

Steps to Open PDBs and Verify Connectivity

1. Connect as SYSDBA

```
sqlplus / as sysdba
```

Expected Response:

```
SQL*Plus: Release 19.0.0.0.0 - Production on <date>
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
```

2. Open PDB1

```
-- Set the session to the CDB root
ALTER SESSION SET CONTAINER = CDB$ROOT;

-- List all PDBs to see their status
SELECT name, open_mode FROM v$pdb;
```

Expected Response:

NAME	OPEN_MODE
-----	-----
PDB\$SEED	READ ONLY
PDB1	MOUNTED
PDB2	MOUNTED

```
-- Set container to PDB1
ALTER SESSION SET CONTAINER = pdb1;

-- Open PDB1
ALTER PLUGGABLE DATABASE pdb1 OPEN;

-- Verify the PDB open mode
SELECT name, open_mode FROM v$pdb WHERE name = 'PDB1';
```

Expected Response:

NAME	OPEN_MODE
-----	-----
PDB1	READ WRITE

3. Open PDB2

```
-- Set the session to the CDB root
ALTER SESSION SET CONTAINER = CDB$ROOT;
```

```
-- Set container to PDB2
ALTER SESSION SET CONTAINER = pdb2;

-- Open PDB2
ALTER PLUGGABLE DATABASE pdb2 OPEN;

-- Verify the PDB open mode
SELECT name, open_mode FROM v$pdb WHERE name = 'PDB2';
```

Expected Response:

NAME	OPEN_MODE
-----	-----
PDB2	READ WRITE

4. Verify the PDBADMIN User and Grant Privileges for PDB1 and PDB2

For PDB1:

```
-- Set container to PDB1
ALTER SESSION SET CONTAINER = pdb1;

-- Verify the PDBADMIN user
SELECT username FROM dba_users WHERE username = 'PDBADMIN';
```

Expected Response:

USERNAME

PDBADMIN

(If no rows are selected, create the user)

```
-- If needed, create the user
CREATE USER pdbadmin IDENTIFIED BY fenago;

-- Grant necessary privileges
GRANT CONNECT, DBA TO pdbadmin;
```

Expected Response:

```
User created.

Grant succeeded.
```

For PDB2:

```
-- Set container to PDB2
ALTER SESSION SET CONTAINER = pdb2;

-- Verify the PDBADMIN user
SELECT username FROM dba_users WHERE username = 'PDBADMIN';
```

Expected Response:

```
USERNAME
-----
PDBADMIN
```

(If no rows are selected, create the user)

```
-- If needed, create the user
CREATE USER pdbadmin IDENTIFIED BY fenago;

-- Grant necessary privileges
GRANT CONNECT, DBA TO pdbadmin;
```

Expected Response:

```
User created.

Grant succeeded.
```

5. Test Connection from Command Line

For PDB1:

```
sqlplus pdbadmin/fenago@localhost:1521/pdb1
```

Expected Response:

```
SQL*Plus: Release 19.0.0.0.0 - Production on <date>
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
```

For PDB2:

```
sqlplus pdbadmin/fenago@localhost:1521/pdb2
```

Expected Response:

```
SQL*Plus: Release 19.0.0.0.0 - Production on <date>
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
```

By following these steps and verifying the expected responses, you can ensure that the PDBs `pdb1` and `pdb2` are open and that the `pdbadmin` user exists with the necessary privileges. Once these steps are successfully executed and verified from the command line, you should be able to connect using SQL Developer as well.

Let's go through the steps to open `pdb1` and `pdb2` and ensure they are correctly configured. Additionally, I will show you how to manually set the environment variables.

Manually Setting Environment Variables

First, set the environment variables. Ensure you replace `<your_values>` with the appropriate values for your environment.

```
xhost +
su - oracle
```

then set the environment

```
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=/u01/app/oracle/product/19.3.0/dbhome_1
export ORACLE_SID=CDBDEV
export PATH=$ORACLE_HOME/bin:$PATH
```

Steps to Open `pdb1` and `pdb2` and Run `catalog.sql` and `catproc.sql`

1. Connect to SQL*Plus as SYSDBA:

```
sqlplus / as sysdba
```

2. Ensure you are in the root container (CDB):

```
SHOW CON_NAME;
```

If it does not show `CDB$ROOT`, switch to the root container:

```
ALTER SESSION SET CONTAINER = CDB$ROOT;
```

3. Open the PDBs:

```
ALTER PLUGGABLE DATABASE pdb1 OPEN;
ALTER PLUGGABLE DATABASE pdb2 OPEN;
```

4. Switch to the PDB and run `catalog.sql` and `catproc.sql` for each PDB:

◦ For `pdb1`:

```
ALTER SESSION SET CONTAINER = pdb1;
@$ORACLE_HOME/rdbms/admin/catalog.sql;
@$ORACLE_HOME/rdbms/admin/catproc.sql;
```

◦ For `pdb2`:

```
ALTER SESSION SET CONTAINER = pdb2;
@$ORACLE_HOME/rdbms/admin/catalog.sql;
@$ORACLE_HOME/rdbms/admin/catproc.sql;
```

5. Exit SQL*Plus:

```
EXIT;
```

Full Command Sequence

Here is the full sequence of commands:

Set Environment Variables

```
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=/u01/app/oracle/product/19.3.0/dbhome_1
export ORACLE_SID=CDBDEV
export PATH=$ORACLE_HOME/bin:$PATH
```

Open and Configure PDBs

```
sqlplus / as sysdba
```

Ensure you are in the root container

```
SHOW CON_NAME;

-- If not in CDB$ROOT
ALTER SESSION SET CONTAINER = CDB$ROOT;
```

Open the PDBs

```
ALTER PLUGGABLE DATABASE pdb1 OPEN;
ALTER PLUGGABLE DATABASE pdb2 OPEN;
```

Switch to `pdb1` and run the scripts

```
ALTER SESSION SET CONTAINER = pdb1;
@$ORACLE_HOME/rdbms/admin/catalog.sql;
@$ORACLE_HOME/rdbms/admin/catproc.sql;
```

Switch to `pdb2` and run the scripts

```
ALTER SESSION SET CONTAINER = pdb2;
@$ORACLE_HOME/rdbms/admin/catalog.sql;
@$ORACLE_HOME/rdbms/admin/catproc.sql;
```

Exit SQL*Plus

```
EXIT;
```

Verify the PDBs

After following the above steps, you should verify that `pdb1` and `pdb2` are open and properly configured.

Check the Status of the PDBs

Reconnect to SQL*Plus:

```
sqlplus / as sysdba
```

Verify the PDBs are open

```
SHOW PDBS;
```

Test the PDB Connections

Finally, use SQL*Plus to connect to `pdb1` and `pdb2` to ensure they are accessible.

```
sqlplus pdbadmin/fenago@localhost:1521/pdb1  
sqlplus pdbadmin/fenago@localhost:1521/pdb2
```

By following these steps, you should be able to open and configure `pdb1` and `pdb2`, and verify their status and connectivity.