

How to migrate SQL in application to different types of databases

esProc – issued by Raqsoft

www.raqsoft.com





Preface



In application development, we will involve many different types of databases. Although the SQL syntax is similar, different databases still have their own dialects, especially their own functions, such as taking the current date: In Oracle, it is written as follows:

```
select sysdate from dual;
```

In SQL server, it is written as follows:

```
select getdate()
```

In this case, if you want to switch applications between different databases, you need to rewrite SQL statements, which is time-consuming and labor-consuming. However, it is much easier to execute SQL with the help of esProc. To solve this problem, esProc provides the sqltranslate function, which can transform SQL statements into different databases.



1) Execute SQL through esProc

- Application structure
- Integrate esProc
- Create SPL script
- Call script file

2) Using SQL directly with JDBC gateway

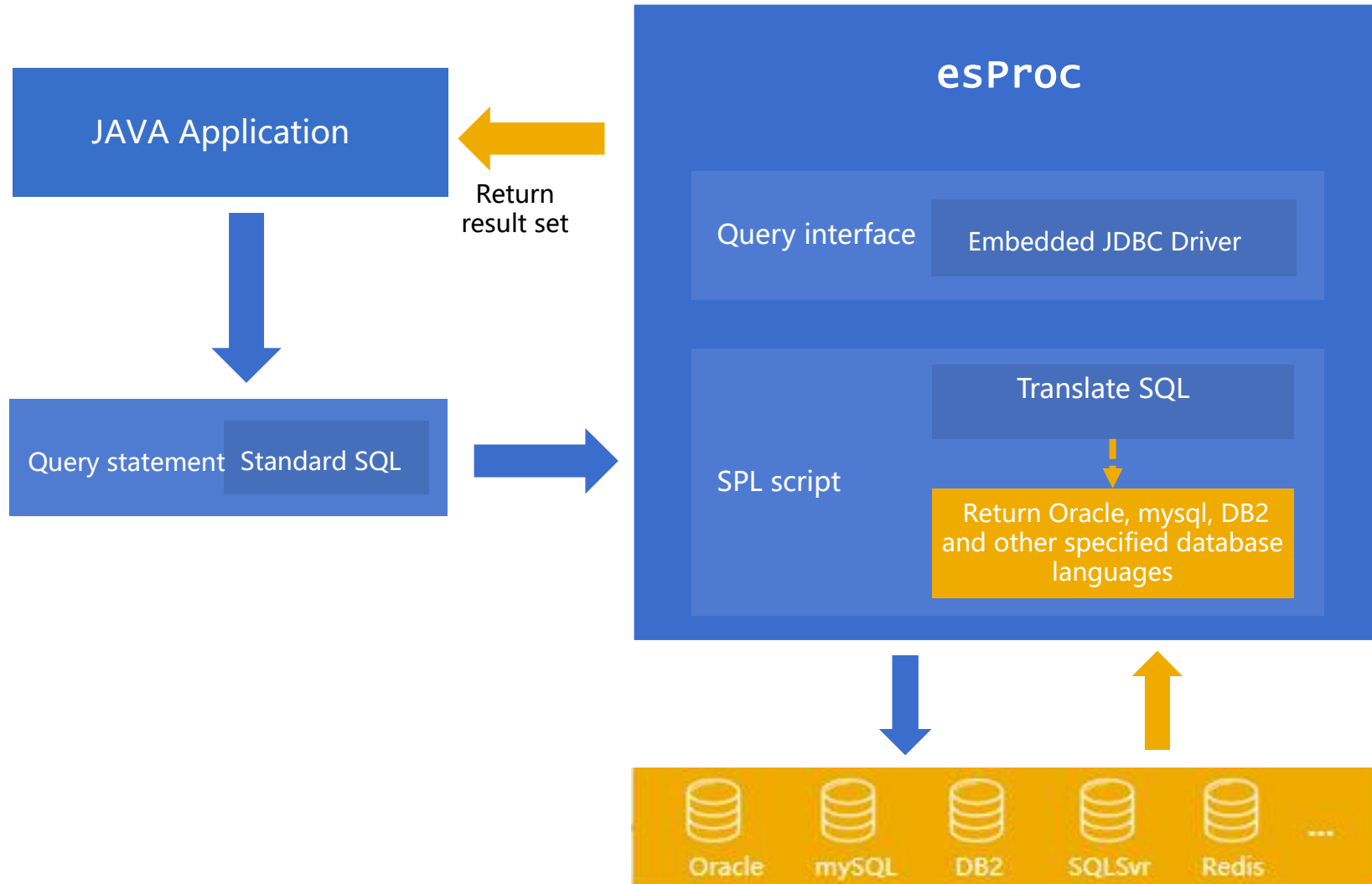
- Application structure
- Write gateway script
- Configure raqsoft.xml file
- Use gateway

3) Use in reporting tools

- Integrate esProc
- Create dataset
- Make report and Preview

4) Maintain database type

Execute SQL through esProc - Application structure



esProc is a kind of middleware between application and data, which can be integrated into Java application;

Java application transfers standard SQL statements to SPL script, translates them into SQL syntax of target database, and returns results after query.

➤ Execute SQL through esProc - Integrate esProc



1. Java application integrates esProc

The operation of Java application integrating esProc can be divided into two steps:

1. Deploying the esProc driver jar packages in Java application
2. Configure the running environment in raqsoftConfig.xml

Among them, the actual database information can also be configured in raqsoftconfig.xml. The following is the main code of data source connection. For detailed operations, please refer to:

<http://c.raqsoft.com.cn/article/1543657951867>

➤ Execute SQL through esProc - Integrate esProc



```
<DBList>
<DB name="dsMysql" > <!--data source name-->
<property name="url" value="jdbc:mysql://192.168.0.110:3306/demo" />
<!-- url connection-->
<property name="driver" value="com.mysql.jdbc.Driver" /> <!--Database driver-->
<property name="type" value="10" /> <!--Database type-->
<property name="user" value="root" /> <!--User name-->
<property name="password" value="sw11"/> <!--Password-->
... </DB>
</DBList>
```

➤ Execute SQL through esProc – Create SPL script



2. Make the called SPL script file to translate SQL. The file name is sqltranslate.dfx

No.	Name	Value	Remark
1	sqlstr		
2	arg1		
3	arg2		

Parameter **sqlstr** represents standard SQL string

Arg1 and **arg2** represent parameters in SQL

The script is as follows:

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

➤ DFX script description



Parameter sqlstr value is set to:

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

Parameter arg1 value is set to: 10000

Parameter arg2 value is set to : 50000

A1: Switch the database type by modifying the parameters in the sqltranslate function:

When the parameter value is Oracle, translate the SQL in sqlstr into Oracle database syntax, and the result is:

```
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)
```

When the parameter value is mysql, translate the SQL in sqlstr into MySQL database syntax, and the result is:

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

A2: Execute the translated SQL in the specified database, for example, use the MySQL data source configured in the previous XML file, then set dsmysql here.

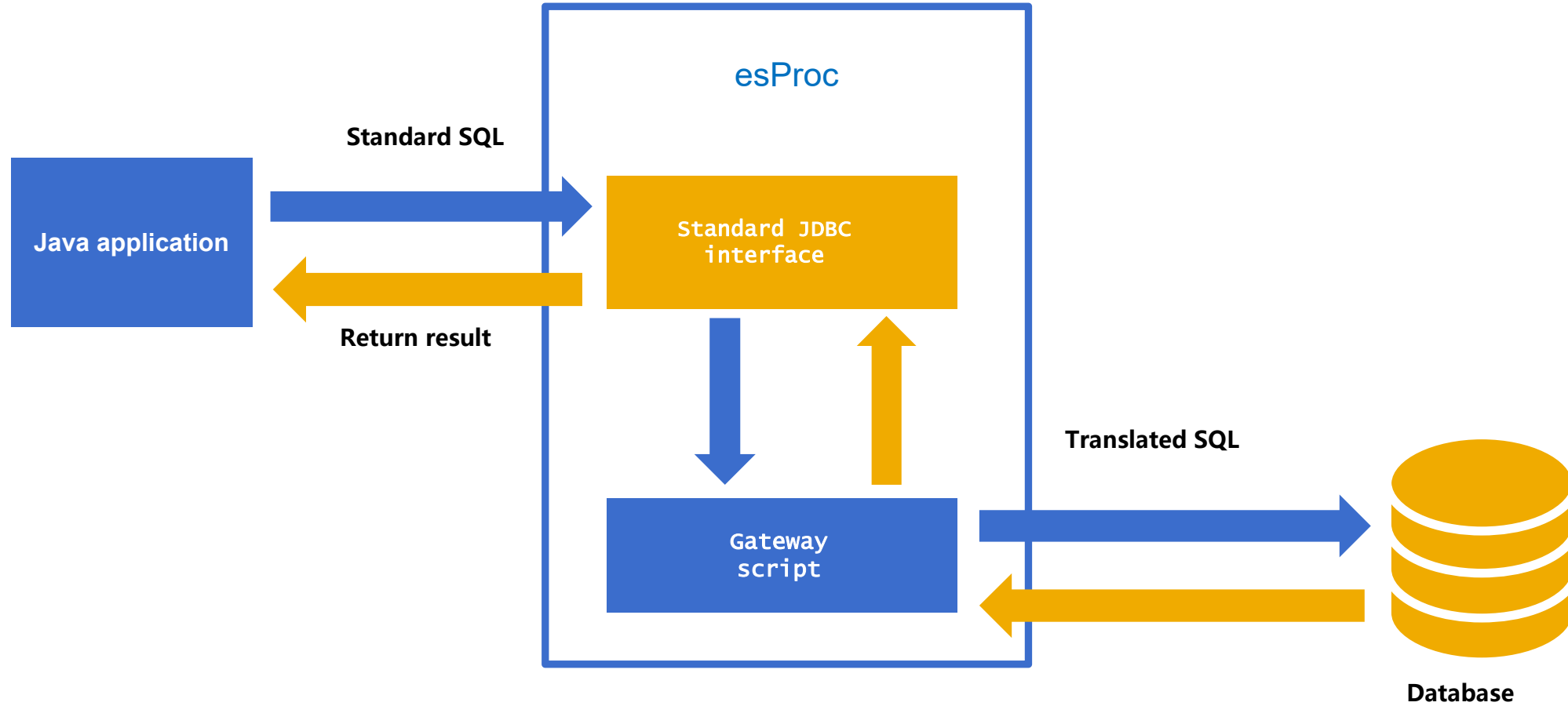
➤ Call DFX script file in Java code



The main code is as follows:

```
// Call the stored procedure through call, where sqltranslate is the file name of DFX, and "?" represents the parameter
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);
//Get result set
ResultSet rs = st.executeQuery();
```

➤ Using SQL directly with JDBC gateway



➤ Using SQL directly with JDBC gateway

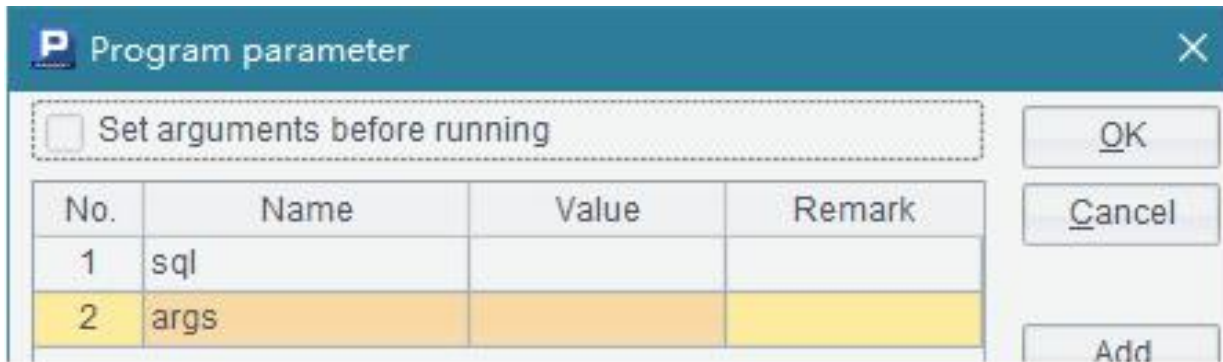


When calling esProc in JDBC mode, you can configure an SPL script as the JDBC gateway. All statements executed in JDBC will be processed and executed by this script first.

We can make use of this to let esProc JDBC receive SQL directly, translate SQL into the syntax of the target database in the gateway, execute it and return the result. In this way, the application uses SQL directly just like accessing the original database.

➤ Using SQL directly with JDBC gateway

1. Make a gateway file named executesql.dfx



There must be two parameters in the gateway file:

sql represents the SQL statement passed in when calling the gateway, **args** represents the parameter in the SQL statement

The gateway file script is as follows:

	A	
1	=sql.sqltranslate("MYSQL")	
2	=A1.split().pos@a("?").count()	//Count parameter numbers
3	=A2("args("+string(~)+")").concat@c()	//Spell out parameters
4	=if(args==null,"A1","A1,"+A3)	
5	=connect("dsMysql").query@x(\${A4})	
6	return A5	

➤ Using SQL directly with JDBC gateway



2. The configuration file `raqsoftconfig.xml` specifies the gateway file in the `<JDBC>` node `< gateway >`

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

For details, please refer to: <http://doc.raqsoft.com.cn/esproc/tutorial/pzraqsoftconfig.html>

➤ Using SQL directly with JDBC gateway



3. Passing parameters to gateway file in Java code

```
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) ");

// Setting parameters

st.setObject(1,10000); //Pass in the first parameter value in the SQL
st.setObject(2,50000); // Pass in the second parameter value in the SQL

st.execute(); // Execute stored procedure
```

➤ Use in reporting tools



Java report tool is a kind of Java application. It can also integrate esProc. Let's see how to implement SQL in the report tool to migrate to different types of databases.

1. Java application integrates esProc

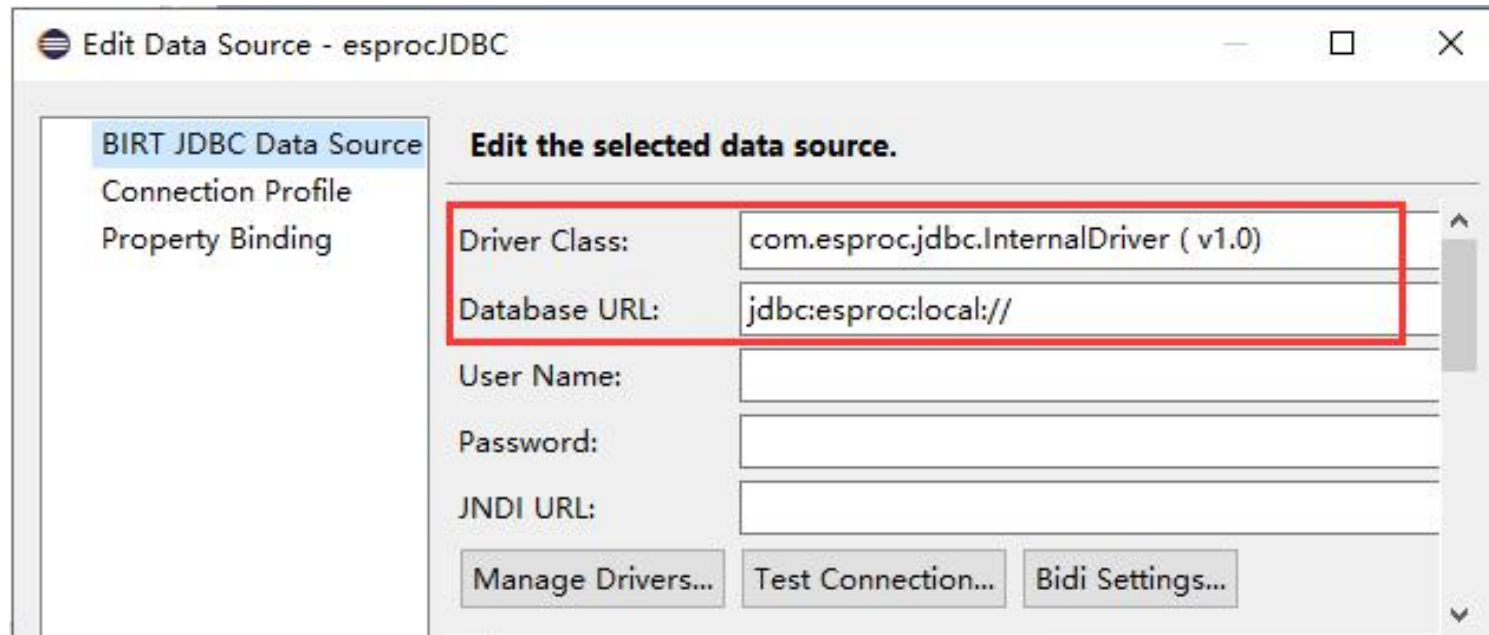
The method of report tool to deploy JDBC is the same as that of the common Java application. In Java application, deploy the esProc driver jar packages, and configure the running environment in raqsoftconfig.xml.

➤ Use in reporting tools



2. Report call esProc script

The DFX script file is still sqltranslate.dfx, which was made before. First, create a data source in the report IDE and point to esProc. Take BIRT report as an example:



➤ Use in reporting tools



Create new stored procedure dataset: select the data source of esProc and call the script file of esProc.

The 'New Data Set' dialog box is shown with the following fields and options:

- Data Source Selection:** A search box containing 'type filter text' and a list box showing 'JDBC Data Source' expanded with 'esprocJDBC' selected.
- Data Set Type:** A dropdown menu set to 'SQL Select Query'.
- Data Set Name:** A text box containing 'sqltranslate'.
- Navigation:** Buttons for '< Back', 'Next >', 'Finish', and 'Cancel'. The 'Next >' button is highlighted with a blue border.

The 'Edit Data Set - Data Set' dialog box is shown with the following fields and options:

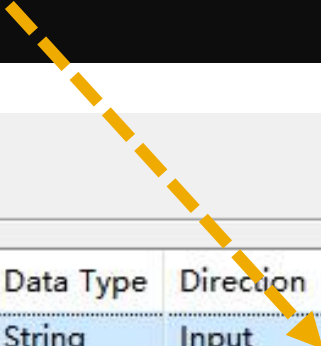
- Navigation:** A vertical list on the left with 'Query' selected. Other options include Data Source, Output Columns, Computed Columns, Parameters, Filters, Property Binding, Settings, and Preview Results.
- Available Items:** A list box containing 'esprocJDBC'.
- Query Text:** A text area containing the query: `1 {call sqltranslate (?,?,?)}`.
- Navigation:** Buttons for '<' and '>' on the left and right sides of the list boxes.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom right. The 'OK' button is highlighted with a blue border.

➤ Use in reporting tools



Set the parameter to standard SQL:

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



Parameters

	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

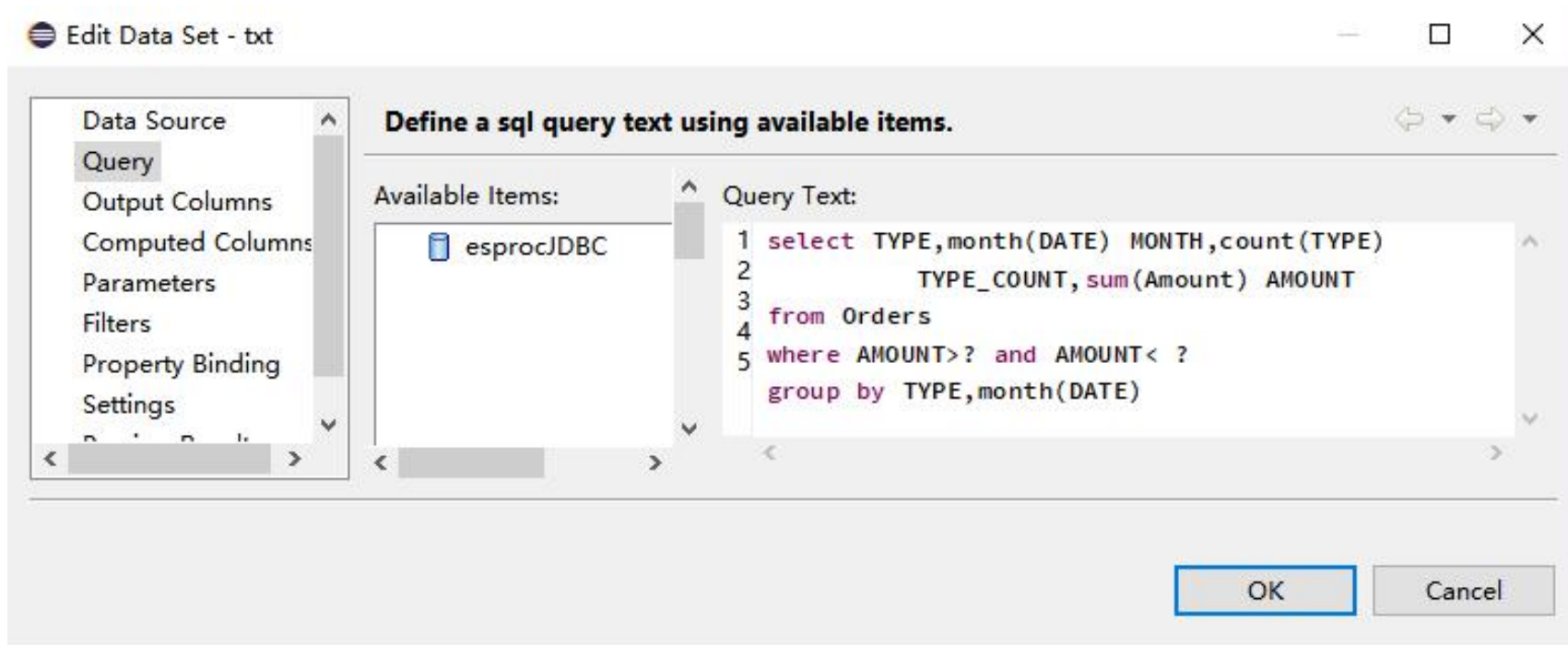
Buttons: New..., Edit..., Remove, Up, OK, Cancel

➤ Use in reporting tools



When used in reports, you can also use the way of JDBC gateway. Similarly, add the < gateway > node configuration gateway file in raqsoftconfig.xml and configure the data source.

The report dataset is configured as follows:



➤ Use in reporting tools



The parameter configuration is as follows:

Parameters

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

Buttons: New..., Edit..., Remove, Up, OK, Cancel

➤ Use in reporting tools



Design report and preview report, the usage of which is the same as ordinary dataset:

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

➤ Maintain database type



As for the SQL conversion function, the database types supported by default are as follows:

Database type	Number
ORACLE	1
SQLSVR	2
DB2	9
MYSQL	10
HSQL	13
TERADATA	14
POSTGRES	15

➤ Maintain database type



The database types supported by SQL translation can also be maintained manually, including the following two situations:

1. For the default database type, you can modify and add database functions
2. Manually add a new database type that does not exist by default

The above two operations can be configured in the function.xml file, which is located in the path /com/raqsoft/dm/sql/ in esproc-bin-xxxx.jar.

➤ Maintain database type



```
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">
```

Among them,

The name attribute indicates the standard function name; the paramcount attribute indicates the number of parameters in the function;

The dbtype attribute indicates the database type number;

The value attribute indicates the corresponding function writing method in the database. If value is empty, the default value is name.



➤ Maintain database type

When adding a database type, users only need to add an Info node in the function node of function.xml, and configure the dbtype and value attributes in the node. Dbtype needs to be set to 0, and value to be the corresponding function of the database.

When using the new database type in sqltranslate, the parameter should be written as **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")
	...