

Lab 6.6: Creating, Locking, and Unlocking Users in PDBLAB1 and PDBLAB2

Objective:

To demonstrate how to create, lock, and unlock a user account in PDBLAB1 and PDBLAB2.

Steps:

1. Connect to SQL*Plus as SYSDBA:

Explanation: This step establishes a connection to the database with SYSDBA privileges, which is required to perform administrative tasks such as creating, locking, and unlocking user accounts.

```
sqlplus / as sysdba
```

2. Create a User Account in PDBLAB1:

◦ Switch to PDBLAB1:

Explanation: This command switches the context to the PDBLAB1 container, allowing you to manage users specific to this PDB.

```
ALTER SESSION SET CONTAINER = PDBLAB1;
```

◦ Create the User Account PDB1_USER :

Explanation: This command creates a user account PDB1_USER in PDBLAB1.

```
CREATE USER PDB1_USER IDENTIFIED BY password;
```

◦ Grant Privileges to PDB1_USER :

Explanation: This command grants necessary privileges to PDB1_USER .

```
GRANT CONNECT, RESOURCE TO PDB1_USER;
```

◦ Verify the User Account Creation in PDBLAB1:

Explanation: This query checks if PDB1_USER is created successfully in PDBLAB1.

```
SELECT username FROM dba_users WHERE username = 'PDB1_USER';
```

3. Lock the User Account in PDBLAB1:

◦ Switch to PDBLAB1 (if not already done):

```
ALTER SESSION SET CONTAINER = PDBLAB1;
```

◦ Lock the User Account PDB1_USER :

Explanation: This command locks the user account PDB1_USER , preventing the user from logging into the database.

```
ALTER USER PDB1_USER ACCOUNT LOCK;
```

◦ Verify the User Account is Locked in PDBLAB1:

Explanation: This query checks the account status of `PDB1_USER` to confirm it is locked.

```
SELECT username, account_status FROM dba_users WHERE username =  
'PDB1_USER';
```

4. Unlock the User Account in PDBLAB1:

- **Switch to PDBLAB1 (if not already done):**

```
ALTER SESSION SET CONTAINER = PDBLAB1;
```

- **Unlock the User Account `PDB1_USER` :**

Explanation: This command unlocks the user account `PDB1_USER`, allowing the user to log into the database again.

```
ALTER USER PDB1_USER ACCOUNT UNLOCK;
```

- **Verify the User Account is Unlocked in PDBLAB1:**

Explanation: This query checks the account status of `PDB1_USER` to confirm it is unlocked.

```
SELECT username, account_status FROM dba_users WHERE username =  
'PDB1_USER';
```

5. Create, Lock, and Unlock a User Account in PDBLAB2:

Explanation: Repeat the above steps for PDBLAB2 to create, lock, and unlock `PDB2_USER`.

- **Switch to PDBLAB2:**

```
ALTER SESSION SET CONTAINER = PDBLAB2;
```

- **Create the User Account `PDB2_USER` :**

```
CREATE USER PDB2_USER IDENTIFIED BY password;
```

- **Grant Privileges to `PDB2_USER` :**

```
GRANT CONNECT, RESOURCE TO PDB2_USER;
```

- **Verify the User Account Creation in PDBLAB2:**

```
SELECT username FROM dba_users WHERE username = 'PDB2_USER';
```

- **Lock the User Account `PDB2_USER` :**

```
ALTER USER PDB2_USER ACCOUNT LOCK;
```

- **Verify the User Account is Locked in PDBLAB2:**

```
SELECT username, account_status FROM dba_users WHERE username =  
'PDB2_USER';
```

- **Unlock the User Account PDB2_USER :**

```
ALTER USER PDB2_USER ACCOUNT UNLOCK;
```

- **Verify the User Account is Unlocked in PDBLAB2:**

```
SELECT username, account_status FROM dba_users WHERE username =  
'PDB2_USER';
```

Detailed Example:

1. **Connect to SQL*Plus as SYSDBA:**

```
sqlplus / as sysdba
```

2. **Create a User Account in PDBLAB1:**

- **Switch to PDBLAB1:**

```
ALTER SESSION SET CONTAINER = PDBLAB1;
```

- **Create the User Account PDB1_USER :**

```
CREATE USER PDB1_USER IDENTIFIED BY password;
```

- **Grant Privileges to PDB1_USER :**

```
GRANT CONNECT, RESOURCE TO PDB1_USER;
```

- **Verify the User Account Creation in PDBLAB1:**

```
SELECT username FROM dba_users WHERE username = 'PDB1_USER';
```

3. **Lock the User Account in PDBLAB1:**

- **Switch to PDBLAB1 (if not already done):**

```
ALTER SESSION SET CONTAINER = PDBLAB1;
```

- **Lock the User Account PDB1_USER :**

```
ALTER USER PDB1_USER ACCOUNT LOCK;
```

- **Verify the User Account is Locked in PDBLAB1:**

```
SELECT username, account_status FROM dba_users WHERE username =  
'PDB1_USER';
```

4. **Unlock the User Account in PDBLAB1:**

- **Unlock the User Account PDB1_USER :**

```
ALTER USER PDB1_USER ACCOUNT UNLOCK;
```

- **Verify the User Account is Unlocked in PDBLAB1:**

```
SELECT username, account_status FROM dba_users WHERE username =
'PDB1_USER';
```

5. Create, Lock, and Unlock a User Account in PDBLAB2:

- **Switch to PDBLAB2:**

```
ALTER SESSION SET CONTAINER = PDBLAB2;
```

- **Create the User Account PDB2_USER :**

```
CREATE USER PDB2_USER IDENTIFIED BY password;
```

- **Grant Privileges to PDB2_USER :**

```
GRANT CONNECT, RESOURCE TO PDB2_USER;
```

- **Verify the User Account Creation in PDBLAB2:**

```
SELECT username FROM dba_users WHERE username = 'PDB2_USER';
```

- **Lock the User Account PDB2_USER :**

```
ALTER USER PDB2_USER ACCOUNT LOCK;
```

- **Verify the User Account is Locked in PDBLAB2:**

```
SELECT username, account_status FROM dba_users WHERE username =
'PDB2_USER';
```

- **Unlock the User Account PDB2_USER :**

```
ALTER USER PDB2_USER ACCOUNT UNLOCK;
```

- **Verify the User Account is Unlocked in PDBLAB2:**

```
SELECT username, account_status FROM dba_users WHERE username =
'PDB2_USER';
```

Explanation:

- **Connecting to SQL*Plus as SYSDBA:** Provides the necessary privileges to perform administrative tasks.
- **Creating Users:** The `CREATE USER` command is used to create new users in the PDBs.
- **Granting Privileges:** The `GRANT CONNECT, RESOURCE` command allows the user to connect to the database and create/manage database objects.
- **Locking/Unlocking User Accounts:** The `ALTER USER ... ACCOUNT LOCK` and `ALTER USER ... ACCOUNT UNLOCK` commands are used to control user access to the database.
- **Verifying Account Status:** Queries are used to ensure the accounts are created, locked, and unlocked as expected.

By following these steps, you will create, lock, and unlock user accounts in PDBLAB1 and PDBLAB2, ensuring that the users' access to the database is correctly managed.