



Concepts and Technologies for ERP Systems

Exercise 3

SS 2003

Database Access in SAP R/3: ABAP Programming with Open SQL and Native SQL

Introduction

This practical exercise continues the previous one where the data model for our banking application has been implemented. In particular, the tables and views that have been defined in exercise 2 are needed here. The goal of this practical exercise is to extend the banking application by producing reports on the account data by means of ABAP programs with embedded Open SQL and Native SQL statements, respectively.

Preparation (only needed when Exercise 2 has not been completed)

All those who do not have finished exercise 2 yet have first to create view ZHPIxxCUST and tables ZHPIxxACC and ZHPIxxENTR. To this end, the following steps are required (as a shortcut to what should have been done already in the previous exercise):

1. **Copy view View** ZHPI00CUST:

- Go to the Data Dictionary (Tools → ABAP Workbench, Development → Dictionary)
- Mark Views
- Specify ZHPI00CUST as object name (name next to the view check box)

- Copy this view (Dictionary Object → Copy)
- Specify the new name according to your name space ZHPIxxCUST

2. Copy tables ZHPI00ACC and ZHPI00ENTR:

- Mark Tables in the Data Dictionary
- Copy tables ZHPI00ACC and ZHPI00ENTR to your name space (Dictionary Object → Copy)

3. Fill these tables with data:

- Activate all objects (the view and the two tables).
- Go to the ABAP editor (Tools → ABAP Workbench, Development → ABAP Editor)
- Specify the name of the ABAP program ZBANKCOPY
- Execute this program (F8) and specify the name of your account table and entry table.

After having completed these additional steps, everything is prepared for this practical exercise.

Question 1: General Account Survey (Open SQL)

Copy program ZHPI00ReportOpenSQL which defines a report on customer, account, and entry data into your name space (i.e., to ZHPIxxReportOpenSQL). In analogy to the previous exercise, you should store all new objects in your development class ZHPIxx. Programs in the ABAP editor can be found at Tools → ABAP Workbench → Development → ABAP Editor or with transactionID SE38, respectively¹. You can add this transactions to your list of favorites (Development).

The report shall list all customers and for each customer, the corresponding accounts and the entry items to these accounts. For this purpose, Open SQL queries are needed that access the corresponding tables of the underlying database via the database interface layer. You have to adjust the copied program to your tables and to extend the (incomplete) Open SQL statements. Do also complete the ABAP statements needed for outputting the data read from the database. After completion, execute the report.

Note: After completion, you have to activate the report. In order to make sure that the program does not contain any syntax errors, you can check it via Program → Check → Syntax. If no syntax errors are detected, you can start your report either with F8 or via Program → Test.

¹By indicating /nTransactionID in the command field (in the button bar next to the ✓ button), you can directly access a transaction within SAP from an arbitrary other transaction. The ABAP editor can be started with /nSE38. When using /oTransactionID, a transaction is launched in a new window.

Question 2: Account Balance Query (Native SQL)

Once having a general overview over all accounts of your bank, you want to provide an additional report for your customers and for the employees at the counter. This report shall be given an account number as input parameter and shall output the name of the account holder and the current account balance. For this program, you plan to use Native SQL. This means that the SQL statement is directly forwarded to the underlying database system without considering the database interface layer. The particular database system and therefore also the particular SQL dialect that has to be used for our SAP R/3 installation can be found at `System → Status...`

A skeleton is also available for this program (`ZHPI00ReportNativeSQL`). Copy this report into your name space (`ZHPIxxReportNativeSQL`) and complete it. The database query should be implemented by a single SQL statement.

When implementing this program, you should address the following questions:

- i.) Are there any schema objects that cannot be used? If so, which ones and why?
- ii.) How can customer data (`ZHPIxxCUST`) be accessed?

How do you evaluate the system support in contrast to Open SQL as you have experienced in the previous question?

Question 3: Account Balance Query (SAP View)

Since you do not want to adapt your report in case your administrator decides to change the underlying database, you plan to produce another version of the account balance query report using a database view. Obviously, the query for this report requires a join between the customer and the account table. Go to the ABAP dictionary and create a database view `ZHPIxxJOIN` for this join. In this view, you have to define the base tables, the join conditions (these are the same as in the Native SQL statement of question 2) and the fields of this view. Generate and activate view `ZHPIxxJOIN`.

Now, you have to use this view in your report. Go to the ABAP editor and copy report `ZHPIxxReportNativeSQL` to `ZHPIxxReportView` (`Program → Copy` and check `Source` and `Text` elements). Change your Native SQL query to an Open SQL query that uses the newly defined view `ZHPIxxJOIN`.

Optional: You can additionally create a further implementation of this report by explicitly embedding the join into the Open SQL statement (use the report name `ZHPIxxReportOpenJoin` for this purpose; just copy `ZHPIxxReportView` to `ZHPIxxReportOpenJoin`, and insert the join condition into the Open SQL statement instead of using view `ZHPIxxJOIN`).

Question 4: Management Summary (Open SQL or Native SQL)

Since you are an experienced ABAP programmer in the meanwhile, the management of your bank asks you to provide an additional report for management summaries. This report has to list the name, the number of accounts and the sum of all account balances for each customer. Create a report named `ZHPIxxManagement`. You can choose whether to use open or native

SQL for data access (you should evaluate what is more appropriate here; by doing this, it is important to consider also the convenience and system support for both variants when making a choice).

This report should be of type **Executable Program**, should have status **Customer production program** and should belong to application **Financial accounting**.

Submission of results: Submission of results is not mandatory. However, you are invited to submit your results by email to heiko.schuldt@hpi.uni-potsdam.de indicating: “ERP: Exercise 3” in the subject line. This email should contain your observations from question 2 and your SAP R/3 user ID (the programs can then be accessed via your development class).