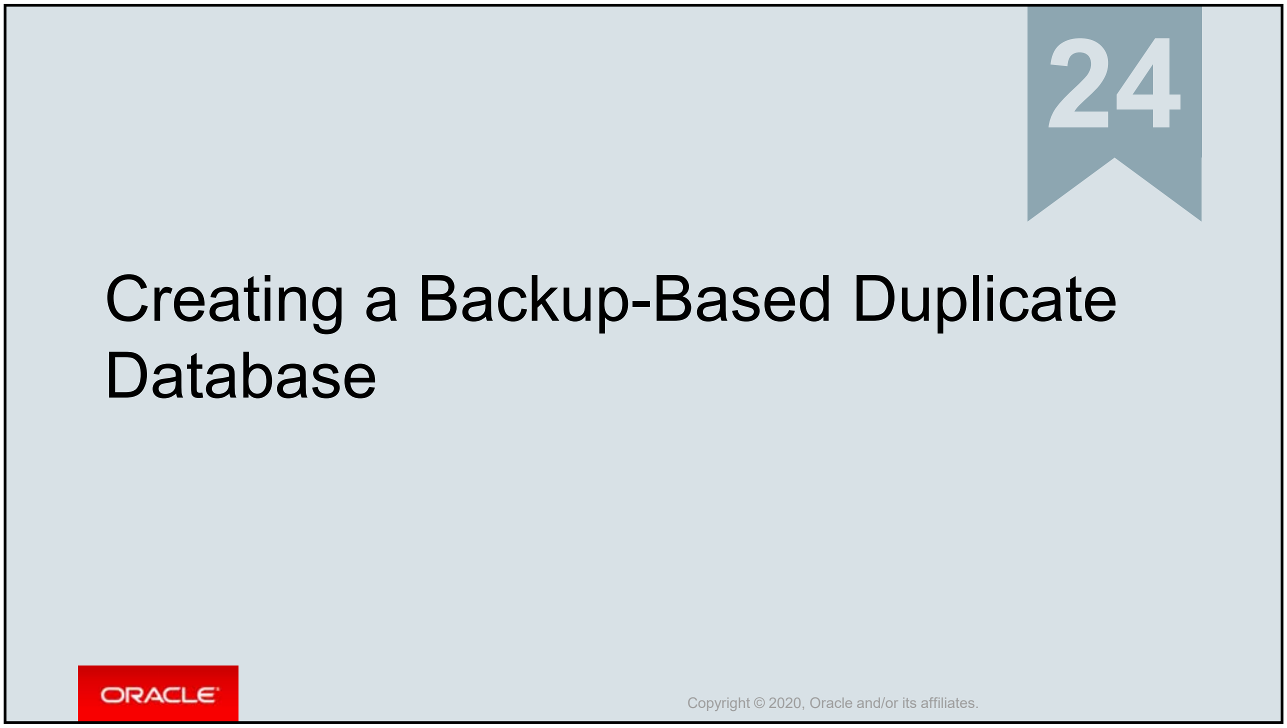
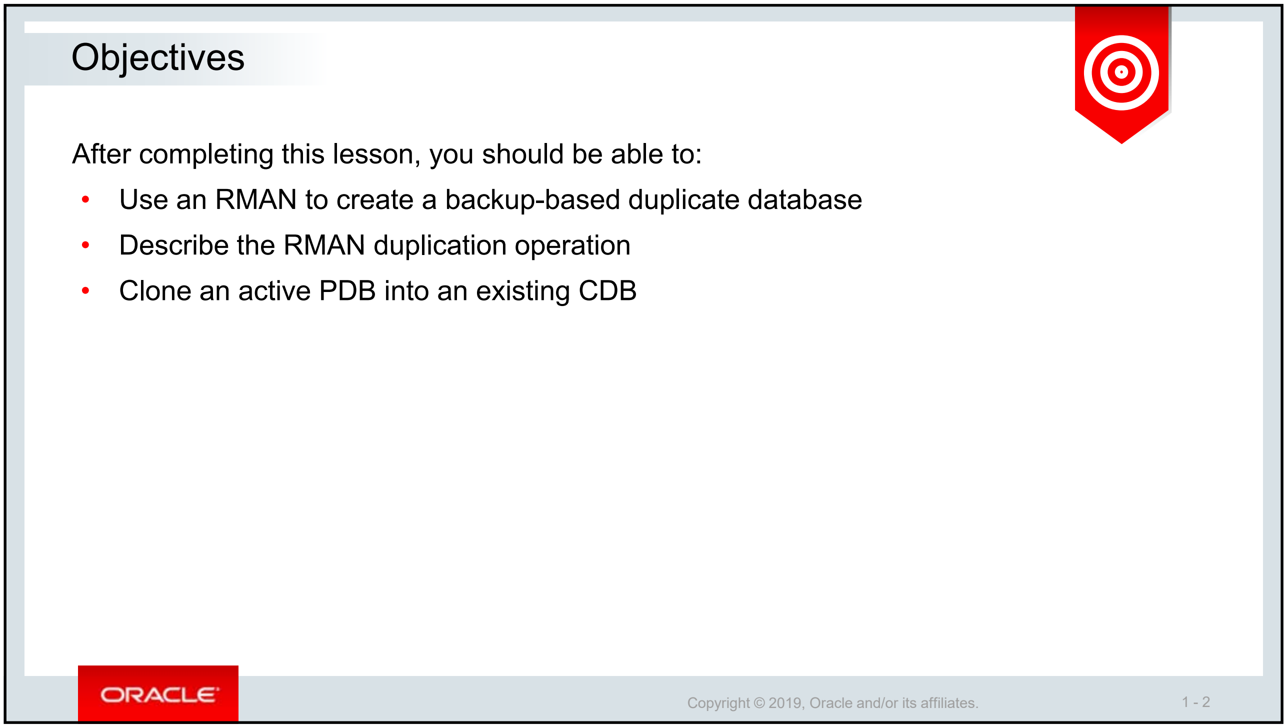


24

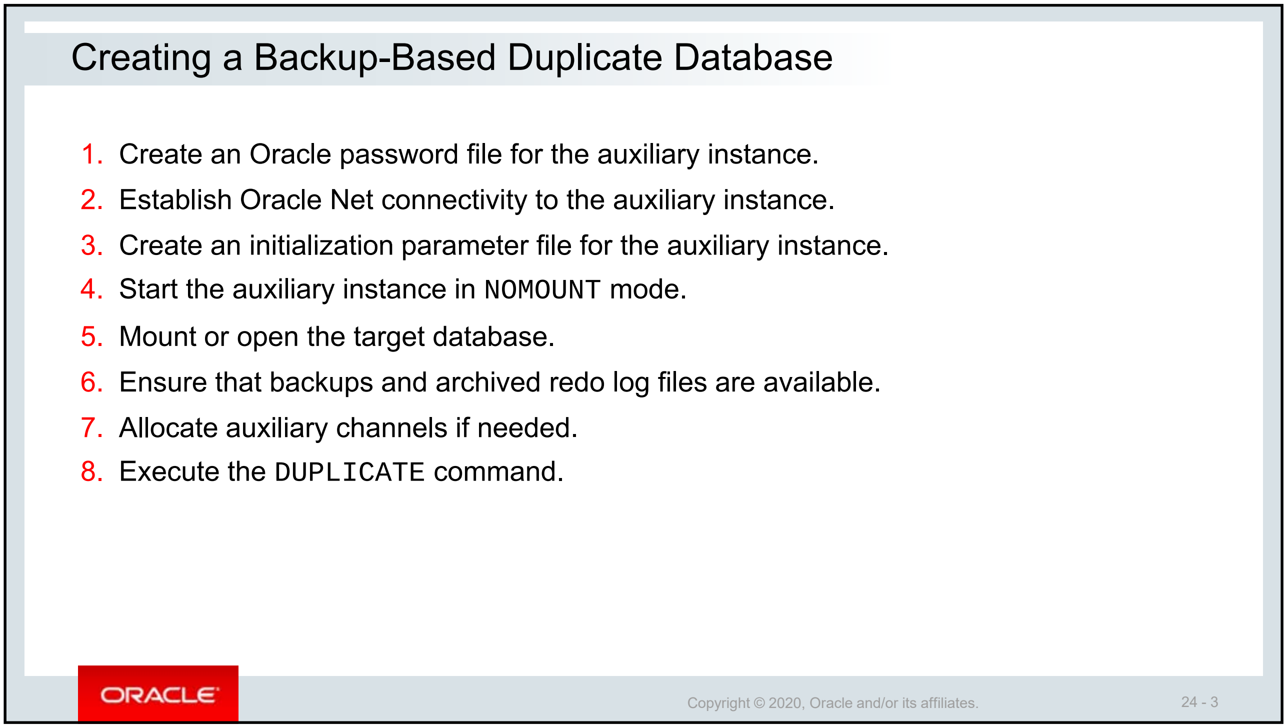
Creating a Backup-Based Duplicate Database



Objectives

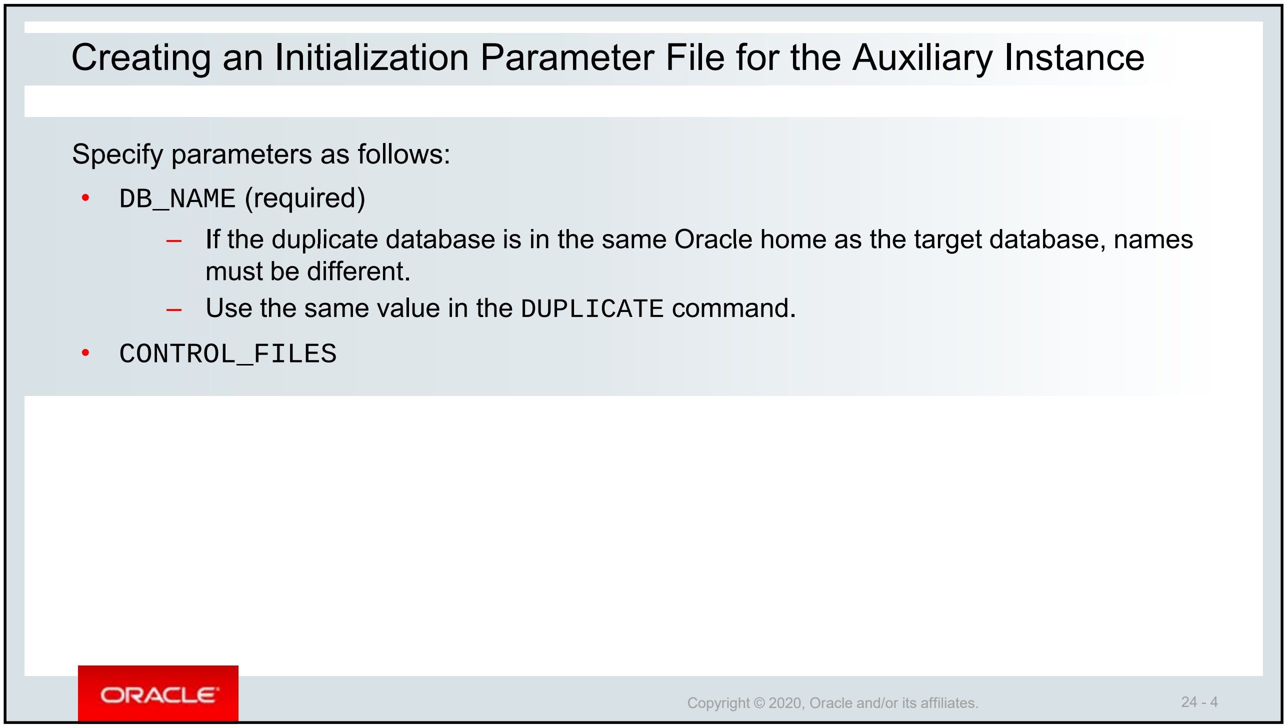
After completing this lesson, you should be able to:

- Use an RMAN to create a backup-based duplicate database
- Describe the RMAN duplication operation
- Clone an active PDB into an existing CDB



Creating a Backup-Based Duplicate Database

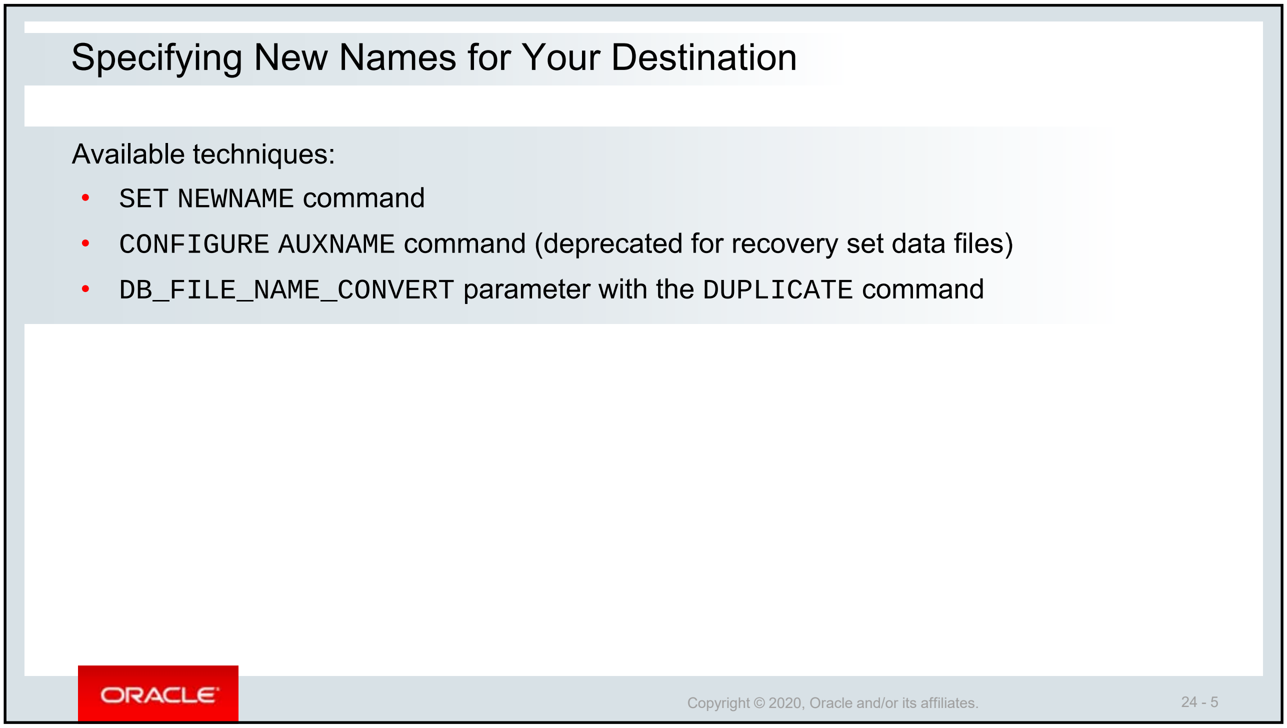
1. Create an Oracle password file for the auxiliary instance.
2. Establish Oracle Net connectivity to the auxiliary instance.
3. Create an initialization parameter file for the auxiliary instance.
4. Start the auxiliary instance in `NOMOUNT` mode.
5. Mount or open the target database.
6. Ensure that backups and archived redo log files are available.
7. Allocate auxiliary channels if needed.
8. Execute the `DUPLICATE` command.



Creating an Initialization Parameter File for the Auxiliary Instance

Specify parameters as follows:

- `DB_NAME` (required)
  - If the duplicate database is in the same Oracle home as the target database, names must be different.
  - Use the same value in the `DUPLICATE` command.
- `CONTROL_FILES`

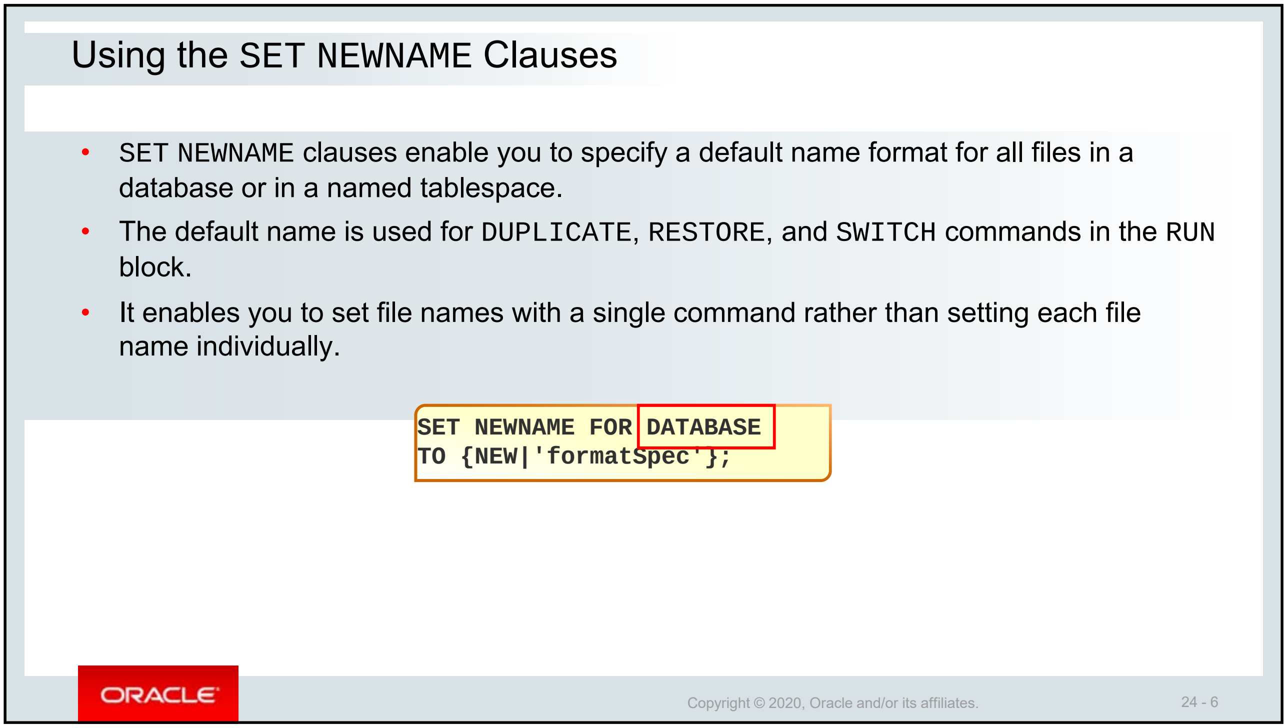


Specifying New Names for Your Destination

Available techniques:

- SET NEWNAME command
- CONFIGURE AUXNAME command (deprecated for recovery set data files)
- DB_FILE_NAME_CONVERT parameter with the DUPLICATE command

ORACLE

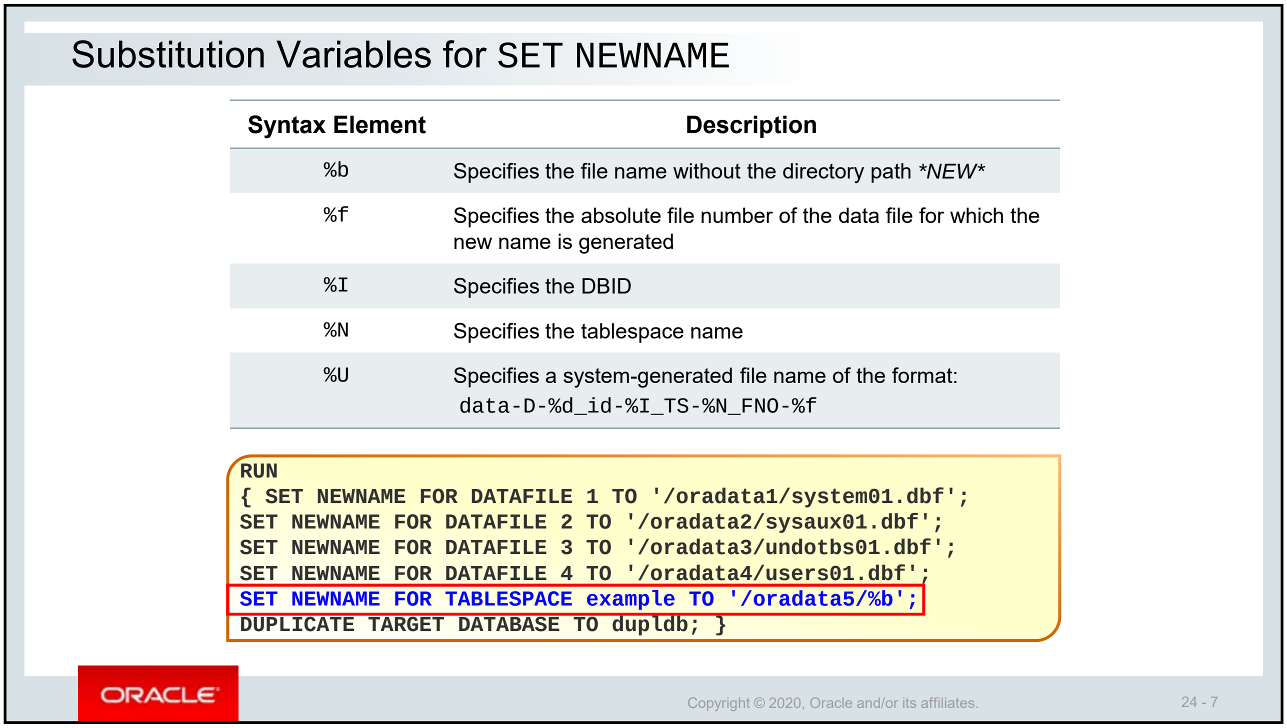


Using the SET NEWNAME Clauses

- SET NEWNAME clauses enable you to specify a default name format for all files in a database or in a named tablespace.
- The default name is used for DUPLICATE, RESTORE, and SWITCH commands in the RUN block.
- It enables you to set file names with a single command rather than setting each file name individually.

```
SET NEWNAME FOR DATABASE
TO {NEW|'formatSpec'};
```

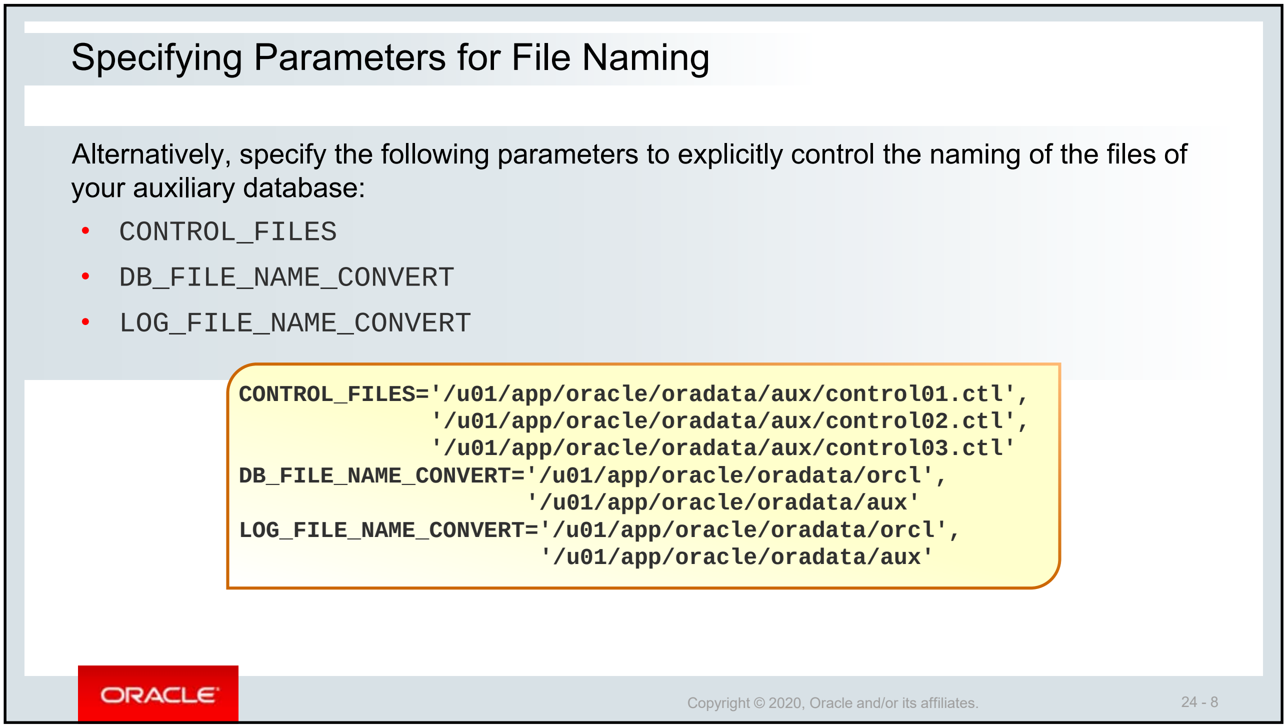
ORACLE



Substitution Variables for SET NEWNAME

Syntax Element	Description
%b	Specifies the file name without the directory path **NEW**
%f	Specifies the absolute file number of the data file for which the new name is generated
%I	Specifies the DBID
%N	Specifies the tablespace name
%U	Specifies a system-generated file name of the format: `data-D-%d_id-%I_TS-%N_FNO-%f`

```
RUN
{ SET NEWNAME FOR DATAFILE 1 TO '/oradata1/system01.dbf';
SET NEWNAME FOR DATAFILE 2 TO '/oradata2/sysaux01.dbf';
SET NEWNAME FOR DATAFILE 3 TO '/oradata3/undotbs01.dbf';
SET NEWNAME FOR DATAFILE 4 TO '/oradata4/users01.dbf';
SET NEWNAME FOR TABLESPACE example TO '/oradata5/%b';
DUPLICATE TARGET DATABASE TO dupldb; }
```

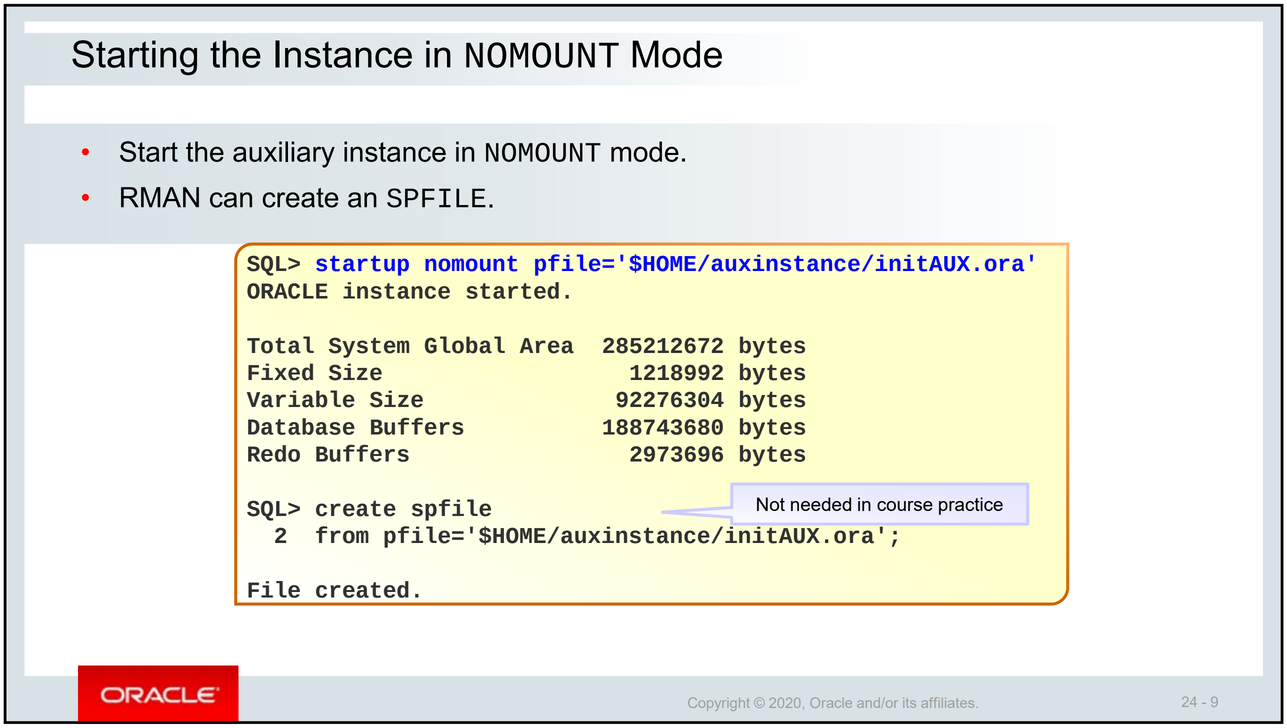


Specifying Parameters for File Naming

Alternatively, specify the following parameters to explicitly control the naming of the files of your auxiliary database:

- `CONTROL_FILES`
- `DB_FILE_NAME_CONVERT`
- `LOG_FILE_NAME_CONVERT`

```
CONTROL_FILES='/u01/app/oracle/oradata/aux/control01.ctl',
              '/u01/app/oracle/oradata/aux/control02.ctl',
              '/u01/app/oracle/oradata/aux/control03.ctl'
DB_FILE_NAME_CONVERT='/u01/app/oracle/oradata/orcl',
                     '/u01/app/oracle/oradata/aux'
LOG_FILE_NAME_CONVERT='/u01/app/oracle/oradata/orcl',
                      '/u01/app/oracle/oradata/aux'
```

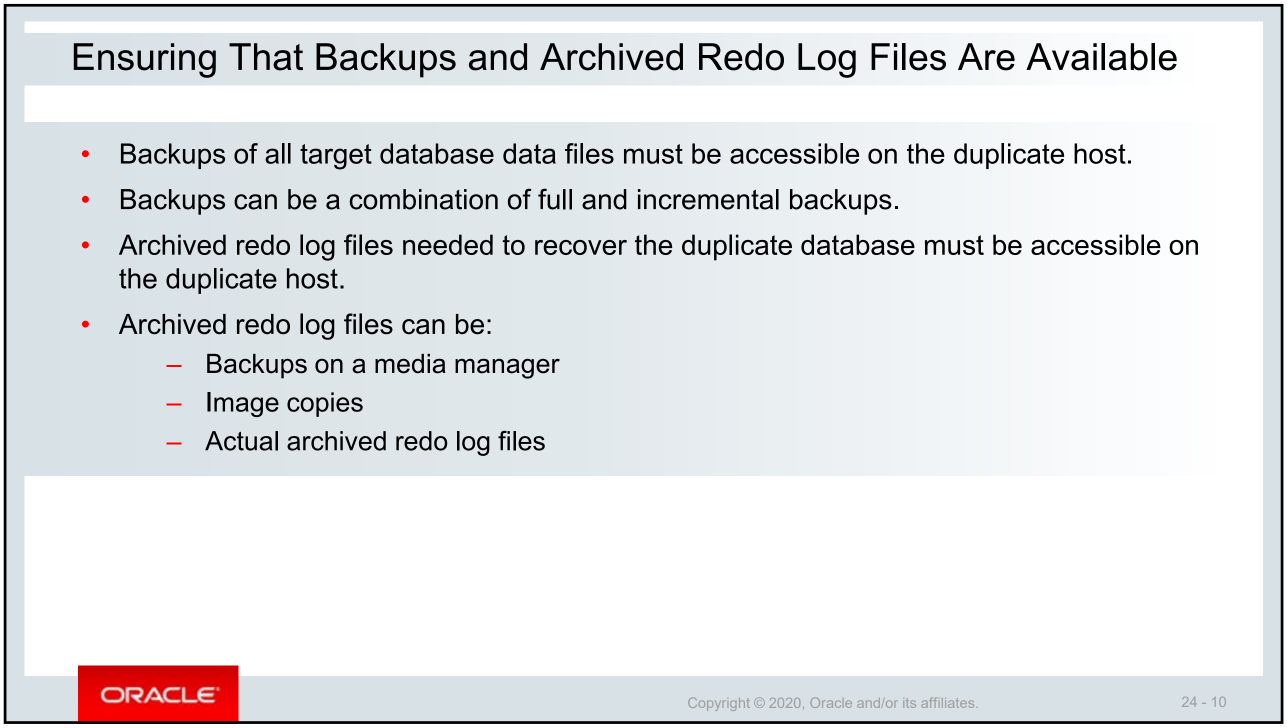


Starting the Instance in NOMOUNT Mode

- Start the auxiliary instance in NOMOUNT mode.
- RMAN can create an SPFILE.

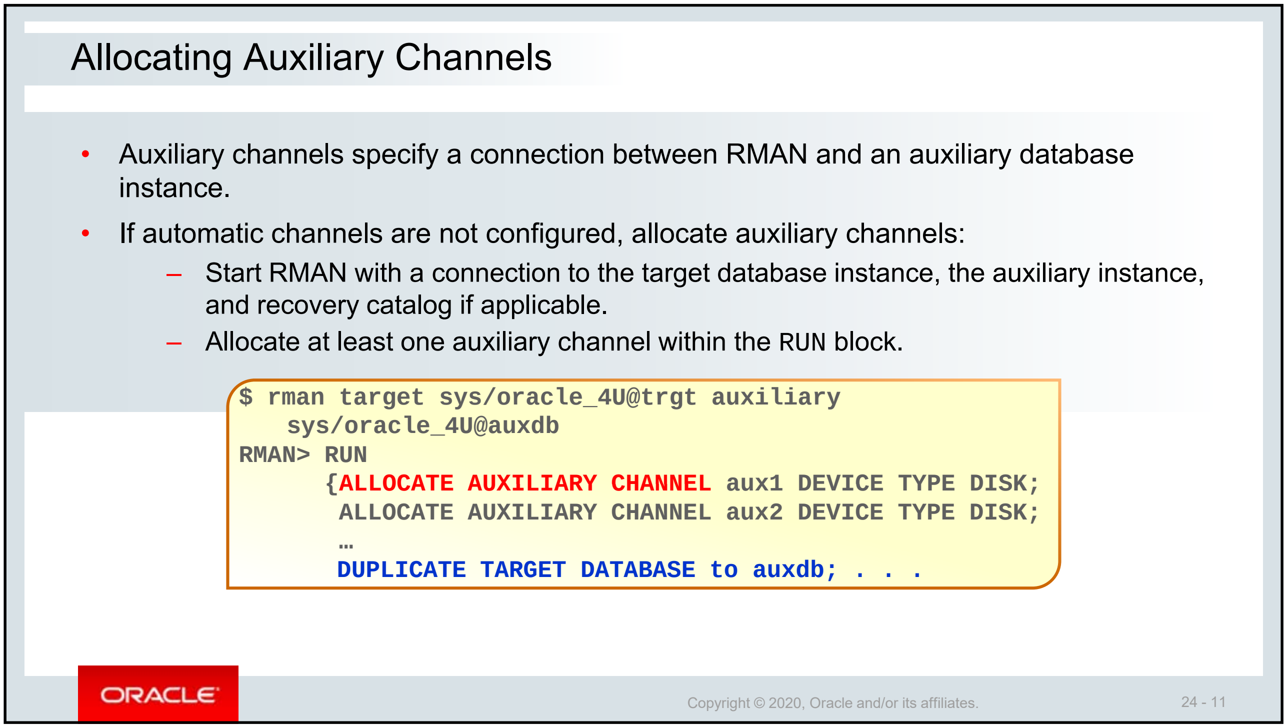
```
SQL> startup nomount pfile='$HOME/auxinstance/initAUX.ora'
ORACLE instance started.

Total System Global Area  285212672 bytes
Fixed Size                  1218992 bytes
Variable Size              92276304 bytes
Database Buffers          188743680 bytes
Redo Buffers                2973696 bytes

SQL> create spfile
  2  from pfile='$HOME/auxinstance/initAUX.ora';

File created.
```

Not needed in course practice



Ensuring That Backups and Archived Redo Log Files Are Available

- Backups of all target database data files must be accessible on the duplicate host.
- Backups can be a combination of full and incremental backups.
- Archived redo log files needed to recover the duplicate database must be accessible on the duplicate host.
- Archived redo log files can be:
  - Backups on a media manager
  - Image copies
  - Actual archived redo log files

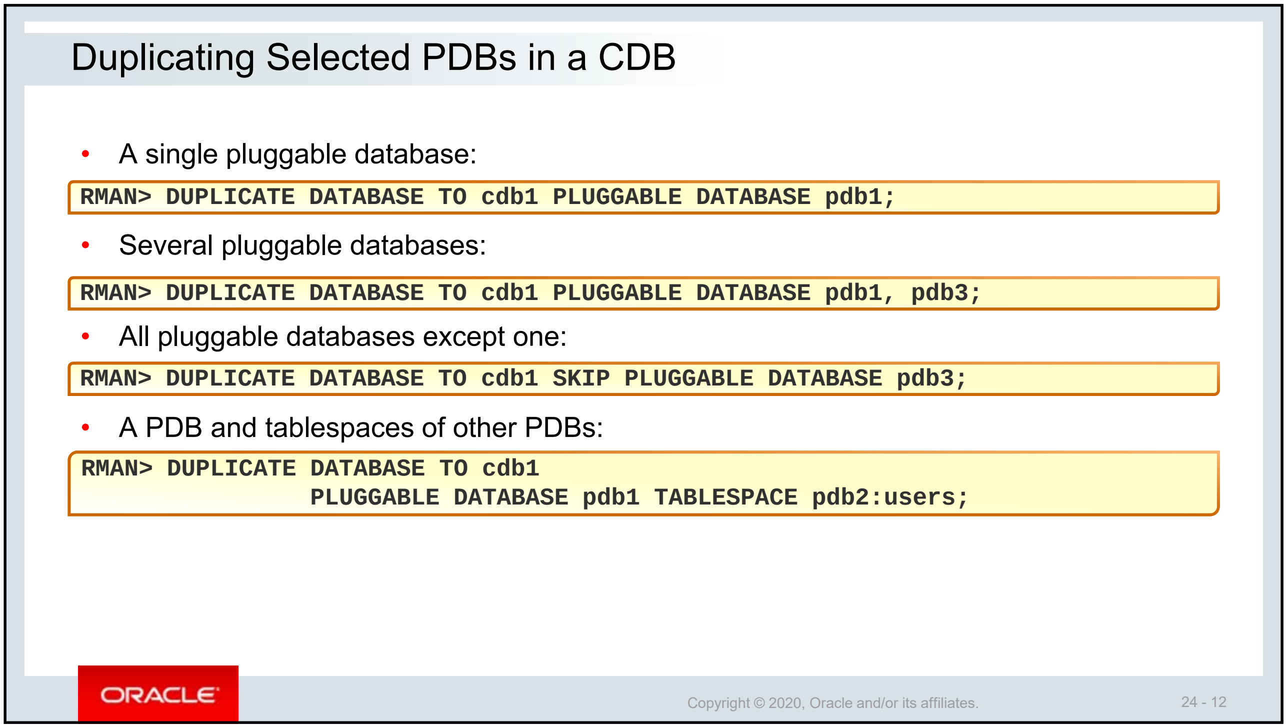


Allocating Auxiliary Channels

- Auxiliary channels specify a connection between RMAN and an auxiliary database instance.
- If automatic channels are not configured, allocate auxiliary channels:
  - Start RMAN with a connection to the target database instance, the auxiliary instance, and recovery catalog if applicable.
  - Allocate at least one auxiliary channel within the RUN block.

```
$ rman target sys/oracle_4U@trgt auxiliary
    sys/oracle_4U@auxdb
RMAN> RUN
       {ALLOCATE AUXILIARY CHANNEL aux1 DEVICE TYPE DISK;
        ALLOCATE AUXILIARY CHANNEL aux2 DEVICE TYPE DISK;
        …
        DUPLICATE TARGET DATABASE to auxdb; . . .
```

ORACLE



Duplicating Selected PDBs in a CDB

- A single pluggable database:

```
RMAN> DUPLICATE DATABASE TO cdb1 PLUGGABLE DATABASE pdb1;
```

- Several pluggable databases:

```
RMAN> DUPLICATE DATABASE TO cdb1 PLUGGABLE DATABASE pdb1, pdb3;
```

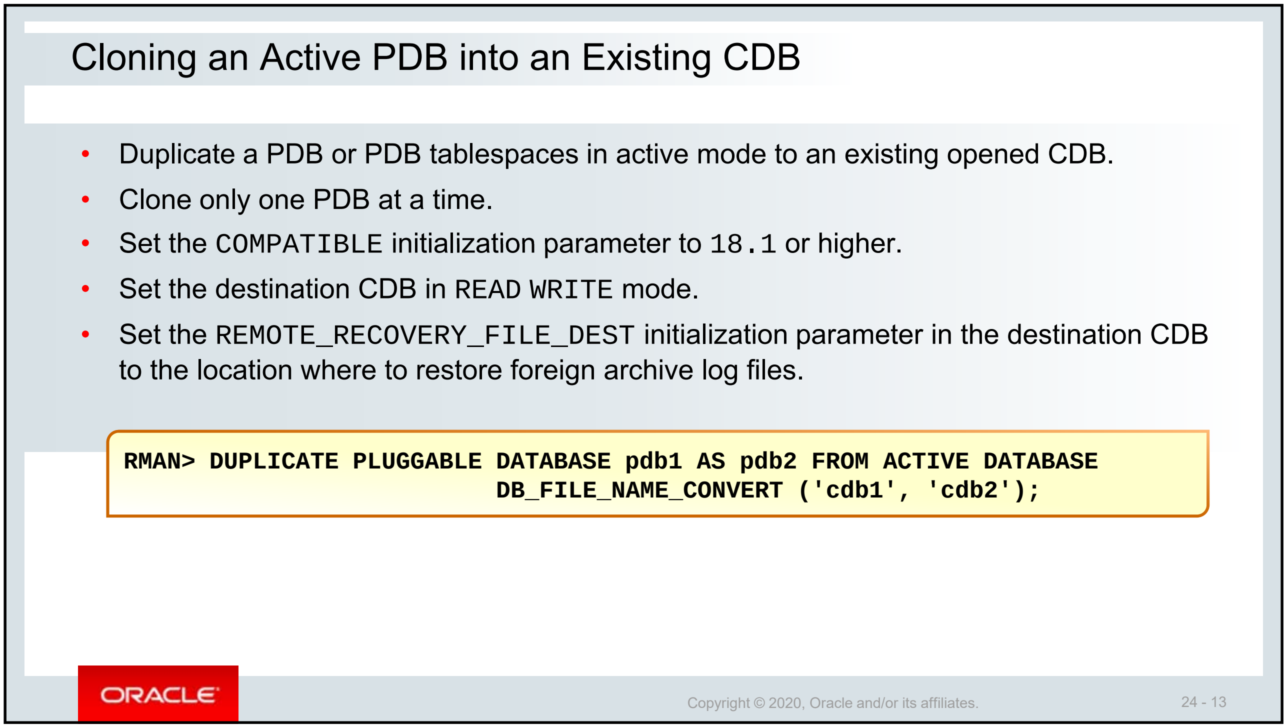
- All pluggable databases except one:

```
RMAN> DUPLICATE DATABASE TO cdb1 SKIP PLUGGABLE DATABASE pdb3;
```

- A PDB and tablespaces of other PDBs:

```
RMAN> DUPLICATE DATABASE TO cdb1
                PLUGGABLE DATABASE pdb1 TABLESPACE pdb2:users;
```

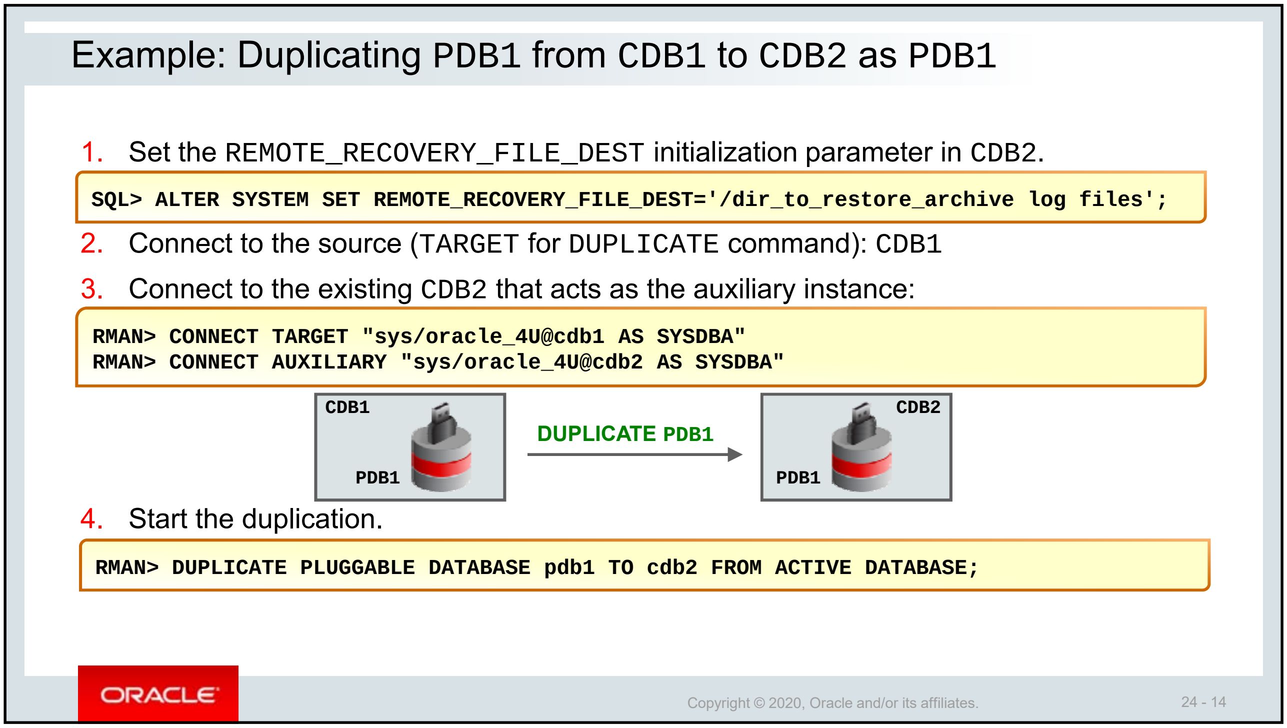
ORACLE



Cloning an Active PDB into an Existing CDB

- Duplicate a PDB or PDB tablespaces in active mode to an existing opened CDB.
- Clone only one PDB at a time.
- Set the `COMPATIBLE` initialization parameter to `18.1` or higher.
- Set the destination CDB in `READ WRITE` mode.
- Set the `REMOTE_RECOVERY_FILE_DEST` initialization parameter in the destination CDB to the location where to restore foreign archive log files.

```
RMAN> DUPLICATE PLUGGABLE DATABASE pdb1 AS pdb2 FROM ACTIVE DATABASE
                                   DB_FILE_NAME_CONVERT ('cdb1', 'cdb2');
```



Example: Duplicating PDB1 from CDB1 to CDB2 as PDB1

1. Set the `REMOTE_RECOVERY_FILE_DEST` initialization parameter in `CDB2`.

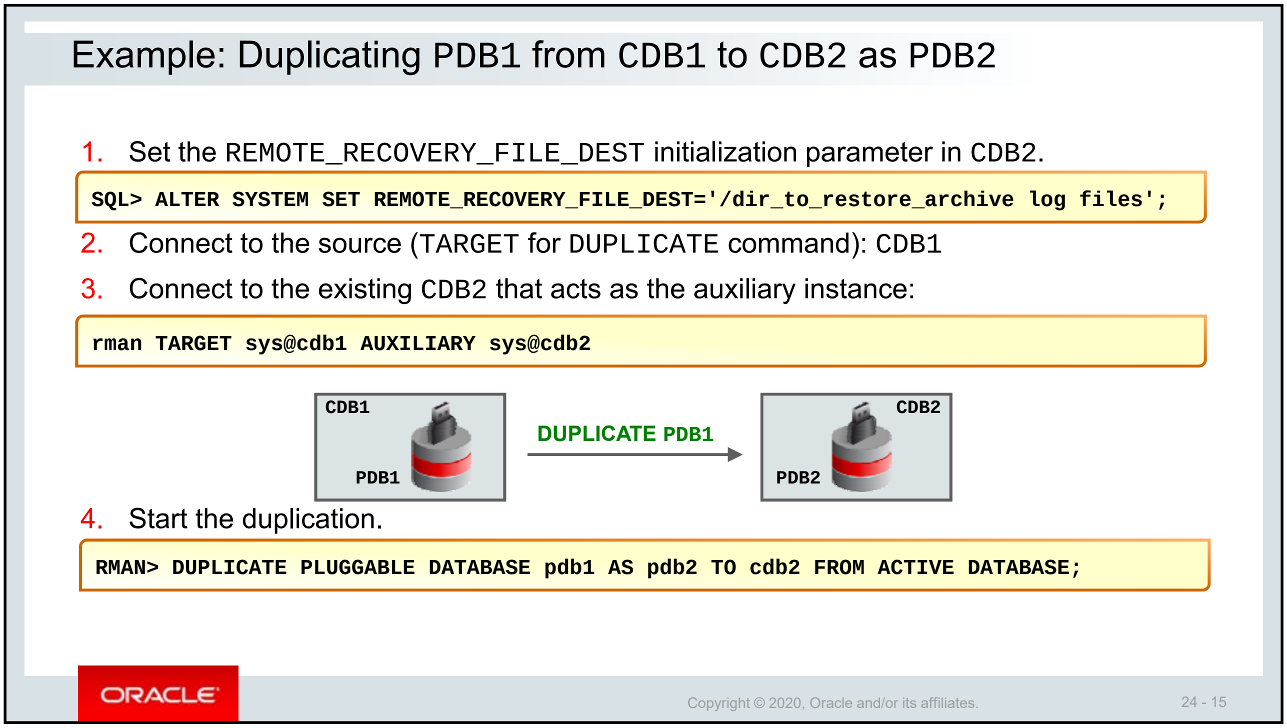
```
SQL> ALTER SYSTEM SET REMOTE_RECOVERY_FILE_DEST='/dir_to_restore_archive log files';
```

2. Connect to the source (`TARGET` for `DUPLICATE` command): `CDB1`
3. Connect to the existing `CDB2` that acts as the auxiliary instance:

```
RMAN> CONNECT TARGET "sys/oracle_4U@cdb1 AS SYSDBA"
RMAN> CONNECT AUXILIARY "sys/oracle_4U@cdb2 AS SYSDBA"
```

4. Start the duplication.

```
RMAN> DUPLICATE PLUGGABLE DATABASE pdb1 TO cdb2 FROM ACTIVE DATABASE;
```



Example: Duplicating PDB1 from CDB1 to CDB2 as PDB2

1. Set the REMOTE_RECOVERY_FILE_DEST initialization parameter in CDB2.

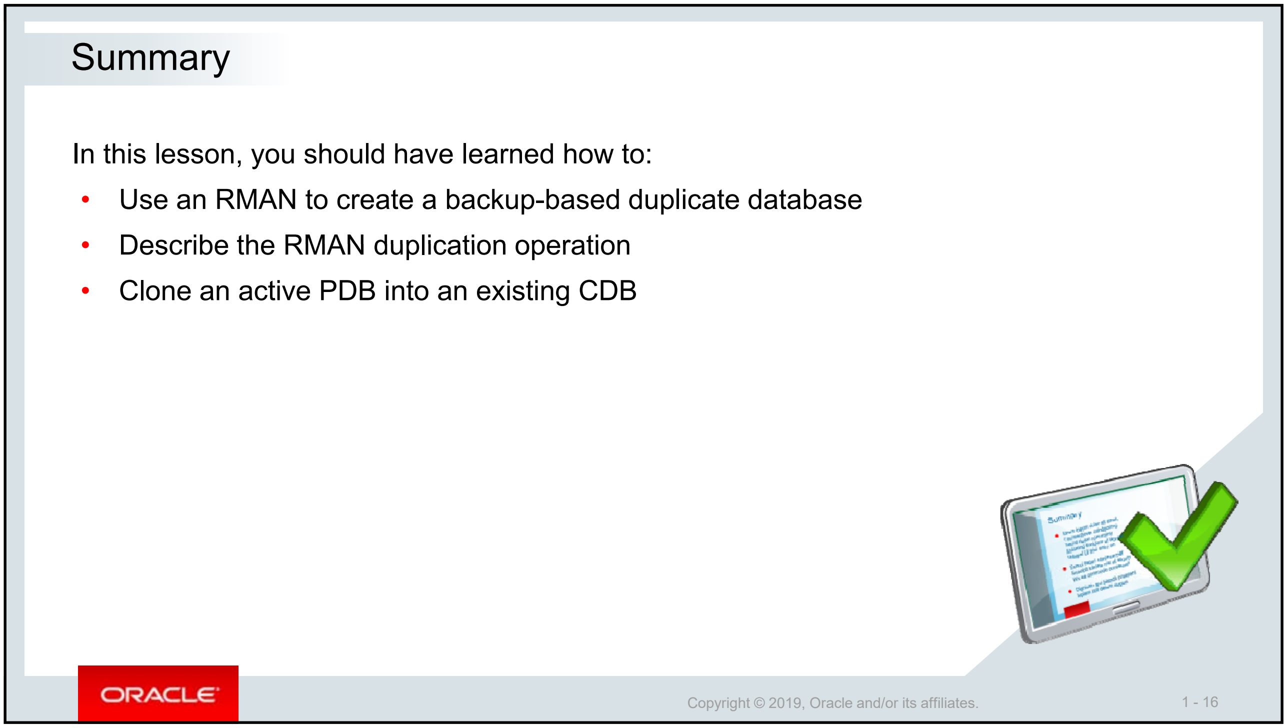
```
SQL> ALTER SYSTEM SET REMOTE_RECOVERY_FILE_DEST='/dir_to_restore_archive log files';
```

2. Connect to the source (TARGET for DUPLICATE command): CDB1
3. Connect to the existing CDB2 that acts as the auxiliary instance:

```
rman TARGET sys@cdb1 AUXILIARY sys@cdb2
```

4. Start the duplication.

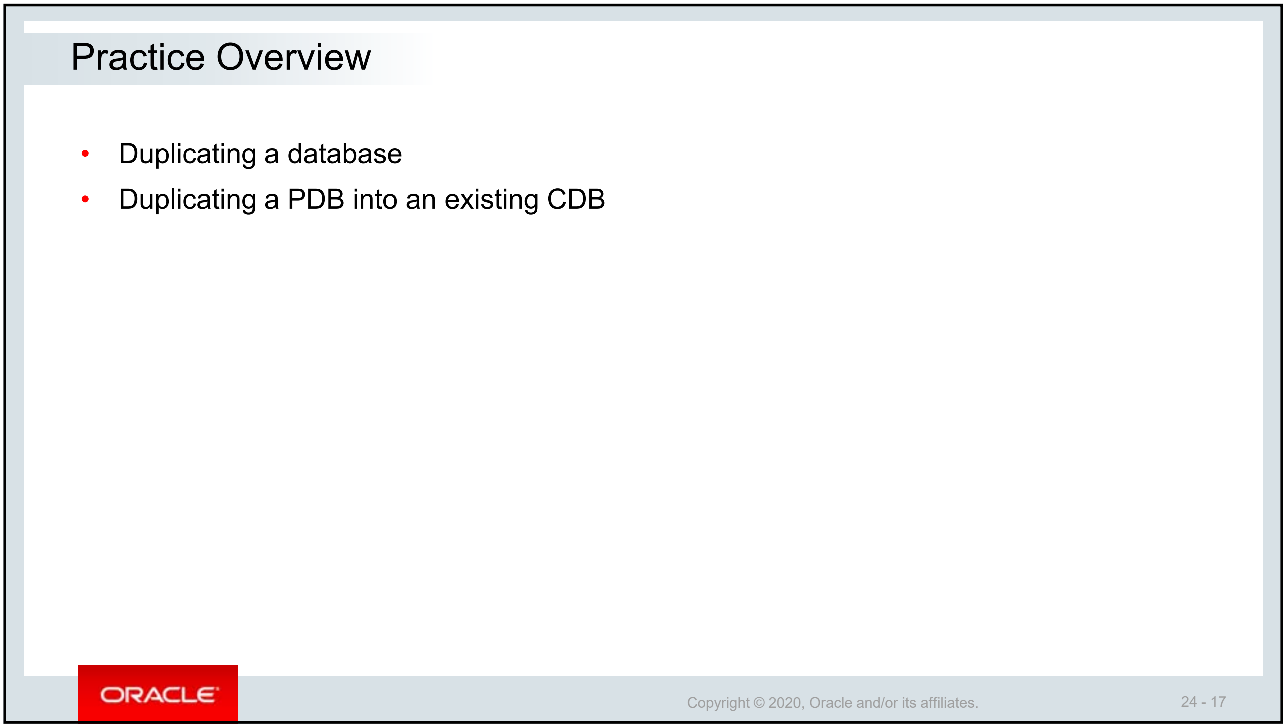
```
RMAN> DUPLICATE PLUGGABLE DATABASE pdb1 AS pdb2 TO cdb2 FROM ACTIVE DATABASE;
```



Summary

In this lesson, you should have learned how to:

- Use an RMAN to create a backup-based duplicate database
- Describe the RMAN duplication operation
- Clone an active PDB into an existing CDB



Practice Overview

- Duplicating a database
- Duplicating a PDB into an existing CDB