Capgemini

ABAP Part I

Lesson 03: DDIC - I

Lesson Objectives

After completing this lesson, participants will be able to -

- Use Data Dictionary to maintain Database Objects
- Work with
  - Domain
  - Data Elements
  - Tables
  - Structures

Data Dictionary

Data Definitions are created and Managed in ABAP Dictionary

Describes the logical structure of objects use in application development

Describes the mapping of data to the underlying Relational Database in tables and views

System Independent interface to the Database

Virtual Database

Provides data for manipulation and processing

Transaction Code : SE11

ABAP Dictionary

Object Types in ABAP Dictionary are

- Tables
  - Defined in Dictionary
  - Independent of Database
- Views
  - Logical Views
- Types
  - Data elements
  - Structures
  - Table Types
- Domain
  - Defines a Value Range
- Search Helps
  - Input Help or F4 Help
- Lock Objects
  - Lock Mechanism to set and release the locks

ABAP Dictionary

TABLES

- Has one or More fields
- Contains data in the form of Rows and Columns

DATA ELEMENTS

- Field of a table refers to Data Element
- Specifies Non-Technical Attributes

DOMAIN

- Specifies Technical Attributes
- Attached to Data Element

Standard Tables

Table	Description
DD02L	List of All Tables
TSTC	List of All Tcodes
TADIR	R/3 Repository Objects
T000	Clients

Types of Table

Table Types

- Transparent Tables
- Pooled Tables
- Cluster Tables

Tables

Transparent Tables

- One-to-one relationship with tables in database
- Most commonly Used
- Holds Application data
- Master data or Transaction data Used by an application
  - Master Data : Vendor Table
  - Transaction data : Orders Placed By Customers

Tables

TABLE FIELDS

- Field Name
  - Should begin with an Alphabet
- Key Flag
  - Determines the Primary Key
- Field Type
  - Data Type
- Field Length
- Decimal Places
  - Number of Decimal Places
- Short Text – Description of the field

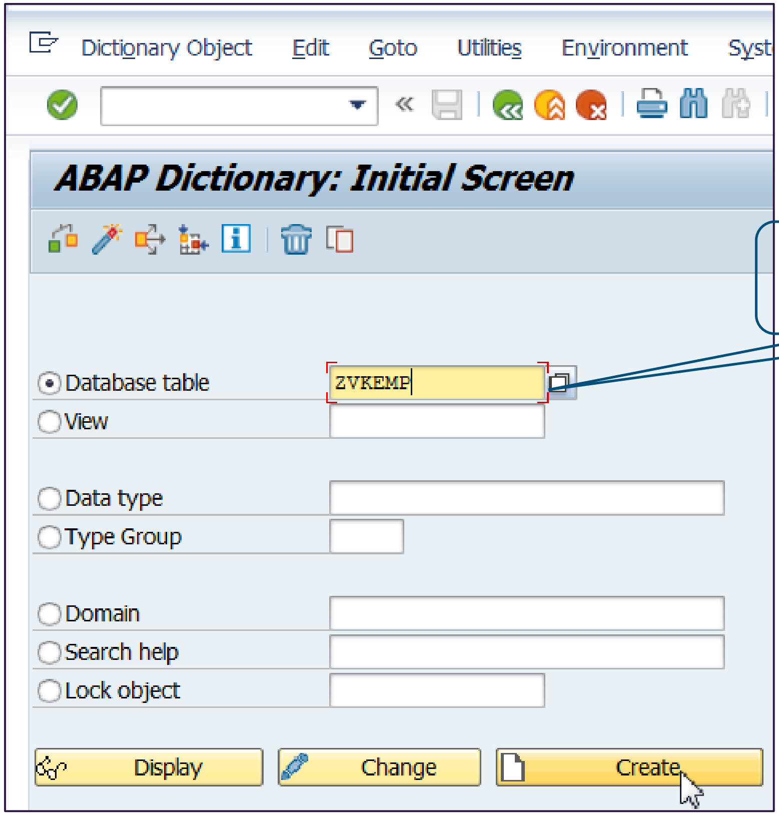
Tables - Transparent

Creation of Tables

- Top-Down Approach
  - Table is first created
  - Data element and Domain are created after creation of Table
  - Easier to Use

- Bottom-Up Approach
  - Domain and Data element are created
  - More intuitive for first timers
  - Cumbersome

Table Creation



ABAP Dictionary: Initial Screen

● Database table ZVKEMP

○ View

○ Data type

○ Type Group

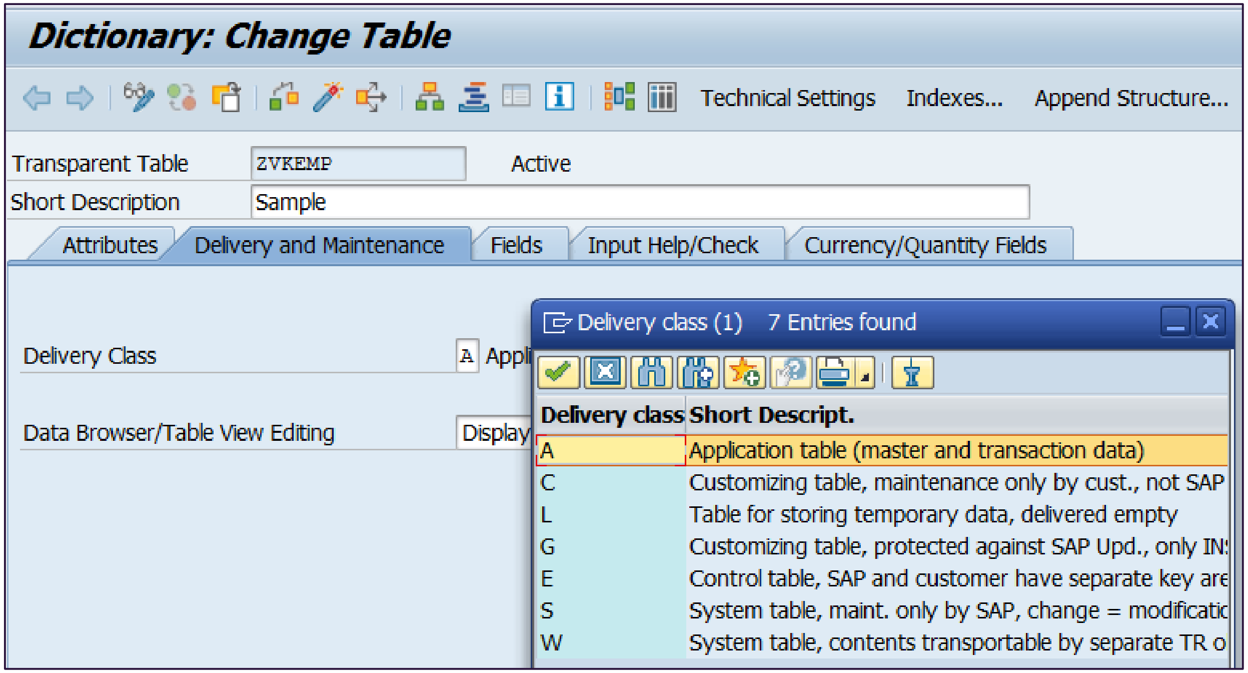
○ Domain

○ Search help

○ Lock object

Specify the Table Name

Table Creation



Dictionary: Change Table

Transparent Table ZVKEMP Active

Short Description Sample

Delivery Class A Appl

Data Browser/Table View Editing Display

Delivery class (1) 7 Entries found

Delivery class	Short Descript.
A	Application table (master and transaction data)
C	Customizing table, maintenance only by cust., not SAP
L	Table for storing temporary data, delivered empty
G	Customizing table, protected against SAP Upd., only IN
E	Control table, SAP and customer have separate key are
S	System table, maint. only by SAP, change = modificatio
W	System table, contents transportable by separate TR o

Choose Delivery Class

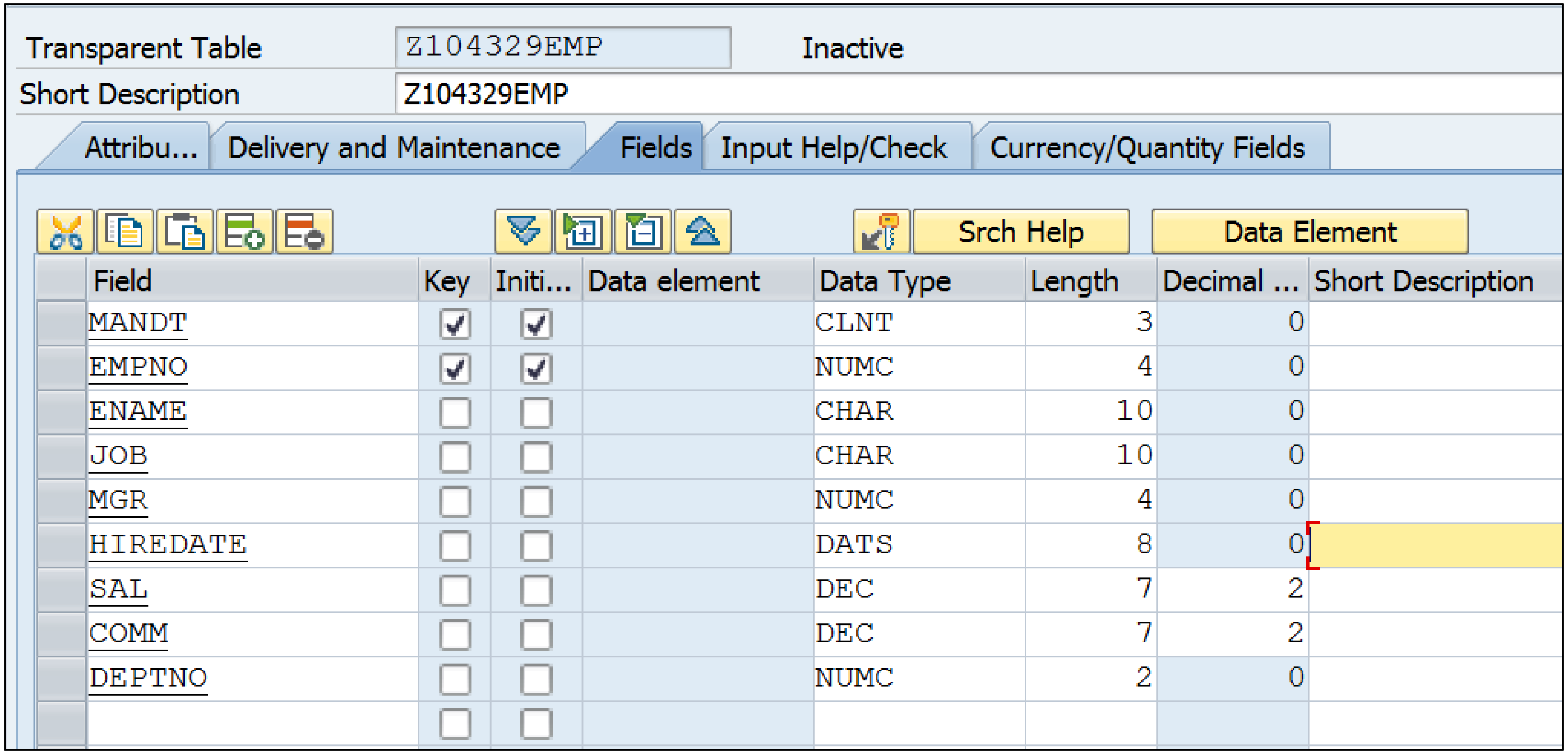
Delivery Classes

The delivery class controls the transport of table data for installation, upgrade, client copy and when transporting between customer systems.

There are the following development classes:

- A: Application table (master and transaction data)
- C: Customer table, data is only maintained by the customer.
- L: Table for storing temporary data
- G: Customer table, SAP may insert new data records but may not overwrite or delete existing ones
- E: System table with its own namespace for customer entries. The customer namespace must be defined in table TRESC
- S: System table, data changes have the status of program changes.
- W: System table (e.g. table of the development environment) whose data is transported with its own transport objects (e.g. R3TR PROG, R3TR TABL, etc.).

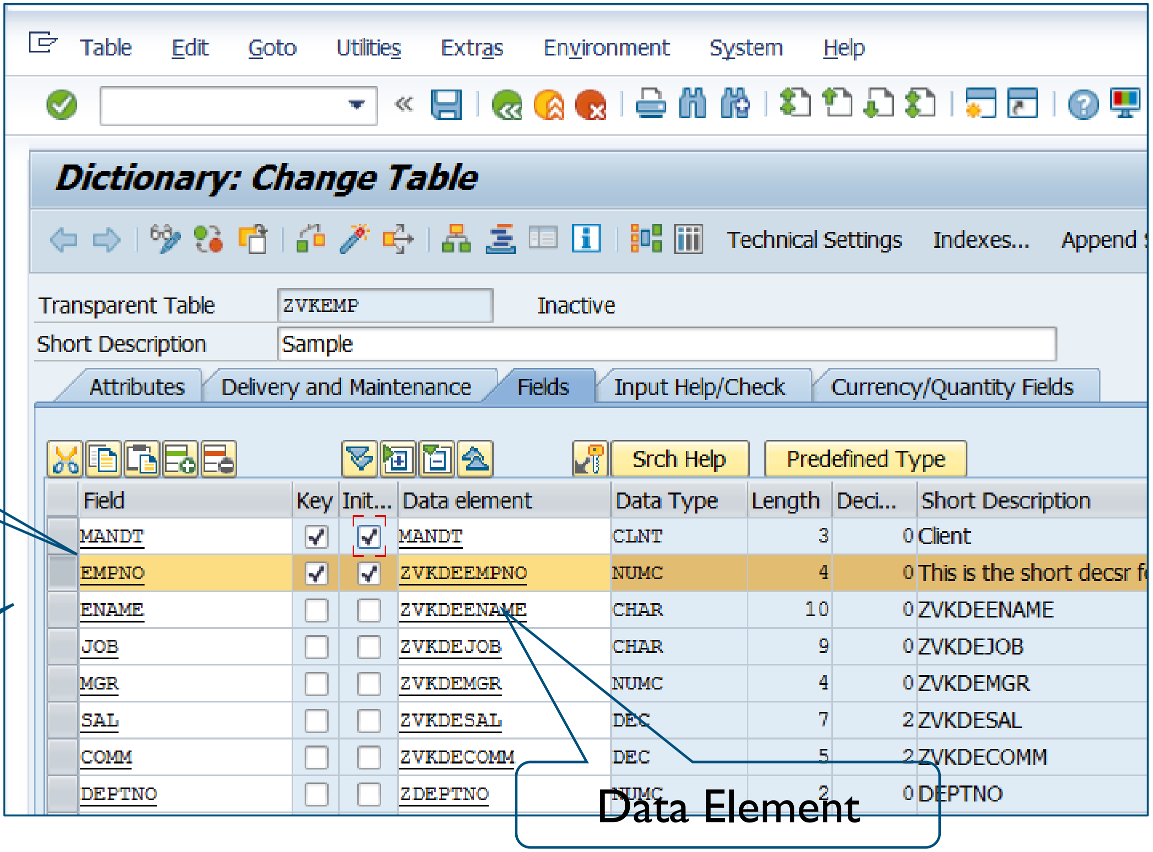
Table Creation - Fields



Transparent Table	Z104329EMP Inactive
Short Description	Z104329EMP

Field	Key	Initi...	Data element	Data Type	Length	Decimal ...	Short Description
MANDT	☑	☑		CLNT	3	0	
EMPNO	☑	☑		NUMC	4	0	
ENAME	□	□		CHAR	10	0	
JOB	□	□		CHAR	10	0	
MGR	□	□		NUMC	4	0	
HIREDATE	□	□		DATS	8	0	
SAL	□	□		DEC	7	2	
COMM	□	□		DEC	7	2	
DEPTNO	□	□		NUMC	2	0	
	□	□					

Table Creation - Fields



Dictionary: Change Table

Transparent Table ZVKEMP Inactive

Short Description Sample

Field	Key	Init...	Data element	Data Type	Length	Deci...	Short Description
MANDT	☑	☑	MANDT	CLNT	3	0	Client
EMPNO	☑	☑	ZVKDEEMPNO	NUMC	4	0	This is the short decsr f
ENAME	□	□	ZVKDEENAME	CHAR	10	0	ZVKDEENAME
JOB	□	□	ZVKDEJOB	CHAR	9	0	ZVKDEJOB
MGR	□	□	ZVKDEMGR	NUMC	4	0	ZVKDEMGR
SAL	□	□	ZVKDESAL	DEC	7	2	ZVKDESAL
COMM	□	□	ZVKDECOMM	DEC	5	2	ZVKDECOMM
DEPTNO	□	□	ZDEPTNO	NUMC	2	0	DEPTNO

Primary Key Field

Specify the field names

Data Element

Demo

Create a custom Table

Table Creation

CONSTRAINTS

- Key Fields must be stored at the beginning of the field list
- Non-Key fields may not occur between two key fields
- Maximum of 16 key fields per table is allowed
- Table may not have more than 249 fields

Tables

Client Dependent Table

- First Field has Data Type CLNT
- Part of PRIMARY KEY Field

Client Independent Table

- A table whose First field is not of Data Type CLNT

Tables (Contd.).

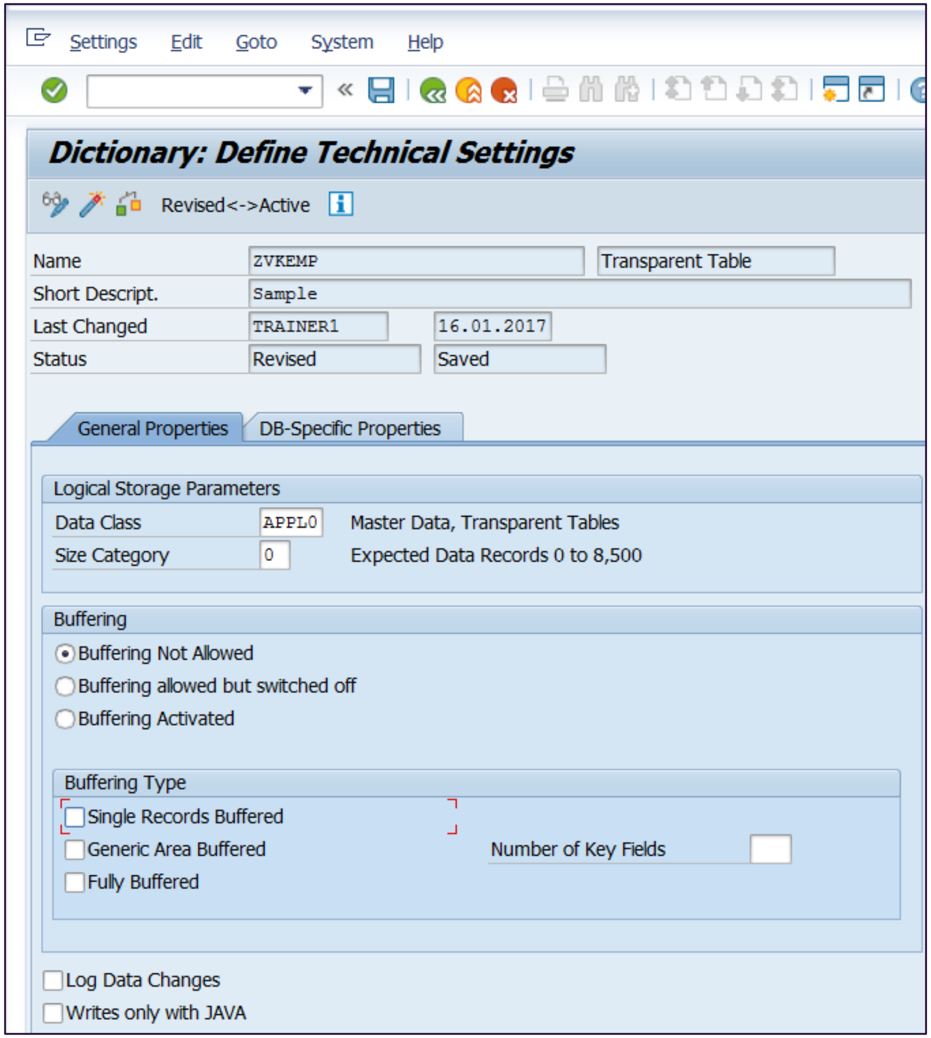
Reference Fields

- Reference Fields required for the following Data Type
  - QUAN
  - CURR
- Reference Fields should of Type
  - UNIT
  - CUKY
- Reference Fields can be in the same table or another table.

Demo

SE11 interface and create a simple table based on pre defined datatypes.

Tables – Technical Settings



Dictionary: Define Technical Settings

Name	ZVKEMP	Transparent Table
Short Descript.	Sample	
Last Changed	TRAINER1	16.01.2017
Status	Revised	Saved

Logical Storage Parameters

Data Class	APPL0	Master Data, Transparent Tables
Size Category	0	Expected Data Records 0 to 8,500

Buffering

● Buffering Not Allowed
○ Buffering allowed but switched off
○ Buffering Activated

Buffering Type

□ Single Records Buffered
□ Generic Area Buffered   Number of Key Fields
□ Fully Buffered

□ Log Data Changes
□ Writes only with JAVA

Data Class

The data class logically defines the physical area of the database where your base database table resided.

Hence, you should choose the data class correctly, the table will automatically created in the appropriate area on the database when it is activated in the dictionary.

The most important data classes are master data, transaction data, organizational data and system data

The data class determines the table space that the table is assigned to.

A tablespace is a physical file on the disk which is used to hold tables

Every table is assigned to one tablespace

Data Class

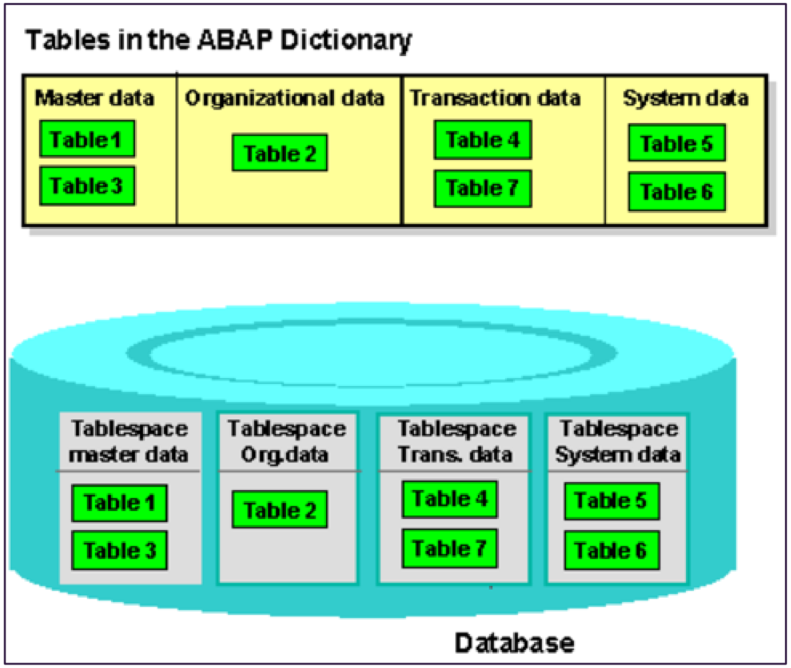
If you choose the data class correctly, your table is automatically assigned to the correct area (tablespace or DBspace) of the database when it is created.

Each data class corresponds to a physical area in which all the tables assigned to this data class are stored.

There are the following data classes:

- APPL0 (Master Data): Data which is seldom changed. An example of master data is the data contained in an address file, such as the name, address and telephone number.
- APPL1 (transaction data): Data that is frequently changed. An example of transaction data is the goods in a warehouse, which change after each purchase order.
- APPL2 (organizational data): Customizing data that is defined when the system is installed and seldom changed. An example is the table with country codes.

Data Class


Tables in the ABAP Dictionary
Master data
Table 1
Table 3
Organizational data
Table 2
Transaction data
Table 4
Table 7
System data
Table 5
Table 6
Tablespace master data
Table 1
Table 3
Tablespace Org.data
Table 2
Tablespace Trans. data
Table 4
Table 7
Tablespace System data
Table 5
Table 6
Database

Size Category

The size category describes the expected storage requirements for the table on the database.

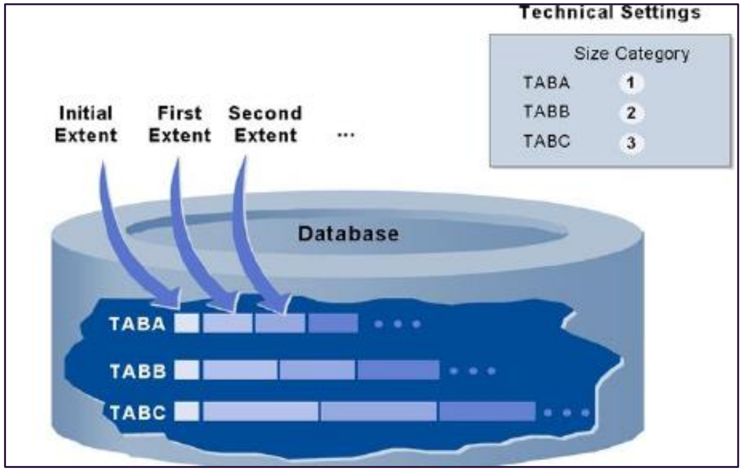
An initial extent is reserved when a table is created on the database.

The size of the initial extent is identical for all size categories.

If the table needs more space for data at a later time, extents are added.

These additional extents have a fixed size that is determined by the size category specified in the ABAP Dictionary.

You can choose a size category from 0 to 4. A fixed extent size, which depends on the database system used, is assigned to each category.



Logging

You can use logging to record and store modifications to the entries of a table.

To activate logging, the corresponding field must be selected in the technical settings.

Logging, however, only takes place if the R/3 system was started with the profile containing parameter rec/client.

Only selecting the flag in the ABAP/4 dictionary is not sufficient to trigger logging.

Table - Buffering

Buffering allows you to access data quicker by letting you access it from the application server instead of the database.

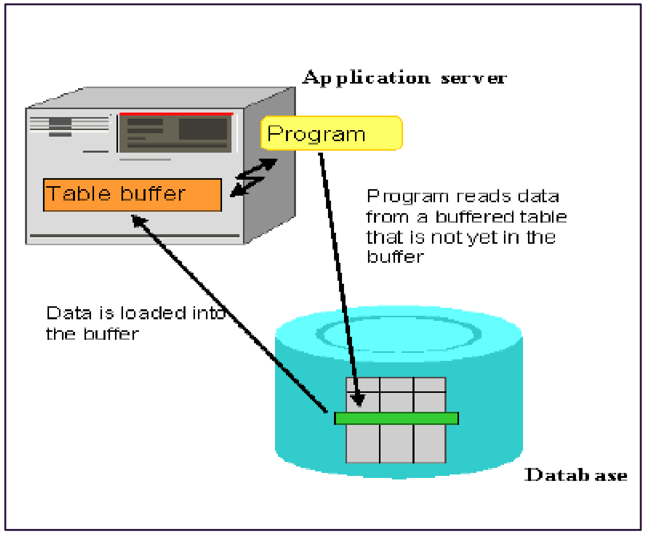

Application server
Program
Table buffer
Program reads data from a buffered table that is not yet in the buffer
Data is loaded into the buffer
Database

Table - Buffering

Buffers reside in each application server

Improves Performance

How are Buffers Filled?

- Program accesses data of a buffered table
- Database Interfaces checks if the data is available in buffer of Application Server
  - If Available, read from buffer
  - If Unavailable, read from database and load into buffer

Tables

Buffer Synchronization

- If Program changes data in a table on Application Server, it is noted in log table by Database Interface
- A synchronization Mechanism runs at a fixed time interval
- Log table is read and buffer contents are invalidated
- In next access, data is read from database table and updated in buffer.

Tables - Buffering

Buffering Tables

- Table that is frequently read and rarely changed
- The key fields of the buffered table should of the Character data types (C, N, D, T)
- By pass the buffer if table data should be read from Database

Table buffering

The buffering type determines which records of the table are loaded into the buffer of the application server when a record of the table is accessed.

There are the following buffering types:

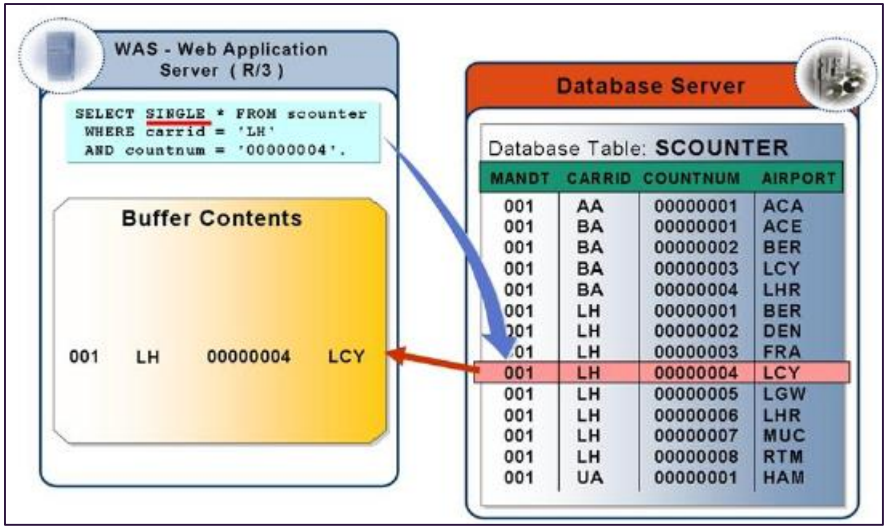
- **Full buffering**: When a record of the table is accessed, all the records of the table are loaded into the buffer.
- **Generic buffering**: When a record of the table is accessed, all the records whose left-justified part of the key is the same are loaded into the buffer.
- **Single-record buffering**: Only the record that was accessed is loaded into the buffer.

Single Record Buffering

Single-record buffering is recommended particularly for large tables in which only a few records are accessed repeatedly with SELECT SINGLE.

If you access a record that was not yet buffered using SELECT SINGLE, there is a database access to load the record. If the table does not contain a record with the specified key, this record is recorded in the buffer as non-existent. This prevents a further database access if you make another access with the same key

All the accesses to the table that do not use SELECT SINGLE bypass the buffer and directly access the database.



```
SELECT SINGLE * FROM scounter
 WHERE carrid = 'LH'
 AND countnum = '00000004'.
```

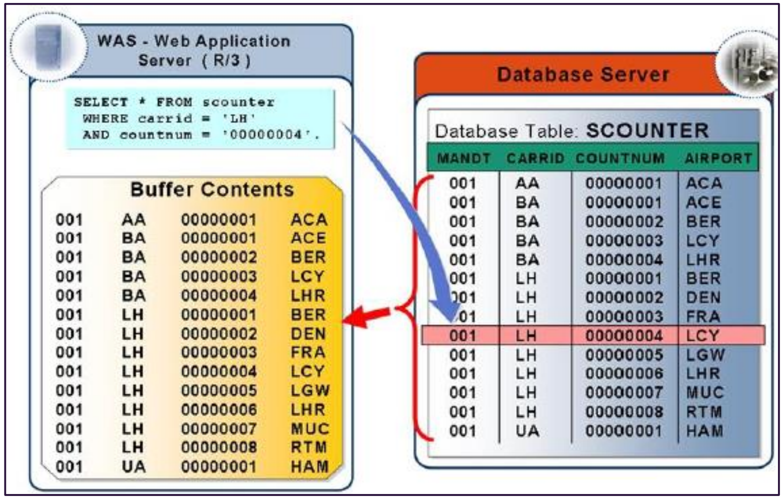
MANDT	CARRID	COUNTNUM	AIRPORT
001	AA	00000001	ACA
001	BA	00000001	ACE
001	BA	00000002	BER
001	BA	00000003	LCY
001	BA	00000004	LHR
001	LH	00000001	BER
001	LH	00000002	DEN
001	LH	00000003	FRA
001	LH	00000004	LCY
001	LH	00000005	LGW
001	LH	00000006	LHR
001	LH	00000007	MUC
001	LH	00000008	RTM
001	UA	00000001	HAM

Full Buffering

When full buffering, the table is either completely or not at all in the buffer. When a record of the table is accessed, all the records of the table are loaded into the buffer.

When you decide whether a table should be fully buffered, you must take the table size, the number of read accesses and the number of write accesses into consideration.

The smaller the table is, the more frequently it is read and the less frequently it is written, the better it is to fully buffer the table.



```
SELECT * FROM scounter
 WHERE carrid = 'LH'
 AND countnum = '00000004'.
```

Buffer Contents

001	AA	00000001	ACA
001	BA	00000001	ACE
001	BA	00000002	BER
001	BA	00000003	LCY
001	BA	00000004	LHR
001	LH	00000001	BER
001	LH	00000002	DEN
001	LH	00000003	FRA
001	LH	00000004	LCY
001	LH	00000005	LGW
001	LH	00000006	LHR
001	LH	00000007	MUC
001	LH	00000008	RTM
001	UA	00000001	HAM

Database Table: SCOUNTER

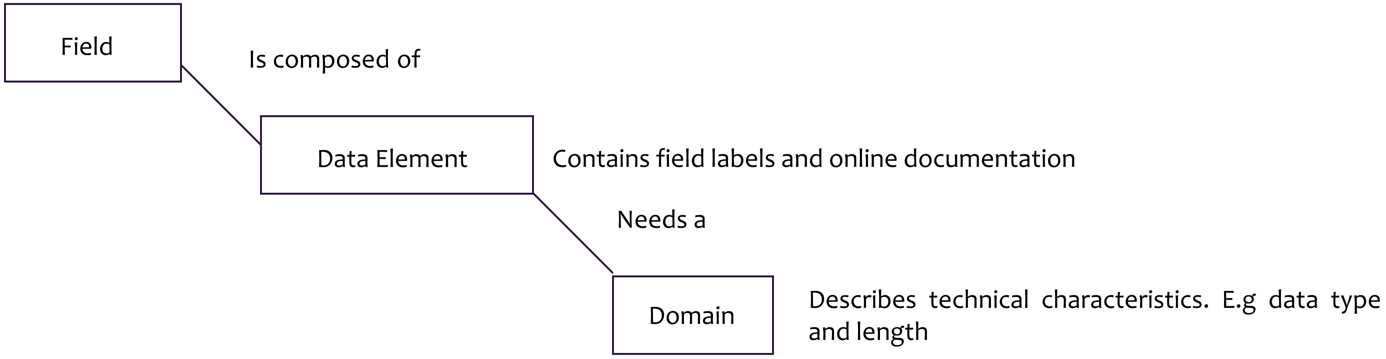
MANDT	CARRID	COUNTNUM	AIRPORT
001	AA	00000001	ACA
001	BA	00000001	ACE
001	BA	00000002	BER
001	BA	00000003	LCY
001	BA	00000004	LHR
001	LH	00000001	BER
001	LH	00000002	DEN
001	LH	00000003	FRA
001	LH	00000004	LCY
001	LH	00000005	LGW
001	LH	00000006	LHR
001	LH	00000007	MUC
001	LH	00000008	RTM
001	UA	00000001	HAM

Generic Buffering

With generic buffering, all records whose generic key fields agree with this record are loaded into the buffer when one record of the table is accessed.

The generic key is left-justified part of the primary key of the table that must be defined when the buffering type is selected.

Data elements , Domain and fields



Field

Is composed of

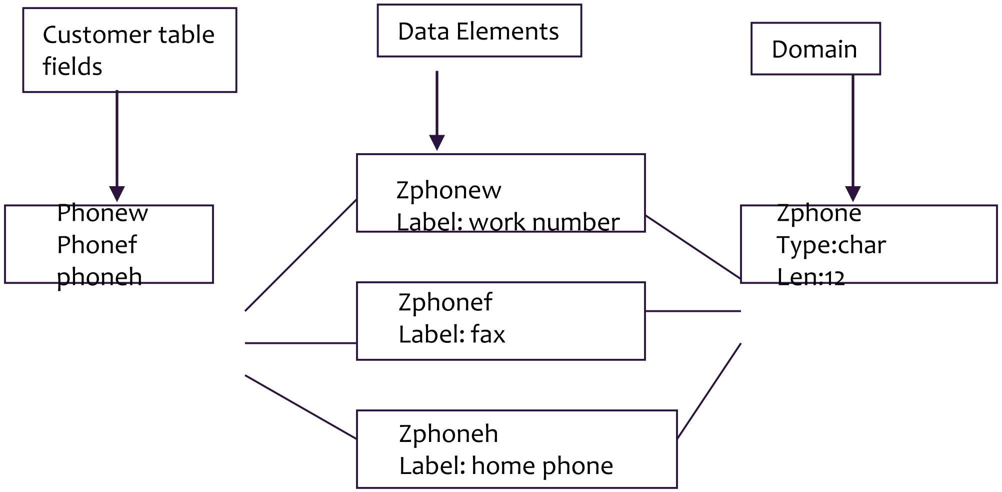
Data Element

Contains field labels and online documentation

Needs a

Domain

Describes technical characteristics. E.g data type and length



Customer table fields

Phonew
Phonef
phoneh

Data Elements

Zphonew
Label: work number

Zphonef
Label: fax

Zphoneh
Label: home phone

Domain

Zphone
Type:char
Len:12

Domain

Specifies the Technical Characteristics of a Field

- Data Type
- Length

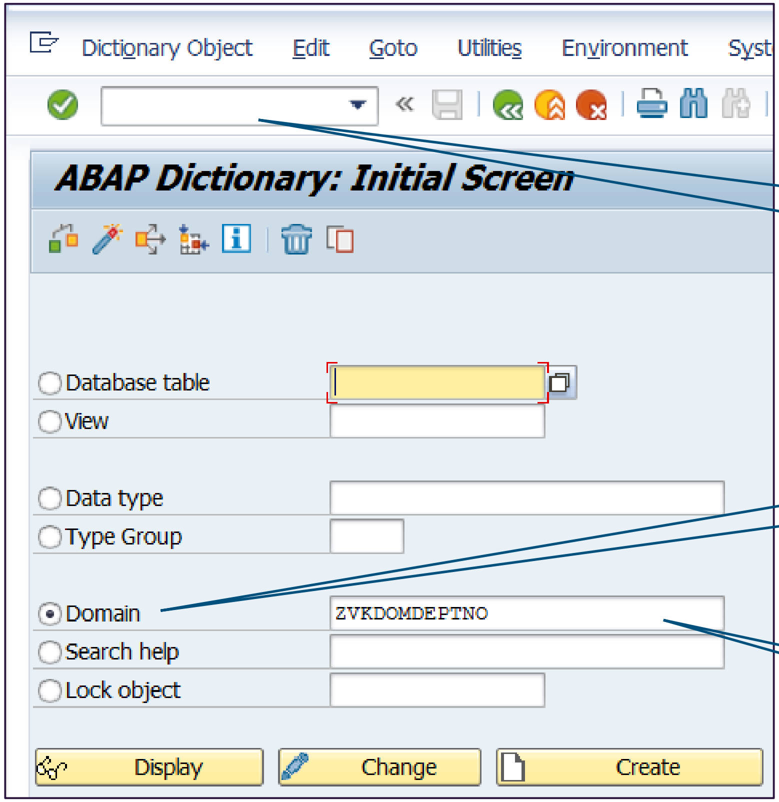
Defines a Value Range

Can be restricted by defining Fixed Values

Define Value Table to check against a Table

Assigned to a Data Element Defines Value Range

Creating Domain



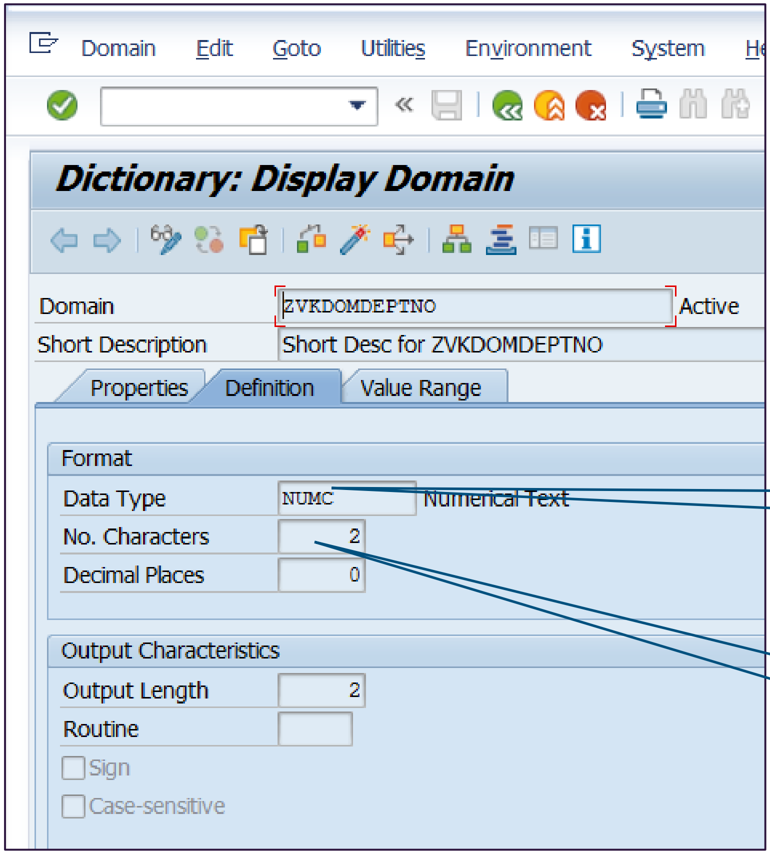
Domain: ZVKDOMDEPTNO

Tcode : SE11

Choose Domain

Specify the Domain Name starting with Y or Z

Creating Domain



Dictionary: Display Domain

Domain: ZVKDOMDEPTNO Active

Short Description: Short Desc for ZVKDOMDEPTNO

Format

Data Type	NUMC	Numerical Text
No. Characters	2	
Decimal Places	0	

Output Characteristics

Output Length	2
Routine	

□ Sign

□ Case-sensitive

Specify the Datatype

Length for the Datatype

Creating Domain (Contd.).

Save the Domain

Specify the Package

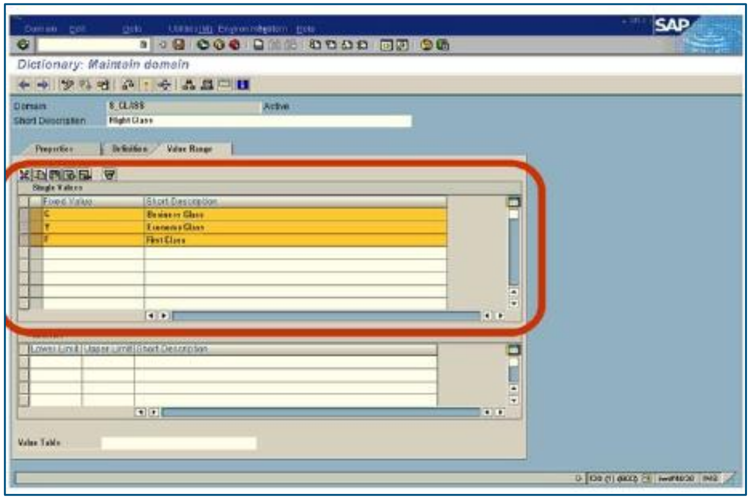
Activate

Domain is ready and can be attached to a Data Element

Domain

Value Range and Fixed Values

- Used to restrict the values in the Domain
- Used in input check in screen templates
- If no other help is defined in field, Value Range or Fixed Values are offered in F4 help.
- Value Range or Intervals can be defined by specifying the upper and lower limits
- Although S_CLASS is a domain of the type C, it would accept no other character besides C/Y/F.



Dictionary: Maintain domain

Domain S_CLASS Active
Short Description Flight Class

Single Values

Fixed Value	Short Description
C	Business Class
Y	Economy Class
F	First Class

Lower Limit	Upper Limit	Short Description

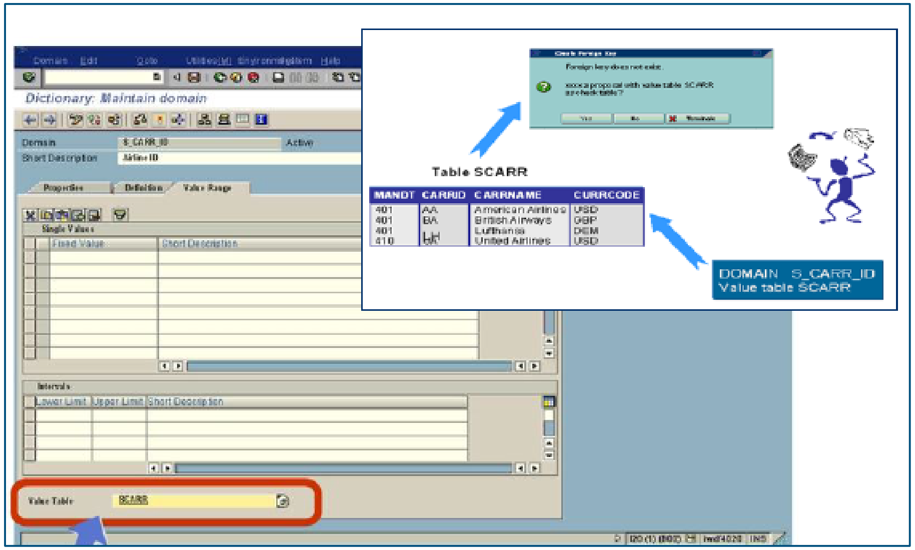
Value Table

Value Table

Maintained at Domain Level

A check is not implemented by simply entering a value table.

The check against the value table only takes effect when a foreign key* has been defined.



Dictionary: Maintain domain

Domain: S_CARR_ID Active
Short Description: Airline ID

Fixed Value	Short Description

Lower Limit	Upper Limit	Short Description

Value Table: SCARR

Foreign key does not exist.

Accept proposal with value table SCARR as check table?

Table SCARR

MANDT	CARRID	CARRNAME	CURRCODE
401	AA	American Airlines	USD
401	BA	British Airways	GBP
401	LH	Lufthansa	DEM
410	UA	United Airlines	USD

DOMAIN S_CARR_ID
Value table SCARR

Domain

Input Check valid for few data types

- Value Range
  - CHAR
  - NUMC
- Fixed Values
  - CHAR
  - NUMC
  - DEC
  - INT1
  - INT2
  - INT4

Data Elements

Specifies the Semantic Characteristics of a Field

Describes an Elementary type or a Reference Type

- Elementary type
  - Defined by built-in data type and length
  - OR
  - Defined directly or specified through a Domain
- Reference Types
  - Defines the type of Reference Variable to a Class or an Interface

Data Elements

Field Label

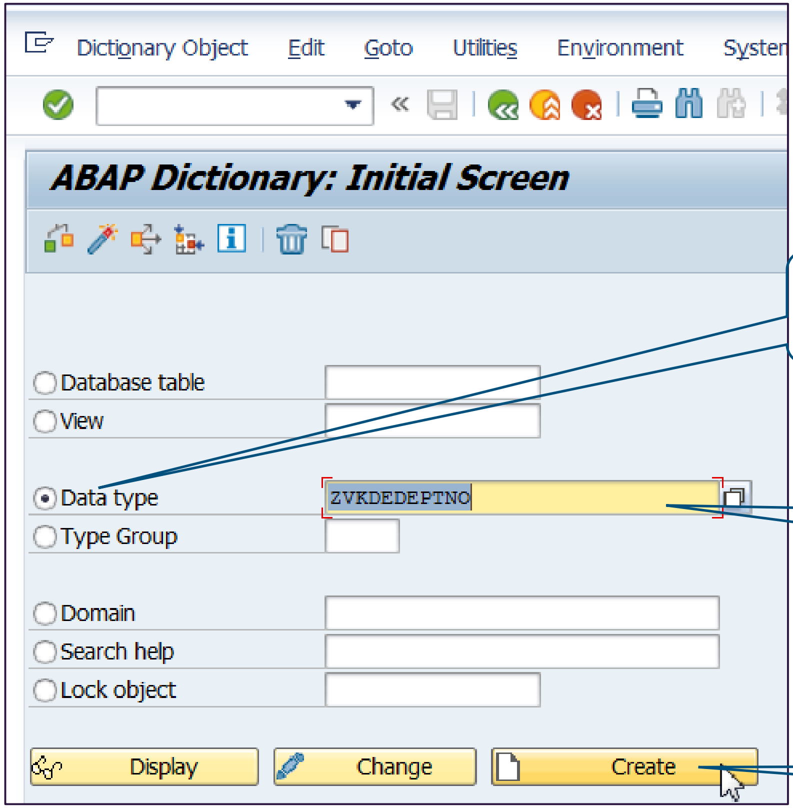
- Field Labels are used to display a screen field

F1 Documentation

- The text appearing in the Field Help (F1 Help) comes from the documentation
- If there is no Documentation, the short text appears

Creating Data Element

Go to Tcode : SE11



ABAP Dictionary: Initial Screen

○ Database table

○ View

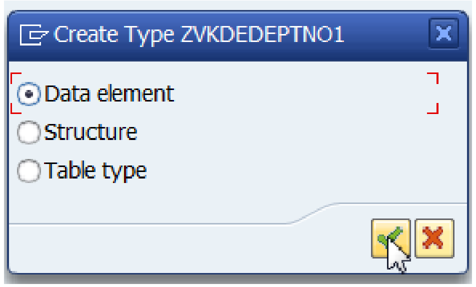
● Data type ZVKDEDEPTNO

○ Type Group

○ Domain

○ Search help

○ Lock object



Create Type ZVKDEDEPTNO1

● Data element

○ Structure

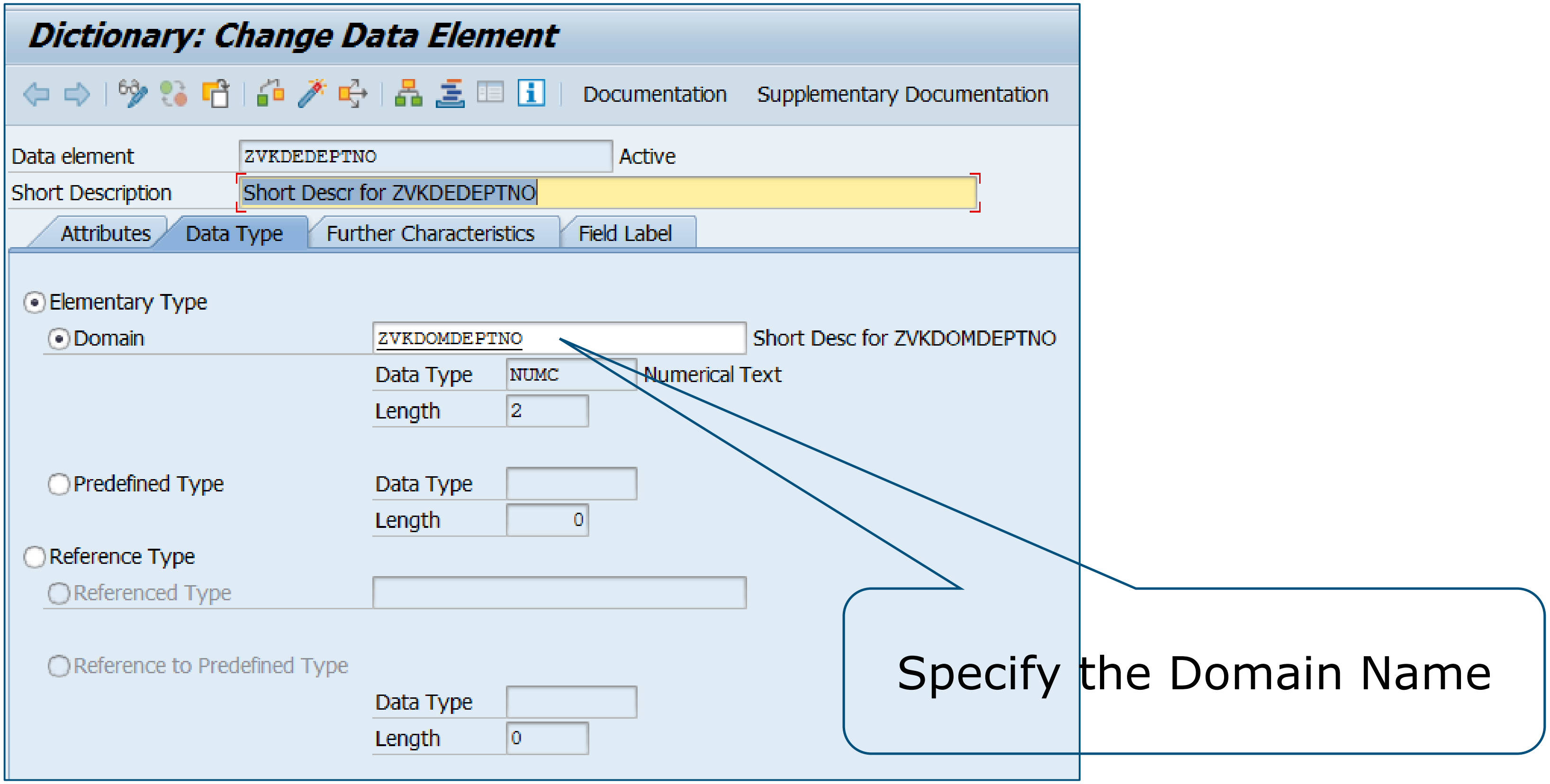
○ Table type

Choose Data Type

Specify the Data Element Name

Click on CREATE

Creating Data Element



Dictionary: Change Data Element

Data element: ZVKDEDEPTNO Active

Short Description: Short Descr for ZVKDEDEPTNO

● Elementary Type

● Domain: ZVKDOMDEPTNO Short Desc for ZVKDOMDEPTNO

Data Type: NUMC Numerical Text

Length: 2

○ Predefined Type

Data Type:

Length: 0

○ Reference Type

○ Referenced Type

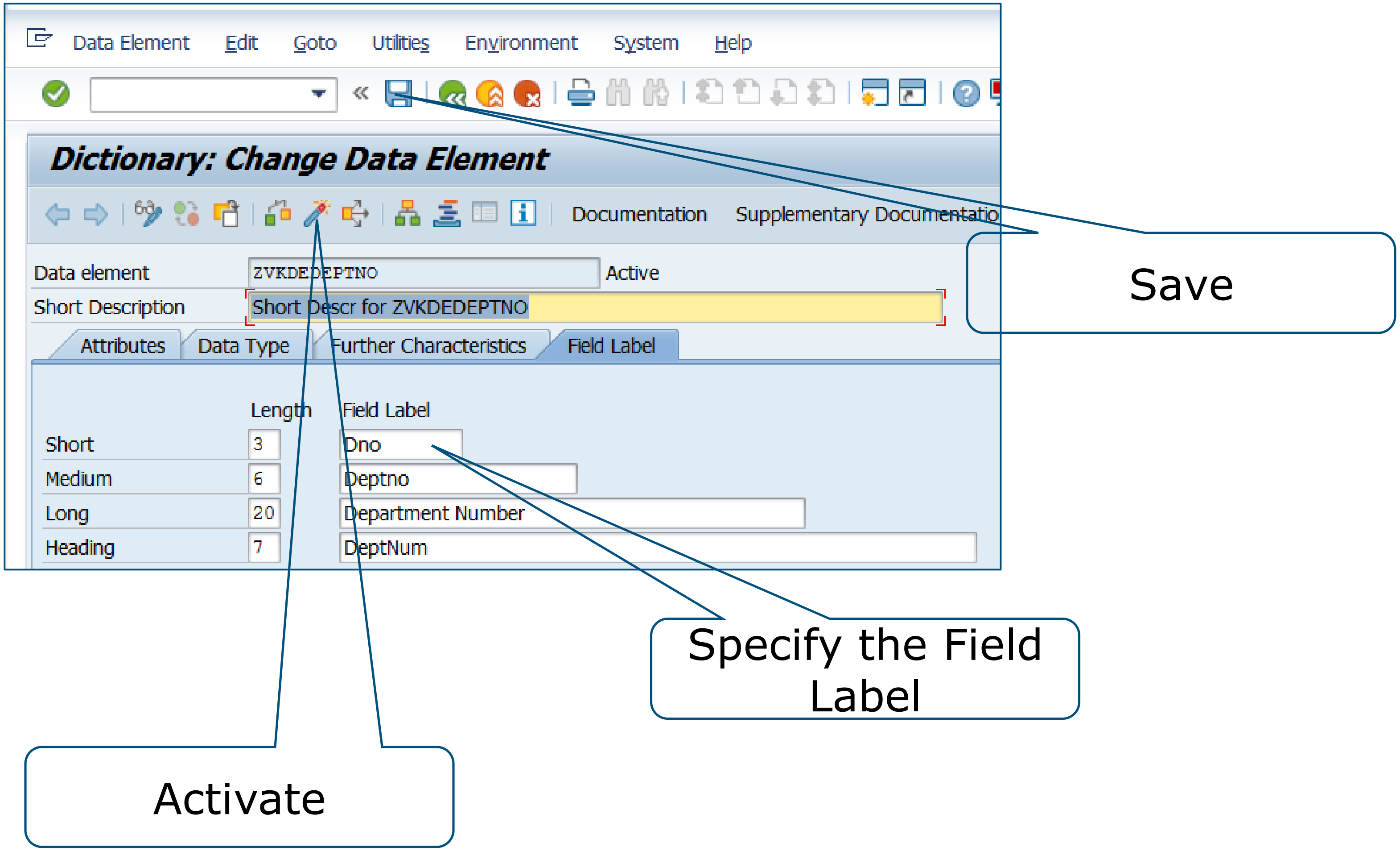
○ Reference to Predefined Type

Data Type:

Length: 0

Specify the Domain Name

Creating Data Element



Dictionary: Change Data Element

Data element: ZVKDEDEPTNO Active

Short Description: Short Descr for ZVKDEDEPTNO

	Length	Field Label
Short	3	Dno
Medium	6	Deptno
Long	20	Department Number
Heading	7	DeptNum

Save

Specify the Field Label

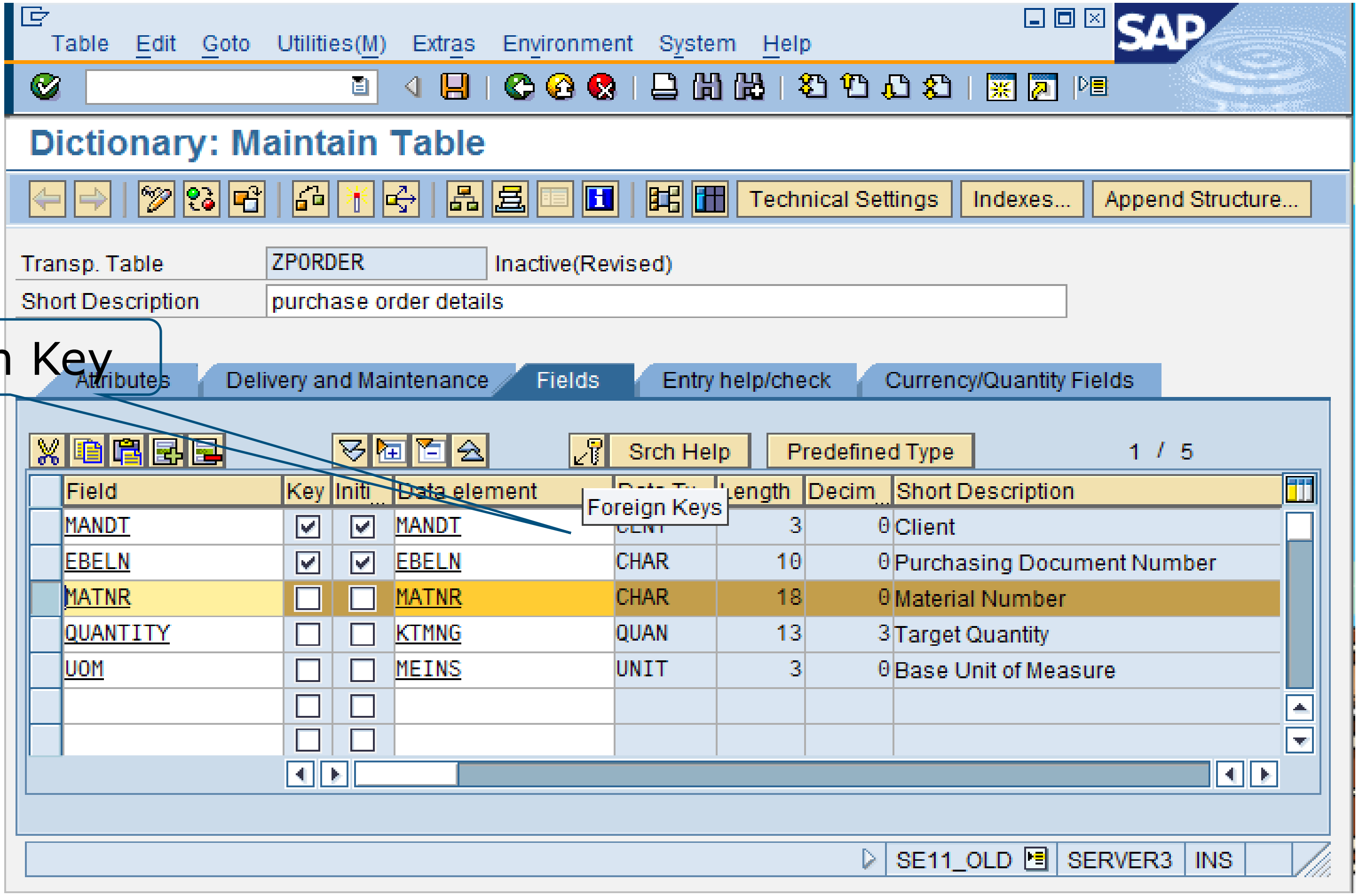
Activate

Demo

Create a simple table based data elements and domain

Tables – Foreign Keys (Contd.).

Creating Foreign Keys



Foreign Key

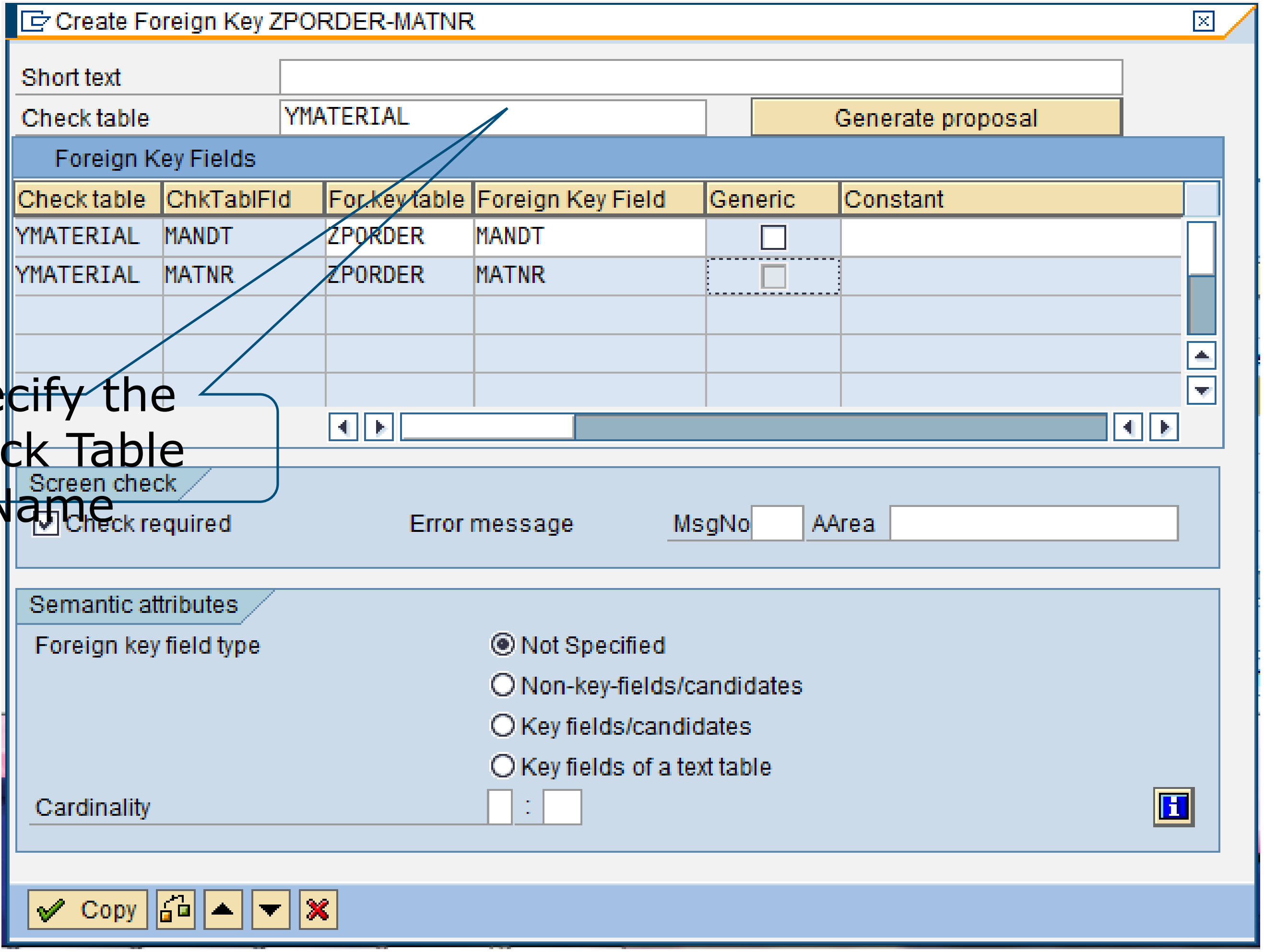
Dictionary: Maintain Table

Transp. Table	ZPORDER Inactive(Revised)
Short Description	purchase order details

1 / 5

Field	Key	Initi...	Data element	Data Ty...	Length	Decim...	Short Description
MANDT	☑	☑	MANDT	CLNT	3	0	Client
EBELN	☑	☑	EBELN	CHAR	10	0	Purchasing Document Number
MATNR	□	□	MATNR	CHAR	18	0	Material Number
QUANTITY	□	□	KTMNG	QUAN	13	3	Target Quantity
UOM	□	□	MEINS	UNIT	3	0	Base Unit of Measure
	□	□					
	□	□					

Tables – Foreign Key (Contd.).



Create Foreign Key ZPORDER-MATNR

Short text	
Check table	YMATERIAL

Foreign Key Fields

Check table	ChkTablFld	For.key table	Foreign Key Field	Generic	Constant
YMATERIAL	MANDT	ZPORDER	MANDT	☐	
YMATERIAL	MATNR	ZPORDER	MATNR	☐	

Screen check

☑ Check required

Error message MsgNo AArea

Semantic attributes

Foreign key field type

- ● Not Specified
- ○ Non-key-fields/candidates
- ○ Key fields/candidates
- ○ Key fields of a text table

Cardinality :

Specify the Check Table Name

Demo

Create primary key , foreign key relationship

Indexes

Copy of Database Table reduced to certain fields

Always in sorted form

Provides faster access to data records

Contains a pointer to corresponding record of actual table

Primary Index contains the key fields of the table

Primary index created automatically when table is activated

Possible to create secondary indexes

Tables - Index

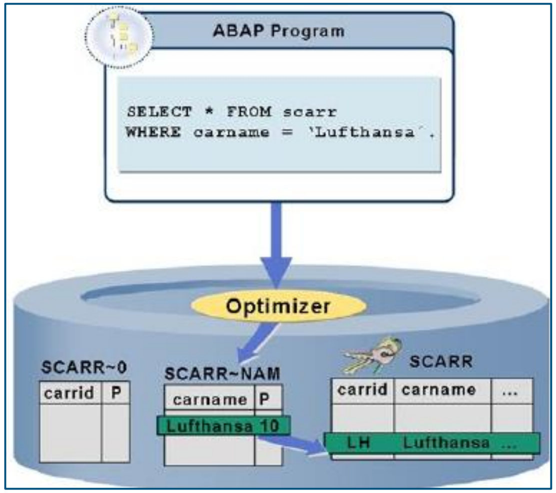
Secondary Index

- Created if the table is frequently accessed using fields which is not a part of primary key
- Index distinguished with a three place identifier
- For certain database systems, the index improves performance
- Unique Index
  - Index field has key function

Tables - Index

Secondary Indexes

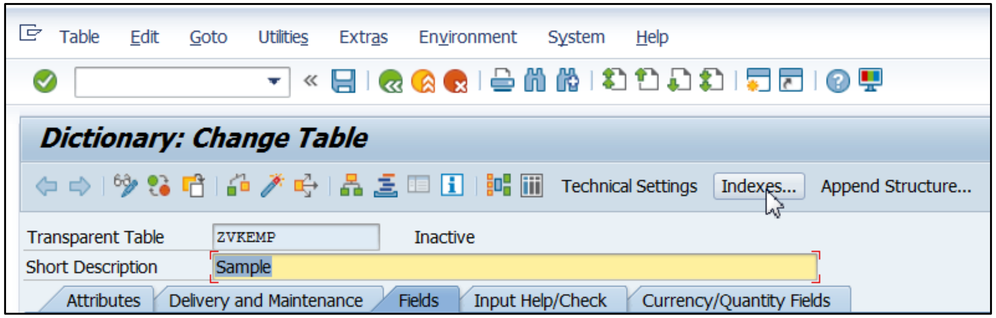
- Only Few indexes should be there in tables where the entries are frequently changed
- The database system does not use suitable indexes for selection even if there is one
- The index used depends on the optimizer used for the database system
- Creating an additional index might have side effects on performance



```
SELECT * FROM scarr
WHERE carname = 'Lufthansa'.
```

Tables - Index

Creating Secondary Index



Transparent Table ZVKEMP Inactive

Short Description Sample

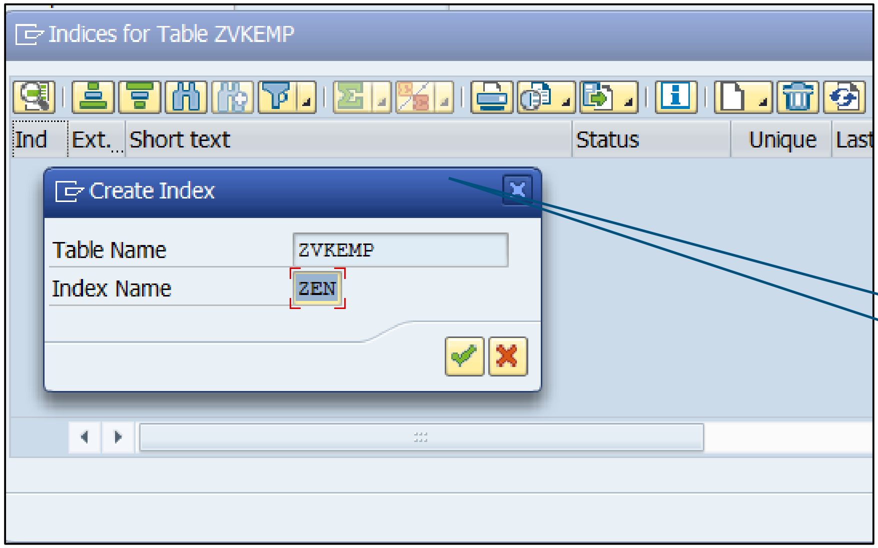
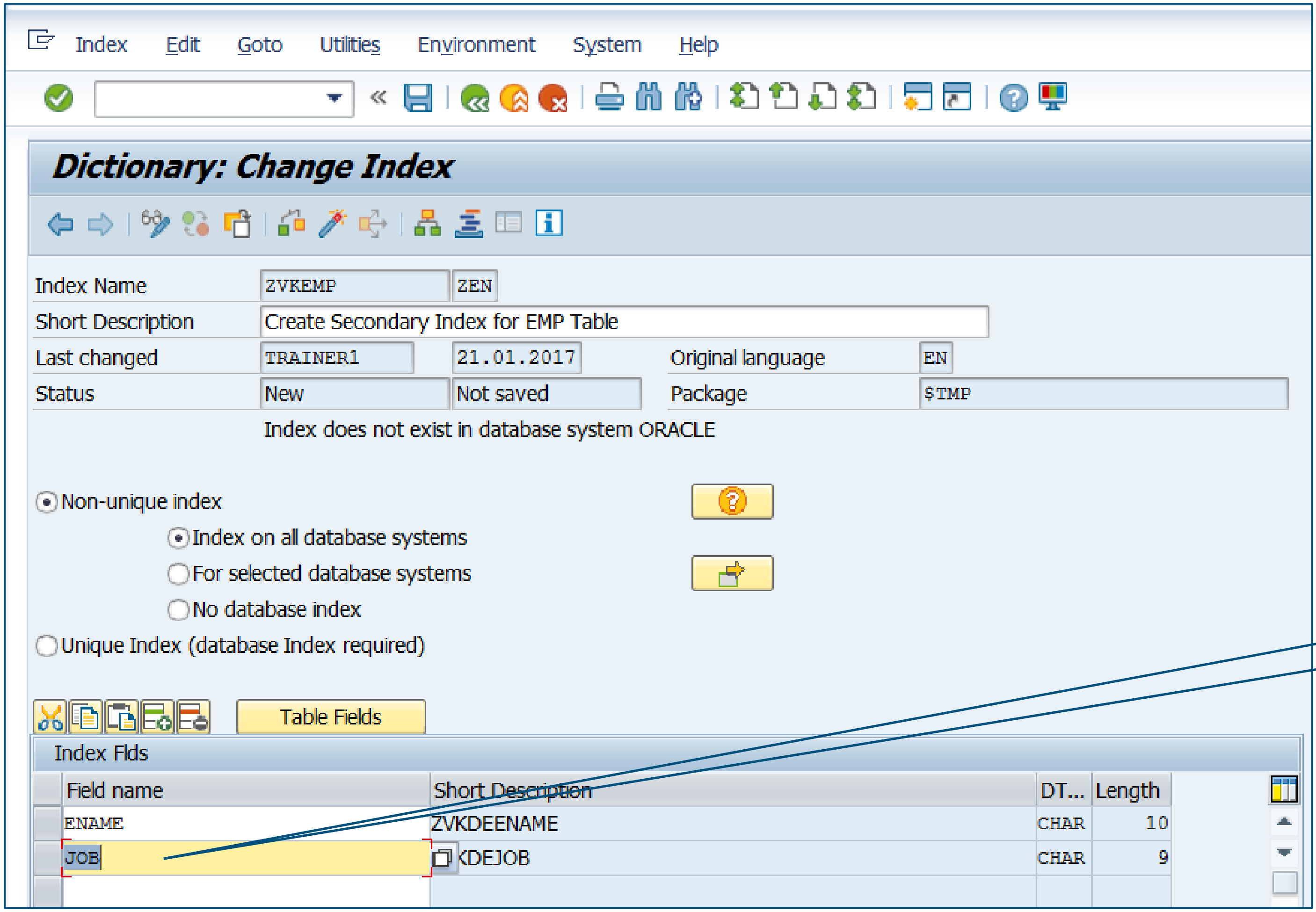


Table Name ZVKEMP

Index Name ZEN

Specify the Index Name

Tables - Index



Dictionary: Change Index

Index Name	ZVKEMP	ZEN		
Short Description	Create Secondary Index for EMP Table			
Last changed	TRAINER1	21.01.2017	Original language	EN
Status	New	Not saved	Package	$TMP

Index does not exist in database system ORACLE

● Non-unique index

- ● Index on all database systems
- ○ For selected database systems
- ○ No database index

○ Unique Index (database Index required)

Index Flds

Field name	Short Description	DT...	Length
ENAME	ZVKDEENAME	CHAR	10
JOB	ZVKDEJOB	CHAR	9

Specify the fields for index

Demo

Create Secondary Index

Tables - Index

The database optimizer decides which index on the table should be used by the database to access data records.

You must distinguish between the primary index and secondary indexes of a table.

The primary index contains the key fields of the table.

The primary index is automatically created in the database when the table is activated.

If a large table is frequently accessed such that it is not possible to apply primary index sorting, you should create secondary indexes for the table.

Standard Tables

Few Frequently Used Tables

Table Name	Description
MARA	Material Master
KNA1	Customer Master
LFA1	Vendor Master
VBAK	Sales Document : Header Data
VBAP	Sales Document : Item Data
EKKO	Purchase Document : Header
EKPO	Purchase Document : Item

Review Question

Question 1: A _________ in the dictionary has a one to one relationship with a table in the database.

Question 2: The ______ determines the table space that the table is assigned to.

Question 3: An _____ can be used to speed up the selection of data records from a table.

Summary

In this lesson, you have learnt:

- To Use Data Dictionary to maintain Database Objects
- To Work with
  - Domain
  - Data Elements