

Oracle Database 10g: Administrator II

Mega Guide

Prepare With Confidence

This PrepLogic Mega Guide was written by certified subject matter experts and published authors to provide you accurate, in-depth exam coverage. All exam objectives are covered in detail, giving you the knowledge and confidence you need to pass your exam.

**PrepLogic***Be Prepared. Be Confident. Get Certified.*

Backup and Recovery

Terminology

Understanding Oracle Corp's backup and recovery terminology may appear too basic to the uninitiated. But, in fact, if you thoroughly understand Oracle's backup and recovery terminology you are well on the way to understanding how backup and recovery works for Oracle databases.

Oracle backups may be consistent (with all internal System Change Numbers or SCNs synchronized). This means the database, its control files and datafile headers are all in sync and represent a single, common transactional state of the database. Consistent backups (cold backups) are made with the database down or "offline," so they are also called offline backups.

Inconsistent, or online backups, are made while the database is up (hot backups). Since control file SCNs may vary from datafile header SCNs, this kind of backup is first restored (copied in from the flash area or tape backup), then recovered (by applying the logs to bring the database to a point of internal consistency). At the conclusion of recovery, all SCNs match.

A whole database backup includes all datafiles and the control file (online redo logs are never backed up). A partial database backup lacks all the files required to make it a whole database backup. Note that a whole database backup may -- or may not -- be a consistent database backup. In other words, the presence of all files in a database backup does not imply anything about whether the transactional state of all those files is consistent.

A full backup includes all blocks of every datafile included in a whole or partial database backup. This differs from a whole database backup -- the full backup does *not* necessarily mean that the complete database has been backed up. In contrast to a full backup, an incremental backup makes a copy of all data blocks changed since a previous backup. Incremental backups or "incrementals" come in 5 levels, from the level 0 or baseline backup, up through the level 4 backup.

A differential incremental backup backs up only data blocks modified since the most recent backup at the same level or lower. It is the default kind of incremental backup.

A cumulative incremental backup is a type of incremental backup that backs up all the data blocks that have been changed since the most recent backup of the next lower level. If this backup is level **n**, then the cumulative incremental backup will backup the data blocks changed since level **n-1** or lower. A cumulative incremental backup is useful because having it may mean you will be able to apply fewer incremental backups when recovering a database, thus reducing the mean time to recover (MTTR). The MTTR is one of several key measures of backup success and may be part of a service level agreement (SLA) with the user community.

A target database is any database that has been backed up so that it may later be recovered. When a target database becomes lost or damaged in some way, we talk about *recovering the database*. This is a loose way of saying we need to fix some damage that has occurred to data or other files in the database. Be aware that Oracle Corp. also has another, very specific use of the word recover. In Oracle's terminology, to fix a damaged database, you will often perform these 2 steps:

1. Restore data from backup (copy data over from a backup file of some sort, possibly transforming it into the proper format for a database datafile, and putting it into proper location as a datafile within the database).
2. Recover the database by applying redo log data to bring any lost datafiles up to consistency with the rest of the database.

So, as used here, recover specifically means to apply logs to one or more lost datafiles in order to bring them to a point of consistency (to the same System Change Number or SCN as the rest of the database). "Recovering a database" in the general sense means performing the two specific steps of *restoring* and then *recovering* datafiles.

Recall that every Oracle database runs in either of two possible modes:

- Archivelog mode
- Noarchivelog mode

When a database is in *Archivelog* mode, if the current online redo log group members become filled up with redo entries, a log switch occurs. At this point, the Archiver background process (ARCn) steps in and writes the contents of full online redo log file members to archived redo logs. This preserves all redo records.

If a database is in *Noarchivelog* mode, no Archiver process runs, and the full online redo logs are *not* archived by ARCn. During a log switch, redo information is over-written and lost.

Archivelog mode means that no transactions are ever lost. A database or its components can be recovered to any point in time, including up to the last transaction committed (also called a recovery to currency). Databases or their components can be backed up while online and accessible to users. Recoveries of datafiles outside of the critical **SYSTEM** and **UNDO** tablespaces can be performed online, while the rest of the system is up and available to users.

Noarchivelog mode databases require that the entire database be recovered to a specific point in time -- the time at which the database backup was made. This could result in lost transactions. The entire database must be shut down to perform this recovery. (This contrasts to Archivelog mode, where many kinds of recovery permit the database to be up and operating while recovery occurs). You must shut down Noarchivelog mode databases to back them up.

To change the archivelog mode:

1. If using a textual parameter file (**PFILE**), set parameter **LOG_ARCHIVE_START = TRUE**. If using a binary parameter file (**SPFILE**), issue:
SQL> alter system set log_archive_start=true scope=spfile;
2. Shutdown the database
3. **STARTUP MOUNT** (you change the archivelog mode in the **MOUNT** state)
4. Issue **ALTER DATABASE ARCHIVELOG** or **ALTER DATABASE NOARCHIVELOG**
5. Open the database by **ALTER DATABASE OPEN**

Set the archivelog destination(s) by initialization parameter **LOG_ARCHIVE_DEST_n**. Name the archive logs by setting initialization parm **LOG_ARCHIVE_FORMAT**. Determine the reliability level for the writing of the archive logs by init parm **LOG_ARCHIVE_MIN_SUCCEED_DEST**.

Backup and Recovery Alternatives

You can work with backup and recovery through 4 basic mechanisms in Oracle:

1. **User-managed.** This means that you manage backup and recovery yourself, manually, and issue any necessary commands to do it. You hand-code any scripts you want to automate backup and recovery tasks, which use operating system commands such as **copy** or **cp**, and **compress**, for example.
2. **Recovery Manager (RMAN).** RMAN is Oracle Corp's bundled backup and recovery tool. Use RMAN if you work on the command line or write (generate) scripts. This is Oracle Corp's recommended approach.
3. **Enterprise Manager (EM).** EM is the 10g GUI that runs from within a web browser. It includes complete provisions for backup and recovery and is the easiest interface to use. You can learn how it operates by pressing its **Show SQL** button on many screens, which displays the line-oriented SQL or RMAN commands the EM generates to implement the actions you selected on the GUI panels. EM is really a web-based GUI interface into RMAN, when it comes to backup and recovery.
4. **Export Utility.** The Export utility performs backups on logical database entities (like tablespaces or tables). It creates logical backups, as opposed to the physical backups you get when you back up physical database files (by copying the datafiles, control files, and archived redo logs). 10g gives you the new **expdp/impdp** utilities. These have more capabilities than and supersede the older **exp/imp** utilities of Oracle 9i and previous releases. (The older **exp/imp** utilities are still present in 10g if you want them).

The exam emphasizes RMAN and its line commands (the test developers correctly figure that if you know how to accomplish tasks using RMAN's command-line interface (CLI) you can perform those same tasks through the easier EM GUI). The Exam does not much cover user-managed and Export methods outside of the few words we give them here.

This chart gives you a rough comparison of the basic features of the main backup/recovery methods. EM capabilities are basically the same as RMAN, since the EM is really just an RMAN GUI interface when it comes to backup and recovery:

Feature:	User-managed:	RMAN:	Exports:
Password file backups	Yes	Not supported	Not supported
PFILE backups	Yes	Not supported	Not supported
SPFILE backups	Yes	Yes	Not supported
Closed database backups	Yes	Yes	Not supported
Open database backups	Yes	Yes	Yes
Incremental backups	Not supported	Yes	Not supported
Detects corrupt blocks	Not supported	Yes	Yes
Automatic backup file	Not supported	Yes	Yes

Backup catalogs	Not supported	Yes	Yes
Media Manager	Yes	Yes	Yes
Platform independent	Not supported	Yes	Yes

This chart shows that RMAN backs up everything in an Oracle database for you except:

1. Password files
2. Text-based initialization parameter files (**PFILEs**)
3. The Oracle binaries themselves

Most sites that use RMAN easily pick up these items in backups by relying on operating-system level disk backups to back them up. But RMAN itself does not back them up.

RMAN Components and Set-up

Recovery Manager (RMAN) is the Oracle database's backup and recovery feature. Its basic components are:

- **EM's web-based GUI interface** – This GUI into RMAN generates RMAN line commands and scripts as needed to carry out the user's directives.
- **RMAN line commands** – A complete set of over 50 commands managing all aspects of Oracle backup and recovery.
- **RMAN Recovery Catalog** – An Oracle database that provides centralized control for backup and recovery. The RMAN recovery catalog exists as a set of tables and tablespaces that may be placed in their own dedicated "RMAN database," or they may co-reside with other applications within any other Oracle database you choose. The Recovery Catalog is sometimes referred to as the "RMAN Catalog" or just as the "Catalog."
- **Media Management Layer (MML)** - The RMAN MML is a component that provides a generic interface (or application programming interface, or API) into external software that controls tape drives and automated tape libraries (ATLs). Oracle Corp does not provide the tape management software; 3rd parties do, and the MML is the RMAN interface into the 3rd party software.

The RMAN Recovery Catalog is optional but most sites use it since it provides extra RMAN features not otherwise available. The only benefit to *not* using it is that you don't have the overhead of supporting it and you save a few hundred megabytes of disk space. If you do not use an "RMAN Catalog," Oracle stores necessary backup/recovery information for any given target database in that database's control files. This causes the control file to grow in size. The initialization parameter **CONTROL_FILE_RECORD_KEEP_TIME** tells how long RMAN info is kept in the control file. It defaults to 7 days and has a maximum of 365 days.

Control files consist of non-reusable and reusable sections. The reusable sections include both circular reuse records and non-circular reuse records. The circular reuse records contain non-critical information that can be over-written. The non-circular reuse records include information on datafiles and redo logs for the target database.

If you use an RMAN Recovery Catalog it contains:

- Backup and recovery information about target databases, including their datafiles and archive logs
- Physical or structural information about target databases
- Scripts, programs or RMAN commands that can be stored, modified, re-used and potentially applied to multiple target databases

The RMAN Catalog is preferably stored in a separate database than any of its target databases and on a separate server.

To create a Recovery Catalog, login to the database server on which you want it to reside, and follow these steps:

1. Create an RMAN user account:
SQL> connect / as sysdba
SQL> create user rman_user identified by rman_user
default tablespace rman_data
temporary tablespace temp;
2. Grant required permissions to the RMAN user account (including **RECOVERY_CATALOG_OWNER**):
SQL> grant connect, resource, recovery_catalog_owner to rman_user;
3. Start the RMAN utility and connect as the RMAN user id:
c:> rman
RMAN> connect catalog rman_user/rman_user ;
4. Create the recovery catalog by the RMAN **CREATE CATALOG** command. Specify the tablespace you want to store the catalog in:
RMAN> create catalog tablespace rman_data ;
RMAN> exit ;

For any database you want to backup through RMAN (any target database), you must first register that database to RMAN. Another way of saying this is that you will create an incarnation of that target database in the catalog (a reference to a version of that database). These steps show how to start the RMAN utility and connect to the target database, how to connect to the recovery catalog, and how to register the target database with the recovery catalog:

1. c:> rman target / (the slash indicates that you are connecting rman to a target database on the same system you are running rman from)
2. RMAN> connect catalog rman_user/rman_user@orarc ;
3. RMAN> register database ;

orarc is the host server where the recovery catalog resides in this example, as accessed through SQL*Net.