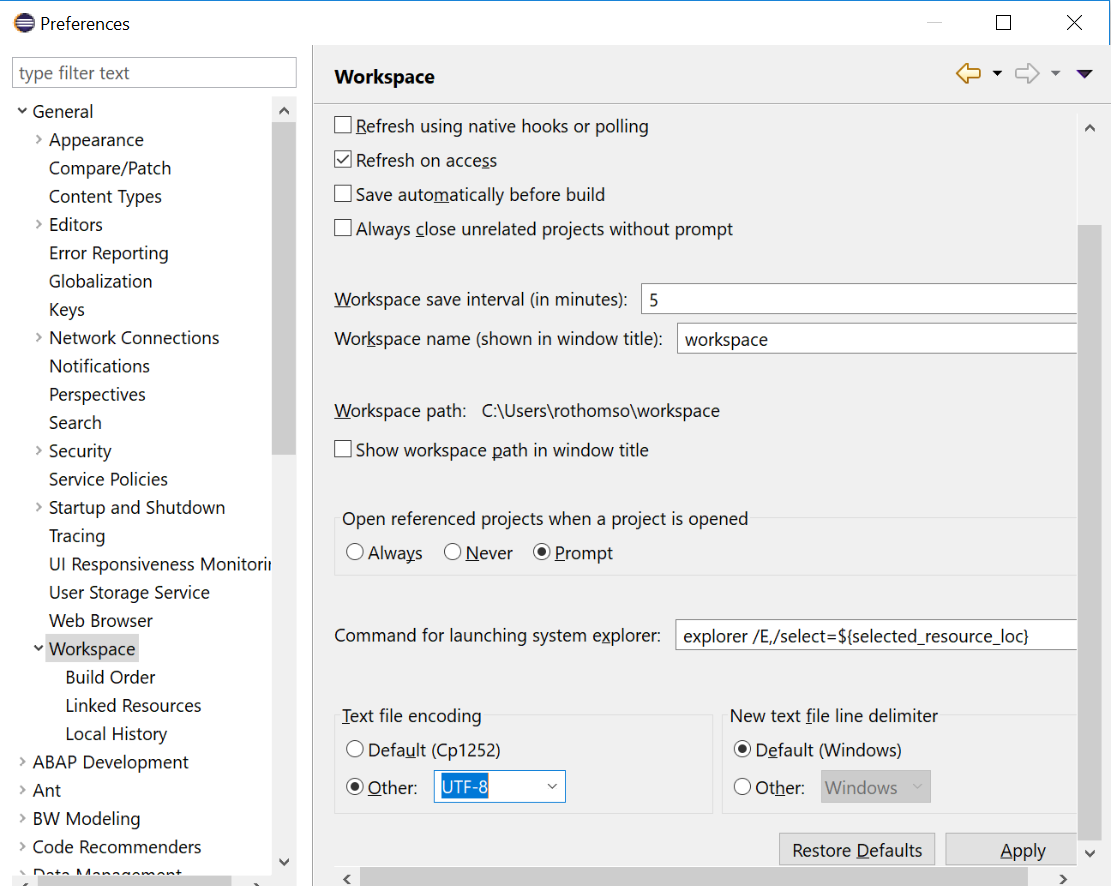
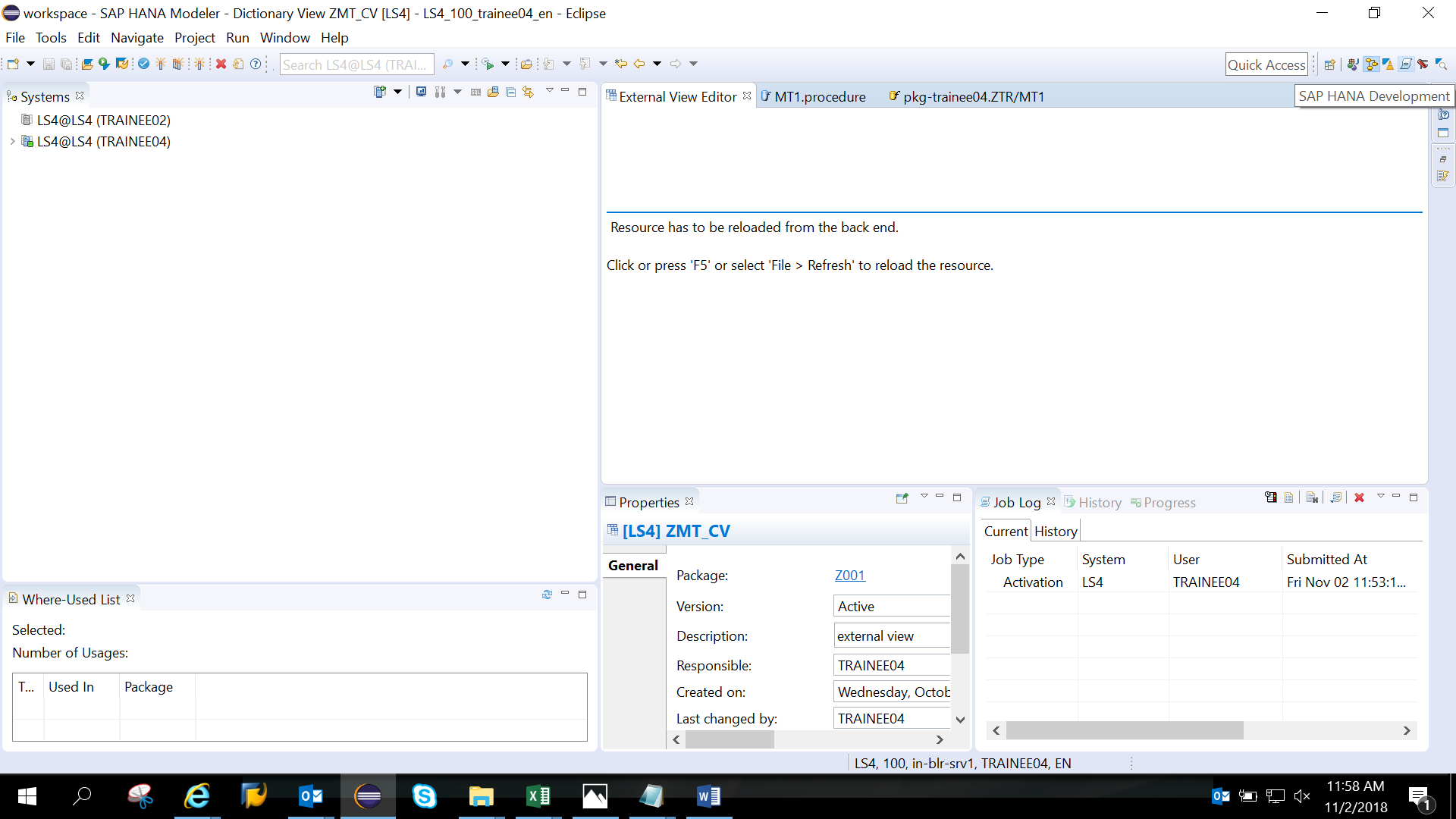
**CREATE PROCEDURES**

**First do the** UTF setting

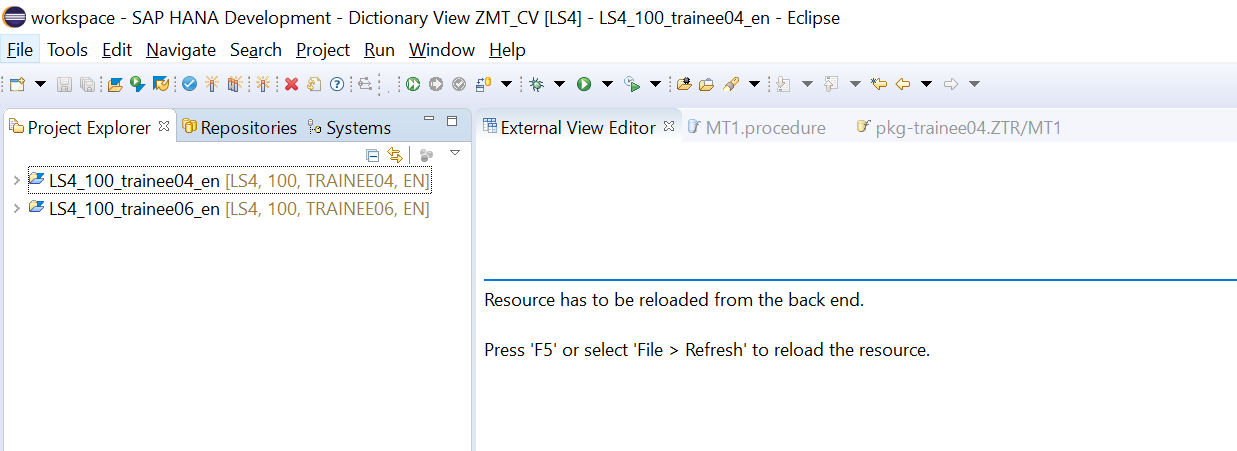
Windows-prefernece-general-workspace-Text file Encoding-other-utf-8



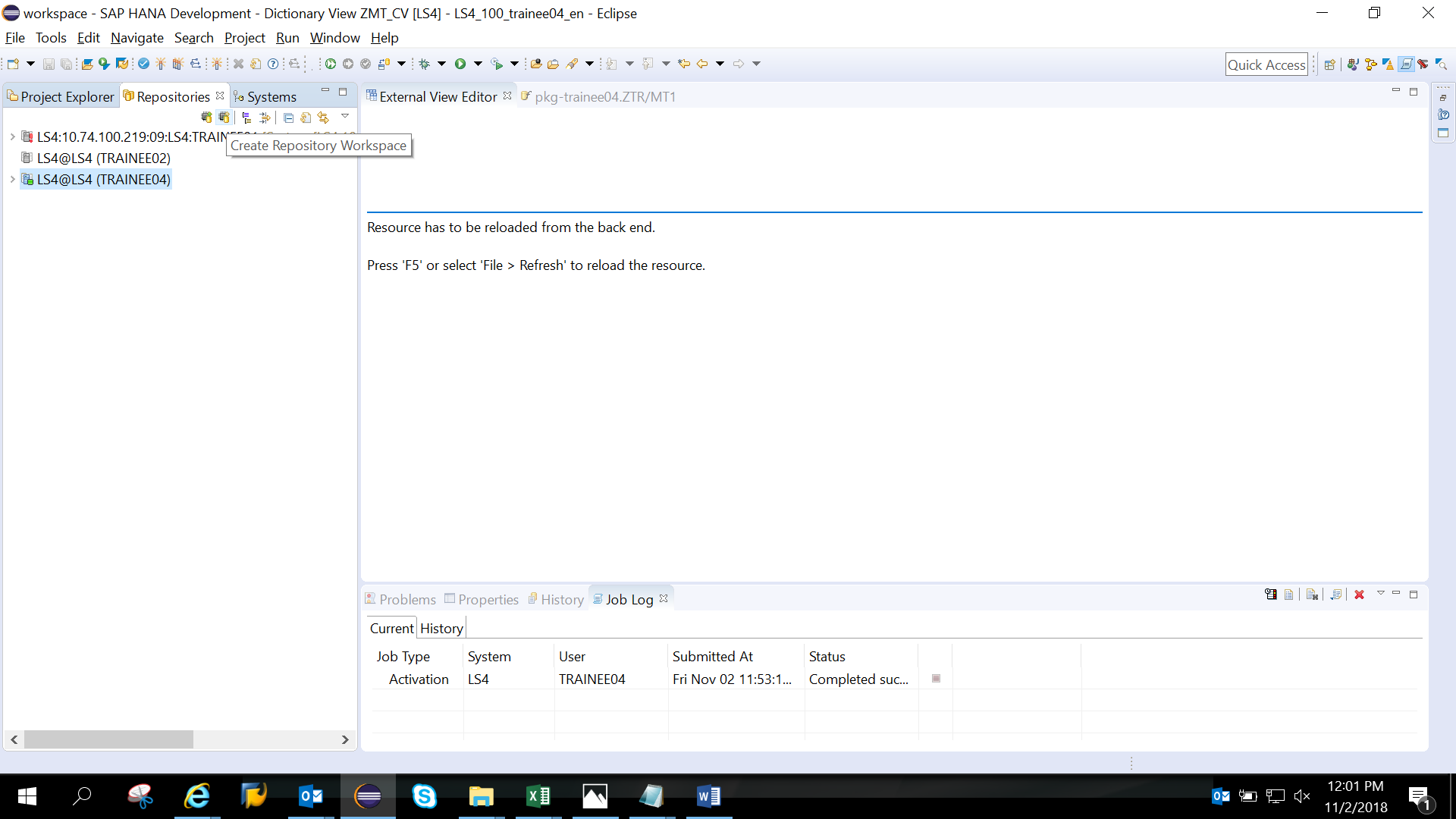
1. Open HANA Development perspective.



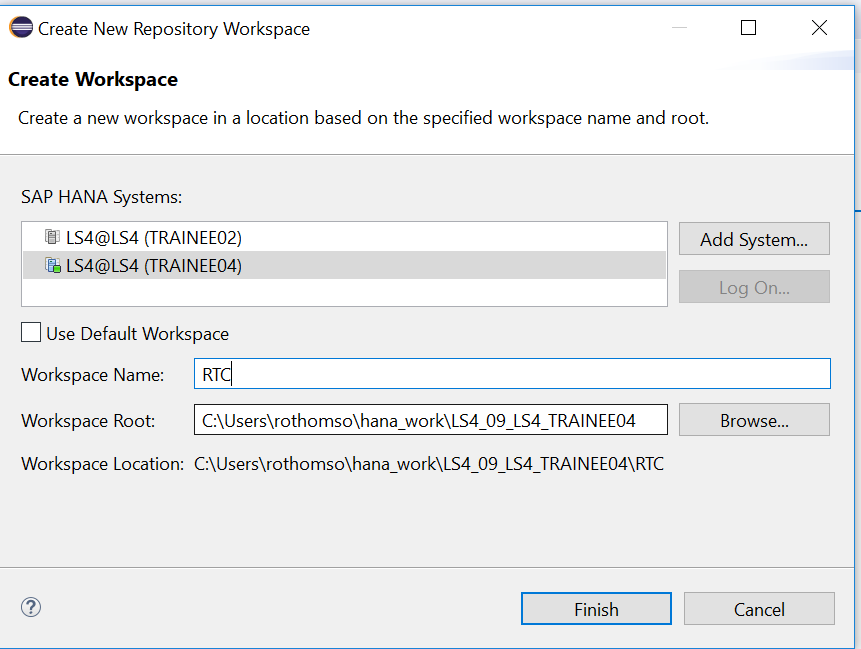
1. Below screen appears



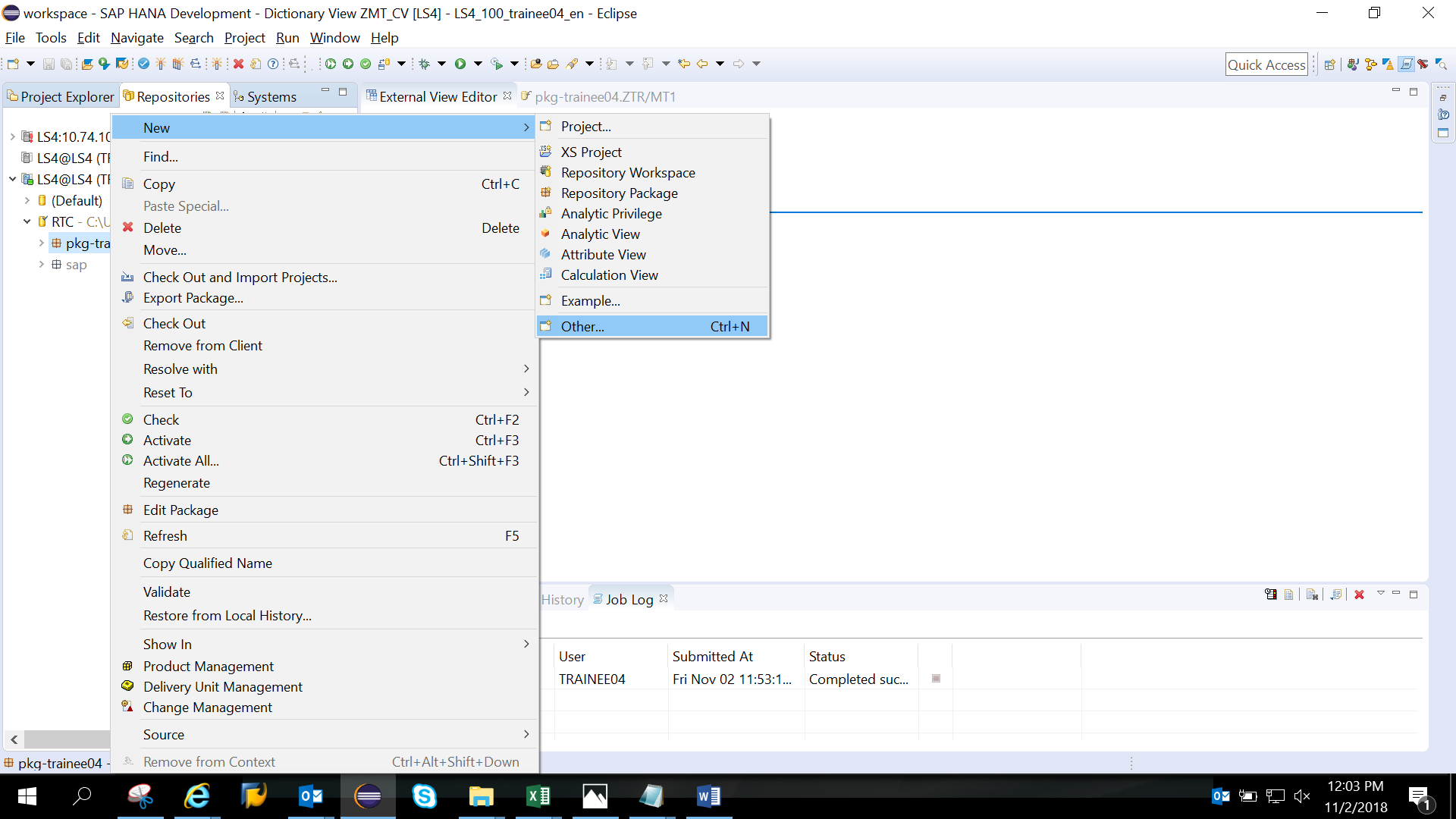
1. In repository tab click on create repository workspace.



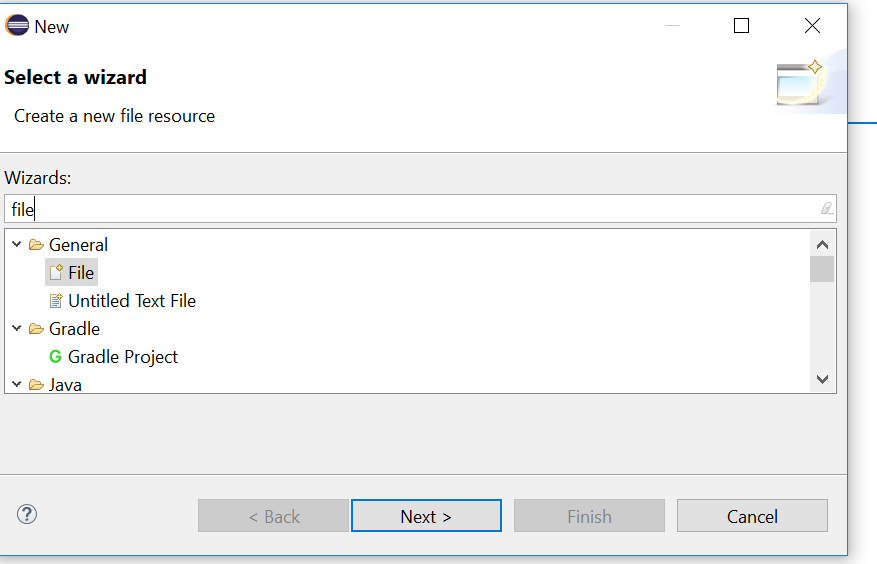
4) **Uncheck default and give details**



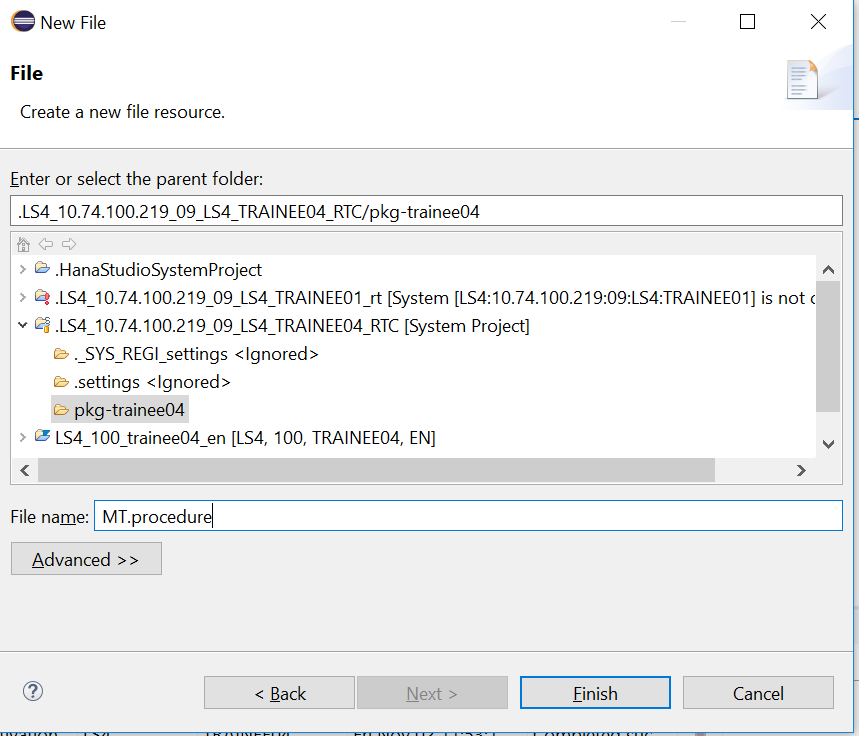
5) Right click on package as shown



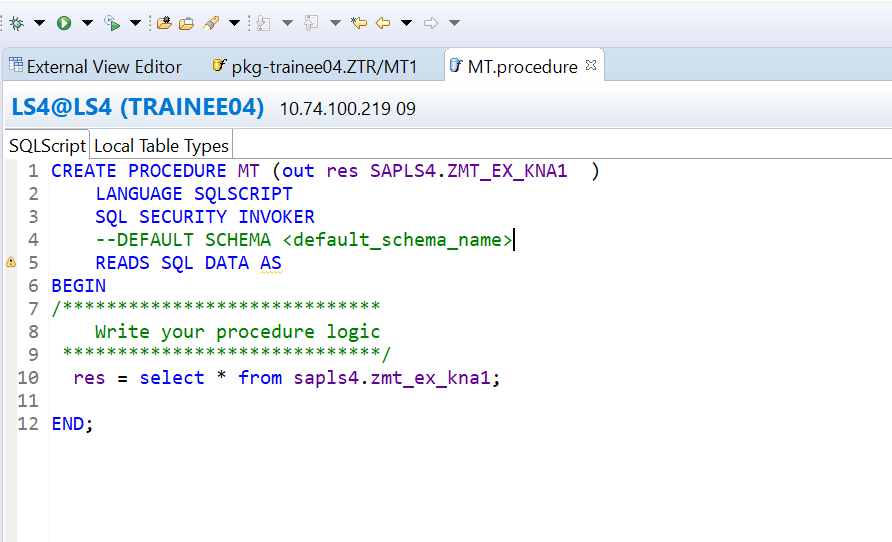
6)Search file



7)



8)edit the template of procedure as shown



CREATE PROCEDURE MTC ( out res SAPLS4.ZMT\_EX\_KNA1 )

LANGUAGE SQLSCRIPT

SQL SECURITY INVOKER

--DEFAULT SCHEMA <default\_schema\_name>

READS SQL DATA AS

BEGIN

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

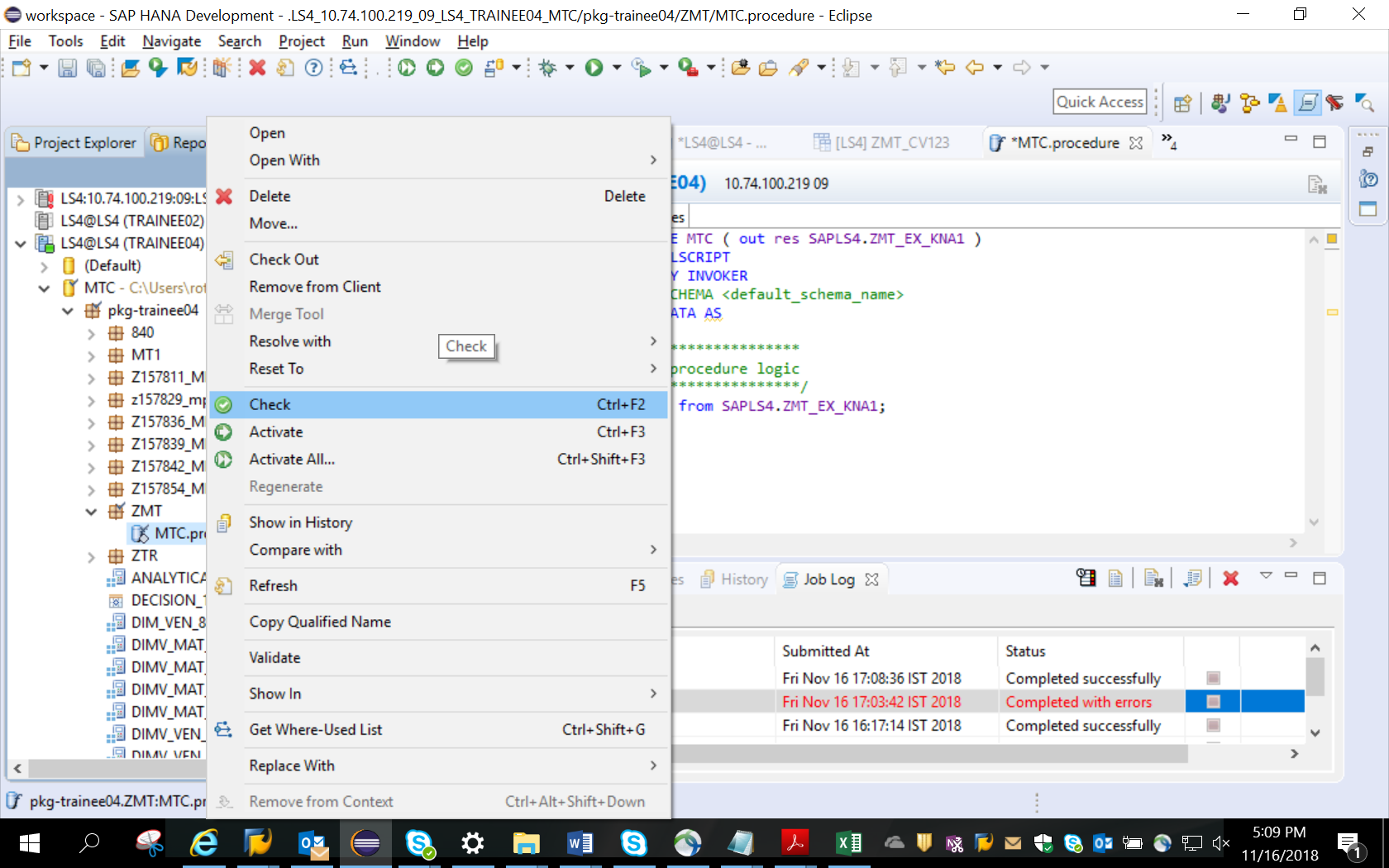
Write your procedure logic

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

res = select \* from SAPLS4.ZMT\_EX\_KNA1;

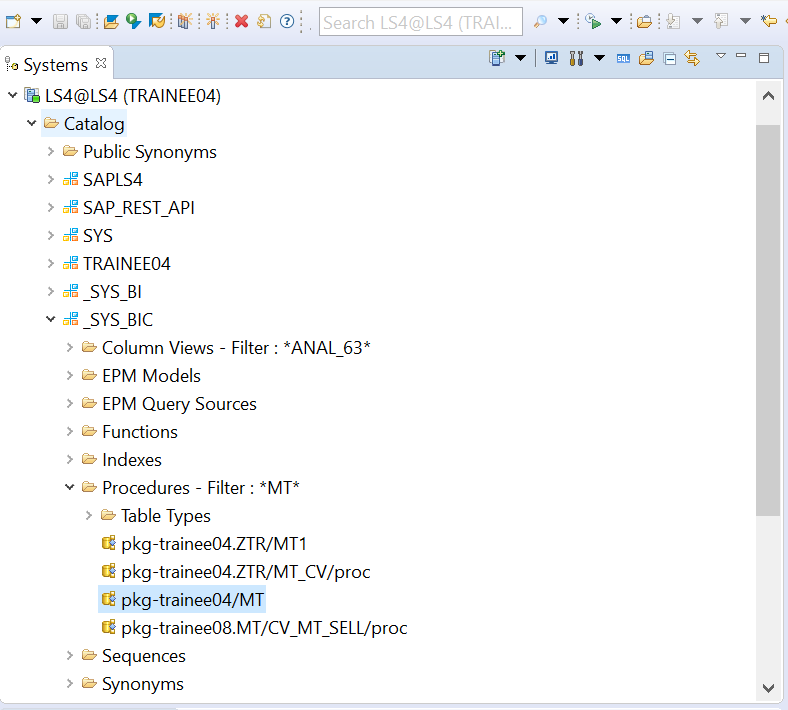
END;

8)Right click on procedure - check -activate it

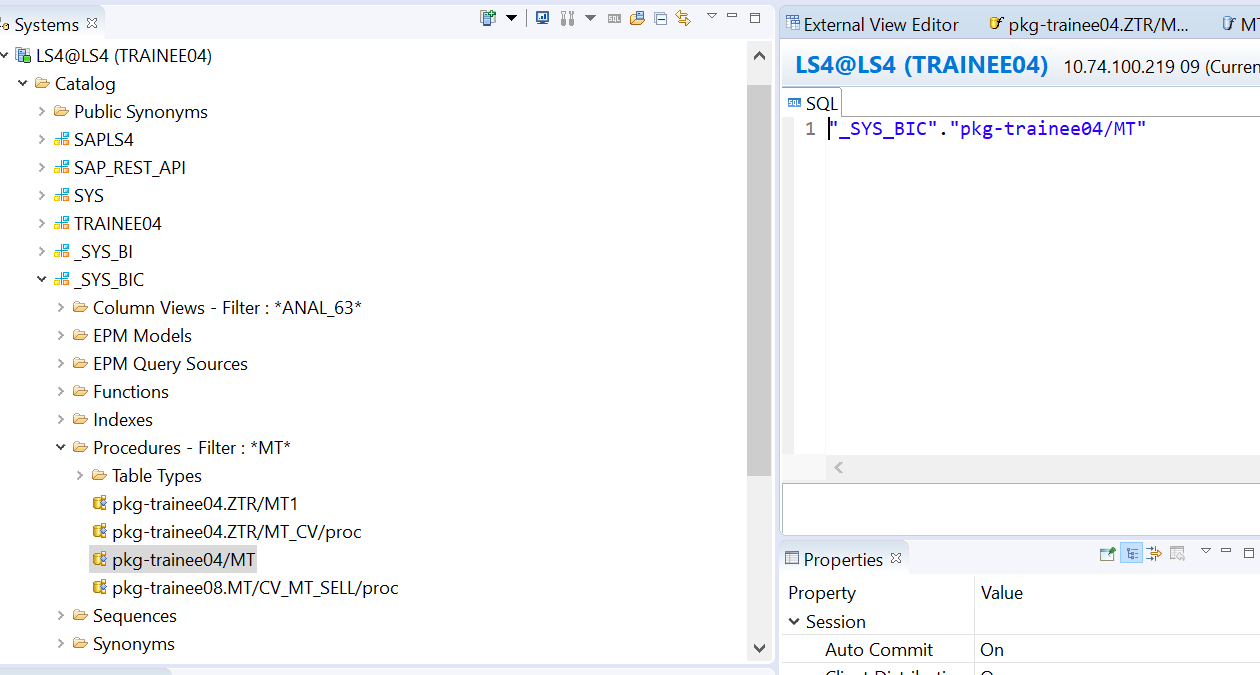


9)In modeler perspective- got to below path and search ur procedure

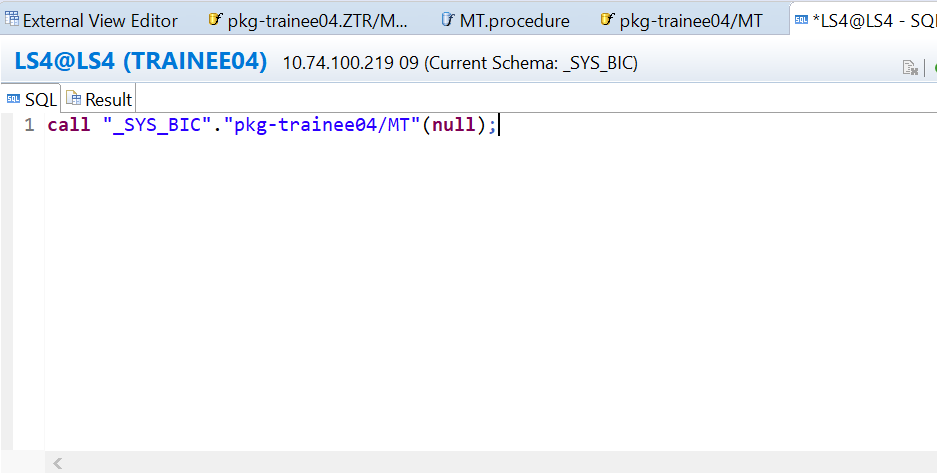
**Modeler perspective -SYS\_BIC-Procedures-MTC.procedure**



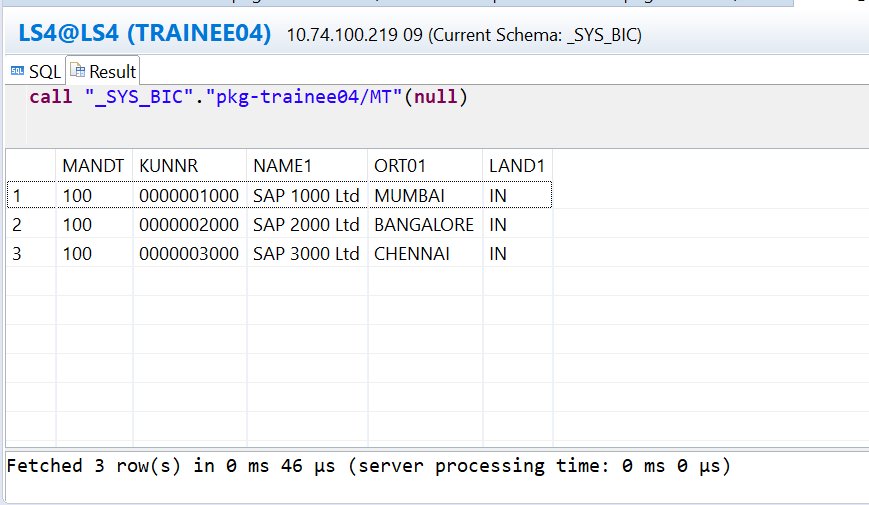
11) Open the SQL editor and drag and drop the procedure to the SQL editor



12) call the procedure by writing call statement.



13) execute



14) To consume this procedure from ABAP code, write ADBC program and replace v\_Sql with below code

V\_SQL = | CALL "\_SYS\_BIC"."pkg-trainee04/MT"(**null**);| .

***ADBC CODE***   
REPORT ZRT.  
  
data : wa\_SALES type ZMT\_EX\_KNA1,  
       it\_SALES type TABLE OF ZMT\_EX\_KNA1.  
  
 DATA: O\_conn   type REF TO cl\_sql\_connection,  
       o\_sql    type REF TO cl\_sql\_statement,  
       o\_result type REF TO cl\_sql\_result\_set,  
       V\_SQL    TYPE STRING,  
       o\_data   type REF TO data.  
  
\*TRY.  
\*Connect to the respective DB according connection  
CALL METHOD cl\_sql\_connection=>get\_connection  
\*  EXPORTING  
\*    con\_name = 'ADM\_HDB'  
\*    sharable = SPACE  
  RECEIVING  
    con\_ref  = O\_CONN  
    .  
\* CATCH cx\_sql\_exception .  
\*ENDTRY.  
  
CREATE OBJECT o\_sql  
  EXPORTING  
    con\_ref = O\_CONN  
\*    tab\_name\_for\_trace =  
    .  
  
V\_SQL =  | call "\_SYS\_BIC"."pkg-trainee04/MT"(null ) | .  
  
  
\*TRY.  
CALL METHOD o\_sql->execute\_query  
  EXPORTING  
    statement   = V\_SQL  
\*    hold\_cursor = SPACE  
  RECEIVING  
    result\_set  = O\_RESULT  
    .  
\* CATCH cx\_sql\_exception .  
\* CATCH cx\_parameter\_invalid .  
\*ENDTRY.  
  
get REFERENCE OF it\_SALES into o\_data.  
\*TRY.  
CALL METHOD o\_result->set\_param\_table  
  EXPORTING  
    itab\_ref             = o\_data  
\*    corresponding\_fields =  
\*    lob\_fields           =  
    .  
\* CATCH cx\_parameter\_invalid .  
\*ENDTRY.  
  
data v\_count type i.  
\*Data Records into internal table  
\*TRY.  
CALL METHOD o\_result->next\_package  
\*  EXPORTING  
\*    upto     = 0  
  RECEIVING  
    rows\_ret = v\_count  
    .  
\* CATCH cx\_sql\_exception .  
\* CATCH cx\_parameter\_invalid\_type .  
\*ENDTRY.  
  
cl\_demo\_output=>display( it\_SALES ).