



Concepts and Technologies for ERP Systems

Exercise 2

SS 2003

SAP R/3: Data Model, Metadata Management, and Customizing

In the first exercise, you have already become acquainted with the SAP R/3 system. Now, we will start with implementing our banking services in SAP R/3. The goal of this exercise is to define the data model needed for these extensions and to define the meta data (SAP tables, attributes, etc.).

Question 1: Data Model & Metadata

a) Data Model

Display the data model “Business Partner” FI___10159 in the Data Modeler (this is the data model we have already analyzed in the last exercise; note that it has three underscores in its name).

- i.) Which relationship (in SAP’s SERM terminology) is defined between the entity types “*Business Partner*” and “*Customer*” (ID 12014)?
- ii.) Which relationship is defined between the entity types “*House Bank*” and “*Company Code - House Bank*”? What are the cardinalities of this relationship?

b) Metadata Management (Dictionary)

Analyze table KNA1 (we have already found out in the last exercise that this table contains the customers).

- i.) How many attributes does KNA1 have?
- ii.) How is this table mapped to table(s) of the underlying database system?
- iii.) What are the names of the key attributes of this table?
- iv.) To how many other tables do foreign key relationships exist?

Register yourself as Developer

Each developer who intends to create objects in SAP R/3 has to be registered and provided with a developer key that is generated by SAP in Walldorf. The first time a new object is stored, the system requests this developer key (note that this does not apply when operational data is stored). We have already started the registration process for all your developxx accounts. A list with all developer keys can be found on the accompanying pages of this lecture in the internal area of the HPI web site at <http://www.hpi.uni-potsdam.de/intern.new/>. You can get the key to your SAP account from this list; it will be needed for the next question.

Question 2: Extension of the SAP Data Model

You realize, after having analyzed the system, that the banking services your company wants to provide, are not adequately supported by the system (at least, we assume that this is the case within our series of exercises). Hence, you have to extend your SAP R/3 system.

The first step is to create a model of your banking application. This model is given in Figure 1 (in the well-known ER notation, not in SAP SERM). It consists of three entities: ZHPIxxCUST, ZHPIxxACC, and ZHPIxxENTR. In what follows, xx has to be replaced by the number of your SAP account. Note that cardinalities are given in the ERM-style which differs from the UML notation.

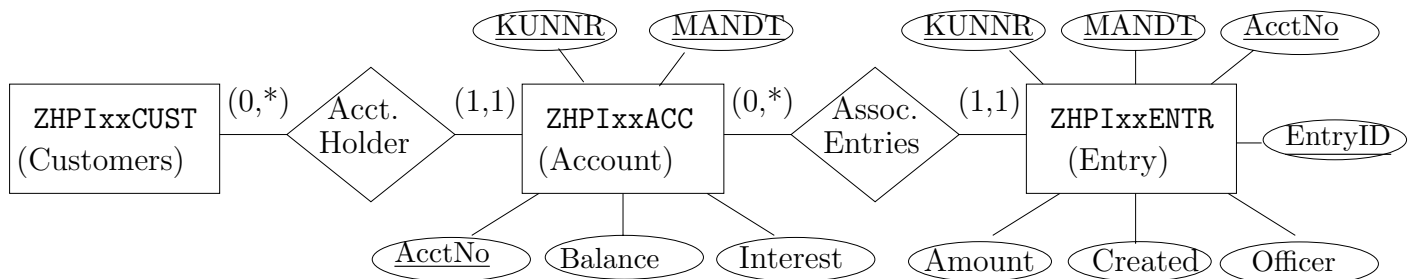


Abbildung 1: ER Model of your Banking Application

a) Create Data Model (Data Modeler)

The first step to your individual SAP R/3 extension is the creation of a new data model ZHPIxxMOD in the Data Modeler.

Hint: Create the new data model ZHPIxxMOD and use within this data model for the three entities the names given above in the ER model, i.e., ZHPI18CUST for user develop18. Just create the entities – without further specifying the attributes. Consider the particular structural restrictions of SAP SERM when arranging the entities. Do also specify the relationships between the entities (Edit → Connect) and their cardinalities.

Important Hint: Store all objects you are newly creating in your own development class ZHPIxx. A development class is a container for logically corresponding objects and strongly facilitates the access to all your objects. To each development class, a workbench request has to be specified. The first time you place a new object in your development class, you have to create such a request. From this point, you can re-use this request for all further objects you insert in the development class.

b) Specification of Entities (Dictionary)

- i.) Obviously, customer data of the original SAP R/3 system should be re-used in your banking application. Hence, create a mapping from the entity type ZHPIxxCUST to a SAP view (in the graphical mode of the Data Modeler: choose entity and click on Dictionary). Define a projection view over table KNA1 for this purpose and name this view ZHPIxxCUST. At this point, you will need your developer key (just copy and paste from the list of developer keys). Choose some (few) attributes of KNA1 you want to include in this view, and make sure that the primary keys MANDT and KUNNR are also within this selection. Save and activate this view.

- ii.) Create a mapping from entity type ZHPIxxACC to a table and define this table according to the ER model of Figure 1.

Hint: Do not use the identifier proposed by the system for this table. Rather, use the name of the corresponding entity type (i.e., ZHPIxxACC).

- iii.) Define the attributes of this table.

Hint: The data elements for MANDT and KUNNR can be taken from the view ZHPIxxCUST (MANDT and KUNNR) you have just created. The data elements for the remaining attributes are already given (ZHPIACCTNO for the account number, ZHPIBAL for both balance and interest). In order to facilitate the import of data, you should stick to this order when defining the attributes.

- iv.) Specify the primary keys given in the ER model for table ZHPIxxACC.
- v.) Specify the foreign keys MANDT and KUNNR, which are dependent from the corresponding attributes of view ZHPIxxCUST.
- vi.) Save and activate (Table → Activate) your account table¹. Via (Table → Check) you can figure out whether your table definition is consistent.

Hint: When creating a table, make sure that you allow table maintenance (otherwise, data can only be modified by ABAP programs but not via the Data Browser). Indicate, in the technical settings, delivery class A and choose data class APPL0.

- vii.) Do the same for the account entries table (ZHPIxxENTR).

Hint: Again, do not use the suggested name but use ZHPIxxENTR. The data elements for MANDT, KUNNR, and ACCTNO can be taken from ZHPIxxACC. The

¹The activation of a newly defined table is very important since only after this, the table is physically created in the underlying database system. The same is true with changes to the schema of a table; these will also take effect after activation.

data elements for the remaining attributes are also predefined (ZHPIENTR for EntryID, ZHPIAMOUNT for amount, ZHPIDATE for Created, and ZHPIOFF for Officer). Create these attributes in this order.

- viii.) Specify the foreign key relationships implicitly included in the model for ZHPIxxENTR. Do also specify the primary keys.
 - ix.) Save and activate the account entry table.
- c) Fill the new Tables with Data (Execution of an ABAP Program)
- Usually, when introducing SAP R/3, legacy applications are replaced and data from these applications has to be loaded into SAP R/3. Similarly, we assume that the required data already exists (but in some other table). Hence, you have to copy this data into your new tables. A nice colleague of yours has already implemented such an ABAP program named ZBANKCOPY. Take a look at this program in the ABAP editor (you do not have to modify anything here; we will start with ABAP programming in the next exercise) and execute the program (just press F8). You have just to indicate the name of your newly generated tables: specify ZHPIxxACC as “ACC_TAB” and ZHPIxxENTR as “ENTR_TAB”.
- d) Extend Contents of Tables (Data Browser)
- Take a look at the data of your tables by using the Data Browser (in the Dictionary: Utilities → Table Contents → Display). Modify some of the tuples or add new tuples (in the Dictionary: Utilities → Table Contents → Create Entries). With this occasion, you can also check whether the system guarantees referential integrity.

Now, with your data model and the tables, you have set the foundations for the integration of banking services into the system. The implementation of programs accessing the data of these tables, and the implementation of transactions and GUIs will start in the next practical exercise.

Final Remark: There is no due date for this exercise; however, since the next exercises will be built on the data model and the tables that are being created here, it is highly recommended to have this exercise completed before you start with the following ones.

Although this exercise is not mandatory, you are invited to submit your results via Email to heiko.schuldt@hpi.uni-potsdam.de indicating “ERP: Exercise 2” in the subject line. Please include the results of question 1 in your mail. For question 2, it is sufficient to mention your SAP account ID. Everything can then be accessed via your development class.