Complete Migration Guide: SQL Server to Oracle Database

Step 1: Pre-Migration Preparation

Before migrating the data and schema, it’s crucial to ensure that everything is ready for the migration process.

Assess the SQL Server Environment:

Inventory: Take an inventory of the SQL Server database, including tables, views, stored procedures, functions, indexes, triggers, and other objects.

Data Quality Check: Clean up any redundant or obsolete data in SQL Server. Ensure that the data is in good condition for migration (e.g., no data corruption, consistent data types).

Assess the Oracle Environment:

Oracle Database Version: Ensure that your Oracle environment is up-to-date and compatible with the version you are migrating to.

Check Resources: Ensure sufficient disk space, memory, and compute resources on the Oracle database server to accommodate the incoming data.

Plan Migration Strategy:

Data Migration Tools: Determine the migration tools that will be used (Oracle SQL Developer, Oracle Data Pump, third-party tools, etc.).

Downtime Planning: If the migration will involve production data, plan for downtime (if required) to ensure minimal disruption during the migration.

Step 2: Install Oracle SQL Developer and jTDS JDBC Driver

To facilitate communication between SQL Server and Oracle, you need to install the necessary tools:

Install Oracle SQL Developer:

Download and install Oracle SQL Developer from the Oracle website.

Oracle SQL Developer is a free tool that allows you to manage Oracle databases and migrate data from various sources like SQL Server.

Install jTDS JDBC Driver:

Download the jTDS JDBC driver (for SQL Server and Sybase) from the official jTDS website.

The jtds-1.3.1-dist driver allows SQL Developer to connect to SQL Server.

Configure SQL Developer to Use jTDS:

Open SQL Developer and navigate to Tools > Preferences > Database > Third Party JDBC Drivers.

Click Add Entry and select the jtds-1.3.1.jar file to enable SQL Developer to connect to SQL Server.

Step 3: Establish Connection to SQL Server

Once the required tools are installed, the next step is to connect SQL Developer to SQL Server.

Create a New Connection in SQL Developer:

Open SQL Developer and click New Connection.

Select SQL Server as the database type (this will now be available after configuring the jTDS driver).

Provide the connection details:

Username: SQL Server user (e.g., dangchou).

Password: SQL Server user password.

Host: The IP address or hostname of the SQL Server.

Port: The port used by SQL Server (default is 1433).

Click Test to ensure that the connection is successful.

Step 4: Schema Migration

Now that you are connected to SQL Server, the next step is migrating the schema (structure of tables, views, procedures, etc.) from SQL Server to Oracle.

Use the Migration Wizard in SQL Developer:

Start Migration: In SQL Developer, go to Tools > Database Migration.

Select SQL Server as the source database and Oracle as the target database.

Connect to SQL Server: Provide the connection details to your SQL Server instance.

Connect to Oracle: Provide the connection details for your Oracle database.

Analyze the SQL Server Schema: SQL Developer will analyze the SQL Server database and generate a list of objects (tables, views, indexes, etc.) for migration.

Mapping Data Types: SQL Developer will attempt to map SQL Server data types to equivalent Oracle data types. You can adjust the mapping manually if needed.

Generate DDL Scripts: SQL Developer will generate DDL scripts to create the schema in Oracle (tables, indexes, constraints, etc.).

Review and Apply the DDL Scripts:

Review the generated DDL scripts for any potential issues (e.g., unsupported SQL Server features or data type differences).

Execute the scripts in Oracle to create the schema.

Step 5: Data Migration

Once the schema is migrated, the next step is transferring the data from SQL Server to Oracle.

Data Migration Using SQL Developer:

Select Tables for Data Transfer: In SQL Developer, select the tables you wish to migrate from SQL Server to Oracle.

Data Transfer Options: Choose from the following options to transfer the data:

Direct Insert: SQL Developer will extract data from SQL Server and insert it into the corresponding Oracle tables.

Export/Import: If needed, export data from SQL Server to a file (e.g., CSV) and import it into Oracle using SQL Developer or Oracle SQL*Loader.

Data Integrity Check: Ensure that the data in Oracle matches the data in SQL Server. This may involve performing queries to verify that row counts, field values, and relationships are correct.

Step 6: Post-Migration Tasks

Once the schema and data are migrated, there are several post-migration tasks to complete.

Check for Compatibility Issues:

Ensure that stored procedures, functions, and triggers from SQL Server are correctly translated into their Oracle equivalents. Some SQL Server-specific code may need to be modified to work in Oracle.

Data Validation:

Perform data validation to ensure that the migrated data is correct. This may involve comparing record counts, checking for missing data, and verifying that data types are correct.

You can run test queries to check that the migrated data is functioning correctly in the Oracle environment.

Application Testing:

If you are migrating a database that is used by an application, test the application thoroughly to ensure that it works correctly with the Oracle database.

Modify any application code that was specific to SQL Server, such as database connection strings, queries, or stored procedure calls.

Performance Tuning:

After the migration, perform performance tuning on Oracle to ensure that the queries and operations are running efficiently.

You may need to create new indexes or optimize existing queries to improve performance.

Backup the Oracle Database:

Once the migration is complete and validated, make sure to take a full backup of the Oracle database to safeguard your data.

Step 7: Final Review and Monitoring

Monitor the Oracle Database:

Regularly monitor the performance and logs of the Oracle database to ensure smooth operation after migration.

Set up monitoring tools to track database health, such as Oracle Enterprise Manager or other third-party tools.

Decommission SQL Server (if required):

If the migration is successful and the Oracle database is now fully operational, you can consider decommissioning the SQL Server instance, depending on your business requirements.

====================GOOD LUCK =====================