15

Backup and Recovery: Concepts

ORACLE

Objectives

After completing this lesson, you should be able to:

- Identify the types of failure that can occur in an Oracle database
- Describe instance recovery
- Describe complete and incomplete recovery

DBA Responsibilities

- Protect the database from failure wherever possible
- Increase the mean time between failures (MTBF)
- Protect critical components by using redundancy
- Decrease the mean time to recover (MTTR)
- Minimize the loss of data

The database administrator (DBA) is typically responsible for ensuring that the database is open and available when users need it. To achieve that goal, the DBA (working with the system administrator):

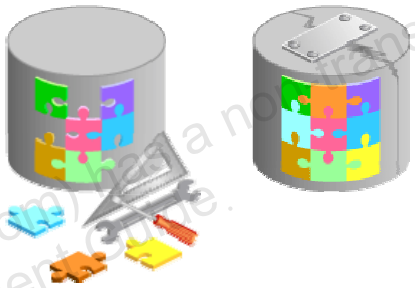
- Anticipates and works to avoid common causes of failure
- Works to increase the mean time between failures (MTBF) that negatively affect availability
- Ensures that hardware is as reliable as possible, that critical components are protected by redundancy, and that operating system maintenance is performed in a timely manner. Oracle Database provides advanced configuration options to increase MTBF, including:
  - Real Application Clusters
  - Oracle Data Guard
- Decreases the mean time to recover (MTTR) by practicing recovery procedures in advance and configuring backups so that they are readily available when needed

- Minimizes the loss of data. DBAs who follow accepted best practices can configure their databases so that no committed transaction is ever lost. Entities that assist in guaranteeing this include:
  - Archive log files (discussed later in this lesson)
  - Flashback technology
  - Standby databases and Oracle Data Guard (discussed in the *Oracle Database 12c: Data Guard Administration* course)

Categories of Failure

Failures can generally be divided into the following categories:

- Statement failure
- User process failure
- Network failure
- User error
- Instance failure
- Media failure



- **Statement failure:** A single database operation (select, insert, update, or delete) fails.
- **User process failure:** A single database session fails.
- **Network failure:** Connectivity to the database is lost.
- **User error:** A user successfully completes an operation, but the operation (dropping a table or entering bad data) is incorrect.
- **Instance failure:** The database instance shuts down unexpectedly.
- **Media failure:** A loss of any file that is needed for database operation (that is, the files have been deleted or the disk has failed).

Statement Failure

Typical Problems	Possible Solutions
Attempts to enter invalid data into a table	Work with users to validate and correct data.
Attempts to perform operations with insufficient privileges	Provide appropriate object or system privileges.
Attempts to allocate space that fail	• Enable resumable space allocation. • Increase owner quota. • Add space to tablespace.
Logic errors in applications	Work with developers to correct program errors.

When a single database operation fails, DBA involvement may be necessary to correct errors with user privileges or database space allocation. DBAs may also need to assist in troubleshooting, even for problems that are not directly in their task area. This can vary greatly from one organization to another. For example, in organizations that use off-the-shelf applications (that is, organizations that have no software developers), the DBA is the only point of contact and must examine logic errors in applications.

To understand logic errors in applications, you should work with developers to understand the scope of the problem. Oracle Database tools may provide assistance by helping to examine audit trails or previous transactions.

Note: In many cases, statement failures are by design and desired. For example, security policies and quota rules are often decided upon in advance. If a user gets an error while trying to exceed his or her limits, it may be desired for the operation to fail and no resolution may be necessary.

User Process Failure

Typical Problems	Possible Solutions
A user performs an abnormal disconnect.	A DBA's action is not usually needed to resolve user process failures. Instance background processes roll back uncommitted changes and release locks.
A user's session is abnormally terminated.	
A user experiences a program error that terminates the session.	Watch for trends. 

User processes that abnormally disconnect from the instance may have uncommitted work in progress that needs to be rolled back. The Process Monitor (PMON) background process periodically polls server processes to ensure that their sessions are still connected. If PMON finds a server process whose user is no longer connected, PMON recovers from any ongoing transactions; it also rolls back uncommitted changes and releases any locks that are held by the failed session.

A DBA's intervention should not be required to recover from user process failure, but the administrator must watch for trends. One or two users disconnecting abnormally is not a cause for concern. A small percentage of user process failures may occur from time to time.

But consistent and systemic failures indicate other problems. A large percentage of abnormal disconnects may indicate a need for user training (which includes teaching users to log out rather than just terminate their programs). It may also be indicative of network or application problems.

Network Failure

Typical Problems	Possible Solutions
Listener fails.	Configure a backup listener and connect-time failover.
Network Interface Card (NIC) fails.	Configure multiple network cards.
Network connection fails.	Configure a backup network connection.

The best solution to network failure is to provide redundant paths for network connections. Backup listeners, network connections, and network interface cards reduce the chance that network failures will affect system availability.

User Error

Typical Causes	Possible Solutions
User inadvertently deletes or modifies data.	Roll back transaction and dependent transactions or rewind table.
User drops a table.	Recover table from recycle bin. Recover table from a backup.

Oracle LogMiner

Users may inadvertently delete or modify data. If they have not yet committed or exited their program, they can simply roll back.

You can use Oracle LogMiner to query your online redo logs and archived redo logs through an Enterprise Manager or SQL interface. Transaction data may persist in online redo logs longer than it persists in undo segments; if you have configured archiving of redo information, redo persists until you delete the archived files. Oracle LogMiner is discussed in the *Oracle Database Utilities* reference.

Users who drop a table can recover it from the recycle bin by flashing back the table to before the drop.

If the recycle bin has already been purged, or if the user dropped the table with the `PURGE` option, the dropped table can still be recovered by using point-in-time recovery (PITR) if the database has been properly configured.

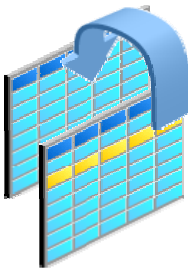
In Oracle Database 12*c*, RMAN enables you to recover one or more tables or table partitions to a specified point in time without affecting the remaining database objects.

Note: Flashback technologies, PITR, and table recovery are discussed in the *Oracle Database 12c: Backup and Recovery Workshop* course and in the *Oracle Database Backup and Recovery User's Guide*.

Flashback Technology

Use Flashback technology for:

- Viewing past states of data
- Winding data back and forth in time
- Assisting users in error analysis and recovery



For error analysis:
Oracle Flashback Query
Oracle Flashback Versions Query
Oracle Flashback Transaction Query

For error recovery:
Oracle Flashback Transaction Backout
Oracle Flashback Table
Oracle Flashback Drop
Oracle Flashback Database

Oracle Database includes Oracle Flashback technology: a group of features that support viewing past states of data—and winding data back and forth in time—without requiring restoring the database from backup. With this technology, you help users analyze and recover from errors. For users who have committed erroneous changes, use the following to analyze the errors:

- **Flashback Query:** View committed data as it existed at some point in the past. The `SELECT` command with the `AS OF` clause references a time in the past through a time stamp or system change number (SCN).
- **Flashback Version Query:** View committed historical data for a specific time interval. Use the `VERSIONS BETWEEN` clause of the `SELECT` command (for performance reasons with existing indexes).
- **Flashback Transaction Query:** View all database changes made at the transaction level.

Possible solutions to recover from user error:

- **Flashback Transaction Backout:** Rolls back a specific transaction and dependent transactions
- **Flashback Table:** Rewinds one or more tables to their contents at a previous time without affecting other database objects

- **Flashback Drop:** Reverses the effects of dropping a table by returning the dropped table from the recycle bin to the database along with dependent objects such as indexes and triggers
- **Flashback Database:** Returns the database to a past time or SCN

Instance Failure

Typical Causes	Possible Solutions
Power outage	Restart the instance by using the `STARTUP` command. Recovering from instance failure is automatic, including rolling forward changes in the redo logs and then rolling back any uncommitted transactions. Investigate the causes of failure by using the alert log, trace files, and Enterprise Manager.
Hardware failure	
Failure of one of the critical background processes	
Emergency shutdown procedures	

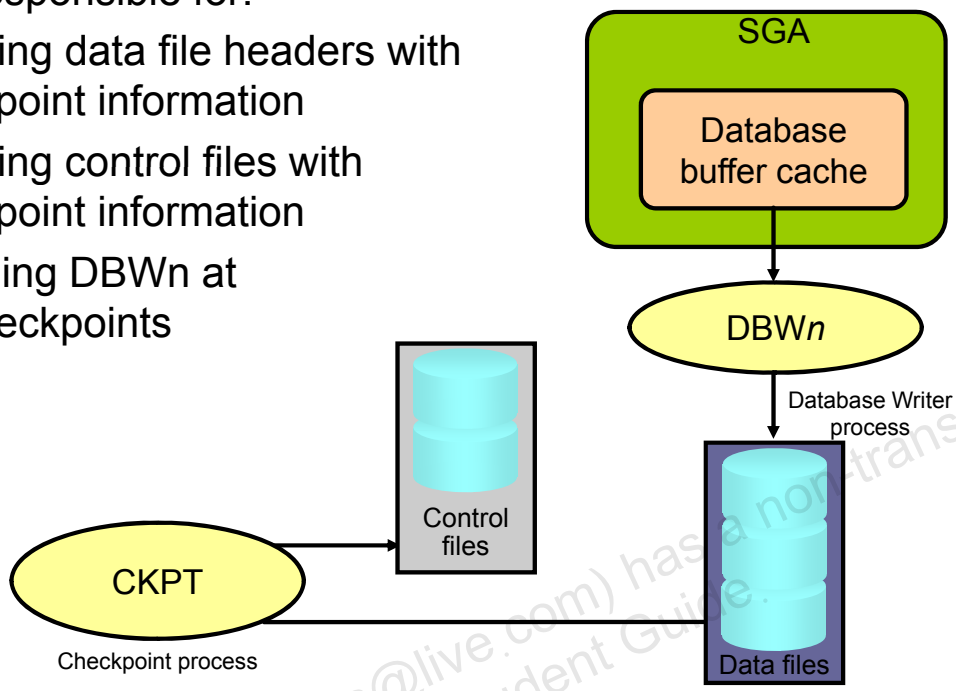
Instance failure occurs when the database instance is shut down before synchronizing all database files. An instance failure can occur because of hardware or software failure or through the use of the emergency `SHUTDOWN ABORT` and `STARTUP FORCE` shutdown commands.

Administrator involvement in recovering from instance failure is rarely required if Oracle Restart is enabled and is monitoring your database. Oracle Restart attempts to restart your database instance as soon as it fails. If manual intervention is required, then there may be a more serious problem that prevents the instance from restarting, such as a memory CPU failure.

Understanding Instance Recovery: Checkpoint (CKPT) Process

CKPT is responsible for:

- Updating data file headers with checkpoint information
- Updating control files with checkpoint information
- Signaling DBWn at full checkpoints



To understand instance recovery, you need to understand the functioning of certain background processes.

Every three seconds (or more frequently), the CKPT process stores data in a control file to document the modified data blocks that DBW*n* has written from the SGA to disk. This is called an “incremental checkpoint.” The purpose of a checkpoint is to identify that place in the online redo log file where instance recovery is to begin (which is called the “checkpoint position”).

In the event of a log switch, the CKPT process also writes this checkpoint information to the headers of data files.

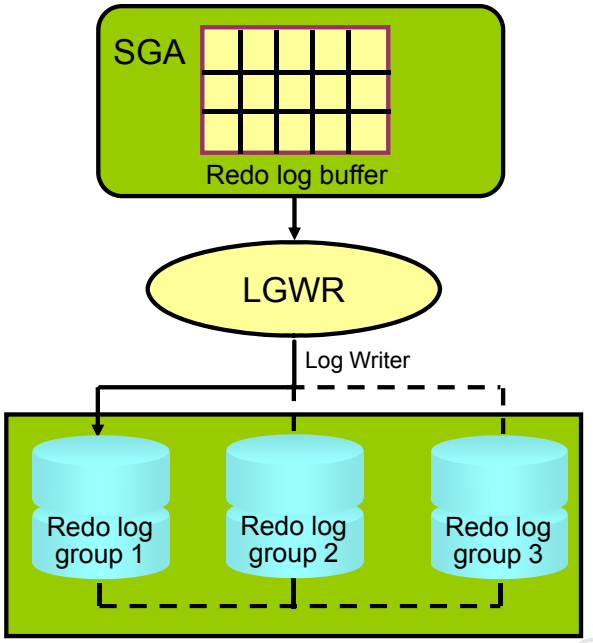
Checkpoints exist for the following reasons:

- To ensure that modified data blocks in memory are written to the disk regularly so that data is not lost in case of a system or database failure
- To reduce the time required for instance recovery (Only the online redo log file entries following the last checkpoint need to be processed for recovery.)
- To ensure that all committed data has been written to data files during shutdown

The checkpoint information written by the CKPT process includes checkpoint position, system change number (SCN), location in the online redo log file to begin recovery, information about logs, and so on.

Note: The CKPT process does not write data blocks to the disk or redo blocks to the online redo log files.

Understanding Instance Recovery: Redo Log Files and Log Writer



Redo log files:

- Record changes to the database
- Should be multiplexed to protect against loss

Log Writer (LGWR) writes:

- At commit
- When one-third full
- Every three seconds
- Before DBW*n* writes
- Before clean shutdowns

ORACLE

Redo log files record changes to the database as a result of transactions and internal Oracle server actions. (A transaction is a logical unit of work consisting of one or more SQL statements run by a user.) Redo log files protect the database from loss of integrity because of system failures caused by power outages, disk failures, and so on. Redo log files should be multiplexed to ensure that the information stored in them is not lost in the event of a disk failure.

The redo log consists of groups of redo log files. A group consists of a redo log file and its multiplexed copies. Each identical copy is said to be a member of that group, and each group is identified by a number. The Log Writer (LGWR) process writes redo records from the redo log buffer to all members of a redo log group until the files are filled or a log switch operation is requested.

It then switches and writes to the files in the next group. Redo log groups are used in a circular fashion.

Best practice tip: If possible, multiplexed redo log files should reside on different disks.

Understanding Instance Recovery

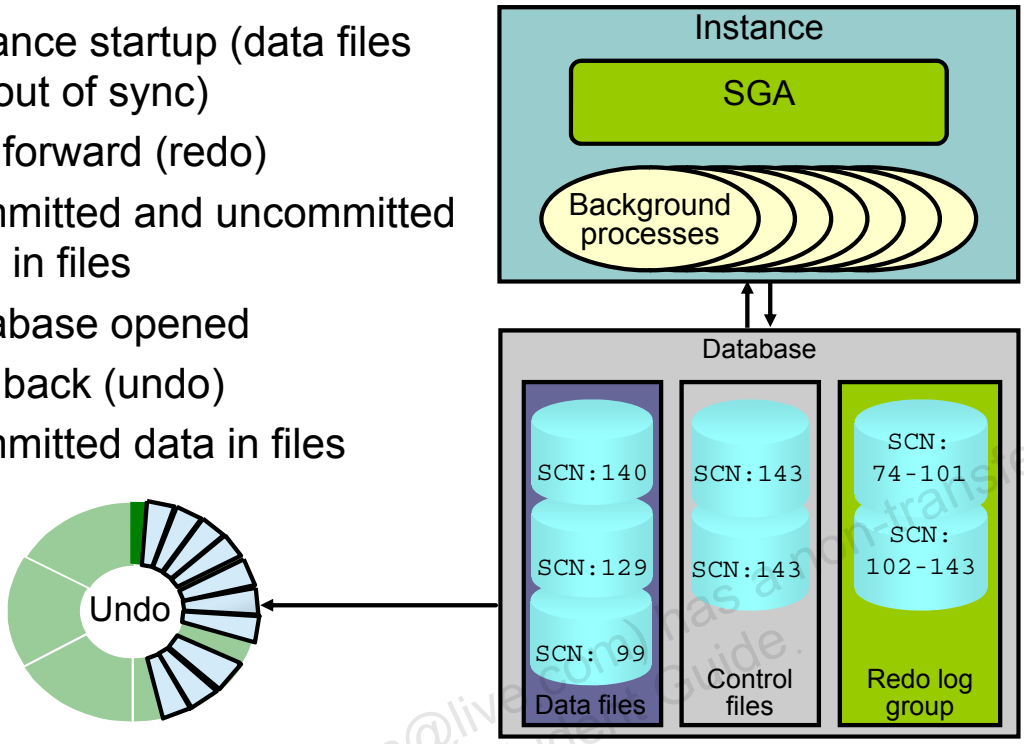
Automatic instance or crash recovery:

- Is caused by attempts to open a database whose files are not synchronized on shutdown
- Uses information stored in redo log groups to synchronize files
- Involves two distinct operations:
  - Rolling forward: Redo log changes (both committed and uncommitted) are applied to data files.
  - Rolling back: Changes that are made but not committed are returned to their original state.

The Oracle database automatically recovers from instance failure. All that needs to happen is for the instance to be started normally. If Oracle Restart is enabled and configured to monitor this database, then this happens automatically. The instance mounts the control files and then attempts to open the data files. When it discovers that the data files have not been synchronized during shutdown, the instance uses information contained in the redo log groups to roll the data files forward to the time of shutdown. Then the database is opened and any uncommitted transactions are rolled back.

Phases of Instance Recovery

1. Instance startup (data files are out of sync)
2. Roll forward (redo)
3. Committed and uncommitted data in files
4. Database opened
5. Roll back (undo)
6. Committed data in files



For an instance to open a data file, the system change number (SCN) contained in the data file's header must match the current SCN that is stored in the database's control files.

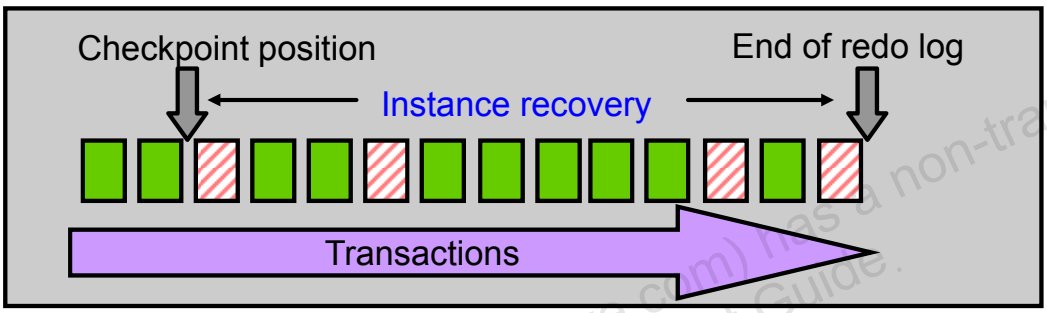
If the numbers do not match, the instance applies redo data from the online redo logs, sequentially "redoing" transactions until the data files are up-to-date. After all data files have been synchronized with the control files, the database is opened and users can log in.

When redo logs are applied, *all* transactions are applied to bring the database up to the state as of the time of failure. This usually includes transactions that are in progress but have not yet been committed. After the database has been opened, those uncommitted transactions are rolled back.

At the end of the rollback phase of instance recovery, the data files contain only committed data.

Tuning Instance Recovery

- During instance recovery, the transactions between the checkpoint position and the end of redo log must be applied to data files.
- You tune instance recovery by controlling the difference between the checkpoint position and the end of redo log.



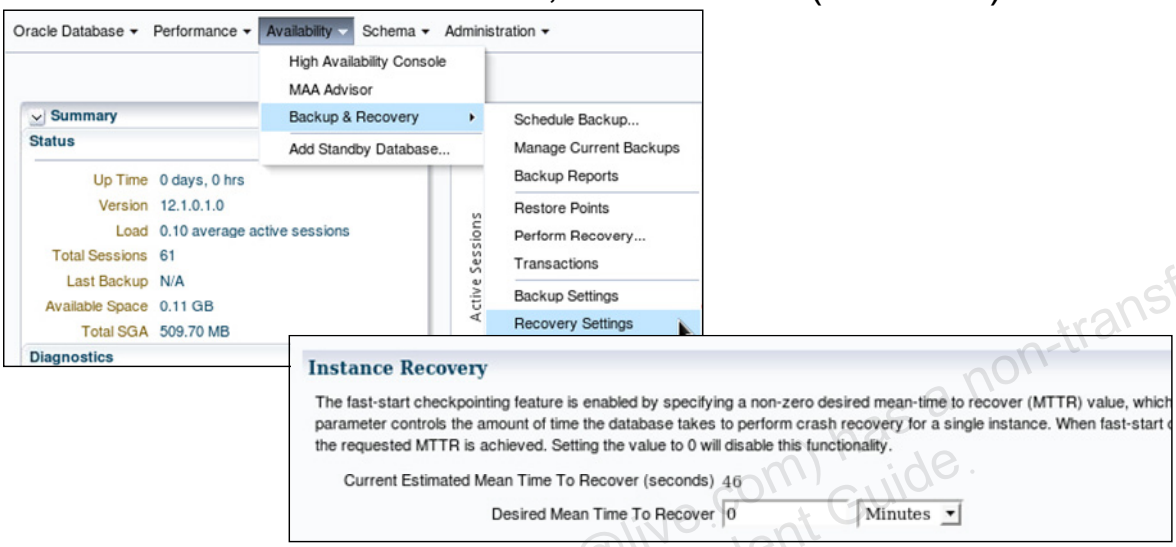
Transaction information is recorded in the redo log groups before the instance returns `commit complete` for a transaction. The information in the redo log groups guarantees that the transaction can be recovered in case of a failure. The transaction information must also be written to the data file. The data file write usually happens at some time after the information is recorded in redo log groups because the data file write process is much slower than the redo writes. (Random writes for data files are slower than serial writes for redo log files.)

Every three seconds, the checkpoint process records information in the control file about the checkpoint position in the redo log. Therefore, the Oracle Database server knows that all redo log entries recorded before this point are not necessary for database recovery. In the graphic in the slide, the striped blocks have not yet been written to the disk.

The time required for instance recovery is the time required to bring data files from their last checkpoint to the latest SCN recorded in the control file. The administrator controls that time by setting an MTTR target (in seconds) and through the sizing of redo log groups. For example, for two redo groups, the distance between the checkpoint position and the end of the redo log group cannot be more than 90% of the smallest redo log group.

Using the MTTR Advisor

- Specify the desired time in seconds or minutes.
- The default value is 0 (disabled).
- The maximum value is 3,600 seconds (one hour).



The `FAST_START_MTTR_TARGET` initialization parameter simplifies the configuration of recovery time from instance or system failure. The MTTR Advisor converts the `FAST_START_MTTR_TARGET` value into several parameters to enable instance recovery in the desired time (or as close to it as possible). Please note, explicitly setting the `FAST_START_MTTR_TARGET` parameter to `0` disables the MTTR Advisor.

The `FAST_START_MTTR_TARGET` parameter must be set to a value that supports the service level agreement for your system. A small value for the MTTR target increases I/O overhead because of additional data file writes (affecting the performance). However, if you set the MTTR target too large, the instance takes longer to recover after a crash.

For assistance in setting the MTTR target by using Enterprise Manager Cloud Control, navigate as follows:

- Performance > Advisors Home > MTTR Advisor
- Availability > Backup & Recovery > Recovery Settings

Media Failure

Typical Causes	Possible Solutions
Failure of disk drive	1. Restore the affected file from backup. 2. Inform the database about a new file location (if necessary). 3. Recover the file by applying redo information (if necessary).
Failure of disk controller	
Deletion or corruption of a file needed for database operation	

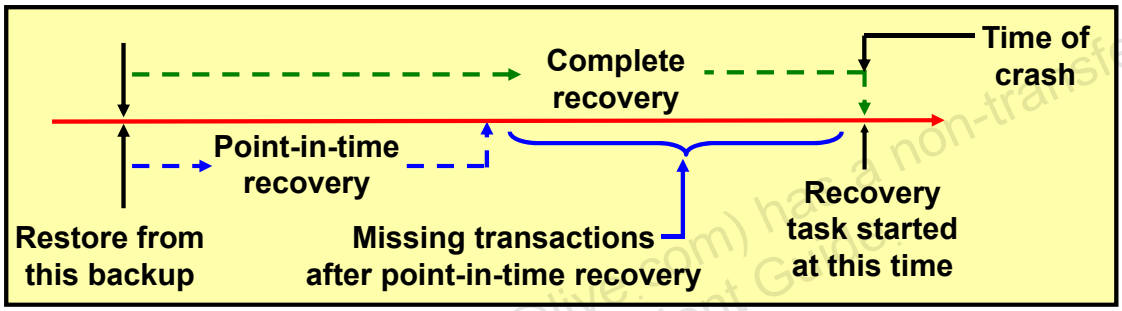
Oracle Corporation defines media failure as any failure that results in the loss or corruption of one or more database files (data, control, or redo log file).

Recovering from media failure requires that you restore and recover the missing files.

Comparing Complete and Incomplete Recovery

Recovery can have two kinds of scope:

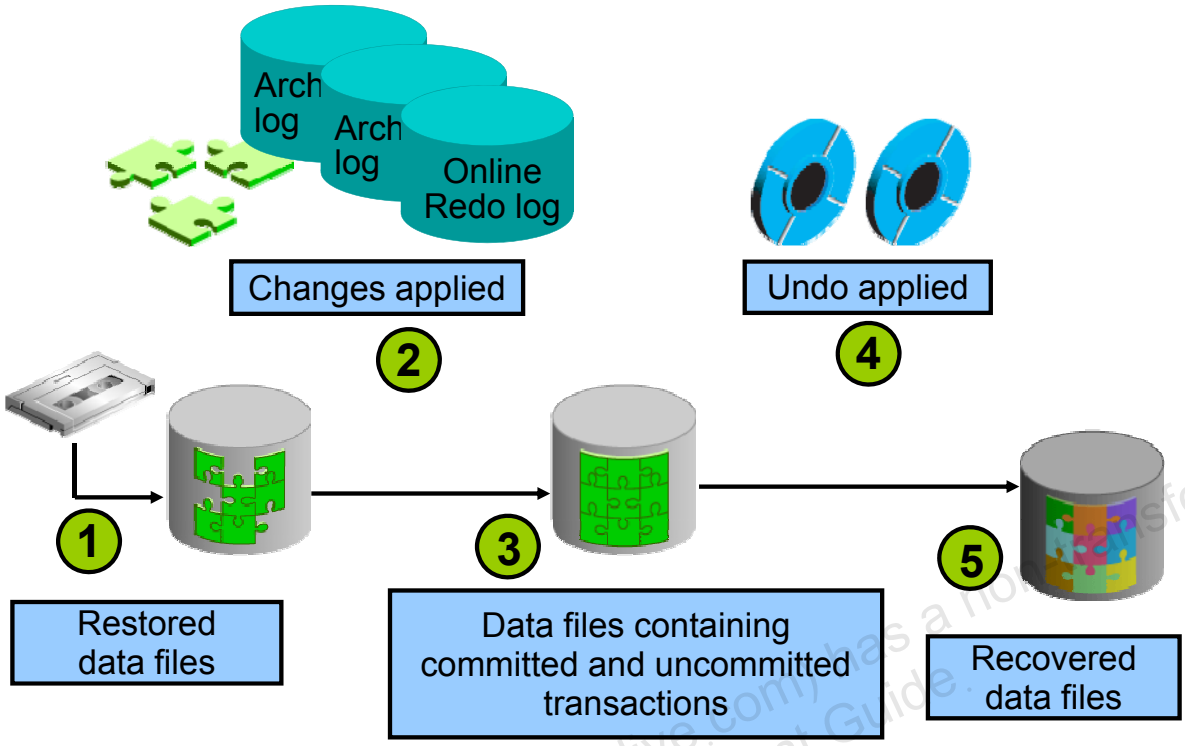
- Complete recovery: Brings the database or tablespace up to the present, including all committed data changes made to the point in time when the recovery was requested
- Incomplete or point-in-time recovery (PITR): Brings the database or tablespace up to a specified point in time in the past, before the recovery operation was requested



When you perform complete recovery, you bring the database to the state where it is fully up-to-date, including all committed data modifications to the present time.

Incomplete recovery, however, brings the database or tablespace to some point of time in the past. This is also known as “point-in-time recovery (PITR).” It means there are missing transactions; any data modifications done between the recovery destination time and the present are lost. In many cases, this is the desirable goal because there may have been some changes made to the database that need to be undone. Recovering to a point in the past is a way to remove the unwanted changes.

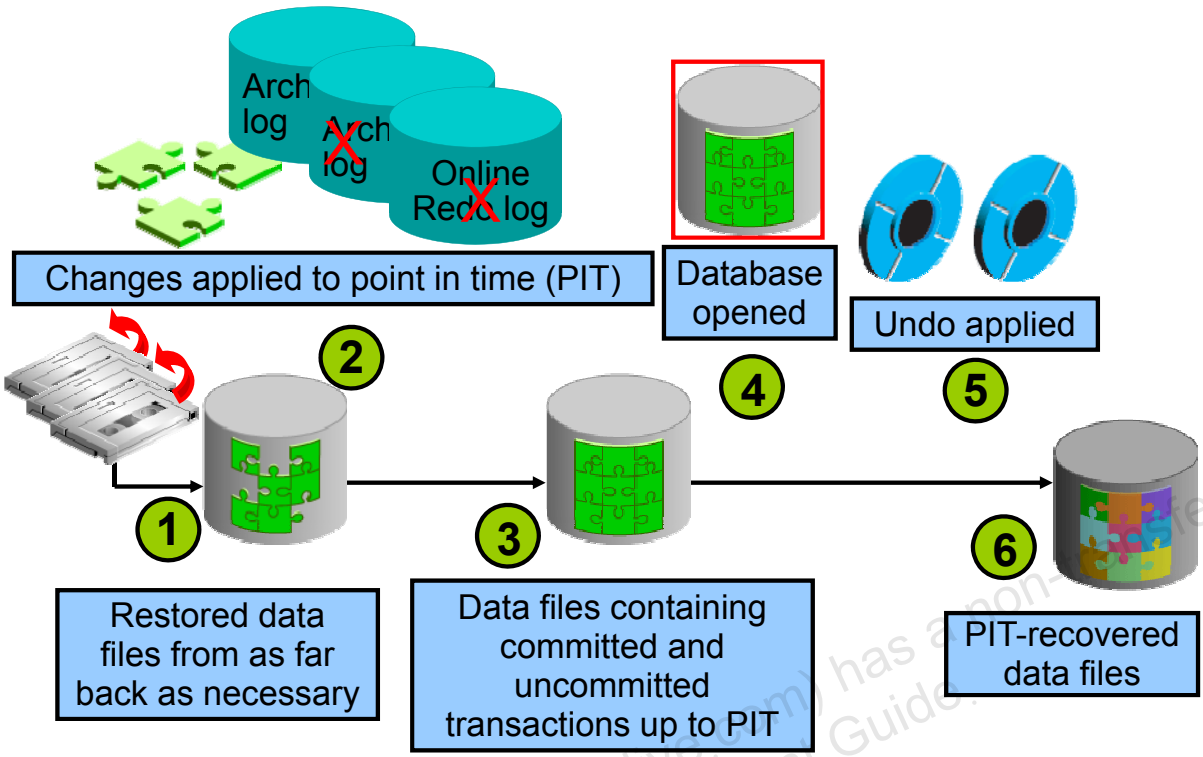
Complete Recovery Process



The following steps describe what takes place during complete recovery:

1. Damaged or missing files are restored from a backup.
2. Changes from incremental backups, archived redo log files, and online redo log files are applied as necessary. The redo log changes are applied to the data files until the current online log is reached and the most recent transactions have been re-entered. Undo blocks are generated during this entire process. This is referred to as rolling forward or cache recovery.
3. The restored data files may now contain committed and uncommitted changes.
4. The undo blocks are used to roll back any uncommitted changes. This is sometimes referred to as transaction recovery.
5. The data files are now in a recovered state and are consistent with the other data files in the database.

Point-in-Time Recovery Process



Incomplete recovery, or database point-in-time recovery (DBPITR), uses a backup to produce a noncurrent version of the database. That is, you do not apply all of the redo records generated after the most recent backup. Perform this type of recovery only when absolutely necessary. To perform point-in-time recovery, you need:

- A valid offline or online backup of all the data files made before the recovery point
- All archived logs from the time of the backup until the specified time of recovery

The steps to perform a point-in-time recovery are as follows:

1. **Restore the data files from backup:** The backup that is used must be from before your target recovery point. This entails either copying files using OS commands or using the RMAN `RESTORE` command.
2. **Use the `RECOVER` command:** Apply redo from the archived redo log files, including as many as necessary to reach the restore point destination.
3. **State of over-recovery:** Now the data files contain some committed and some uncommitted transactions because the redo can contain uncommitted data.
4. **Use the `ALTER DATABASE OPEN` command:** The database is opened before undo is applied. This is to provide higher availability.

5. **Apply undo data:** While the redo was being applied, redo supporting the undo data files was also applied. So the undo is available to be applied to the data files in order to undo any uncommitted transactions. That is done next.
6. **Process complete:** The data files are now recovered to the point in time that you chose.

Oracle Flashback Database is the most efficient alternative to DBPITR. Unlike the other flashback features, it operates at a physical level and reverts the current data files to their contents at a past time. The result is like the result of a DBPITR, including the OPEN RESETLOGS, but Flashback Database is typically faster because it does not require you to restore data files and requires only limited application of redo compared to media recovery.

Oracle Data Protection Solutions

Backup and Recovery Objective	Recovery Time Objective (RTO)	Oracle Solution
Physical data protection	Hours/Days	Recovery Manager Oracle Secure Backup
Logical data protection	Minutes/Hours	Flashback Technologies
Recovery analysis	Minimize time for problem identification and recovery planning	Data Recovery Advisor

Disaster Recovery Objective	Recovery Time Objective (RTO)	Oracle Solution
Physical data protection	Seconds/Minutes	Data Guard Active Data Guard

Oracle provides an appropriate data protection solution depending on your backup and recovery objective and RTO:

- Oracle Recovery Manager (RMAN) is the core Oracle Database software component that manages database backup, restore, and recovery processes.
- Oracle Secure Backup (OSB) is Oracle's enterprise-grade tape backup management solution for both database and file system data.
- Oracle Database Flashback technologies are a set of data recovery solutions that enable human errors to be reversed by selectively and efficiently undoing the effects of a mistake.
- The Data Recovery Advisor provides intelligent database problem identification and recovery capabilities.
- Data Guard and Active Data Guard enable physical standby databases to be open for read access while being kept synchronized with the production database through media recovery.

Quiz

Statement failure is never by design and always requires the DBA to address the issue.

a. True
b. False

Answer: b

Summary

In this lesson, you should have learned how to:

- Identify the types of failure that can occur in an Oracle database
- Describe instance recovery
- Describe complete and incomplete recovery