ORACLE

12

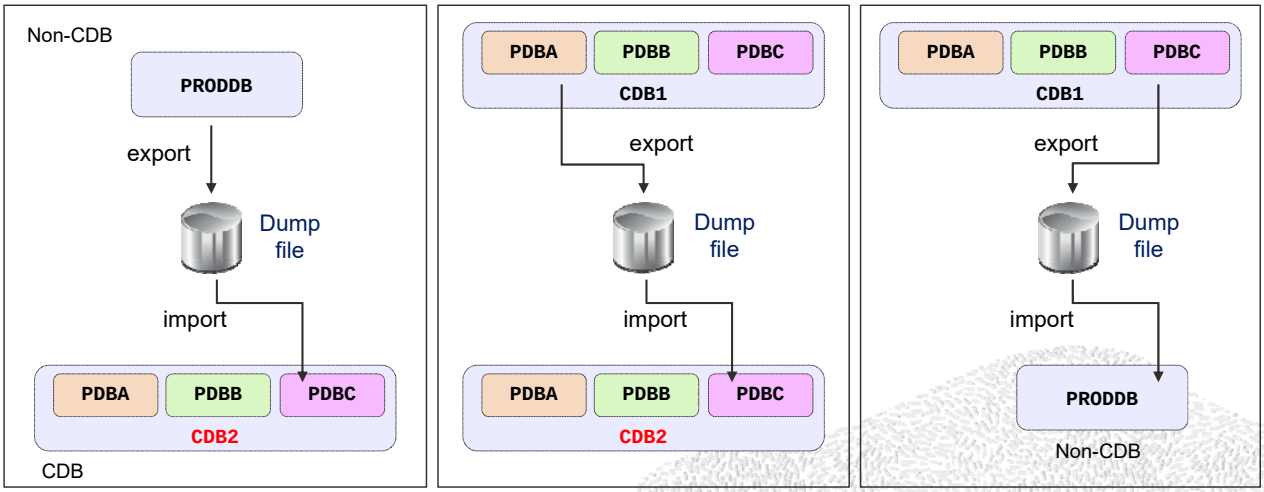
Data Movement

Objectives

After completing this lesson, you should be able to:

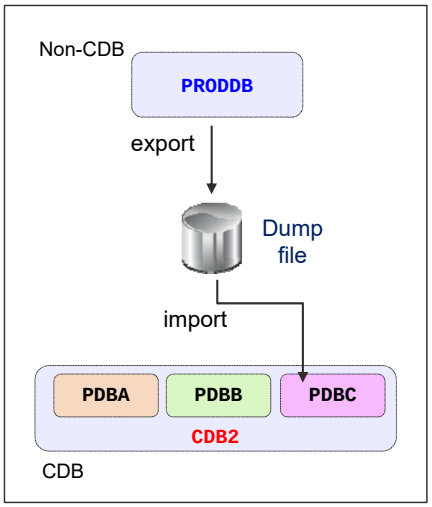
- Export from a non-CDB and import into a PDB
- Export from a PDB and import into a PDB
- Export from a PDB and import into a non-CDB
- Use SQL*Loader to load data into a PDB

Using Oracle Data Pump with PDBs



Use the PDB service name to export from or import into a PDB.

Exporting from Non-CDB and Importing into PDB



1. Export **PRODDB** with FULL clause:

```
$ expdp system@PRODDB FULL=Y DUMPFILE=proddb.dmp
```

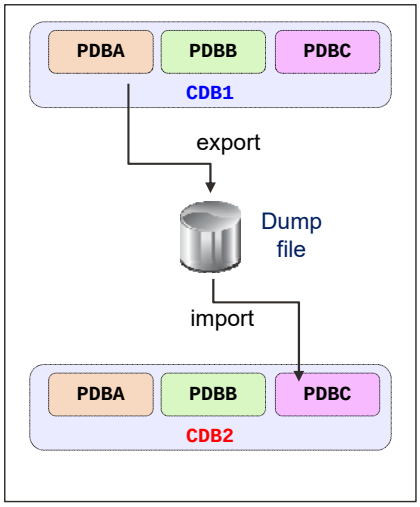
2. If **PDBC** does not exist in **CDB2**, create **PDBC** in **CDB2**:

```
SQL> CONNECT sys@CDB1
SQL> CREATE PLUGGABLE DATABASE PDBC …;
```

3. Open **PDBC**.
4. Create a Data Pump directory in **PDBC**.
5. Copy the dumpfile to the Data Pump directory.
6. Create same **PRODDB** tablespaces in **PDBC** for new local users' objects.
7. Import into **PDBC** with FULL and REMAP clauses:

```
$ impdp system@PDBC FULL=Y DUMPFILE=proddb.dmp
```

Exporting and Importing Between PDBs



1. Export PDBA from CDB1 with FULL clause:

```
$ expdp system@PDBA FULL=Y …
```

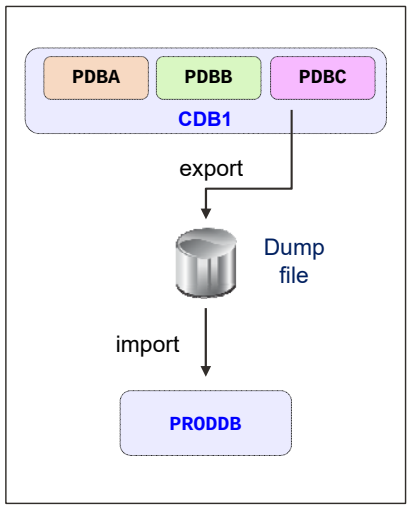
2. If PDBC does not exist in CDB2, create PDBC in CDB2:

```
SQL> CONNECT sys@CDB2
SQL> CREATE PLUGGABLE DATABASE PDBC …;
```

3. Open PDBC.
4. Create a Data Pump directory in PDBC.
5. Copy the dumpfile to the directory.
6. Create same PDBA tablespaces in PDBC for new local users objects.
7. Import into PDBC of CDB2 with FULL and REMAP clauses:

```
$ impdp system@PDBC FULL=Y REMAP_SCHEMA=c##u:lu…
```


Exporting from PDB and Importing into Non-CDB



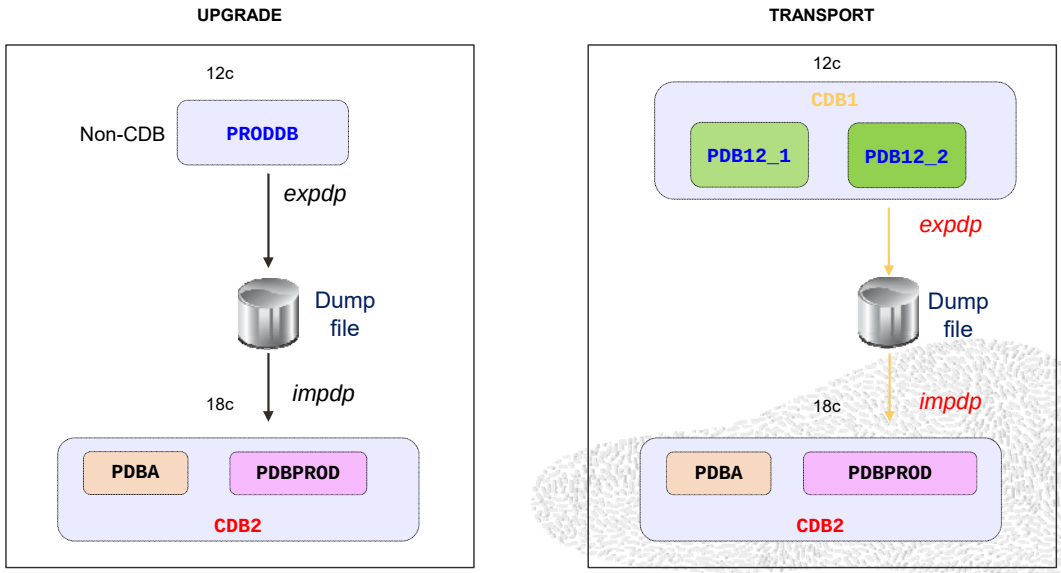
1. Export PDBC of CDB1 with FULL clause:

```
$ expdp system@PDBC FULL=Y …
```

2. Import into PRODDB with FULL and REMAP clauses:

```
$ impdp system@PRODDB FULL=Y
      REMAP_SCHEMA=c##u:local_u
```

Full Transportable Export/Import: Overview



7

Full Transportable Export/Import: Usage

A full transportable export exports all objects and data necessary to create a complete copy of the database.

- TRANSPORTABLE=ALWAYS parameter
- FULL parameter

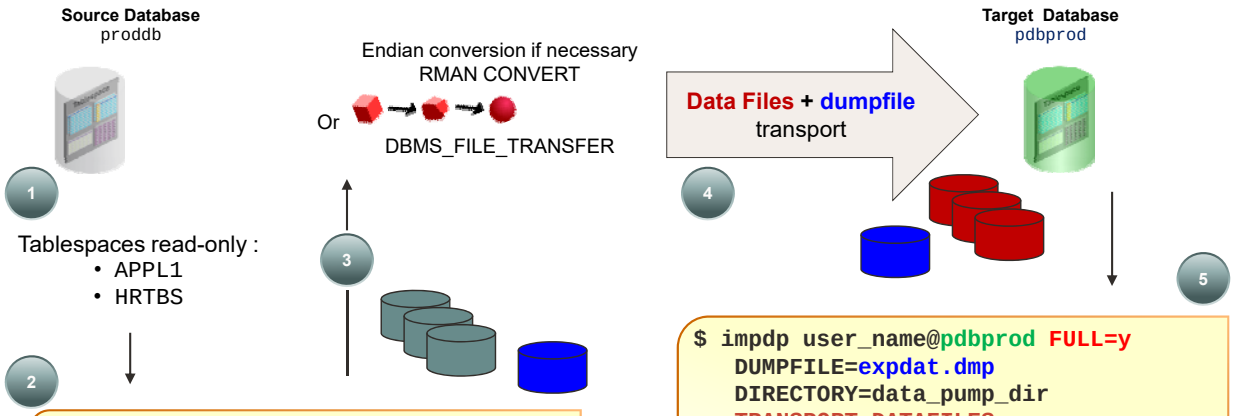
```
$ expdp user_name@pdb FULL=y DUMPFILE=expdat.dmp DIRECTORY=data_pump_dir
                         TRANSPORTABLE=always
```

A full transportable import imports a dump file only if it has been created using the transportable option during export.

- TRANSPORT_DATAFILES
- If the NETWORK_LINK is used, it requires TRANSPORTABLE=ALWAYS parameter.

8

Full Transportable Export/Import: Example




```
$ expdp user_name@proddb FULL=y
      DUMPFILE=expdat.dmp
      DIRECTORY=data_pump_dir
      TRANSPORTABLE=always
      LOGFILE=export.log
```

```
$ impdp user_name@pdbprod FULL=y
    DUMPFILE=expdat.dmp
    DIRECTORY=data_pump_dir
    TRANSPORT_DATAFILES=
    '/oracle/oradata/prod/file1.dbf',
    '/oracle/oradata/prod/file2.dbf',
    '/oracle/oradata/prod/file3.dbf'
    LOGFILE=import.log
```


Transporting a Database Over the Network: Example

Transport a database over the network: perform an import using the NETWORK_LINK parameter.

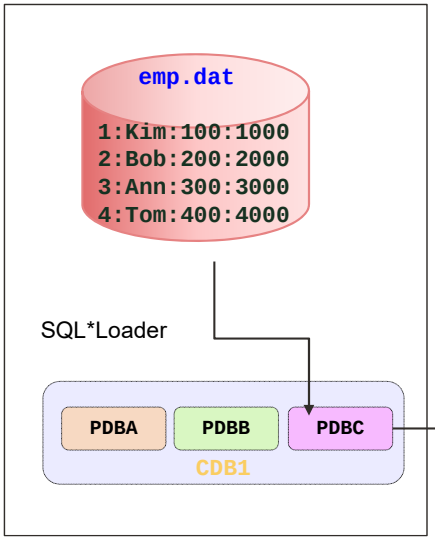
1. Create a database link in the target to the source database.
2. Make the user-defined tablespaces in the source database read-only.
3. Transport the data files for all of the user-defined tablespaces from the source to the target location.
4. Perform conversion of the data files if necessary.
5. Import in the target database.

```
$ impdp username@pdbname full=Y network_link = sourcedb
      transportable = always
      transport_datafiles = '/oracle/oradata/prod/sales01.dbf',
                            '/oracle/oradata/prod/cust01.dbf'
      logfile=import.log
```

Additional features of Oracle Data Pump

- Oracle Data Pump Ability to Exclude ENCRYPTION Clause on Import
- Oracle Data Pump Allows Tablespaces to Stay Read-Only During TTS Import
- Oracle Data Pump Support for Resource Usage Limitations
- Oracle Data Pump Prevents Inadvertent Use of Protected Roles

Using SQL*Loader with PDBs

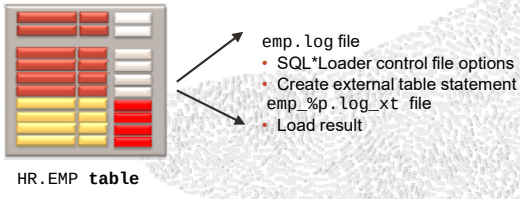


1. Use SQL*Loader Express Mode to insert rows into the HR.EMP table in PDBC.

```
$ sqlldr system@PDBC TABLE=hr.emp
```

– No need to prepare a control file:
  – The table columns must be scalar data types (character, number, or datetime).
  – SQL*Loader uses table column definitions to determine input data types.

2. Use log files to verify load operation.



Summary

In this lesson, you should have learned how to:

- Export from a non-CDB and import into a PDB
- Export from a PDB and import into a PDB
- Export from a PDB and import into a non-CDB
- Use SQL*Loader to load data into a PDB

13

Practice 12: Overview

- 12-1: Performing a full transportable export/import from a 12c non-CDB into a 19c PDB
- 12-2: Performing a full transportable export/import from an 18c PDB into a 19c PDB

14