Practice

Using External Tables

Practice Target

In this practice, you will create external tables with `ORACLE_LOADER` and `ORACLE_DATAPUMP` access drivers.

Practice Overview

In this practice, you will perform the following tasks:

- Create an external table with `ORACLE_LOADER` access driver.
- Create an external table with `ORACLE_DATAPUMP` access driver.

Assumptions

- This practice assumes that `srv1` is up and running from the **CDB** snapshot.
- This practice assumes that `winsrv` is up and running.

A. Create an External Table with ORACLE_LOADER Access Driver

In this section of the practice, you will generate CSV file from PDB1 in srv1, transfer it to winsrv, then create an external table in the database in winsrv linked to the CSV file.

1. Start Putty to srv1 as oracle, then in the staging folder, create a subdirectory named as 'ext'.

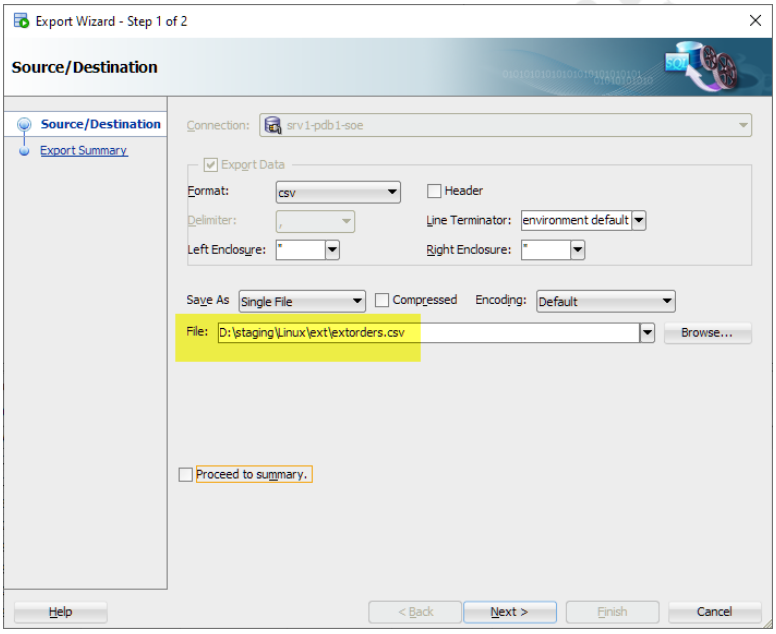
```
mkdir /media/sf_staging/ext
```

2. Start SQL Developer and login to PDB1 as SOE.

3. Submit the following query to retrieve list of the records to be saved in an external file. It returns 100 order records.

```
SELECT ORDER_ID, TO_CHAR(ORDER_DATE,'DD-MM-RRRR HH24:MI:SS') ORDER_DATE, CUSTOMER_ID,
ORDER_STATUS, DELIVERY_TYPE, ORDER_TOTAL
FROM ORDERS FETCH FIRST 100 ROWS ONLY;
```

4. Export the output of the query into a CSV file named as 'extorders.csv' without headers. Save the CSV file in the ext sub-directory.



5. Copy the CSV file to the shared folder configured in winsrv

6. In winsrv, login to the VirtualBox window as oracle.

7. Verify that the CSV file is seen in the mapped Z drive.

8. Copy the CSV file from the Z drive to D:\temp

9. In winsrv, open a command line window, login to orawindb as SYS and create a directory object that points to D:\temp in the shared folder. Grant access on the directory to HR.

```
sqlplus sys/ABcd##1234@orawindb as sysdba
CREATE OR REPLACE DIRECTORY EXTDIR AS 'D:\temp';
GRANT READ, WRITE ON DIRECTORY EXTDIR TO HR;
```

10. Login to the database as HR and create an external table linked to the CSV file.

    The file is a text file, so we have only the option to use ORACLE_LOADER access driver.

```
conn hr/ABcd##1234

CREATE TABLE ext_orders
 (ORDER_ID       NUMBER(12),
  ORDER_DATE     DATE,
  CUSTOMER_ID    NUMBER(12),
  ORDER_STATUS   NUMBER(2),
  DELIVERY_TYPE  VARCHAR2(15),
  ORDER_TOTAL    NUMBER(8,2)
 )
ORGANIZATION EXTERNAL
 ( TYPE ORACLE_LOADER
   DEFAULT DIRECTORY extdir
   ACCESS PARAMETERS
   ( RECORDS DELIMITED BY NEWLINE
     BADFILE EXTDIR:'extorders.bad'
     LOGFILE EXTDIR:'extorders.log'
     FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
     MISSING FIELD VALUES ARE NULL
     ( ORDER_ID,
       ORDER_DATE DATE MASK 'DD-MM-YYYY HH24:MI:SS',
       CUSTOMER_ID,
       ORDER_STATUS,
       DELIVERY_TYPE,
       ORDER_TOTAL
      )
   )
   LOCATION ('extorders.csv')
  )
REJECT LIMIT UNLIMITED;
```

11. Retrieve the external table attributes from the data dictionary view USER_EXTERNAL_TABLES.

```
SELECT TYPE_NAME, ACCESS_TYPE FROM USER_EXTERNAL_TABLES WHERE TABLE_NAME='EXT_ORDERS';
```

12. Display the structure of the external table.

    It has the same structure as the structure defined in the CREATE TABLE statement.

```
DESC EXT_ORDERS
```

13. Retrieve data from the external file.

    Observe that the date column is displayed in the session `NLS_DATE_FORMAT` setting, not in the format saved in the external file. This proves that Oracle database engine recognized the datatype of the column as a `DATE` data type. Furthermore, this means we can apply the Date/Time functions on the `DATE` columns in the external tables.

```
SELECT * FROM ext_orders;
```

14. Check out the files created in the `ext` sub-directory

    Observe that the `SQL_LOADER` driver considers all the columns (except `ORDER_DATE`) as of data type `CHAR(255)`. The database internally performs the conversion from this data type to the data type of the column as defined in the table definition.

15. Open the CSV file and make a corrupted data in one of its records.

16. Query the external table. It should return 99 records.

```
SELECT * FROM ext_orders;
```

17. Check out the files generated in the `ext` directory.

    You should see a bad file generated containing the rejected row. The file contains the rejected row but it does not tell why it was rejected. Having a look at the log file helps on detecting the root cause.

Clean up

18. Drop the external table.

```
DROP TABLE EXT_ORDERS ;
```

19. Delete the files (not the directories) in the `ext` directory and `D:\temp`

B. Create an External Table with ORACLE_DATAPUMP Access Driver

In this section of the practice, you will create an external table in srv1 to unload a query results into a dump file. Then, you will move the dump file and link it to an external table in winsrv.

20. In srv1, invoke SQL*Plus and login to the database as SYS. Then create a directory object that is linked to the ext sub-directory and grant access on it to SOE.

```
sqlplus / as sysdba
ALTER SESSION SET CONTAINER=PDB1;
CREATE DIRECTORY EXTDIR AS '/media/sf_staging/ext';
GRANT READ, WRITE ON DIRECTORY EXTDIR TO SOE;
```

21. As SOE, run the following statement to **unload** the query below into the ext_orders table using ORACLE_DATAPUMP driver.

Observe that we do not need to define the column datatypes. They are automatically defined by the database from the query.

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

CREATE TABLE ext_orders
 (ORDER_ID       ,
  ORDER_DATE     ,
  CUSTOMER_ID    ,
  ORDER_STATUS   ,
  DELIVERY_TYPE  ,
  ORDER_TOTAL
 )
ORGANIZATION EXTERNAL
  (
    TYPE ORACLE_DATAPUMP
    DEFAULT DIRECTORY extdir
    LOCATION ('orders.dmp')
  )
AS
SELECT ORDER_ID, ORDER_DATE, CUSTOMER_ID, ORDER_STATUS, DELIVERY_TYPE, ORDER_TOTAL
FROM ORDERS FETCH FIRST 100 ROWS ONLY;
```

22. Check out the content of the ext directory.

You should see a dump file and a log file.

```
host ls -al /media/sf_staging/ext
```

23. Check out the contents of the log file to verify no issue was raised when unloading the dump file.

```
host cat /media/sf_staging/ext/EXT_ORDERS_***.log
```

24. Display the structure of EXT_ORDERS.

```
desc EXT_ORDERS
```

25. Query the `EXT_ORDERS` table.

    The query retrieves its data from the dump file. It does not run the query (used to unload the dump file) anymore.

```
set linesize 150
SELECT ORDER_ID, TO_CHAR(ORDER_DATE,'DD-MON-YY') ORDER_DATE, CUSTOMER_ID, ORDER_STATUS,
DELIVERY_TYPE, ORDER_TOTAL
FROM EXT_ORDERS ;
```

26. Try deleting or updating the `EXT_ORDERS`.

    Once an external table is created, then no data may be added, updated or deleted from the external table.

```
DELETE EXT_ORDERS;
```

27. Copy the generated dump file to the shared folder configured in `winsrv`.

28. In `winsrv`, copy the dump file from z drive to `D:\temp`

29. In `winsrv`, login to `orawindb` as `HR` and create an external table linked to dump file.

    Observe in the statement that we need to define the column data types. This means, having the dump file alone is not enough to be able to load it to an external table. We need to know the table structure specifications as well.

```
conn hr/ABcd##1234@orawindb

CREATE TABLE ext_orders
( ORDER_ID        NUMBER(12),
  ORDER_DATE      TIMESTAMP(6) WITH LOCAL TIME ZONE,
  CUSTOMER_ID     NUMBER(12),
  ORDER_STATUS    NUMBER(2),
  DELIVERY_TYPE   VARCHAR2(15),
  ORDER_TOTAL     NUMBER(8,2)
)
ORGANIZATION EXTERNAL
 ( TYPE ORACLE_DATAPUMP
   DEFAULT DIRECTORY extdir
   LOCATION ('orders.dmp')
 )
/
```

30. Query the external table.

```
SELECT ORDER_ID, TO_CHAR(ORDER_DATE,'DD-MON-YY') ORDER_DATE, CUSTOMER_ID, ORDER_STATUS,
DELIVERY_TYPE, ORDER_TOTAL
FROM EXT_ORDERS ;
```

Clean up

31. Drop the external tables.

```
conn SOE/ABcd##1234@pdb1
DROP TABLE EXT_ORDERS;

conn hr/ABcd##1234@orawindb
DROP TABLE EXT_ORDERS;
```

32. Drop the directory objects.

```
conn sys/ABcd##1234@pdb1 as sysdba
DROP DIRECTORY EXTDIR;

conn sys/ABcd##1234@orawindb as sysdba
DROP DIRECTORY EXTDIR;
```

33. Exit from SQL Developer

34. In `winsrv`, delete the files in `D:\temp`

35. In the hosting PC, delete the subdirectory `ext` created in the sharing folder.

Summary

- External tables using `ORACLE_LOADER` access driver are used to link text data files to external tables. Those external tables can be used in SQL and PL/SQL as the ordinary internal database tables. However, external tables with `ORACLE_LOADER` access driver cannot be used to unload database objects into external files.

- External tables using `ORACLE_DATAPUMP` access driver are used to upload data into binary dump files. Those files can then be moved to other Oracle databases and link it to external tables using the same access driver.