Preface

I am known to the system as DEVELOP30. Therefore, I named my development class ZHPI30, etc.

Question 1: General Account Survey (Open SQL)

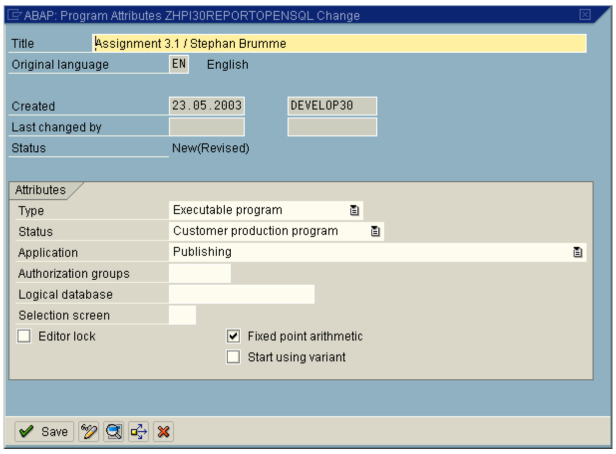
Copy program ZHPI00ReportOpenSQL which defines a report on customer, account, and entry data into your name space (i.e., to ZHPIxxReportOpenSQL). In analogy to the previous exercise, you should store all new objects in your development class ZHPIxx. Programs in the ABAP editor can be found at Tools → 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.

The given source code missed a few important statements. Neither the syntax nor the semantics are complete. Nevertheless, I present the original lines:

```
*&---------------------------------------------------------------------*
*& Report  ZHPIxxREPORTOPENSQL                                          *
*&                                                                     *
*&---------------------------------------------------------------------*
*& Outputs customers, their accounts and their entries                 *
*& Exercise 3                                                          *
*& Skeleton that has to be extended/completed                          *
*& using Open SQL                                                      *
*&---------------------------------------------------------------------*

REPORT  ZHPIxxREPORTOPENSQL .


TABLES: ZHPICUST,                                           .
   " Declaration of tables/views needed
   " Have to be adopted to own table names

SELECT *                                                    .
   " (tuple-wise) selection of all customers
   " from own customer view zhpixxcust
   ULINE.
   WRITE: / 'Customer: ',                   , ZHPICUST-NAME1 COLOR 3.
      " output of attributes Kunnr and Name1 per customer



   SELECT
        WHERE KUNNR =  ZHPIxxCUST-                   .
        " nested (tuple-wise) selection of all associated accounts
        " from own account table
        " link between customer view and account table
        " via shared (foreign) key
        " Open-SQL additionally uses the clientID (mandt)
        " transparently

      If              .       " do the following only when tuples exist
         WRITE: / '                   AccountNo:',            ,
                '        Balance: ',                          .
            " Outputs account number and balance.

           SELECT

                                                              .
```

```
                         " nested selection of the associated account entries
                         " link via foreign keys

             WRITE: / '                             ' ,
                    '                           Entries:',
                    '         ', ZHPIxxENTR-AMOUNT.
                    " output of entry amounts.

                ENDSELECT.  "  inner select statement
             endif.
             ENDSELECT.     "  middle select statement
          SKIP.             "  new line after each customer
          ENDSELECT.        "  outer select statement
```

The modified version is a “Executable program”, has the status “Customer production program” and its application domain is “Publishing”. All changes are highlighted blue.


ABAP: Program Attributes ZHPI30REPORTOPENSQL Change

Title	Assignment 3.1 / Stephan Brumme	
Original language	EN	English
Created	23.05.2003	DEVELOP30
Last changed by		
Status	New(Revised)	

Attributes

Type	Executable program
Status	Customer production program
Application	Publishing
Authorization groups	
Logical database	
Selection screen	

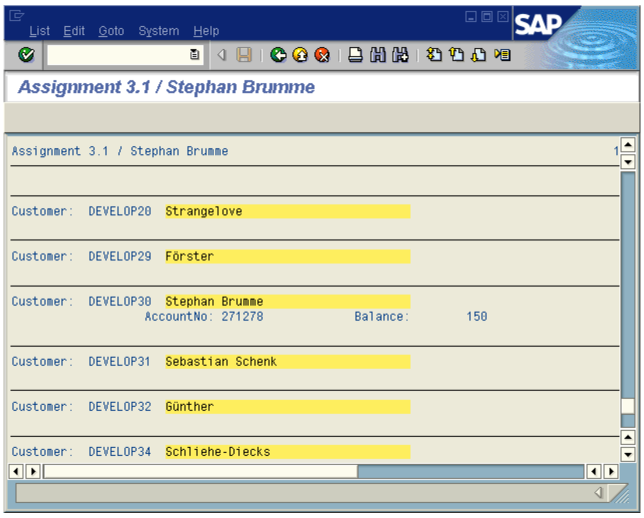
☐ Editor lock
☑ Fixed point arithmetic
☐ Start using variant

Save

```
  REPORT ZHPI30REPORTOPENSQL .

  *&---------------------------------------------------------------------*
  *& Report  ZHPI30REPORTOPENSQL                                         *
  *&                                                                     *
  *&---------------------------------------------------------------------*
  *& Outputs customers, their accounts and their entries                 *
  *& Exercise 3                                                          *
  *& Skeleton that has to be extended/completed                          *
  *& using Open SQL                                                      *
  *&---------------------------------------------------------------------*

  TABLES: ZHPI30CUST, ZHPI30ACC, ZHPI30ENTR                   .
     " Declaration of tables/views needed
     " Have to be adopted to own table names

  SELECT * FROM ZHPI30CUST
     .
     " (tuple-wise) selection of all customers
     " from own customer view zhpixxcust
     ULINE.
     WRITE: / 'Customer: ', ZHPI30CUST-KUNNR, ZHPI30CUST-NAME1 COLOR 3.
        " output of attributes Kunnr and Name1 per customer



     SELECT * FROM ZHPI30ACC
          WHERE KUNNR = ZHPI30CUST-KUNNR                  .
          " nested (tuple-wise) selection of all associated accounts
          " from own account table
          " link between customer view and account table
          " via shared (foreign) key
          " Open-SQL additionally uses the clientID (mandt)
          " transparently

        If  SY-SUBRC = 0   .  " do the following only when tuples exist
```

```
   WRITE: / '                   AccountNo:', ZHPI30ACC-ACCTNO,
          '        Balance: ', ZHPI30ACC-BALANCE              .
      " Outputs account number and balance.

     SELECT * FROM ZHPI30ENTR
     WHERE KUNNR  = ZHPI30ACC-KUNNR AND
           ACCTNO = ZHPI30ACC-ACCTNO
                                                         .
            " nested selection of the associated account entries
            " link via foreign keys

 WRITE: / '                              ' ,
        '                           Entries:',
        '        ', ZHPI30ENTR-AMOUNT.
        " output of entry amounts.

      ENDSELECT.  "  inner select statement
   endif.
   ENDSELECT.     "  middle select statement
SKIP.             "  new line after each customer
ENDSELECT.        "  outer select statement
```

The snippet actually works as proved by this astonishing convincing screenshot:



```
Assignment 3.1 / Stephan Brumme                                          1

Customer:  DEVELOP28  Strangelove

Customer:  DEVELOP29  Förster

Customer:  DEVELOP30  Stephan Brumme
                    AccountNo: 271278            Balance:        150

Customer:  DEVELOP31  Sebastian Schenk

Customer:  DEVELOP32  Günther

Customer:  DEVELOP34  Schliehe-Diecks
```

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 ?

The skeleton turned out to be fairly incomplete:

```
*&---------------------------------------------------------------------*
*& Report  ZHPIxxREPORTNATIVESQL                                       *
*&                                                                     *
*&---------------------------------------------------------------------*
*& account balance query                                               *
*& Exercise 3                                                          *
*& Adapt/extend the given skeleton                                     *
*& using Native SQL                                                    *
*&---------------------------------------------------------------------*

REPORT  ZHPIxxREPORTNATIVESQL .

        " Declaration of variables that are to be selected
        " (Host variables in Native SQL)
DATA:   KNR LIKE         ,
        NAME             ,
        ACCT_BAL               .

        " Declaration of input parameter acct_no
PARAMETER: ACCT_NO                           .

        " Selection of customer number, customer name (name1)
        " and account balance
        " Native SQL statement that is sumbitted directly to the DBMS
        " should be terminated neither with a semicolon ;
        " nor with a full stop . like ABAP statements
EXEC SQL.

  INTO :KNR,



ENDEXEC.

" Formatting and output of query results
   ULINE.
```

There were some issues I had to solve: first of all, ZHPI30CUST is not a real database table but a view that only exists in the SAP system. The underlying database does not have any information about it, however, I as the developer do have: accessing KNA1 instead of ZHPI30CUST fixed that problem. Furthermore, I faced a miraculous issue: there is only a single SQL statement allowed per EXEC-SQL-block. So splitting up the three statements in three blocks resolved that problem, too.

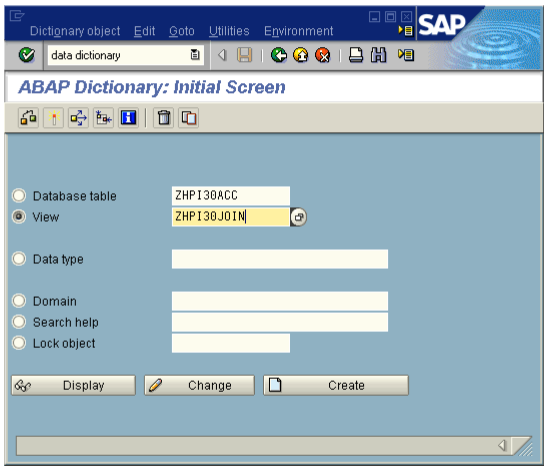
```
*&---------------------------------------------------------------------*
*& Report  ZHPI30REPORTNATIVESQL                                       *
*&                                                                     *
*&---------------------------------------------------------------------*
*& account balance query                                               *
*& Exercise 3                                                          *
*& Adapt/extend the given skeleton                                     *
*& using Native SQL                                                    *
*&---------------------------------------------------------------------*

REPORT  ZHPI30REPORTNATIVESQL .

TABLES: ZHPI30ACC, ZHPI30CUST, ZHPI30ENTR.

        " Declaration of variables that are to be selected
        " (Host variables in Native SQL)
DATA:   KNR LIKE ZHPI30CUST-KUNNR,
        NAME LIKE ZHPI30CUST-NAME1,
        ACCT_BAL LIKE ZHPI30ACC-BALANCE  .

        " Declaration of input parameter acct_no
PARAMETER: ACCT_NO LIKE ZHPI30ACC-ACCTNO                          .

        " Selection of customer number, customer name (name1)
        " and account balance
        " Native SQL statement that is sumbitted directly to the DBMS
        " should be terminated neither with a semicolon ;
        " nor with a full stop . like ABAP statements

" Get balance
EXEC SQL.
  SELECT ZHPI30ACC.BALANCE INTO :ACCT_BAL
  FROM ZHPI30ACC
  WHERE ZHPI30ACC.ACCTNO = :ACCT_NO
ENDEXEC.

" Determine customer's ID
EXEC SQL.
  SELECT ZHPI30ACC.KUNNR INTO :KNR
  FROM ZHPI30ACC
  WHERE ZHPI30ACC.ACCTNO = :ACCT_NO
ENDEXEC.

" Retrieve customer's name
EXEC SQL.
  SELECT KNA1.NAME1 INTO :NAME
  FROM KNA1, ZHPI30ACC
  WHERE ZHPI30ACC.ACCTNO = :ACCT_NO
    AND KNA1.KUNNR = ZHPI30ACC.KUNNR
ENDEXEC.


" Formatting and output of query results
WRITE: / KNR, '  ', NAME, '  ', ACCT_BAL.
   ULINE.
```

Question 3: Account Balance Query (SAP View)

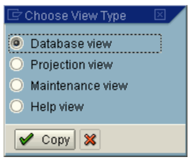
Since you do not want to adapt your report in case your administrator decides to change the underlying database, you plan to produce another version of the account balance query report using a database view. Obviously, the query for this report requires a join between the customer and the account table. 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).

Setting up a new SAP database view can be performed with a just few mouse clicks.

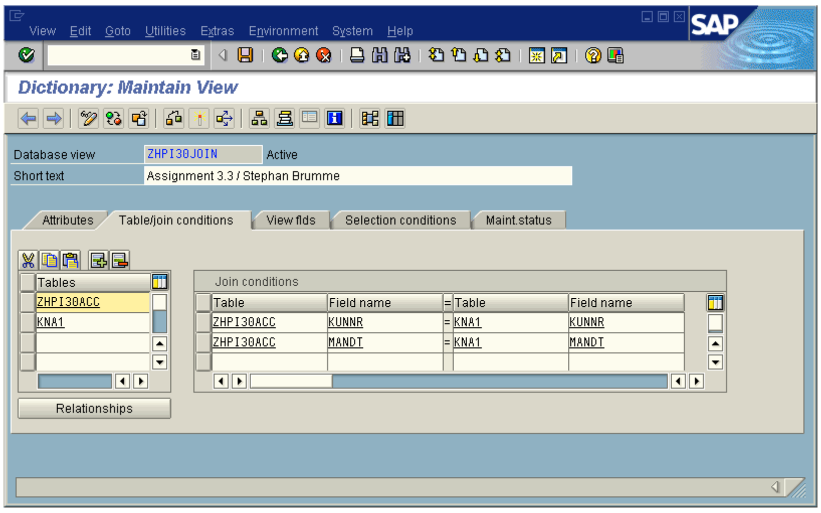

ABAP Dictionary: Initial Screen

Database table	ZHPI30ACC
[✓ View]	ZHPI30JOIN
Data type	
Domain	
Search help	
Lock object	


Choose View Type

[✓ Database view]
Projection view
Maintenance view
Help view

The view conencts ZHPI30ACC and KNA1. The latter table must be used instead of ZHPI30CUST because ZHPI30CUST is a view and one cannot create a view on views.


Dictionary: Maintain View

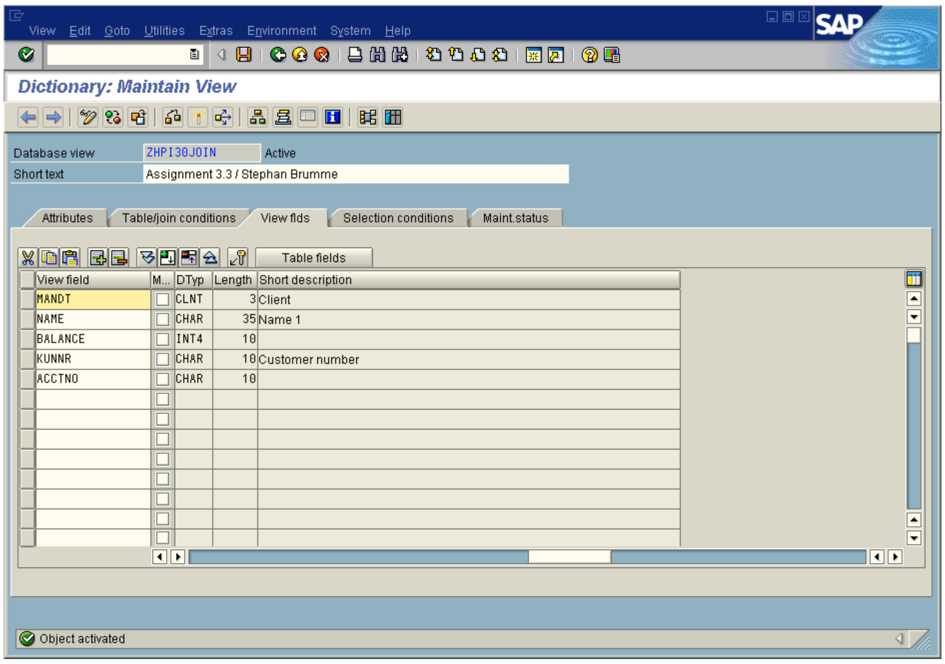
Database view ZHPI30JOIN Active
Short text Assignment 3.3 / Stephan Brumme

Tables
ZHPI30ACC
KNA1

Join conditions

Table	Field name	=	Table	Field name
ZHPI30ACC	KUNNR	=	KNA1	KUNNR
ZHPI30ACC	MANDT	=	KNA1	MANDT

The fields can be obtained from the following screenshot:



Dictionary: Maintain View

Database view	ZHPI30JOIN Active
Short text	Assignment 3.3 / Stephan Brumme

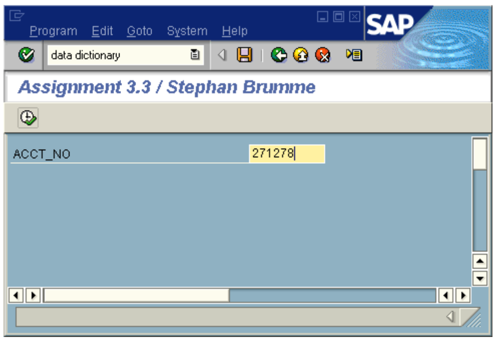
View field	M...	DTyp	Length	Short description
MANDT	☐	CLNT	3	Client
NAME	☐	CHAR	35	Name 1
BALANCE	☐	INT4	10	
KUNNR	☐	CHAR	10	Customer number
ACCTNO	☐	CHAR	10	

Object activated

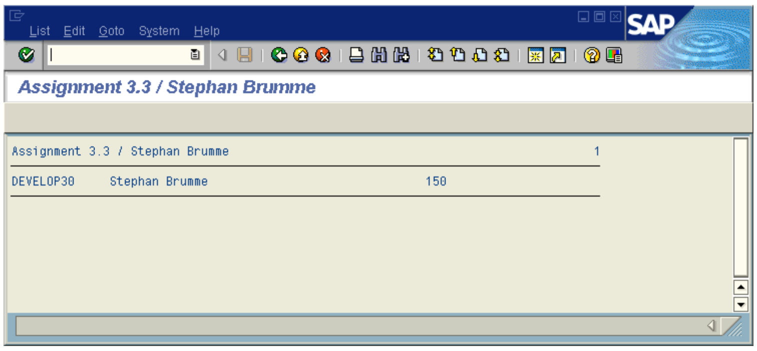
The task requires these OpenSQL statements:

```
*&---------------------------------------------------------------------*
*& Report  ZHPI30REPORTVIEW                                            *
*&                                                                     *
*&---------------------------------------------------------------------*
*& account balance query                                               *
*& Exercise 3                                                          *
*&---------------------------------------------------------------------*

REPORT  ZHPI30REPORTVIEW .

        " Declaration of variables that are to be selected
        " (Host variables in Native SQL)
DATA:   KNR LIKE ZHPI30JOIN-KUNNR,
        NAME LIKE ZHPI30JOIN-NAME,
        ACCT_BAL LIKE ZHPI30JOIN-BALANCE.

        " Declaration of input parameter acct_no
PARAMETER: ACCT_NO LIKE ZHPI30JOIN-ACCTNO.

        " Selection of customer number, customer name (name1)
        " and account balance
        " Native SQL statement that is sumbitted directly to the DBMS
        " should be terminated neither with a semicolon ;
        " nor with a full stop . like ABAP statements

TABLES: ZHPI30JOIN.

" Get balance, kunnr and name all at once
SELECT BALANCE KUNNR NAME
    INTO (ACCT_BAL, KNR, NAME)
  FROM ZHPI30JOIN
  WHERE ACCTNO = :ACCT_NO.
ENDSELECT .

" Formatting and output of query results
WRITE: / KNR, '  ', NAME, '  ', ACCT_BAL.
   ULINE.
```

The program found – when looking for account 271278 – my own account (as expected). Both the name and the balance are accurate.


Program Edit Goto System Help
data dictionary
SAP
Assignment 3.3 / Stephan Brumme

ACCT_NO 271278



```
Assignment 3.3 / Stephan Brumme                                  1
DEVELOP30     Stephan Brumme                       150
```

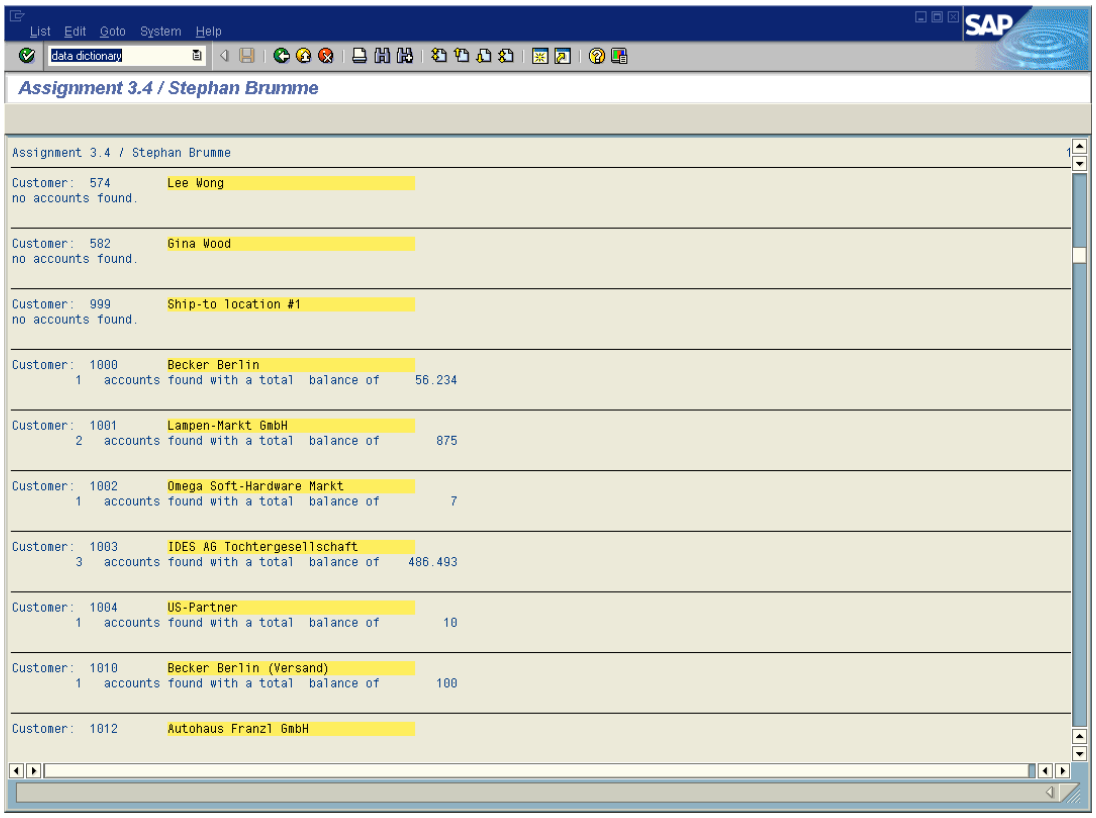
Question 4: Management Summary (Open SQL or Native SQL)

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.

OpenSQL “feels” more suitable to me: its superior integration in the SAP environment (syntax check etc.) and the portability feature truly convinced me. I do not need any of the advanced SQL commands and thus all algorithms can be implemented with plain OpenSQL.

```
*&---------------------------------------------------------------------*
*& Report  ZHPI30MANAGEMENT                                            *
*&                                                                     *
*&---------------------------------------------------------------------*
*& Exercise 4                                                          *
*&---------------------------------------------------------------------*

REPORT  ZHPI30MANAGEMENT.

TABLES: ZHPI30CUST, ZHPI30ACC.

DATA:   NO_OF_ACC TYPE I,
        TOTAL_BALANCE TYPE ZHPIBAL.

" Get balance, kunnr and name all at once
SELECT * FROM ZHPI30CUST
   .
   " (tuple-wise) selection of all customers
   " from own customer view zhpixxcust
   ULINE.
   WRITE: / 'Customer: ', ZHPI30CUST-KUNNR, ZHPI30CUST-NAME1 COLOR 3.
      " output of attributes Kunnr and Name1 per customer



   SELECT COUNT( * ) SUM( BALANCE )
      FROM ZHPI30ACC
      INTO (NO_OF_ACC, TOTAL_BALANCE)
      WHERE KUNNR = ZHPI30CUST-KUNNR .

      If  NO_OF_ACC = 0 .
          WRITE: / 'no accounts found.' .
      endif.

      If  NO_OF_ACC > 0 .
             WRITE: / NO_OF_ACC, ' accounts found with a total',
                                 ' balance of ', TOTAL_BALANCE .
      endif.

   SKIP.            "  new line after each customer
ENDSELECT.       "  outer select statement
```

Some output:



```
Assignment 3.4 / Stephan Brumme

Customer:  574        Lee Wong
no accounts found.

Customer:  582        Gina Wood
no accounts found.

Customer:  999        Ship-to location #1
no accounts found.

Customer:  1000       Becker Berlin
         1   accounts found with a total  balance of     56.234

Customer:  1001       Lampen-Markt GmbH
         2   accounts found with a total  balance of        875

Customer:  1002       Omega Soft-Hardware Markt
         1   accounts found with a total  balance of          7

Customer:  1003       IDES AG Tochtergesellschaft
         3   accounts found with a total  balance of    486.493

Customer:  1004       US-Partner
         1   accounts found with a total  balance of         10

Customer:  1010       Becker Berlin (Versand)
         1   accounts found with a total  balance of        100

Customer:  1012       Autohaus Franzl GmbH
```