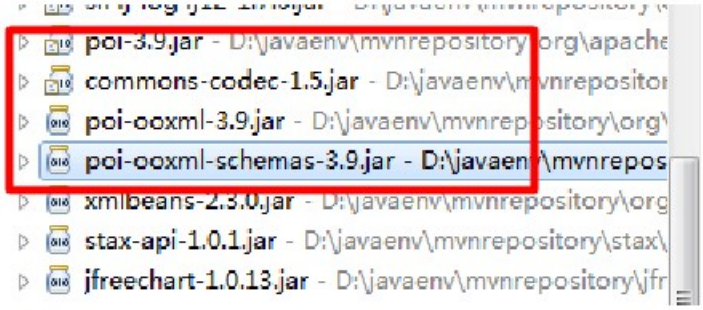
1） Jxl　　只能操作 excel 2003
2） Poi　　可以操作整个 office（excel、doc(vb 宏)、ppt、visio）；包括所有的 excel 版本

	Excel2003　xls	Excel2007 及以上版本 xlsx
基础	OLE 技术实现，控件	OOXML 技术
问题：支持数据量有限（二进制）	大数据量时，查询数据效率低	在 2007 版本，微软对 office 进行重构，基于 xml，查询效率特别高
单 sheet	行：65536；列：256；	行：1048576；列：16384

五．应用场景

应用于将数据写入 excel，用户可以共享数据。作为备份数据（不包括大字段）、还原数据。

1.导入 POI 的相关 jar 包



不包括 commons-codec-1.5.jar

Maven 项目：
Pom.xml 文件中的 xml 配置：

```xml
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi</artifactId>
    <version>3.11</version>
</dependency>
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-ooxml</artifactId>
    <version>3.11</version>
</dependency>
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-ooxml-schemas</artifactId>
    <version>3.11</version>
</dependency>
```

2.需求：需要 8 步

a) 创建一个工作簿 workbook

b) 创建一个工作表 sheet

c) 创建一个行对象 row (下标起始值为 0)

d) 创建一个单元格对象 cell (下标起始值为 0)

e) 给单元格设置内容

f) 设置单元格的样式，设置字体和字体的大小

g) 保存，关闭流对象

h) 下载

3.实现代码

```
@Test
    public void testPoi() throws IOException{
        //1.创建一个工作簿
        Workbook wb = new HSSFWorkbook();//作用于excel2003 版本
        //2.创建工作表
        Sheet sheet = wb.createSheet();
        //3.创建行对象
        Row row = sheet.createRow(3);
        //4.创建单元格
        Cell cell = row.createCell(3);
        //5.设置单元格的内容
        cell.setCellValue("传智播客  宋江");
        //6.设置单元格的样式
        CellStyle cellStyle = wb.createCellStyle();
        Font font = wb.createFont();

        font.setFontHeightInPoints((short)28);//字体大小
        font.setFontName("华文行楷");//字体名称

        cellStyle.setFont(font);//设置单元格样式的字体

        cell.setCellStyle(cellStyle);//将单元格样式作用于单元格
        //7.保存，关闭流
        OutputStream os = new FileOutputStream("E:\\img\\出货表.xls");
        wb.write(os);
        os.close();
        //8.下载（不能测试）
    }
```

六．出货表的打印

1.是按船期进行统计的。

向客户咨询得到的。


```
<td class="tableContent">
    <input type="text" style="width:90px;" name="inputDate"
     value="2015-07"
    onclick="WdatePicker({el:this,isShowOthers:true,dateFmt:'yyyy-MM'});"/>
</td>
```

2.数据来源哪些表？

通过分析得出： 来自于购销合同表，货物表

3.思考怎么查？

Hql 语句只要查询货物就可以了
因为货物中有一个关联的对象（contract 属性），它的 HQL 语句如下：

```
from ContractProduct cp where cp.contract.shipTime = ?
```

4.出货表页面的进入

只要在该页面提供一个日历选择框就可以选择船期了，再使用 POI 实现数据打印

```
/**
 * 进入出货表的打印页面
 */
@Action(value = "outProductAction_toedit", results = {
        @Result(name = "toedit", location = "/WEB-INF/pages/cargo/outproduct/jOutProduct.jsp") })
public String toedit() throws Exception {

    return "toedit";
}
```

5.实现出货表的打印

出货表的打印思路分以下步骤来进行。

1. 创建 POI 报表相关对象
2. 实现大标题的制作
3. 实现小标题的制作
4. 实现 POI 报表的下载输出

创建 POI 报表的相关对象

```
/**
 * 实现POI打印
 */
@Action(value="outProductAction_print")
public String print() throws Exception {
    //1.创建工作簿
    Workbook wb = new HSSFWorkbook();
    //2.创建工作表Sheet
    Sheet sheet = wb.createSheet();
    //设置一些通用变量
    Row nRow = null;
    Cell  nCell = null;
    int cellNo=1,rowNo=0;

    //设置列宽信息
    sheet.setColumnWidth(0, 1);
    sheet.setColumnWidth(1,26);
    sheet.setColumnWidth(2, 11);
    sheet.setColumnWidth(3, 29);
    sheet.setColumnWidth(4, 12);
    sheet.setColumnWidth(5, 15);
    sheet.setColumnWidth(6, 10);
    sheet.setColumnWidth(7, 10);
    sheet.setColumnWidth(8, 8);
```

大标题制作

```
//==============================制作大标题
nRow = sheet.createRow(rowNo++);//创建行对象
nRow.setHeightInPoints(36);//设置行高
nCell = nRow.createCell(cellNo);//创建单元格对象
nCell.setCellValue(inputDate.replace("-0", "-").replace("-", "年")+"月份出货表");
//实现第一行的单元格合并
sheet.addMergedRegion(new CellRangeAddress(0,0,1,8));//横向合并单元格
//设置样式
nCell.setCellStyle(this.bigTitle(wb));
```

小标题制作

```
//============================制作小标题
nRow = sheet.createRow(rowNo++);
nRow.setHeightInPoints(26.25f);//行高
String titles[] ={"客户","订单号","货号","数里","工厂","工厂交期","船期","贸易条款"};
for (String title : titles) {
    nCell = nRow.createCell(cellNo++);
    nCell.setCellValue(title);
    nCell.setCellStyle(this.title(wb));
}
```

内容制作

```
//============================制作内容
//查询出符合条件的出货记录
List<ContractProduct> cpList = contractProductService.findByShipTime(inputDate);
for (ContractProduct cp : cpList) {
    nRow = sheet.createRow(rowNo++);//创建一个行对象
    nRow.setHeightInPoints(24f);

    cellNo=1;
    //得到单元格对象  客户
    nCell = nRow.createCell(cellNo++);//创建单元格对象
    nCell.setCellValue(cp.getContract().getCustomName());//设置单元格内容
    nCell.setCellStyle(this.text(wb));

    //订单号
    nCell = nRow.createCell(cellNo++);//创建单元格对象
    nCell.setCellValue(cp.getContract().getContractNo());//设置单元格内容
    nCell.setCellStyle(this.text(wb));

    //货号
    nCell = nRow.createCell(cellNo++);//创建单元格对象
    nCell.setCellValue(cp.getProductNo());//设置单元格内容
    nCell.setCellStyle(this.text(wb));

    //数里
    nCell = nRow.createCell(cellNo++);//创建单元格对象
    nCell.setCellValue(cp.getCnumber());//设置单元格内容
    nCell.setCellStyle(this.text(wb));

   //工厂
   nCell = nRow.createCell(cellNo++);//创建单元格对象
   nCell.setCellValue(cp.getFactoryName());//设置单元格内容
   nCell.setCellStyle(this.text(wb));

   //工厂交期
   nCell = nRow.createCell(cellNo++);//创建单元格对象
   nCell.setCellValue(UtilFuns.dateTimeFormat(cp.getContract().getDeliveryPeriod()));//设置单元格内容
   nCell.setCellStyle(this.text(wb));

   //  船期
   nCell = nRow.createCell(cellNo++);//创建单元格对象
   nCell.setCellValue(UtilFuns.dateTimeFormat(cp.getContract().getShipTime()));//设置单元格内容
   nCell.setCellStyle(this.text(wb));

   //贸易条款
   nCell = nRow.createCell(cellNo++);//创建单元格对象
   nCell.setCellValue(cp.getContract().getTradeTerms());//设置单元格内容
   nCell.setCellStyle(this.text(wb));
}
```

下载 Excel 表格

```
//============================下载Excel文件
DownloadUtil downLoadUtil = new DownloadUtil();

ByteArrayOutputStream baos = new ByteArrayOutputStream();
wb.write(baos);
HttpServletResponse response = ServletActionContext.getResponse();
downLoadUtil.download(baos, response, "出货表.xls");
return NONE;
```

七．正确的方向

1.模板打印

制作模板文件（模板文件路径：/make/xlsprint/tOUTPRODUCT.xls）

2.模板打印的步骤

1.导入（加载）模板文件，从而得到一个工作簿
2.读取工作表
3 读取行
4.读取单元格
5. 读取单元格样式
6. 设置单元格内容
7.其它单元格就可以使用读到的样式了

八．作业

1.读程

2.往后编写：出口报运模块