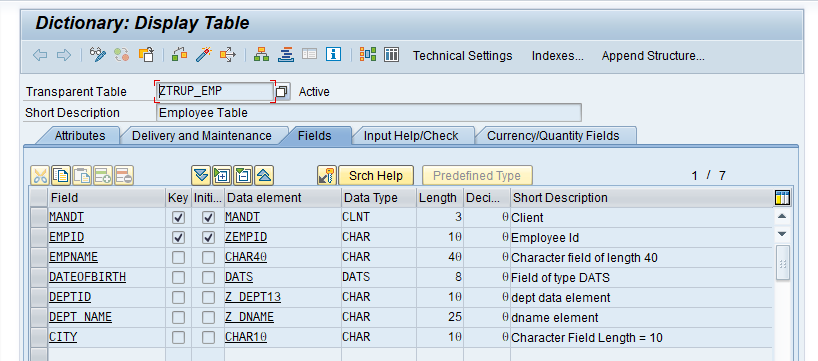
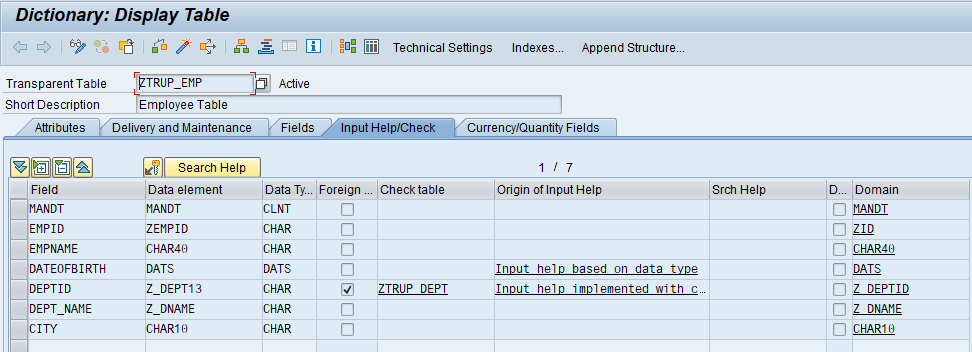
**DDIC - TMG Events Demo # 1.**

***DDIC – TMG theme:*** *In this demo, you will get how to use TMG events.*

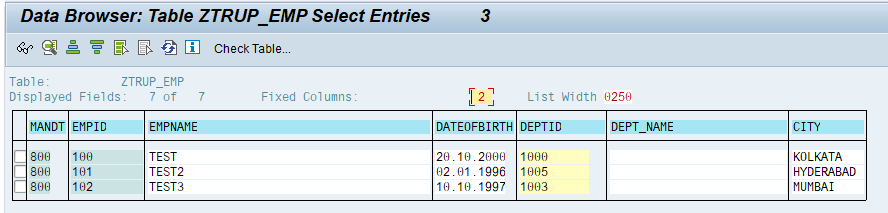
**Step # 1:** Go to transaction code SE11 and create the table (ZTRUP\_EMP) and enter a few entries in the table.



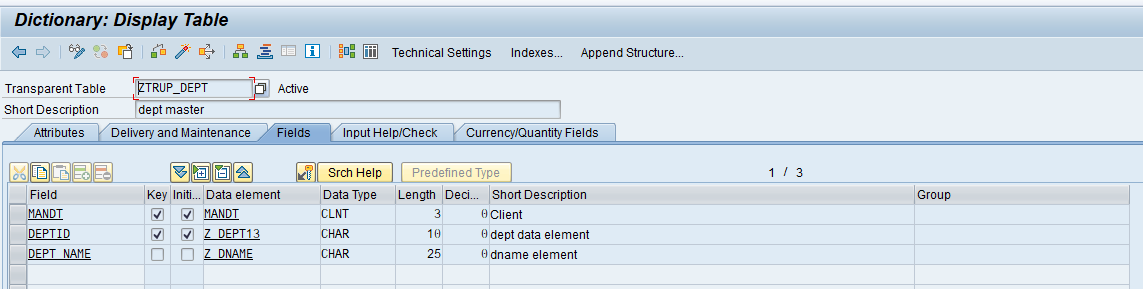
For the field DEPTID created a check table ZTRUP\_DEPT.



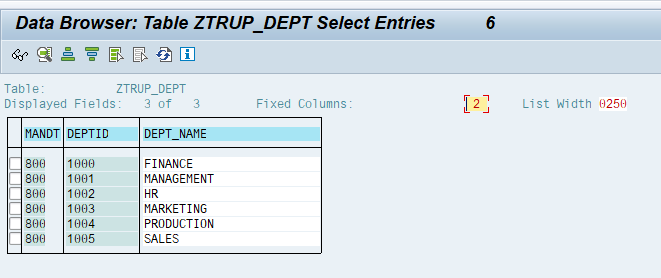
Test Data of ZTRUP\_EMP



**Step # 2:** Go to transaction code SE11 and create the table (ZTRUP\_DEPT) and enter a few entries in the table.

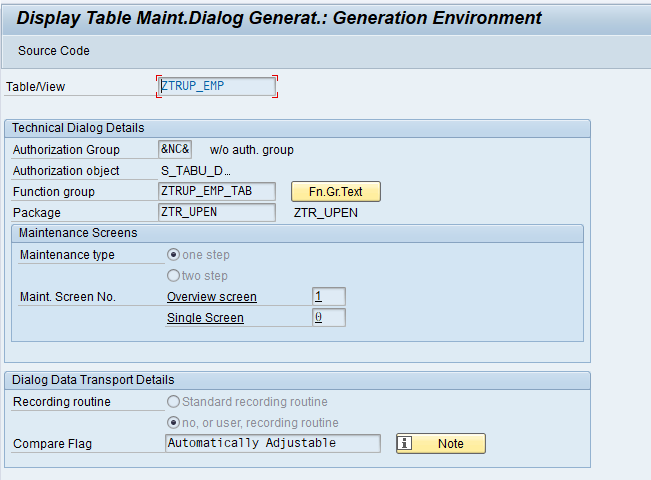


Test Data of ZTRUP\_DEPT Table.

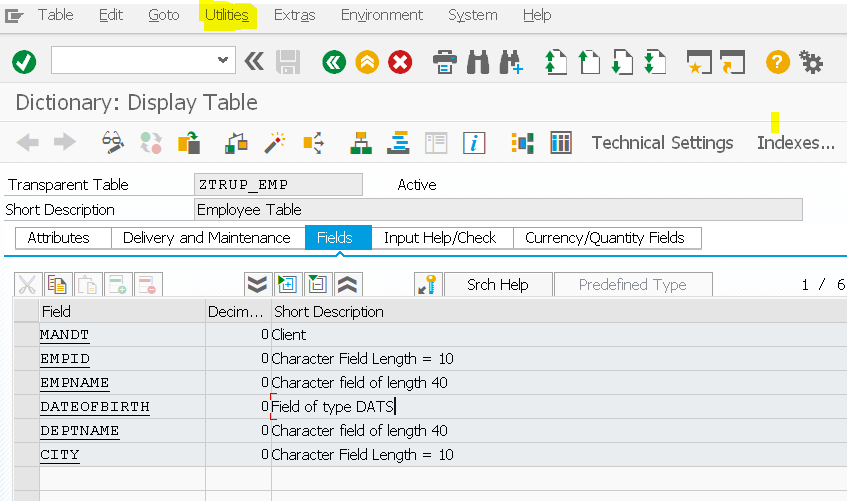


**Step # 3:** Go to transaction code SE11 and open the database table ZTRUP\_EMP. Utilities->Table Maintenance Generator.

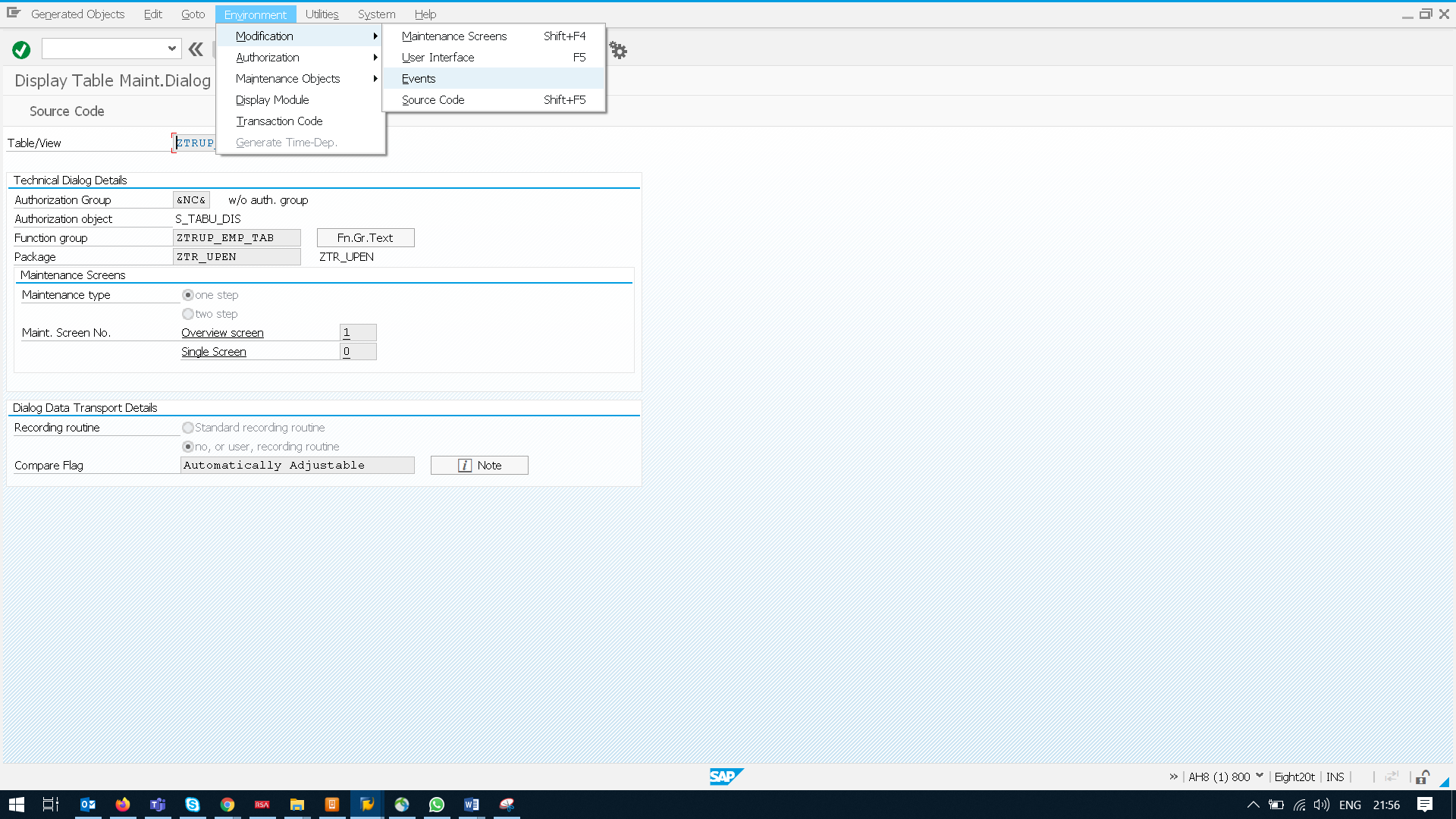
TMG Screen:



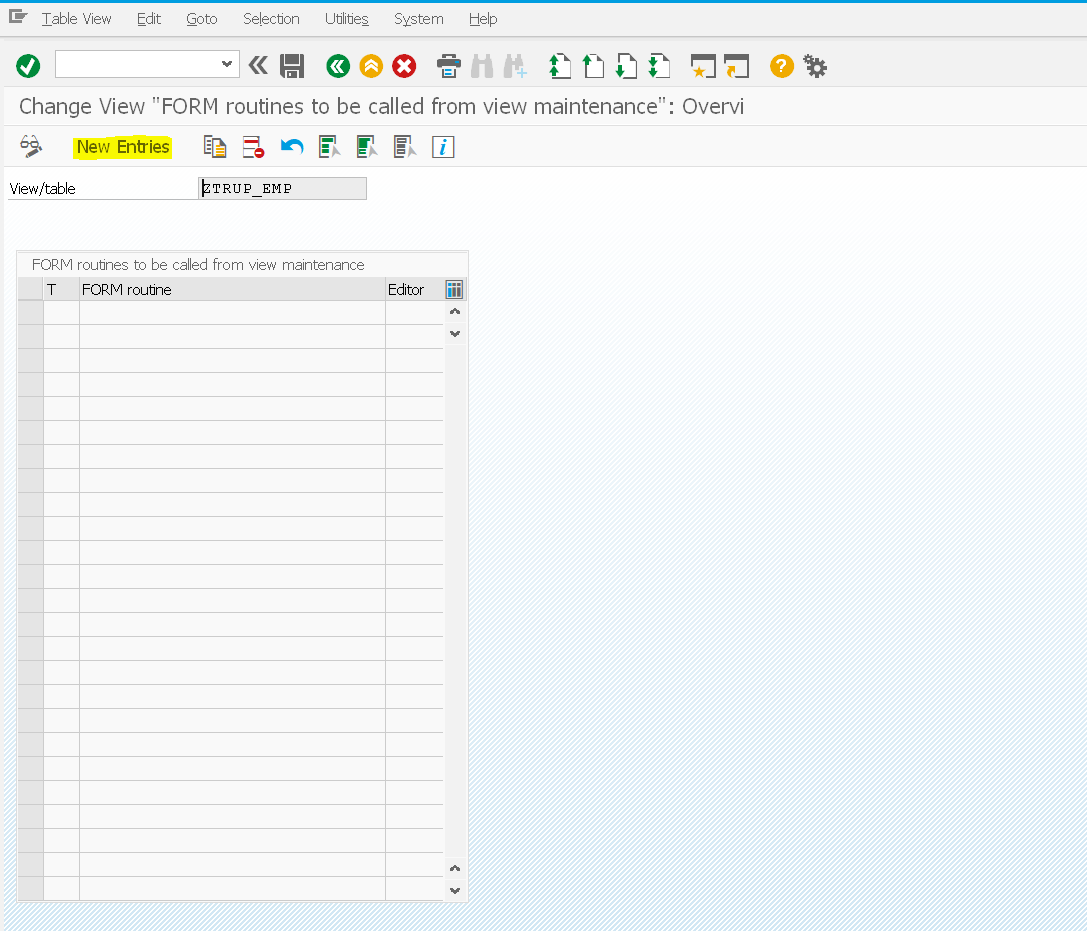
Utilities->Table Maintenance Generator.



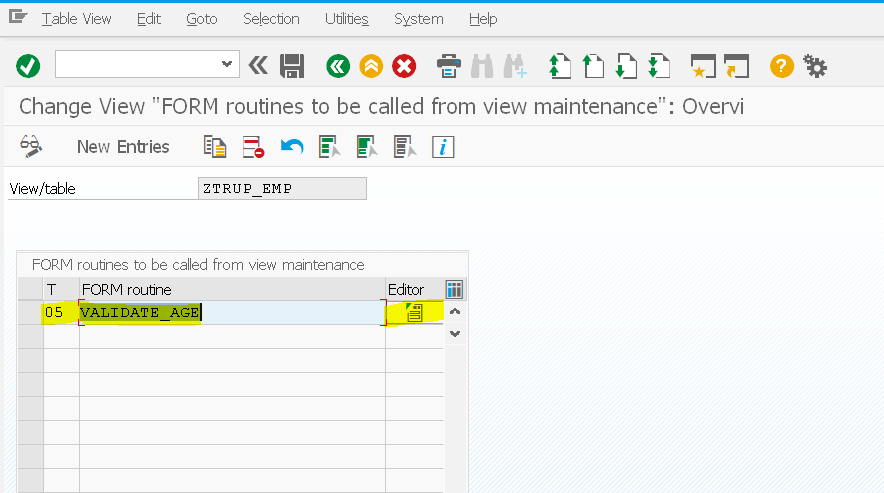
**Step # 4:** In the TMG Screen - Go to Environment->Modification->Events



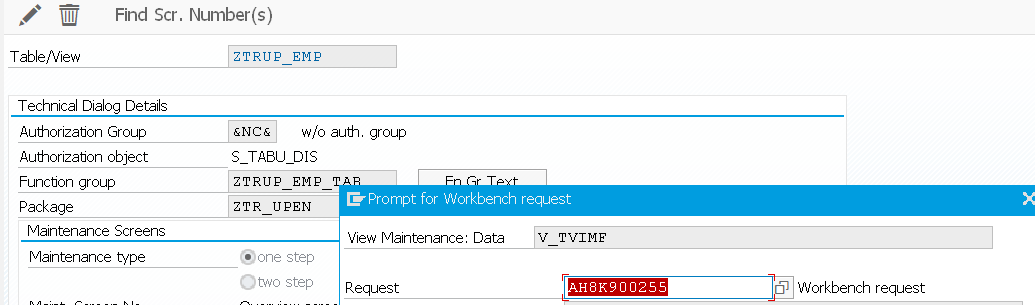
**Step # 5:** Now add a new entry in TMG Events Maintenance Overview



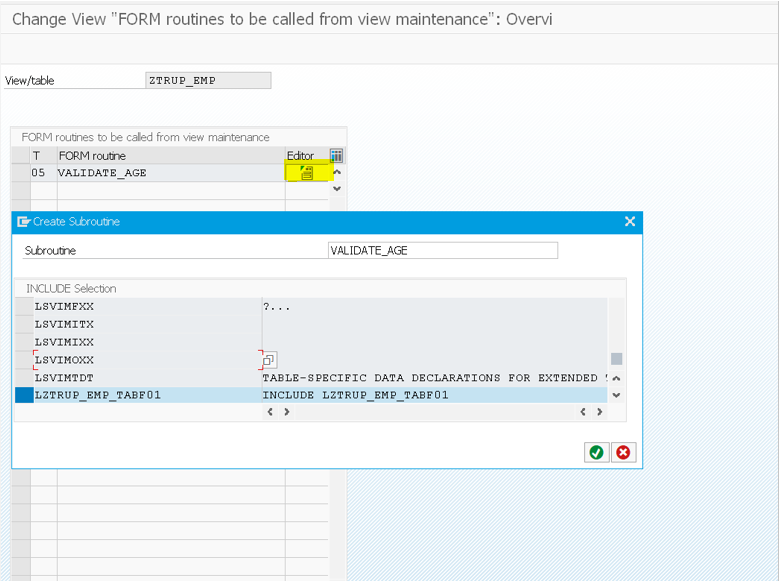
**Step # 6:** Provide the TMG Event as per the requirement and provide the sub routine name (VALIDATE\_AGE ) – 05 Creating a new entry.



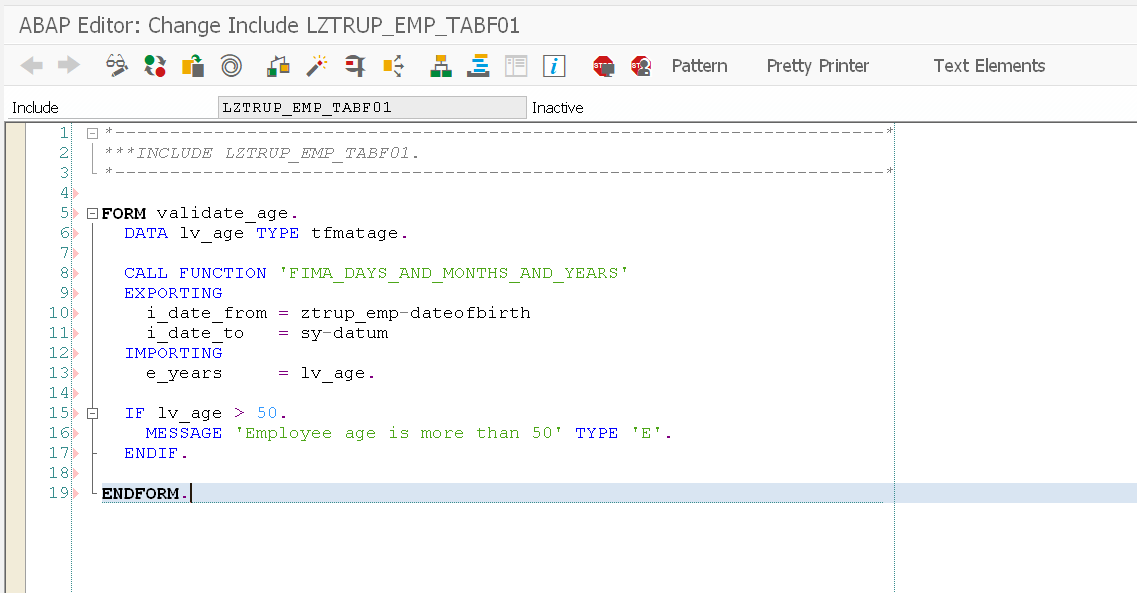
**Step # 7:** Select the TR after clicking save.



**Step # 8:** Now click on Editor Button in the Maintenance screen and select the existing include or create a new include.

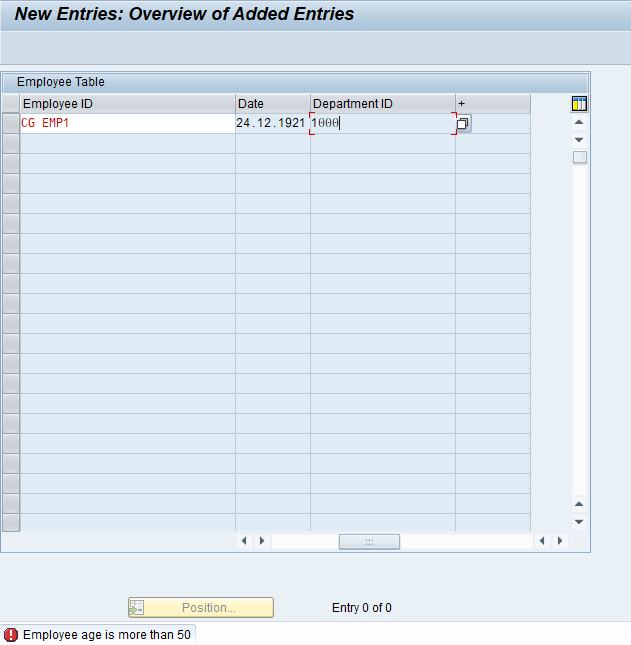


**Step # 9:** Create the subroutine with the same name as provided in previous screen. Write the logic as per the requirement. Note: For events 01, 05 and 21, the loop occurs on each data and store it into structure (same names as table name).



**Step # 10:** Go to transaction code SE11 and enter a few entries in the table (ZTRUP\_EMP).

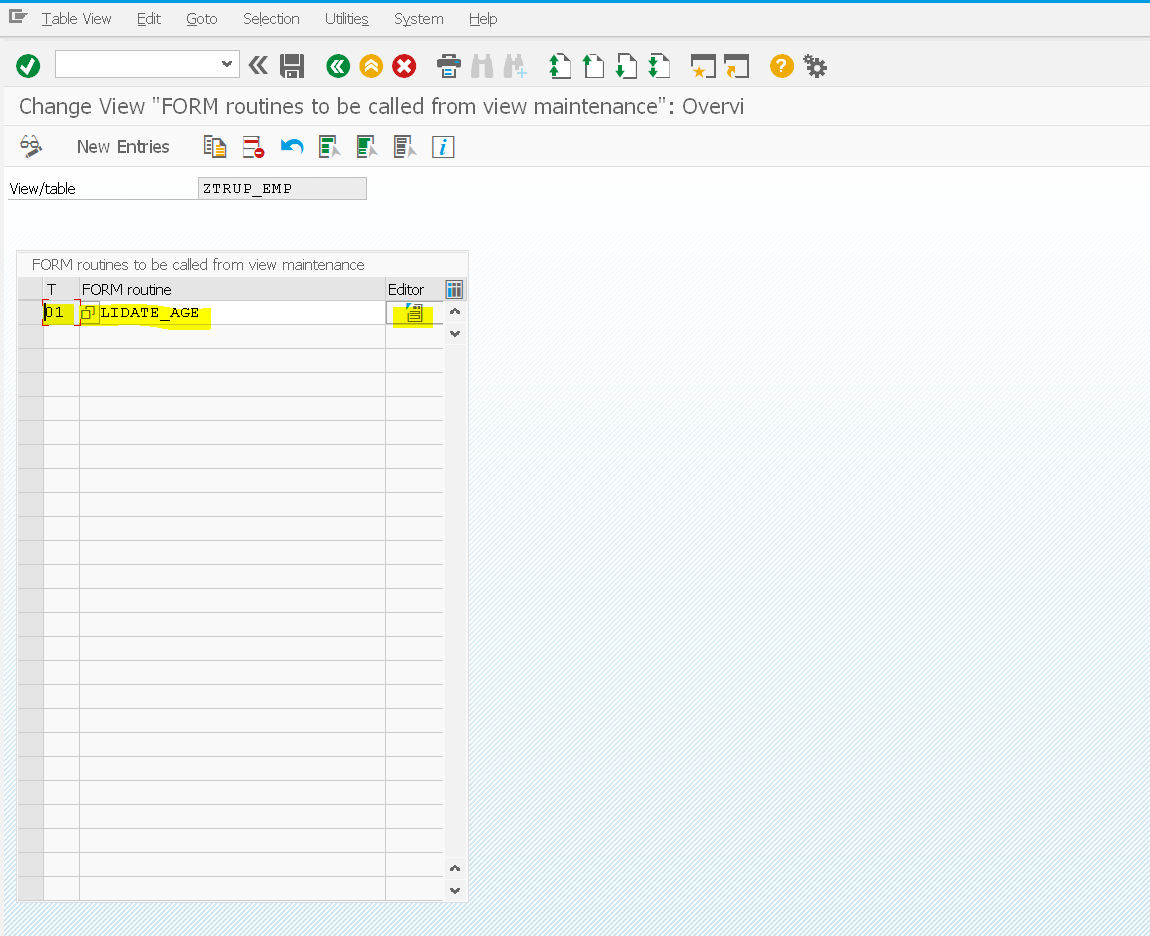
Note: TMG – Event was triggered when you enter the age above 50 Years date (24.12.1921). Check the error message at the status screen.



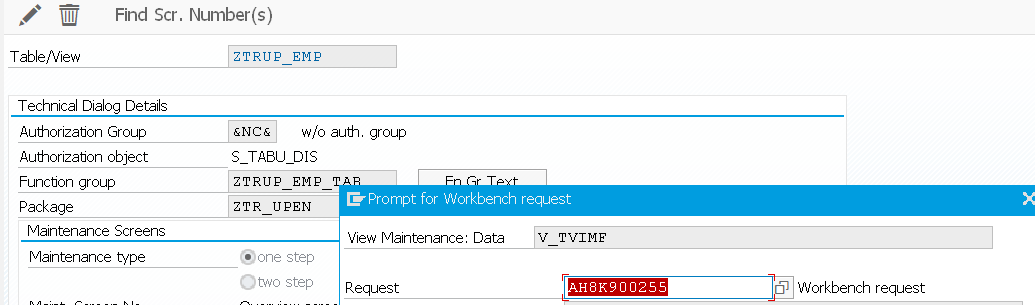
**Using TMG Event 01:**

This event will be triggered before you save the entries.

**Step # 1:** Go to TMG Event maintenance screen, add new entry with Event 01, and provide the subroutine name.



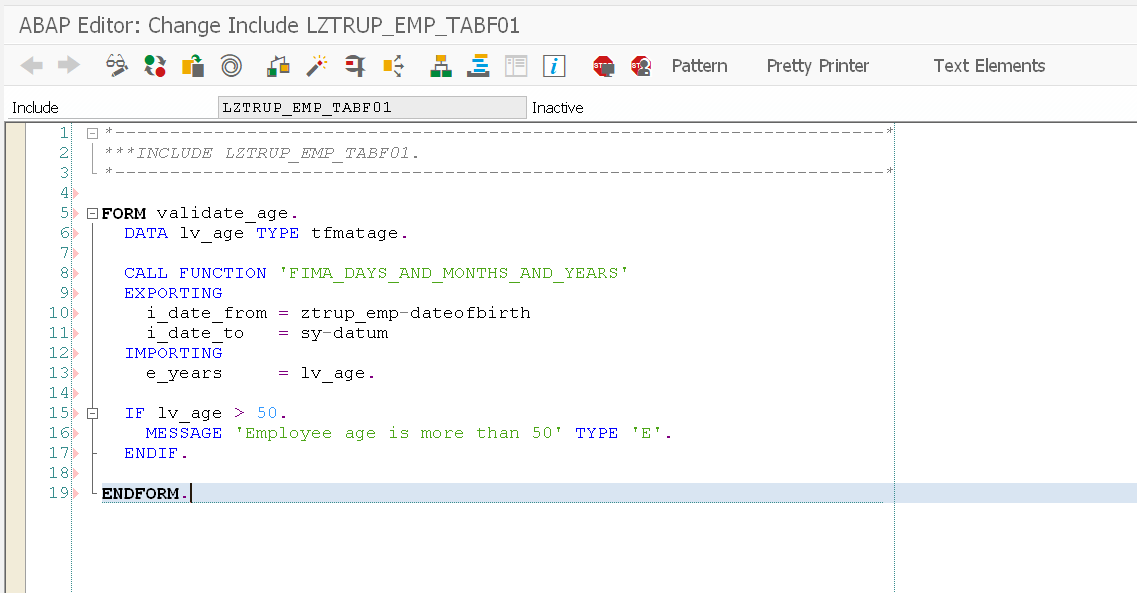
**Step # 2:** Select the TR after clicking save.



**Step # 3:** Create the subroutine with the same name as provided in previous screen.

Write the logic as per the requirement.

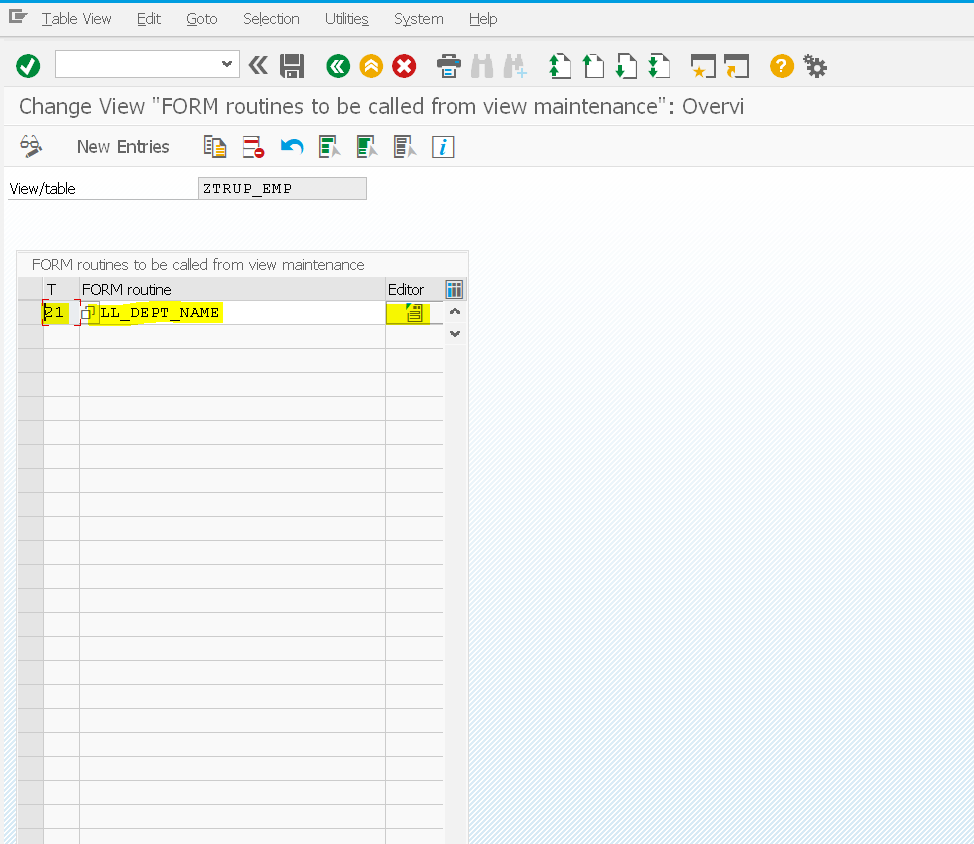
Note: For events 01, 05 and 21, the loop occurs on each data and store it into structure (same names as table name).



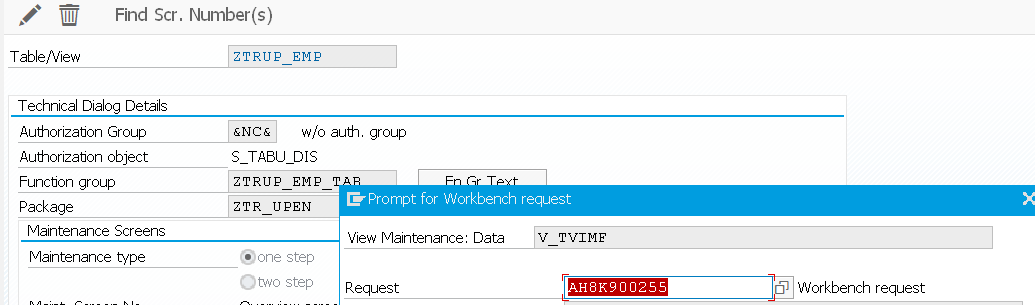
**Using TMG Event 21:**

This event will be triggered to fill hidden columns in the table.

**Step # 1:** Go to TMG Event maintenance screen, add new entry with Event 21, and provide the subroutine name.



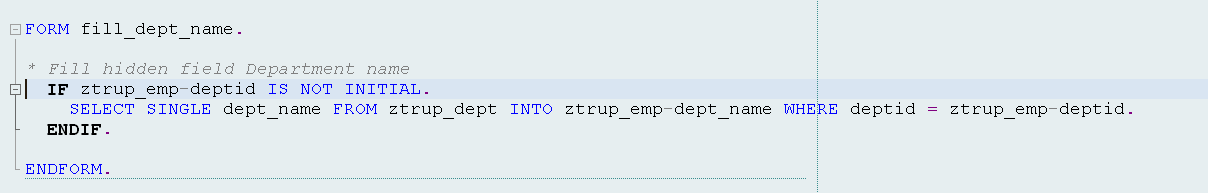
**Step # 2:** Select the TR after clicking save.



**Step # 3:** Create the subroutine with the same name as provided in previous screen.

Write the logic as per the requirement; fill the hidden field for each row of the table contents.

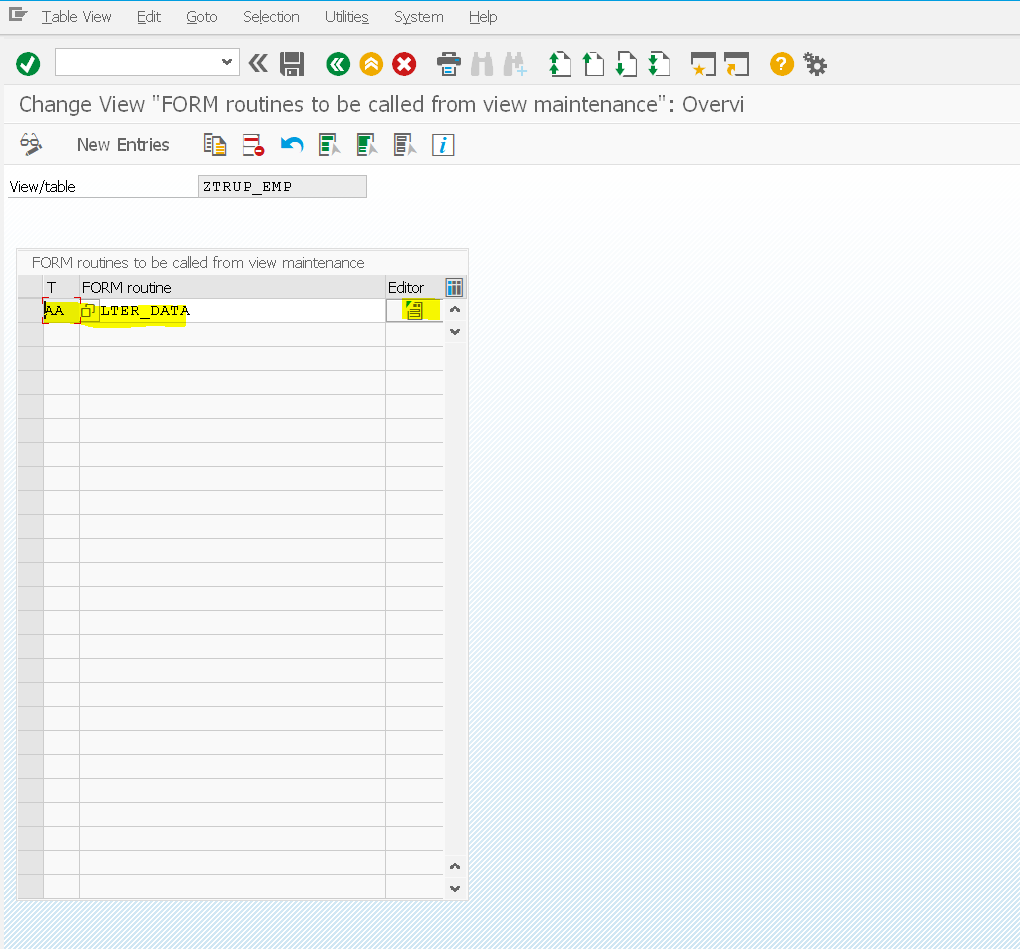
Note: For events 01, 05 and 21, the loop occurs on each data and store it into structure (same names as table name).



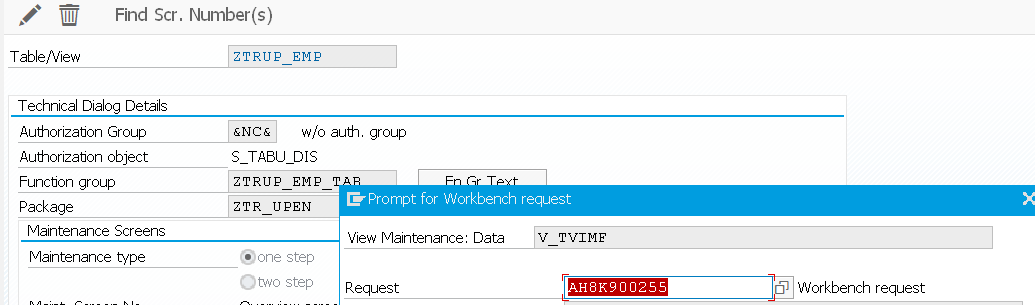
**Using TMG Event AA:**

This event will be triggered before showing the content of table in TMG. This event execute for fetching data instead of standard fetch from the table. This can be used to filter data based on some conditions.

**Step # 1:** Go to TMG Event maintenance screen, add new entry with Event AA, and provide the subroutine name.

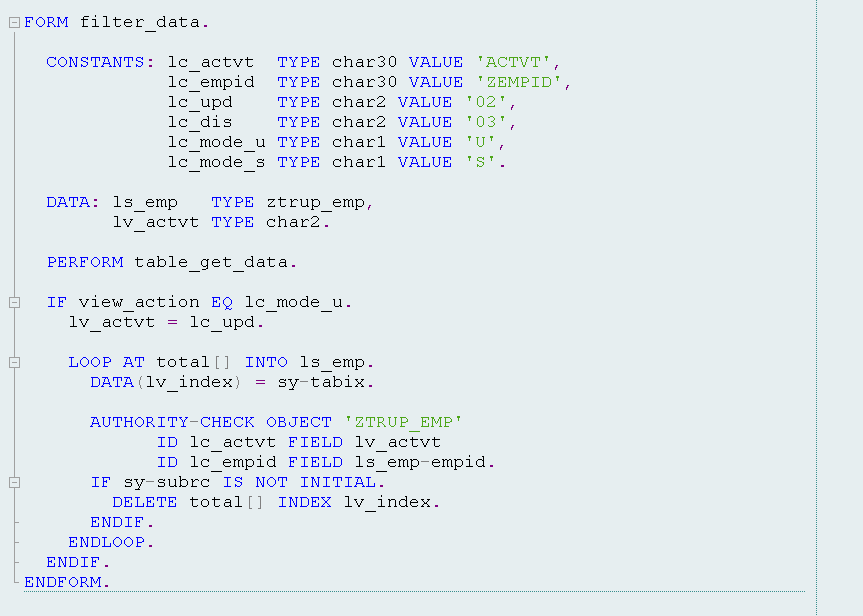


**Step # 2:** Select the TR after clicking save.



**Step # 3:** Create the subroutine with the same name as provided in previous screen.

Write the logic as per the requirement; this can be used to filter data.



**The TMG events are as follows.**

01 Before saving the data in the database

02 After saving the data in the database

03 Before deleting the data displayed

04 After deleting the data displayed

05 Creating a new entry

06 After completely performing the function 'Get original'

07 Before correcting the contents of a selected field

08 After correcting the contents of a selected field

09 After getting the original of an entry

10 After creating the header entries for the change task (E071)

11 After changing a key entry for the change task (E071K)

12 After changing the key entries for the change task (E071K)

13 Exit editing (exit main function module)

14 After lock/unlock in the main function module

15 Before retrieving deleted entries

16 After retrieving deleted entries

17 Do not use. Before print: Event 26

18 After checking whether the data has changed

19 After initializing global variables, field symbols, etc.

20 after input in date subscreen (time-dep. tab./views)

21 Fill hidden fields

22 Go to long text editor for other languages

23 Before calling address editing screen

24 After restricting an entry (time-dep. tab./views)

25 Individual authorization checks

26 Before creating a list

27 After creation or copying a GUID (not a key field)

28 After entering a date restriction for time-dep. views

AA Instead of the standard data read routine

AB Instead of the standard database change routine

AC Instead of the standard 'Get original' routine

AD Instead of the standard RO field read routine

AE Instead of standard positioning code

AF Instead of reading texts in other languages

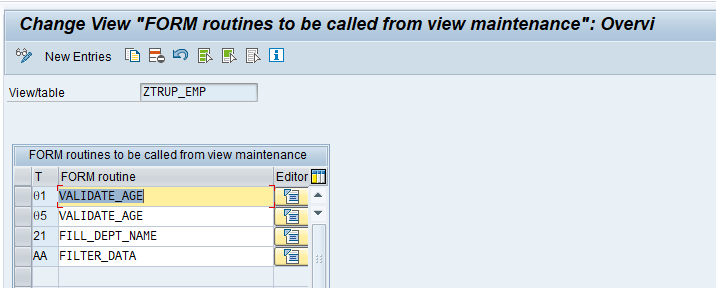
AG Instead of 'Get original' for texts in other languages

AH Instead of DB change for texts in other languages

ST GUI menu main program name

AI Internal use only

**List of events handled and its Code;**



Code:

\*----------------------------------------------------------------------\*  
\*\*\*INCLUDE LZTRUP\_EMP\_TABF01.  
\*----------------------------------------------------------------------\*  
  
FORM validate\_age.  
  DATA lv\_age TYPE tfmatage.  
  
  CALL FUNCTION 'FIMA\_DAYS\_AND\_MONTHS\_AND\_YEARS'  
    EXPORTING  
      i\_date\_from = ztrup\_emp-dateofbirth  
      i\_date\_to   = sy-datum  
    IMPORTING  
      e\_years     = lv\_age.  
  
  IF lv\_age > 50.  
    MESSAGE 'Employee age is more than 50' TYPE 'E'.  
  ENDIF.  
  
ENDFORM.  
  
FORM fill\_dept\_name.  
  
\* Fill hidden field Department name  
  IF ztrup\_emp-deptid IS NOT INITIAL.  
    SELECT SINGLE dept\_name FROM ztrup\_dept INTO ztrup\_emp-dept\_name WHERE deptid = ztrup\_emp-deptid.  
  ENDIF.  
  
ENDFORM.  
  
FORM filter\_data.  
  
  CONSTANTS: lc\_actvt  TYPE char30 VALUE 'ACTVT',  
             lc\_empid  TYPE char30 VALUE 'ZEMPID',  
             lc\_upd    TYPE char2 VALUE '02',  
             lc\_dis    TYPE char2 VALUE '03',  
             lc\_mode\_u TYPE char1 VALUE 'U',  
             lc\_mode\_s TYPE char1 VALUE 'S'.  
  
  DATA: ls\_emp   TYPE ztrup\_emp,  
        lv\_actvt TYPE char2.  
  
  PERFORM table\_get\_data.  
  
  IF view\_action EQ lc\_mode\_u.  
    lv\_actvt = lc\_upd.  
  
    LOOP AT total[] INTO ls\_emp.  
      DATA(lv\_index) = sy-tabix.  
  
      AUTHORITY-CHECK OBJECT 'ZTRUP\_EMP'  
            ID lc\_actvt FIELD lv\_actvt  
            ID lc\_empid FIELD ls\_emp-empid.  
      IF sy-subrc IS NOT INITIAL.  
        DELETE total[] INDEX lv\_index.  
      ENDIF.  
    ENDLOOP.  
  ENDIF.  
ENDFORM.