

ABAP Part I

Lesson 04: DDIC

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
 - Views
 - Table Types
 - Search Helps
 - Lock Objects



Data Dictionary

- Data Definitions are created and Managed in ABAP Dictionary
- Describes the logical structure of objects used 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



Copyright © Capgemini 2015. All Rights Reserved 3

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



Copyright © Capgemini 2015. All Rights Reserved. 4

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



Copyright © Capgemini 2015. All Rights Reserved 5

Standard Tables

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



Copyright © Capgemini 2015. All Rights Reserved

6

Types of Table

- Table Types
 - Transparent Tables
 - Pooled Tables
 - Cluster Tables



Copyright © Capgemini 2015. All Rights Reserved

7

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



Copyright © Capgemini 2015. All Rights Reserved. 8

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



Copyright © Capgemini 2015. All Rights Reserved 9

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



Copyright © Capgemini 2015. All Rights Reserved 10

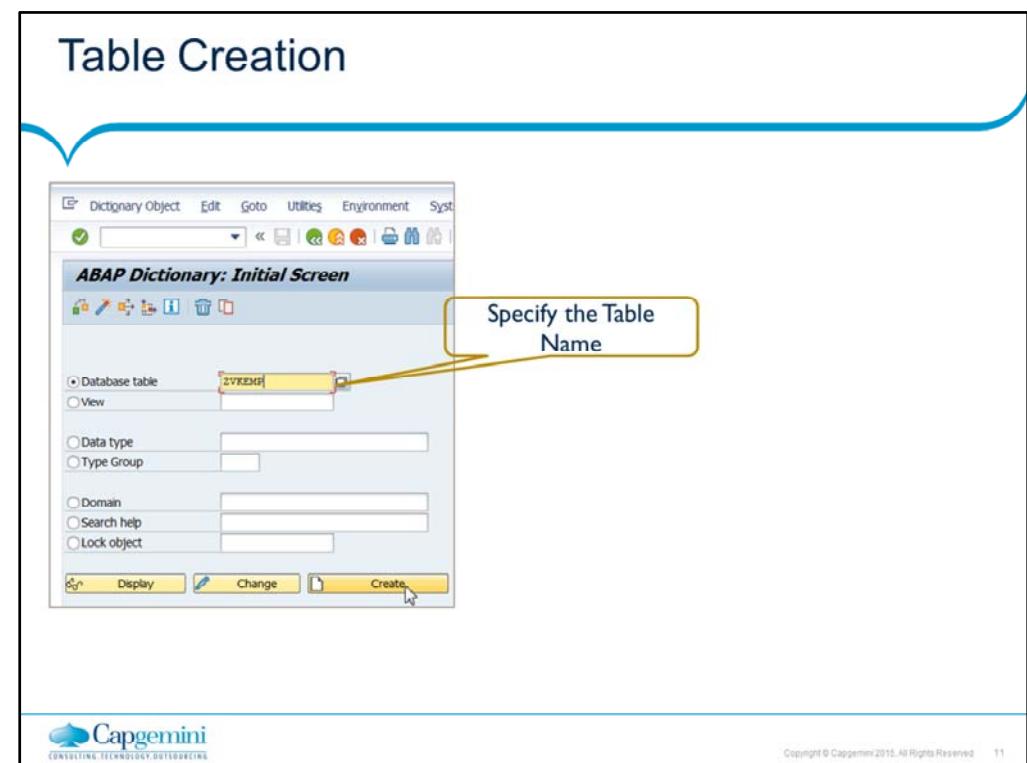
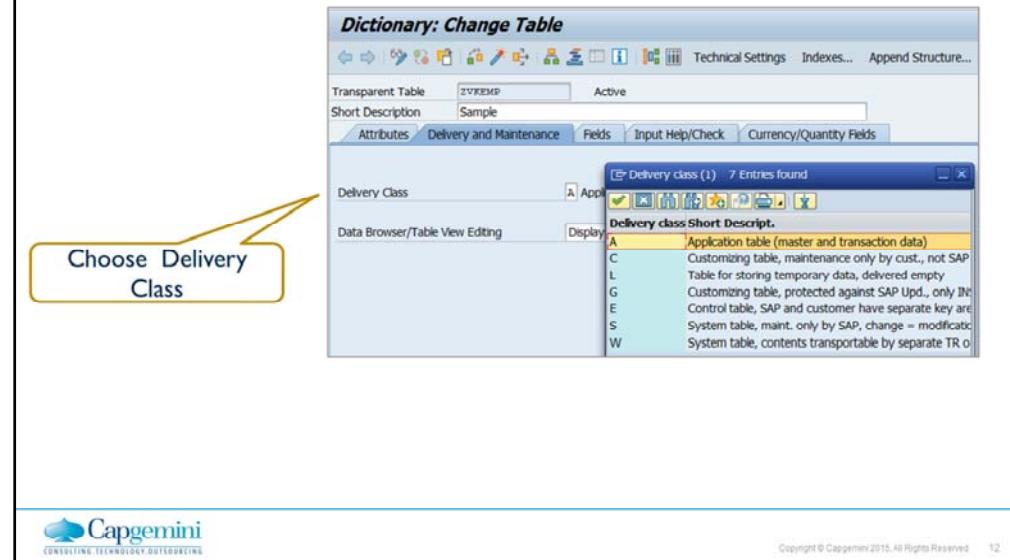


Table Creation (Contd.).



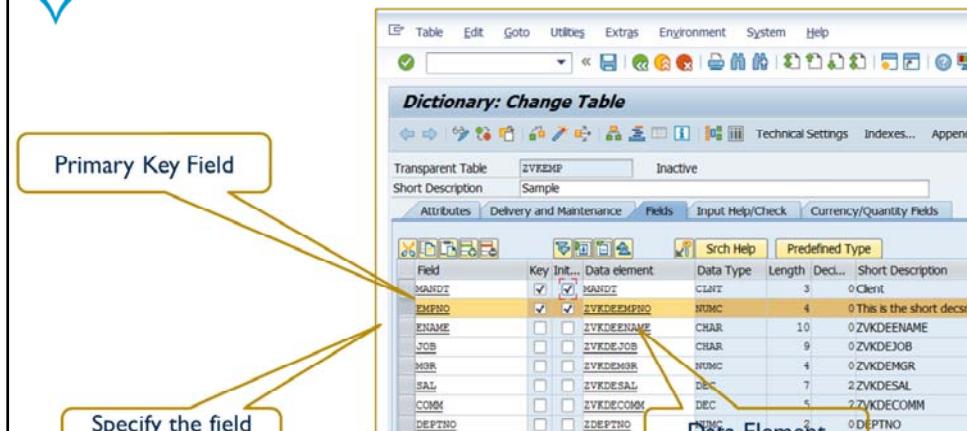
Delivery Classes

- 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.).



Copyright © Capgemini 2015. All Rights Reserved 13

Table Creation - Fields



Tables Creation (Contd.).

▪ 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



Copyright © Capgemini 2015. All Rights Reserved 15

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
 - CURRE

- Reference Fields should of Type

- UNIT
 - CUKY

- Reference Fields can be in the same table or another table.



Copyright © Capgemini 2015. All Rights Reserved 17

Demo

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



Tables – Technical Settings

Dictionary: Define Technical Settings

Name: EVKEMP (Transparent Table)

Short Descript.: Sample

Last Changed: TRAINER1 (16.01.2017)

Status: Revised (Saved)

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
- Fully Buffered

Number of Key Fields:

Log Data Changes:

Writes only with JAVA:

Capgemini CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 19

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



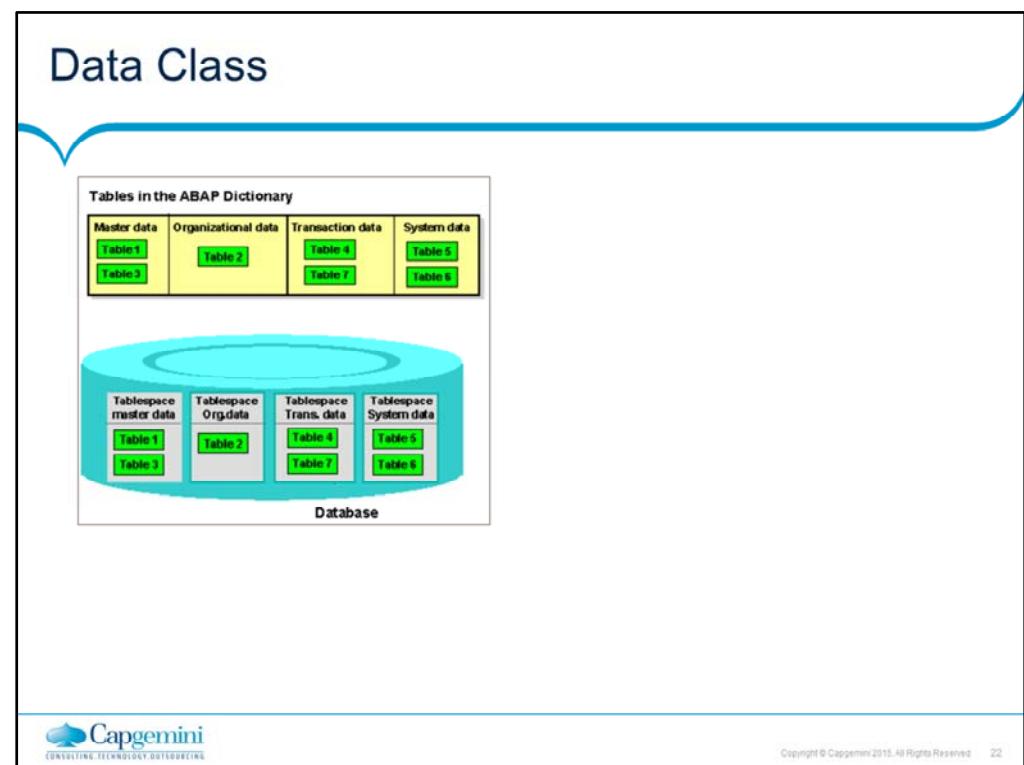
Copyright © Capgemini 2015. All Rights Reserved 20

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.

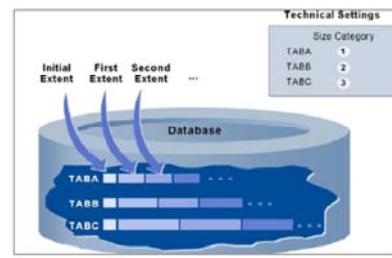


Copyright © Capgemini 2015. All Rights Reserved 21



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.



Copyright © Capgemini 2015. All Rights Reserved 23

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

You can choose a size category from 0 to 4.

You are setting the initial extents and the next extents as well as the maximum number of extents for this table

An extent is the amount of space allocated for a table

An initial extent is the amount of space allocated when the table is created

When you create a table, the system reserves initial space (an initial extent) in the database. If more space is required at a later time due to data entries, additional memory is added depending on the selected size category.

Correctly assigning a size category therefore ensures that you do not create a large number of small extents. It also prevents storage space from being wasted when creating extents that are too large.

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.



Copyright © Capgemini 2015. All Rights Reserved 24

The parameter rec/client can have the following settings:

rec/client = ALL All clients should be logged

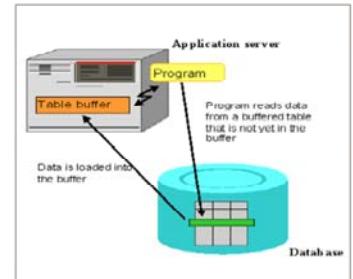
rec/client = 000 ... Only the specified clients should be logged

rec/client = OFF Logging is not enabled on this system

The data modifications are logged independently of the update. You can display the logs with the Transaction Table History (SCU3).

Table - Buffering

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



Copyright © Capgemini 2015. All Rights Reserved 25

The R/3 system manages and synchronizes the buffers on the individual application servers.

If an application program accesses data of a table, the database interface determines whether this data lies in the application server.

If this is the case, the data is read directly from the buffer.

If the data is not in the buffer of the application server, it is read directly from the database and loaded into the buffer

The buffer can therefore satisfy the next access to this data.

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



Copyright © Capgemini 2015. All Rights Reserved 26

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.



Copyright © Capgemini 2015. All Rights Reserved 27

Tables - Buffering

- Buffering Tables

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



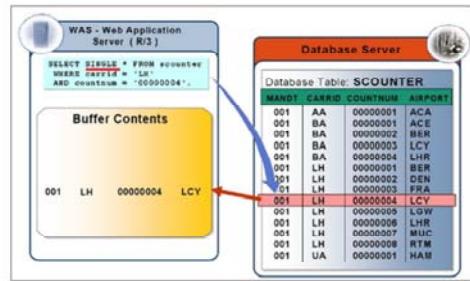
Copyright © Capgemini 2015. All Rights Reserved 28

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.



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.

The diagram illustrates the concept of full buffering. On the left, a 'WAS - Web Application Server (R/3)' window displays a SQL query result:

```
SELECT * FROM scountry
WHERE country = 'LH'
AND countnum = '00000004'.
```

The 'Buffer Contents' section shows the following data:

	MANDT	CARRIER	COUNTRYNAME	AIRPORT
001	AA	00000001	ACA	
001	BA	00000001	ACE	
001	BA	00000002	AFR	
001	BA	00000002	BER	
001	BA	00000002	CDY	
001	BA	00000004	LHR	
001	BA	00000004	BER	
001	LH	00000002	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	

On the right, a 'Database Server' window shows the 'SCOUNTER' table:

MANDT	CARRIER	COUNTRYNAME	AIRPORT
001	AA	00000001	ACA
001	BA	00000001	ACE
001	BA	00000002	AFR
001	BA	00000002	BER
001	BA	00000002	CDY
001	BA	00000004	LHR
001	BA	00000004	BER
001	LH	00000001	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

A red arrow points from the WAS query result to the Database Server table, indicating that the entire table is loaded into memory.

Fully buffering is also advisable for tables having frequent accesses to records that do not exist.

Since all the records of the table reside in the buffer, it is already clear in the buffer whether or not a record exists.

The data records stored in the buffer sorted by table key.

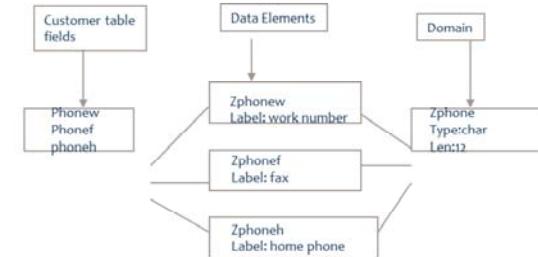
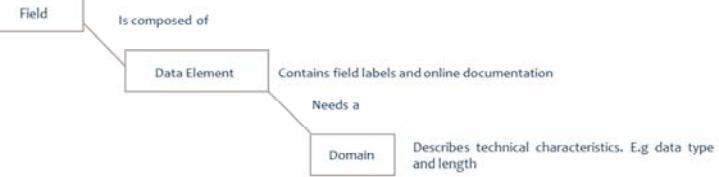
Generic Buffering

- 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.



Copyright © Capgemini 2015. All Rights Reserved 32

Data elements , Domain and fields

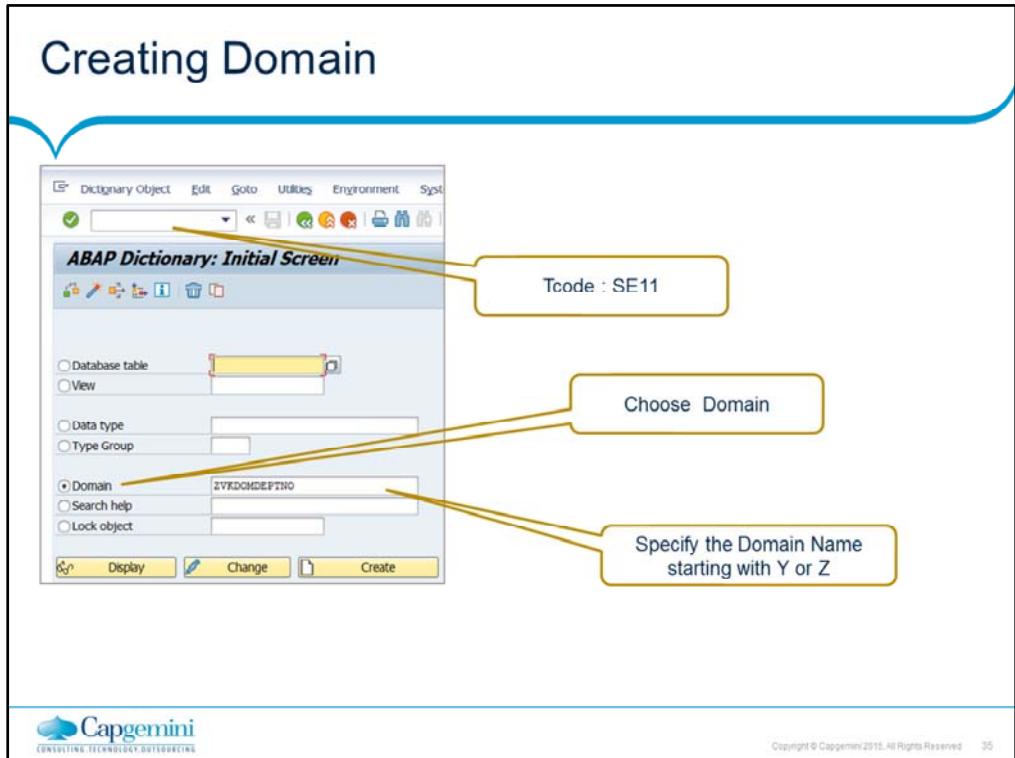


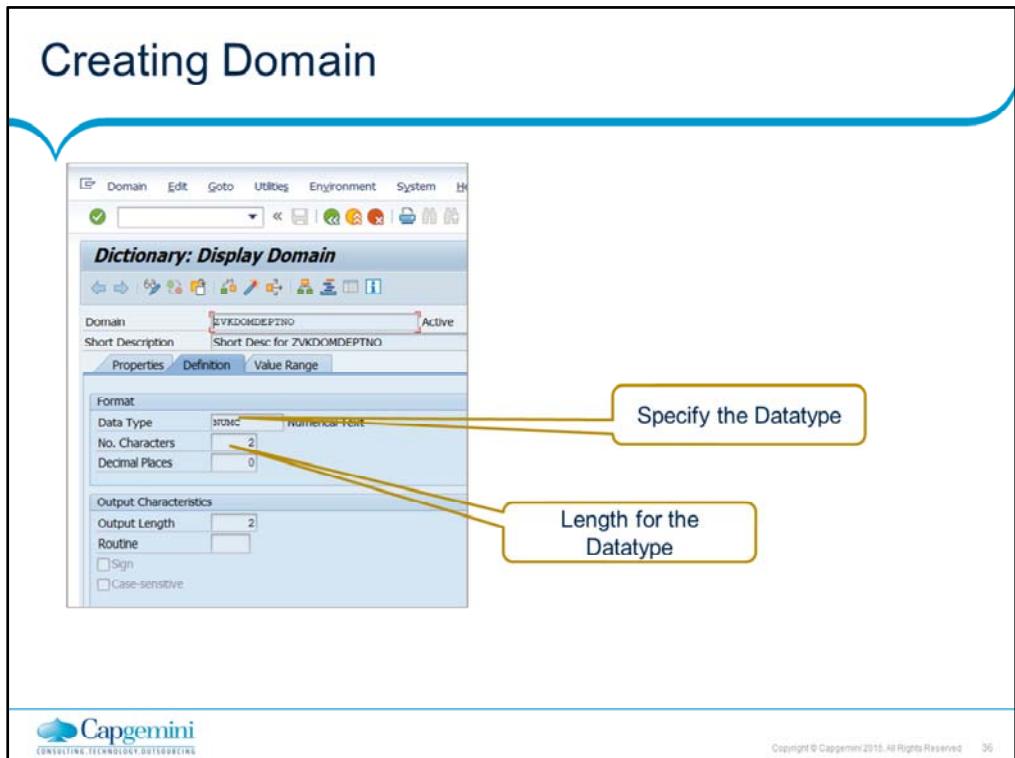
Domain

- 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



Copyright © Capgemini 2015. All Rights Reserved 34





Creating Domain (Contd.).

- Save the Domain
- Specify the Package
- Activate
- Domain is ready and can be attached to a Data Element

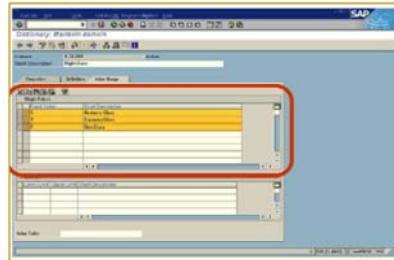


Copyright © Capgemini 2015. All Rights Reserved 37

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.



Copyright © Capgemini 2015. All Rights Reserved 38

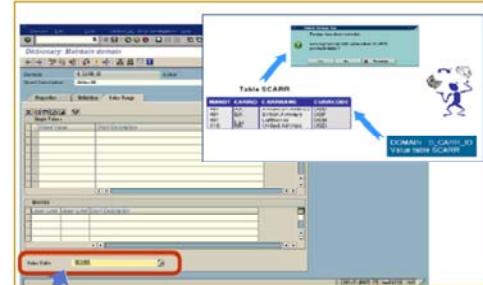
The domain describes the value range of a field by specifying its data type and field length. If only a limited set of values is allowed, they can be defined as fixed values. Specifying fixed values causes the value range of the domain to be restricted by these values. Fixed values are immediately used as check values in screen entries. There is also an F4 help.

Fixed values can either be listed individually or defined as an interval.

Fixed values are only checked in screens. No check is made when data records are inserted in a table by an ABAP program.

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.



Copyright © Capgemini 2015. All Rights Reserved 39

The value range of a field can also be defined by specifying a value table in the domain.

In contrast to fixed values, however, simply specifying a value table does not cause the input to be checked. There is no F4 help either.

If you enter a value table, the system can make a proposal for the foreign key definition.

A value table only becomes a check table when a foreign key is defined.

If you refer to a domain with a value table in a field, but no foreign key was defined at field level, there is no check

Domain

- Input Check valid for few data types
 - Value Range
 - CHAR
 - NUMC
 - Fixed Values
 - CHAR
 - NUMC
 - DEC
 - INT1
 - INT2
 - INT4



Copyright © Capgemini 2015. All Rights Reserved 40

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



Copyright © Capgemini 2015. All Rights Reserved 41

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



Copyright © Capgemini 2015. All Rights Reserved 42

Creating Data Element

■ Go to Tcode : SE11

The screenshot shows the ABAP Dictionary: Initial Screen. A callout points to the 'Data type' radio button, which is selected. Another callout points to the text input field containing 'ZVKDEDEPTNO1', which is highlighted in yellow. A third callout points to the 'Create' button at the bottom right of the screen.

Choose Data Type

Specify the Data Element Name

Click on CREATE

ABAP Dictionary: Initial Screen

ZVKDEDEPTNO1

Data type

Create

Copyright © Capgemini 2015. All Rights Reserved 43

Creating Data Element

Dictionary: Change Data Element

Data element **ZVKDEDEPTNO** Active
Short Description **Short Descr for ZVKDEDEPTNO**

Attributes Data Type Further Characteristics Field Label

Elementary Type
 Domain **ZVKDOMDEPTNO** Short Desc for ZVKDOMDEPTNO
Data Type **NUMC** Numerical Text
Length **2**

Predefined Type Data Type Length **0**

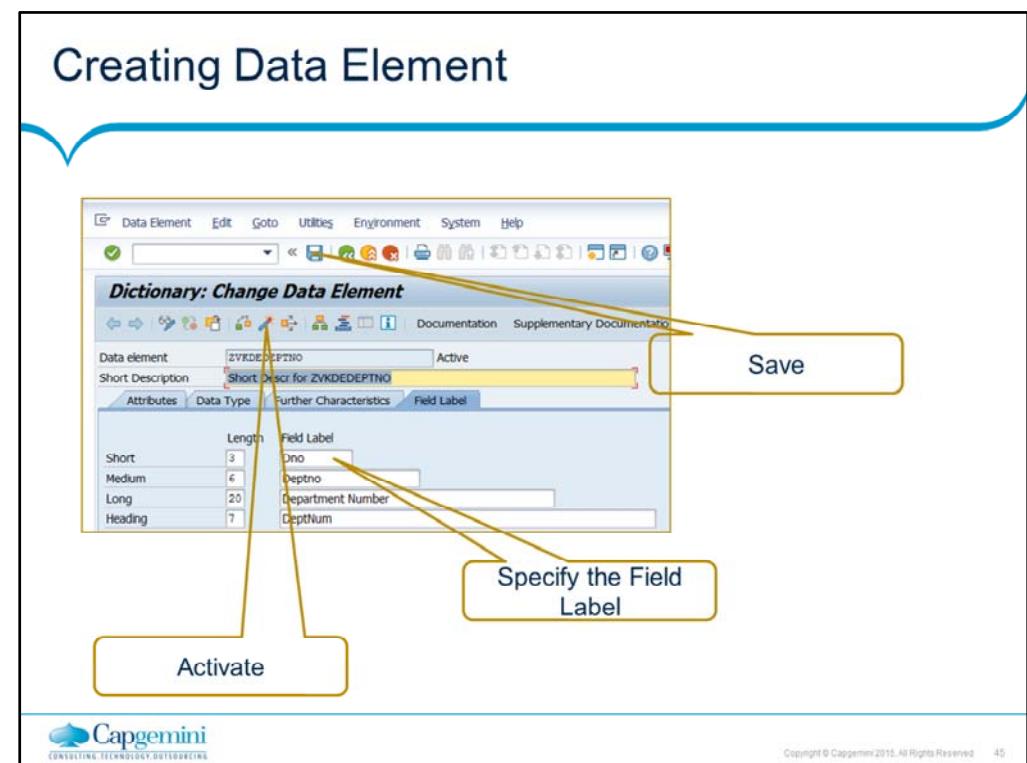
Reference Type Referenced Type

Reference to Predefined Type Data Type Length **0**

Specify the Domain Name

 Capgemini CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 44



Demo

- Create a simple table based data elements and domain



Copyright © Capgemini 2015. All Rights Reserved 46

Tables (Contd.).

▪ Pooled Tables

- Many-to-One Relationship with the table in Database
- SAP Proprietary Construct
- Stored in a Table Pool
 - Table Pools Hold large number of small Tables
- When activated, a single table is created in database
- Define Pooled tables within R/3 and assign them to the table pool



Tables (Contd.).

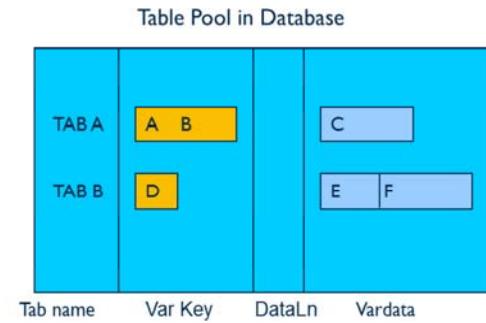
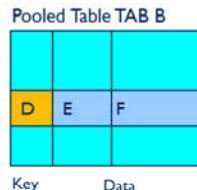
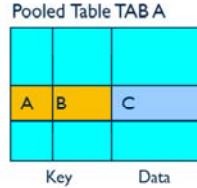
- Holds CUSTOMIZING data
 - Codes, Field Validations, number ranges, parameters
 - Country Code table, Exchange Rate Table, etc...
- Data in Customizing Table is set by Functional Consultant during the initial Implementation



Copyright © Capgemini 2015. All Rights Reserved 48

Pooled Tables

- Definition of table pool contains 2 Key fields
 - Tabname
 - Varkey (Contains Entry from all key fields)



Cluster Table

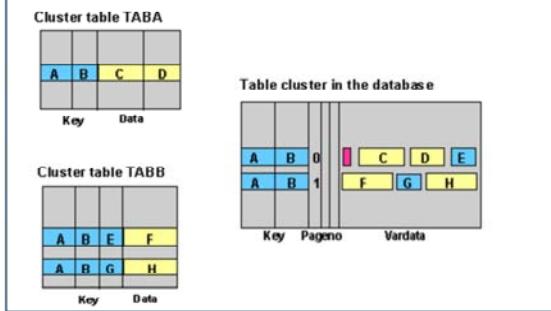
- Many-to-One relationship with table in database
- Cluster tables are stored in Table Cluster in Database
- SAP Proprietary
- Used when the tables have a part of Primary Key in common
- Data accessed Simultaneously
- Contain Fewer tables than Pool tables
- In a single I/O, all the related rows in a cluster table are retrieved
- Reduces the no. of Database reads and improves Performance



Copyright © Capgemini 2015. All Rights Reserved 50

Cluster Table

- The records of all cluster tables with the same key are stored under one key in the assigned table cluster. The values of the key fields are stored in the corresponding key fields of the table cluster.



Demo

- Create Cluster table and Pool table



Pooled and Clustered table

- Restrictions

- Secondary Indexes cannot be created
- Cannot Use ABAP/4 Constructs
 - Select DISTINCT
 - GROUP BY
- Cannot Use Native SQL
- Cannot Specify Field names in ORDER BY except for Primary Key



Copyright © Capgemini 2015. All Rights Reserved 53

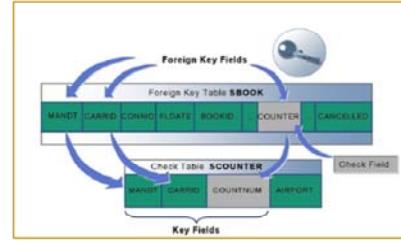
Tables

- Foreign Keys

- Defines Relationship between Tables
 - CHECK Table
 - FOREIGN Key Table

- Create Value Checks for input fields

- Link several tables in a view or a Lock Object



Copyright © Capgemini 2015. All Rights Reserved 54

A foreign key creates a link between two tables T1 and T2. Every primary key field from T2 (check table) is assigned a field from table T1 (foreign key field). The fields from T1 assigned to primary key fields are marked as foreign key fields.

The most important function of the foreign key is the support of data integrity. The foreign key fields can only accept values which appear in the primary key of the check table. During input the values of the foreign key fields can thus be checked against the entries of the assigned key fields of the check table.

Foreign keys are used to ensure that the data is consistent. Data that has been entered is checked against existing data to ensure that it is consistent

Tables – Foreign Keys

FOREIGN Key Table T1			
Field 1	Field 2	Field 3	Field 4
Primary Key			

CHECK Table T2		
Field 5	Field 6	Field 7

Primary
Key

Tables – Foreign Keys (Contd.).

Creating Foreign Keys

The screenshot shows the SAP Dictionary: Maintain Table interface for table ZPORDER. The table has a short description of "purchase order details". The "Foreign Key" tab is selected. A yellow box highlights the "Key" column, which contains checkboxes for MANDT, EBELN, and MATNR. A yellow arrow points from the text "Creating Foreign Keys" to the "Key" column. The table structure is as follows:

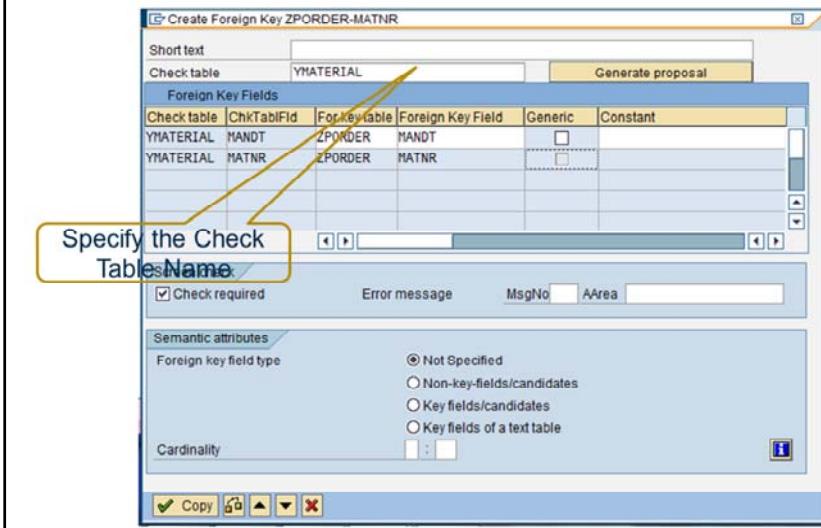
Field	Key	Type	Length	Decim	Short Description
MANDT	<input checked="" type="checkbox"/>	CHAR	3	0	Client
EBELN	<input checked="" type="checkbox"/>	CHAR	10	0	Purchasing Document Number
MATNR	<input type="checkbox"/>	CHAR	18	0	Material Number
QUANTITY	<input type="checkbox"/>	QUAN	13	3	Target Quantity
UOM	<input type="checkbox"/>	UNIT	3	0	Base Unit of Measure

At the bottom right of the SAP interface, there are buttons for SE11_OLD, SERVER3, and INS.

Capgemini CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 56

Tables – Foreign Key (Contd.).



Demo

- Create primary key , foreign key relationship



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

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



Copyright © Capgemini 2015. All Rights Reserved 60

Primary index: The primary index contains the key fields of the table and a pointer to the non-key fields of the table. The primary index is created automatically when the table is created in the database.

Secondary index: Additional indexes could be created considering the most frequently accessed dimensions of the table.

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

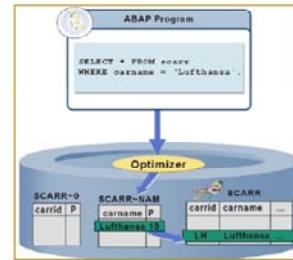


Copyright © Capgemini 2015. All Rights Reserved 61

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



Tables - Index

- Creating Secondary Index

The screenshot shows the SAP Dictionary: Change Table interface. At the top, there's a toolbar with various icons and a menu bar. Below the toolbar, the table details are shown: Transparent Table ZVKEMP, Short Description Sample, and several tabs like Attributes, Delivery and Maintenance, Fields, Input Help/Check, and Currency/Quantity Fields. A sub-menu is open under the 'Indexes...' tab. In the center, a dialog box titled 'Create Index' is displayed, showing 'Table Name' as ZVKEMP and 'Index Name' as ZER. A yellow callout box points to the 'Index Name' field with the text 'Specify the Index Name'. The background shows a list of existing indices for the table.

Dictionary: Change Table

Transparent Table ZVKEMP Inactive

Short Description Sample

Attributes Delivery and Maintenance Fields Input Help/Check Currency/Quantity Fields

Indexes...

Indices for Table ZVKEMP

Ind Ext_ Short text Status Unique Last

Create Index

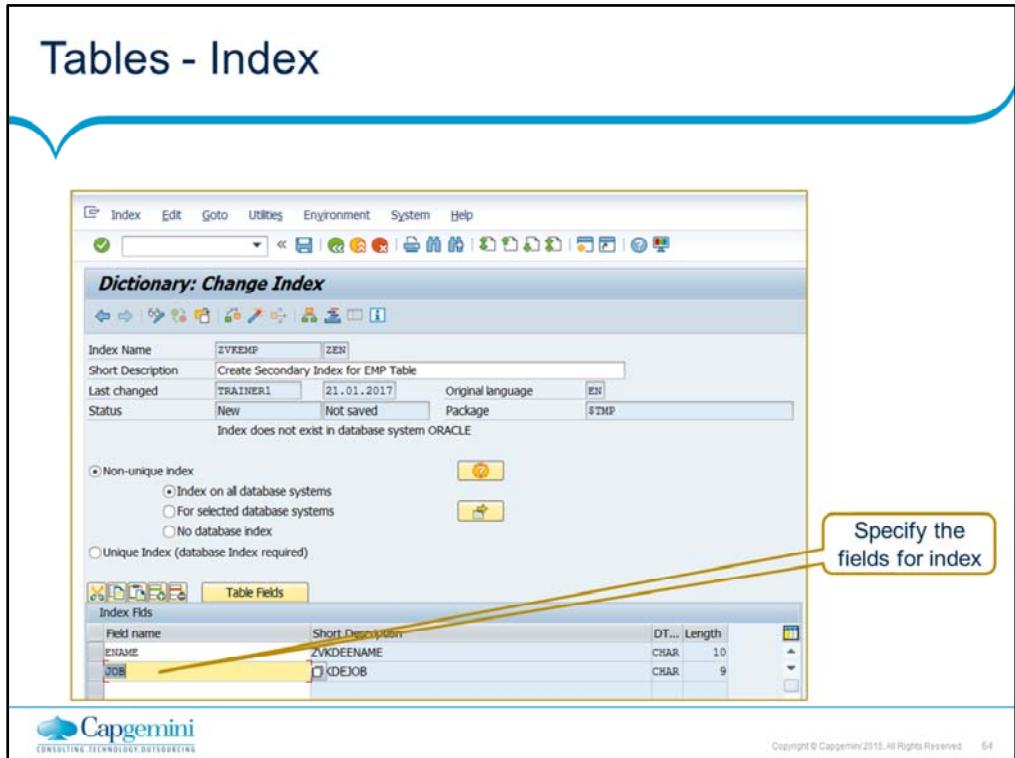
Table Name ZVKEMP

Index Name ZER

Specify the Index Name

Capgemini CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 63



Demo

- Create Secondary Index



 Capgemini
CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 65

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.



Copyright © Capgemini 2015. All Rights Reserved 66

Structures

- Structures
 - Contains Fields
 - User Defined Data Type
 - Fields can refer to
 - An elementary data type
 - Another structure
 - Table type
 - Types :
 - Flat Structures
 - Nested Structures
 - Deep Structures



Copyright © Capgemini 2015. All Rights Reserved 67

Structures (Contd.).

- Flat Structure

- In database Tables only Flat Structures can be included

Field A	Field B
---------	---------

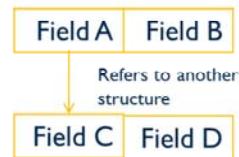
The image cannot currently be displayed.



Copyright © Capgemini 2015. All Rights Reserved. 68

Structures (Contd.).

▪ Nested Structure



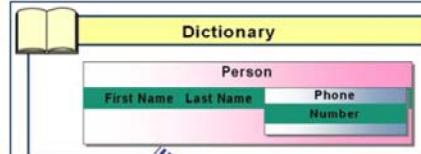
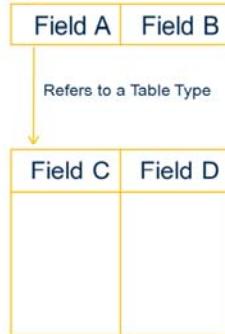
This image cannot currently be displayed.



Copyright © Capgemini 2015. All Rights Reserved 69

Structures (Contd.).

■ Deep Structure



ABAP Program

```
DATA wa_pers TYPE person.  
DATA wa_tel LIKE LINE OF wa_pers-telephone.  
  
wa_pers-name-firstname = 'Walter'.  
wa_pers-name-lastname = 'Schweizer'.  
wa_tel-number = '+49-6227-727272'.  
INSERT wa_tel INTO TABLE wa_pers-telephone.  
wa_tel-number = '+49-6227-742888'.  
INSERT wa_tel INTO TABLE wa_pers-telephone.  
  
READ TABLE wa_pers-telephone INDEX 1 INTO wa_tel.
```

Structures

- Create a Structure

The screenshot shows the ABAP Dictionary: Initial Screen. The 'Data type' radio button is selected, and the text 'ZPLT_STK' is entered into the associated input field. Two callout boxes provide instructions: one pointing to the input field with the text 'Specify the Structure Name', and another pointing to the 'Create' button with the text 'Click on CREATE'.

ABAP Dictionary: Initial Screen

Specify the Structure Name

Click on CREATE

Capgemini

Copyright © Capgemini 2015. All Rights Reserved 71

Structures

Choose Structure

Specify the fields of a Structure

The screenshot shows the SAP ABAP Dictionary interface. At the top, a modal window titled "Create Type ZFLT_STR1" has a radio button selected for "Structure". Below it, the main dictionary screen displays a structure named "ZFLT_STR" with a short description "Flat Structure". The "Components" tab is selected, showing three components: MATNR, MBRSH, and MTART, all defined as "Types" with "MATNR" as their component type. The table also includes columns for "Data-type", "Length", "Deci", and "Short Description".

Component	Type	Component-type	Data-type	Length	Deci	Short Description
MATNR	Types	MATNR	CHAR	18	0	Material number
MBRSH	Types	MBRSH	CHAR	1	0	Industry sector
MTART	Types	MTART	CHAR	4	0	Material type

Copyright © Capgemini 2015. All Rights Reserved 72

Demo

- Create a Flat, Nested and Deep Structure



Copyright © Capgemini 2015. All Rights Reserved 73

Structures (Contd.).

- Maintain Foreign Keys if required
- Save
- Activate the Structure

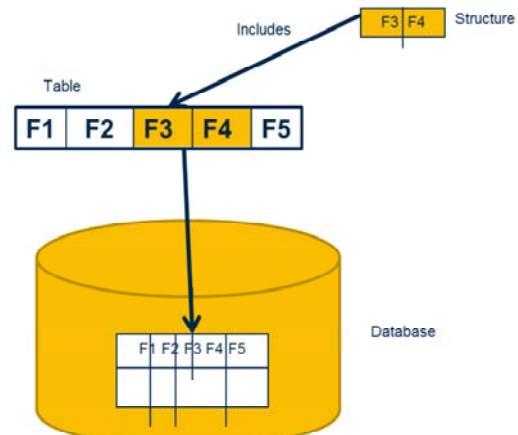


Copyright © Capgemini 2015. All Rights Reserved 74

Structures (Contd.).

- Includes

- To add fields of another structure in Tables or Structures



Structures (Contd.).

- Only Flat Structures can be included
- A Structure can be included more than once
- The field name of the structure should not be longer than 16 places



Copyright © Capgemini 2015. All Rights Reserved 76

Table Types

- Defines the structure of an Internal Table in ABAP
- Commonly used in ABAP Programs



Modifying Standard Tables

- Add fields to Standard Tables Using
 - Append Structures
 - Structure included in Single Table
 - Customizing includes
 - Structure can be included in multiple Tables
 - Is already integrated into SAP tables by SAP
 - The customer fills it with the desired additional fields



Copyright © Capgemini 2015. All Rights Reserved 78

Append Structures

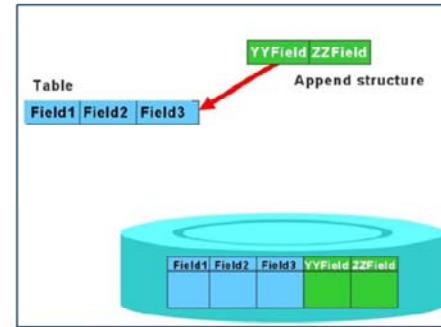
- Append Structures
 - Used for Table Enhancement
 - To Insert New Fields into Tables
 - Structure that is assigned to exactly one table
 - Created in customer namespace
 - Field Names begin with YY or ZZ
 - Created in Customer Namespace
 - Customers can create an append structure for an SAP table (without SAP preparation)
 - Multiple append structures can be used with a single SAP table



Copyright © Capgemini 2015. All Rights Reserved 79

Append Structures

- Append structures allow you to attach fields to a table without the need to modify the table itself.
- If you copy a table that has an append structure attached to it, the fields in the append structure become normal fields in the target table



the following points must be considered when using append structures:

You cannot create append structures for pool and cluster tables.

If a long field (data type **LCHR** or **LRAW**) occurs in a table, it cannot be extended with append structures. This is because long fields must always be in the last position of the field list, that is, they must be the last field of the table. No fields from an append structure may be added after them.

If you use an append structure to expand an SAP table, the field names in your append structure should be in the customer namespace, that is, they must begin with either **YY** or **ZZ**. This prevents name collisions with new fields inserted in the standard table by SAP.

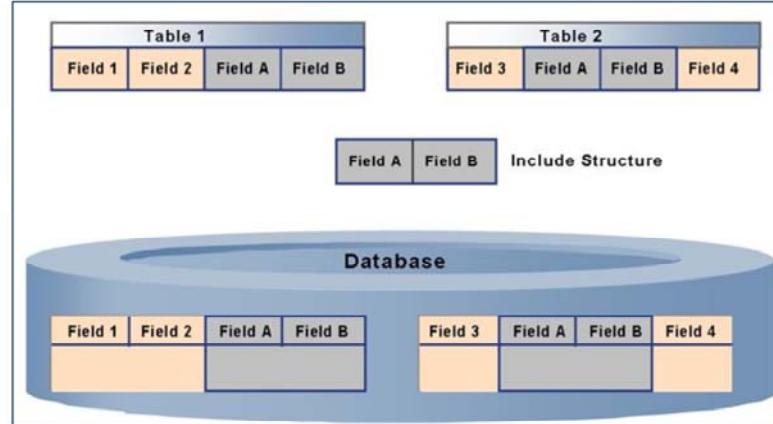
Demo

- Create an Append structure on a table created



Customizing Includes

- Structure Satisfying a Special Naming Convention CI_
- Can be included in Several Tables



Customizing Includes

- Some of the tables and structures delivered with the R/3 standard contain special include statements: These are known as **Customizing includes**
- Customizing includes are created by SAP, but the customer supply the fields for the include.
- Customizing includes begin with CI_ and is part of the customer namespace
- One Customizing include can be inserted into more than one table.



Copyright © Capgemini 2015. All Rights Reserved 53

Customizing Includes

- Consider the table RKPF which uses the Customizing include. A field can be added to CI_COBL. The field becomes a part of table RKPF after the include is activated.

Dictionary: Display Table

Transparent Table: RKPF Active
Short Description: Document Header: Reservation

Attributes Delivery and Maintenance Fields Input Help/Check Currency/Quantity Fields

Field	Key	Init...	Data element	Data Type	Length	Deci...	Short Description
VFZNR			JV_PART	CHAR	10	0	Partner account number
FIPOS			FIPOS	CHAR	14	0	Commitment Item
.INCLUDE			CI_COBL	STRU	0	0	
RECID			JV_RECINDI	CHAR	2	0	Internal Recovery Indicator
FKBER			FKBER	CHAR	16	0	Functional Area
DABRZ			DABRZ	DATS	8	0	Reference date for settlement
FISTL			FISTL	CHAR	16	0	Funds Center
GEBER			BP_GEBER	CHAR	10	0	Fund
PRZNR			CO_PRZNR	CHAR	12	0	Business Process
LSTAR			LSTAR	CHAR	6	0	Activity Type
GRANT_NBR			GM_GRANT_NBR	CHAR	20	0	Grant
BUDGET_PD			FM_BUDGET_PERIOD	CHAR	10	0	FM: Budget Period

Capgemini CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 54

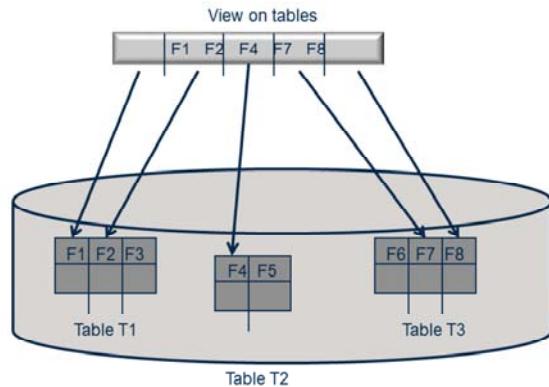
Demo

- Create an Include structure on a table created



Views

- View is data derived from one or more tables
- Used in ABAP Program for data selection
- Data is not stored physically



Views

- Structure of a View - Starting Situation

SCARR (T1)	
CARRID	CARNAME
LH	Lufthansa
DL	Delta Airline

SFLIGHT (T2)		
CARRID	CONNID	PLANETYPE
LH	401	A319
LH	402	DC-10-10
DL	106	A310-300
DL	1699	A319

Resulting sets (using Join without ON condition)				
T1~CARRID	T1~CARNAME	T2~CARRID	T2~CONNID	T2~PLANETYPE
LH	Lufthansa	LH	401	A319
LH	Lufthansa	LH	402	DC-10-10
LH	Lufthansa	DL	106	A310-300
LH	Lufthansa	DL	1699	A319
DL	Delta Airline	LH	401	A319
DL	Delta Airline	LH	402	DC-10-10
DL	Delta Airline	DL	106	A310-300
DL	Delta Airline	DL	1699	A319

Cross-product
of tables SCARR
and SFLIGHT

Structure of a View - Join Condition

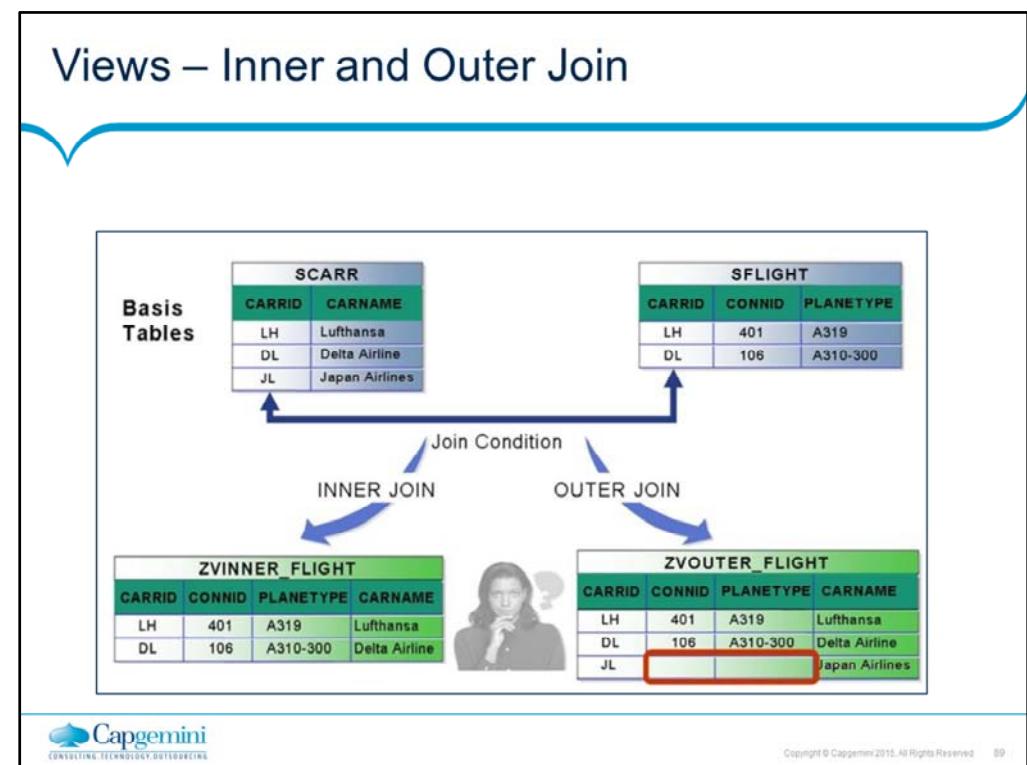
Resulting sets (using Join with ON condition)

T1~CARRID	T1~CARNAME	T2~CARRID	T2~CONNID	T2~PLANETYPE
LH	Lufthansa	LH	401	A319
LH	Lufthansa	LH	402	DC-10-10
LH	Lufthansa	DL	106	A310-300
LH	Lufthansa	DL	1699	A319
DL	Delta Airline	LH	401	A310
DL	Delta Airline	LH	402	DC-10-10
DL	Delta Airline	DL	106	A310-300
DL	Delta Airline	DL	1699	A319

Reduce
the cross-
product



Copyright © Capgemini 2015. All Rights Reserved 58



Types of Views

- Database View
- Projection View
- Maintenance View
- Help View



Database View

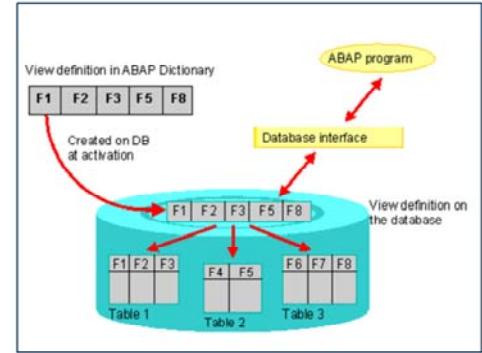
- Provides application specific view on data distributed in tables
- Created in the database
- accessed using Open SQL and Native SQL
- If there is only one table in the view , change access is possible
- Contains only transparent table
- Implements Inner Join



Copyright © Capgemini 2015. All Rights Reserved 91

Database View

- Data about an application object is often distributed on several database tables.
- A database view provides an application-specific view on such distributed data.
- Database views are defined in the ABAP Dictionary.
- A database view is automatically created in the underlying database when it is activated.



Application programs can access the data of a database view using the database interface. You can access the data in ABAP programs with both OPEN SQL and NATIVE SQL. However, the data is actually selected in the database. Database views implement an **inner join**. If the database view only contains a single table, the maintenance status can be used to determine if data records can also be inserted with the view. If the database view contains more than one table, you can only read the data. A database view may only contain transparent tables.

Demo

- Create a Database View

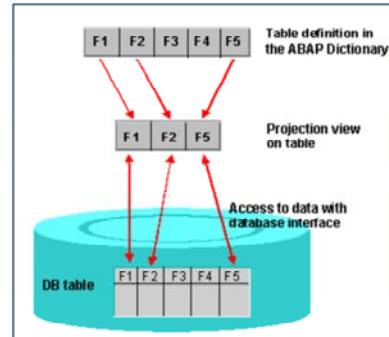


 Capgemini
CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 93

Projection View

- Used to hide fields of a table
- Contains exactly one table
- Selection conditions cannot be defined



There is no corresponding object in the database for a projection view. It is also possible to access pooled tables or cluster tables with a projection view.
The maintenance status of the view controls how the data of the table can be accessed with the projection view

Projection View

- Structure of a View - Field Selection (Projection)

SFLIGHT				
CARRID	CONNID	PRICE	CURRENCY	PLANETYPE
LH	401	666,00	EUR	A319
LH	401	666,00	EUR	DC-10-10
DL	106	611,01	USD	A319
DL	1699	422,94	USD	DC-10-10

Projection (Restriction of existing table fields)

ZVP_SFLIGHT (View)			
CARRID	CONNID	PRICE	CURRENCY
LH	401	666,00	EUR
LH	401	666,00	EUR
DL	106	611,01	USD
DL	1699	422,94	USD

Capgemini
CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 95

Demo

- Create a projection view

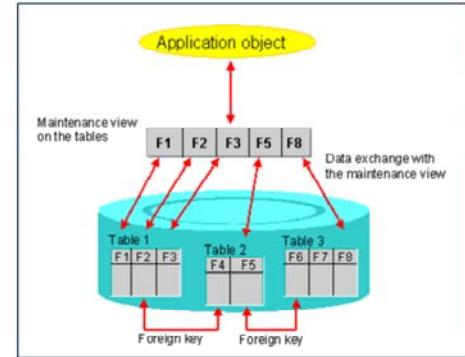


 Capgemini
CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 96

Maintenance View

- Maintenance views offer easy ways to maintain complex application objects.
- Data distributed on several tables often forms a logical unit, for example an application object, for the user.
- You want to be able to display, modify and create the data of such an application object together



Copyright © Capgemini 2015. All Rights Reserved 97

A maintenance view permits you to maintain the data of an application object together. The data is automatically distributed in the underlying database tables. The maintenance status determines which accesses to the data of the underlying tables are possible with the maintenance view. All the tables in a maintenance view must be linked with foreign keys.

Demo

- Create a Maintenance view



 Capgemini
CONSULTING TECHNOLOGY OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 98

Help view

- You have to create a help view if a view with outer join is needed as selection method of a search help.
- The selection method of a search help is either a table or a view.
- If you have to select data from several tables for the search help, you should generally use a database view as selection method.
- However, a database view always implements an inner join.
- If you need a view with outer join for the data selection, you have to use a help view as selection method.
- A help view implements an outer join, i.e. all the contents of the primary table of the help view are always displayed.

Demo

- Create a Help view



About Search Help

- Used to Display list of all possible input values for a screen field on the press of F4
- Useful when the field requires the input of a formal key
- Has to be assigned to the screen field
- Types of Search Helps
 - Elementary
 - Describes a search path
 - Defines where the data of the hitlist should be read from
 - Collective
 - Combines several elementary search helps
 - Offers Alternative search paths



Copyright © Capgemini 2015. All Rights Reserved 101

Selection Method

- Possible input values are determined at runtime by database selection
- If the values are from a single table, the corresponding table is selected as selection method
- If the values are from multiple tables, they must be linked with a view (Database or Help View) which is selected in the Selection Method



Copyright © Capgemini 2015. All Rights Reserved 102

Search Help Parameters

- Defines the fields of selection method that should be used in input help
- Data element should be assigned to Search Help Parameter
- Import and Export Parameters
 - Import Parameter – Information from the screen is copied to help process
 - Export Parameter – Values from hitlist is returned to input template



Copyright © Capgemini 2015. All Rights Reserved 103

Attaching Search Help to Screen Fields

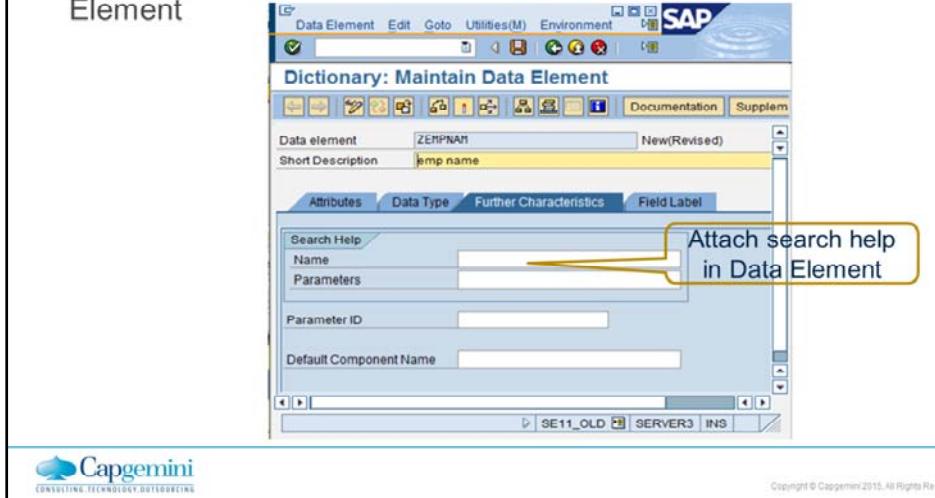
- Search helps can be attached to fields by the following ways
 - Attaching to Data Element
 - Attaching to Check Tables
 - Attaching to a table field or Structure Field
 - Attaching to Screen Fields



Copyright © Capgemini 2015. All Rights Reserved 104

Attaching to Data Element

- The search help is used by all screens that refer to this data element
- Export parameter of the search help must be assigned to Data Element



Demo

- Create an Elementary search help and Collective search help



Copyright © Capgemini 2015. All Rights Reserved 106

Attaching to Screen Field

- A search help can be directly assigned to Screen field by
 - Specifying name of search help in Screen Painter Attributes
 - Specifying the name of Search help in ABAP reports in PARAMETERS or SELECT-OPTIONS statement using AS SEARCH PATTERN.



Copyright © Capgemini 2015. All Rights Reserved 107

Creating Search Helps (Contd.).

- Specify
 - Import
 - Export
 - LPos
 - Spos
- Save and Activate the search help



Copyright © Capgemini 2015. All Rights Reserved 106

What are Lock Objects?

- Lock Mechanism is used by R/3 to synchronize simultaneous access to same data by several users
- Locks are set and released by calling Function Modules, which are automatically generated from the definition of lock objects
- Lock Arguments
 - Consists of Key Fields of the Tables
 - Used as input parameter in the function module for setting and releasing locks



Lock Mode

- A lock mode can be assigned to each table in the lock object
- Lock mode
 - Defines how the users can access the locked records of the table
 - Write Lock
 - Locked Data can be displayed or edited by a single user.
 - Can be requested several times from the same transaction and are processed successively
 - A request for another Exclusive lock or Shared Lock is rejected
 - Read Lock
 - More than one user can access the locked data at the same time in display mode
 - A request for another shared lock is accepted
 - Exclusive But not Cumulative
 - Can be called only once from the transaction
 - All other lock requests are rejected



Copyright © Capgemini 2015. All Rights Reserved 110

Function Modules for Lock Requests

- Activating a lock object automatically creates function modules
 - ENQUEUE_<lock Object Name> - for setting the lock
 - DEQUEUE_<lock Object Name> - for releasing the lock
- The above function modules can be called directly from SE37 or an ABAP program created in SE38
- The TCode SM12 can be used to check whether a lock has been applied
- This can be done after executing the Enqueue function



Copyright © Capgemini 2015. All Rights Reserved 111

Summary

■ In this lesson, you have learnt:

- To Use Data Dictionary to maintain Database Objects
- To Work with
 - Domain
 - Data Elements
 - Tables
 - Structures
 - Views
 - Table Types
 - Search Helps
 - Lock Objects



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.

