Recovery Techniques

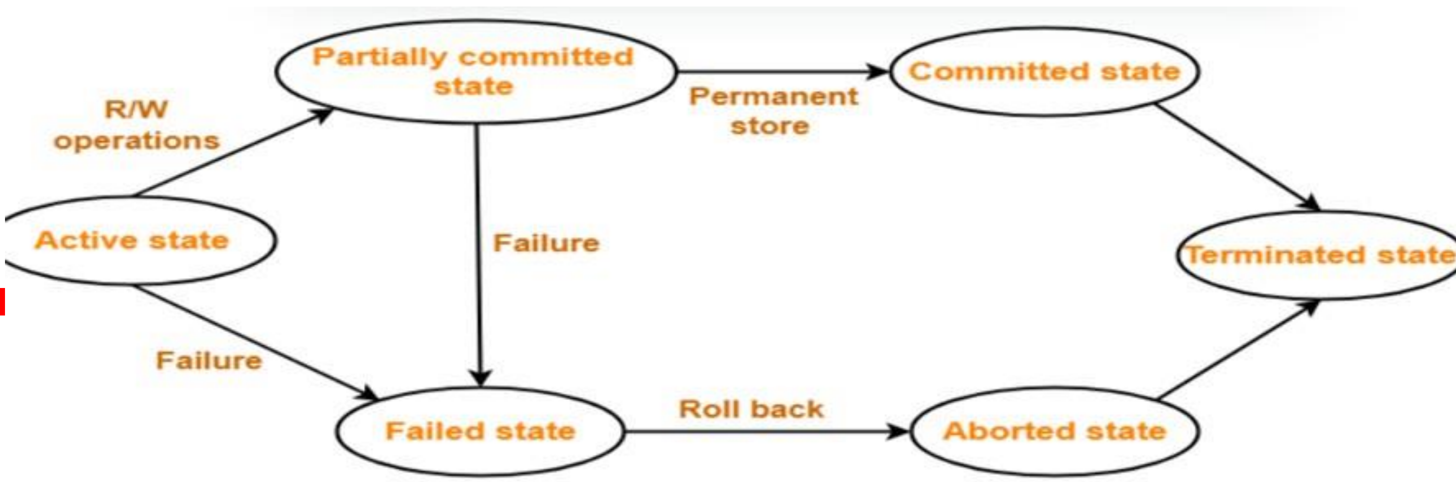
Introduction:

- Data recovery is the process of restoring the database of the most recent consistent state that existed just before the failure.
- Data recovery is important in managing a database system.
- Having a good plan will help to make a quick and successful recovery from failure.

Purpose of Database Recovery:

To bring the database into the last consistent state, which existed prior to the failure.

To preserve transaction (ACID) properties:
(a) Atomicity
(b) Consistency
(c) Isolation
(d) Durability

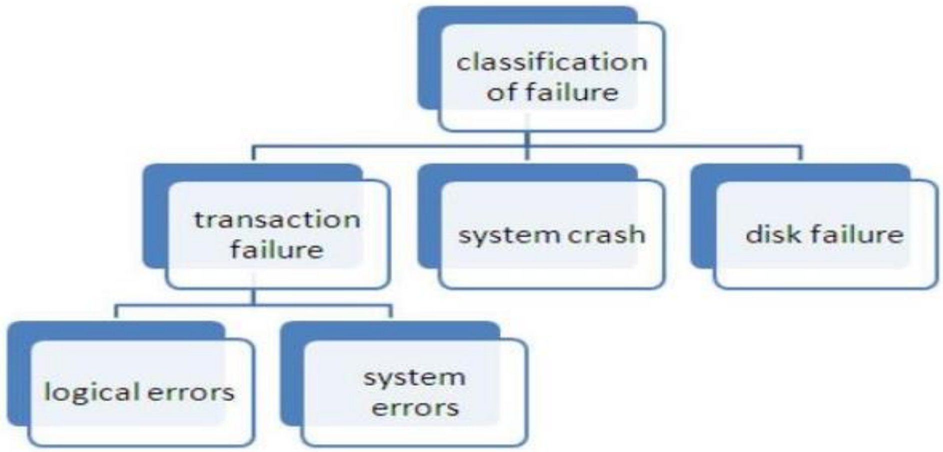


Transaction States in DBMS

Database Failure:

A failure is a state where the data inconsistency is visible to transactions if they are scheduled for execution. In databases usually three types of failures are encountered:

(a) Transaction failure
(b) System failure
(c) Disk failure



Causes of Database Failure:

- Database server computer or program crashes
- Database client computer or program crashes
- Network failure between client and server
- Data on a disk drive is corrupted
- A transaction executes an illegal operation
- Two or more transactions deadlock
- Transactions introduce errors into the database

Need of Database Recovery:

- The storage of data usually includes four types of media with an increasing amount of reliability: the main memory, the magnetic disk, the magnetic tape, and the optical disk.
- Many different forms of failure can affect database processing and/or transaction, and each of them has to be dealt with differently.
- Some data failures can affect the main memory only, while others involve non-volatile or secondary storage also.

Among the sources of failure are:

- Due to hardware or software errors, the system crashes, which ultimately resulting in loss of main memory.
- Failures of media, such as head crashes or unreadable media that results in the loss ofportions of secondary storage.
- There can be application software errors, such as logical errors that are accessing thedatabase that can cause one or more transactions to abort or fail.

- Natural physical disasters can also occur, such as fires, floods, earthquakes, or powerfailures.
- Carelessness or unintentional destruction of data or directories by operators or users.
- Damage or intentional corruption or hampering of data (using malicious software or files) hardware or software facilities.

Whatever the grounds of the failure are, there are two principal things that you have toconsider:

- Failure of main memory, including that database buffers.
- Failure of the disk copy of that database.

Recovery Techniques:

Log Based Recovery

- Logs are the continuation of records which are used to oversee records of the activities during an exchange.
- Logs are composed before the real change and put away on a steady stockpiling media.

Log Based Recovery procedure works in three distinct habits as follows –

- Conceded Update
- Quick Update
- Checkpoint

Conceded Update Method

- In this technique, an information base isn't truly refreshed on a circle until after an exchange arrives at its submitting point.
- After it, the updates are put away perseveringly in the log and afterward stayed connected with the information base.
- Before the submitting point, the exchange refreshes are overseen in the nearby exchange workspace like cradles.
- In the event that an exchange comes up short prior to coming to the submit point, it won't have changed the information base.
- Subsequently, there is no compelling reason to UNDO.
- So, it is important to REDO the impact of the tasks of a submitted exchange from the log, since then impact may not yet have been recorded.

Quick Update Method

- In this technique, the information base might be refreshed by certain activities of an exchange before the exchange compasses it submit point.
- These activities are reliably recorded in the sign on circle viably composing before adjusted.
- If an exchange prematurely ends after keeping record of a few changes to the information base, however before submit point, the impact of its procedure on the data set should be fixed.

Reserving/Buffering

- In this at least one circle, pages that incorporate information things to be refreshed are stored into principal memory supports and afterward refreshed in memory prior to being composed back to plate.
- An assortment of in-memory cushions called the DBMS reserve is monitored by DBMS for holding these cradles.
- A catalogue is utilized to monitor which information base things are in the cradle.

RAID

- Redundant Array of Independent Disk (RAID) combines multiple small, inexpensive disk drives into an array of disk drives which yields performance more than that of a Single Large Expensive Drive (SLED).
- RAID is also called Redundant Array of Inexpensive Disks.
- Storing the same data in different disk increases the fault-tolerance.
- The array of Mean Time Between Failure (MTBF) = MTBF of an individual drive, which is divided by the number of drives in the array.
- Because of this reason, the MTBF of an array of drives are too low for many application requirements.

Types of RAID:

RAID-0

- RAID Level-0 is not redundant.
- Since no redundant information is stored, performance is very good, but the failure of any disk in the array results in data loss.
- A single record is divided into strips typically 512 bytes and is stored across all disks.
- The record can be accessed quickly by reading all disks at the same time, called as striping.

RAID-1

- RAID Level-1 provides redundancy by writing all data into two or more drives.
- The performance is faster on reads and slower on writes compared to a single drive.
- If anyone drives fails, no data is lost.
- This method is called mirroring.

RAID-2

- RAID Level-2 is used for Hamming error correction codes and is used with drives which do not have built-in error detection.

RAID-3

- RAID Level-3 stripes data at a byte level across several drives, with parity stored on one drive.
- Byte- level stripping hardware supports efficient use.

RAID-4

- RAID Level-4 that stripes data at a block level across several drives, with parity stored on one drive.
- Parity information allows recovery from the failure of any single drive.
- The performance of the level-4 array is good for reads.
- Writes, however, require that parity data be updated each time.
- Because only one drive in the array stores redundant data.
- The cost per megabyte is low.

RAID-5

- RAID Level-5 is like level 4 but distributes parity among the drives.
- This can speed up small writes in the multiprocessing system.
- The performance for reads is lower than a level-4 array.
- The cost per megabyte is the same as level-4.

SUMMARY:

Levels	Summary
RAID-0	It is the fastest and most efficient array type but offers no fault-tolerance.
RAID-1	It is the array of choice for a critical, fault tolerant environment.
RAID-2	It is used today because ECC is embedded in almost all modern disk drives.
RAID-3	It is used in single environments which access long sequential records to speed up data transfer.
RAID-4	It offers no advantages over RAID-5 and does not support multiple simultaneous write operations.
RAID-5	It is the best choice in a multi-user environment. However, at least three drives are required for the RAID-5 array.