Oracle SQL Learning Environment Setup

Transitional SQL

May 26, 2025

Contents

1 Introduction

Welcome to the Oracle SQL Learning Environment Setup guide. This document provides step-by-step instructions to install and configure Oracle Database (Express Edition or 23c Free Developer Release) and Oracle SQL Developer on Debian 12, Windows, and macOS. The goal is to establish a robust local environment for practicing Oracle SQL and PL/SQL.

Core Tools

- **Oracle Database Free:** Includes Oracle Database Express Edition (XE) and Oracle Database 23c Free Developer Release. We'll use "Oracle Database Free" as a general term.
- **Oracle SQL Developer:** A graphical IDE for interacting with your Oracle database.

General Prerequisites (All Operating Systems)

- **Oracle Account:** A free Oracle account is required for software downloads. Create one at https://profile.oracle.com/myprofile/account/create-account.jspx.
- **System Resources:**
  - RAM: Minimum 4GB (8GB+ recommended).
  - Disk Space: Minimum 15-20GB free space.
  - Always check specific version requirements on the Oracle download page.
- **Java Development Kit (JDK):** Oracle SQL Developer requires a JDK (typically version 11, 17, or newer). Oracle JDK or OpenJDK can be used.

2 Debian 12 Setup (via Docker)

For Debian 12, using Docker is highly recommended due to the absence of official `.deb` packages from Oracle for its database, simplifying installation and avoiding potential dependency issues.

2.1 Install Docker Engine

If Docker is not already installed on your Debian 12 system:

```
# Update package lists
sudo apt update

# Install prerequisites
sudo apt install apt-transport-https ca-certificates curl
    ↪ software-properties-common gnupg lsb-release -y

# Add Docker's official GPG key
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o
    ↪ /usr/share/keyrings/docker-archive-keyring.gpg

# Set up the stable repository
echo "deb [arch=$(dpkg --print-architecture)
    ↪ signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
    ↪ https://download.docker.com/linux/debian $(lsb_release -cs) stable" | sudo
    ↪ tee /etc/apt/sources.list.d/docker.list > /dev/null

# Install Docker Engine
sudo apt update
sudo apt install docker-ce docker-ce-cli containerd.io docker-compose-plugin -y
```

```

# Add your user to the docker group
sudo usermod -aG docker ${USER}

# IMPORTANT: Log out and log back in for group changes to take effect.
# Alternatively, run 'newgrp docker' in your current terminal session.
```

Docker Installation

Verify the installation:

```
docker --version
```

Listing 2: Verify Docker

2.2 Pull and Run Oracle Database Free Docker Image &

Oracle provides official Docker images. We will use the Oracle Database 23c Free image.

```
docker pull container-registry.oracle.com/database/free
```

Listing 3: Pulling Oracle Database Free Docker Image

Run the container:

```
# Replace 'YourStrongPassword!' with a strong password.
# Specify a password to be used for database accounts. Oracle recommends that
# the password entered should be at least 8 characters in length, contain at
# least 1 uppercase character, 1 lower case character and 1 digit [0-9].
# Note that the same password will be used for SYS, SYSTEM and PDBADMIN accounts.

# For data persistence (recommended), create a local directory first:
mkdir -p /media/oracle_test_data
# Then run with the -v (persistent volume for dataset storage) option:
docker run -d --name oracledbai -p 1521:1521 \
  --ulimit nofile=1024:65536 \
  --ulimit nproc=2047:16384 \
  --ulimit stack=10485760:33554432 \
  --ulimit memlock=3221225472 \
  -e ORACLE_SID=FREE \
  -e ORACLE_PWD=verboTemlos_. \
  -e INIT_CPU_COUNT=2 \
  -e INIT_PROCESSES=300 \
  -e ENABLE_ARCHIVELOG=true \
  -e ENABLE_FORCE_LOGGING=true \
  -v /home/dc/oracle_db_data/free23ai:/media/oracle_test_data \
  container-registry.oracle.com/database/free
```

Run Oracle DB Container

Docker Run Parameters Explained

- `-d`: Detached mode (runs the container in the background).
- `-name oracledbai`: Assigns the name "oracledbai" to your container for easier management.
- `-p 1521:1521`: Maps port 1521 on your host machine to port 1521 inside the container. This is the default port for Oracle SQL*Net listener.
- `-ulimit <type>=<soft>:<hard>`: Sets resource limits (ulimits) for the container. These are often recommended for database stability and performance.
  - `nofile=1024:65536`: Maximum number of open file descriptors (soft limit: 1024, hard limit: 65536).

  - `nproc=2047:16384`: Maximum number of processes.
  - `stack=10485760:33554432`: Stack size in bytes (soft: 10MB, hard: 32MB).
  - `memlock=3221225472`: Maximum locked-in-memory address space in bytes (approx 3GB). Important for Oracle's System Global Area (SGA) if memory locking is used.
- `-e ORACLE_SID=FREE`: Sets the environment variable `ORACLE_SID` to "FREE". This defines the Oracle System Identifier (database name).
- `-e ORACLE_PWD=verboTemlos_.`: Sets the environment variable `ORACLE_PWD`. This password will be used for the SYS, SYSTEM, and PDBADMIN database accounts. **Important: Replace 'verboTemlos_.' in the command with your own strong, unique password.**
- `-e INIT_CPU_COUNT=2`: Sets the initial CPU count parameter for the database instance configuration.
- `-e INIT_PROCESSES=300`: Sets the initial processes parameter for the database instance configuration (e.g., `PROCESSES` initialization parameter).
- `-e ENABLE_ARCHIVELOG=true`: Enables archivelog mode for the database. This is crucial for point-in-time recovery and online backups.
- `-e ENABLE_FORCE_LOGGING=true`: Enables force logging mode. This ensures all database changes are logged, even for operations that might otherwise use `NOLOGGING`.
- `-v /home/dc/oracle_db_data/free23ai:/media/oracle_test_data`: Mounts a volume. This maps the host directory `/home/dc/oracle_db_data/free23ai` to the directory `/media/oracle_test_data` inside the container. This is used for persistent storage of database files, so your data survives container restarts or removal.
- `container-registry.oracle.com/database/free`: Specifies the Docker image to use for creating the container. In this case, it's the Oracle Database Free image from Oracle's container registry.

2.3 Check Container Status and Database Readiness

Then open Docker Desktop, go to the containers click on the *oracledbai* container, see the logs and wait for messages like *DATABASE IS READY TO USE!* or "Pluggable database FREEPDB1 opened read write". This may take several minutes. For programmatic logging.

```
docker ps # Lists running containers
docker logs -f oracleDBDebian # Follow container logs
```

Listing 5: Checking Container Status

2.4 Install Oracle SQL Developer on Debian

1. **Install JDK:**

```
sudo apt update
sudo apt install openjdk-17-jdk
```

Listing 6: Install OpenJDK 17

2. **Download SQL Developer:** If you already have VS Code, then just isntall the Extension called Oracle SQL Developer Extension for VSCode. Or navigate to the Oracle SQL Developer download page: https://www.oracle.com/database/sqldeveloper/technologies/download/. Download the "Other Platforms" (`.zip`) file.

3. **Extract and Run SQL Developer:**

```
# Example: If downloaded to ~/Downloads and extracting to ~/OracleTools
mkdir -p ~/OracleTools
unzip ~/Downloads/sqldeveloper-*.zip -d ~/OracleTools
cd ~/OracleTools/sqldeveloper
# On first run, it might prompt for the JDK path.
# Provide the path, e.g., /usr/lib/jvm/java-17-openjdk-amd64
./sqldeveloper.sh
```

SQL Developer on Debian

3 Windows Setup

For Windows, Oracle provides a native installer for its free database versions.

3.1 Download Oracle Database Free

1. Go to the Oracle Database download page (e.g., for 23c Free: https://www.oracle.com/database/free/get-started/).
2. Download the Windows installer (typically a `.zip` file).

3.2 Install Oracle Database Free

1. Extract the downloaded `.zip` file to a suitable location.
2. Run `setup.exe` as an Administrator.
3. Follow the on-screen prompts:
   - Accept the license agreement.
   - Choose an installation folder (e.g., `C:\Oracle\product\23c\dbhomeFREE`).
   - Set a strong password for `SYS`, `SYSTEM`, and `PDBADMIN` users when prompted. **Remember this password.**
   - The installer will perform checks and complete the installation. This can take some time.
   - Note the connection details provided at the end (Hostname: `localhost`, Port: `1521`, Default PDB Name: e.g., `FREEPDB1` or `XEPDB1`).

3.3 Download and Install Oracle SQL Developer

1. Visit the SQL Developer download page: https://www.oracle.com/database/sqldeveloper/technologies/download/.
2. Download the "Windows 64-bit with JDK included" for simplicity, or the version without JDK if you have a compatible JDK already installed and configured.
3. If you downloaded the version with JDK, extract the `.zip` file to a folder (e.g., `C:\SQLDeveloper`).
4. Run `sqldeveloper.exe` from the extracted folder.
5. If using the version without JDK, ensure your JDK is installed and `JAVA_HOME` is set. SQL Developer might prompt for the JDK path on first launch.

4 macOS Setup (via Docker)

Oracle does not offer native installers for recent Oracle Database versions on macOS, especially for Apple Silicon. Docker is the recommended method.

4.1 Install Docker Desktop for Mac

1. Download Docker Desktop for Mac from https://www.docker.com/products/docker-desktop.
2. Open the downloaded `.dmg` file and drag Docker.app to your Applications folder.
3. Launch Docker Desktop.

Run the container:

```
# Replace 'YourStrongPassword!' with a strong password.
# Specify a password to be used for database accounts. Oracle recommends that
# the password entered should be at least 8 characters in length, contain at
# least 1 uppercase character, 1 lower case character and 1 digit [0-9].
# Note that the same password will be used for SYS, SYSTEM and PDBADMIN accounts.

# For data persistence (recommended), create a local directory first:
mkdir -p /media/oracle_test_data
# Then run with the -v (persistent volume for dataset storage) option:
docker run -d --name oracledbai -p 1521:1521 \
  --ulimit nofile=1024:65536 \
  --ulimit nproc=2047:16384 \
  --ulimit stack=10485760:33554432 \
  --ulimit memlock=3221225472 \
  -e ORACLE_SID=FREE \
  -e ORACLE_PWD=verboTemlos_. \
  -e INIT_CPU_COUNT=2 \
  -e INIT_PROCESSES=300 \
  -e ENABLE_ARCHIVELOG=true \
  -e ENABLE_FORCE_LOGGING=true \
  -v /home/dc/oracle_db_data/free23ai:/media/oracle_test_data \
  container-registry.oracle.com/database/free
```

Run Oracle DB Container

Docker Run Parameters Explained

- `-d`: Detached mode (runs the container in the background).
- `-name oracledbai`: Assigns the name "oracledbai" to your container for easier management.
- `-p 1521:1521`: Maps port 1521 on your host machine to port 1521 inside the container. This is the default port for Oracle SQL*Net listener.
- `-ulimit <type>=<soft>:<hard>`: Sets resource limits (ulimits) for the container. These are often recommended for database stability and performance.
  - `nofile=1024:65536`: Maximum number of open file descriptors (soft limit: 1024, hard limit: 65536).
  - `nproc=2047:16384`: Maximum number of processes.
  - `stack=10485760:33554432`: Stack size in bytes (soft: 10MB, hard: 32MB).
  - `memlock=3221225472`: Maximum locked-in-memory address space in bytes (approx 3GB). Important for Oracle's System Global Area (SGA) if memory locking is used.
- `-e ORACLE_SID=FREE`: Sets the environment variable `ORACLE_SID` to "FREE". This defines the Oracle System Identifier (database name).

- `-e ORACLE_PWD=verboTemlos_.`: Sets the environment variable `ORACLE_PWD`. This password will be used for the SYS, SYSTEM, and PDBADMIN database accounts. **Important: Replace 'verboTemlos_.' in the command with your own strong, unique password.**
- `-e INIT_CPU_COUNT=2`: Sets the initial CPU count parameter for the database instance configuration.
- `-e INIT_PROCESSES=300`: Sets the initial processes parameter for the database instance configuration (e.g., `PROCESSES` initialization parameter).
- `-e ENABLE_ARCHIVELOG=true`: Enables archivelog mode for the database. This is crucial for point-in-time recovery and online backups.
- `-e ENABLE_FORCE_LOGGING=true`: Enables force logging mode. This ensures all database changes are logged, even for operations that might otherwise use `NOLOGGING`.
- `-v /home/dc/oracle_db_data/free23ai:/media/oracle_test_data`: Mounts a volume. This maps the host directory `/home/dc/oracle_db_data/free23ai` to the directory `/media/oracle_test_data` inside the container. This is used for persistent storage of database files, so your data survives container restarts or removal.
- `container-registry.oracle.com/database/free`: Specifies the Docker image to use for creating the container. In this case, it's the Oracle Database Free image from Oracle's container registry.

4.2 Check Container Status

```
docker ps
docker logs -f oracleDBMac
```

Listing 9: Checking Container Status on macOS

Wait for the database readiness messages.

4.3 Install Oracle SQL Developer on macOS

1. **Install JDK (if needed):** Download from Oracle or use Homebrew: `brew install openjdk@17`.
2. **Download SQL Developer:** From https://www.oracle.com/database/sqldeveloper/technologies/download/, get the macOS application (`.dmg` or `.app.zip`).
3. **Install:**
   - If `.dmg`, open it and drag `SQLDeveloper.app` to your `/Applications` folder.
   - If `.app.zip`, extract it and move `SQLDeveloper.app` to `/Applications`.
   - On first launch, you might need to right-click (or Control-click) and select "Open" due to Gatekeeper.
   - If prompted for a JDK path, provide it (e.g., `/Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home` or the path from Homebrew install like `/usr/local/opt/openjdk@17/libexec/openjdk.jdk/Contents/Home`).

5 Connecting with Oracle SQL Developer (All Systems)

Once your Oracle Database is running and SQL Developer is launched:

5.1 Create a New Database Connection

1. In SQL Developer's "Connections" panel (usually left side), click the green "+" icon or right-click "Connections" and select "New Connection...".

5.2 Configure Connection Details

Connection Parameters

- **Connection Name:** A descriptive name (e.g., `LocalOracle23cFree`).
- **Authentication Type:** `Default`.
- **Username:** `sys` (for initial admin access) or `system`.
- **Password:** The password set during database installation or Docker container creation.
- **Role:** `SYSDBA` (if connecting as `sys`), `Default` for `system` or other users.
- **Connection Type:** `Basic`.
- **Hostname:** `localhost` (for both native Windows and Docker setups due to port mapping).
- **Port:** `1521` (default).
- **Service Name (Recommended for PDBs):**
  - Oracle 23c Free (Docker/Native): `FREEPDB1` (default) or your custom PDB name.
  - Oracle XE (Windows): Typically `XEPDB1` or `ORCLPDB1`.
- **SID (Alternative, or for CDB root):**
  - Oracle 23c Free (CDB): `FREE`.
  - Oracle XE (CDB): `XE`.

Note: For development, connecting to a Pluggable Database (PDB) via its Service Name is preferred.

5.3 Test, Save, and Connect

1. Click **Test**. A "Status: Success" message indicates a correct configuration.
2. Click **Save** to store the connection.
3. Click **Connect**. A SQL Worksheet will open.

You are now ready to execute SQL and PL/SQL!

6 Initial User Setup (Recommended)

While `SYS` or `SYSTEM` can be used, creating a dedicated user for your coursework is good practice. Connect as `SYS` with the `SYSDBA` role and execute the following in a SQL Worksheet:

```
-- On newer Oracle versions, this might be needed to create users without C## prefix
ALTER SESSION SET "_ORACLE_SCRIPT"=true;

CREATE USER myCourseUser IDENTIFIED BY YourSecureUserPassword;

-- Basic connection and resource privileges
GRANT CONNECT, RESOURCE TO myCourseUser;
GRANT CREATE VIEW TO myCourseUser;

-- Tablespace quota (essential for creating tables)
GRANT UNLIMITED TABLESPACE TO myCourseUser;
-- For production, grant specific quotas on specific tablespaces.
```

```
13
14 -- Optionally, for more convenience in a learning environment:
15 -- GRANT CREATE SESSION, CREATE TABLE, CREATE SEQUENCE, CREATE PROCEDURE, CREATE
       ↪ TRIGGER TO myCourseUser;
16 -- Or, for full schema owner capabilities in a dev environment (use with
       ↪ understanding):
17 -- GRANT DBA TO myCourseUser;
18
19 /*
20 If connected to CDB$ROOT and creating a user in a specific PDB:
21 You might need to switch context if the user is not a common user.
22 ALTER SESSION SET CONTAINER = FREEPDB1; -- Or your PDB name
23 -- Then run CREATE USER and GRANT statements.
24 */
```

Create Course User

Replace `YourSecureUserPassword` with a strong password for this new user. Afterward, create a new SQL Developer connection for `myCourseUser`.

7 Troubleshooting Tips

Common issues and their resolutions:

Firewall: Ensure port `1521` is not blocked by your system's firewall.

- **Password Mismatch:** Double-check case-sensitivity and accuracy of the database password.
- **Listener Not Running/Reachable:**
  - Windows: Check Oracle Listener service in `services.msc`.
  - Docker: Listener is managed within the container. Ensure container is running and port is mapped.
  - Use `lsnrctl status` (from command line if Oracle client tools installed, or inside Docker container via `docker exec -it <containerName> lsnrctl status`).
- **Incorrect Service Name/SID:** This is a frequent issue. Verify the correct PDB service name.
  - As SYS: `SELECT name, pdb FROM v$services WHERE pdb IS NOT NULL;` or `SHOW PDBS;`
- **Docker Logs:** For Docker setups, `docker logs <containerName>` provides crucial startup and error information.
- **SQL Developer JDK Issues:** Ensure SQL Developer uses a compatible JDK. The path can often be set in `sqldeveloper.conf` (Linux/macOS) or `ide.conf`.
- **Oracle Documentation:** The official Oracle documentation is the definitive resource for installation and troubleshooting.

Happy Oracle SQL Querying!