# 10. Backup and Recovery Overview.

*Abstract*: Backup and recovery in an Oracle database environment can be simple or complex depending on the requirements of the business environment the database is supporting. Oracle provides methods for supporting such environments, and each of these methods requires different levels of complexity for backup and recovery operations.

First of all, there are multiple types of failures that may occur in an Oracle database environment. Each of these can result in different types of recovery operations. You must understand these types of failure in order to make the correct recovery decisions.

Once you understand Oracle backup and recovery and the possible types of Oracle database failures, you will be able to create a backup and recovery strategy. When you are determining this strategy, you need to consider a number of issues. First of all, keep in mind that in order for backup and recovery to be successful, everyone from the technical team through management must understand the requirements and the effects of the backup and recovery strategy. After this strategy is agreed upon and in place, a disaster recovery plan can be created based upon this strategy. When you are creating your disaster recovery plan, it's important that you understand the options for high availability, as well as the options for configuring your database for recoverability.

After you have successfully created this plan, the final step is to test it. This lesson takes you step-by-step through the basic principles of backup and recovery: it introduces you to the types of failures that may occur, and to the backup and recovery strategy. In the end, you should be comfortable with your knowledge of what is involved in the Oracle backup, restore, and recovery process. You should understand enough about the different types of failures so that you can identify the appropriate course of action to implement in a recovery situation. This level of understanding will not only make your job as a DBA easier, but it should also make it much more comfortable.

**Contents**

**Objectives:**

- Describe the basics of database backup, restore and recovery.
- List the types of failure that may occur in an Oracle environment.
- Define a backup and recovery strategy.

# 1. Database Backup, Restoration, and Recovery

If you understand and can identify the aspects of the Oracle database that are required for normal operation, you will understand what must be backed up, restored, and recovered.

The Oracle database is made up of a set of physical structures that must be present and consistent for the database to function normally. At a minimum, these physical structures consist of data files, redo logs, control files, and initialization files.

If any of these files are not present, the database may not start up or it may halt during normal operations. All of these files must be backed up on a regular basis to disk, tape, or both. Such a backup can consist of a user-managed backup or Recovery Manager (RMAN)-based backup. A *user-managed backup* consists of any custom backup; such a backup is usually performed in an OS script such as a Unix shell script or the DOS-based batch script. These scripts execute database commands and OS commands to copy the necessary database files to disk or tape. An *RMAN-based backup* is performed by the Oracle recover manager utility, which is part of the Oracle software. RMAN backups are performed by executing standard RMAN commands or scripts.

Both of these backups can be used to restore the necessary database files from disk or tape to the desired location. The *restore* process consists of copying the database files from tape or disk to a desired location so that database recovery can begin.

The *recovery* process consists of starting the database and making it consistent using a complete or partial backup copy of some of the physical structures of the database. Recovery has many options depending on the type of backups that are performed. We discuss the different types of recovery in user-managed and RMAN-based situations in a different lesson.

# 2. Types of Failure in Oracle Environments

There are two major categories of database failures: *non-media failures* and *media* (*disk*) *failures*. Non-media failures consist of four types of failures, which are typically less critical in nature. Media failures have only one type of failure, which is generally more critical in nature—the inability to read or write from a database file.

## *2.1. Non-Media Failures*

This type of failure is made up of statement failures, process failures, instance failures, and user errors, and it is almost always less critical than a media failure. In most cases, statement, process, and instance failures are automatically handled by Oracle and require no DBA intervention. User error can require a manual recovery performed by the DBA.

*Statement failure* consists of a syntax error in the statement, and Oracle usually returns an error number and description.

*Process failure* occurs when the user program fails for some reason, such as when there is an abnormal disconnection or a termination. The process monitor (PMON) process usually handles cleaning up the terminated process.

*Instance failure* occurs when the database instance abnormally terminates due to a power spike or outage. Oracle handles this automatically upon start-up by reading through the current online redo logs and applying the necessary changes back to the database.

*User error* occurs when a table is erroneously dropped or data is erroneously removed.

## *2.2. Media, or Disk, Failures*

These failures are the most critical. A media failure occurs when the database fails to read or write from a file that it requires. For example, a disk drive could fail, a controller supporting a disk drive could fail, or a database file could be removed, overwritten, or corrupted. Each type of media failure that occurs requires a different method for recovery.

The basic steps you should take to perform media recovery are as follows:

1. Determine which files will need to be recovered: data files, control files, and/or redo logs.
2. Determine which type of media recovery is required: complete or incomplete, opened database, or closed database. (You will learn more about these types of recovery in a different lesson.)
3. Restore backups of the required files: data files, control files, and offline redo logs (archived logs) necessary to recover.
4. Apply offline redo logs (archived logs) to the data files.
5. Open the database at the desired point, depending on whether you are performing a complete or an incomplete recovery.
6. Perform frequent testing of the process. Create a test plan of typical failure scenarios.

# 3. Defining a Backup and Recovery Strategy

To create a solid *backup and recovery strategy*, you must keep in mind six major requirements:

- The amount of data that can be lost in the event of a database failure
- The length of time that the business can go without the database in the event of a database failure
- Whether the database can be offline to perform a backup, and if so, the length of time that it can remain offline
- The types of resources available to perform backup and recovery
- The procedures for undoing changes to the database, if necessary
- The cost of buying and maintaining hardware and performing additional backups versus the cost of replacing or re-creating the data lost in a disaster

All of these requirements must clearly be understood before you plan a backup and recovery strategy.

## *3.1. Losing Data in a Database Failure*

The amount of data that can be lost in a failure helps determine the backup and recovery strategy that gets implemented. For instance, if losing a week's worth of data in the event of failure is tolerable, then a weekly backup may be a possible option. On the other hand, if no data can be lost in the event of failure, then weekly backups would be out of the question and backups would need to be performed daily.

## *3.2. Surviving without the Database in a Database Failure*

If the company database were to fail during an outage, how long would it take for the business to be negatively affected? Generally, this question can be answered by management. If all data is entered manually by data entry, the downtime could be relatively long without hurting the business operations. The business could potentially operate normally by generating orders or forms that could be entered into the database later. This type of situation could have minimal effect on the business.

On the other hand, a financial institution that sends and receives data electronically 24 hours a day can't afford to be down for any time at all, and if it were, business operations would most definitely be impaired. The electronic transactions could be unusable until the database was recovered.

After you determine how long the business could survive without a database, you can use the *mean time to recovery* (*MTTR*) to figure out the average amount of time the database could be down if it were to fail. The MTTR is the average time it takes to recover from certain types of failure. You should record each type of failure that is tested so that you can then determine an average recovery time. The MTTR can help determine mean recovery times for different failure scenarios. You can document these times during your testing cycles.

## 3.3. Performing an Offline Backup

To determine whether it is possible to perform a database backup if the database is offline or shut down, you must first know how long the database can afford to be out of commission.

For example, if the database is being used with an Internet site that has national or international access, or if it is being used with a manufacturing site that works two or three shifts across different time zones and has nightly batch processing, then it would have to be available 24 hours a day. In this case, the database would always need to remain online, with the exception of scheduled downtimes for maintenance. In this case, an online backup, or *hot backup*, would need to be performed. This type of backup is done when the database is online or running.

Businesses that don't require 24-hour availability and do not have long batch processing activities in the evening could potentially afford to have the database offline on regular nightly intervals for an offline backup, or *cold backup*. In this scenario, each site should conduct their own backup tests with factors unique to their environment to determine how long it would take to perform a cold backup. If the database downtime is acceptable for that site, then a cold backup could be a workable solution.

## 3.4. Knowing Your Backup and Recovery Resources

The personnel, hardware, and software resources available to the business also affect the backup and recovery strategy. Personnel resources would include at least adequate support from a database administrator (DBA), system administrator (SA), and operator. The DBA would be responsible for the technical piece of the backup, such as user-managed scripts or Recovery Manager (RMAN) scripts. A user-managed backup is an OS backup written in an OS scripting language, such as the Korn shell in the Unix OS. RMAN is an automated tool from Oracle that can perform the backup and recovery process. The SA would be involved in some aspects of the scripting, tape backup software, and tape hardware. The operator might be involved in changing tapes and ensuring that the proper tape cycles are followed.

The hardware resources could include an automated tape library (ATL), a stand-alone tape drive, adequate staging disk space for scripted hot backups and exports, adequate archived log disk space, and third disk mirrors. Many storage subsystem hardware vendors are offering their own third disk mirror options or equivalents. These options create disk copies of 100 gigabytes and greater in just a few minutes. All types of disk subsystems should be at least mirrored or use some form of Redundant Array of Inexpensive Disks (RAID), such as RAID 5, where performance is not compromised.

The software resources could include backup software, scripting capabilities, and tape library software. The Oracle RMAN utility comes with the Oracle9i Server software and is installed when selecting all components of Oracle9i Enterprise Server.

The technical personnel, the DBA and SA at a minimum, are generally responsible for informing the management of the necessary hardware and software to achieve the desired recovery goals.

*NOTE*: RAID is essentially fault tolerance that protects against individual disk crashes. There are multiple levels of RAID. RAID 0 implements disk striping without redundancy. RAID 1 is standard disk mirroring. RAID 2–5 offer some form of parity-bit checking on separate

disks. RAID 5 has become the most popular in recent years, with many vendors offering their own enhancements to RAID 5 for increased performance. RAID 0 + 1 has been a longtime fault-tolerance and performance favorite for Oracle database configurations. This is due to redundancy protection and strong write performance. However, with the new RAID 5 enhancements (performed by some storage array vendors to include large caches or memory buffers), the write performance has improved substantially. RAID 0 + 1 and RAID 5 both can be viable configurations for Oracle databases.

## 3.5. Undoing Changes to the Database

There are three primary ways of undoing changes to the database; one is a manual approach, the other two methods use Oracle features to undo the data.

- Manually—by reexecuting code or rebuilding tables
- Using Oracle LogMiner to recover dropped objects
- Using an Oracle9i feature called Flashback Query

Whether it is possible to undo changes to the database with the manual approach depends on the sophistication of the code releases and the configuration management control for the application in question.

If the configuration control is highly structured with defined release schedules, then undoing changes may not be necessary. A highly structured release schedule would reduce the possibility of data errors or dropped database objects.

On the other hand, if the release schedule tends to be unstructured, the potential for data errors from developers can be higher. It is a good idea to prepare for these issues in any case. A full export can be done periodically, which would give the DBA a static copy of all the necessary objects within a database. Although exports have limitations, they can be useful for repairing data errors because individual users and tables can be extracted from the export file. Additionally, individual tablespace backups can be performed more frequently on high-use tablespaces.

Oracle LogMiner was first introduced in Oracle8/8i. This utility rebuilds data from redo log–generated transactions. LogMiner allows you to rebuild erroneously dropped tables by performing a series of steps that include building an external data dictionary and identifying the transactions that must be reloaded. LogMiner is run by using Procedural Language SQL (PL/SQL) procedures to build the LogMiner utility. Table 1 describes the PL/SQL procedures involved with using LogMiner. The Data Manipulation Language (DML) can be seen in the v$logmnr_contents table after the PL/SQL procedures have been executed.

TABLE 1. LogMiner PL/SQL Procedures

| **PL/SQL Procedure** | **Purpose** |
|---|---|
| sys.dbms_logmnr_d.build | Builds data dictionary |
| dbms_logmnr.add_logfile | Accesses desired redo log file |

| | |
|---|---|
| dbms_logmnr.start_logmnr | Begins using LogMiner session |

Oracle Flashback Query is a feature to Oracle9i. This feature allows a user to access past versions of dictionary tables. This feature works by generating a picture of data as it was in the past by using undo data. This is performed by identifying all data that has been modified since undo was created and retained against the retention policy. Then the corresponding undo data is retrieved.

The Flashback Query feature is performed by executing the PL/SQL dbms_flashback package. Further, Automatic Undo Management must be enabled by setting the init.ora file to the UNDO_MANAGEMENT = AUTO parameter.

There also must be an undo tablespace parameter designated, such as UNDO_TABLESPACE = UNDOTBS. This must be set in the init.ora before the Flashback Query can be used. The parameter that controls the length of retention of the Flashback Query is called UNDO_RETENTION. This parameter may be set with UNDO_RETENTION = n with n being an integer value in seconds.

*NOTE*: Flashback Query cannot query data that is greater than five days old. This is true even if the UNDO_RENTENTION parameter is set to a value greater than five days old.

### *3.6. Weighing the Costs*

Additional hardware is usually needed to perform adequate testing and failover for critical databases. When this additional hardware is unavailable, the risk of an unrecoverable database failure is greater.

The cost of the additional hardware should be weighed against the cost of re-creating the lost data in the event of an unrecoverable database failure. This type of cost comparison will cause the management team to identify the steps necessary to manually re-create lost data, if this can be done.

Once the steps for re-creating lost data are identified and the associated costs are determined, these costs can be compared to the cost of additional hardware. This additional hardware would be used for testing backups and as a system failover if a production server was severely damaged.

## 4. Testing a Backup and Recovery Plan

One of the most important (but also most overlooked) components of the recovery plan is testing. Testing should be done before and after the database that you are supporting is in production. Testing validates that your backups are working, and gives you the peace of mind that recovery will work when a real disaster occurs.

You should document and practice scenarios of certain types of failures so that you are familiar with them, and you should make sure that the methods to recover from these types of failures are clearly defined. At a minimum, you should document and practice the following types of failures:

- Loss of a system tablespace
- Loss of a non-system tablespace
- Loss of a current online redo log
- Loss of the whole database

Testing recovery should include recovering your database to another server, such as a test or development server. The cost of having additional servers available on which to perform testing can be intimidating for some businesses, and it can be one deterrent for adequate testing. Test servers are absolutely necessary, however, and businesses that fail to perform this requirement can be at risk of severe data loss or an unrecoverable situation.

*TIP*: One way that you can test recovery is to create a new development or testing environment and recover the database to a development or test server in support of a new software release. Database copies are often necessary to support new releases of the database and application code prior to moving it to production. RMAN provides the DUPLICATE command in support of this. Manual OS tools in Unix, such as ufsrestore, tar, and cpio from tape or copying from a disk staging area, are often used for scripted backups.

**Real World Scenario**
**Driving Adequate Testing with the Service Level Agreement**

Let's look at this real-life example to determine how much testing is enough for recovery preparation.

Many companies try to reduce implementation and support costs by not allocating sufficient resources for testing. In this particular case, a manufacturing company is trying to determine how much testing is enough to get by.

Up until this point, this company had performed all of its data processing on a mainframe-customized environment, so the customized Oracle database environment is new to them. As a result, when the company switched to an Oracle-based environment, they did not set up a service level agreement with the information technology (IT) department so that their database environment would be maintained properly. Because they lacked this service level agreement from customers of the database, they didn't have the instrument necessary to drive the testing, nor did they have the necessary resources to perform the testing. Their reasoning for this lack of resources was based on the premise that testing is a costly effort, and the results of it may never be needed. In this case, it was hard for the company to justify the expense of resources (IT staff and equipment) to perform testing. What this company was not aware of was that, at a minimum, certain recovery tests should be performed to validate the backup process and to train the database administration staff for certain common failures.

Also, because there was no service agreement with the database customers, there was never a formalized testing strategy to support the recovery of the database. As a result, within one year after the company converted their manufacturing environment to a custom Oracle-based solution, there was an outage. As a result, the company experienced a corrupt online redo log, but because the company had a customized hot backup strategy, a backup was available.

It wasn't immediately apparent to those working on the problem that the failure was due to a corrupt

redo log. As a result, over six hours was wasted by the DBA group before the problem was accurately diagnosed. Once it was, the staff was not sure of the exact type of recovery to perform. Because of this, more time was wasted and more anxiety was created than necessary Finally, the problem was diagnosed with the help of Oracle Support, who determined that the company would need to perform an incomplete recovery prior to the identified corruption point in the online redo log.

With a service level agreement from the database customers and the IT department, a formalized set of expectations would have been set. These expectations could have lead to the proper testing of the database. As a result, most of these problems could have been reduced.

## 5. Summary

Oracle provides numerous options for backup and recovery, which support varying business requirements. These options are founded on some basic principles of database file structures required in the backup, restore, and recovery processes. One of these principles is that the necessary Oracle database file structures are backed up on a regular basis so that if necessary, the restore and recovery of these files can be performed.

There are two major categories of failure: non-media and media based failures. Understanding these types of failure within these categories will allow you to develop a backup and recovery strategy. The result of this strategy is how you will respond in a failure situation to best meet the situation at hand.

It is important to have a solid of understanding of the backup, restore, and recovery processes of an Oracle database. Equally important is having an understanding of the types of failures or problems that could cause you to recover a database. Without understanding the importance and appropriate course of action to take for certain failures, significant down time or mistakes can occur. Making sure the database is open and available for use is one of the most important responsibilities of the DBA.

## References

[1] Oracle9i DBA Fundamentals II.