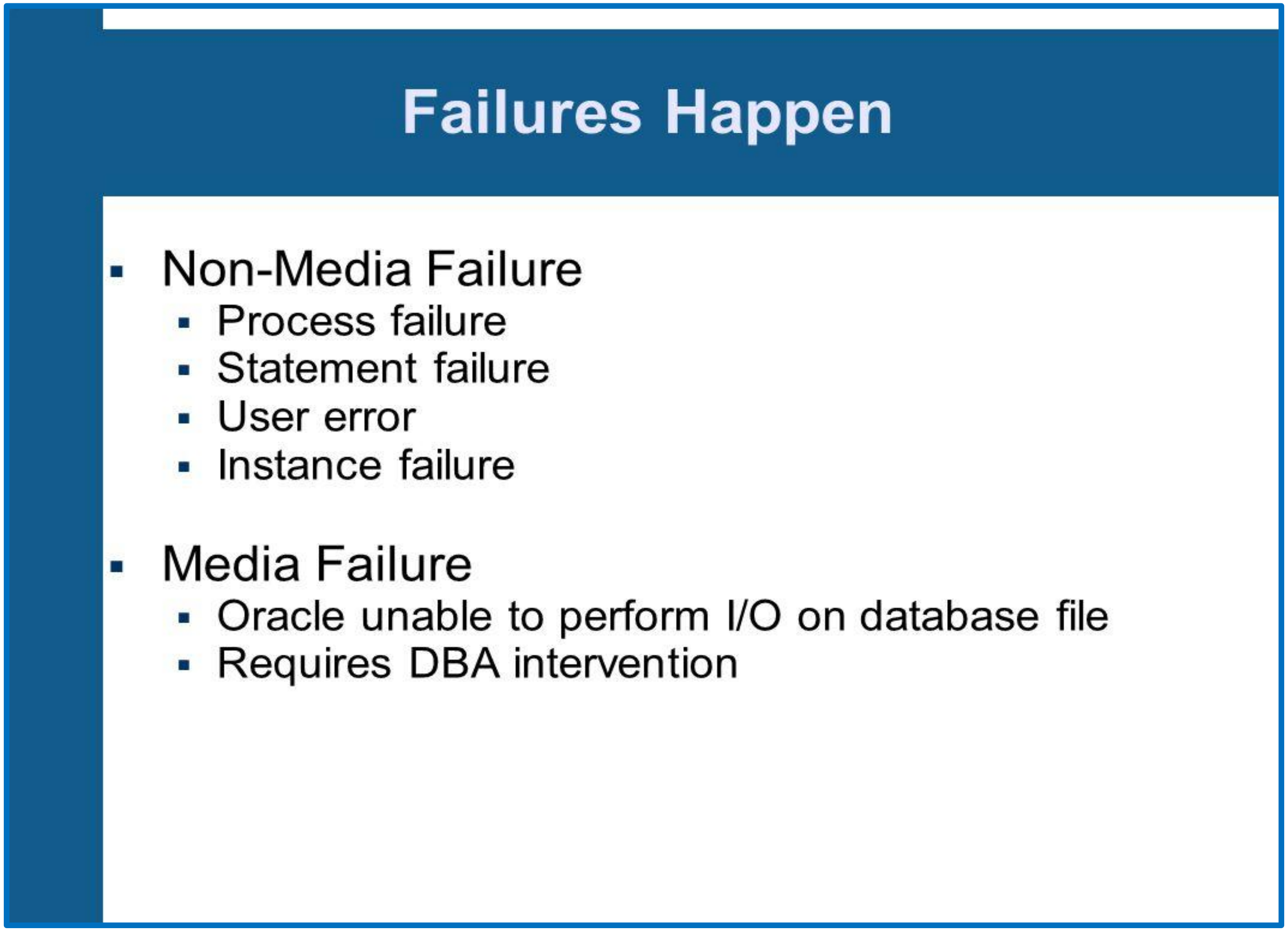
Unit 4: Database Backup, Restore, and Recovery

Backup and Recovery Overview, Database backup, restoration and recovery, defining a backup and recovery strategy, Backup and Recovery options; Data Dump; User-Managed Backup and Recovery; Configuring RMAN; RMAN Backups, Restore and Recovery; High Availability Features; Oracle Data Guard; Flashback operations



Failures Happen

- Non-Media Failure
  - Process failure
  - Statement failure
  - User error
  - Instance failure
- Media Failure
  - Oracle unable to perform I/O on database file
  - Requires DBA intervention

- Database must be protected from failures to protect crucial data from losing. There are two major categories of database failures:
  - non-media failures: less critical in nature (statement failures, process failures, instance failures, and user errors), and
  - media (disk) failures: more critical in nature—the inability to read or write from a database file.

- **Non 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

- **Media or Disk Failure**
  - 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 step to perform media recovery are:**
  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.
  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.

- **Backup and Recovery Overview**
  - Backup and recovery is one of the most important aspect of database administration.
  - A backup is a representative copy of data. This copy can include important parts of a database such as the control file, redo logs and data files.
  - A backup protects data from application error and acts as safeguard against unexpected data loss, by providing a way to restore original data.

Backups are divided into:

- **Physical backups**
  - Physical backup is copying the data files either when the database is up and running (HOT BACKUP) or when the database is shutdown (COLD BACKUP)
  - Physical backups are copies of physical database files. The phrase "backup and recovery" usually refers to the transfer of copied files from one location to another, along with the various operations performed on these files.
- **Logical backups**
  - Logical backup is using SQL statements. Export using exp tool is logical.
  - In contrast, logical backups are those that contains data that is exported using SQL commands and stored in a binary file. Logical backups are used to supplement Oracle server. To recover a restored backup, data is updated using redo records from the transaction log.

Different type of backup strategy may include:

- **The entire database (whole)**
  - A whole database backup includes all data files and at least one control file (remember that all control files within a database are identical).
- **A portion of the database (partial)**
  - Partial database backups may include zero or more tablespaces, zero or more data files, and may or may not include a control file.

Backup type may be:

- **All information from all data files (full)**
  - Full backups make a copy of every data block within the files being backed up that contains data.
- **Only information that has changed since some previous backup (incremental)**
  - Incremental backups make a copy of all data blocks that have changed since some previous backup.

Backup mode may be:

- **Offline (consistent, cold)**
  - Offline backups (also known as consistent backups) are taken while the database is not open. They are consistent because at the time of the backup, the SCN data file headers matches the SCN in the control files.
- **Online (inconsistent, hot)**
  - Online backups (also known as hot or inconsistent backups) are taken while the database is open. The backups are inconsistent because with the database open there is no guarantee that the data files are synchronized with the control files. Inconsistent backups require recovery in order to be used.

Backups may be stored as:

- **Image copies:**
  - Image copies are duplicates of data or archived log files (similar to simply copying the files using operating system commands).
  - Image copies are exact byte-for-byte copies of files.
- **Backup sets:**
  - Backup sets are copies of one or more data or archived log files. With backup sets, empty data blocks are not stored, thereby causing backup sets to use less space on disk or tape.
  - Backup sets can be compressed to further reduce the space requirements of the backup.
  - Backup sets are logical entities produced by the RMAN BACKUP command.

- The advantage of creating a backup as an image copy is improved granularity of the restore operation. With an image copy only the file or files need to be retrieved from tape.
- With backup sets the entire backup set must be retrieved from tape before you extract the file or files that are needed.
- The advantage of creating backups as backup sets is better space usage. Most databases contain 20% or more empty blocks. Image copies back every single data block up, even if the data block is empty. Backup sets significantly reduce the space required by the backup.

- **Recovery**
    - A major responsibility of the database administrator is to prepare for the possibility of hardware, software, network, process, or system failure. If such a failure affects the operation of a database system, you must usually recover the database and return to normal operation as quickly as possible. Recovery should protect the database and associated users from unnecessary problems and avoid or reduce the possibility of having to duplicate work manually.

- **The Physical data Structures used in recovering data are:**
  - Data files and data blocks
  - Control Files
  - Rollback Segments
  - Redo Log Files

Datafiles and Data Blocks:

- A data file is a file which is part of an Oracle database that stores data - including user data and undo data and are collectively known as tablespaces.
- Every Oracle database has one or more physical datafiles. The datafile is divided into smaller units called data blocks. Data blocks are the smallest units of storage that the database can use or allocate.
- The first block of every datafile is header that contains important information such as file size, block size, tablespace, and creation timestamp. Whenever the database is opened, Oracle checks to see that the datafile header information matches the information stored in the control file. If it doesn't match then recovery is necessary.
- Oracle reads data in a datafile during normal operation and stores it in the buffer cache. The Database Writer (DBWR) or DB Writer later writes from buffer cache to disk. The more data that accumulates in memory without being written in disk, the longer the recovery time it will take.

Control files:

- The control file is the file that contains the record of the physical structures of the database and their status. Several types of information stored in the control file related to backup and recovery are:
  - Database information (RESETLOGS SCN and time stamp)
  - Tablespace and datafile records (filenames, datafile checkpoints, read/write status, offline ranges)
  - Information about redo threads (current online redo log)
  - Log records (log sequence numbers, SCN (System change number)range in each log)
  - A record of past RMAN backups
  - Information about corrupt datafile blocks

Control files:

- Every time a user mounts database, its control file is used to identify the datafiles and online redo log files that must be opened for database operation. If physical structure of database changes, a new datafile or redo log file is created, Oracle then modifies the database's control file to reflect the change.
- The control file should be backed up whenever the structure of database changes. Loss of the control file makes recovery from a data loss much more difficult.

Redo Log Files:

- An Oracle database requires at least two online redo log groups, and in each group there is at least one online redo log member, an individual redo log file where the changes are recorded.
- Redo logs record all changes made to a database's data files. Each time data is changed in the database, that change is recorded in the online redo log first, before it is applied to the datafiles.
- At intervals, the database rotates through the online redo log groups, storing changes in the current online redo log.

Redo Log Files:

- Because the redo log contains a record of all changes to the datafiles, if a backup copy of a datafile from some point in time and a complete set of redo logs from that time forward are available, the database can reapply changes recorded in the redo logs, in order to re-construct the datafile contents at any point between the backup time and the end of the last redo log. However, this is only possible if the redo log has been preserved.
- Therefore, preserving the redo logs is a major part of most backup strategies. The first level of preserving the redo log is through a process called archiving. The database can copy online redo log groups that are not currently in use to one or more archive locations on disk, where they are collectively called the archived redo log. Individual files are referred to as archived redo log files. After a redo log file is archived, it can be backed up to other locations on disk or on tape, for long term storage and use in future recovery operations.

There are three basic types of recovery:

- instance recovery
- crash recovery
- media recovery

First two are performed by Oracle automatically at instance startup. Latter requires the user to issue command

- **Instance recovery:**
  - It occurs in an open database when one instance discovers that another instance has crashed. The surviving instance automatically uses the redo log to recover the committed data in database buffers that was lost when the instance failed.

- **Crash recovery:**
  - It occurs when either a single-instance database crashes or all instance of multi-instance database crash. In crash recovery, an instance must first open the database and then execute recovery operation.

- **Media recovery:**
  - It is executed on user's command, usually in response to media failure. In media failure, online or archived redo logs can be used to make a restored backup current or to update it to a specific point in time. Media recovery can restore the whole database, a tablespace, or a datafile and recover them to some non-current time, media recovery is being performed.

Defining a Backup and Recovery Strategy

- **Backup Strategies**
- Before you create an Oracle database, decide how to protect the database against potential media failures. If you do not develop a backup strategy before creating your database, then you may not be able to perform recovery if a disk failure damages the datafiles, online redo log files, or control files.
  1. The golden rule of backup and recovery is: the set of disks or other media that contain the redundancy set should be separate from the disks that contain the datafiles, online redo logs, and control files.
  2. Choosing the Database Archiving Mode
     1. Backing Up a NOARCHIVELOG Database
     2. Backing Up an ARCHIVELOG Database
  3. Multiplexing Control Files, Online Redo Logs, and Archived Redo Logs

Defining a Backup and Recovery Strategy

- **Backup Strategies**

1. Performing Backups Frequently and Regularly
2. Performing Backups Before and After You Make Structural Changes
3. Backing Up Often-Used Tablespaces
4. Performing Backups After Unrecoverable Operations
5. Performing Whole Database Backups After Opening with the RESETLOGS Option
6. Archiving Older Backups
7. Knowing the Constraints for Distributed Database Backups
8. Exporting Data for Added Protection and Flexibility
9. Avoiding the Backup of Online Redo Logs
10. Keeping Records of the Hardware and Software Configuration of the Server

- **Restore and Recovery Strategies**
  1. Testing Backup and Recovery Strategies
  2. Validating Backups and Restores Using RMAN
  3. Planning a Response to Media Failures
  4. Planning a Response to Datafile Block Corruption
  5. Planning the Response to Non-Media Failures

Testing Backup and Recovery Strategies:

- 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.

Parallel Instance Recovery:

- The goal of the parallel recovery feature is to use computed and I/O parallelism to reduce the elapsed time required to perform crash recovery, single-instance recovery, or media recovery. Parallel recovery is most effective at reducing recovery time when several data files on several disks are being recovered concurrently.
- Parallel recovery can speed up both instance recovery and media recovery.

Parallel Recovery using RMAN

- For RMAN, the restore and application of incremental backups are parallelized using channel allocation. The RECOVERY_PARALLELISM parameter determines the number of concurrent processes that participates in recovery. Setting the parameter to 0 or 1 invokes serial recovery. With RESTORE and RECOVER statements, Oracle can automatically parallelize all three stages of recovery.
  1. Restoring Datafiles: When restoring datafiles, the number of channels you allocate in the RMAN recover script effectively sets the parallelism RMAN uses.
  2. Applying Incremental Backups: When you are applying incremental backups, the number of channels you allocate determines the potential parallelism.

Parallel Recovery using RMAN …

3. Applying Redo Logs: Oracle applies redo logs using a specific number of parallel processes as determined by your setting for the RECOVERY_PARALLELISM parameter. The parameter specifies the number of redo application server processes that participates in the instance or media recovery.

- During parallel recovery, one process reads the log files sequentially and dispatches redo information to several recovery processes that apply the changes from the log files to the datafiles.

Parallel Instance Recovery

- Parallel execution can also improve recovery processing. For the parallel instance recovery, the parallel execution processes must be running when the instance starts up. Use PARALLEL_MIN_SERVERS to define the number of parallel execution servers available for parallel recovery and PARALLEL_MIN_SERVERS to set limit on number of parallel execution processes available for recovery

Parallel Media Recovery

- The PARALLEL clause in the RECOVER DATABASE statement determines the degree of parallelism in media recovery. If you do not give value for RECOVERY_PARALLELISM and if it has some non-zero value, it will be used as a default degree of parallelism for the media recovery.

Parallel recovery using Operating System utilities:

You can parallelize instance and media recovery in two ways by:

  - Setting the RECOVERY_PARALLELISM parameter
  - Specifying RECOVER statement options

Setting the RECOVERY_PARALLELISM parameter:

- The RECOVERY_PARALLELISM parameter specifies the number of redo application server processes participating in instance or media recovery. One process reads log files sequentially and dispatches redo information to several recovery processes that apply the changes from the log files to the datafiles. A value of 0 or 1 indicates that recovery is performed serially by one process and the value cannot exceed the value of PARALLEL_MAX_SERVERS parameter.

Specifying RECOVER statement options:

- When you specify RECOVER statement to parallelize instance and media recovery, the allocation of recovery processes to instance is operating system specific. The DEGREE keyword of the PARALLEL clause can either signify the number of processes on each instance of a parallel server or the number of processes to spread across all instances.

Recovering from Non-critical Losses:

Loss of file can be caused by:

- User error
- Application error

There are some files whose loss can be tolerated without going through the restore and recover process, known as "non-critical" loss. A noncritical file loss is one where the database can continue to function.

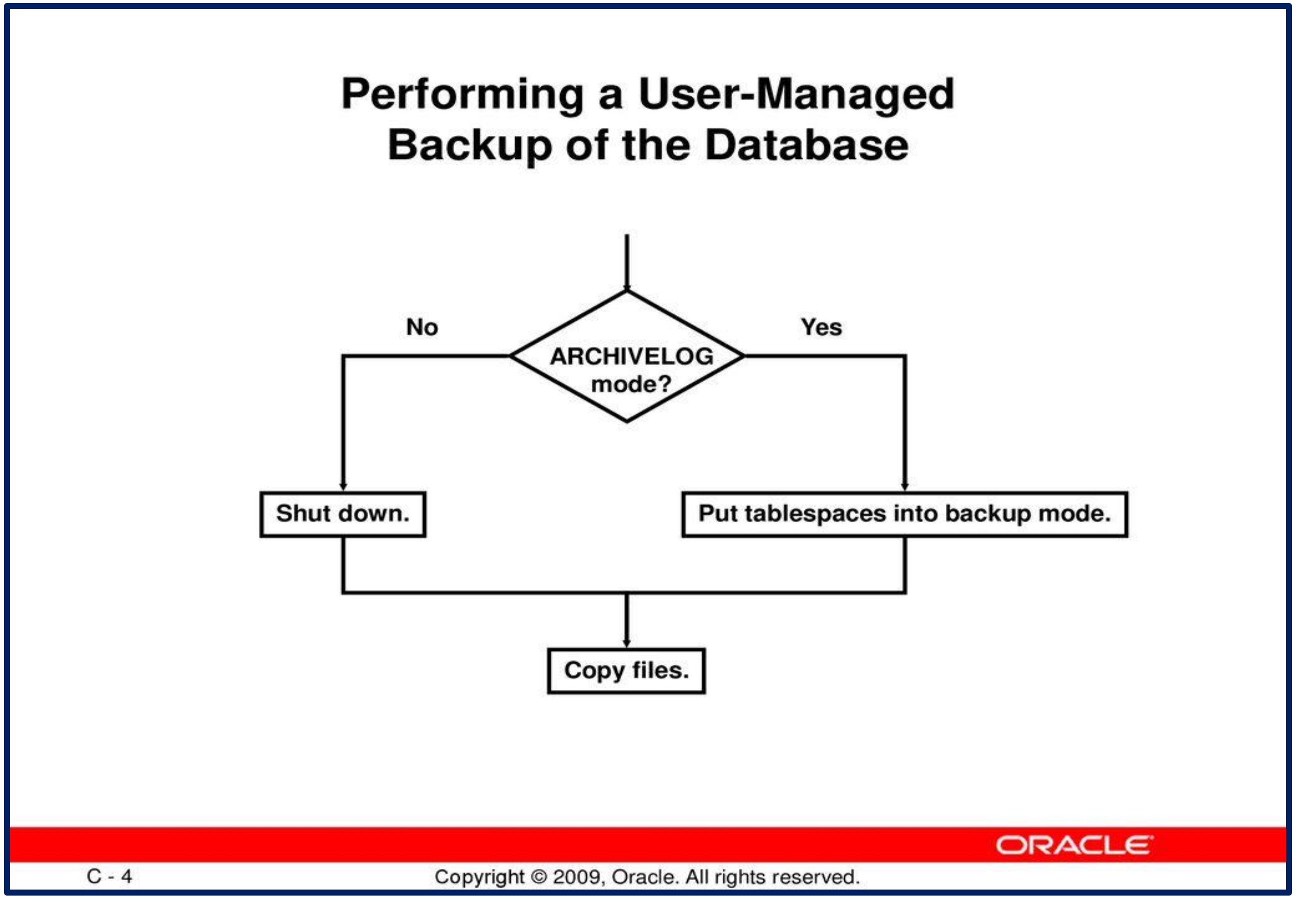
You fix the problem by taking one of these actions:

- Create a new file.
- Rebuild the file.
- Recover the lost or damaged file.

- A **data dump** is the transfer of a large amount of data between two systems, often over a network connection. For example, a database can perform a data dump to another computer or servers on a network, where it could then be utilized by other software applications or analyzed by a person. Some websites receive a data dump from outside systems and publish that data to for visitors to review or use..

User-Managed Backup and Recovery

- User-managed backup and recovery is any strategy in which Recovery Manager (RMAN) is not used as the principal backup and recovery tool. The basic user-managed backup strategy is to make periodic backups of datafiles and archived logs with operating system commands.
- The basic user-managed procedure for recovering from a media failure is as follows:
  - Restore database file backups with operating system commands.
  - Recover restored datafiles with the SQL*Plus RECOVER statement.
  - If the database is closed, then open it for normal use; if it is open, then bring the recovered tablespaces back online.



Performing a User-Managed Backup of the Database

ARCHIVELOG mode?

No

Yes

Shut down.

Put tablespaces into backup mode.

Copy files.

ORACLE

C - 4

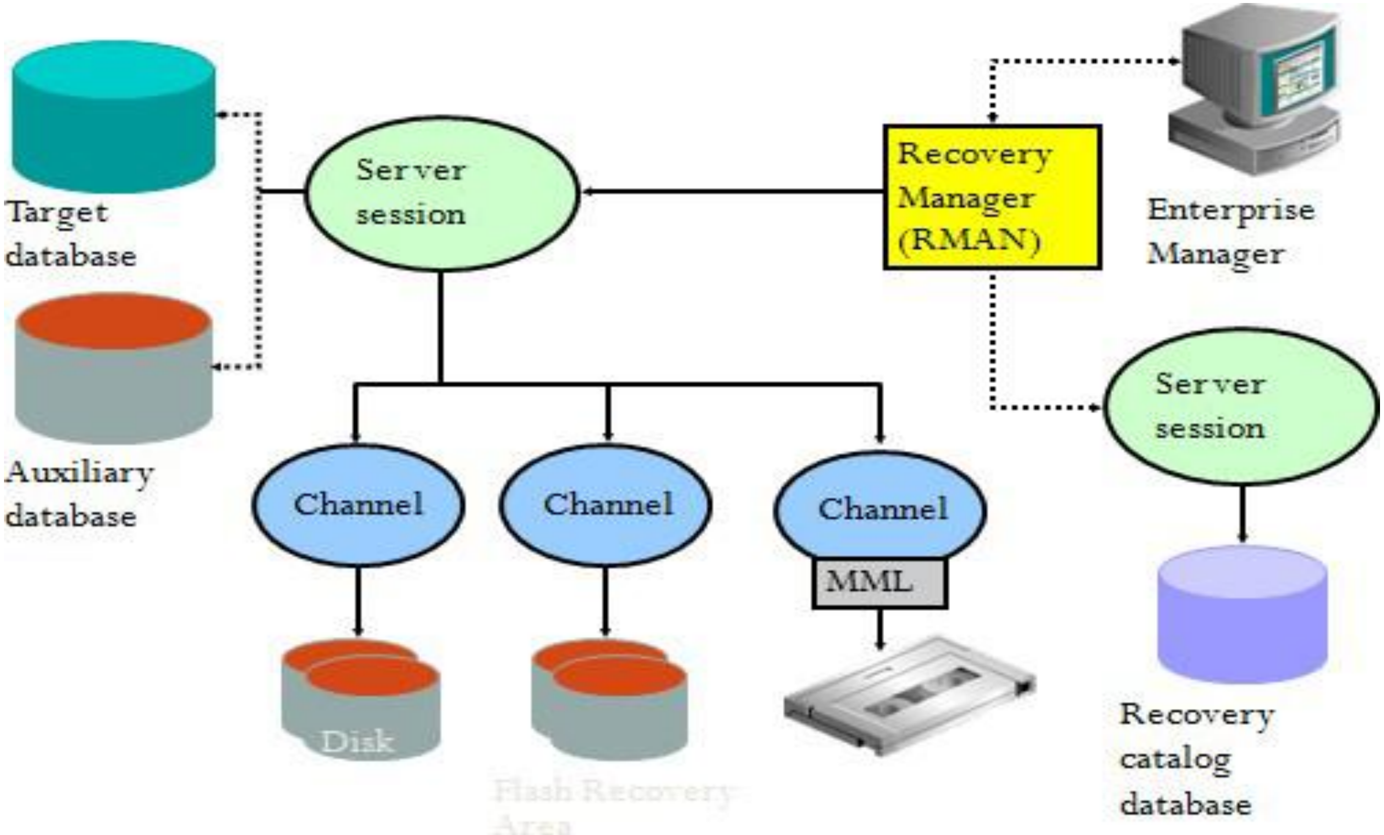
Oracle Recovery Manager (RMAN)

- Recovery Manager (RMAN) is an Oracle utility that can back up, restore, and recover database files. It is a feature of the Oracle database server and does not require separate installation.
- Database administrators (DBAs) can use RMAN to protect data on Oracle databases rather than requiring data backup administrators to initiate them.

Why use RMAN?

- It’s FREE (with your Oracle license).
- Backups can be checked for corruption before it become a problem.
- Minimize downtime after failure.
- Recover single blocks of data.
- Backup directly to tape.
- Creating duplicate instances, including standbys for failover.
- Backups can be taken online without the additional overhead of traditional online backups.
- Automatically includes new datafiles and tablespaces without manual intervention.
- Only backs up used data blocks.
- Allows for compressed backups.
- Works with third-party media management.
- Allows for centralized management and reporting of backups.

RECOVERY MANAGER COMPONENTS:



Target Database

- The target database is the database that RMAN is backing up, restoring, or recovering.
- You can use a single recovery catalog in conjunction with multiple target databases. For example, assume that your data center contains 10 databases of varying sizes. You can use a single recovery catalog located in a different data center to manage the metadata from all of these databases.

RMAN Repository:

- The RMAN repository is the collection of metadata about the target databases that RMAN uses for backup, recovery, and maintenance. RMAN always stores this information in records in the control file. The version of this information in the control file is the authoritative records of RMAN's backups of your database. This is one reason why protecting your control file is an important part of your backup strategy. RMAN can conduct all necessary backup and recovery operations using just the control file to store the RMAN repository information, and maintain all records necessary to meet your configured retention policy.

Media Management Interface:

- To store backups on tape, RMAN requires a media manager. A media manager is a software program that loads, labels, and unloads sequential media such as tape drives used to back up and recover data.

RMAN Channels

- Before you can execute a backup or recovery using RMAN, you must allocate a channel between the backup processes and the operating system. The following operations in RMAN must have at least one channel allocated to operate correctly:
  - BACKUP
  - COPY
  - RESTORE
  - RECOVER
- To allocate a channel, you use the RMAN command ALLOCATE CHANNEL. When you allocate the channel, you also specify the type of device that will be used for the operation that will occur through that channel. Multiple channels can be allocated, allowing for multiple backup sets or file copies to run in parallel by the execution of just a single RMAN command. Each time you issue an ALLOCATE CHANNEL command, a separate connection between the backup processes and the operating system is created.

Automatic Storage Management (ASM):

- Automatic Storage Management (ASM) is an integrated, high-performance database file system and disk manager. ASM is based on the principle that the database should manage storage instead of requiring an administrator to do it. ASM eliminates the need for you to directly manage potentially thousands of Oracle database files.
- ASM groups the disks in your storage system into one or more disk groups each of which comprises of several physical disks that are controlled as a single unit. The physical disks are knows as ASM disks, while the files that reside on the disks are known as ASM files. The locations and names for the files are controlled by ASM.

ASM provides the following benefits:

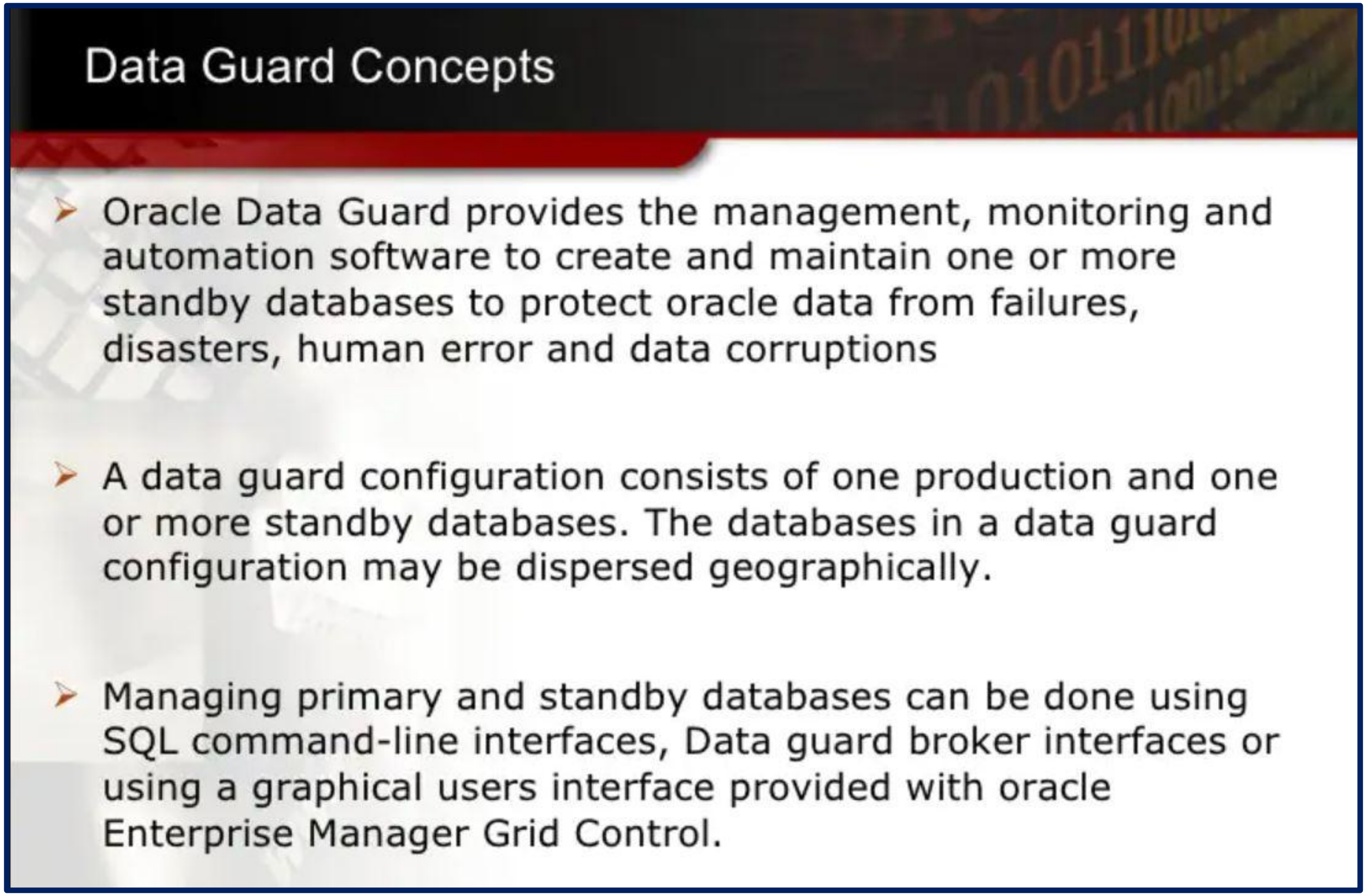
- Striping—ASM spreads data evenly across all disks in a disk group to optimize performance and utilization. This even distribution of database files eliminates the need for regular monitoring and I/O performance tuning.
- Mirroring— Mirroring means keeping redundant copies, or mirrored copies, of each extent of the file, to help avoid data loss caused by disk failures. ASM can increase availability by optionally mirroring any file. ASM mirrors at the file level, unlike operating system mirroring, which mirrors at the disk level. The mirrored copy of each file extent is always kept on a different disk from the original copy. If a disk fails, ASM can continue to access affected files by accessing mirrored copies on the surviving disks in the disk group. ASM supports 2-way mirroring, where each file extent gets one mirrored copy, and 3-way mirroring, where each file extent gets two mirrored copies.

ASM provides the following benefits …

- Online storage reconfiguration and dynamic rebalancing—ASM permits you to add or remove disks from your disk storage system while the database is operating. When you add a disk, ASM automatically redistributes the data so that it is evenly spread across all disks in the disk group, including the new disk. This redistribution is known as rebalancing. It is done in the background and with minimal impact to database performance. When you request to remove a disk, ASM first rebalances by evenly relocating all file extents from the disk being removed to the other disks in the disk group.
- Managed file creation and deletion—ASM further reduces administration tasks by enabling files stored in ASM disk groups to be Oracle-managed files. ASM automatically assigns filenames when files are created, and automatically deletes files when they are no longer needed.

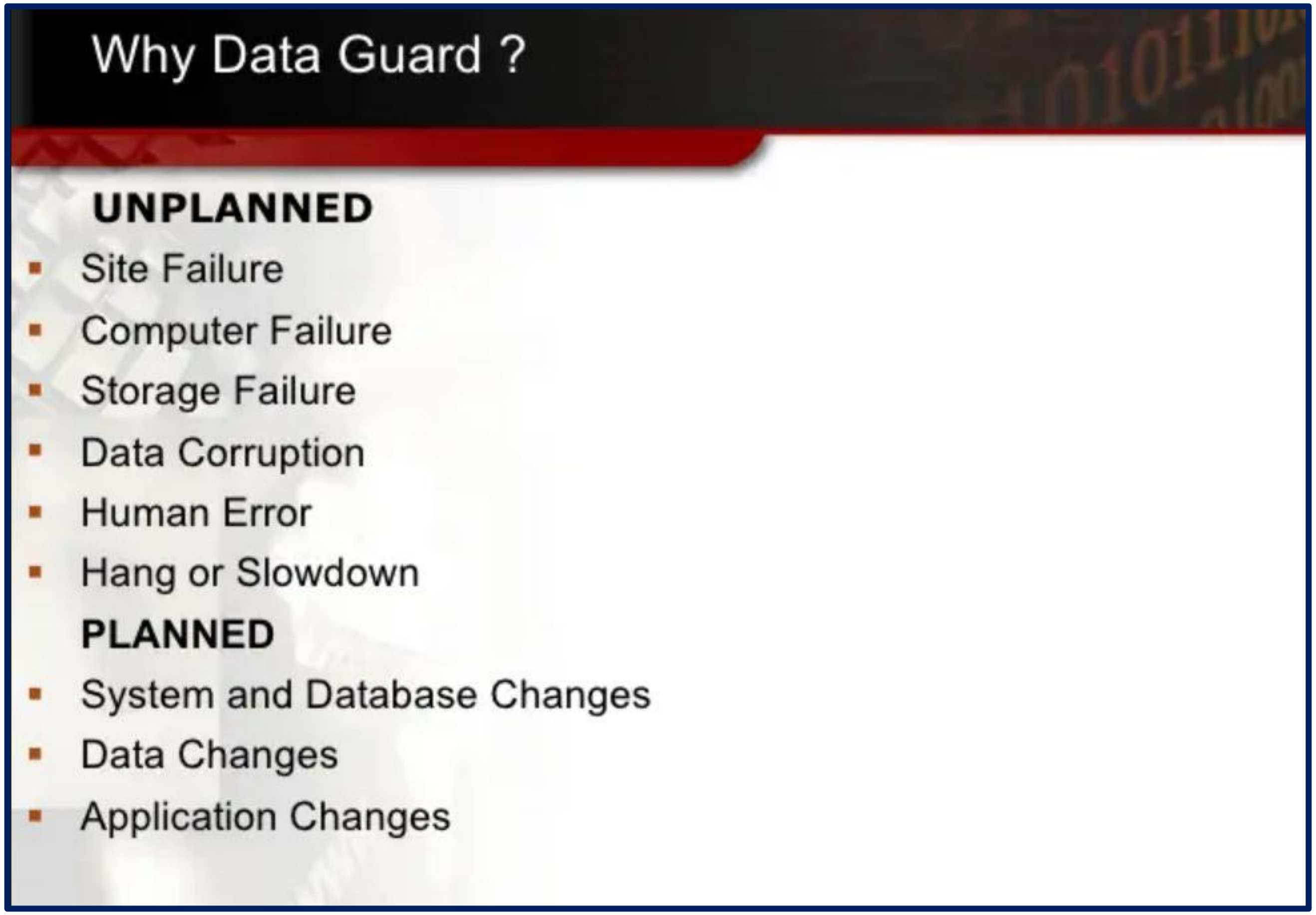
In summary ASM provides following functionalities:

- Manages group of disks, called disk groups.
- Manages disk redundancy within a disk group.
- Provides near-optimal I/O balancing without any manual tuning.
- Enables management of database objects without specifying mount points and filenames.
- Supports large files.



Data Guard Concepts

- Oracle Data Guard provides the management, monitoring and automation software to create and maintain one or more standby databases to protect oracle data from failures, disasters, human error and data corruptions
- A data guard configuration consists of one production and one or more standby databases. The databases in a data guard configuration may be dispersed geographically.
- Managing primary and standby databases can be done using SQL command-line interfaces, Data guard broker interfaces or using a graphical users interface provided with oracle Enterprise Manager Grid Control.



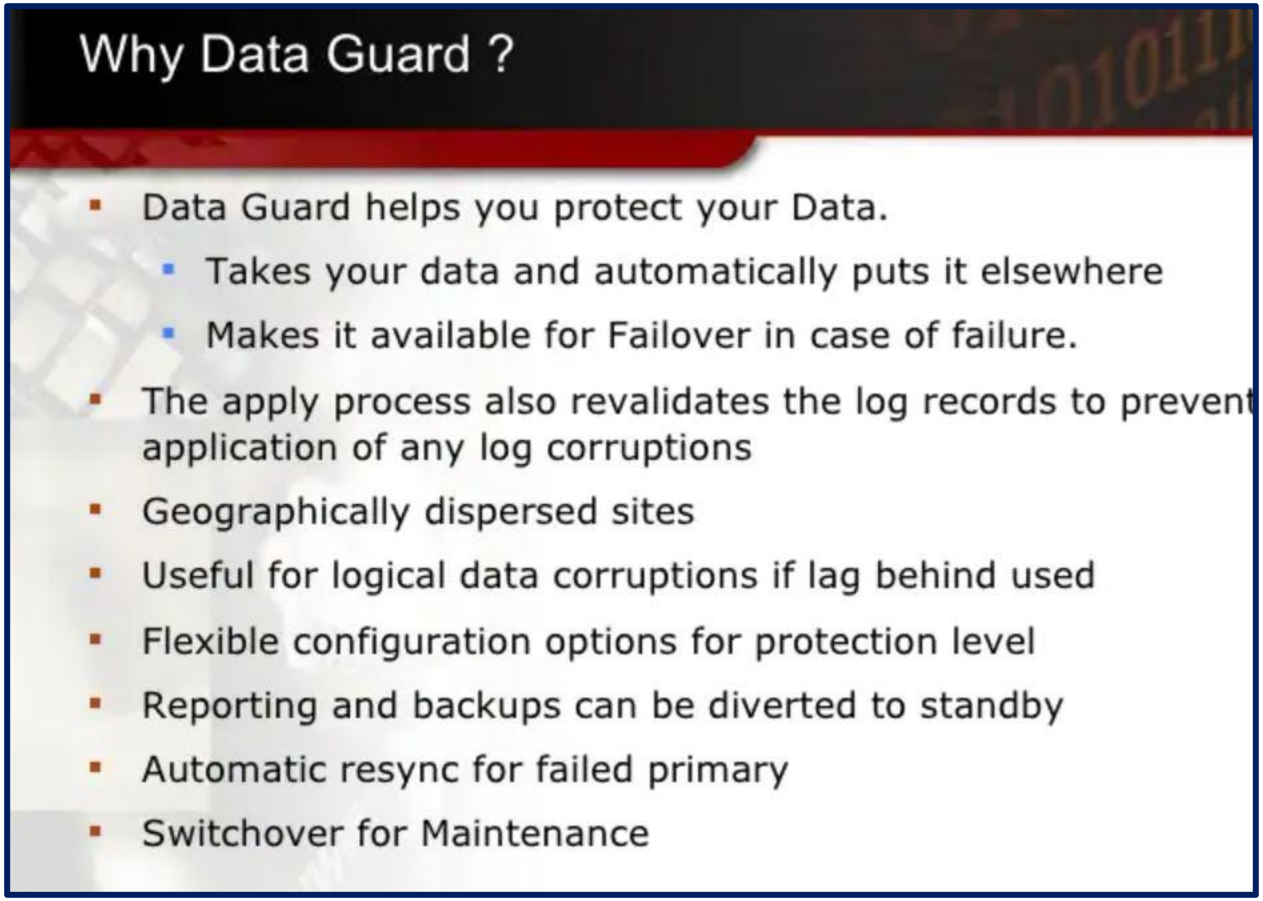
Why Data Guard ?

UNPLANNED

- Site Failure
- Computer Failure
- Storage Failure
- Data Corruption
- Human Error
- Hang or Slowdown

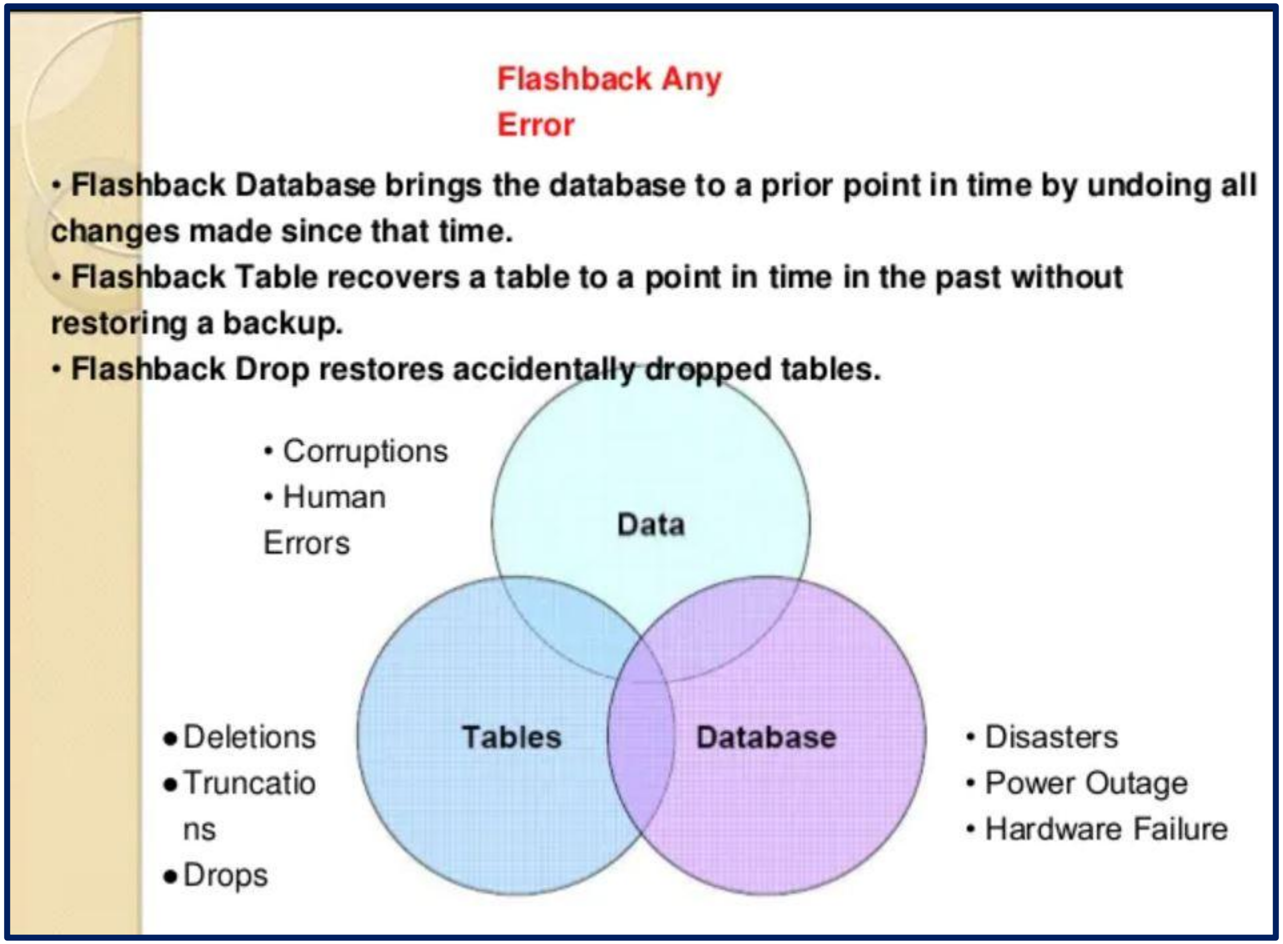
PLANNED

- System and Database Changes
- Data Changes
- Application Changes



Why Data Guard ?

- Data Guard helps you protect your Data.
  - Takes your data and automatically puts it elsewhere
  - Makes it available for Failover in case of failure.
- The apply process also revalidates the log records to prevent application of any log corruptions
- Geographically dispersed sites
- Useful for logical data corruptions if lag behind used
- Flexible configuration options for protection level
- Reporting and backups can be diverted to standby
- Automatic resync for failed primary
- Switchover for Maintenance



Flashback Any Error

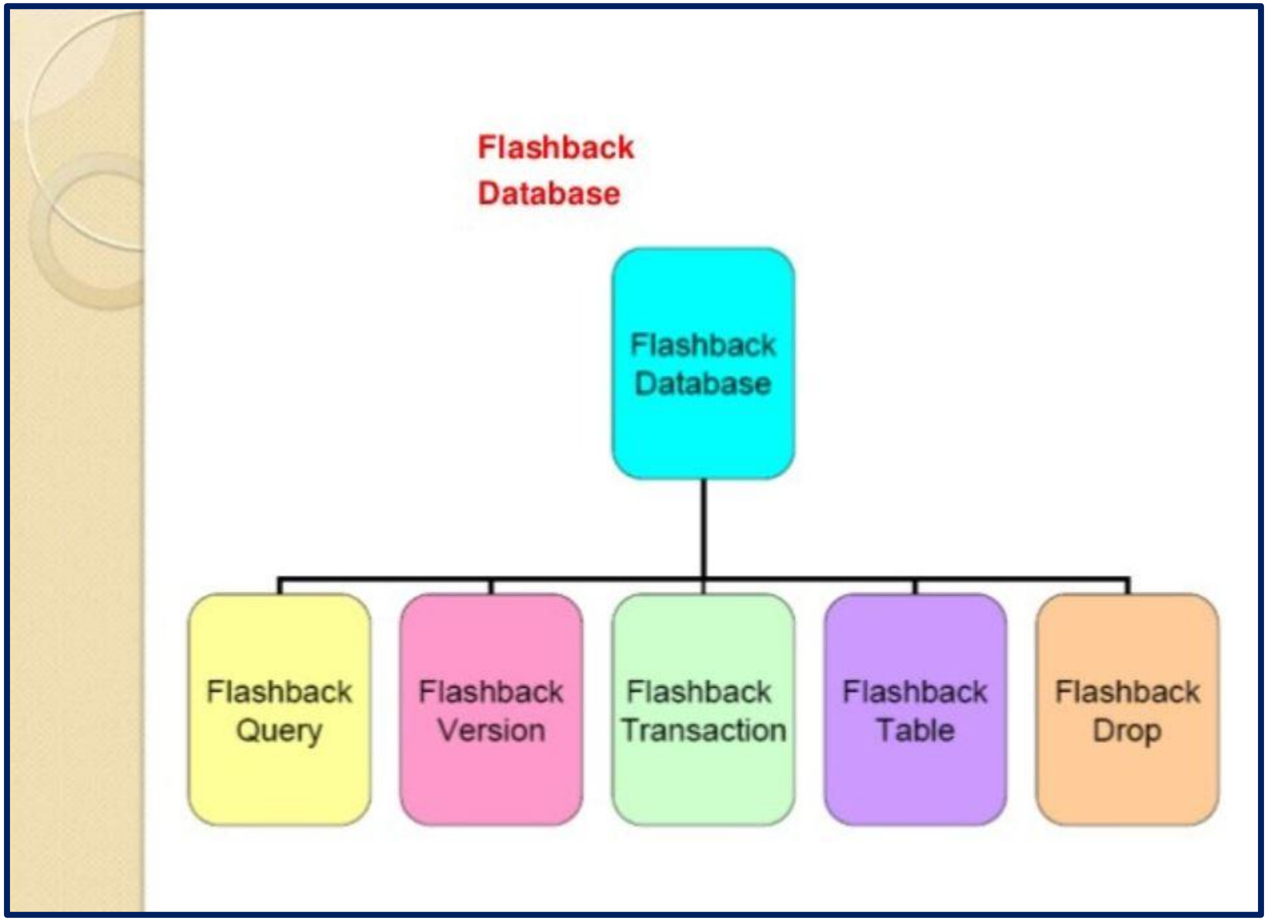
- Flashback Database brings the database to a prior point in time by undoing all changes made since that time.
- Flashback Table recovers a table to a point in time in the past without restoring a backup.
- Flashback Drop restores accidentally dropped tables.

- Corruptions
- Human Errors

Data
Tables
Database

- Deletions
- Truncatio ns
- Drops

- Disasters
- Power Outage
- Hardware Failure



Flashback Database

Flashback Database

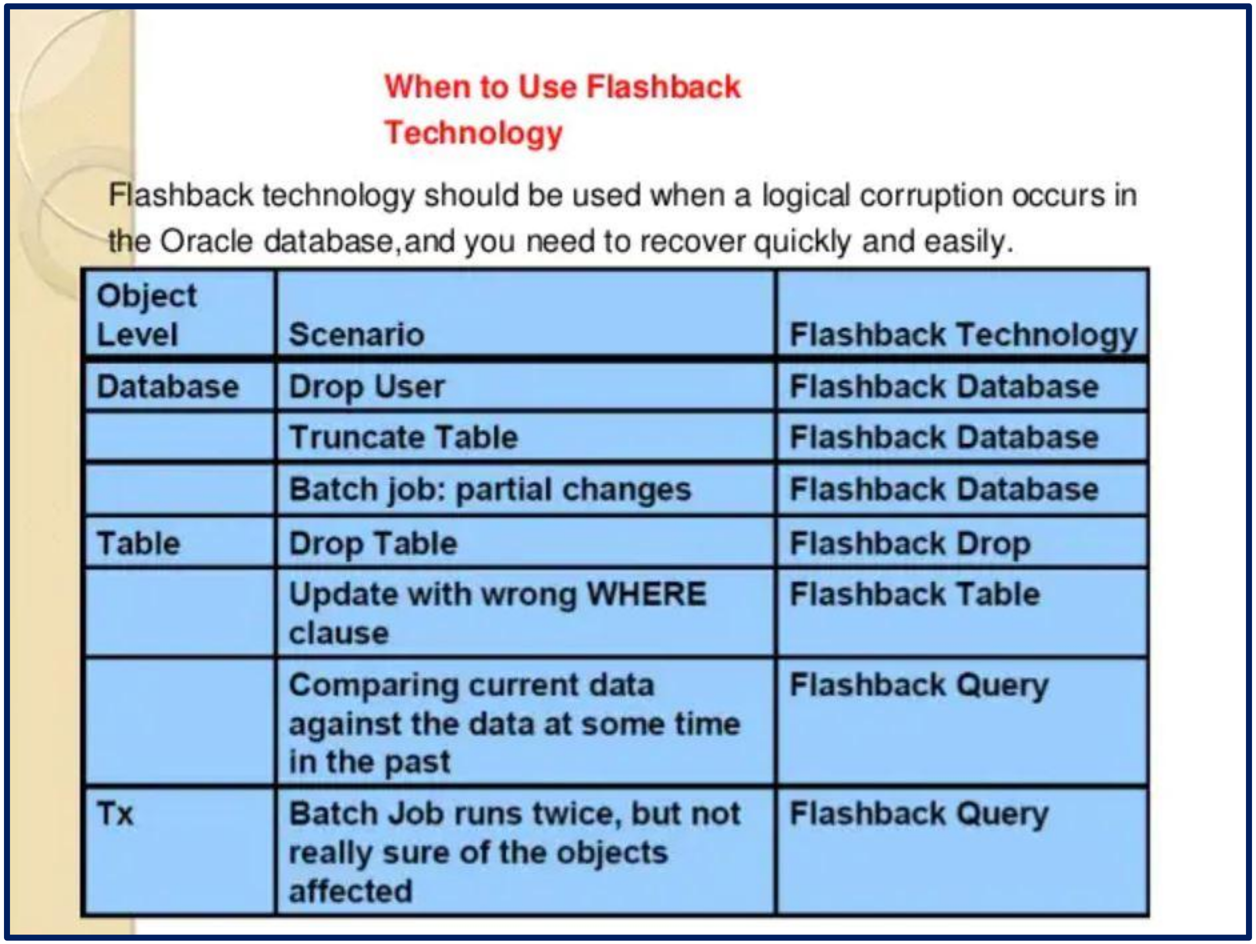
Flashback Query

Flashback Version

Flashback Transaction

Flashback Table

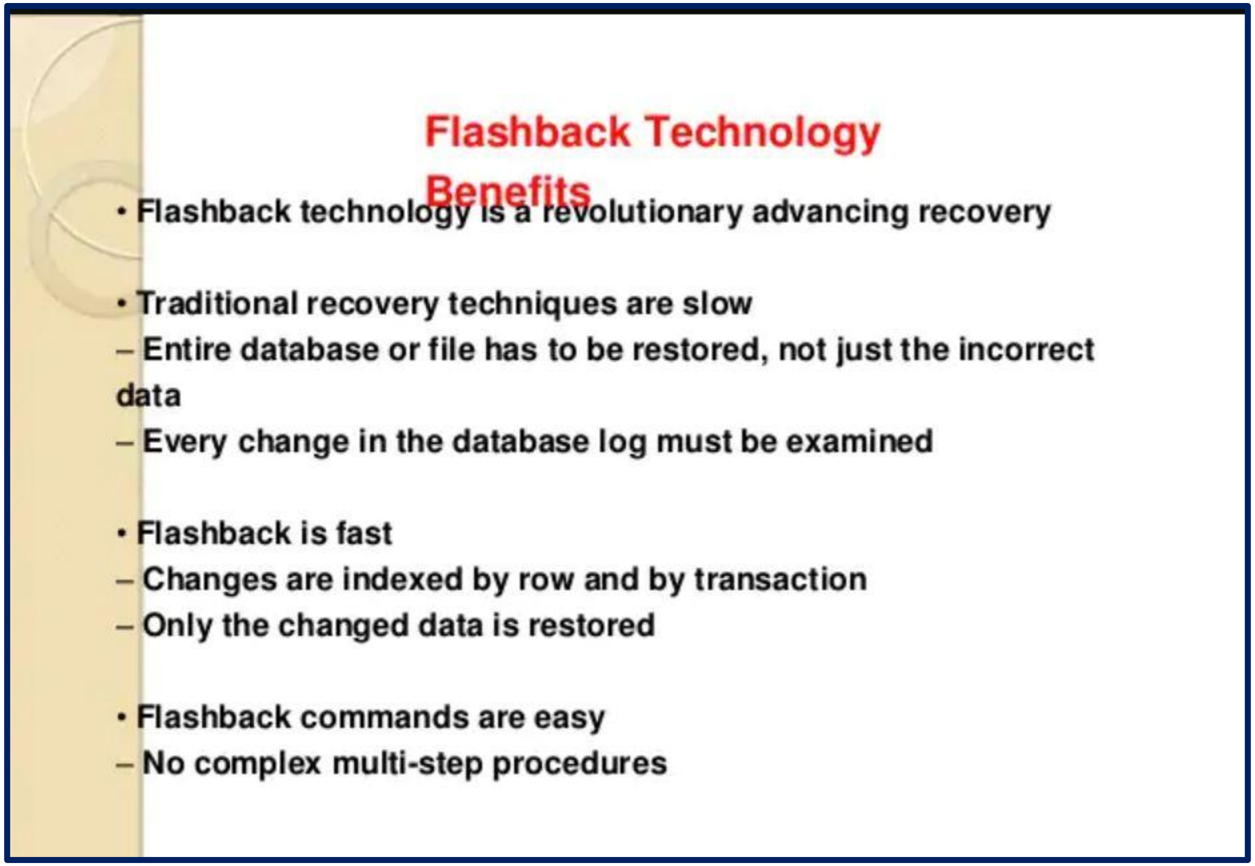
Flashback Drop



When to Use Flashback Technology

Flashback technology should be used when a logical corruption occurs in the Oracle database,and you need to recover quickly and easily.

Object Level	Scenario	Flashback Technology
Database	Drop User	Flashback Database
	Truncate Table	Flashback Database
	Batch job: partial changes	Flashback Database
Table	Drop Table	Flashback Drop
	Update with wrong WHERE clause	Flashback Table
	Comparing current data against the data at some time in the past	Flashback Query
Tx	Batch Job runs twice, but not really sure of the objects affected	Flashback Query



Flashback Technology Benefits

- Flashback technology is a revolutionary advancing recovery

- Traditional recovery techniques are slow

– Entire database or file has to be restored, not just the incorrect data

– Every change in the database log must be examined

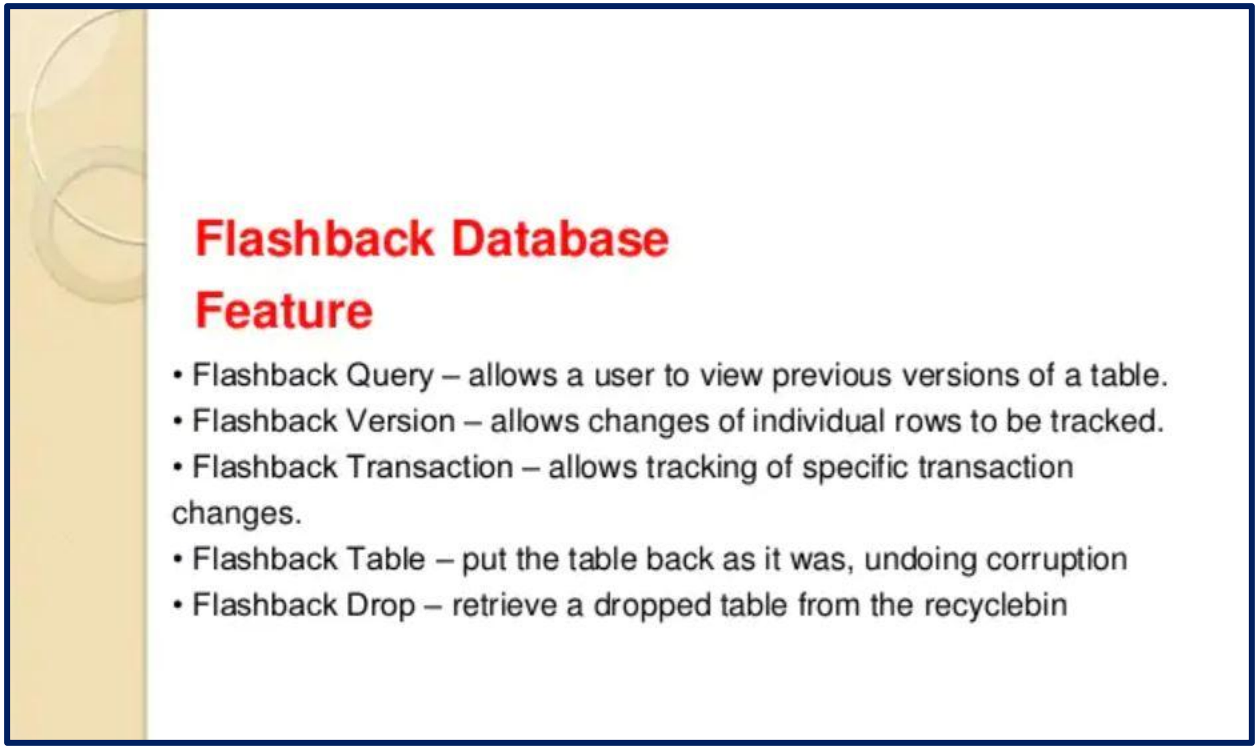
- Flashback is fast

– Changes are indexed by row and by transaction

– Only the changed data is restored

- Flashback commands are easy

– No complex multi-step procedures



Flashback Database Feature

- Flashback Query – allows a user to view previous versions of a table.
- Flashback Version – allows changes of individual rows to be tracked.
- Flashback Transaction – allows tracking of specific transaction changes.
- Flashback Table – put the table back as it was, undoing corruption
- Flashback Drop – retrieve a dropped table from the recyclebin

Thank You