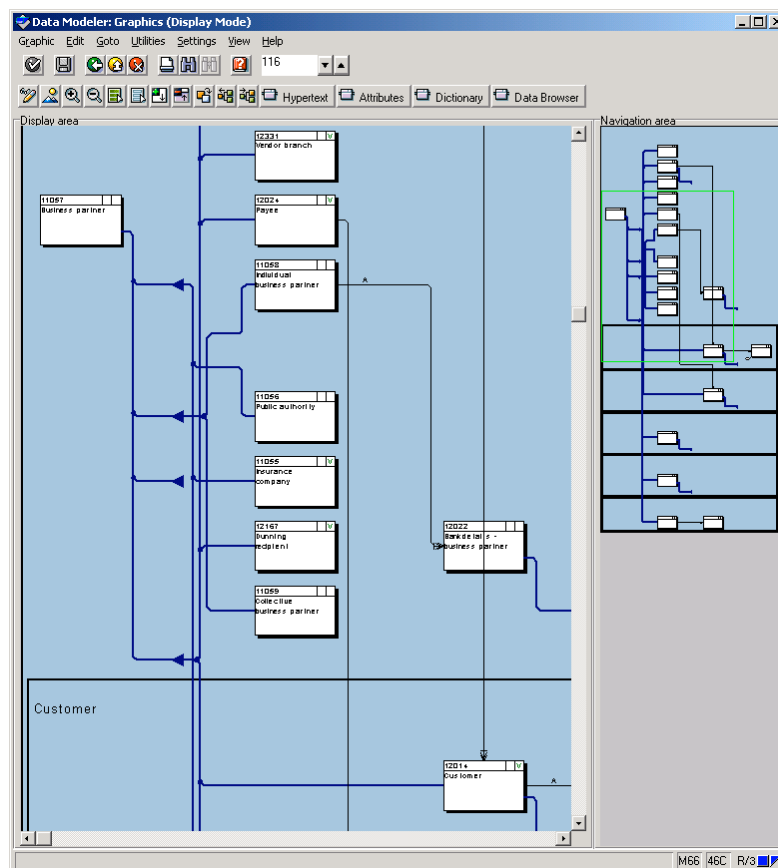


Question 1

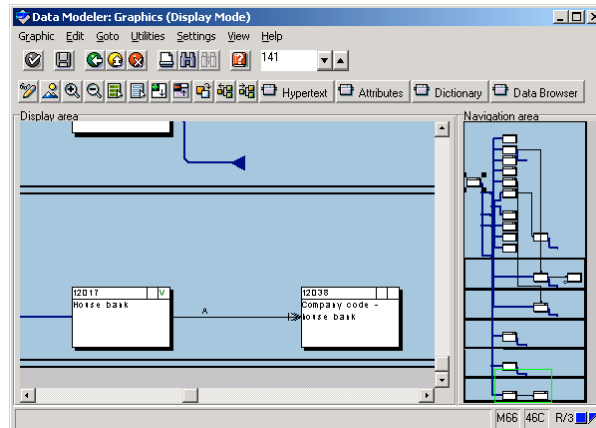
- a) *Data Model: Display the data model "Business partner" FI__10159 in the Data Modeler (this is the data model we have already analyzed in the last exercise; note that it has three underscores in its name).*
 i) *Which relationship ((in SAP's SERM terminology) is defined between the entity types "Business Partner" and "Customer" (ID 12014) ?*

Spec. category ... to ent.type	Short description: specialization cat. Short text: Entity type
0001	Individuality
11058	Individual business partner
11059	Collective business partner
0002	Type of business relations
11055	Insurance company
11056	Public authority
12017	House bank
15047	Vendor
16001	Customer
16096	Sales partners
16400	Competitor
0003	Role
11063	Lessor
12014	Customer
12016	Vendor
12023	Payer
12024	Payee
12167	Dunning recipient



“Customer” (located in the lower right corner of the screenshot) is a specialisation of “Business Partner” (upper left part).

ii) Which relationship is defined between the entity types “House Bank” and “Company Code - House Bank” ? What are the cardinalities of this relationship ?



The 1:cm-aggregation assigns arbitrarily “Company Code – House Bank” to each “House Bank”.

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 ?

Unfortunately, I did not find a dialog which explicitly tells me the exact number of attributes. Nevertheless, the scrollbar displays a tooltip while dragging it. Its upper limit – when scrolled to the bottom – are astonishing 140. Therefore, I suppose that the number of KNA1’s attributes should be 140.

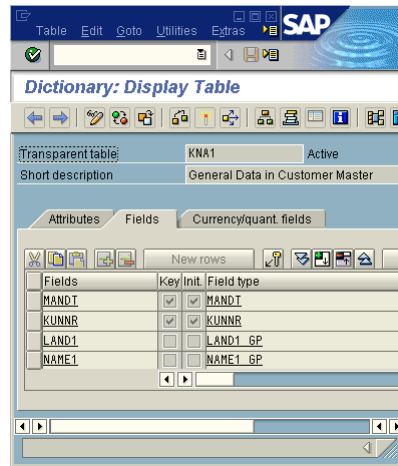
ii) How is this table mapped to table(s) of the underlying database system ?

The screenshot shows the SAP Dictionary: Display Table for KNA1. The table is marked as a 'transparent table' and is active. The short description is 'General Data in Customer Master'. The 'Fields' tab is selected, showing a list of fields with their attributes.

Fields	Key	Init	Field type	Data...	Lgth.	Dec.p...	Check table	Short text
MANDI	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDI	CLNT	3	0	I000	Client
KUNNR	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	KUNNR	CHAR	10	0		Customer numr
LAND1	<input type="checkbox"/>	<input type="checkbox"/>	LAND1 GP	CHAR	3	0	I005	Country key
NAME1	<input type="checkbox"/>	<input type="checkbox"/>	NAME1 GP	CHAR	35	0		Name 1
NAME2	<input type="checkbox"/>	<input type="checkbox"/>	NAME2 GP	CHAR	35	0		Name 2
ORT01	<input type="checkbox"/>	<input type="checkbox"/>	ORT01 GP	CHAR	35	0		City
PSTLZ	<input type="checkbox"/>	<input type="checkbox"/>	PSTLZ	CHAR	10	0		Postal Code
REGIO	<input type="checkbox"/>	<input type="checkbox"/>	REGIO	CHAR	3	0	I005S	Region (State,
SORTL	<input type="checkbox"/>	<input type="checkbox"/>	SORTL	CHAR	10	0		Sort field

As a transparent table, KNA1 maps 1:1 to a database table (see first line, marked by a dotted line).

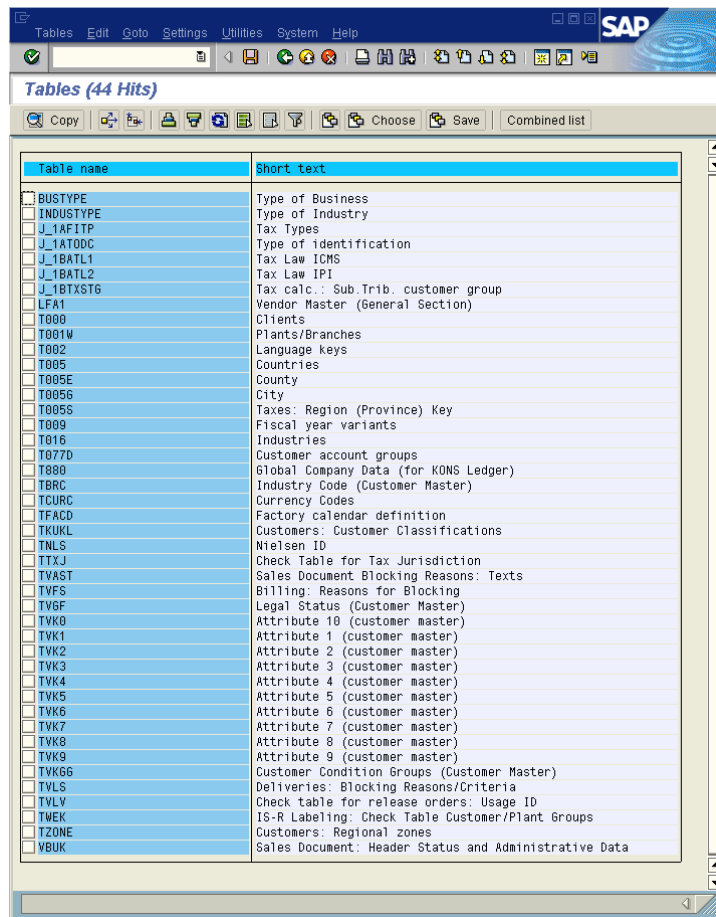
iii) What are the names of the key attributes of this table ?



The key attributes – set checkbox in column “key” – are MANDT and KUNNR.

iv) To how many other tables do foreign key relationships exist ?

Quite a lot: 44.



Question 2

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.

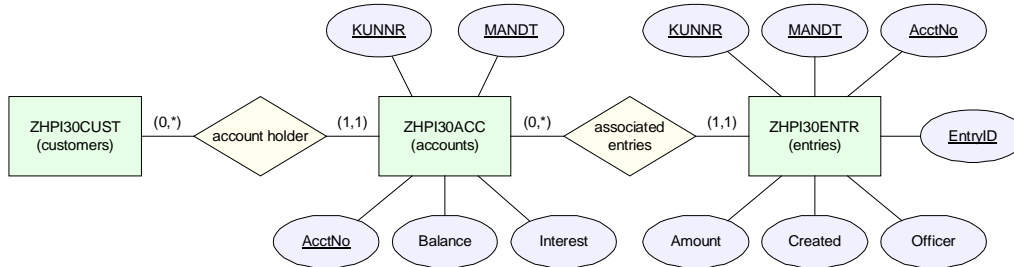


Figure 1: ER Model of your Banking Application

I am registered in the system as DEVELOP30.

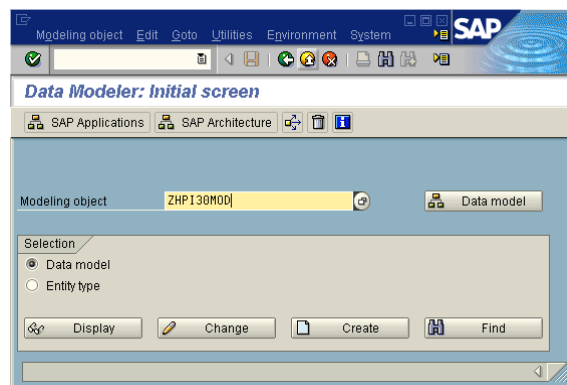
a) Create Data Model (Data Modeler)

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

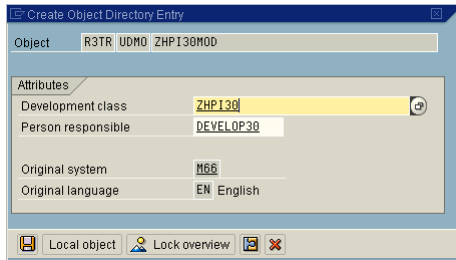
Hint: Create the new data model ZHPIxxMOD and use within this data model for the three entities the names given above in the ER model, i.e., ZHPI18CUST for user develop18. Just create the entities – without further specifying the attributes. Consider the particular structural restrictions of SAP SERM 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.

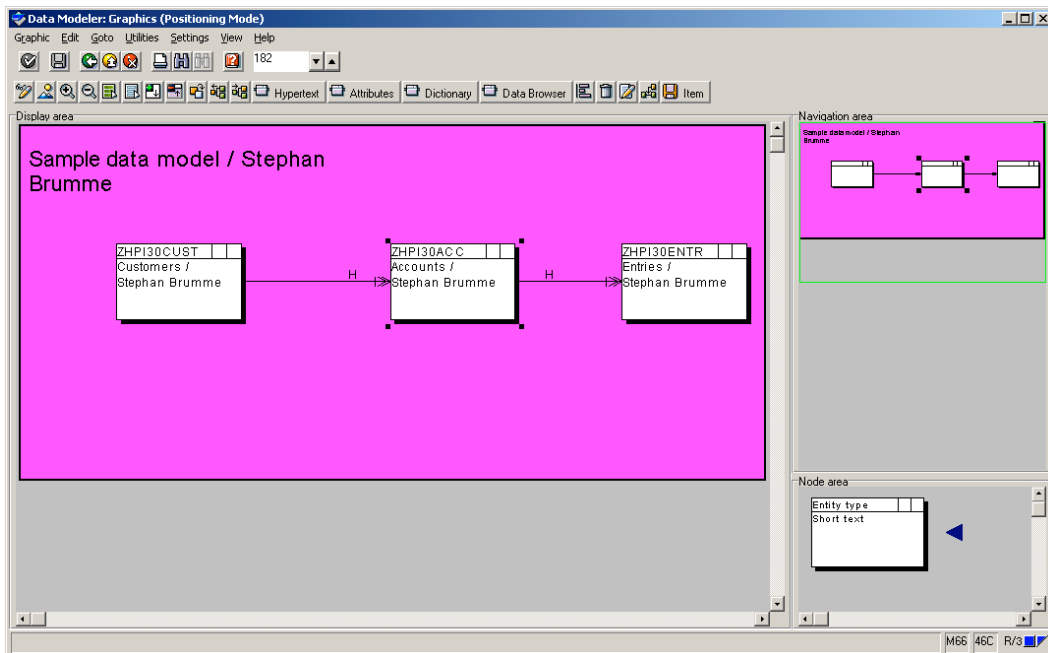
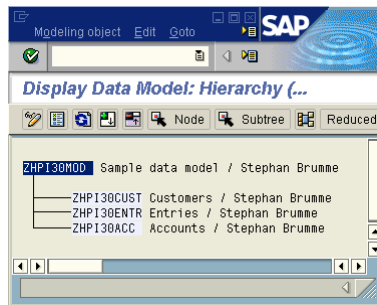
My new model ZHPI30MOD has been created in just a few seconds.



The next step, setting up the development class ZHPI30, caused no further riddles. Furthermore, I passed my ID to the initial request dialog which in turn generated a request ID.



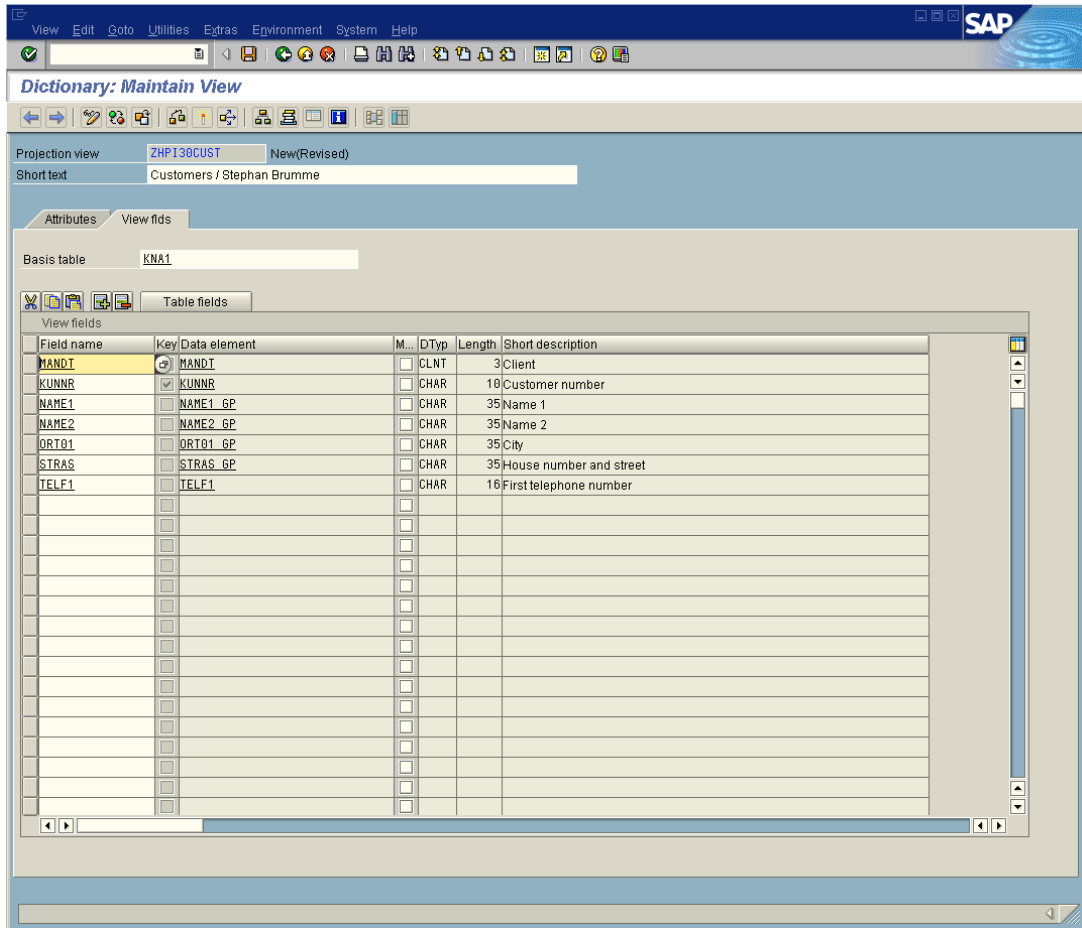
During the following minutes, I built all three entities and connected them (see second screenshot, the one with purple background).



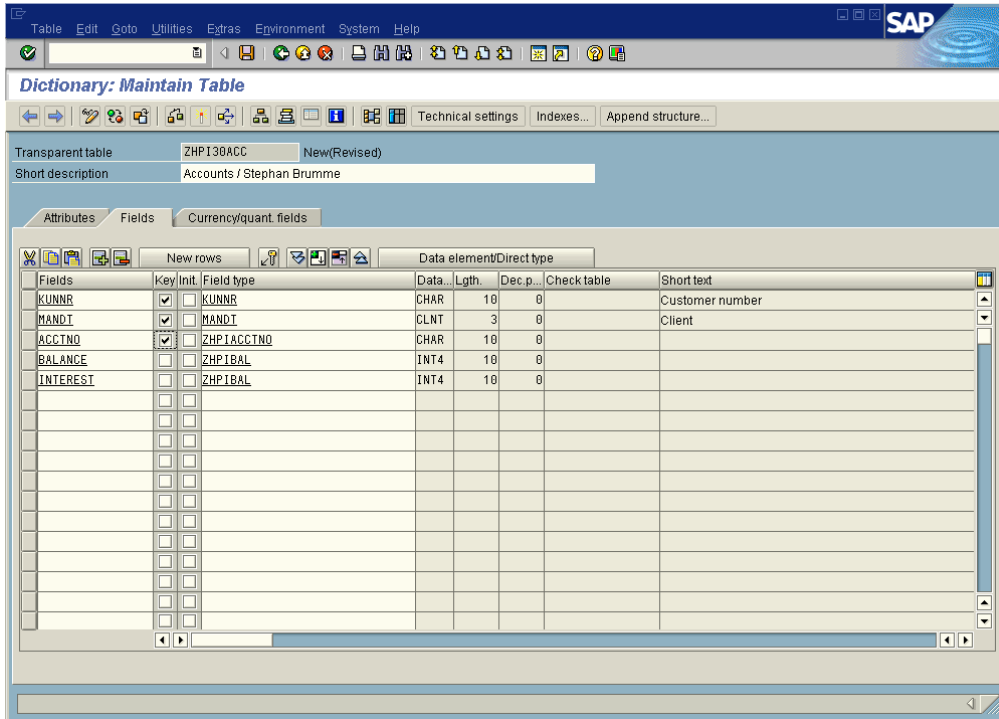
b) Specification of Entities (Dictionary)

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.

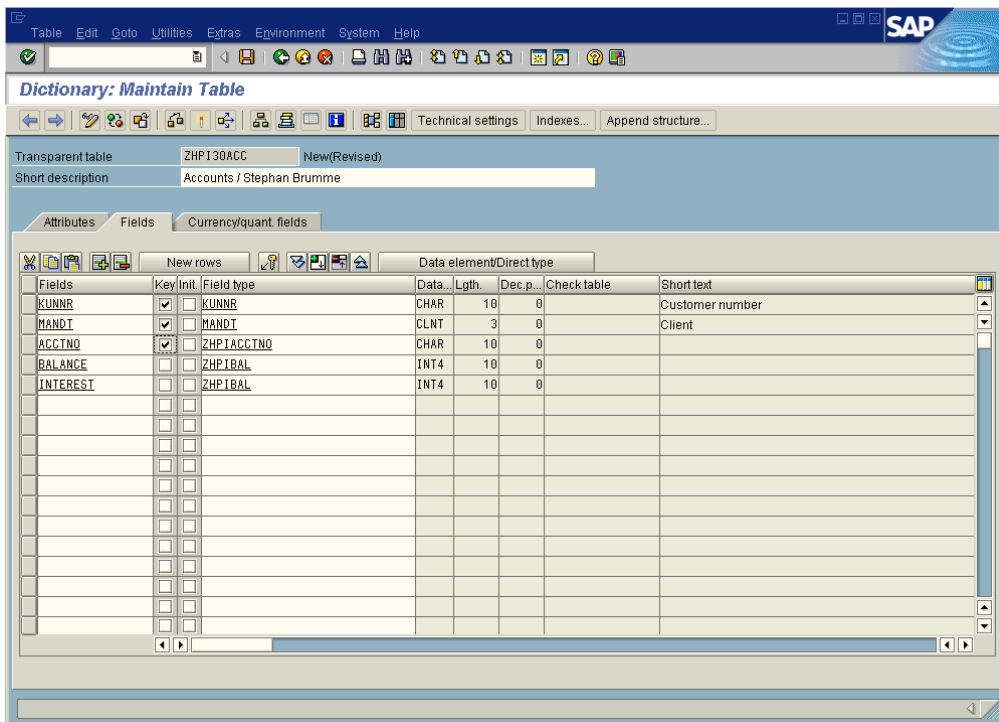
My view consists of seven attributes: MANDT, KUNNR, NAME1, NAME2, ORT01, STRAS and TELF1.



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).



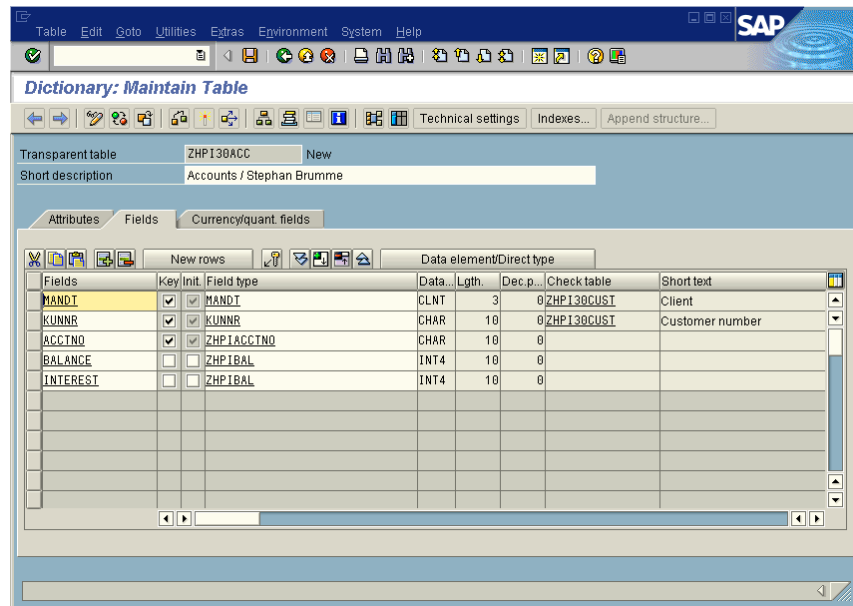
iii) Define the attributes of this table. Hint: The data elements for MANDT and KUNNR can be taken from the view ZHPIxxCUST (MANDT and KUNNR) you have just created. The data elements for the remaining attributes are already given (ZHP1ACCTNO for the account number, ZHP1BAL for both balance and interest). In order to facilitate the import of data, you should stick to this order when defining the attributes.



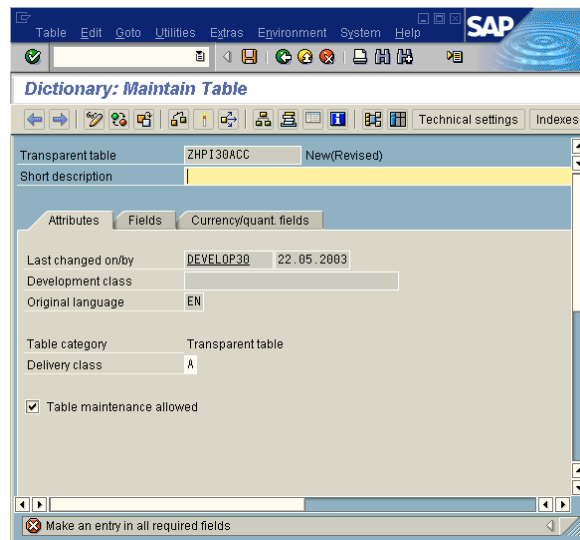
iv) Specify the primary keys given in the ER model for table ZHPIxxACC.

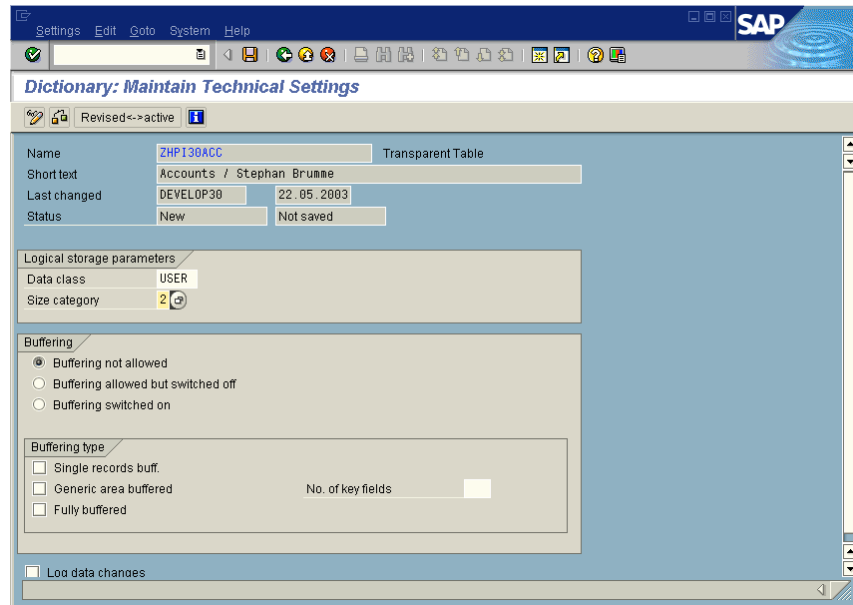
See screenshot above.

v) Specify the foreign keys MANDT and KUNNR, which are dependent from the corresponding attributes of view ZHPIxxCUST.

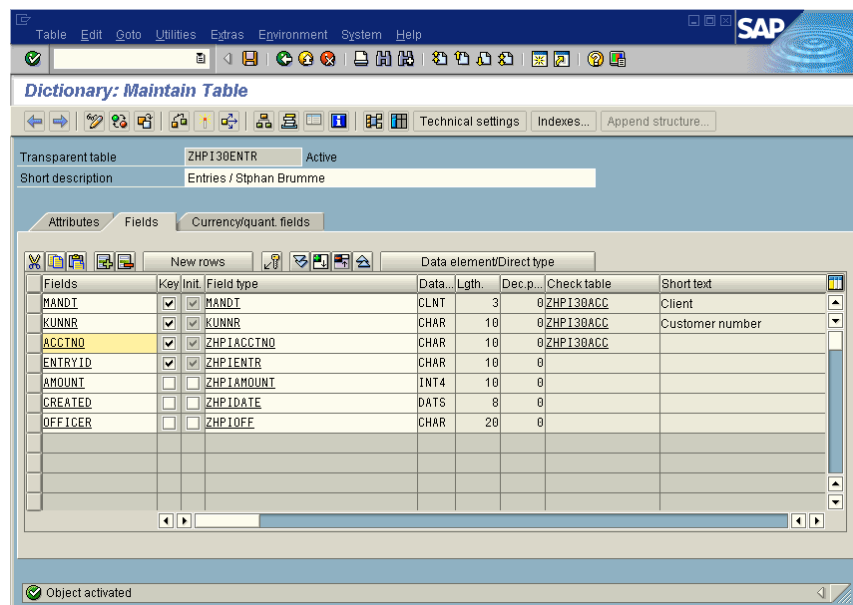


vi) Save and activate (Table → Activate) your account table1. Via (Table → Check) you can figure out whether your table definition is consistent. Hint: When creating a table, make sure that you allow table maintenance (otherwise, data can only be modified by ABAP programs but not via the Data Browser). 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 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.

Because I just had to redo the same steps that I took for ZHPI30ACC, I present only a single screenshot (the one above). It covers ix) as well.

ix) Save and activate the account entry table.

See the screenshot above.

- 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”.*

I immediately found the ABAP program. Entering the appropriate table names filled them with many tuples.

```

REPORT ZBANKCOPY .
PARAMETERS: ACC_TAB(20) TYPE C, "Input parameters
            ENTR_TAB(20)  TYPE C.

TABLES: ZHPIACC,           "Base Tables
        ZHPIENTRY.

DATA: KONTO_TUPEL LIKE ZHPIACC, "stores a
      BUCHUNGS_TUPEL LIKE ZHPIENTRY. "single record

* "Daten Zeilen/Tupelweise aus Vorgabe auslesen
SELECT * FROM ZHPIACC INTO KONTO_TUPEL.
* "Eine Zeile in neue Tabelle einfüegen
MODIFY (ACC_TAB) FROM KONTO_TUPEL.
ENDSELECT.

* "Daten Zeilen/Tupelweise aus Vorgabe auslesen
SELECT * FROM ZHPIENTRY INTO BUCHUNGS_TUPEL.
* "Eine Zeile in neue Tabelle einfüegen
MODIFY (ENTR_TAB) FROM BUCHUNGS_TUPEL.
ENDSELECT.
COMMIT WORK.
    
```

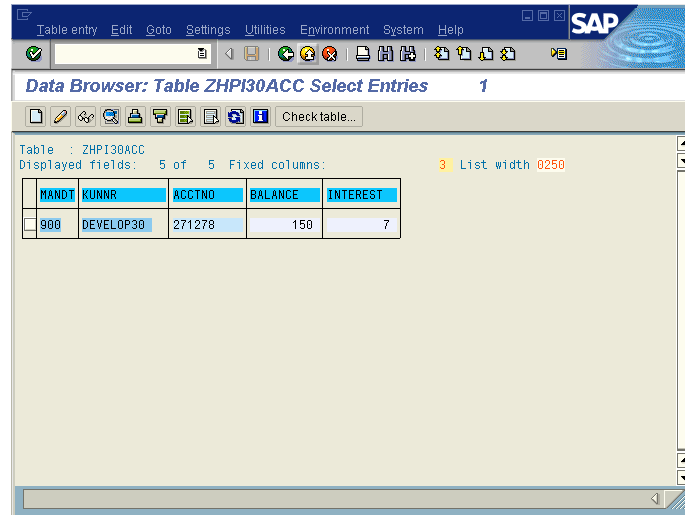
ACC_TAB	zhp130acc
ENTR_TAB	zhp130entr

SAP R/3

The following tables ZHP130ACC and ZHP130ENTR have been filled with data.
Have fun with the definition of your banking services in R/3!

- 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.*

I added a new account (no. 271278) owned by me with a balance of 150 and an interest rate of 7%.



The screenshot shows the SAP Data Browser interface for table ZHPI30ACC. The window title is "Data Browser: Table ZHPI30ACC Select Entries 1". The table has 5 columns: MANDT, KUNNR, ACCTNO, BALANCE, and INTEREST. The displayed data is as follows:

MANDT	KUNNR	ACCTNO	BALANCE	INTEREST
900	DEVELOP30	271278	150	7