Practice

Using Database Links

Practice Target

In this practice you will create and use a database link to the `SOE` schema in `PDB1`

Practice Assumptions

This practice assumes the following:

- The virtual machine `srv1` up and running with its **CDB** database
- The virtual machine `winsrv` up and running

Creating and Using a Database Link to the SOE Schema in PDB1 in srv1

In this section of the practice, you will create a database link in the database in winsrv that points to the SOE schema in PDB1 in the srv1.

In the following steps, you will configure a connection in WINSRV to PDB1 in srv1.

1. In winsrv, start Notepad then use it to open the tnsnames.ora file located in D:\oracle\product\19.0.0\db_1\network\admin.

2. Add the following network service name to it and save the changes on the file.

```
soesrv1 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = srv1)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = pdb1.localdomain)
    )
  )
```

3. Test connecting to the added connection name. The connection should succeed.

```
sqlplus soe/ABcd##1234@soesrv1
```

4. Open a Putty session to srv1 as oracle

5. In srv1, login to PDB1 as SYS and check the GLOBAL_NAMES value.

We configured pdb1 in tnsnames.ora file in the previous practice.

In the practice environment database, GLOBAL_NAMES is disabled (FALSE). This means we can create database links with any name.

However, in this practice, we will create a database link with global database name of pdb1 in srv1.

```
sqlplus sys/ABcd##1234@pdb1 as sysdba
show parameter GLOBAL_NAMES
```

6. Retrieve the global name assigned to pdb1

The global name will be used to create the database link in winsrv.

```
SELECT * FROM GLOBAL_NAME;
```

7. In `winsrv` database, run the following code to create a user and grant basic privileges to it.

Observe that the user does not have any privileges to create a table and it does not have quota on any user tablespace.

```
conn / as sysdba
CREATE USER USER1 IDENTIFIED BY ABcd##1234 DEFAULT TABLESPACE USERS ;
GRANT CREATE SESSION, CREATE DATABASE LINK, CREATE SYNONYM TO USER1;
```

8. Login as `USER1` and run the following code to create a database link.

This is a fixed-user database link that points to `SOE` in `srv1`.

```
conn USER1/ABcd##1234

CREATE DATABASE LINK PDB1.LOCALDOMAIN
 CONNECT TO soe IDENTIFIED BY ABcd##1234 USING 'soesrv1';
```

9. Test the database link is working by querying any table in `srv1`.

Using database link, `USER1` is able to access `SOE` objects in a remote database.

```
SELECT COUNT(*) FROM ORDERS@PDB1.LOCALDOMAIN;
```

In real life scenario, referring to objects in remote databases using the format `OBJECT_NAME@DB_LINK_NAME` is tedious for a schema with so many objects. A common way to tackle this challenge is by using synonyms.

In the following steps, you will produce code to create synonyms for `soe` tables and execute them in `winsrv`.

10. In `srv1`, run the following code to produce code to create synonyms for `SOE` tables.

It is a common practice to generate code using `SELECT` statement this way.

```
conn soe/ABcd##1234@pdb1

SELECT 'CREATE SYNONYM ' || TABLE_NAME || ' FOR ' || TABLE_NAME ||
'@PDB1.LOCALDOMAIN;' CODE FROM USER_TABLES;
```

11. In `srv1` Putty window, highlight the produced code to copy it automatically into the clipboard.

12. In `winsrv`, paste the code into `USER1` command line prompt to execute it.

```
CONN USER1/ABcd##1234

CREATE SYNONYM CUSTOMERS FOR CUSTOMERS@PDB1.LOCALDOMAIN;
CREATE SYNONYM ADDRESSES FOR ADDRESSES@PDB1.LOCALDOMAIN;
CREATE SYNONYM CARD_DETAILS FOR CARD_DETAILS@PDB1.LOCALDOMAIN;
CREATE SYNONYM WAREHOUSES FOR WAREHOUSES@PDB1.LOCALDOMAIN;
CREATE SYNONYM ORDER_ITEMS FOR ORDER_ITEMS@PDB1.LOCALDOMAIN;
CREATE SYNONYM ORDERS FOR ORDERS@PDB1.LOCALDOMAIN;
CREATE SYNONYM INVENTORIES FOR INVENTORIES@PDB1.LOCALDOMAIN;
CREATE SYNONYM PRODUCT_INFORMATION FOR PRODUCT_INFORMATION@PDB1.LOCALDOMAIN;
CREATE SYNONYM LOGON FOR LOGON@PDB1.LOCALDOMAIN;
CREATE SYNONYM PRODUCT_DESCRIPTIONS FOR PRODUCT_DESCRIPTIONS@PDB1.LOCALDOMAIN;
CREATE SYNONYM ORDERENTRY_METADATA FOR ORDERENTRY_METADATA@PDB1.LOCALDOMAIN;
```

13. Test accessing `ORDERS` table in the remote database via its synonym.

The data is retrieved by querying the synonym.

```
SELECT COUNT(*) FROM ORDERS;
```

Extra Exercise

Consider making a database link from `srv1` to the database in `winsrv`.

Cleanup

14. In `winsrv`, drop `USER1`

```
conn / as sysdba
DROP USER USER1 CASCADE;
```

15. Shutdown `winsrv` and take a snapshot of it in VirtualBox. Delete old snapshots (if there is any).

Summary

Database links allow a local database user to access objects in a remote database.