SAP-SYSTEMS APPLICATIONS & PRODUCTS IN DATA PROCESSING.

--CLIENT & SERVER thins and difference between them.

--R3 REAL TIME 3 TIER

--CLIENT SERVER IS A SET OF SOFTWARE COMPONENTS WHICH INCLUDES SET OF SERVICE REQUESTERS & SERVICE PROVIDERS.

--A small exe file or something request is known client.

--A single PC or machine or group of machines it acts like a service providers known as server.

--R1 – stand alone,

--r3-padb

--s/4 – simple logic, HDB

--work process

--**RICEFW** – reports,interface,conversion, enhancements,forms,workflows

ERP – Enterprise Resource Planning

--E →Huge Business Organization

--R -> Money (Fico), Manpower (HR),Material(MM),Machinery(PP),Marketing(SD),Methods(PS)…

--P ->Effective utilization and optimization of resources

ERP DEFINATION: effective utilization and optimization of resources i.e. 6m's into your org. and integrate them.

→>Reports are internal, forms are external.

ERP Process: It attempts to integrate all the departments and its subdepartments across to a company into a single system or server is known as Erp process.

ERP Products: SAP,ORACLE.

SAP Modules – Functional, technical—abap/4, Basis /admin,bi,Xi/pi

RICEF w-Reports Interface conversion enhancements forms workflow.

ECC-ERP Central Components

S4-HANA – High performance Analytical appliance.

EE-ENTERPRISE EDITION

WRITE:'WELCOME TO SAP UST' **USED FOR PRINTING THE STATEMENTS IN SAP.

CONTROL STATEMENTS

→data a type i VALUE 123.

WRITE a.

#1

*data lv_a type i VALUE 123.

```
*WRITE lv_a.
**#2
*data lv_a(20) type c VALUE 'ust hyd'.
*WRITE lv_a.
**#  3
*data :lv_a TYPE i VALUE 10,
*      lv_b TYPE i VALUE 20,
*      lv_c TYPE i.
*
*lv_c =  lv_a + lv_b.
*
*WRITE: 'the value is' , lv_c.
**#  4
*PARAMETERS:  p_a TYPE i DEFAULT 10,
*             p_b TYPE i DEFAULT 20.
*
*data  lv_c TYPE i.
*
*lv_c =  p_a + p_b.
*
*WRITE: 'the value is' , lv_c.
# 6
*
*WRITE: 'The current date is' ,  sy-datum,
*       'Time : '  , sy-uzeit,
*       'sap user', sy-uname.



*
**CONSTANTS CS_ABC type i VALUE 210.
**
**cs_abc = 234.
**WRITE cs_abc.
*
```

```
*

****# 7

data: begin of b1,  " FIELD STRING
       empid      type i value  123,
       empname(20) type c value 'UST',
     end of b1.
data: begin of b2,
       empid       type i,
       empaddrs(30) type c,
       empname(20)  type c,

     end of b2.
     move b1 to b2.
*move-corresponding b1 to b2.
write :/ b2-empid , b2-empname , b2-empaddrs.
```

JOINS :

The joins are used to create a link between the tables.

```
*TYPES: BEGIN OF ls_tab,
*        vbeln TYPE vbeln_va,
*        audat TYPE audat,
*        vbtyp TYPE vbtypl,
*        trvog TYPE trvog,
*        auart TYPE auart,
*      END OF ls_tab.
*
*DATA : lt_tab TYPE STANDARD TABLE OF ls_tab,
*      wa_tab TYPE ls_tab.
*
*SELECT-OPTIONS s_vbeln FOR wa_tab-vbeln.
*
*START-OF-SELECTION.
*
```

```
*  SELECT vbeln audat vbtyp trvog auart
*    INTO TABLE lt_tab
*    FROM vbak
*    WHERE vbeln IN s_vbeln.
*
*  IF sy-subrc EQ 0.
*    WRITE : 'sales order info ', sy-dbcnt.
*  ELSE.
*    WRITE : 'sales order records are not found ', sy-dbcnt.
*  ENDIF.
*
*END-OF-SELECTION.
*  LOOP AT lt_tab INTO wa_tab.
*    WRITE : / wa_tab-vbeln,
*    wa_tab-audat,
*    wa_tab-vbtyp,
*    wa_tab-trvog,
*    wa_tab-auart.
*  ENDLOOP.
```

*****# 2 two tables**

```
*TYPES: BEGIN OF ls_tab,
*        vbeln TYPE vbeln_va,  "  vbak
*        audat TYPE audat,
*        vbtyp TYPE vbtypl,
*        trvog TYPE trvog,
*        auart TYPE auart,
*
*        posnr TYPE posnr_va,  "   vbap
*        matnr TYPE matnr,
*        matwa TYPE MATWA,
*      END OF ls_tab.
*
*DATA : lt_tab TYPE STANDARD TABLE OF ls_tab,
*      wa_tab TYPE ls_tab.
```

```
*
*SELECT-OPTIONS s_vbeln FOR wa_tab-vbeln.
*START-OF-SELECTION.
*  SELECT vbak~vbeln
*    vbak~audat
*    vbak~vbtyp
*    vbak~trvog
*    vbak~auart
*
*     vbap~posnr
*     vbap~matnr
*    vbap~matwa
*
*    INTO TABLE lt_tab
*    FROM vbak
*
*    INNER JOIN vbap
*    ON vbak~vbeln = vbap~vbeln
*
*    WHERE vbak~vbeln IN s_vbeln.
*
*  IF sy-subrc EQ 0.
*    WRITE : 'sales order info ', sy-dbcnt.
*  ELSE.
*    WRITE : 'sales order records are not found ', sy-dbcnt.
*  ENDIF.
*
*END-OF-SELECTION.
*  LOOP AT lt_tab INTO wa_tab.
*    WRITE : / wa_tab-vbeln,
*     wa_tab-posnr,
*    wa_tab-matnr,
*    wa_tab-audat,
*    wa_tab-vbtyp,
```

```
*    wa_tab-trvog,
*    wa_tab-auart,
*      wa_tab-matwa.
*  ENDLOOP.
```

****# 2 two tables with alias**

```
*TYPES: BEGIN OF ls_tab,
*        vbeln TYPE vbeln_va,  "  vbak
*        audat TYPE audat,
*        vbtyp TYPE vbtypl,
*        trvog TYPE trvog,
*        auart TYPE auart,
*
*        posnr TYPE posnr_va,  "   vbap
*        matnr TYPE matnr,
*        matwa TYPE matwa,
*      END OF ls_tab.
*
*DATA : lt_tab TYPE STANDARD TABLE OF ls_tab,
*      wa_tab TYPE ls_tab.
*
*SELECT-OPTIONS s_vbeln FOR wa_tab-vbeln.
*
*START-OF-SELECTION.
*
*  SELECT so_h~vbeln
*    so_h~audat
*    so_h~vbtyp
*    so_h~trvog
*    so_h~auart
*
*     so_i~posnr
*     so_i~matnr
*    so_i~matwa
*
```

```
*    INTO TABLE lt_tab
*    FROM vbak AS so_h
*
*    INNER JOIN vbap AS so_i
*    ON so_h~vbeln = so_i~vbeln
*
*    WHERE so_h~vbeln IN s_vbeln.
*
*  IF sy-subrc EQ 0.
*    WRITE : 'sales order info ', sy-dbcnt.
*  ELSE.
*    WRITE : 'sales order records are not found ', sy-dbcnt.
*  ENDIF.
*
*END-OF-SELECTION.
*  LOOP AT lt_tab INTO wa_tab.
*    WRITE : / wa_tab-vbeln,
*     wa_tab-posnr,
*    wa_tab-matnr,
*    wa_tab-audat,
*    wa_tab-vbtyp,
*    wa_tab-trvog,
*    wa_tab-auart,
*      wa_tab-matwa.
*
*  ENDLOOP.
```

**# 3 leftouter join

```
*TYPES: BEGIN OF ls_tab,
*        vbeln TYPE vbeln_va,  "  vbak
*        audat TYPE audat,
*        vbtyp TYPE vbtypl,
*        trvog TYPE trvog,
*        auart TYPE auart,
*
```

```
*        posnr TYPE posnr_va,  "   vbap
*        matnr TYPE matnr,
*        matwa TYPE matwa,
*      END OF ls_tab.
*
*DATA : lt_tab TYPE STANDARD TABLE OF ls_tab,
*      wa_tab TYPE ls_tab.
*
*SELECT-OPTIONS s_vbeln FOR wa_tab-vbeln.
*
*START-OF-SELECTION.
*
*  SELECT so_h~vbeln
*    so_h~audat
*    so_h~vbtyp
*    so_h~trvog
*    so_h~auart
*
*     so_i~posnr
*     so_i~matnr
*    so_i~matwa
*
*    INTO TABLE lt_tab
*    FROM vbak AS so_h
*
*    LEFT OUTER JOIN vbap AS so_i
*    ON so_h~vbeln = so_i~vbeln
*
*    WHERE so_h~vbeln IN s_vbeln.
*
*  IF sy-subrc EQ 0.
*    WRITE : 'sales order info ', sy-dbcnt.
*  ELSE.
*    WRITE : 'sales order records are not found ', sy-dbcnt.
```

```
*  ENDIF.
*
*END-OF-SELECTION.
*  LOOP AT lt_tab INTO wa_tab.
*    WRITE : / wa_tab-vbeln,
*     wa_tab-posnr,
*    wa_tab-matnr,
*    wa_tab-audat,
*    wa_tab-vbtyp,
*    wa_tab-trvog,
*    wa_tab-auart,
*      wa_tab-matwa.
* ENDLOOP.
```

*# 2 two tables with alias

```
*TYPES: BEGIN OF ls_tab,
*        vbeln TYPE vbeln_va,  "  vbak
*        audat TYPE audat,
*        vbtyp TYPE vbtypl,
*        trvog TYPE trvog,
*        auart TYPE auart,
*
*        posnr TYPE posnr_va,  "   vbap
*        matnr TYPE matnr,
*        matwa TYPE matwa,
*      END OF ls_tab.
*
*DATA : lt_tab TYPE STANDARD TABLE OF ls_tab,
*      wa_tab TYPE ls_tab.
*
*SELECT-OPTIONS s_vbeln FOR wa_tab-vbeln.
*START-OF-SELECTION.
*
*  SELECT so_h~vbeln
*    so_h~audat
```

```
*    so_h~vbtyp
*    so_h~trvog
*    so_h~auart
*
*     so_i~posnr
*     so_i~matnr
*    so_i~matwa
*
*    INTO TABLE lt_tab
*    FROM vbak AS so_h
*
**     LEFT OUTER JOIN vbap AS so_i
*
*   RIGHT OUTER JOIN vbap AS so_i
*    ON so_h~vbeln = so_i~vbeln
*
*    WHERE so_h~vbeln IN s_vbeln.
*
*  IF sy-subrc EQ 0.
*    WRITE : 'sales order info ', sy-dbcnt.
*  ELSE.
*    WRITE : 'sales order records are not found ', sy-dbcnt.
*  ENDIF.
*
*END-OF-SELECTION.
*  LOOP AT lt_tab INTO wa_tab.
*    WRITE : / wa_tab-vbeln,
*     wa_tab-posnr,
*    wa_tab-matnr,
*    wa_tab-audat,
*    wa_tab-vbtyp,
*    wa_tab-trvog,
*    wa_tab-auart,
*      wa_tab-matwa.
```

* ENDLOOP.

Day-3

→String functions

a)translate(upper/lower)->restrictions from user to provide only lower or upper

b)concatenate-→separated by delimeters,

SPACE KEYWORD

OBLIGATORY KEYWORD,NO SPACES

C)SPLIT

D)OFFSETTING

E) STRLEN AND THE TYPE SHOULD BE INTEGER

F) REPLACE

READ,SHIFT.,CONDENSE,NOGAP

TRANSLATE P_A TO LOWER CASE .

PROGRAMS ON STRINGS:

```
WRITE P_A.
 CONCATENATE
*PARAMETERS: P_A(10),
*          P_B(20),
*          P_DEL.
DATA LV_DEST(30) TYPE C.
******CONCATENATE P_A P_B INTO LV_DEST.
**CONCATENATE P_A P_B INTO LV_DEST SEPARATED BY '*'.
*CONCATENATE P_A P_B INTO LV_DEST SEPARATED BY SPACE.
*CONCATENATE P_A P_B INTO LV_DEST SEPARATED BY P_DEL.
*WRITE LV_DEST.
```

****SPLIT**

```
*PARAMETERS p_src(30).
*data: lv_a(15), lv_b(15).
*SPLIT p_src at ',' INTO lv_a lv_b.
*WRITE : / lv_a , lv_b.
```

****offsetting

```
*PARAMETERS p_a(20).
*data lv_dest(2).
*lv_dest = p_a+3(4).
```

*lv_dest = p_a+0.

*WRITE lv_dest.

**STRLEN

*PARAMETERS P_A(20).

*DATA LV_LEN TYPE I.

*LV_LEN = STRLEN(P_A) .

*WRITE LV_LEN.

*REPLACE

PARAMETERS P_SRC(20).

*DO.

IF P_SRC CA ','.

 REPLACE ',' WITH '$' INTO P_SRC.

  ELSE.

   EXIT.

   ENDIF.

* ENDDO.

WRITE P_SRC.

DDIC-Data Dictionary:

CREATION OF TABLES:

TO INSERT RECORDS INTO A TABLE GOTO MENU BAR ,UTILITIES→TABLE-CONTENTS→CREATE ENTRIES

Sm12→to delete the locks.

TMG→TABLE MAINTENANCE GENERATOR.

The basic functionality of the foreign keys is validation of data while joining two tables.

The domain and the data elements names should start with z or y and can or can't be same.

Table Entry Edit Goto Settings Utilities(M) System Help

Data Browser: Table Z280_TABS1 Select Entries 11

More Exit

Table: Z280_TABS1

Displayed Fields: 6 of 6 Fixed Columns: 2 List Width 0250

MANDT	UIN	NAM	DPTS	SITE	COMPENSATION
100	19101	ROHIT	MI	MUMBAI	45000
100	19102	GILL	GT	GUJARAT	77000
100	19103	KOHLI	RCB	BANGALORE	18000
100	19104	KL	RCB	BANGALORE	10000
100	19105	PANT	DC	DELHI	33000
100	19106	JADEJA	CSK	CHENNAI	80000
100	19107	MS DHONI	CSK	CHENNAI	70000
100	19108	JASMPRIT	MI	MUMBAI	85000
100	19109	SHAMI	GT	GUJARAT	85000
100	19110	KULDEEP	DC	DELHI	85000
100	19111	SHARDUL THAKUR	CSK	CHENNAI	85000

Continue

Type here to search ENG IN 11:03 01-05-2024

Table Edit Goto Utilities Extras Environment System Help

Dictionary: Display Table

Technical Settings Append Structure... More Exit

Transparent Table: Z280_TABS1 Active

Short Description: EMPLOYEE TABLE FOR THE ORGANIZATION

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

Search Help 1 / 6

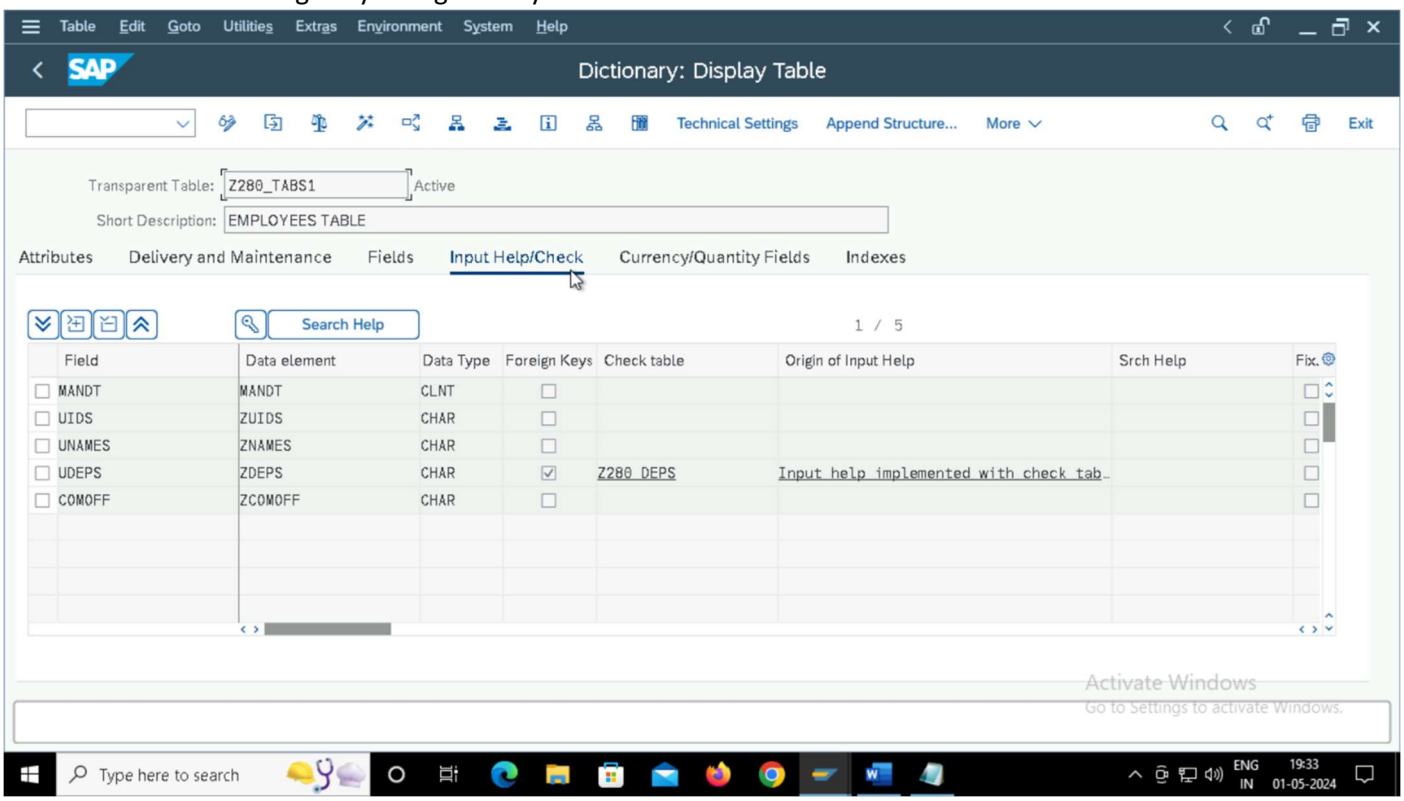
Field	Data element	Data Type	Foreign Keys	Check table	Origin of Input Help	Srch Help	Fix.
MANDT	MANDT	CLNT	☐				☐
UIN	ZUIN	CHAR	☐				☐
NAM	ZNAM	CHAR	☐				☐
DPTS	ZDPTS	CHAR	☑	Z280_TABS2	Input help implemented with check tab...		☐
SITE	ZSITE	CHAR	☑	Z280_TABS2	Input help implemented with check tab...		☐
COMPENSATION	ZCOMPENSATION	CHAR	☐				☐

Origin of Input Help

Type here to search ENG IN 11:03 01-05-2024

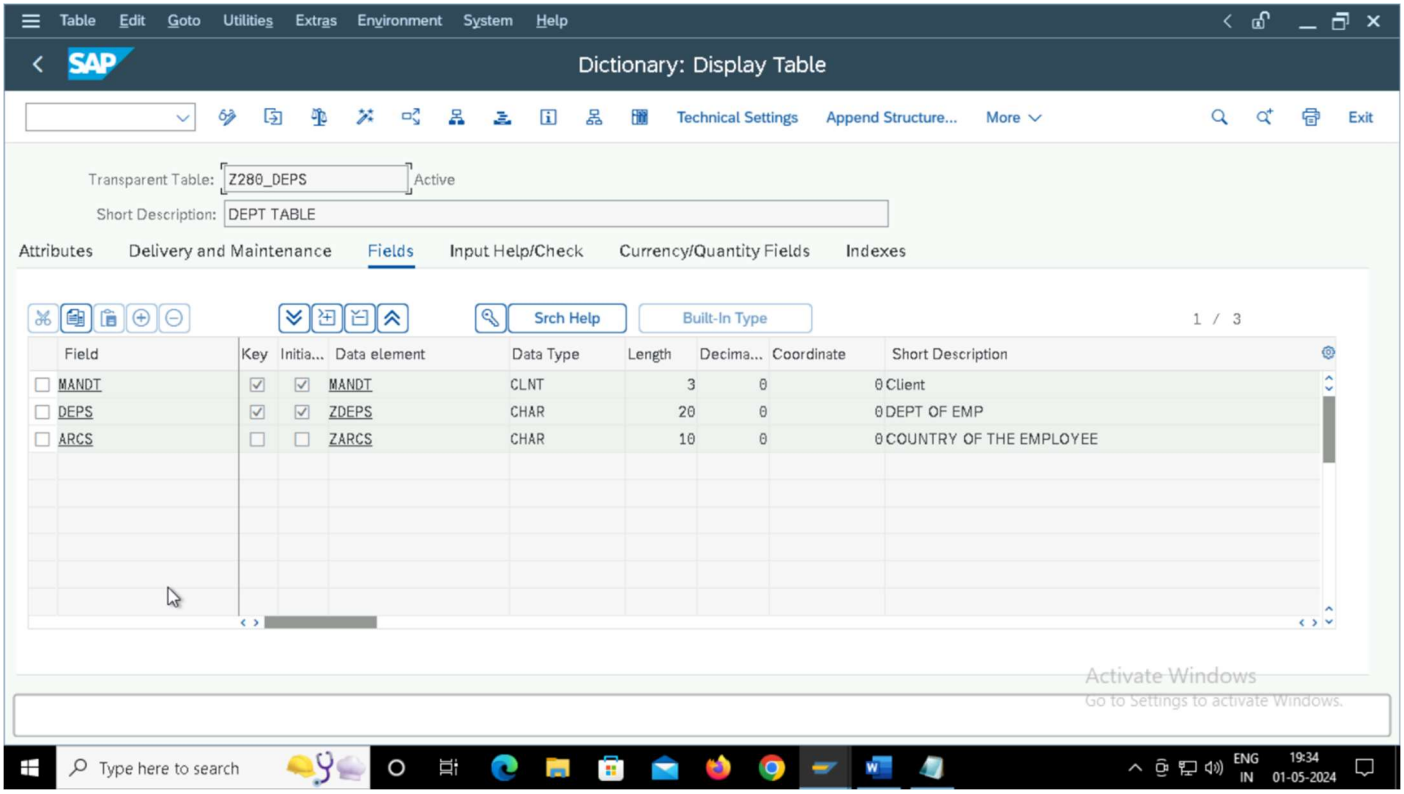
FOREIGN-Keys:

These foreign keys are generally used for connection of the tables and for the data validation.



INDEXES:

The indexes are generally used for the creating secondary primary and for the fields which you frequently used for the navigation.



CREATION OF THE FUNCTION GROUPS:STEPS

SE80→

SELECT THE FUNCTION GROUP FROM THE DROP DOWN

→START FUNCTION GROUP NAME WITH Z/Y

→CLICK ON ENTER AND YES

->PROVIDE SOME SHORT DESCRIPTION. AND SAVE IT YOUR PACKAGE.(CREATE A NEW TR) & SAVE .

NOTE:THERE ARE 2 INCLUDES INSIDE THE FUNCTION GROUP THEY ARE 1)TOP INCLUDE,2)UXX INCLUDE.

RIGHT CLICK ON THE FUNCTION GROUP NAME AND ACTIVATE.

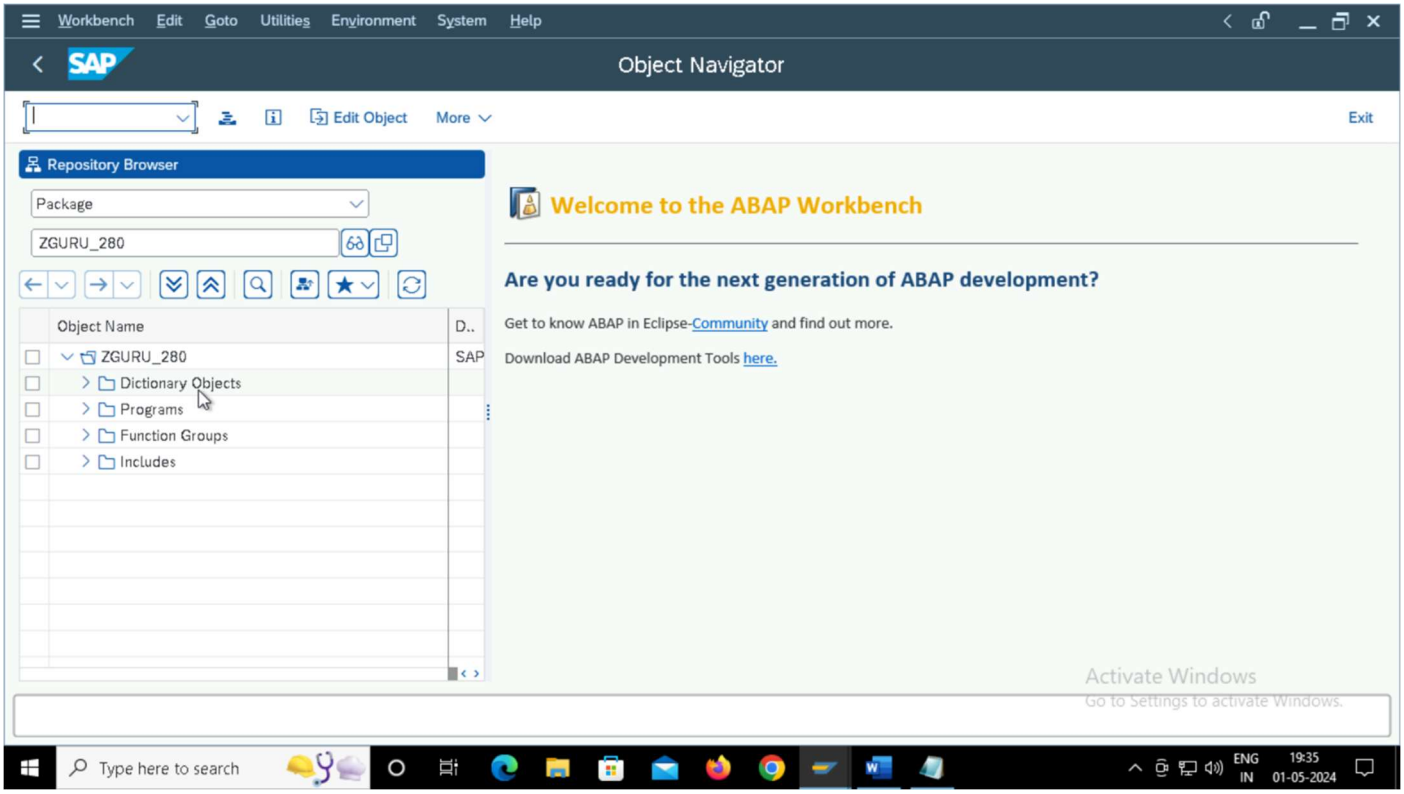
Step:2:
Create a TMG on a table and open the table in the change mode and goto utilities and click on tmg.

In the authorization group press f4,→&NC&.

In the function group insert box insert the function group from the first step.

Select the first ready button i.e. 1 step in the application tool bar find the screen number & in the pop select first button i.e propose screen number and continue.

Click on the create button besides the find screen number and save it.



Constraints:/declaration of variables:

Data-Variables : Tablename-fieldname,dataelement.

Parameters : Tablename-fieldname,dataelement.

Select options : tablename-fieldname.

Select : field name

Where : field name

Internal Tables:

The temporary table which is created during the execution of the program and is created during the runtime and after the execution the table will be deleted and no memory space is required.

A)With Header-Line,

B)Without Header-Line,

c)Work-Area.

```
→TABLES kna1.
*data: lv_kunnr TYPE kna1-kunnr, " Customer Number
*     lv_name1 type kna1-name1, " Name 1
*     lv_ort01 TYPE kna1-ort01, " city
*     lv_land1 TYPE kna1-land1. " country
*select kunnr name1 ort01 land1
*      into ( lv_kunnr, lv_name1 , lv_ort01 , lv_land1 )
*      from kna1
*      WHERE kunnr = '0000000006'.
*  WRITE : / lv_kunnr, lv_name1 , lv_ort01 , lv_land1.
* ENDSELECT.
*# 3   without header
*TABLES kna1.
*DATA: lt_tab type kna1 occurs 0 WITH HEADER LINE.
** in this case itab body and header names are same.
*data lv_kunnr TYPE kunnr.
*select-OPTIONS s_kunnr for lv_kunnr.  " 1  to 100
*PARAMETERS p_land1 TYPE kna1-land1.  " us
```

```
*
*SELECT *   "  1/2
*******************       INTO lt_Tab   " into HEADER
*      INTO TABLE  lt_Tab   " into ITAB BODY
*      FROM kna1
*       wHERE kunnr in s_kunnr and
*            land1 eq p_land1.
*
** APPEND lt_tab to lt_tab.  " h to b
***ENDSELECT.
*
*loop at lt_Tab INTO lt_Tab.  " b to h
*
* WRITE: / lt_tab-kunnr,   " h-f
*         lt_tab-name1,
*         lt_tab-ort01,
*         lt_tab-land1.
*
*  ENDLOOP.
```

*# 4 WITH USER DEFINED WORK AREA

```
*TABLES kna1.
*
*TYPES : BEGIN OF LS_TAB,
*******************  KUNNR TYPE KNA1-KUNNR. " V TYPE T-F
*     lv_KUNNR TYPE KUNNR ,  "  V TYPE DTL.
*     NAME1 TYPE NAME1_GP ,
*     ORT01 TYPE ORT01_GP,
*     LAND1 TYPE LAND1_GP,
*     END OF LS_TAB.
*
*DATA : LT_tAB TYPE STANDARD TABLE OF LS_TAB, " ITAB
*      WA_TAB TYPE LS_tAB.   " WORK AREA
*
*data lv_kunnr TYPE kunnr.
```

```
*select-OPTIONS s_kunnr for lv_kunnr.  " 1  to 100
*PARAMETERS p_land1 TYPE kna1-land1.  " us
*
*SELECT  KUNNR NAME1 ORT01 LAND1
*      INTO WA_Tab   " into WORK AREA
*      FROM kna1
*       wHERE kunnr in s_kunnr and
*            land1 eq p_land1.
*
* APPEND WA_tab to lt_tab.  " W to b
* ENDSELECT.
*
*loop at lt_Tab INTO WA_Tab.  " b to W
*
* WRITE: / WA_tab-lv_kunnr,   " W-f
*         WA_tab-name1,
*         WA_tab-ort01,
*         WA_tab-land1.
*
*  ENDLOOP.
```

OPTIMIZED-CODE

```
*TABLES kna1.
*
*TYPES : BEGIN OF LS_TAB,   " LOCAL STRS
*****************  KUNNR TYPE KNA1-KUNNR. " V TYPE T-F
*     KUNNR TYPE KUNNR ,  "  V TYPE DTL.
*     NAME1 TYPE NAME1_GP ,
*     ORT01 TYPE ORT01_GP,
*     LAND1 TYPE LAND1_GP,
*
*     END OF LS_TAB.
*
*DATA : LT_tAB TYPE STANDARD TABLE OF LS_TAB, " ITAB
*      WA_TAB TYPE LS_tAB.   " WORK AREA
```

```
*
*data lv_kunnr TYPE kunnr.
*select-OPTIONS s_kunnr for lv_kunnr.  " 1  to 100
*PARAMETERS p_land1 TYPE kna1-land1.  " us
*
*SELECT  KUNNR NAME1 ORT01 LAND1
***********      INTO WA_Tab   " into WORK AREA
*  INTO TABLE LT_tAB
*      FROM kna1
*       wHERE kunnr in s_kunnr and
*            land1 eq p_land1.
*
************** APPEND WA_tab to lt_tab.  " W to b
************** ENDSELECT.
*
*loop at lt_Tab INTO WA_Tab.  " b to W
*
* WRITE: / WA_tab-kunnr,   " W-f
*        WA_tab-name1,
*        WA_tab-ort01,
*        WA_tab-land1.
*  ENDLOOP.
```

DOWNLOADING FILES INTO THE LOCAL SYSTEM FORM THE SAP SERVER:

Goto se38/reptran/package name and the other details/click on execute

Structures:

they are of 2 types i.e local and global ,

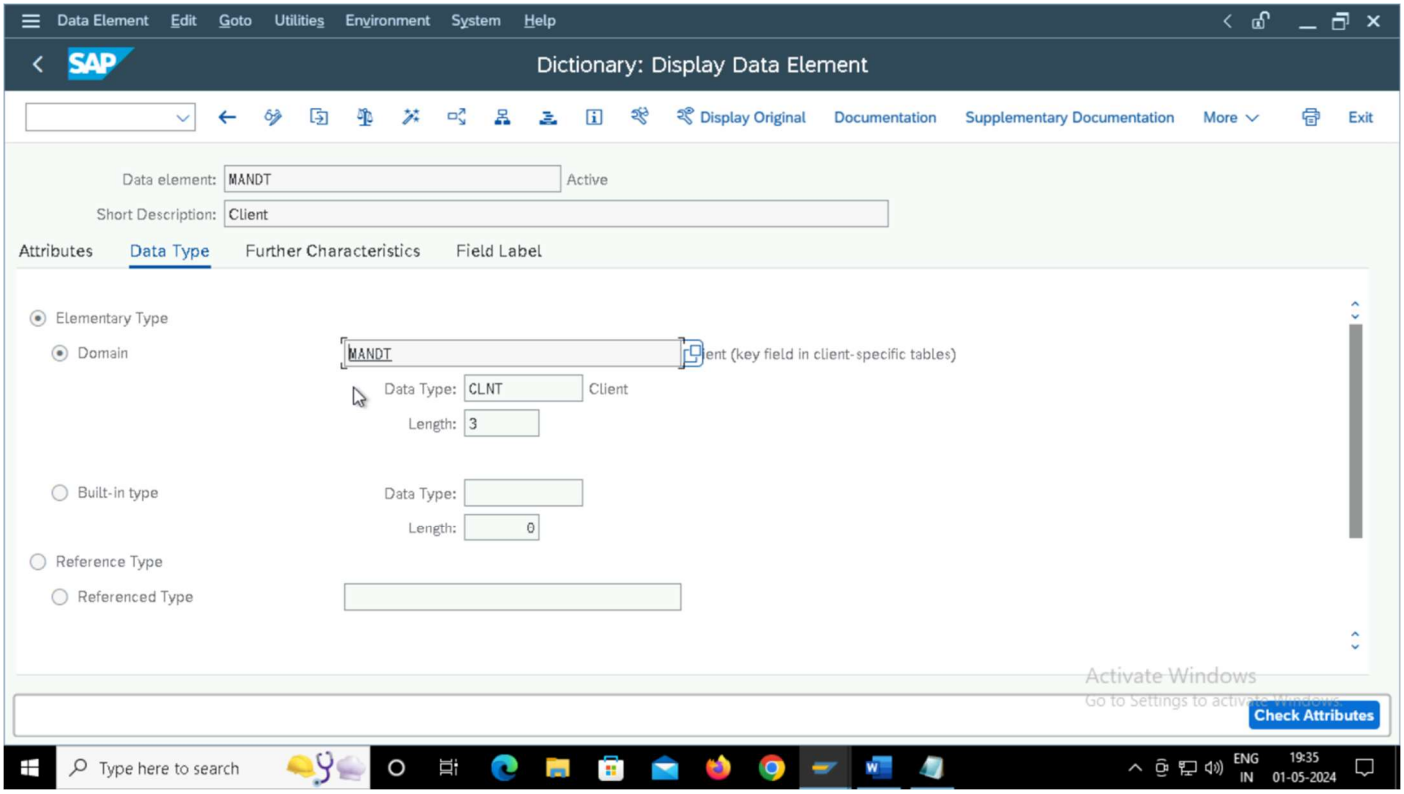
Local: in se38 /80 with a keyword called' types' we can define the structures,and they are reusable within the same program .

The structure doesn't contain any data its only the definition.

Global: in se11 ->flat structure, nested structure and deep structure/complex/table types/line types.

Flat structure/global stru:

se11->data type ready button provide the structure name z/y->click on create->copy the data elements from the predefined table._>save and activate.



VIEWS:

The views are virtual tables created and can't occupy any memory .They are of 4 types.

a)Database view : works on inner joins between the tables.

b)Projection View : Used for data confidentiality.

c)Maintenance View : Used for making mass changes in the database.

d)Help View : Used for working with the left joins .

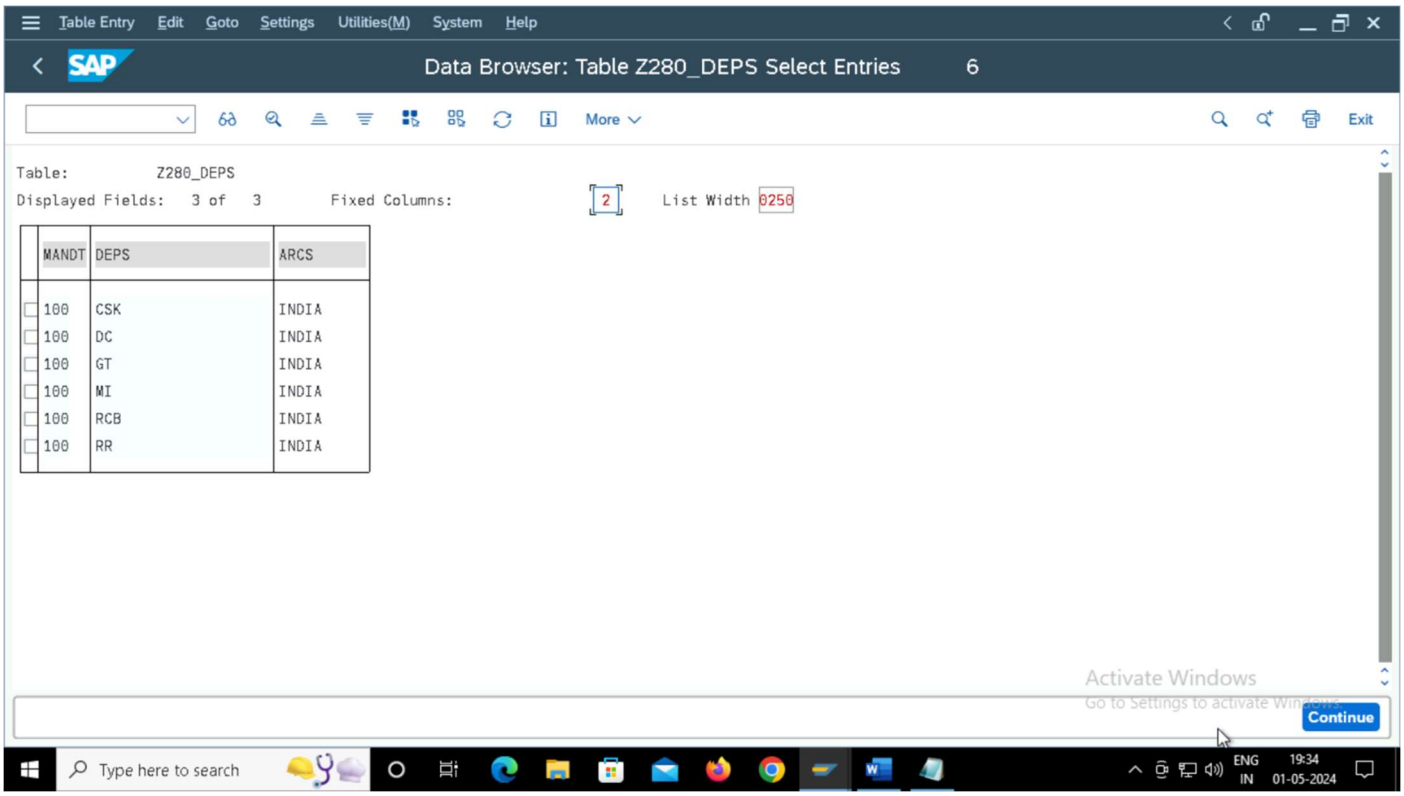


Table View Edit Goto Selection Utilities System Help

Change View "DEPT TABLE": Overview

New Entries More Display Exit

DEPT TABLE

DEPARTMENT	COUNTRY
CSK	INDIA
DC	INDIA
GT	INDIA
MI	INDIA
RCB	INDIA
RR	INDIA

Position... Entry 1 of 6

Save Cancel

Table View Edit Goto Selection Utilities System Help

Change View "EMPLOYEES TABLE": Overview

New Entries More Display Exit

EMPLOYEES TABLE

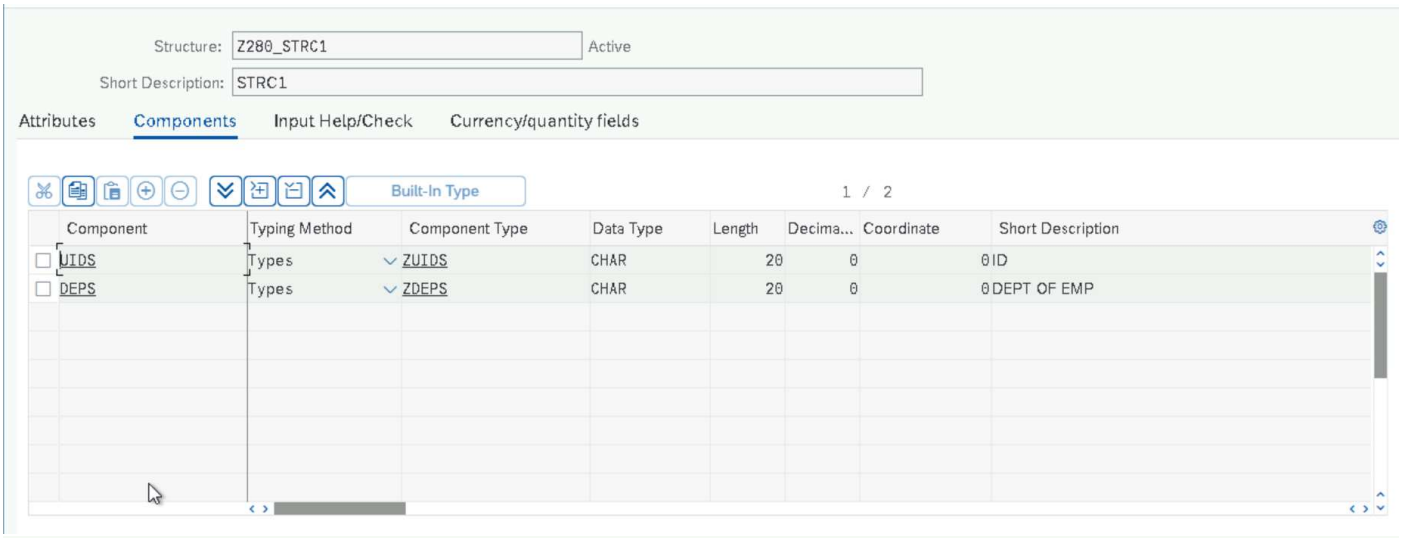
UID	NAME	DEPARTMENT	SALARY
19101	ROHIT	MI	45000
19102	GILL	GT	33000
19103	KOHLI	RCB	18000
19104	KL	RCB	10000
19105	MSD	CSK	70000
19106	JADEJA	CSK	80000
19107	PANT	DC	33000
19108	SHAMI	GT	1000
19109	BOOM	MI	1000
19110	CHAHAL	RR	1000
19111	KULDEEP	DC	1000

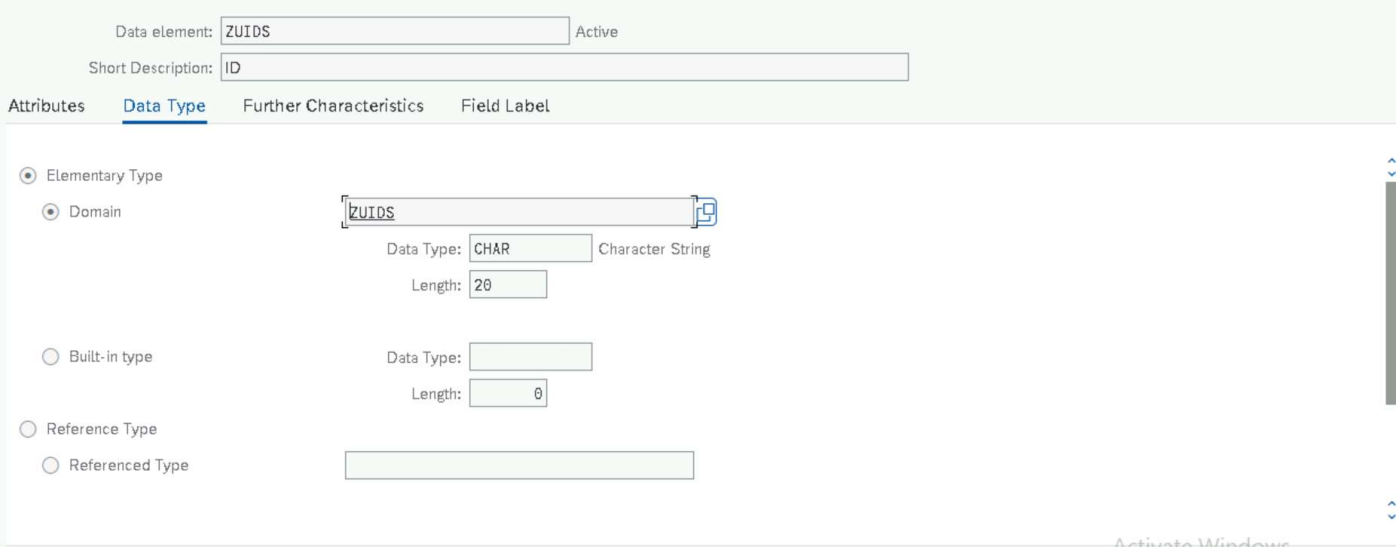
Position... Entry 1 of 11

Save Cancel

Structures :

The structures can be declared both internally and externally. The global defined structures can be reused.

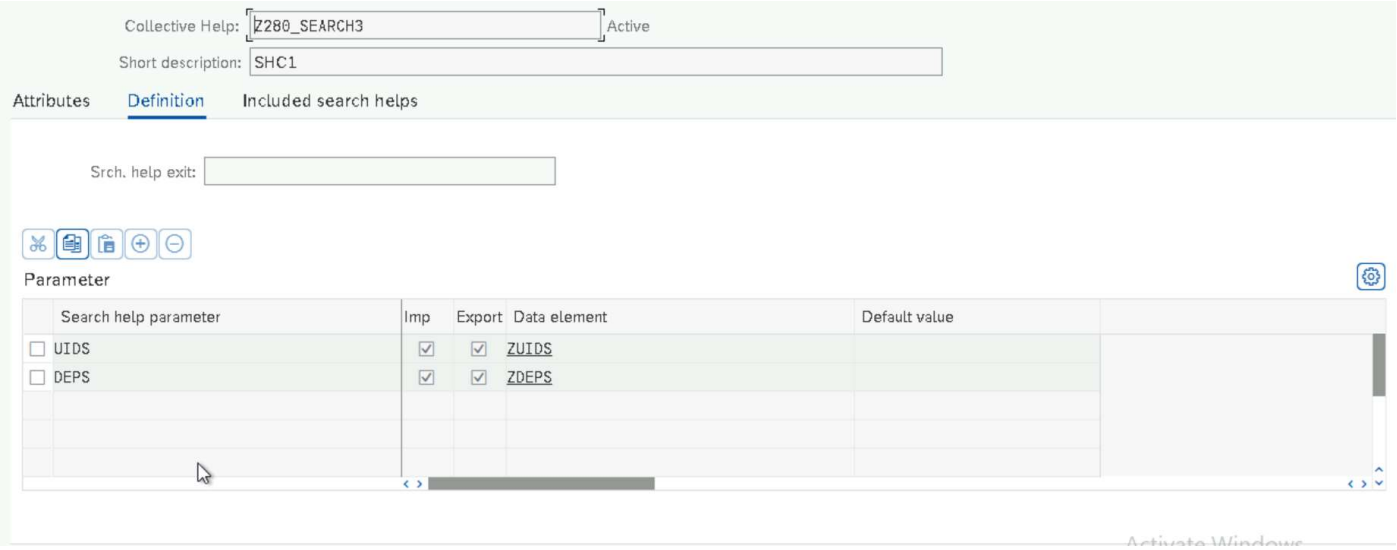




SEARCH HELP:

The Search help is used for giving the users to give the options for the input.

They are of 2 types : a)Elemental Search b)Collective Search.



SEARCH HELP FOR DEPS AND TABS1 TABLES SH2

DEPARTMENT
CSK
CSK
DC
DC
GT
GT
MI
MI
RCB
RCB
RR