Oracle Data Pump (Expdp and Impdp)/Backup and Restore in Oracle Database 10g, 11g, 12c, 18c, 19c

Oracle Data Pump enables high-speed movement of data and metadata from one database to another. This technology is the basis for the following Oracle Database data movement utilities:

- **Data Pump Export (Export):** Export is a utility for unloading data and metadata into a set of operating system files called a **dump file set**. The dump file set is made up of one or more binary files that contain table data, database object metadata, and control information.

- **Data Pump Import (Import):** Import is a utility for loading an export dump file set into a database. You can also use Import to load a destination database directly from a source database with no intervening files, which allows export and import operations to run concurrently, minimizing total elapsed time.

Getting Started

For the examples to work we must first unlock the SCOTT account and create a directory object it can access. The directory object is only a pointer to a physical directory, creating it does not actually create the physical directory on the file system of the database server.

```
CONN / AS SYSDBA
ALTER USER scott IDENTIFIED BY tiger ACCOUNT UNLOCK;

CREATE OR REPLACE DIRECTORY test_dir AS '/u01/app/oracle/oradata/';
GRANT READ, WRITE ON DIRECTORY test_dir TO scott;
```

Below is the attached example from command line:

```
SQL> conn sys/password@kbc as sysdba
Connected.
SQL> select name from v$database;

NAME
---------
KBC

SQL> ALTER USER scott IDENTIFIED BY password ACCOUNT UNLOCK;

User altered.

SQL> CREATE OR REPLACE DIRECTORY test_dir AS 'C:\test_dir';

Directory created.

SQL> GRANT READ, WRITE ON DIRECTORY test_dir TO scott;

Grant succeeded.
```

Table Exports/Imports

The TABLES parameter is used to specify the tables that are to be exported. The following is an example of the table export and import syntax.

```
expdp scott/tiger@db10g tables=EMP,DEPT directory=TEST_DIR dumpfile=EMP_DEPT.dmp
logfile=expdpEMP_DEPT.log

impdp scott/tiger@db10g tables=EMP,DEPT directory=TEST_DIR dumpfile=EMP_DEPT.dmp
logfile=impdpEMP_DEPT.log
```

Here is the attached example in command line:

```
C:\Users\Sapana>expdp scott/password@kbc tables=EMP,DEPT directory=test_dir
dumpfile=EMP_DEPT.dmp logfile=expdpEMP_DEPT.log
```

So to restore (import) table EMP, DEPT in ORCL database, create user scott in ORCL database and grant dba to scott.

```
NAME
---------
ORCL

SQL> create user scott identified by password;

User created.

SQL> create or replace directory test_dir as 'C:\test_dir';

Directory created.

SQL> grant write, read on directory test_dir to scott;

Grant succeeded.

SQL> grant dba to scott;

Grant succeeded.

SQL> exit
```

Then, finally import table EMP_DEPT to scott schema.

```
C:\Users\Sapana>impdp scott/password@ORCL tables=EMP,DEPT directory=test_dir
 dumpfile=EMP_DEPT.dmp logfile=impdpEMP_DEPT.log
```

For example output files see expdpEMP_DEPT.log and impdpEMP_DEPT.log.

The TABLE_EXISTS_ACTION=APPEND parameter allows data to be imported into existing tables.

Schema Exports/Imports

The OWNER parameter of exp has been replaced by the SCHEMAS parameter which is used to specify the schemas to be exported. The following is an example of the schema export and import syntax.

```
expdp scott/tiger@db10g schemas=SCOTT directory=TEST_DIR dumpfile=SCOTT.dmp
logfile=expdpSCOTT.log

impdp scott/tiger@db10g schemas=SCOTT directory=TEST_DIR dumpfile=SCOTT.dmp
logfile=impdpSCOTT.log
```

For example output files see expdpSCOTT.log and impdpSCOTT.log.

Database Exports/Imports

The FULL parameter indicates that a complete database export is required. The following is an example of the full database export and import syntax.

```
expdp system/password@db10g full=Y directory=TEST_DIR dumpfile=DB10G.dmp
logfile=expdpDB10G.log

impdp system/password@db10g full=Y directory=TEST_DIR dumpfile=DB10G.dmp
logfile=impdpDB10G.log
```

For an example output file see expdpDB10G.log.

That database user performing the export will need `DATAPUMP_EXP_FULL_DATABASE` role, and the user performing the import will need the `DATAPUMP_IMP_FULL_DATABASE` role.