eBOX
AN AMPHISOFT PRODUCT

Transaction Recovery and Save Points

Content

- Transaction Recovery
- Crash Recovery
- Failure Classification
- Transaction failure
- Log-based recovery Or Manual Recovery
- Save Points
- Isolation Levels
- SQL Facilities for Concurrency and Recovery

Transaction Recovery

- Recovery is the process of restoring a database to the correct state in the event of a failure.
- It ensures that the database is reliable and remains in consistent state in case of a failure.

Database recovery can be classified into two parts;

1. **Rolling Forward** applies redo records to the corresponding data blocks.
2. **Rolling Back** applies rollback segments to the data files. It is stored in transaction tables.

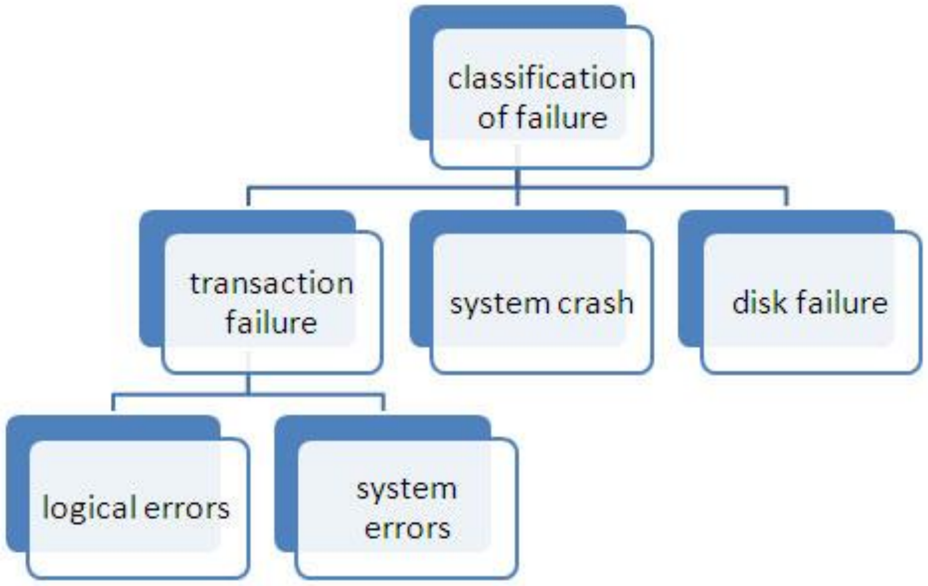
Crash Recovery

- DBMS is a highly complex system with hundreds of transactions being executed every second.
- The durability and robustness of a DBMS depends on its complex architecture and its underlying hardware and system software.
- If it fails or crashes amid transactions, it is expected that the system would follow some sort of algorithm or techniques to recover lost data.

Failure Classification

To see wherever the matter has occurred, we tend to generalize a failure into numerous classes, as follows:

- Transaction failure
- System crash
- Disk failure



Failure Classification

- **Transaction failure:** A transaction needs to abort once it fails to execute or once it reaches to any further extent from wherever it can’t go to any extent further. This is often known as transaction failure wherever solely many transactions or processes are hurt. The reasons for transaction failure are:
- Logical errors
- System errors

Failure Classification

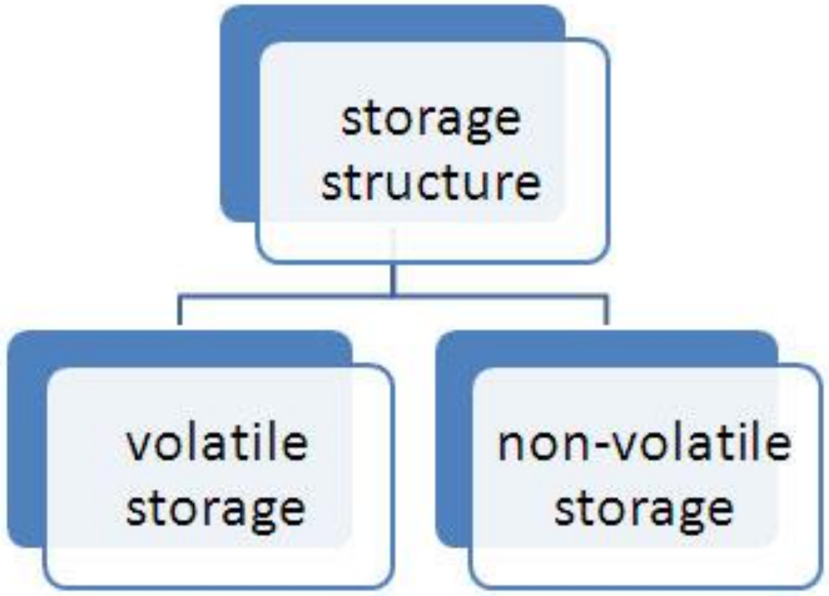
- **Logical errors:** Where a transaction cannot complete as a result of its code error or an internal error condition.
- **System errors:** Wherever the information system itself terminates an energetic transaction as a result of the DBMS isn’t able to execute it, or it’s to prevent due to some system condition. to Illustrate, just in case of situation or resource inconvenience, the system aborts an active transaction.

Failure Classification

- **System crash:** There are issues – external to the system – that will cause the system to prevent abruptly and cause the system to crash. For instance, interruptions in power supply might cause the failure of underlying hardware or software package failure.
- **Disk failure:** In early days of technology evolution, it had been a typical drawback wherever hard-disk drives or storage drives accustomed to failing oftentimes. Disk failures include the formation of dangerous sectors, unreachability to the disk, disk crash or the other failure, that destroys all or a section of disk storage.

Categories of Storage Structure:

- We have already described the storage system. In brief, the storage structure can be divided into two categories –



Categories of Storage Structure:

- **Volatile storage** – A volatile storage cannot survive system crashes. Volatile storage devices are placed very close to the CPU; normally they are embedded onto the chipset itself. For example, main memory and cache memory are examples of volatile storage.
- **Non-volatile storage** – These memories are made to survive system crashes. They are huge in data storage capacity, but slower in accessibility. Examples may include hard-disks, magnetic tapes, flash memory, and non-volatile (battery backed up) RAM.

Log-based recovery Or Manual Recovery:

- Log could be a sequence of records, which maintains the records of actions performed by dealing. It’s necessary that the logs area unit written before the particular modification and hold on a stable storage media, that is failsafe. Log-based recovery works as follows:
- The log file is unbroken on a stable storage media.
- When a transaction enters the system and starts execution, it writes a log regarding it.

Log-based recovery Or Manual Recovery:

The database can be modified using two approaches –

- **Deferred database modification** – All logs are written on to the stable storage and the database is updated when a transaction commits.
- **Immediate database modification** – Each log follows an actual database modification. That is, the database is modified immediately after every operation.

Transactions Recovery

- Transaction recovery involves the transaction log file that is used as a write-ahead log, plus journal files maintained on a per-database basis.
- Log files contain short-term recovery information regarding active databases, while the journal files contain long-term information used for auditing and disaster recovery.

Types of Transaction Recovery

Recovery information is divided into two types:

- **Undo (or Rollback) Operations**
  - ✓ Undo or transaction back out recovery is performed by the DBMS Server.
- **Redo (or Cache Restore) Operations**
  - ✓ A Redo recovery operation is database-oriented. Redo recovery is performed after a server or an installation fails.

Save Points

- **SAVEPOINT** – creates points within the groups of transactions in which to ROLLBACK.

The SAVEPOINT Command

- A SAVEPOINT is a point in a transaction when you can roll the transaction back to a certain point without rolling back the entire transaction.

Syntax for a SAVEPOINT command:

-- **SAVEPOINT SAVEPOINT_NAME;**

Isolation Levels

- Isolation levels determine the type of phenomena that can occur during the execution of concurrent transactions.

Three phenomena define SQL Isolation levels for a transaction:

- Dirty Reads
- Non-Repeatable Reads
- Phantom Read

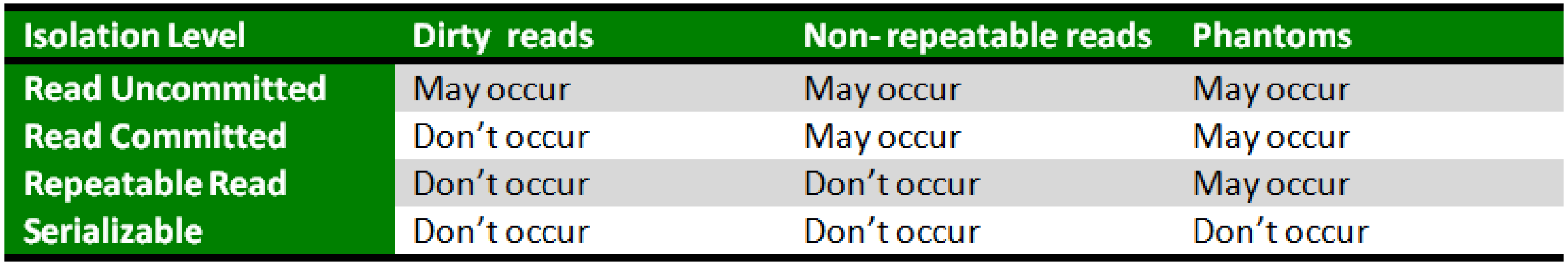
SQL Facilities

- Based on these phenomena, The SQL standard defines four isolation levels :

1. Read Uncommitted **2. Read Committed**

3. Repeatable Read **4. Serializable**

The relationship between isolation levels, read phenomena and locks



Isolation Level	Dirty reads	Non-repeatable reads	Phantoms
Read Uncommitted	May occur	May occur	May occur
Read Committed	Don't occur	May occur	May occur
Repeatable Read	Don't occur	Don't occur	May occur
Serializable	Don't occur	Don't occur	Don't occur

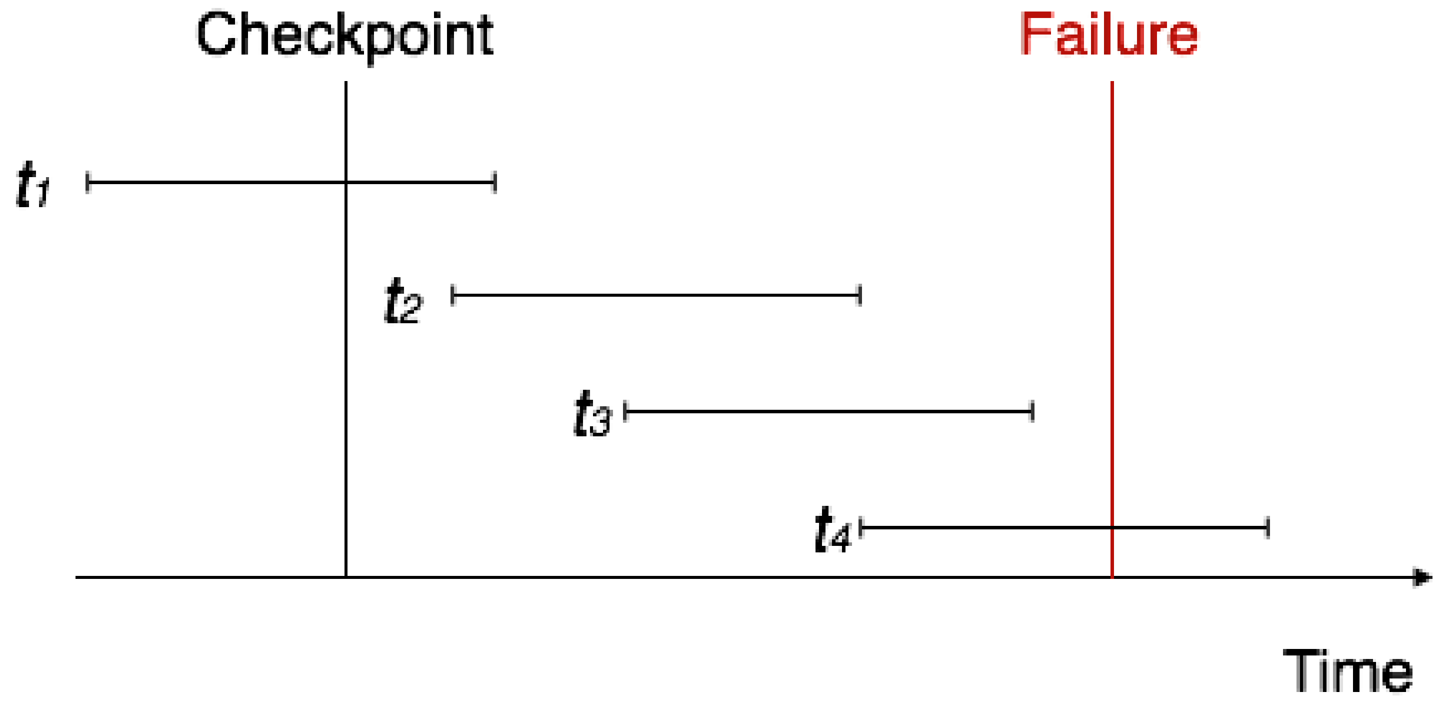
Recovery with Concurrent Transactions

- When more than one transaction are being executed in parallel, the logs are interleaved. At the time of recovery, it would become hard for the recovery system to backtrack all logs, and then start recovering.

Checkpoint

- Checkpoint is a mechanism where all the previous logs are removed from the system and stored permanently in a storage disk. Checkpoint declares a point before which the DBMS was in consistent state, and all the transactions were committed.

Recovery with Concurrent Transactions



Recovery with Concurrent Transactions

- The recovery system reads the logs backwards from the end to the last checkpoint.
- It maintains two lists, an undo-list and a redo-list.
- If the recovery system sees a log with <T_n, Start> and <T_n, Commit> or just <T_n, Commit>, it puts the transaction in the redo-list.
- If the recovery system sees a log with <T_n, Start> but no commit or abort log found, it puts the transaction in undo-list.

THANK YOU

eBOX

AN AMPHISOFT PRODUCT