Capgemini

SAP HANA

Lesson Name: Open SQL-Revisited

Lesson Objectives

After completing this lesson, participants will be able to –

- Know about basics of OPEN SQL
- Features of New OPEN SQL

Contents

Introduction to OPEN SQL

- SQL consists of statements that *perform operations on the central database*

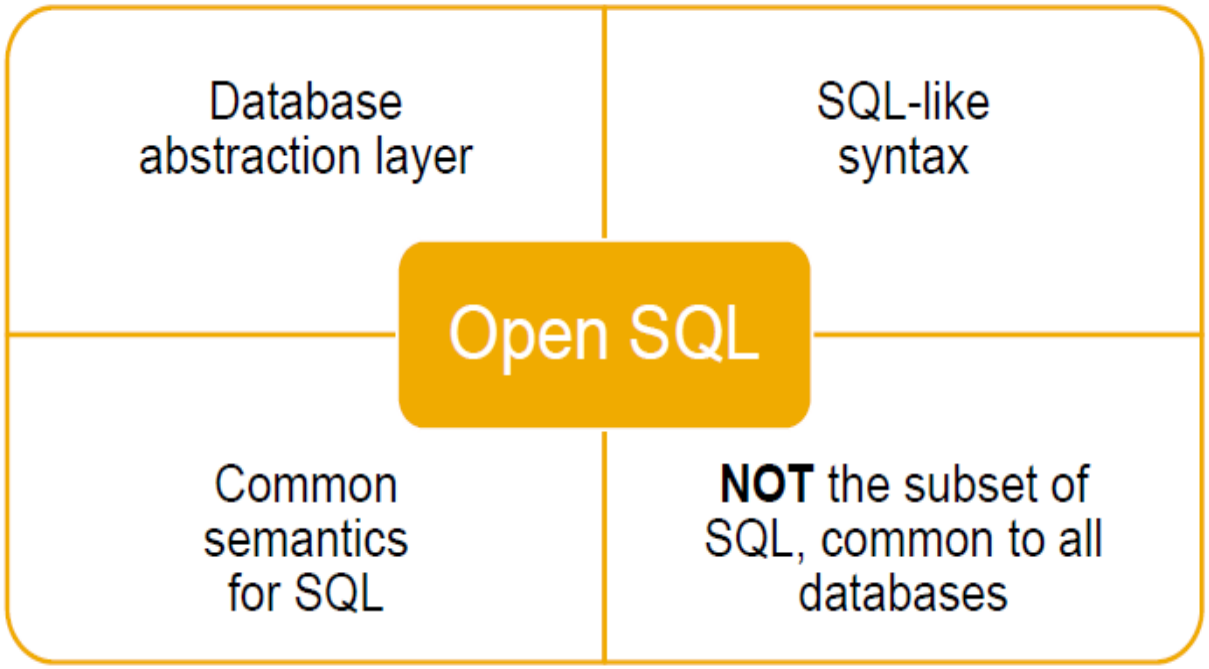
- Open SQL *is called Open because it is database independent*.

Open = Platform independent.

- Open SQL is the only DB abstraction layer that defines a common semantic for all SAP-supported databases.

- There is very less efforts in migrating Open SQL from one database to another because it has the same semantics on all databases..

Introduction to OPEN SQL

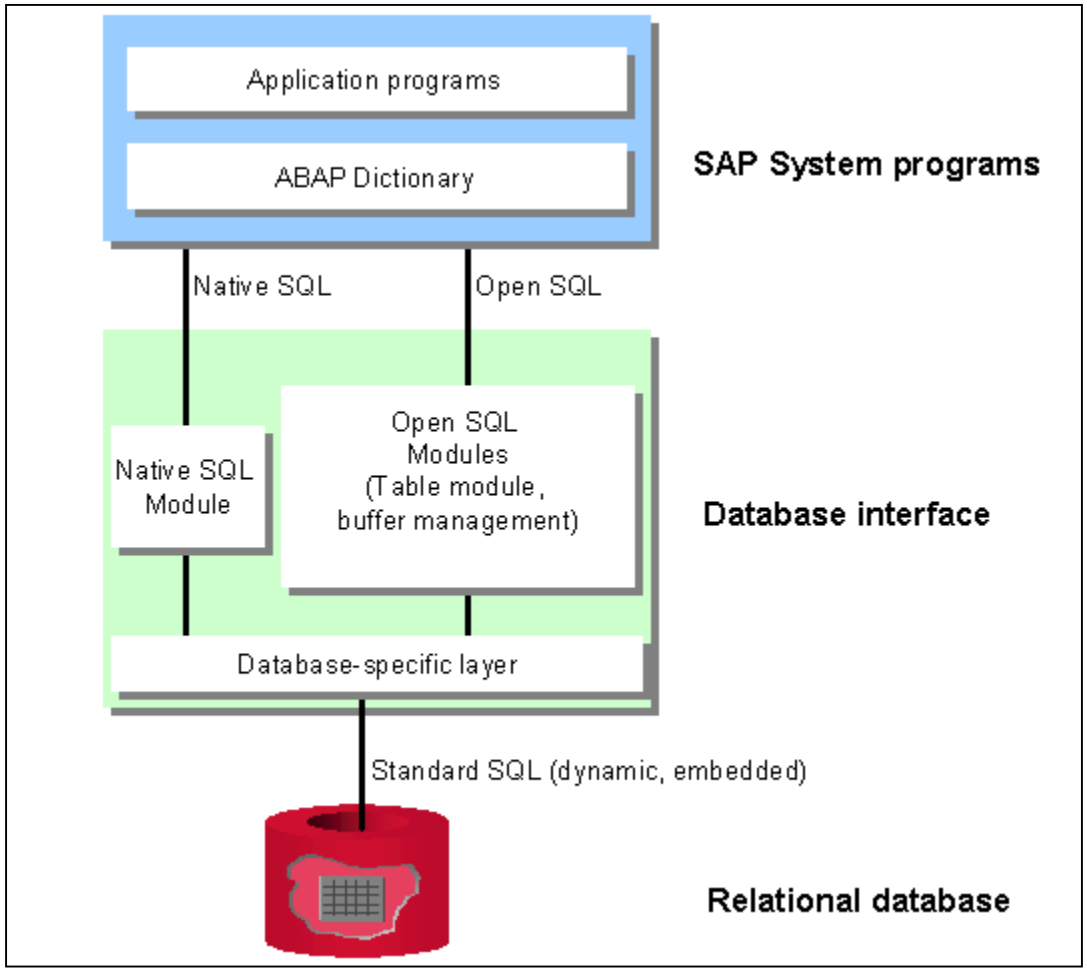


Open SQL is the only DB **abstraction layer** that defines a **common semantic** for all SAP-supported databases!

Features of OPEN SQL

- Native SQL allows you to use database-specific SQL statements in an ABAP/4 program.
- Open SQL allows you to access the database tables declared in the ABAP dictionary regardless of the database platform that the R/3 system is using.
  - This means that you can use database tables that are not administered by ABAP dictionary.

Features of OPEN SQL



Open SQL - Performance Rules:

To improve the performance of the ABAP program, one should take care of the following rules-

Keep the Result Set Small

- Using the where clause.
- If only one record is required from the database, use SELECT SINGLE whenever possible .

Minimize the Amount of Data Transferred

- Restrict the number of lines
- If only certain fields are required from a table, use the SELECT <field1> <field2> INTO ... statement
- Restrict no of columns
- Use aggregate functions

Using Internal Tables to Buffer Records

- To avoid executing the same SELECT multiple times (and therefore have duplicate selects), an internal table of type HASHED can be used to improve performance.

Open SQL - Performance Rules:

Minimize the Number of Data Transfers

- Avoid nested select loops
- An alternative option is to use the SELECT .. FOR ALL ENTRIES statement. This statement can often be a lot more efficient than performing a large number of SELECT or SELECT SINGLE statements during a LOOP of an internal table.
- Use dictionary views
- Use Joins in the FROM clause
- Use subqueries in the where clause

Minimize the Search Overhead

- Use index fields in the where clause
- When accessing databases, always ensure that the correct index is being used .

Reduce the Database Load

- Buffering
- Logical databases
- Avoid repeated database access

Limitations of OPEN SQL:

Open SQL can *only work with database tables* that *have been created in the ABAP Dictionary*.

OPEN SQL does not support DML statements like create table.

OPEN SQL does not support “advanced” SQL statements like TRUNCATE, MERGE, ROLLUP.

In OPEN SQL you can’t use aggregate functions like sum, avg.

OPEN SQL does not support most column functions like SUBSTR, CONCAT (||) and “case expression” in both the select and where clauses.

Limitations of OPEN SQL:

OPEN SQL does not allow you to write predicates (where conditions) between more than one column … so you can't write the following condition:
where t1.col1 <> t1.col2

OPEN SQL does not allow to write predicates (where conditions) on columns of a table joined with left (or right) join.

So the following SQL is not valid:
select … from t1 left join t2 where t2.col = 'x'

Features of New Open SQL

- Syntax enhancements
- SELECT list enhancements
- Aggregation functions
- Literal Values
- Arithmetical expressions
- Open SQL enhancements

New features of OPEN SQL:

Select List enhancements:

- Conditional expressions like CASE statement can be used in Select .

CASE Expression

```
"simple case
SELECT so_id,
         CASE delivery_status
           WHEN ' ' THEN 'OPEN'
           WHEN 'D' THEN 'DELIVERED'
           ELSE delivery_status
         END AS delivery_status_long
   FROM snwd_so
   INTO TABLE @DATA(lt_simple_case).

"searched case
SELECT so_id,
        CASE
          WHEN gross_amount > 1000
            THEN 'High volume sales order'
          ELSE ' '
        END AS volumn_order
    FROM snwd_so
    INTO TABLE @DATA(lt_searched_case).
```

New features of OPEN SQL:

Aggregate functions:

Aggregate functions operate on multiple records to calculate one value from a group of values.

Eg. Select Sum(Sales) from table_name where Column1='ABC';

Sum() - returns the sum of the numeric values in a given column

Max() - returns the maximum of the numeric values in a given column

```
SELECT bp_id,
       company_name,
       so~currency_code,
       SUM( so~gross_amount )
        AS total_amount
  FROM snwd_so AS so
  INNER JOIN snwd_bpa AS bpa
  ON bpa~node_key = so~buyer_guid
  INTO TABLE @DATA(lt_result)
  WHERE so~delivery_status = ' '
  GROUP BY
    bp_id,
    company_name,
    so~currency_code
  HAVING SUM( so~gross_amount ) > 10000000.
```

New features of OPEN SQL:

Literal Values can be used in the SELECT list

```
SELECT so~so_id,
       'X' AS literal_x,
       42 AS literal_42
  FROM snwd_so AS so
  INTO TABLE @DATA(lt_result).

DATA lv_exists TYPE abap_bool
               VALUE abap_false.

SELECT SINGLE @abap_true
  FROM snwd_so
  INTO @lv_exists.

IF lv_exists = abap_true.
  "do some awesome application logic
ELSE.
  "no sales order exists
ENDIF.
```

New features of OPEN SQL:

Arithmetic Expressions

- Expressions like +, -, *, DIV, MOD, ABS, FLOOR, CEIL can be used in the SELECT statement

```
DATA lv_discount TYPE p LENGTH 1 DECIMALS 1
                 VALUE '0.8'.

SELECT ( 1 + 1 ) AS two,
       ( @lv_discount * gross_amount )
           AS red_gross_amount,
       CEIL( gross_amount )
           AS ceiled_gross_amount
  FROM snwd_so
  INTO TABLE @DATA(lt_result).
```

New features of OPEN SQL:

Open SQL is enhanced

- SQL Expressions is enhanced using
  - HAVING clause
  - JOIN statements
  - Client handling

HAVING Clause

```
SELECT bp_id,
       company_name,
       so~currency_code,
       SUM( so~gross_amount )
        AS total_amount
  FROM snwd_so AS so
  INNER JOIN snwd_bpa AS bpa
  ON bpa~node_key = so~buyer_guid
  INTO TABLE @DATA(lt_result)
  WHERE so~delivery_status = ' '
  GROUP BY
    bp_id,
    company_name,
    so~currency_code
  HAVING SUM( so~gross_amount ) > 10000000.
```

```
SELECT
  bp_id,
  company_name,
  so~currency_code,
  so~gross_amount
FROM snwd_so AS so
INNER JOIN snwd_bpa AS bpa
  ON so~buyer_guid = bpa~node_key
  USING CLIENT '111'
INTO TABLE @DATA(lt_result).
```

Summary

In this lesson, you have learnt:

- Basic Concepts of Open SQL
- Features of Open SQL
- Open SQL Syntaxes and Statements
- Performance Rules and Limitations of Open SQL

Review Questions

OPEN SQL Statements are those statements which are used to -------- or --------- database table data.

For OPEN SQL statements insertion in database table is possible in ------- way/ways.

Open SQL in ABAP application server is the ------- ---------layer calling an SQL like syntax.