

## Handling Case-Sensitive Data

Recently, I was looking at a requirement document to display the purchase requisition created by a user who wants to query purchase requisition data by the name of the requester. I have arrived at the conclusion that I need a case-insensitive database query.

But, what is a case-insensitive where clause? A little research revealed that this was nothing new. To implement a case-insensitive where clause in ABAP, we can use the native SQL UPPER () construct. This seemed like an easy nut to crack. But, as I soon found out, I had a lot to learn.

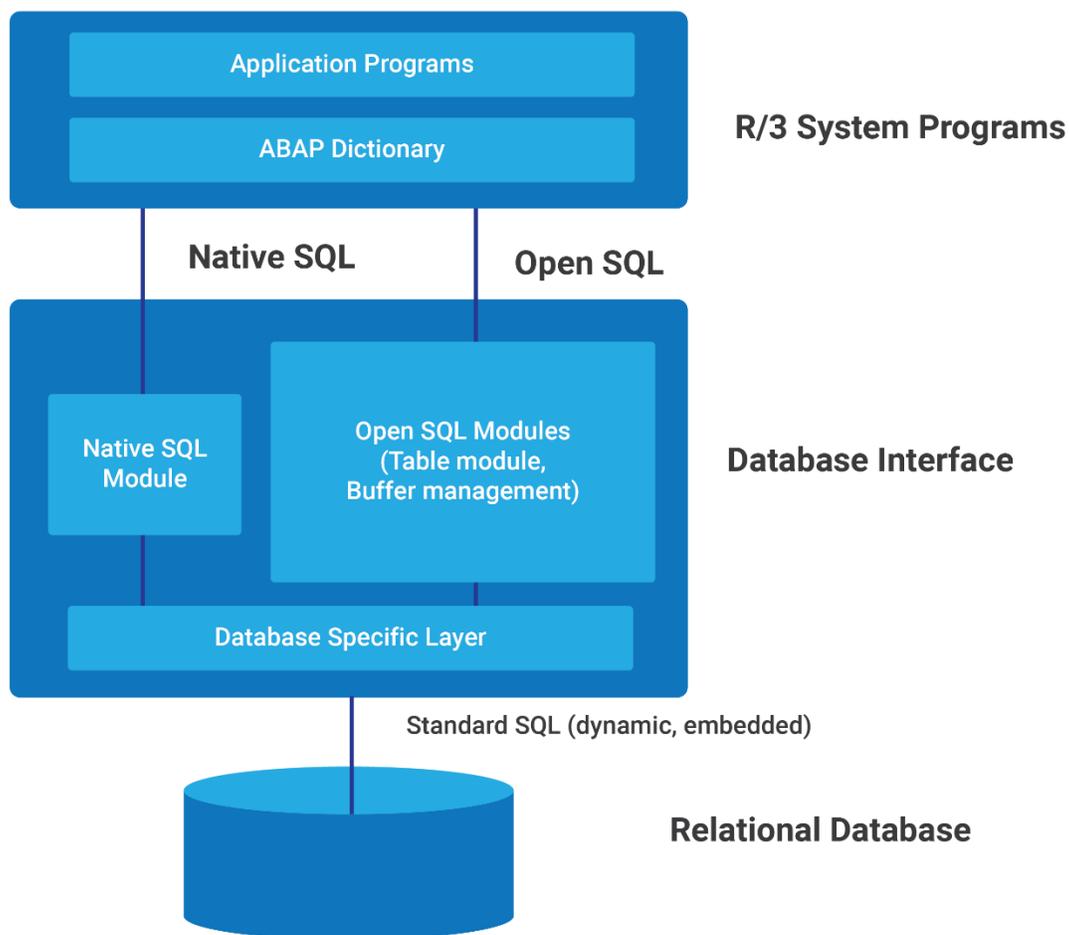
Native SQL statements embedded between EXEC SQL and ENDEXEC do not fall within the scope of ABAP and do not follow ABAP syntax as well. Mainly, database-specific SQL statements can be embedded and passed unchanged from the Native SQL interface to a database system for execution. The full SQL language scope of the relevant database can be used and the addressed database tables do not have to be declared in the ABAP Dictionary. In addition, a small set of SAP-specific Native SQL statement is available that can only be specified between EXEC SQL and ENDEXEC. This set is subject to special handling by the Native SQL interface.

ABAP Native SQL allows the inclusion of database-specific SQL statements in the ABAP program. Native SQL statements bypass the interface of R/3 database. In order to make sure that transactions in R/3 system are consistent, one should not use any transaction control statements (ROLLBACK WORK, COMMIT), or any statements which set transaction parameters by using Native SQL.

### Native SQL - Key points to remember

- The Native SQL statements should code in between the EXEC SQL and ENDEXEC statement
- These statements define an area in the ABAP program in which one or more Native SQL statements can be specified statically. The area between EXEC and ENDEXEC is not checked completely by the syntax check
- The statement ENDEXEC closes an embedded Native SQL statement introduced using EXEC SQL
- The period should be at the end of the EXEC SQL and ENDEXEC statements
- A period is not required after native SQL statements
- All Native SQL statements bypass SAP buffering and automatic client handling is not performed

- SAP ABAP comment statements (\*) will not be considered as a comment if they are coded in between EXEC SQL & ENDEXEC
- The statement ENDEXEC sets the system fields SY-SUBRC and SY-DBCNT. While using the obsolete addition PERFORMING, implicit cursor processing is carried out and the system fields are set for every reading
- In native SQL statements, the data is transferred between the database table and the ABAP program using host variables
- The host variable in the native SQL statement is represented with colon (:) preceding
- Only flat elementary fields and flat structures with elementary components can be used as host variables with one exception. If a structure is specified after the INTO clause by Native SQL, it is transformed by the Native SQL interface as if its components were specified as individual fields separated by commas



*Fig: Difference between Native SQL and Open SQL Approaches*

Let us see an example on how to use Native SQL syntax to fetch the Case Sensitive data

```

DATA: lv_afnam TYPE eban-afnam.
DATA : c1 TYPE cursor.

PARAMETERS: p_afnam TYPE eban-afnam.

EXEC SQL.
  OPEN C1 FOR
    SELECT AFNAM
      FROM EBAN
     WHERE MANDT = :SY-MANDT AND
           UPPER(AFNAM) = :p_afnam
ENDEXEC.
DO.
  EXEC SQL.
    FETCH NEXT C1 INTO :lv_afnam
  ENDEXEC.
  IF sy-subrc <> 0.
    EXIT.
  ELSE.
    WRITE : lv_afnam.
  ENDIF.
ENDDO.
EXEC SQL.
  CLOSE C1
ENDEXEC.

```

With ABAP 7.51 you can also write:

```

DATA:
  query TYPE string VALUE `ERROR`,
  rows TYPE i       VALUE 100.

SELECT arbgb, msgnr, text
  FROM t100
  WHERE sprsl = 'E' AND
        upper( text ) LIKE @query
  ORDER BY arbgb, msgnr, text
  INTO TABLE @DATA(result2)
  UP TO @rows ROWS.

```

The new built-in function UPPER in combination with LIKE enables a case insensitive search in Open SQL.

UPPER is not only available in Open SQL but in ABAP CDS also. With ABAP 7.51 one can define a CDS view as follows:

```

@AbapCatalog.sqlViewName: 'DEMOCDSUPPER'
@AccessControl.authorizationCheck: #NOT_REQUIRED
define view Demo_Cds_Upper
  as select from
    t100

```

```
{  
  sprsl,  
  arrgb,  
  msgnr,  
  text,  
  upper(text) as upper_text
```

Another SELECT statement uses the SQL function UPPER directly to access the same data source T100.

Since SQL functions cannot be used on the left side of LIKE in the DDL of the ABAP CDS, the view returns a help field which is evaluated in Open SQL. The results are the same. Thus, the search effected by the examples is not case-sensitive.

```
SELECT arrgb, msgnr, text  
  FROM demo_cds_upper  
  WHERE sprsl = 'E' AND  
        upper_text LIKE @query  
  ORDER BY arrgb, msgnr, text  
  INTO TABLE @DATA (result)  
  UP TO @rows ROWS.
```

With the help of the above examples, we can understand how a case-sensitive data can be handled with native SQL statements and CDS views in various SAP versions.

## Contact for further details



### **Madhulatha Gummadi**

Sr. Technical Consultant - ERP SAP Technical

[madhulathag.in@mouritech.com](mailto:madhulathag.in@mouritech.com)

**MOURI Tech**