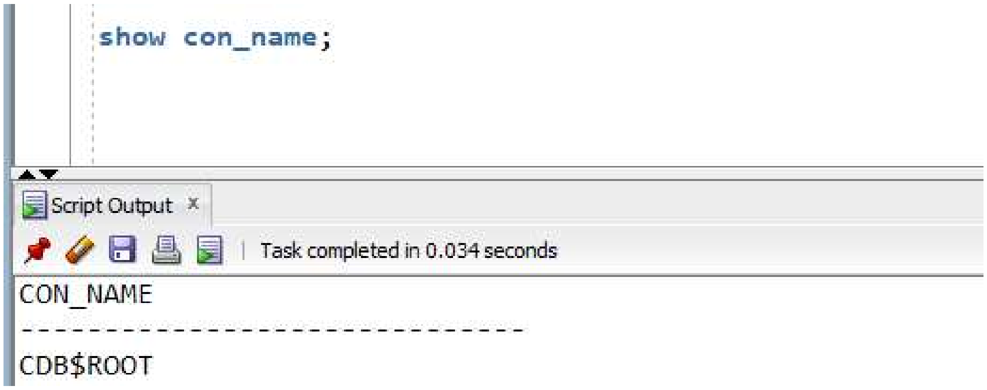
Basic commands and usage in Oracle Architecture

After, login to oracle instance. If we want to check, where we are below command can be used.

SHOW CON_NAME; → It will return the current container you are connected.



```
show con_name;
```

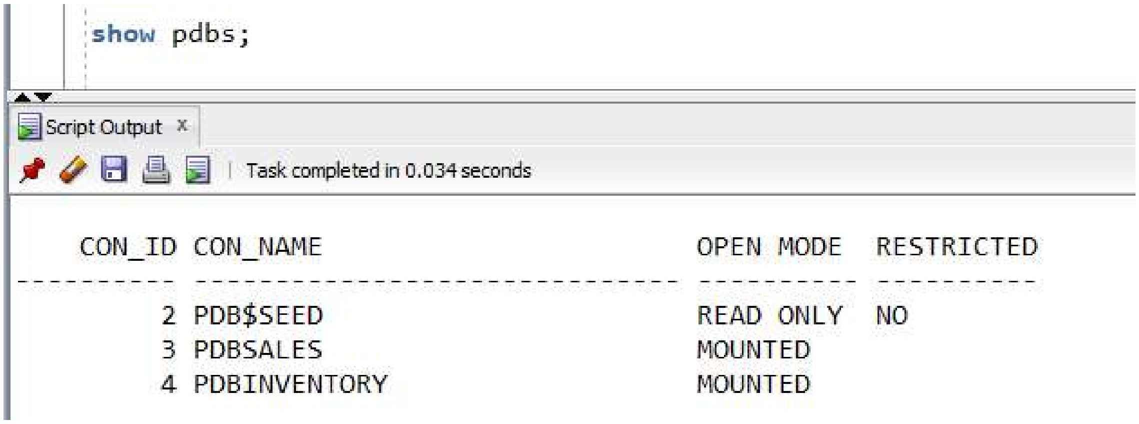
Task completed in 0.034 seconds

```
CON_NAME
------------------------------
CDB$ROOT
```

The Oracle architecture is like, container database is at top level and then pluggable databases will be created as per application and use.

If you want to list the existing pluggable databases in a container database, below command can be used.

SHOW PDBS; → It will list all the pluggable databases in a container database.



```
show pdbs;
```

Task completed in 0.034 seconds

```
    CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                       READ ONLY  NO
         3 PDBSALES                       MOUNTED
         4 PDBINVENTORY                   MOUNTED
```

From above output, we can see various pluggable databases and its state (OPEN MODE). Let's deep dive into database states.

1. MOUNTED
2. OPEN STATE

3. CLOSED
4. MIGRATE

1. MOUNTED State

- The PDB is attached to the **CDB (Container Database)** but is not accessible for queries or DML operations.

- Metadata for the PDB is available, but the actual data is not accessible.

✅ **Usage**: This state is often used for maintenance tasks like **backup, recovery**, or **database cloning**.

2. OPEN State

The PDB can be opened in different modes:

a) READ WRITE Mode

- The PDB is fully operational for both **read** and **write** activities.

✅ **Usage**: Normal operation for active users and applications.

b) READ ONLY Mode

- The PDB allows **SELECT** operations but prevents any **modifications** (INSERT, UPDATE, DELETE).

✅ **Usage**: Ideal for **reporting** or **auditing** tasks where you want to prevent changes.

c) RESTRICTED Mode

- Access is limited to **DBA users** with **RESTRICTED SESSION** privileges.

✅ **Usage**: Useful for **maintenance** tasks where you want to limit access to specific users.

3. CLOSE State

- The PDB is **offline** and cannot be accessed.

4.**MIGRATE State**

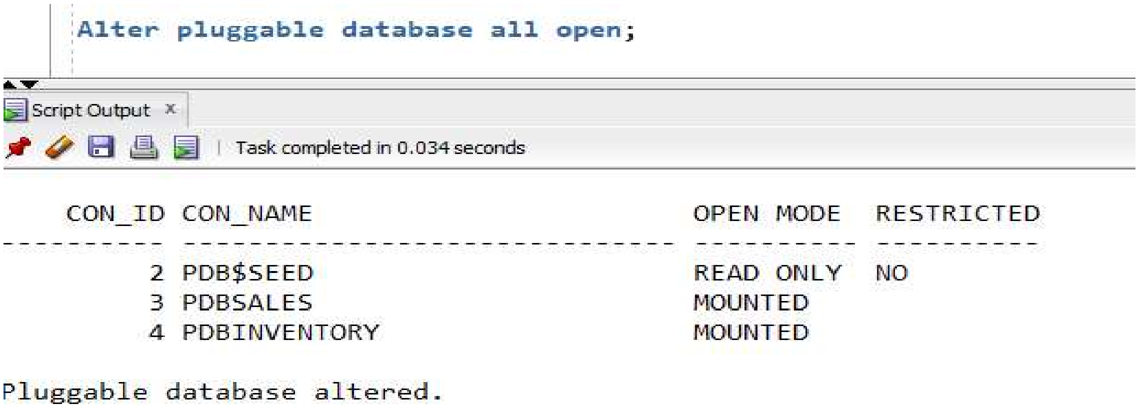
In **Oracle Pluggable Databases (PDBs)**, the **MIGRATE** state is a special state used during **PDB relocation or upgrade**. It is not a typical operational state like MOUNTED or OPEN but is specifically used when moving a **Pluggable Database** from one **Container Database (CDB)** to another or during **version upgrades**.

To open the database for DML operations, we need to use alter command as below.

ALTER PLUGGABLE DATABASE PDBSALES OPEN;

It will open the particular database to open state. If we want to open all existing databases, we can use all keyword.

ALTER PLUGGABLE DATABASE ALL OPEN;

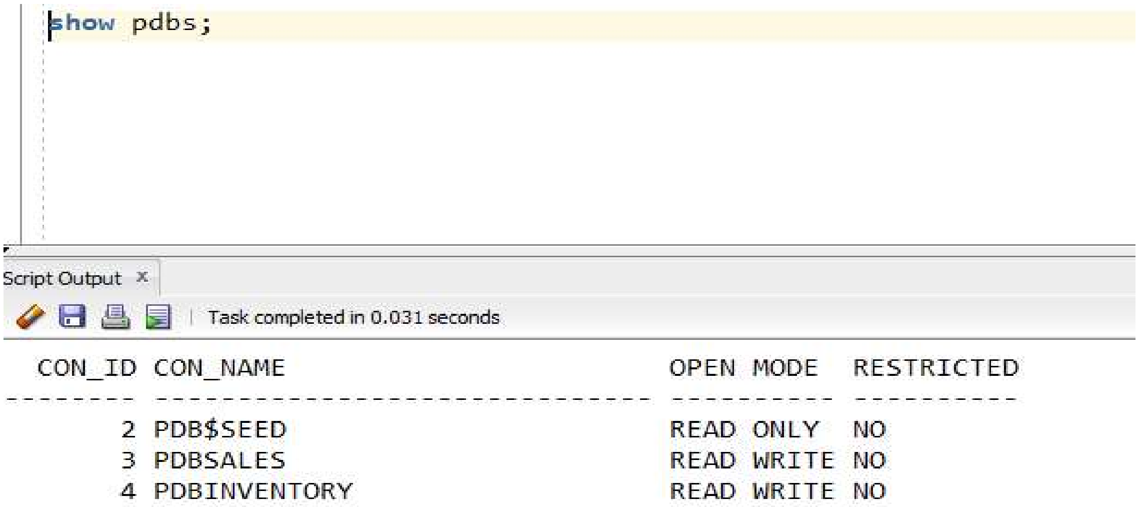


```
Alter pluggable database all open;
```

Task completed in 0.034 seconds

```
    CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                       READ ONLY  NO
         3 PDBSALES                       MOUNTED
         4 PDBINVENTORY                   MOUNTED

Pluggable database altered.
```

We can PDBs were altered and open to DML operations.

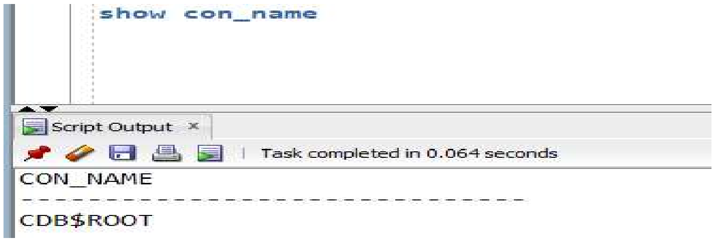


```
show pdbs;
```

Task completed in 0.031 seconds

```
  CON_ID CON_NAME                       OPEN MODE  RESTRICTED
-------- ------------------------------ ---------- ----------
       2 PDB$SEED                       READ ONLY  NO
       3 PDBSALES                       READ WRITE NO
       4 PDBINVENTORY                   READ WRITE NO
```

We can see current container is root container, but the main data insertion, updation will be happening on PDBs only. So to switch to other container, we need to alter the session.

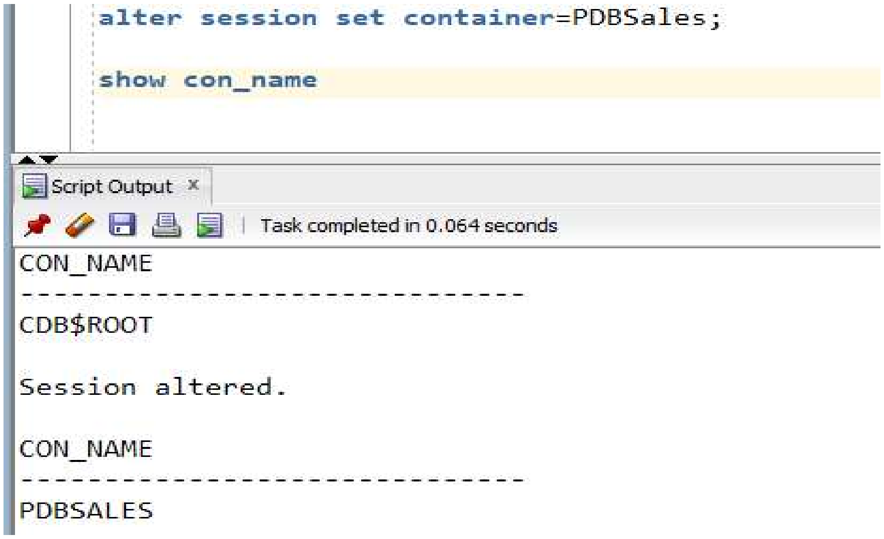


```
show con_name
```

Task completed in 0.064 seconds

```
CON_NAME
------------------------------
CDB$ROOT
```

ALTER SESSION SET CONTAINER=PDBName;



```
alter session set container=PDBSales;

show con_name
```

Task completed in 0.064 seconds

```
CON_NAME
------------------------------
CDB$ROOT

Session altered.

CON_NAME
------------------------------
PDBSALES
```

TABLE SPACE: It is a logical storage unit to store data, tables, indexes.

Oracle divides a database into one or more logical storage units called tablespaces.

Each tablespace consists of one or more files called datafiles. A datafile physically stores the data objects of the database such as tables and indexes on disk.

In other words, Oracle logically stores data in the tablespaces and physically stores data in datafiles associated with the corresponding tablespaces.

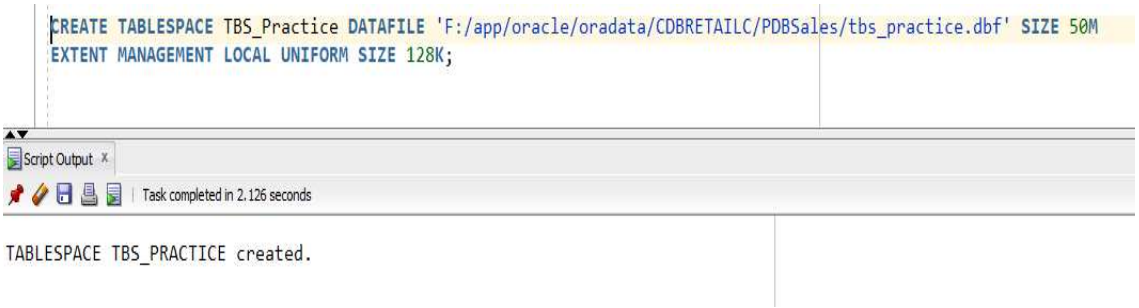
As per industry standards, Table spaces will be created as per application specific data to be stored. Below is the syntax to create a table space.

CREATE TABLESPAE 'NAME'

DATAFILE 'Path' SIZE 10M

EXTENT MANAGEMENT LOCAL

UNIFORM SIZE 128K;



```
CREATE TABLESPACE TBS_Practice DATAFILE 'F:/app/oracle/oradata/CDBRETAILC/PDBSales/tbs_practice.dbf' SIZE 50M
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K;
```

Task completed in 2.126 seconds

```
TABLESPACE TBS_PRACTICE created.
```

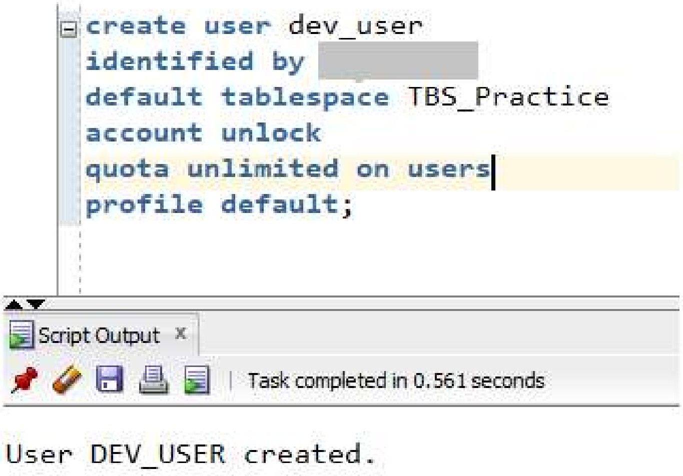
Next is to create a user for day-to-day operations with limited access on data to be used. Because we cannot provide sys permission to all users, it may damage data integrity, security.

While creating a user we need to specify the user name and password and also it requires a tablespace to be defined along with quota.

Quota means the amount of storage can this particular user can utilize on a data file.

create user username identified by password

default tablespace ‘ ‘ ;

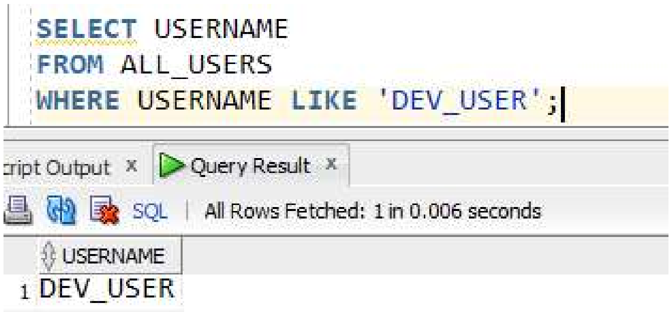


```
create user dev_user
identified by [redacted]
default tablespace TBS_Practice
account unlock
quota unlimited on users
profile default;
```

Task completed in 0.561 seconds

```
User DEV_USER created.
```

We can view the created user by retrieving from Users table.



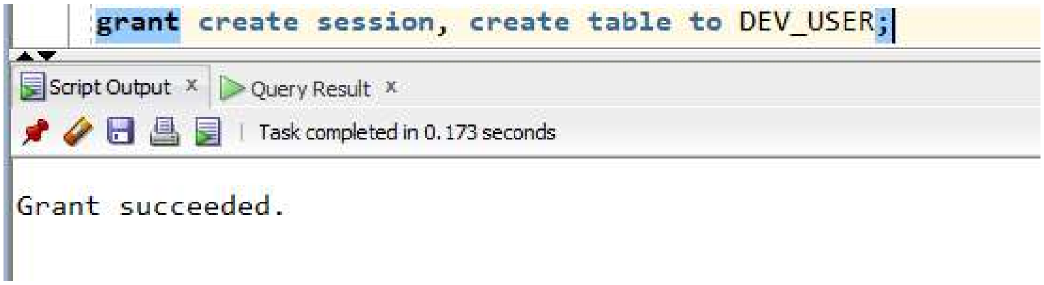
```
SELECT USERNAME
FROM ALL_USERS
WHERE USERNAME LIKE 'DEV_USER';
```

All Rows Fetched: 1 in 0.006 seconds

	USERNAME
1	DEV_USER

After user creation we need to provide the permissions to user with sys user. Then only the newly created user can perform required operations.

Below we are granting create session means it can used to connect to DB and create Table means user is able to create Table.



```
grant create session, create table to DEV_USER;
```

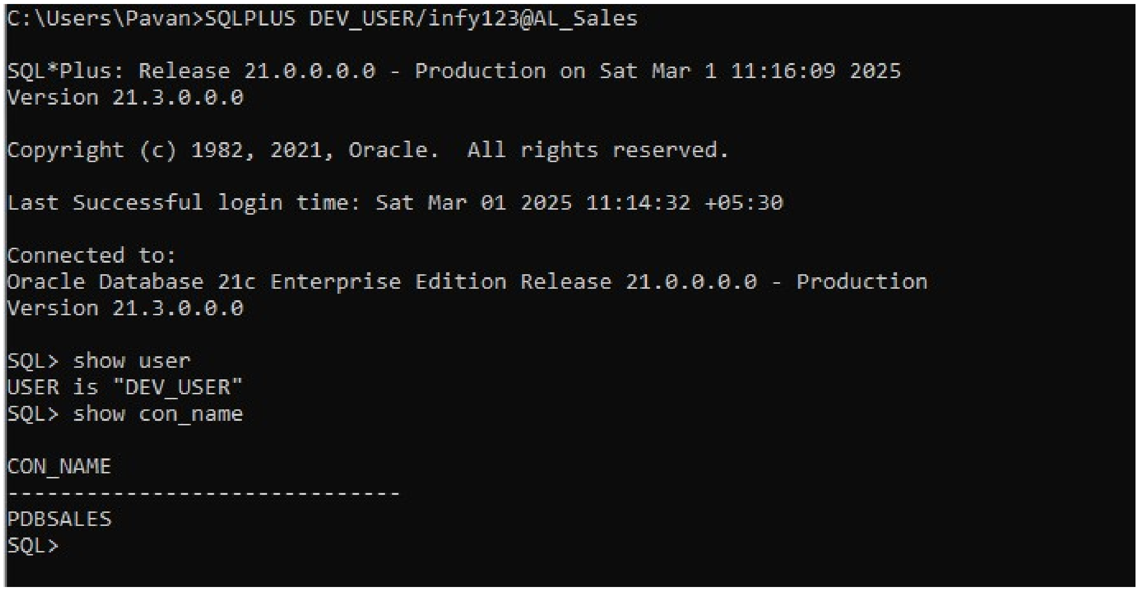
Task completed in 0.173 seconds

```
Grant succeeded.
```

Testing user:

Below command can verify, we are successfully connected PDBSales (AL_Sales – It is a alias name for PDB. It is used to connect to specified DB with short name, called as TNS Name. We need to create a Listener and TNS Names for this service to be used.)

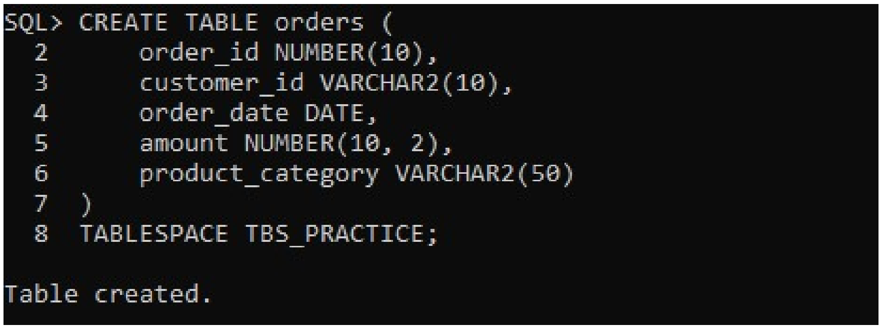
SQLPLUS USERNAME/PASSWORD@AL_DB;



```
C:\Users\Pavan>SQLPLUS DEV_USER/infy123@AL_Sales

SQL*Plus: Release 21.0.0.0.0 - Production on Sat Mar 1 11:16:09 2025
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle.  All rights reserved.

Last Successful login time: Sat Mar 01 2025 11:14:32 +05:30

Connected to:
Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> show user
USER is "DEV_USER"
SQL> show con_name

CON_NAME
------------------------------
PDBSALES
SQL>
```

Create a Table orders:

As we had given permission to create Table, we can able to create a Table as below.



```
SQL> CREATE TABLE orders (
  2      order_id NUMBER(10),
  3      customer_id VARCHAR2(10),
  4      order_date DATE,
  5      amount NUMBER(10, 2),
  6      product_category VARCHAR2(50)
  7  )
  8  TABLESPACE TBS_PRACTICE;

Table created.
```