

# 怎样在应用中集成自助报表功能

润乾软件出品





**应用的集成结构**

**文件数据源的自助报表**

**数据库数据源的自助报表**

**字段标题与显示值**

**权限控制**

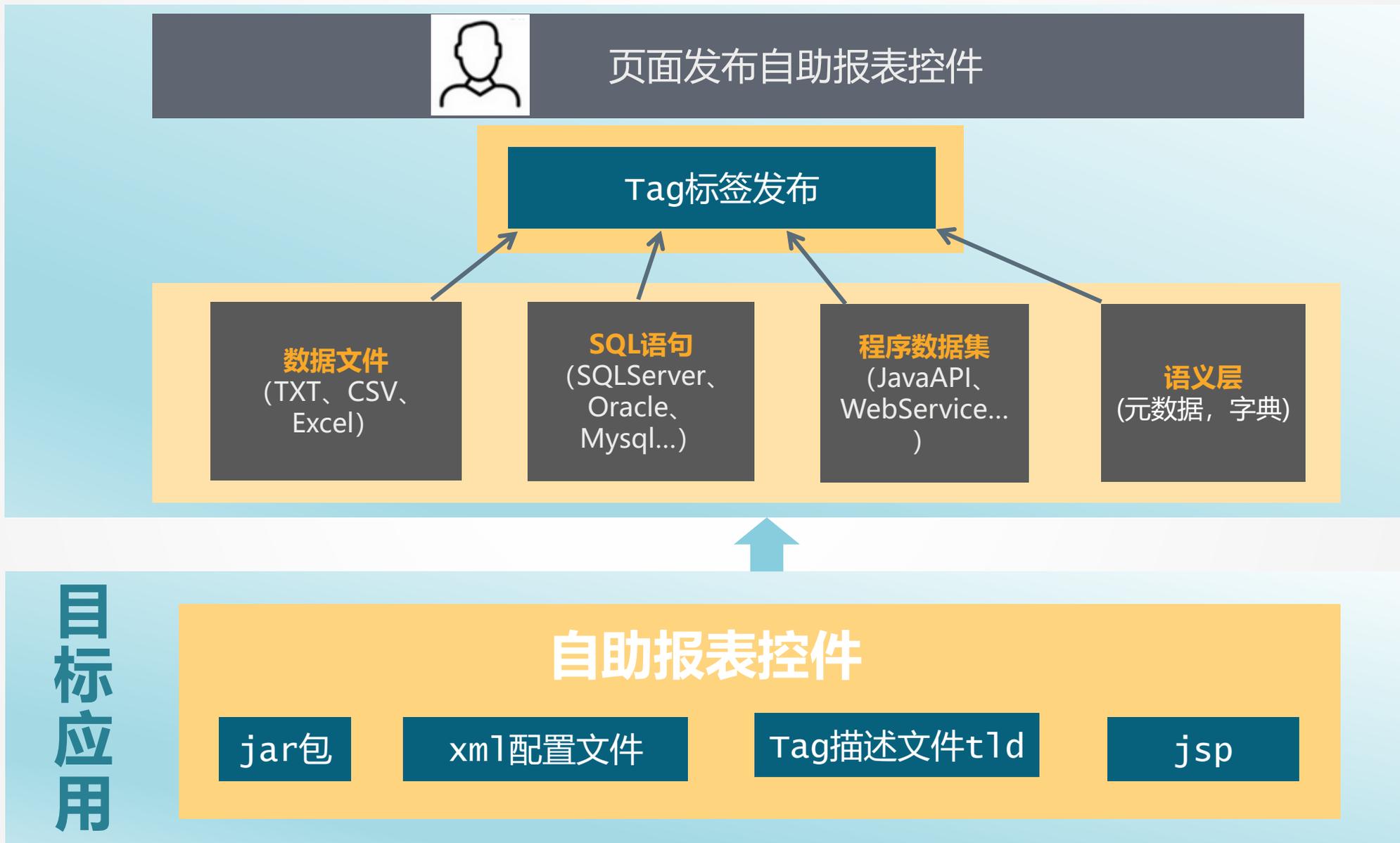


# - 应用的集成结构 -

PART ONE



## 应用的集成结构





应用层



多维分析提供了标准的应用接口，可将自助报表无缝集成到应用中。集成时只需按照标准的 java 开发三件套即可，分别为 jar 包、xml 配置文件、以及 jsp 和 tag 描述文件 tld。



## 自助报表控件目录结构说明

raqsoft	必须的目录，目录名不可以调整	
jre-6u24-windows-i586.exe	jre的安装包，客户端打印时如果检测到没有安装java环境，会自动下载此包，适合windows操作系统	
raqsoftReportApplet.jar	打印的applet包	
raqsoft.cer	applet打印签名证书	
WEB-INF	必须的目录，放置应用配置文件和jar	
	web.xml	配置servlet
	classes	类路径
	raqsoftConfig.xml	报表的核心配置文件
	raqsoftInput.tld	填报报表tag标签
	raqsoftReport.tld	报表tag标签
	raqsoftAnalyse.tld	分析控件的tag标签
	raqsoftQuery.tld	查询控件的tag标签
	raqsoftReportLog.properties	配置润乾报表的日志文件路径、日志输出信息级别
	lib	WEB应用所需引用的jar包

所有文件中只有web.xml文件不能简单复制，需要将润乾报表V2018的web.xml文件的内容复制到目标应用的web.xml文件中。其内容在web.xml中的顺序，要遵从相关应用服务器的规定。如果不是和已有应用集成，而是新建应用，那么直接复制web.xml也无妨。

目录存放路径：**【润乾报表V2018安装根目录】** \report\web\webapps\demo，使用自助报表功能需将上述内容全部集成到WEB应用中。《程序员参考》**WEB应用配置**小节中介绍了目录中的所有文件目录结构及用途，如有需要可做进一步查阅。



## 配置授权文件和自助报表需使用的数据源

打开[WEB应用根目录]/WEB-INF/raqsoftConfig.xml:

```
<?xml version="1.0" encoding="UTF-8"?><Config Version="1"><Runtime >
//配置应用需要用的数据源
<DBList>
<!--数据库类型编号，编号依次代表的数据库类型为：[UNKNOWN:0; ORACLE:1; SQLSVR:2; SYBASE:3;
SQLANY:4; INFMIX:5; FOXPRO:6; ACCESS:7; FOXBAS:8; DB2:9; MYSQL:10; KINGBASE:11; DERBY:12;
HSQL:13; TERADATA:14; POSTGRES:15; DATALOGIC:16; IMPALA:17; ESSBASE:101;] -->
<DB name="dqldemo">
  <property name="url" value="jdbc:hsqldb:hsqldb://127.0.0.1/reportdemo" />
  <property name="driver" value="org.hsqldb.jdbcDriver" />
  <property name="type" value="13" />
  <property name="user" value="sa" />
  <property name="password" />
  ....
</DB>... .. </DBList> ... ..
//集算器选项配置，安装包缺省提供了内置集算器授权，此步可略过
<Esproc><license/> ... .. </Esproc>
//报表选项配置
<Report><license> D:/report.lic</license>... ..</Report>... ..
</Config>
```

**注意：**连接数据源时需将对应的数据库驱动包放在能够加载到的类路径下



# -文件数据源的自助报表-

PART TWO



准备数据 (Excel, txt, csv)



使用Tag标签发布



页面展现



自定义页面菜单





## 2. Tag标签发布

在jsp中定义Tag标签的属性及值，用Tag标签完成多维分析的解析及运算，按用户配置的属性要求将HTML文件流输出到客户端。

Test. jsp:

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
```

```
<%@ taglib uri="/WEB-INF/raqsoftAnalyse.tld" prefix="raqsoft" %>
```

```
<raqsoft:analysev2
```

```
    dfxFile="WEB-INF/files/dfx/official/readFile.dfx" //指定用来读取数据文件的readFile.dfx
```

```
    dfxParams="f=WEB-INF/data.xls" //readFile.dfx中定义了名为f的参数，用来接收xls的文件路径，因此这里将xls的文件路径作为参数值传递给f
```

```
/>
```

	A	B
1	if (pos(lower(f), ".csv")==len(f)-3)	>r=file(f).import@t(, ",")
2	else if (pos(lower(f), ".txt")==len(f)-3)	>r=file(f).import@t()
3	else if (pos(lower(f), ".xlsx")==len(f)-4)	>r=file(f).xlsimport@t()
4	else if (pos(lower(f), ".xls")==len(f)-3)	>r=file(f).xlsimport@t()
5	else if (pos(lower(f), ".ctb")==len(f)-3)	>r=file(f).create().cursor()
6	else if (pos(lower(f), ".lbt")==len(f)-3)	>r=file(f).cursor@b()
7	return r	

**注意：**readFile.dfx默认存放于【润乾报表V2018安装根目录】/report/web/webapps/demo/WEB-INF/files/dfx/official目录下



### 3. 页面展现

数据集创建完成，可随意在复杂条件中拖拽表头和指标做自助查询。





## 4. 自定义页面菜单

分析界面上方默认显示toolbar工具条，可通过在<script/>中添加JS API属性去掉toolbar工具条。

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib uri="/WEB-INF/raqsoftAnalyse.tld" prefix="raqsoft" %>
<raqsoft:analysev2
    dfxFile="WEB-INF/files/dfx/official/readFile.dfx"
    dfxParams="f=WEB-INF/files/dfx/official/data.xls" />
<script>
    guideConf.showToolBar = 'no'; //是否显示上部toolbar工具条。
</script>
```



**注意：**更多JS API的属性介绍可参考在线帮助文档《分析教程》[JS API控制初始化界面](#)小节。



# -数据库数据源的自助报表-

PART THREE



1

定义物理数据源

2

Tag标签发布SQL数据集

3

二进制缓存文件提升性能

4

非缓存SQL数据集



# 1. 定义物理数据源

在 [web 应用根目录]/WEB-INF/raqsoftConfig.xml 中定义SQL数据集要使用的物理数据源

```
<DBList>

<DB name="dqldemo">

<property name="url" value="jdbc:hsqldb:hsq://127.0.0.1/demo" ></property>

<property name="driver" value="org.hsqldb.jdbcDriver" ></property>

<property name="type" value="13" ></property>

<property name="user" value="sa" ></property>

<property name="password" ></property>

... ..

</DB>

... ..

</DBList>
```

**注意：**需将数据源对应的数据库驱动包放在能够加载到的类路径下



## 2. Tag标签发布SQL数据集

在jsp中定义Tag标签的属性及值，用Tag标签完成多维分析的解析及运算，按用户配置的属性要求将HTML文件流输出到客户端。

Test.jsp:

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
```

```
<%@ taglib uri="/WEB-INF/raqsoftAnalyse.tld" prefix="raqsoft" %>
```

```
<raqsoft:analysev2
```

```
    dataSource="dqldemo" //指定SQL需要使用的数据源
```

```
    ql="select * from employee" //SQL语句
```

```
/>
```

使用Tag 标签添加数据集是直接为自助报表页面设置初始数据集，程序会自动生成临时文件，并将计算结果存入。  
缓存文件类型支持TXT和二进制两种格式。  
程序默认保存为TXT类型。

EID	NAME	STATE	SALARY
1	Rebecca	California	7000
2	Ashley	New York	11000
3	Rachel	New Mexico	9000
4	Emily	Texas	7000
5	Ashley	Texas	16000
6	Matthew	California	11000
7	Alexis	Illinois	9000
8	Megan	California	11000
9	Victoria	Texas	2000
10	Ryan	Pennsylvania	13000
11	Jacob	Texas	12000
12	Jessica	New York	7000
13	Daniel	Florida	10000
14	Alyssa	Florida	4000



## 3. 二进制缓存文件

Test.jsp中添加JS API 脚本：

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib uri="/WEB-INF/raqsoftAnalyse.tld" prefix="raqsoft" %>
<raqsoft:analysev2
    dataSource="dqldemo" //指定SQL需要使用的数据源
    ql="select * from employee" //SQL语句
/>
<script>
    guideConf.dataFileType = 'binary' ; //binary是二进制文件，它以游标方式读取，可支持超出内存的
数据集，计算性能也更好。此功能需要大数据版的支持。该属性的默认为text文本文件类型。
</script>
```

自助报表控件提供了JS API控制初始化界面的方法以及通过js脚本对初始化后的界面进行修改的方法。如需进一步学习，可查阅《[分析教程](#)》。



## 缓存文件的命名及生命周期说明：

- 1、默认缓存文件命名规则以时间毫秒数组合而成；
- 2、缓存文件的父目录路径中包含” temp”字样时，如：/Raqsofttemp/.txt 或 /temp/.txt，程序会在 Session 超时时自动清理该缓存文件；
- 3、生成缓存文件时，如果发现缓存文件不存在，则会自动根据数据集的配置重新生成缓存文件；
- 4、如果缓存文件路径中没有 temp 字样，则不会对缓存文件做任何删除处理，由客户的系统自己管理这些缓存文件的生命周期。



## 4. 非缓存SQL数据集

将数据集计算结果缓存入文件的方式对于数据量不大的情况可以处理，但有时却希望利用数据库的计算能力（毕竟数据库还可以集群分布等），因此可以使用**非缓存SQL数据集**。

非缓存SQL数据集将 SQL 语句传递给数据库，查询后分析界面会直接将结果集中的字段列出。

### Test.jsp:

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib uri="/WEB-INF/raqsoftAnalyse.tld" prefix="raqsoft" %>
<% String sqlld = request.getParameter( "sqlldvalue" );%>
<raqsoft:analysev2 />
<script>
guideConf.sqlld="<%=sqlld%>"; //指定sqlld
var sqlDatasets = [{
  sqlld:"sqlld1" //指定数据集名称，不可重复
  ,dataSource:"dqldemo" //指定SQL语句使用的数据源
  ,sql:"SELECT * FROM EMPLOYEE" //指定数据集SQL语句
  ,fields:null},... ..].... ..
</script>
```



# 4. 非缓存SQL数据集

访问test.jsp, 在 URL 为 sqlIdvalue参数传入数据集名称, 使用名称为 sqlId1 的数据集:

1	李强	0	中国
2	王胜	1	中国
3	赵明	1	中国
4	杜邦	2	美国
5	罗斯	2	英国
6	冯平平	0	中国
7	郭苗苗	0	中国
8	王强	3	中国

复杂条件	
	上表头
DEPTID ▼ COUNTRY ▼	ID计数 ▼ +

非缓存SQL数据集的分析过程是通过在原始SQL上追加更多的查询条件、分组、聚合子句方式实时在数据库中查询, 如:  
*select f1 , sum(f2) from ({ 原始 sql}) t where ... group by ... having ...*



# -字段标题与显示值-

PART FOUR



## 字段标题与显示风格

# 直接在SQL上修改字段标题与显示值

如下所示，可在SQL中通过 **as** 给每个字段标题重命名；使用**left join**将部门编号显示为部门名称：  
SELECT EMPLOYEE.Name **as 雇员姓名**, DEPARTMENT.NAME **as 部门**, EMPLOYEE.COUNTRY **as 国家** FROM  
EMPLOYEE **LEFT JOIN DEPARTMENT ON EMPLOYEE.DEPTID=DEPARTMENT.ID**

SQL数据集 保存

DQL表 数据库查询 (DQL/SQL) DFX文件 DFX脚本 报表文件

dqldemo

```
SELECT EMPLOYEE.Name as 雇员姓名, DEPARTMENT.NAME as 部门, EMPLOYEE.COUNTRY as 国家 FROM  
EMPLOYEE LEFT JOIN DEPARTMENT ON EMPLOYEE.DEPTID=DEPARTMENT.ID
```

q1532312490216.txt 查询数据, 缓存入文件

雇员查询LEFTJOIN

雇员姓名	部门	国家
张勇	销售部	中国
李强	销售部	中国
王胜	产品部	中国
赵明	产品部	中国
杜邦	测试部	美国
罗斯	测试部	英国
冯平平	销售部	中国
郭苗苗	销售部	中国
王强	研发部	中国
陈家禾	研发部	比利时



## 字段标题与显示风格

还可以在自助报表界面中修改字段标题

Figure 1: Initial state of the data analysis interface. The table shows columns 'NAME' and 'COUNTRY'. The 'NAME' field is highlighted in red. The '数据分析师' (Data Analyst) panel shows 'NAME' and 'COUNTRY' fields. The '复杂条件' (Complex Conditions) panel shows 'NAME' and 'COUNTRY' fields.

NAME	COUNTRY
张勇	中国
李强	中国
王胜	中国
赵明	中国
杜邦	美国

Figure 2: The '编辑' (Edit) button is highlighted in red in the 'NAME' field's context menu. The '编辑' dialog box is shown with '姓名' (Name) entered in the text field.

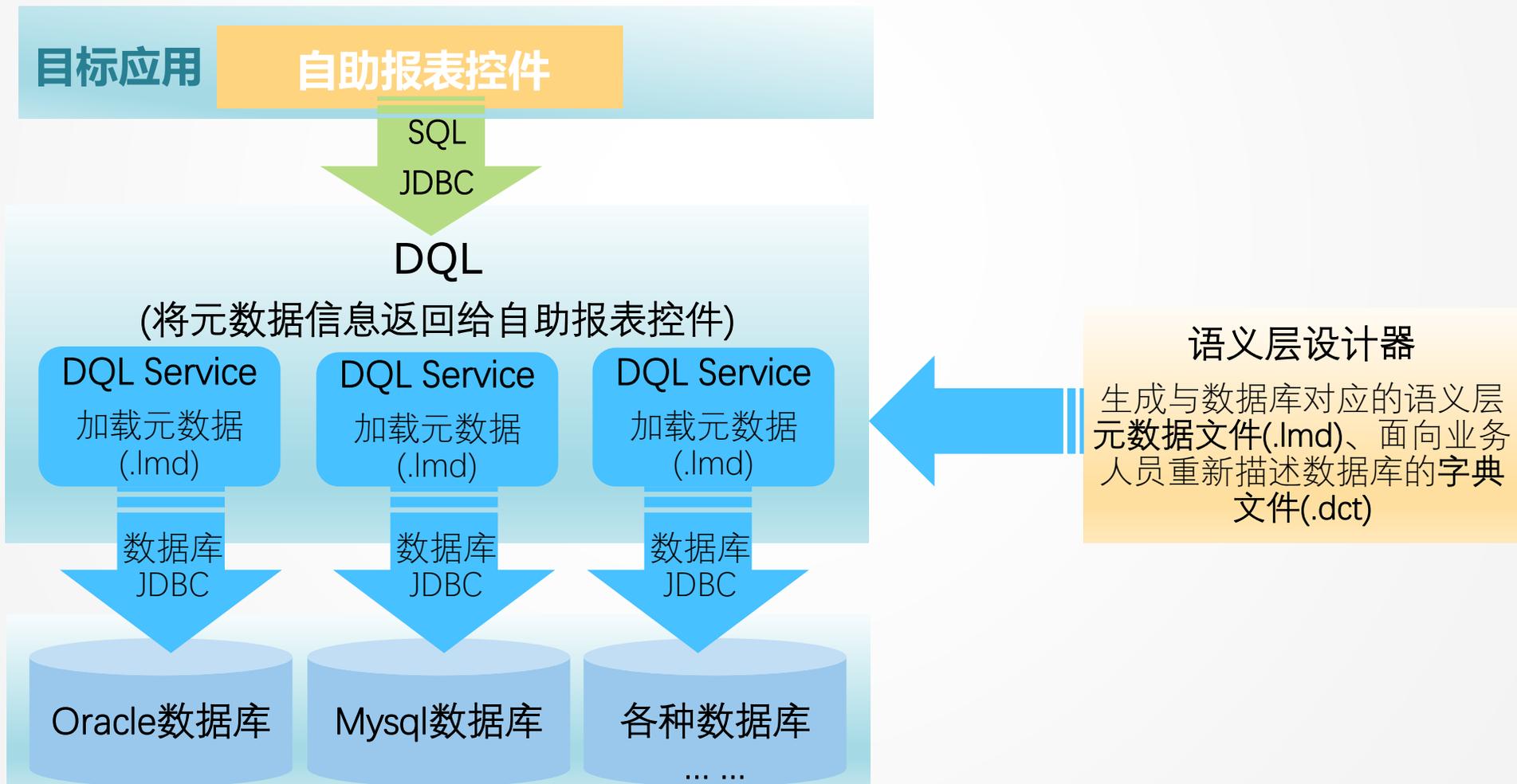
Figure 3: The final state of the data analysis interface. The table shows columns '姓名' and 'COUNTRY'. The 'NAME' field is highlighted in red. The '数据分析师' (Data Analyst) panel shows 'NAME' and 'COUNTRY' fields. The '复杂条件' (Complex Conditions) panel shows '姓名' and 'COUNTRY' fields.

姓名	COUNTRY
张勇	中国
李强	中国
王胜	中国
赵明	中国
杜邦	美国



## 字段标题与显示值

但每次针对SQL语句或界面修改显示太麻烦，如何能够自动生成标题和显示值呢？  
——语义层





1 创建语义层文件(元数据、字典)

2 启动DQL Server服务

3 配置数据源信息

4 使用Tag标签发布

5 页面展现



# 语义层

元数据 (.lmd) — 描述数据库中，各个表中的数据结构以及各类表间关系

字典 (.dct) — 定义表或字段的别名

可视 (.vsb) — 设定表或字段的可见



## 步骤一：配置物理数据源

打开元数据层设计器，系统菜单选择【数据源】，配置数据源并连接，连接之前，要保证所连接数据源已启动。

以下演示数据为润乾自带的dqldemo数据库，该数据库不用配置，通过【服务 - 启动示例数据库】菜单可直接启动数据库。





步骤二：创建元数据文件，文件后缀为.lmd。



文件菜单栏下  
选择【新建】，  
选择元数据



系统菜单下选择  
【导入数据库表】，  
选中需要的  
物理表，导入表  
结构



步骤三：定义字段标题。

应用层

将回款表中的销售ID字段重命名为销售

表名	回款单	物理表名	回款单		
记录数	0	表描述			
<p>常规 外键 伪字段 基础表 汇总表 子表</p>					
<p>字段    输入字段名关键字进行查找: <input type="text"/></p>					
序号	字段名	主键	数据来源	数据类型	字段描述
1	回款单ID	<input type="checkbox"/>	回款单ID	整数	
2	回款日期	<input type="checkbox"/>	回款日期	日期	
3	客户ID	<input type="checkbox"/>	客户ID	字符串	
4	金额	<input type="checkbox"/>	金额	双精度数	
5	销售	<input type="checkbox"/>	销售ID	整数	



步骤四：定义显示值。

表名  物理表名

记录数  表描述

常规 外键 伪字段 基础表 汇总表 子表

字段

序号	字段名	主键	数据来源	数据类型	字段描述
1	雇员ID	<input checked="" type="checkbox"/>	雇员ID	整数	
2	姓名	<input type="checkbox"/>	姓名	字符串	
3	职务	<input type="checkbox"/>	职务	字符串	
4	出生日期	<input type="checkbox"/>	出生日期	日期	
5	雇佣日期	<input type="checkbox"/>	雇佣日期	日期	
6	上级	<input type="checkbox"/>	上级	整数	
7	籍贯城市编码	<input type="checkbox"/>	籍贯城市编码	整数	

为雇员表  
设置主键

表 视图 维和层 导入元数据

维

序号	维名	源字段名	时间维类型
1	雇员	雇员.雇员ID	缺省
2	回款单_回款单ID	回款单.回款单ID	缺省

给维字段设置  
维名，便于页  
面显示值下拉  
列表显示更友  
好



步骤五：生成字典文件，定义雇员维的显示值，文件后缀为.dct。

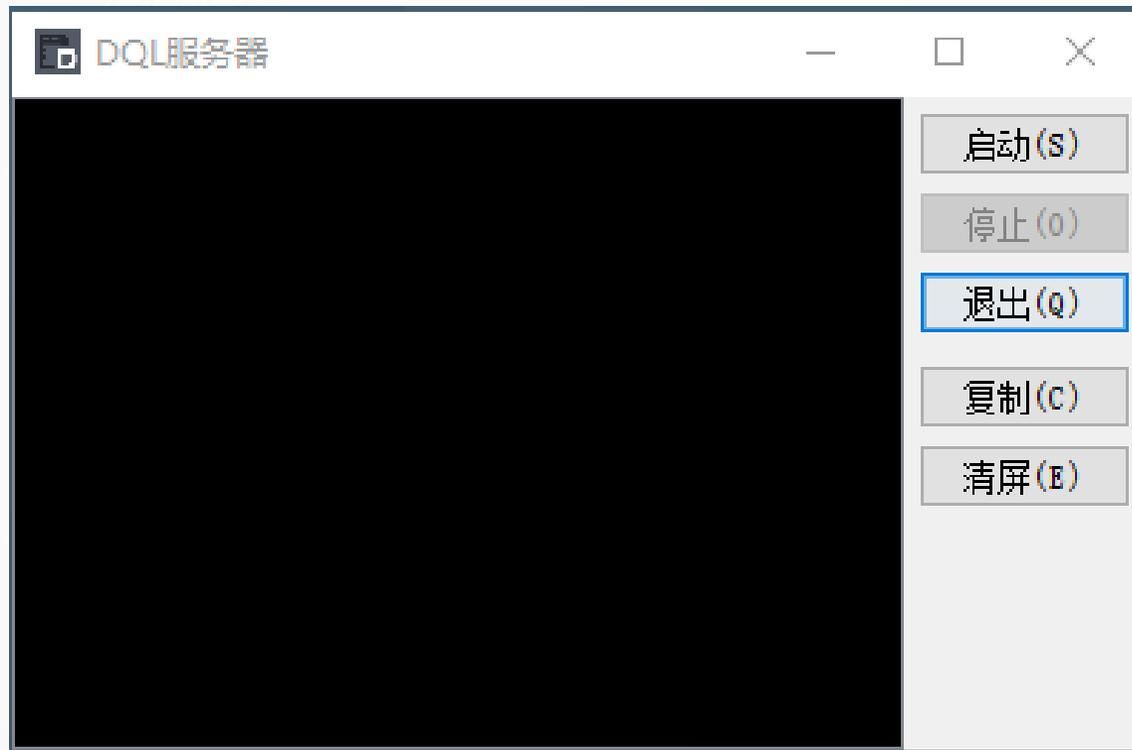
打开前一步做好的元数据文件，系统菜单下选择【生成字典】

序号	维标题	维名称	代码列名称	代码值标题	显示列名称	显示值标题
1	雇员	雇员	雇员.雇员ID		雇员.姓名	

设置雇员的显示列字段，销售名字就可以用于分析结果和过滤条件的下拉显示



需要使用 DQL Server 服务器对元数据文件(.lmd)进行语义解析，DQL 服务器控制台中启动页面如下所示，启动前要保证元数据中使用的物理数据库为启动状态：



- (1) DQL 这个词在界面中出现过多次，想要对DQL做更多了解可参考《[分析教程](#)》；
- (2) 元数据文件需要部署到 DQL Server 服务器，如需了解DQL Server的部署，可以到《[分析教程](#)》中查看关于 DQL 服务器部署集成的专题；
- (3) 字典文件是部署到 web 应用目录下，并在发布分析控件的页面里通过 JS API 去灵活设置，关于这部分的介绍，可以到《[分析教程](#)》JS API 相关主题中查看。



在[WEB应用根目录]/WEB-INF/raqsoftConfig.xml中添加物理数据源和DQL Server数据源

```

<?xml version="1.0" encoding="UTF-8"?><Config Version="1"><Runtime >
//配置应用需要用的数据源
<DBList>
<!--数据使用的物理数据源-->
<DB name="dqldemo">
  <property name="url" value="jdbc:hsqldb:hsq://127.0.0.1/reportdemo" />
  <property name="driver" value="org.hsqldb.jdbcDriver" />
<!--数据库类型编号，编号依次代表的数据库类型为：[UNKNOWN:0; ORACLE:1; SQLSVR:2; SYBASE:3; SQLANY:4; INFMIX:5;
FOXPRO:6; ACCESS:7; FOXBAS:8; DB2:9; MYSQL:10; KINGBASE:11; DERBY:12; HSQL:13; TERADATA:14; POSTGRES:15;
DATALOGIC:16; IMPALA:17; ESSBASE:101;] -->
  <property name="type" value="13" />
  <property name="user" value="sa" />
  <property name="password" />
  ... ..
</DB>
<!--DQL Server数据源-->
<DB name="DataLogic">
  <property name="url" value="jdbc:datalogic://127.0.0.1:3366/datalogic" />
  <property name="driver" value="com.datalogic.jdbc.LogicDriver" />
  <property name="type" value="16" />
  <property name="user" value="root" />
  <property name="password" value="root" />
  ... ..
</DB> </DBList> ... ..
</Config>

```



在jsp中定义Tag标签的属性及值，用Tag标签完成多维分析的解析及运算，按用户配置的属性要求将HTML文件流输出到客户端。

Test. jsp:

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
```

```
<%@ taglib uri="/WEB-INF/raqsoftAnalyse.tld" prefix="raqsoft" %>
```

```
<raqsoft:analysev2
```

```
    dataSource="DataLogic"           //指定raqsoftConfig.xml中定义的DQL Server数据源
```

```
    ... .. />
```

```
<script>
```

```
    guideConf.dct="WEB-INF/test.dct"; //指定字典文件路径
```

```
</script>
```





如何将上一步分析结果中的销售 ID 显示为销售名称？





如何只看部分销售的回款情况呢？销售列点【数据过滤】菜单，设置过滤条件：

销售 ▾

- 升序
- 降序
- √ 不排序
- 编辑
- 数据过滤**
- 显示格式
- 显示值
- 删除

简洁显示 (简洁显示会隐藏掉“新增条件、调整条件层次”的功能, 更能凸显条件之间的层次关系)

→ 销售      等于      1,3,5,6

AND | OR

- 张颖
- 王伟
- 李芳
- 郑建杰
- 赵军
- 孙林
- 金士鹏
- 刘英玫
- 张雪眉

张颖	13267
李芳	15623
赵军	37014
孙林	719



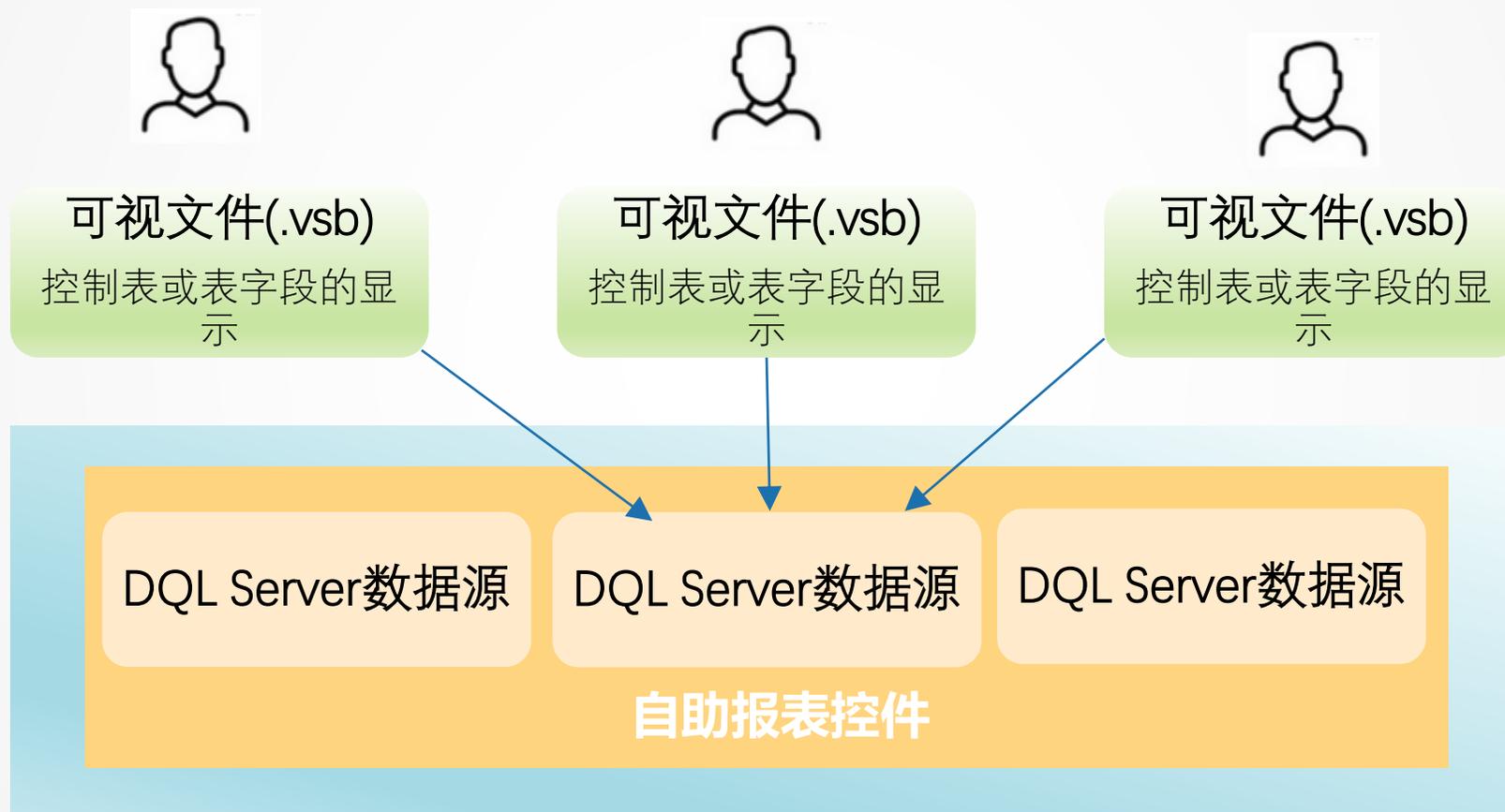
# -权限控制-

PART FIVE



## 权限控制

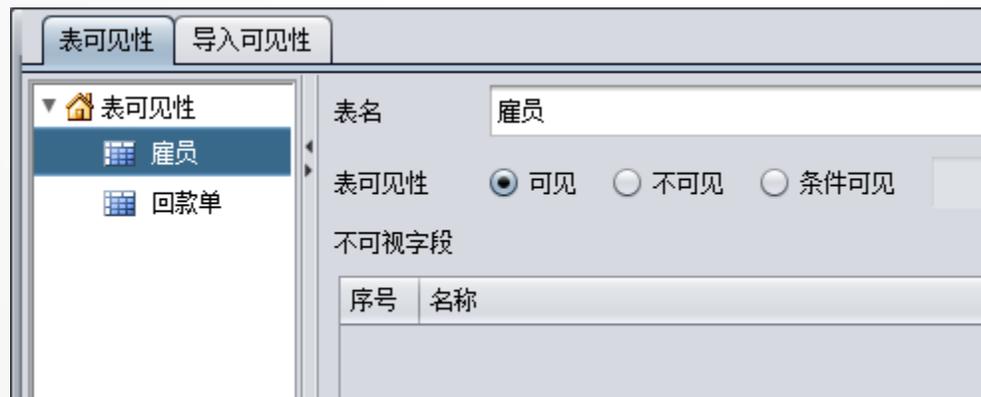
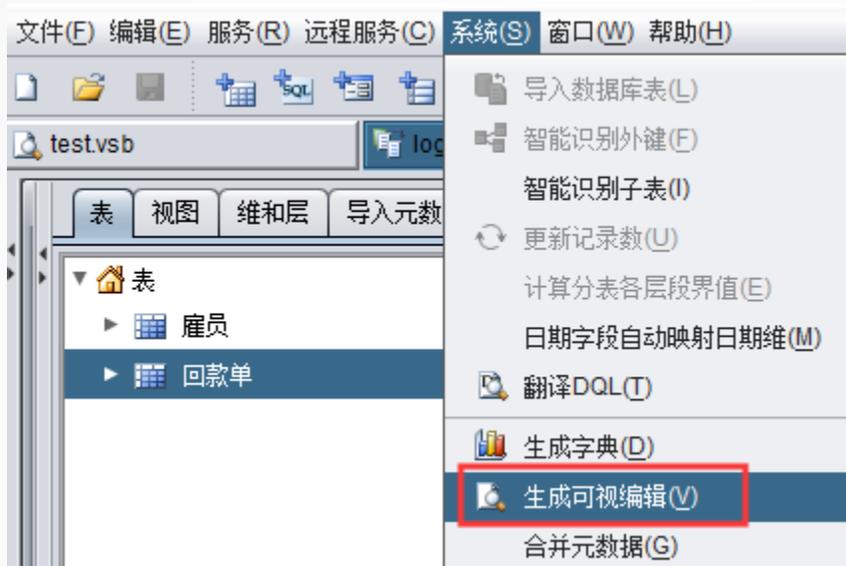
自助报表中的权限控制是使用语义层的 .vsb（可视文件）控制表或字段的可见。





## 权限控制

打开上一节做好的元数据文件，系统菜单下选择【生成可视编辑】，文件后缀为.vsb。



## 权限控制



将雇员表设为不可见，并且为回款单添加条件可见表达式，过滤出销售ID为3的回款记录：





## 权限控制

新建Test.jsp内容如下:

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="com.raqsoft.*" %>
<%@ page import="java.util.*" %>
<%@ taglib uri="/WEB-INF/raqsoftAnalyse.tld" prefix="raqsoft" %>
```

```
<raqsoft:analysev2
dataSource="DataLogic"
/>
<script>
guideConf.vsb="/WEB-INF/test.vsb"; //指定要引用的可视文件
</script>
```

报表名称

销售	金额
3	335
3	266
3	2646
3	3740
3	982

数据分析设置器

- 回款单
  - 回款单ID
  - 回款日期
  - 客户ID
  - 金额
  - 销售

	上表头
	指标标题
左表头	销售 ▼
	金额 ▼
	+



多个用户使用同一个数据源时，需要根据不同情况为每个用户分别制作可视文件(.vsb)。为了减少工作量，如何能让多个用户公用一个可视文件呢？

**宏**——利用宏可以在可视文件中动态进行表达式的替换，根据用户传入的不同宏值，返回不同的记录。



## 步骤一：创建可视文件，添加DQL宏

打开前面做好的可视文件(.vsb)，修改回款单条件可见表达式。表达式支持使用DQL宏，宏的语法为 $\${宏名称}$ 。宏只能用在条件可见里，其中宏名称就是 $\{\}$ 中变量名称；宏值就是 $\{\}$ 里面变量的值。

下面以回款单为例，介绍如何通过宏的设置，让销售可以使用同一个可视文件(.vsb)分别查询出各自的回款情况：





## 步骤二：在多维分析JSP中添加DQL宏的引入

多维分析集成到应用系统，通常由应用系统负责提供当前登录用户的信息。本例为方便演示，直接在jsp指定固定宏值，需要手动改这个值模拟不同登录用户。

新建Test.jsp内容如下：

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="com.raqsoft.*" %>
<%@ page import="java.util.*" %>
<%@ taglib uri="/WEB-INF/raqsoftAnalyse.tld" prefix="raqsoft" %>
<%
    String userId="3";//填入宏值
    String vsb = "/WEB-INF/test.vsb";//指定要引用的可视文件
    String dataSource = request.getParameter( "dataSource" );
    if (dataSource == null) dataSource = "DataLogic";
    if(vsb==null) vsb = "";
    vsb = vsb.replaceAll("\\\\", "/");
    //在session中设置DQL表的权限条件
    ArrayList filters = new ArrayList();
    session.setAttribute("_raqsoft_filters_",filters);
    if (dataSource.length()>0) {
        DQLTableFilter f = new DQLTableFilter("default",dataSource);
        filters.add(f);
        if (vsb.length()>0) f.setVsb(vsb);
        f.getParamValues().put("雇员",userId); //为vsb中条件可见的DQL宏传递宏值 }%>
<raqsoft:analysev2
    dataSource="DataLogic"
    fixedTable="回款单"
/>
<script>guideConf.vsb="/WEB-INF/test.vsb"; </script>
```



### 步骤三：访问多维分析页面，做自助报表查询

报表名称

销售	客户ID	金额
3	WHITC	335
3	DUMON	266
3	PICCO	982
3	WHITC	2646
3	PICCO	3740
3	FRANS	23
3	PICCO	230
3	WHITC	478
3	WHITC	483
3	PICCO	248

数据分析设置器

报表列表

添加报表

回款单

- 回款单ID
- 回款日期
- 客户ID
- 金额
- 销售

上表头

指标标题

左表头

销售

客户ID

金额

自助报表查询拖拽回款单表的指标时，会自动在DQL语句后拼上条件可见表达式，如：

```
SELECT * FROM 回款单 where 销售 = ${雇员}
```

假设雇员为“李芳”，从登录的session界面获取到雇员值为”李芳“，李芳的雇员ID为3，然后将宏值3传递进去：最终生成的DQL语句为：SELECT \* FROM 回款单 WHERE 销售 = 3

想要了解更多 请联系我们



THANK YOU