

1. What are the different SQL statements used in abap/4?

a) **Open SQL.**

They are a set of ABAP/4 commands, which perform operation on database tables. The results of these operations and the associated error messages are independent of database systems used. Open SQL thus offers a unified SQL syntax and semantics of different database systems.

To avoid incompatibilities between different database tables and also to make ABAP programs independent of the database system in use, SAP has created a set of separate SQL statements called Open SQL. Open SQL contains a subset of standard SQL statements as well as some enhancements, which are specific to SAP. Using Open SQL enables you to access any database tables available to the SAP system, regardless of the manufacturer.

There are no DDL and DCL statements in Open SQL.

The open SQL command set include the following command –

SELECT, INSERT, UPDATE, DELETE, MODIFY, OPEN CURSOR, FETCH, CLOSE CURSOR, COMMIT WORK, ROLLBACK WORK.

All SQL Statements used in ABAP other than native SQL.

NOTE : TO Execute an open SQL command:

- 1) The addressed database system is supported by SAP.
- 2) The database table is defined in ABAP dictionary.

b) **Native SQL.**

These are database specific SQL statements or the ANSI SQL which all RDBMS supports..

Syntax

EXEC SQL [PERFORMING <form>].
<Native SQL statement> [;]
ENDEXEC.

Advantage :

- 1) Tables which are not declared in ABAP dictionary can be accessed. (e.g. TABLES belonging to sys or system user of oracle etc.)
- 2) To use some of the special features supported by the database specific SQL. (e.g. Passing hints to Oracle optimizer.)

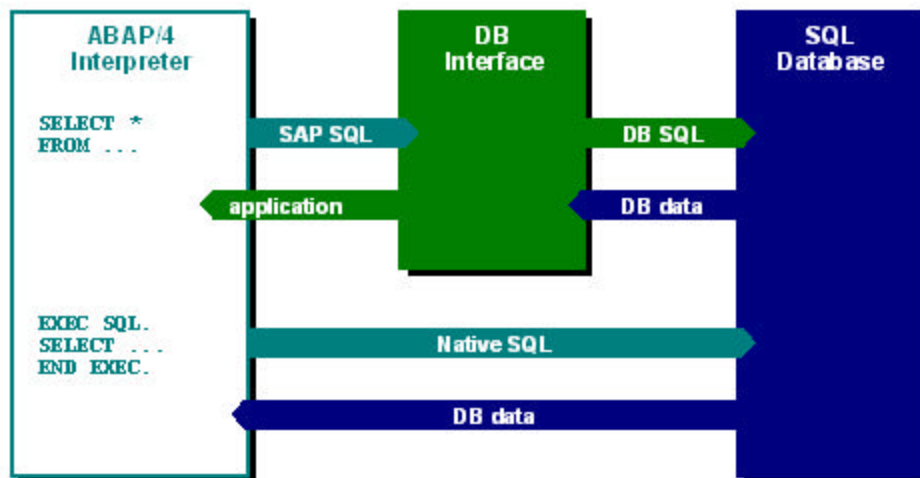
Disadvantage .:

- 1) No syntax check is performed what ever is written between EXEC & ENDEXEC.
- 2) ABAP program containing database-specific SQL statements will **not** run under different database systems.
- 3) There is no automatic client handling for client dependent tables.
- 4) Care has to be taken during migration to higher versions.

SAP R/3 Document – FAQ SQL / Performance

NOTE : Use of Native SQL is highly discouraged by SAP.

The difference between Open SQL and Native SQL is shown in the following diagram:



A database interface translates SAP's Open SQL statements into SQL commands specific to the database in use. Native SQL statements access the database directly.

NOTE :

2. How to take care for performance in ABAP development?

- ◆ **Keep the data selection small.**
 - To avoid transporting unnecessary data across the network
 - Always use the WHERE clause
 - Avoid selecting useless data that you filter out later (using CHECK, for example).
 - Use the indexes of the relevant database tables to make your WHERE clause more efficient, by checking all index fields for equality (EQ, =) and using the AND operator
 - Avoid using complex WHERE clauses, since the system has to break them down into several individual statements for the database system
 - If possible, avoid using the NOT operator in the WHERE clause, because it is not supported by database indexes; invert the logical expression instead.
- ◆ **Transport as little data as possible.**
 - Transport only the fields of the database table that you really need. If you do not need all of the fields in a table, use a field list in the SELECT clause instead of SELECT *.
 - Use the aggregate functions in the SELECT clause for calculations, instead of transporting large amounts of data and then performing the equivalent calculation.
 - Use the UPDATE statement sparingly: Only update the columns that have actually changed, and do not overwrite the entire line.

SAP R/3 Document – FAQ SQL / Performance

- Note here that the addition INTO CORRESPONDING FIELDS in the INTO clause of the SELECT statement is only effective for large amounts of data, because the time required to compare the field names is otherwise too great.
- Consider using the DISTINCT option if you are expecting a lot of duplicate table entries.
- **Use fewer database accesses**
 - Transfer all of the data at once from the database into internal tables
 - Where possible, avoid accessing the same data more than once (for example, by using SELECT before an UPDATE or DELETE statement).
 - Avoid nested SELECT loops. Instead, use an internal table and a second SELECT statement with the FOR ALL ENTRIES addition.
 - In exceptional cases, you can also select data using a separate cursor.
- **Using Database Buffering**
 - Saving database tables in local buffers can save a considerable amount of time. Wherever possible, use buffered data, and only use the BYPASSING BUFFER addition where absolutely necessary.
 - Note that the following additions automatically bypass the buffer: DISTINCT, SINGLE FOR UPDATE, and aggregate functions in the SELECT clause.

TIPS:

- ❑ You can check the performance of your SQL or ABAP functions by using the **SQL Trace** and **'Runtime analysis'** utilities.
- ❑ For tips on how to improve the performance of ABAP tasks, choose transaction **SE30**

3. What is Runtime analysis and its benefits?

Definition: The Runtime Analysis tool lets you analyze the performance of any transaction or program created within the ABAP Workbench. The Runtime Analysis tool creates lists that reveal expensive statements, summarize table accesses, and show the hierarchy of a program's overall flow. This information enables you to detect and analyze problems resulting from:

- Overuse of or unnecessary calling of modularization units (such as subroutines or function modules) and ABAP statements
- CPU-intensive programming functions
- User-programmed functions that replace existing ABAP statements
- Inefficient and unnecessary database accesses
- find out the database tables accessed by an ABAP program.
- find out resource utilization for the distributed environment of processing for all the 3 tiers of SAP.

4. What is SQL trace tool?

The SQL Trace tool lets you examine the database calls of reports and transactions. This tool shows you:

- the SQL statements that your application uses.
- which values the system uses for specific database accesses and changes
- how the system translates ABAP OPEN SQL commands (such as SELECT) into standard SQL commands

SAP R/3 Document – FAQ SQL / Performance

- where your application makes unnecessary database accesses or repeated accesses
- where your application positions COMMIT statements
- what database accesses or changes occur in the update section of your application
- which index and with what cost it is used for data retrieval.

NOTE : Only one user can use the trace tool at a time. So, it is very important to switch off the tool as soon as the execution is over.

5. What is Explain SQL function?

The *Explain SQL* function provides you with an analysis of a database's strategy for accessing any ABAP Dictionary object. You can use this analysis to identify the indexes used for database access.

6. When to buffer a table?

A table should be buffered when it is

- Rather small in size
- Accessed mostly for read purposes
- Changed very infrequently

Tables which are very good candidate for buffering:

- Control tables / customizing tables
- “SMALL” master data tables

7. Which SQL statement bypasses the table buffer?

Following are the SQL statements which bypasses table buffer:

- SELECT ----- BYPASSING BUFFER.
- SELECT from database views (projection views are OK)
- SELECT ---- DISTINCT -----.
- SELECT ---- COUNT, SUM, AVG, MIN, MAX.
- SELECT ---- ORDER BY (other than the primary key)
- SELECT ---- FOR UPDATE. (EXEC ----- ENDEXEC)
- Native SQL statements.

NOTE : Avoid this statements when working with buffered tables

8. What is an expensive select statement?

- Long response time
- Many data blocks are scanned to find the selected records.
- Long DB request time caused by too many buffer gets.

9. Explain the open SQL statement modify?

This is used for Adding or changing lines in DB tables.

To insert a line into a database table, regardless of whether the primary key of this line already exists, you use the MODIFY statement.

There are two possibilities:

- If the database table contains no line with the same primary key as the line to be inserted, MODIFY works like INSERT, i.e. the line is added.

SAP R/3 Document – FAQ SQL / Performance

- If the database already contains a line with the same primary key as the line to be inserted, MODIFY works like UPDATE, i.e. the line is changed.

Note : For performance reasons, you should use MODIFY only if you cannot distinguish between these two options in your ABAP program.

10 . How automatic client handling is switched off in Open SQL?

The CLIENT SPECIFIED option switches off automatic client handling. You can then specify the client in a WHERE condition, and fill the client field in table work areas.