

如何让应用中的SQL适应不同类型数据库

集算器 - 润乾软件出品



› 序言



应用开发中，我们会涉及多种不同类型的数据库，虽然SQL语法大体是相似的，但是不同数据库仍然有各自的方言，特别是自有函数，比如取当前日期，ORACLE的写法：

```
select sysdate from dual;
```

而SQLServer的写法为：

```
select getdate()
```

这时如果想要让应用在不同数据库间切换，就需要改写SQL语句，耗时耗力。不过，若借助集算器执行SQL，就简单多了，针对此类问题，集算器提供了sqltranslate函数，可以将SQL语句转换不同数据库。

1) 通过集算器执行SQL

- 应用结构
- 集成集算器
- 制作SPL脚本
- 调用脚本文件

2) 利用JDBC网关直接执行SQL

- 应用结构
- 编写网关脚本
- 配置raqsoft.xml文件
- 使用网关

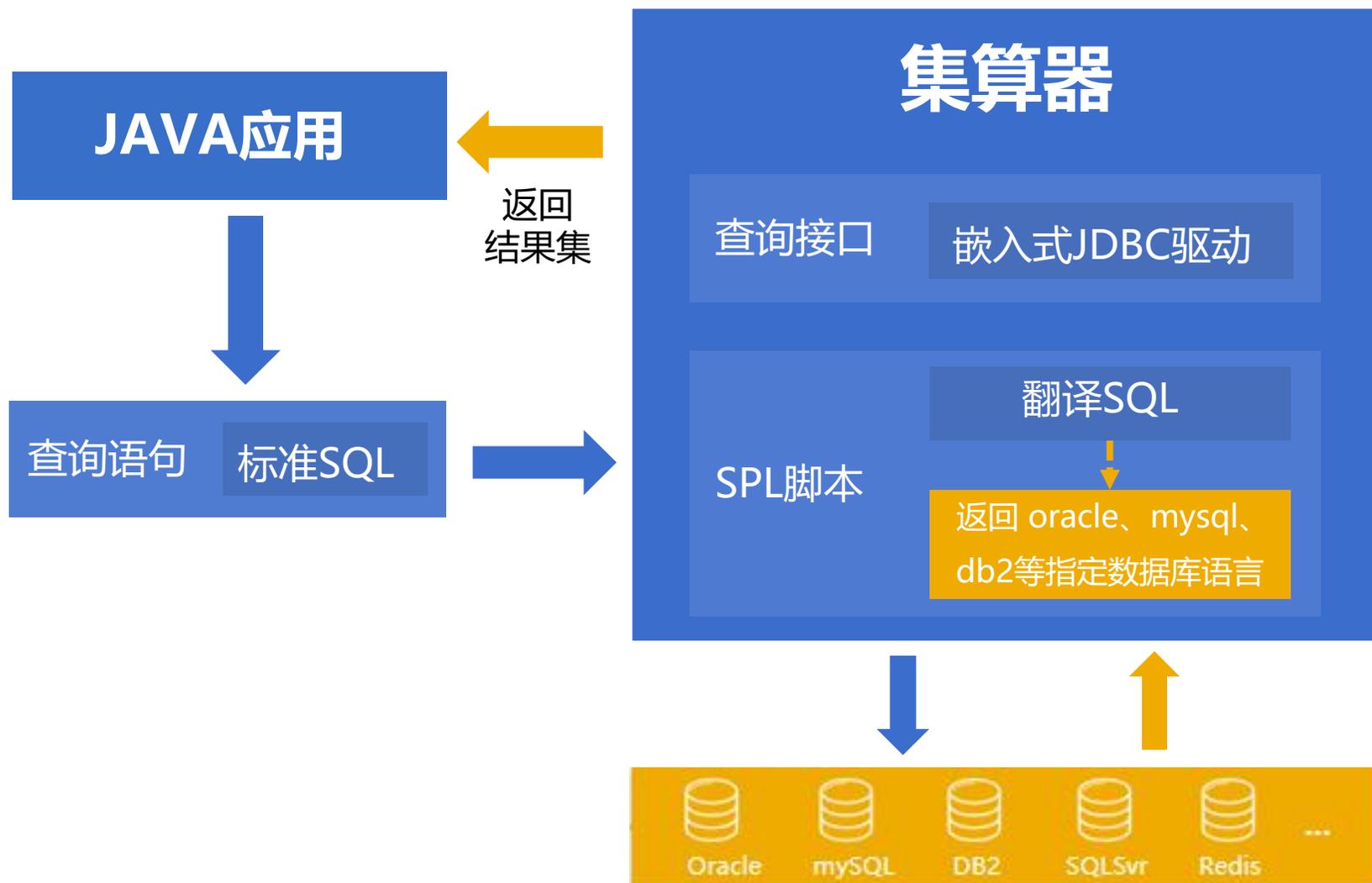
3) 在报表工具中使用

- 集成集算器
- 创建数据集
- 制作报表并预览

4) 维护数据库类型



通过集算器执行SQL-应用结构



集算器是一款应用与数据之间的计算中间件，可被集成到Java应用中；

Java应用将标准的SQL语句传递到SPL脚本，翻译成目标数据库的SQL语法，查询后返回结果。



通过集算器执行SQL-集成集算器

1、Java应用集成集算器

Java应用集成集算器操作可分为两步：

1. 在Java应用中部署集算器驱动jar包
2. 在raqsoftConfig.xml中配置运行环境

其中，实际用到的数据库信息也可在raqsoftConfig.xml中配置，以下为数据源连接的主要代码，

详细操作可参考乾学院文档：<http://c.raqsoft.com.cn/article/1543657951867>

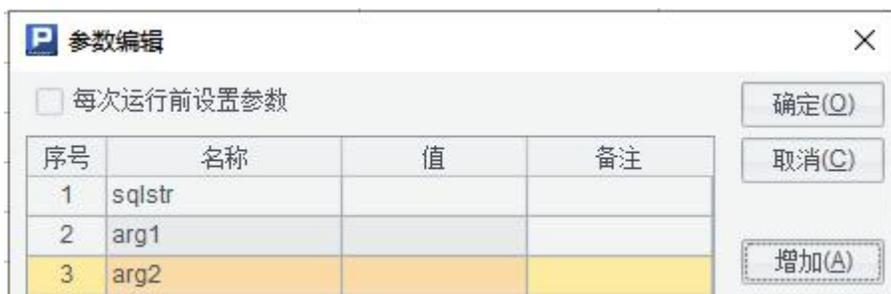
通过集算器执行SQL-集成集算器



```
<DBList>
<DB name="dsMysql" > <!--数据源名称-->
<property name="url" value="jdbc:mysql://192.168.0.110:3306/demo" />
<!-- url连接-->
<property name="driver" value="com.mysql.jdbc.Driver" /> <!--数据库驱动-->
<property name="type" value="10" /> <!--数据库类型-->
<property name="user" value="root" /> <!--用户名-->
<property name="password" value="sw11"/> <!--密码-->
... </DB>
</DBList>
```

通过集算器执行SQL-制作SPL脚本

2、制作被调用的SPL脚本文件，用来翻译SQL，文件命名为sqltranslate.dfx



参数 `sqlstr` 表示标准SQL串
`arg1`、`arg2` 表示SQL语句中的参数

脚本内容如下：

	A
1	=sqlstr.sqltranslate("MYSQL")
2	=connect("dsMysql").query@x(A1,arg1,arg2)

› dfx脚本说明



参数sqlstr值设为:

```
select TYPE,month(DATE) MONTH,count(TYPE) TYPE_COUNT,sum(Amount) AMOUNT from Order
where Amount >? and Amount < ?
group by TYPE,month(DATE)
```

参数arg1值设为: 10000 参数arg2值设为: 50000

A1: 通过修改函数sqltranslate中的参数来切换数据库类型:

当参数值为ORACLE时, 将sqlstr中的SQL翻译为Oracle数据库语法, 结果为:

```
select TYPE,EXTRACT(MONTH FROM DATE) MONTH,count(TYPE) TYPE_COUNT, sum(Amount) AMOUNT
from Order
where Amount >? and Amount < ?
group by TYPE,EXTRACT(MONTH FROM DATE)
```

当参数值为MYSQL时, 将sqlstr中的SQL翻译为MySQL数据库语法, 结果为:

```
selectTYPE,month(DATE) MONTH,count(TYPE) TYPE_COUNT,sum(Amount) AMOUNT
from Order
where Amount >? and Amount < ?
group by TYPE,month(DATE)
```

A2: 在指定数据库中执行翻译后的SQL, 例如使用前面xml文件中配置的mysql数据源, 则此处设置为dsMysql。

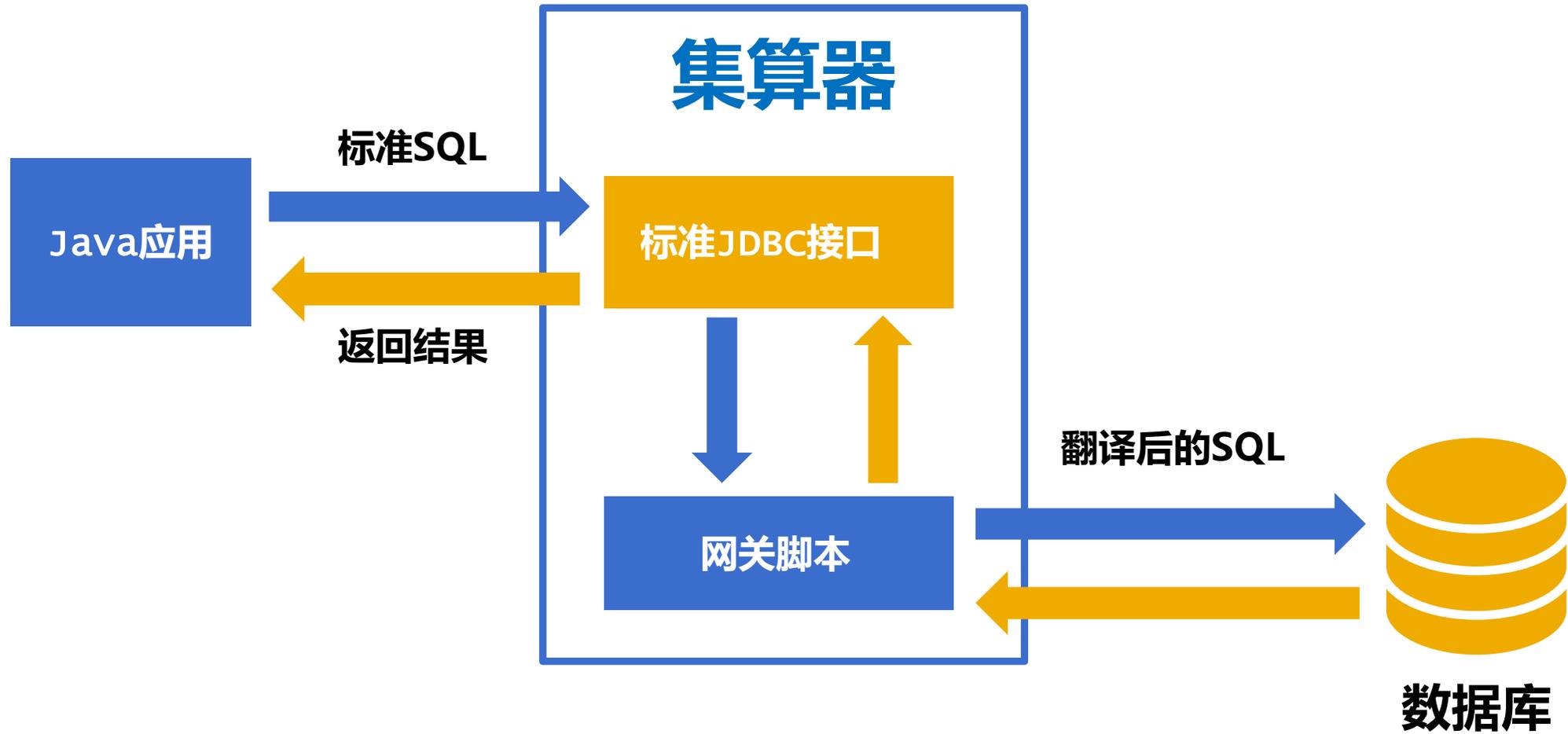


在java代码中调用dfx脚本文件

主要代码如下:

```
//通过call调用存储过程，其中sqltranslate是dfx的文件名，?表示参数
st = con.prepareCall("call sqltranslate(?,?)");
st.setObject(1, "select TYPE,month(DATE) MONTH,count(TYPE)
                TYPE_COUNT,sum(Amount) AMOUNT
                from Order
                where Amount >?
                group by TYPE,month(DATE)");
st.setObject(2, 10000);
st.setObject(3, 50000);
//获取结果集
ResultSet rs = st.executeQuery();
```

利用JDBC网关直接使用SQL





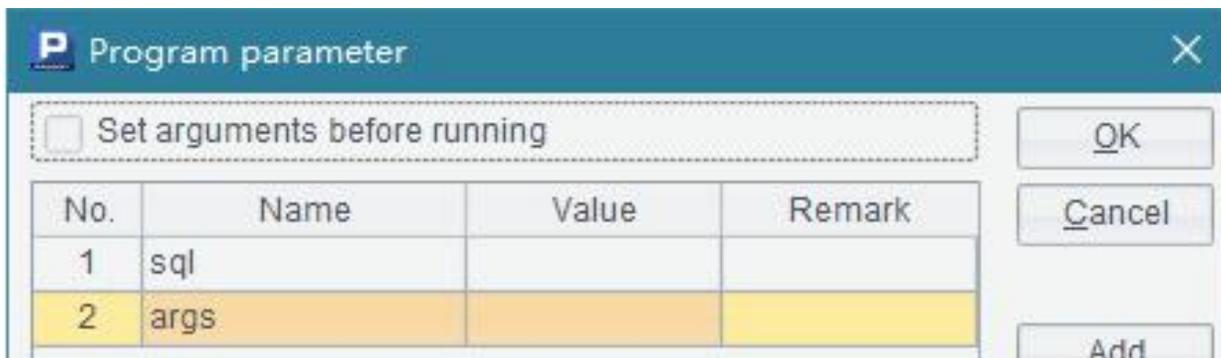
利用JDBC网关直接使用SQL

在用JDBC模式调用集算器时，可以配置一个SPL脚本作为JDBC网关，在JDBC中执行的所有语句都会先交由这个脚本处理执行。

我们可以利用这一点让集算器JDBC直接接收SQL，在网关中把SQL翻译成目标数据库的语法，执行后并返回结果。这样，应用就像访问原来的数据库一样直接使用SQL了。

利用JDBC网关直接使用SQL

1、制作网关文件，命名为executeSQL.dfx



网关文件中必须有的两个参数：

sql表示调用网关时传入的SQL语句，

args表示SQL语句中的参数

网关文件脚本如下：

	A	
1	=sql.sqltranslate("MYSQL")	
2	=A1.split().pos@a("?").count()	//计算参数个数
3	=A2.("args("+string(~)+")").concat@c()	//拼出参数
4	=if(args==null,"A1","A1,"+A3)	
5	=connect("dsMysql").query@x(\${A4})	
6	return A5	



› 利用JDBC网关直接使用SQL

2、配置文件raqsoftConfig.xml, 在<JDBC>节点中的<gateway>指定网关文件指定网关文件。

```
<JDBC>  
<load>Runtime</load>  
<gateway>executeSQL.dfx</gateway>  
</JDBC>
```

详细内容可参考文档: <http://doc.raqsoft.com.cn/esproc/tutorial/pzraqsoftconfig.html>

➤ 利用JDBC网关直接使用SQL



3、Java代码中传递参数给网关文件

```
st = con.prepareCall("select TYPE,month(DATE)  
    MONTH,count(TYPE) TYPE_COUNT,sum(Amount) AMOUNT  
    from order_wines  
    where Amount >? and Amount<?  
    group by TYPE,month(DATE) ");
```

//设置参数

```
st.setObject(1,10000); //传入SQL中的第一个参数值  
st.setObject(2,50000); //传入SQL中的第二个参数值
```

```
st.execute(); //执行存储过程
```



在报表工具中使用

Java报表工具属于Java应用的一种，也可以集成集算器，下面看下如何在报表工具中实现SQL适应不同类型数据库。

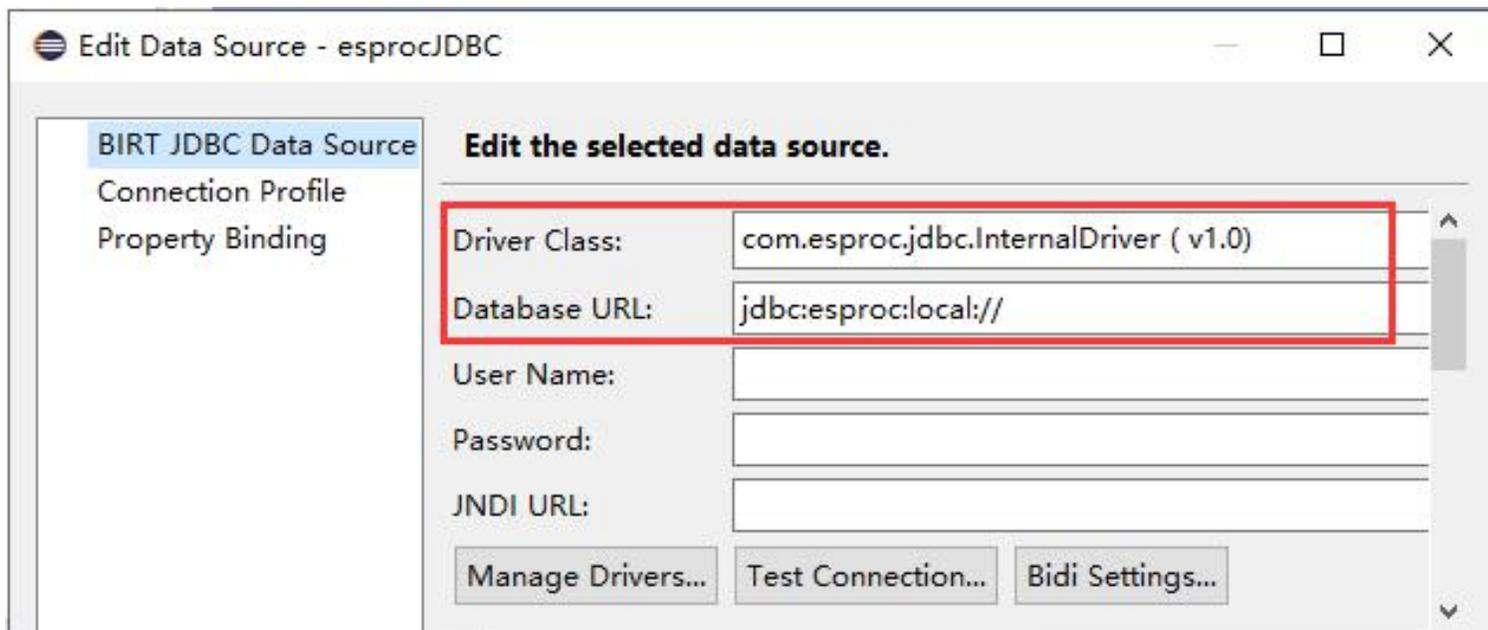
1、Java应用集成集算器

报表工具部署集算器JDBC的方法，和普通Java应用原理相同，在Java应用中部署集算器驱动jar包，并在raqsoftConfig.xml中配置运行环境即可。

在报表工具中使用

2、报表调用集算器脚本

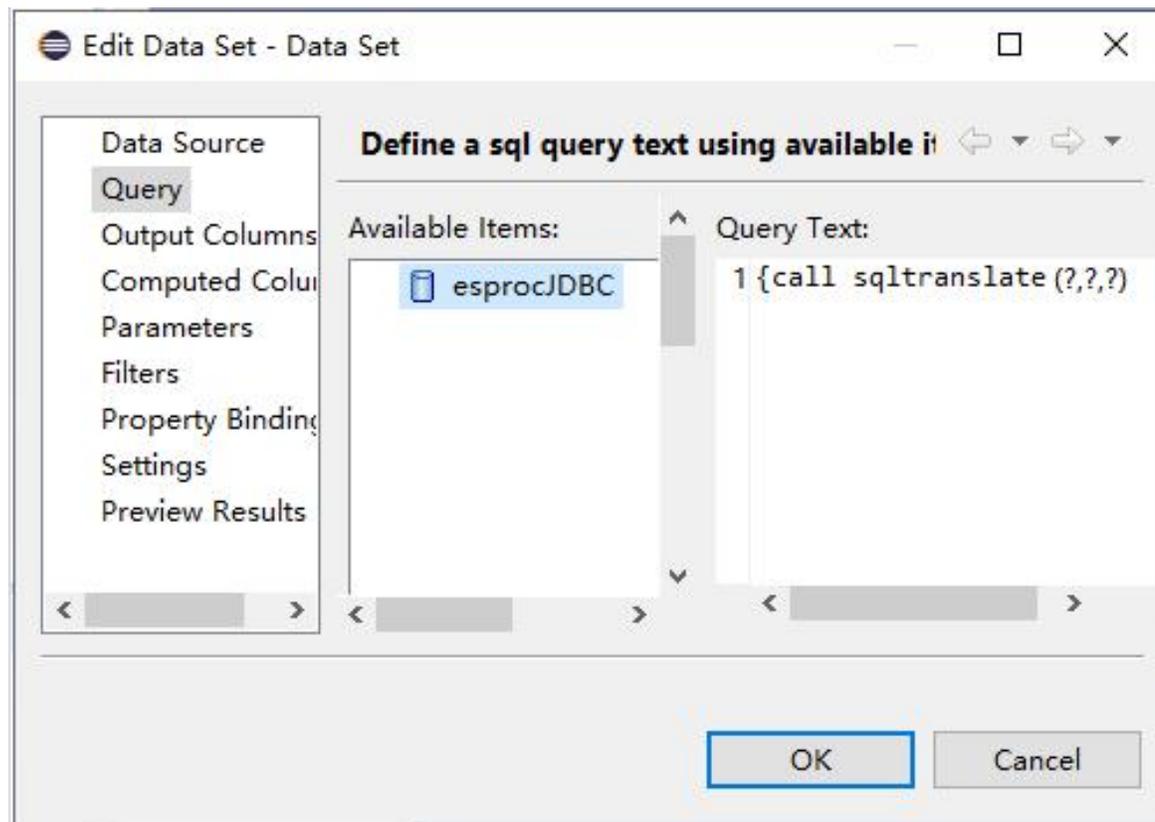
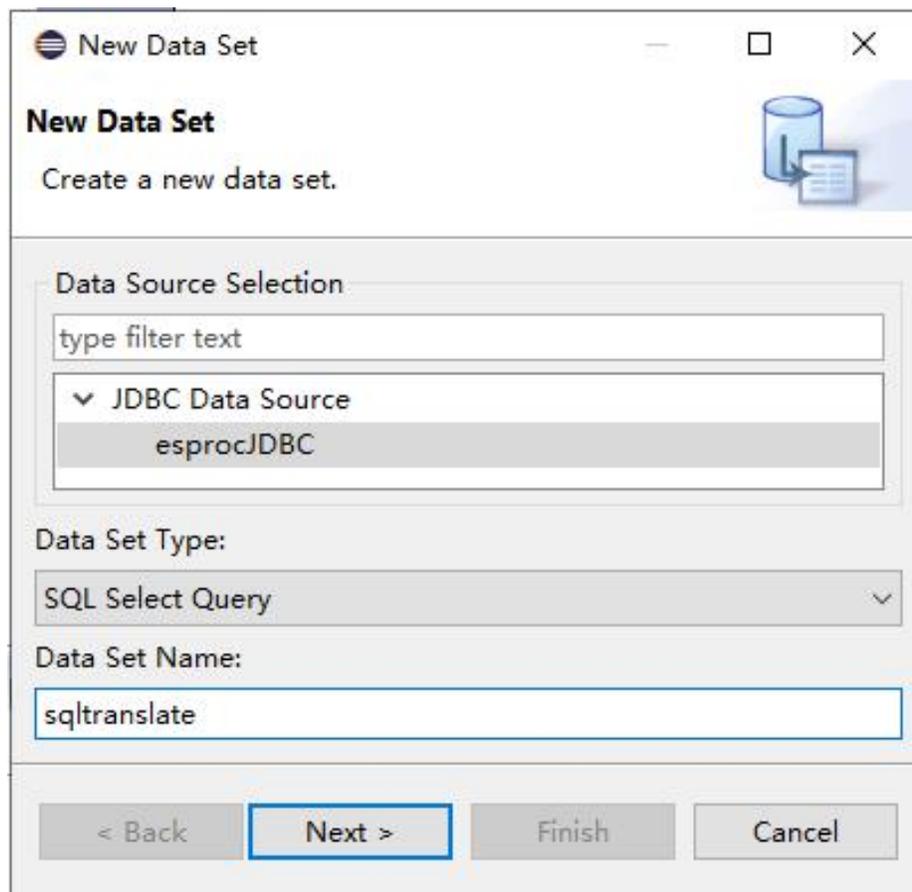
dfx脚本文件依然为前面制作好的`sqltranslate.dfx`，首先在报表IDE中建立数据源，并指向集算器，下面以BIRT报表为例：



在报表工具中使用



新建存储过程数据集：选择集算器数据源，调用集算器脚本文件



在报表工具中使用

设定参数，参数值为标准SQL:

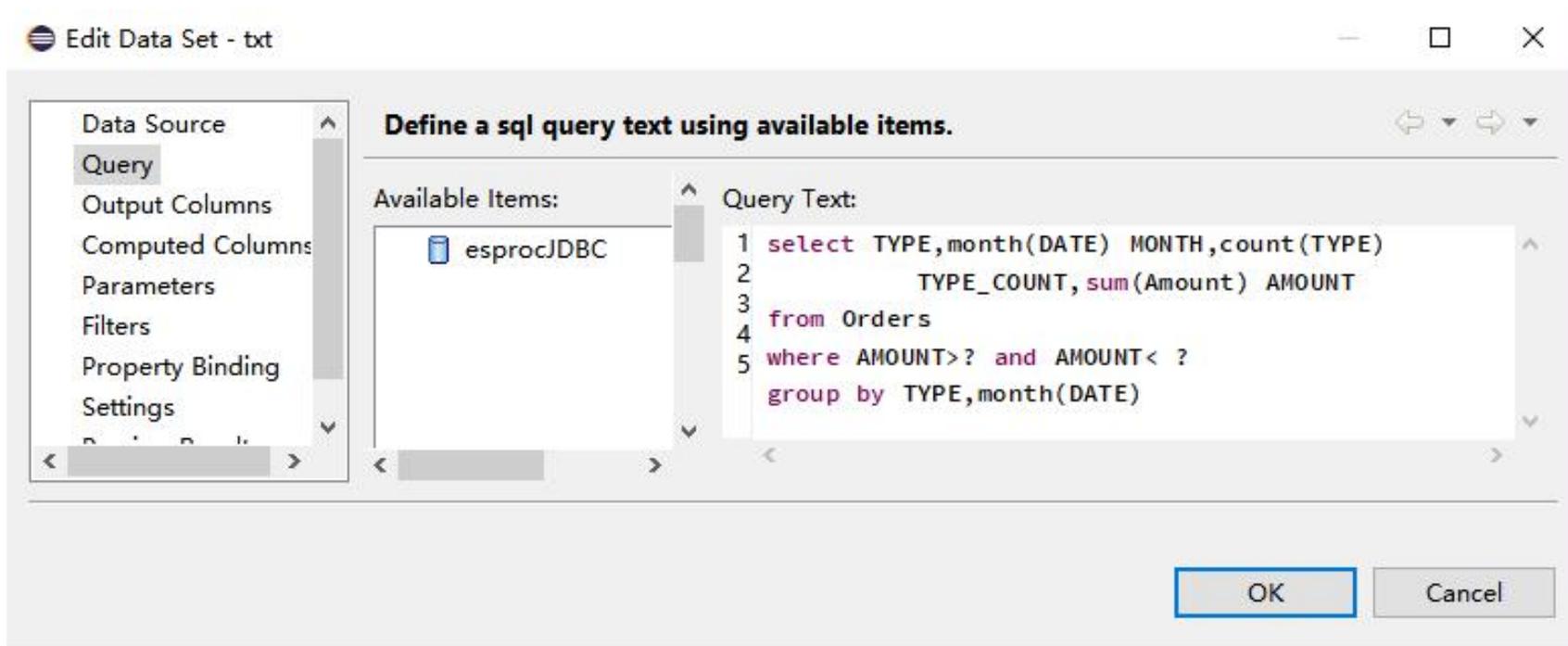
```
select TYPE,month(DATE) MONTH,count(TYPE) TYPE_COUNT, sum(Amount) AMOUNT  
from Order  
where AMOUNT>? and AMOUNT< ?  
group by TYPE,month(DATE)
```

	Name	N..	Data Type	Direction	Default Value	Linked To
1	sql		String	Input	select TYPE,m...	None
2	arg1		Integer	Input	10000	None
3	arg2		Integer	Input	50000	None

在报表工具中使用

在报表中使用时，也可以用JDBC网关的方式，同样的，在 raqsoftConfig.xml中添加<gateway>节点配置网关文件，并进行数据源配置。

此时报表数据集配置如下：



在报表工具中使用



参数配置如下：

	Name	N..	Data Type	Direction	Default Value	Linked To
1	arg1		Integer	Input	10000	None
2	arg2		Integer	Input	50000	None



在报表工具中使用

设计报表，预览报表，其用法和普通数据集一样：

report4.rptdesign

TYPE	MONTH	TYPE_COUNT	AMOUNT
[TYPE]	[MONTH]	[TYPE_COUNT]	[AMOUNT]
Footer Row			

BIRT Report Viewer

Showing page 1

TYPE	MONTH	TYPR_COUNT	AMOUNT
Wines	1	4248	1.0632367298000015E8
Wines	2	3834	9.536151428999996E7
Wines	3	4247	1.0532002361000012E8
Wines	4	4110	1.0332843257000005E8
Wines	5	4247	1.0693380060999994E8
Wines	6	4109	1.0085924089999986E8
Wines	7	4248	9.870287694000016E7
Wines	8	4244	1.0688678374000014E8
Wines	9	4111	9.844279791999968E7
Wines	10	4247	1.0167565124E8
Wines	11	4108	1.0034451196000013E8
Wines	12	4247	1.0441973171000035E8



维护数据库类型

关于SQL转换功能，默认支持的数据库类型有以下几种：

数据库类型	编号
ORACLE	1
SQLSVR	2
DB2	9
MYSQL	10
HSQL	13
TERADATA	14
POSTGRES	15



维护数据库类型

关于SQL翻译支持的数据库类型，用户还可手动进行维护，包含以下两种情况：

1. 对于默认存在的数据库类型，可对数据库函数进行修改和增加
2. 手动增加默认不存在的新的数据库类型

以上两种操作可在function.xml文件中进行配置，该文件位于dm.jar中/com/raqsoft/dm/sql/路径下。

维护数据库类型



function.xml

```
<FUNCTION name="HOUR" paramcount="1">
  <INFO dbtype="1" value="EXTRACT(HOUR FROM ?)" />
  <INFO dbtype="2" value="DATEPART(HH, ?)" />
  <INFO dbtype="9" value="" />
  <INFO dbtype="10" value="" />
  <INFO dbtype="13" value="" />
  <INFO dbtype="14" value="EXTRACT(HOUR FROM ?)" />
  <INFO dbtype="15" value="EXTRACT(HOUR FROM ?)" />
</FUNCTION>
<FUNCTION name="INDEXOF" paramcount="3">
  <INFO dbtype="1" value="INSTR(?2,?1,?3)" />
  <INFO dbtype="2" value="CHARINDEX(?1,?2,?3)" />
  <INFO dbtype="9" value="LOCATE(?1,?2,?3)" />
  <INFO dbtype="10" value="LOCATE(?1,?2,?3)" />
  <INFO dbtype="13" value="LOCATE(?1,?2,?3)" />
  <INFO dbtype="14" value="N/A" />
  <INFO dbtype="15" value="N/A" />
</FUNCTION>
<FUNCTION name="INDEXOF" paramcount="2">
  <INFO dbtype="1" value="INSTR(?2,?1)" />
  <INFO dbtype="2" value="CHARINDEX(?1,?2)" />
  <INFO dbtype="9" value="LOCATE(?1,?2)" />
  <INFO dbtype="10" value="LOCATE(?1,?2)" />
  <INFO dbtype="13" value="LOCATE(?1,?2)" />
  <INFO dbtype="14" value="INDEX(?2,?1)" />
  <INFO dbtype="15" value="strpos(?2,?1)" />
</FUNCTION>
<FUNCTION name="INT" paramcount="1">
```

其中,

name属性表示标准函数名称;

paramcount属性表示函数中的参数个数;

dbtype属性表示数据库类型编号;

value属性表示数据库中对应的函数写法,
value为空则默认为name的值。



维护数据库类型

新增数据库类型时，用户只需在function.xml中的FUNCTION节点中增加INFO节点，并在该节点中配置dbtype及value属性，dbtype需设置为0，value设置为该数据库的对应函数即可。

在sqltranslate中使用新增的数据库类型时，参数应写为**OTHER**。

```
<FUNCTION name="DATE" paramcount="1">  
  <INFO dbtype="0" value="FORM_UNIXTIME(?, 'YYYY-MM-DD') " />  
  <INFO dbtype="1" value="TO_DATE(?, 'YYYY-MM-DD') " />  
  <INFO dbtype="2" value="CONVERT(varchar(100), ?, 23) " />  
  <INFO dbtype="9" value="DATE(?) " />  
  <INFO dbtype="10" value="DATE_FORMAT(?, '%Y-%m-%d') " />  
  <INFO dbtype="13" value="TO_DATE(?, 'YYYY-MM-DD') " />
```

	A
1	=sqlstr.sqltranslate("OTHER")
	...