20

Duplicating a Database

ORACLE

Objectives

After completing this lesson, you should be able to:

- List the purposes of creating a duplicate database
- Choose a technique for duplicating a database
- Duplicate a database with RMAN
- Use an RMAN backup to duplicate a database
- Duplicate a database based on a running instance

Using a Duplicate Database

- Using a duplicate database to:
  - Test backup and recovery procedures
  - Recover objects by creating an export and importing the objects into the production database
- Creating a duplicate database:
  - With the RMAN `DUPLICATE` command
  - On the same or separate hosts
  - With the identical content, or subset of source
  - Performed by auxiliary channels for backup-up based duplication
  - Performed by target channels for active database duplication

Using a Duplicate Database

A duplicate database is a copy of your target database with a new, unique database identifier (`DBID`). You can operate it independently of the target database to:

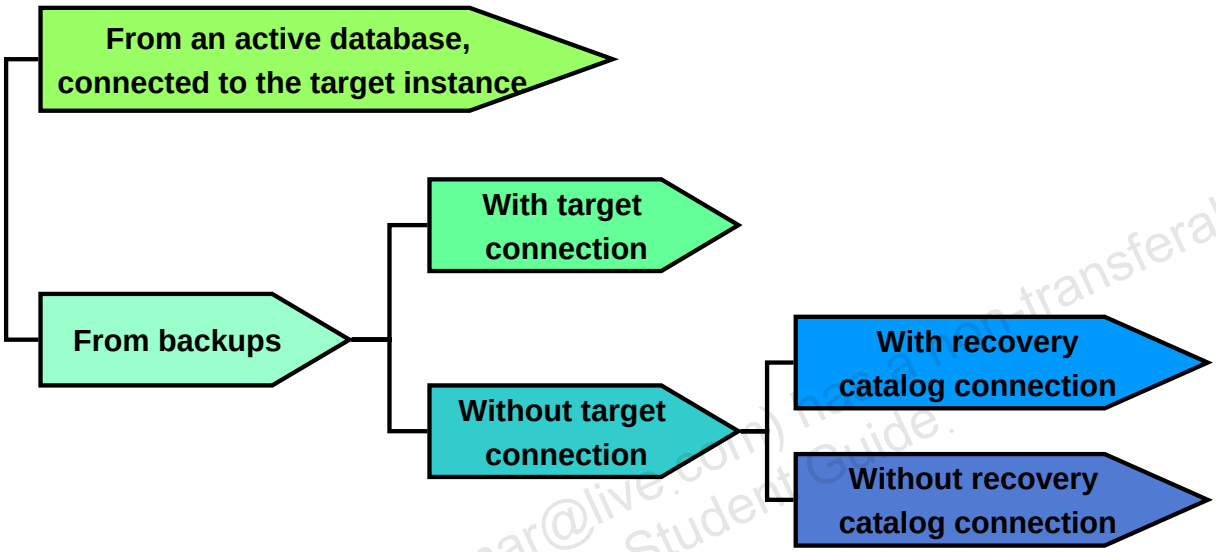
- Test backup and recovery procedures
- Recover objects that were inadvertently dropped from the target database by creating an export containing the objects in the duplicate database and importing them into the production database. Although you probably find that Flashback Query, Flashback Drop, and Flashback Table are a much easier and faster solution to recover objects.

Creating a duplicate database:

- You can use the RMAN `DUPLICATE` command to create a duplicate database on the same host or separate hosts.
- The duplicate database can include the same content or only a subset from the source database (more details later in this lesson).
- The principal work of the duplication is performed by the auxiliary channels. These channels correspond to a server session on the auxiliary instance on the destination host for backup-based duplication.
- For active database duplication, the target channels perform the work of pushing data file copies to the auxiliary instance.

Choosing Database Duplication Techniques

Choosing a technique to duplicate your database—always with connection to the auxiliary instance:



ORACLE

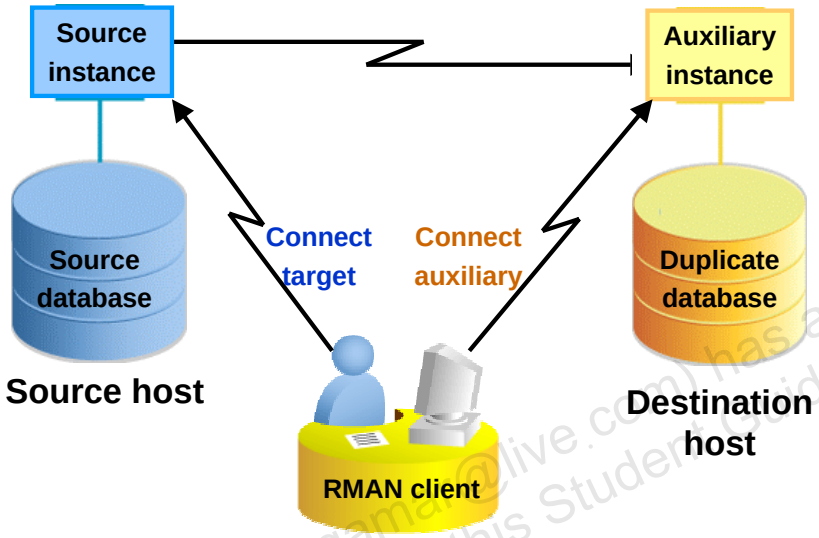
Database Duplication Techniques

You can duplicate a source database to a destination database, which can be on the same or different computers. The database instance associated with the duplicate database is called the auxiliary instance. All duplication techniques require a connection to the auxiliary instance. The diagram shows you the following techniques for database duplication:

- From an active database, connected to the target and auxiliary instances
- From backup, connected to the target and auxiliary instances
- From backup, connected to the auxiliary instance, not connected to the target, but with recovery catalog connection
- From backup, connected to the auxiliary instance, not connected to the target and the recovery catalog

Duplicating an Active Database

- With network (no backups required)
- Including customized SPFILE
- Via Enterprise Manager or RMAN command line



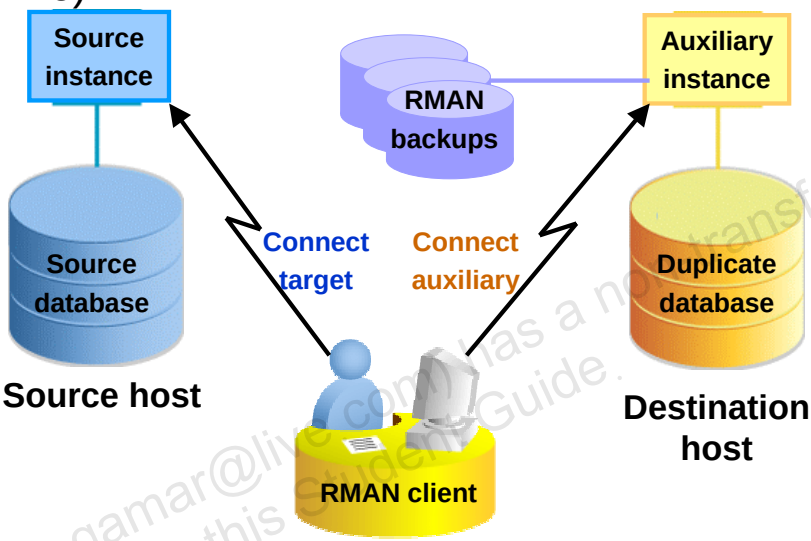
Duplicating an Active Database

You can instruct the source database to perform online image copies and archived log copies directly to the auxiliary instance by using Enterprise Manager or the FROM ACTIVE DATABASE clause of the RMAN DUPLICATE command. Backups are not needed for this operation. RMAN connects as TARGET to the source database instance and as AUXILIARY to the auxiliary instance (as shown in the slide).

The database files are copied from the source to a destination or auxiliary instance via an interinstance network connection. RMAN then uses a “memory script” (one that is contained only in memory) to complete recovery and open the database.

Duplicating a Database with a Target Connection

- Connecting to the target (source database)
- Connecting to the auxiliary instance
- Optionally, connecting to the recovery catalog (or using target control file)



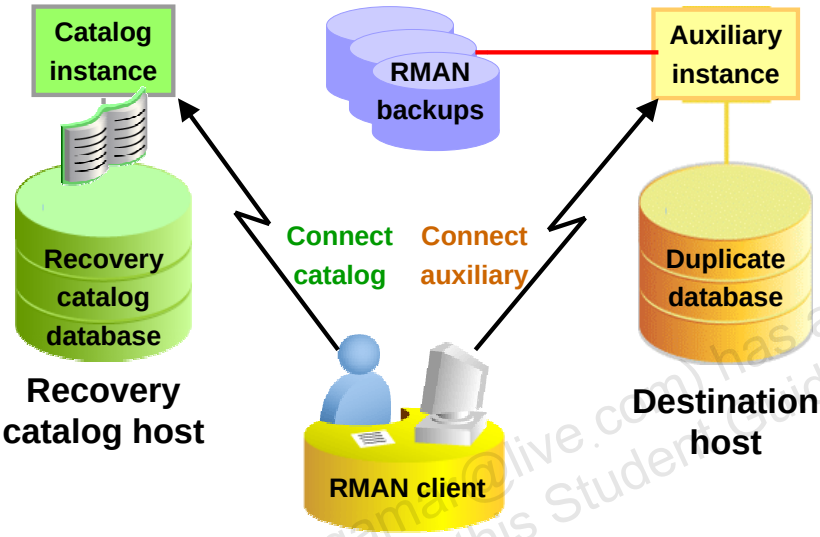
Duplicating a Database with a Target Connection

When you duplicate a database with a target database connection, RMAN can obtain metadata about backups either from the target database control file or from the recovery catalog.

The diagram illustrates backup-based duplication with a target connection. RMAN connects to the source database instance and the auxiliary instance. Optionally, RMAN can connect to a recovery catalog database (not shown in the graphic). The destination host must have access to the RMAN backups required to create the duplicate database.

Duplicating a Database with Recovery Catalog Without Target Connection

- Connecting to a recovery catalog for backup metadata
- Connecting to the auxiliary instance, which must have access to the RMAN backups



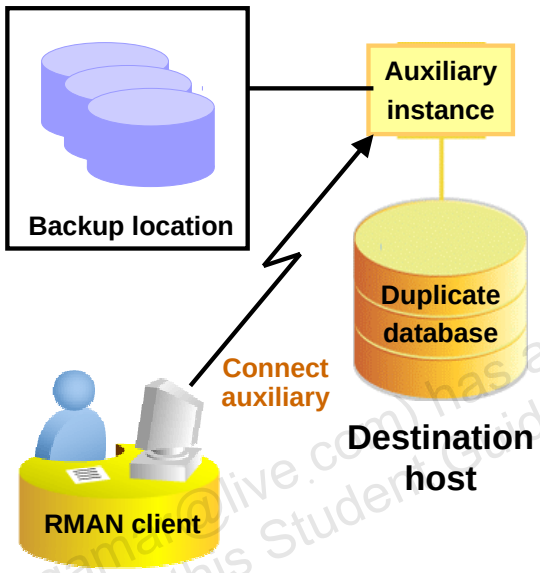
Duplicating a Database Without Target Connection

When you duplicate a database without a target database connection, but with a recovery catalog, RMAN uses the recovery catalog to obtain metadata about the backups.

The diagram illustrates backup-based duplication without a target connection. RMAN connects to a recovery catalog database instance and the auxiliary instance. The destination host must have access to the RMAN backups required to create the duplicate database.

Duplicating a Database Without Recovery Catalog or Target Connection

Connecting to the auxiliary instance, which must have access to a disk `BACKUP LOCATION`



ORACLE

Duplicating a Database Without Recovery Catalog or Target Connection

When you duplicate a database without a target database connection and without a recovery catalog, RMAN uses a `BACKUP LOCATION` where all necessary backups and copies reside.

The diagram illustrates backup-based duplication without connections to the target or to the recovery catalog database instance. A disk backup location containing all the backups or copies for duplication must be available to the destination host.

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 a Backup-Based Duplicate Database

It is important to understand these basic steps and the RMAN database duplication process.

If you are using the Enterprise Manager interface, wizards can perform most steps for you. If you are creating a duplicate database with the command-line interface, you need to perform the steps manually. You can also use the EM interface as a test or sample, and use the output log as a basis for scripting your own database duplication.

The basic steps for creating a duplicate database are outlined in the slide. More detail is provided in this lesson for some of the steps.

Creating an Initialization Parameter File for the Auxiliary Instance

Specify parameters as follows:

- DB_NAME
  - 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.
- DB_BLOCK_SIZE
  - Specify the same value as set for the target database.

ORACLE

Creating an Initialization Parameter File for the Auxiliary Instance

You must create a text initialization parameter file for the auxiliary instance. The text initialization parameter file must reside on the same host as the RMAN client that you use to execute the DUPLICATE command.

Take note of the requirements for each of the following parameters:

- **DB_NAME:** If the target database and the duplicate database are in the same Oracle home, you must set DB_NAME to a different name. If they are in different Oracle homes, you must ensure that the name of the duplicate database differs from the other names in its Oracle home. Be sure to use the same database name that you set for this parameter when you execute the DUPLICATE command.
- **DB_BLOCK_SIZE:** The block size of the auxiliary database must match the block size of the target database. Specify the same value in the initialization parameter file for the auxiliary database as set in the initialization parameter file for the target database. If the parameter is not set in the initialization parameter file for the target database, do not set it in the auxiliary instance initialization parameter file.

In addition, be sure to verify the settings of all initialization parameters that specify path names. Verify that all specified paths are accessible on the duplicate database host.

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

Specifying New Names for Your Destination

You can use the following techniques to specify new names for data files:

- Include the SET NEWNAME FOR DATAFILE command within a RUN block to specify new names for the data files.
- Use the CONFIGURE AUXNAME command.
  CONFIGURE AUXNAME is an alternative to SET NEWNAME. The difference is that after you configure the auxiliary name the first time, additional DUPLICATE commands reuse the configured settings. In contrast, you must reissue the SET NEWNAME command every time you execute the DUPLICATE command.

  Note: SET NEWNAME replaces CONFIGURE AUXNAME for recovery set data files
- Specify the DB_FILE_NAME_CONVERT parameter with the DUPLICATE command.

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'};
```


Using the SET NEWNAME Clauses

You can use SET NEWNAME to specify the default name format for all data files in a named tablespace and all data files in the database.

The order of precedence for the SET NEWNAME command is as follows:

1. SET NEWNAME FOR DATAFILE and SET NEWNAME FOR TEMPFILE
2. SET NEWNAME FOR TABLESPACE
3. SET NEWNAME FOR DATABASE

Example:

```
RUN
{
SET NEWNAME FOR DATABASE TO '/u01/app/oracle/oradata/dupldb/%b';
DUPLICATE TARGET DATABASE TO dupldb
LOGFILE
GROUP 1 ('/u01/app/oracle/oradata/dupldb/redo01a.log',
'/u01/app/oracle/oradata/dupldb/redo01b.log') SIZE 50M REUSE,
GROUP 2 ('/u01/app/oracle/oradata/dupldb/redo02a.log',
'/u01/app/oracle/oradata/dupldb/redo02b.log') SIZE 50M REUSE,
GROUP 3 ('/u01/app/oracle/oradata/dupldb/redo03a.log',
'/u01/app/oracle/oradata/dupldb/redo03b.log') SIZE 50M REUSE;
}
```

Substitution Variables for SET NEWNAME

Syntax Element	Description
%b	Specifies the file name without the directory path
%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

ORACLE

Substitution Variables for SET NEWNAME

When issuing SET NEWNAME FOR DATABASE or SET NEWNAME FOR TABLESPACE, you must specify substitution variables in the TO <filename> clause to avoid name collisions. Specify at least one of the following substitution variables: %b, %f, and %U. %I and %N are optional variables.

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'
```

ORACLE

Specifying Parameters for File Naming

RMAN generates names for the required database files when you execute the DUPLICATE command. You can control the naming of the files by specifying the following initialization parameters in the auxiliary instance initialization parameter file:

- **CONTROL_FILES:** Specify the names of the control files in this parameter. If you do not set the names via this parameter, the Oracle server creates an Oracle-managed control file in a default control destination. Refer to the SQL CREATE CONTROLFILE command in the SQL Reference manual for specific information.
- **DB_FILE_NAME_CONVERT:** This parameter is used to specify the names of data files for the auxiliary database. It has the format DB_FILE_NAME_CONVERT = '*string1*' , '*string2*', where *string1* is the pattern of the target database file name and *string2* is the pattern of the auxiliary database file name. You can also specify the DB_FILE_NAME_CONVERT parameter as an option to the DUPLICATE DATABASE command.
- **LOG_FILE_NAME_CONVERT:** This parameter is used to specify the names of the redo log files for the auxiliary database. It has the format LOG_FILE_NAME_CONVERT = '*string1*', '*string2*', where *string1* is the pattern of the target database file name and *string2* is the pattern of the auxiliary database file name. You can also use the LOGFILE clause of the DUPLICATE DATABASE command to specify redo log file names.

Specifying Parameters for File Naming (continued)

As an alternative to using the initialization parameters to control the naming of the files, you can use the following techniques to rename the redo log files:

- Use the `LOGFILE` clause of the `DUPLICATE` command.
- Set the Oracle Managed Files initialization parameters: `DB_CREATE_FILE_DEST`, `DB_CREATE_ONLINE_DEST_`*n*, or `DB_RECOVERY_FILE_DEST`.

Starting the Instance in NOMOUNT Mode

- Start the auxiliary instance in NOMOUNT mode.
- Create a server parameter file (SPFILE) from the text initialization parameter file you used to start the instance.

```
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.
```


Starting the Instance in NOMOUNT Mode

After you have created the text initialization parameter file, invoke SQL*Plus to start the auxiliary instance in NOMOUNT mode.

After you invoke SQL*Plus, create a server parameter file (SPFILE) from your text initialization parameter file. You can execute the CREATE SPFILE before or after you have started the instance. You should create the SPFILE in the default location so that you do not need to specify the PFILE option with the DUPLICATE command. RMAN shuts down the auxiliary instance and restarts it as part of the duplication process, so you must specify the PFILE option if you do not use an SPFILE.

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

Ensuring That Backups and Archived Redo Log Files Are Available

The backups needed to restore the data files must be accessible on the duplicate host. You do not need a whole database backup. RMAN can use a combination of full and incremental backups of individual data files during the duplication process.

Archived redo logs required to recover the duplicate database to the desired point in time must also be accessible. The archived redo log files can be backups, image copies, or the actual archived redo logs. The backups or copies can be transferred to the local disk of the duplicate database node or mounted across a network by some means such as network file system (NFS).

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

Allocating Auxiliary Channels

If you do not have automatic channels configured, manually allocate at least one auxiliary channel before issuing the DUPLICATE command. The ALLOCATE AUXILIARY CHANNEL command must be within the same RUN block as the DUPLICATE command.

The channel type specified on the ALLOCATE AUXILIARY CHANNEL command must match the media where the backups of the target database are located.

- If the backups reside on disk, you can allocate more than one channel to reduce the time it takes for the duplication process.
- For tape backups, you can specify the number of channels that correspond to the number of devices available.

The auxiliary instance must be started with the NOMOUNT option and the target database must be mounted or open.

Understanding the RMAN Duplication Operation

When you execute the `DUPLICATE` command, RMAN performs the following operations:

1A. Creates a control file server parameter file for the auxiliary instance (for active and for backup-based duplication with target connection) , *or:*

1B. Restores from backup (for standby database and for backup-based duplication without target connection)

2. Mounts the backup control file
3. For backup-based duplication: Selects the backups for restoring the data files to the auxiliary instance
4. Restores the target data files to the duplicate database
5. Performs incomplete recovery using all available incremental backups and archived redo log files

ORACLE

Understanding the RMAN Duplication Operation

When you execute the `DUPLICATE` command, RMAN performs the operations listed in the slide.

1A. RMAN creates a default server parameter file for the auxiliary instance if the following conditions are true:
- Duplication does not involve a standby database.
- Server parameter files are not being duplicated.
- The auxiliary instance was not started with a server parameter file.

1B. RMAN restores from backup—always for the standby database, and for backup-based duplication without target connection.

2. RMAN mounts the restored or the copied backup control file from the active database.
3. For backup-based duplication: RMAN uses the RMAN repository to select the backups for restoring the data files to the auxiliary instance.
4. RMAN restores and copies the duplicate data files.
5. RMAN recovers the data files with incremental backups and archived redo log files to a noncurrent point in time. RMAN must perform database point-in-time recovery, even when no explicit point in time is provided for duplication. Point-in-time recovery is required because the online redo log files in the source database are not backed up and cannot be applied to the duplicate database. The farthest point of recovery of the duplicate database is the most recent redo log file archived by the source database.

Understanding the RMAN Duplication Operation

When you execute the `DUPLICATE` command, RMAN performs the following operations:

6. Shuts down and restarts the auxiliary instance in `NOMOUNT` mode
7. Creates a new control file, which then creates and stores the new `DBID` in the data files
8. Opens the duplicate database with the `RESETLOGS` option
9. Creates the online redo log files for the duplicate database

Note: The database duplication process attempts to resume from the point-of-failure upon reexecution.

Understanding the RMAN Duplication Operation (continued)

6. RMAN shuts down and restarts the database instance in NOMOUNT mode.
7. RMAN creates a new control file, which then creates and stores the new, unique database identifier `DBID` in the data files of the duplicated database.
8. RMAN opens the duplicate database with the `RESETLOGS` option.
9. RMAN creates the online redo log files for the duplicate database.

Note: If the `DUPLICATE DATABASE` command fails, you can re-execute the `DUPLICATE DATABASE` command and the duplication process attempts to resume from the point-of-failure.

Specifying Options for the DUPLICATE Command

You can specify the following options with the DUPLICATE command:

Option	Purpose
SKIP READONLY	Excludes read-only tablespaces
SKIP TABLESPACE	Excludes named tablespaces
TABLESPACE	Includes named tablespaces
NOFILENAMECHECK	Prevents checking of file names
OPEN RESTRICTED	Enables RESTRICTED SESSION automatically

Specifying Options for the DUPLICATE Command

Specify additional options when executing the DUPLICATE command as appropriate.

SKIP READONLY: Use to exclude read-only tablespace data files.

SKIP TABLESPACE: Use to exclude tablespaces from the target database. You cannot exclude the SYSTEM tablespace or tablespaces containing undo or rollback segments.

TABLESPACE: Use to include tablespaces from the target database.

NOFILENAMECHECK: Use to prevent RMAN from checking whether target database data files with the same name as duplicate database data files are in use. You must specify this option when the target database and duplicate database data files and redo log files use the same names. You would typically use this when you create a duplicate database on a host that has the same disk configuration, directory structure, and file names as the target database host. If you do not specify NOFILENAMECHECK in this situation, RMAN returns an error.

OPEN RESTRICTED: Use to enable RESTRICTED SESSION automatically after the database is opened.

Using Additional DUPLICATE Command Options

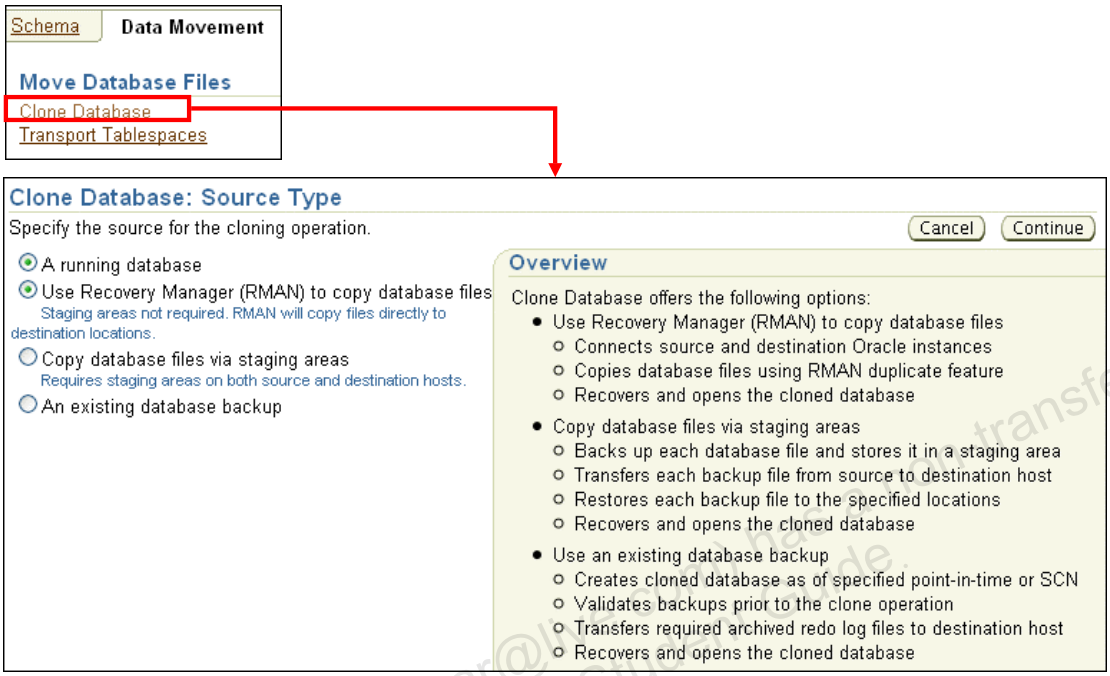
Option	Purpose
NOREDO	Signals RMAN that the application of redo logs should be suppressed during recovery Must be used with targetless DUPLICATE when target database is in NOARCHIVELOG mode at backup time Can also be used to explicitly state that no archived redo log files should be applied
UNDO TABLESPACE	Must be specified when target database is not open and there is no recovery catalog connection so that RMAN does not check the tablespace for SYS-owned objects

Using Additional DUPLICATE Command Options

The following additional options for the DUPLICATE command are introduced with Oracle Database 11*g* Release2:

- **NOREDO:** The NOREDO option is used to signal RMAN that redo logs should not be applied during the recovery phase of the duplication operation. This option should be specified when the database was in NOARCHIVELOG mode at the time of the backup or when the archived redo log files are not available for use during the duplication operation. This option is appropriate if a database that is currently in ARCHIVELOG mode is being duplicated to a point-in-time when it was in NOARCHIVELOG mode.
  If you are performing a targetless DUPLICATE and the database is in NOARCHIVELOG mode, you must use this option to inform RMAN of the database mode. Without a connection to the target database, RMAN cannot determine the mode.
- **UNDO TABLESPACE:** RMAN checks that there are no objects belonging to the SYS user in any of the duplicated tablespaces during non-whole database duplication. The SYSTEM, SYSAUX, and undo segment tablespaces are excluded from this check. However, if the target database is not open and a recovery catalog is not being used during the duplication, RMAN cannot obtain the undo tablespace names. So you must use the UNDO TABLESPACE option to provide the names of undo segment tablespaces.

Using EM to Clone a Database



Using EM to Clone a Database

You can also use Enterprise Manager (EM) to create a duplicate (clone) database. From the EM home page, navigate to Data Movement > Clone database. The screenshot displays the Clone Database: Source Type page.

You can choose from the following venues for the clone operation:

- **Running instance:** You can specify a running instance to be cloned.
- **Staging area:** A disk area specified on the source and destination hosts. A backup is created and stored here, then put in the destination staging area, and read from on that destination host to create the clone database.
- **Existing backup:** If you already have a backup that reflects the database in the state you want it cloned, you can use that.

Quiz

Select all statements that are true about database duplication:

1. You can duplicate a database with or without connection to the auxiliary instance.
2. You can duplicate a database with or without connection to the recovery catalog.
3. You can duplicate a database with or without target connection.
4. You can duplicate a database only when you have RMAN backups.
5. You always have to manually re-create control files on the auxiliary instance.

Answer: 2, 3

Summary

In this lesson, you should have learned how to:

- List the purposes of creating a duplicate database
- Choose a technique for duplicating a database
- Duplicate a database with RMAN
- Use an RMAN backup to duplicate a database
- Duplicate a database based on a running instance

Practice 20 Overview: Duplicating a Database

This practice covers cloning a database and using utilities to complete the setup of a functioning duplicated database.