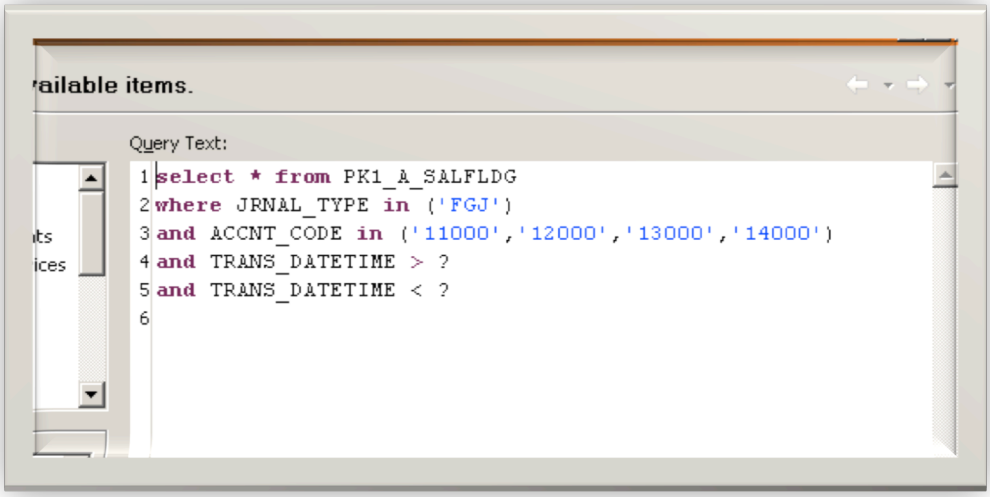
BIRT display or show the SQL query on the report design

Sometimes, when creating reports, it is useful, usually for debugging purposes to be able to see the SQL statement that is being executed against the database. In this tutorial we look at how we can output the SQL statement to the report itself, to view what is going on. In addition, our SQL statement uses parameters so we also look at how to include these in our output.

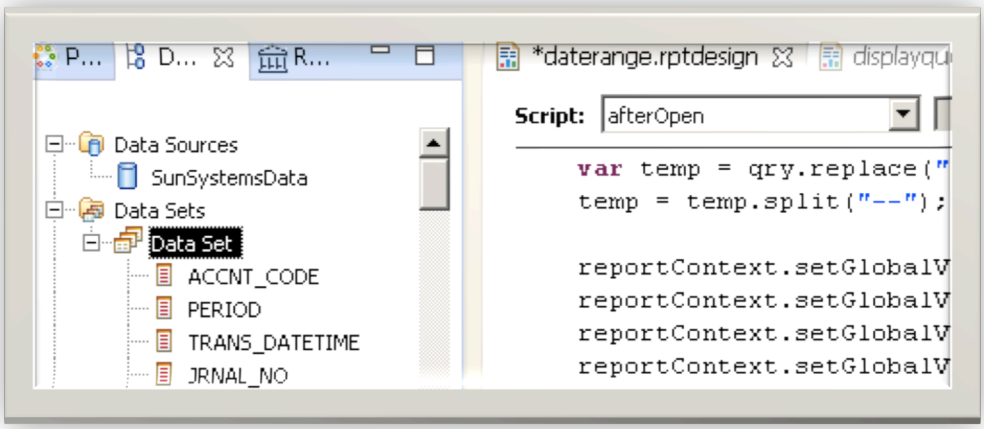
The SQL

Here is our SQL query. Notice the use of the two parameters:



The Script

In order to output this we need to add some script into the AfterOpen event in the scripting section for our data set.



Here is the full script:

```
importPackage(Packages.java.lang);
var qry = this.queryText;
var temp = qry.replace("?","--");
temp = temp.split("--");

reportContext.setGlobalVariable("gtemplen", temp.length);
reportContext.setGlobalVariable("gtemp1", temp[0]);
reportContext.setGlobalVariable("gtemp2", temp[1]);
reportContext.setGlobalVariable("gtemp3", temp[2]);

var newqry = "";
for(i=0; i<temp.length; i++){
      if(i==0){
            newqry = temp[i] +
this.getInputParameters().values().toArray()[i];
      }
      else{
            newqry = newqry + temp[i] +
this.getInputParameters().values().toArray()[i];
      }
}

reportContext.setGlobalVariable("myquery", newqry);
```

Lets break that down and see what it does

This first part Imports the java libraries and sets a variable “qry” equal to the querytext (i.e. the SQL statement) from the report

```
importPackage(Packages.java.lang);
var qry = this.queryText;
```

The next part replaces “?” (all the parameter place holders) with “--“ because we need to break the string down into the parts before and after each parameter and split doesn’t work with question marks.

The split then creates an array called temp containing each part of the split string.

```
var temp = qry.replace("?","--");
temp = temp.split("--");
```

The next lines are in here for learning purposes so we can see what is going on. These are global variables that we can put out on the report to show us the content of temp.length and the various parts of the strings generated by the split – i.e. the different parts of the array. There will be a different number of elements to the array depending on how many question marks there were in your original SQL statement. So you will have to add an extra gtempN line for each one. We could have put this into a loop, but decided to keep it simple in this example.

```
reportContext.setGlobalVariable("gtemplen", temp.length);
reportContext.setGlobalVariable("gtemp1", temp[0]);
reportContext.setGlobalVariable("gtemp2", temp[1]);
reportContext.setGlobalVariable("gtemp3", temp[2]);
```

Next we sets a variable called newquery to blank and start a loop for each part of the temp array.

For the first part of the temp array (i==0) we set the newquery variable equal to the first part of the temp array (temp[i]) concatenated to the first input parameter.

The next part triggers when dealing with the rest of the array elements (i.e. when I is not equal to zero), setting newquery equal to the previous value for newquery concatenated with the next parameter.

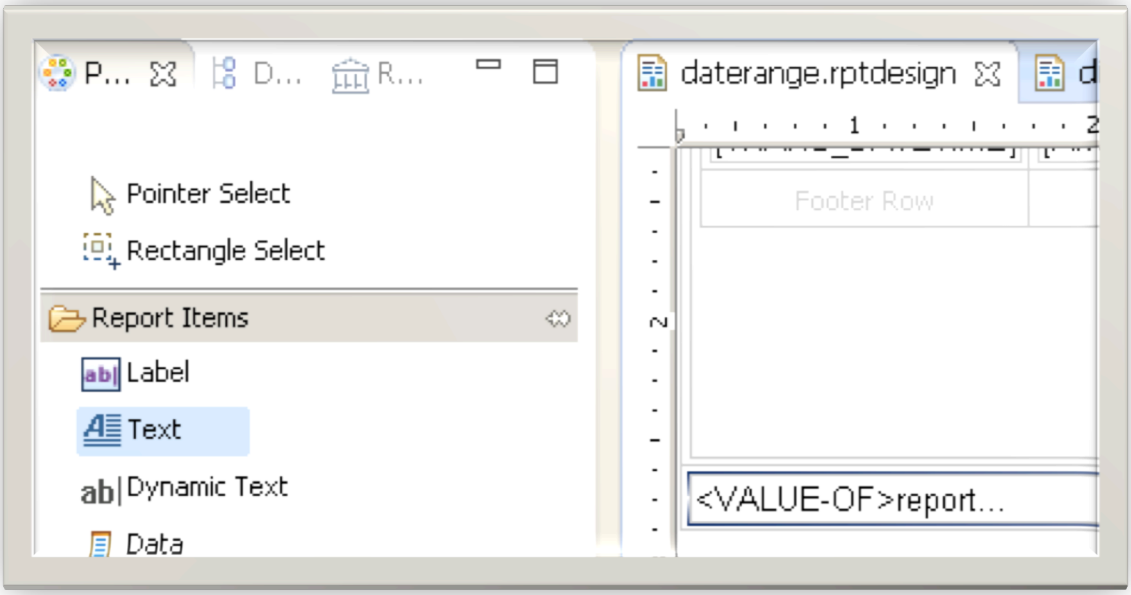
```
var newqry = "";
for(i=0; i<temp.length; i++){
      if(i==0){
            newqry = temp[i] +
this.getInputParameters().values().toArray()[i];
      }
      else{
            newqry = newqry + temp[i] +
this.getInputParameters().values().toArray()[i];
      }
}
```

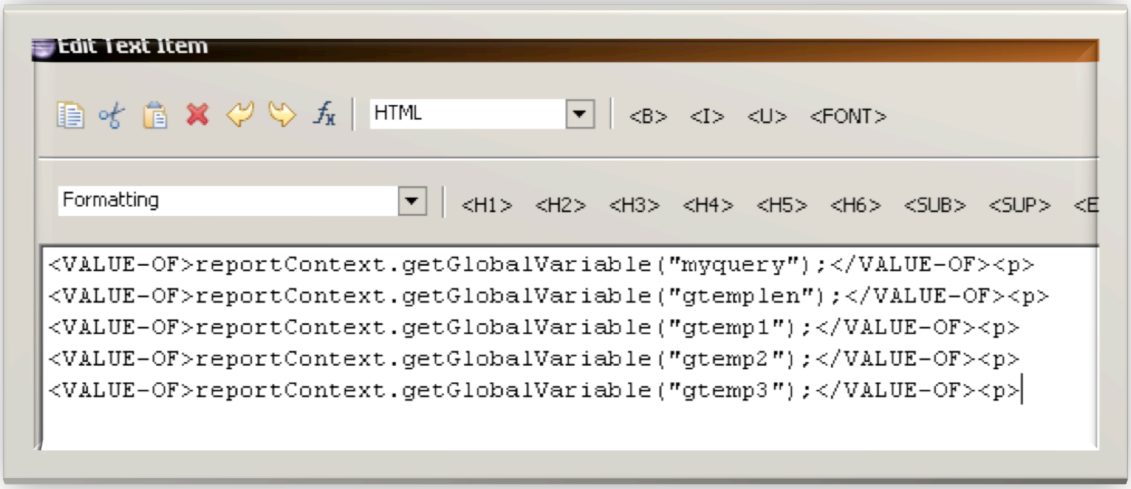
Finally we add all of the newquery variable into a global variable “myquery”, for output.

```
reportContext.setGlobalVariable("myquery", newqry);
```

The Output

When it comes to the output stage, in the report we can have a text field, containing the output from each of the global arrays





The Result

On our report we can see the various global variables, containing the SQL being displayed: