Unit 4

Crash Recovery

1. Introduction

When a DBMS crashes, all or a portion of the data can become unusable. Appropriate procedures must be followed to restore, validate, and return the system to normal. Recovery–that is, the return to a fully operational environment after a hardware or software failure–is an important process. Moreover, the effects of a system failure on the organization must be curtailed to minimize any substantial financial loss. Actions must be taken to prevent DBMS failures or resolve them quickly if they occur.

There are many causes of DBMS failure. When a DBMS fails, it falls into an incorrect state and will likely contain erroneous data. Typical causes of DBMS failures include errors in the application program, an error by the terminal user, an operator error, loss of data validity and consistency, a hardware error, media failures, an error introduced by the environment, and errors caused by mischief or catastrophe.

2. Transaction Failure Classification

Typically, the three major types of failure that result from a major hardware or software malfunction are

i. Transaction
ii. System and
iii. Media failure

Various types of failure
i. Transaction failure
ii. System failure
iii. Media failure

These failures may be caused by a natural disaster, computer crime, or user, designer, developer, or operator error. Each type of failure is described below:

i. **Transaction failure:** Transaction failures occur when the transaction is not processed and the processing steps are rolled back to a specific point in the processing cycle. Transaction failure can occur when some, but not all, physical databases are updated at the same time.

ii. **System failure:** System failure can be caused by bugs in the data base, operating system, or hardware. In each case, the transaction processing is terminated without control of the application. Data in the memory is lost; however, disk storage remains stable. The system must recover in the amount of time it takes to complete all interrupted transactions. At one transaction per second, the system should recover in a few seconds. System failures may occur as often as several times a week.

iii. **Media failure:** Disk crashes or controller failures can occur because of disk-write bugs in the operating system release, hardware errors in the channel or controller, head crashes, or media degradation. These failures are rare but costly.

By identifying the type of DBMS failure, an organization can define the state of activity to return to after recovery. To design the database recovery procedures, the potential failures must be identified and the reliability of the hardware and software must be determined.

Types of Failures

2

Oct. 2018 – *4M*
Explain different types of failures in detail.
Oct. 2017 – *3M*
What are different types of failures?

i. **A computer failure (system crash):** A hardware, software, or network error occurs in the computer system during transaction execution. Hardware crashes are usually *media failures*, *for example*, main memory failure.

ii. **A transaction or system error:** Some operation in the transaction may cause it to fail, such as integer overflow or division by zero. Transaction failure may also occur because of erroneous parameter values or because of a logical programming error. In addition, the user may interrupt the transaction during its execution.

iii. **Logical errors or exceptional conditions detected by the transaction:** During transaction execution, certain conditions may occur that necessitate cancellation of the transaction. *For example*, data for transaction may not be found. Notice that an exceptional condition, such as insufficient account balance in a banking database, may

cause a transaction, such as a fund withdrawal to be cancelled. This exception should be programmed in the transaction itself, and hence would not be considered a failure.

iv. **Concurrency control enforcement:** The concurrency control method may decide to abort the transaction, to be restarted later, because it violates serializability or because several transactions are in a state of deadlock.

v. **Disk failure:** Some disk blocks may lose their data because of a read or write malfunction or because of a disk read/write head crash. This may happen during read or write operation of the transaction.

vi. **Physical problems and catastrophes:** This refers to an endless list of problems that includes power or air – conditioning failure, fire, theft, sabotage, overwriting disks or tapes by mistake, and mounting of a wrong tape by an operator.

3. Recovery Concepts

Recovery from the transaction failures usually means that the database is restored to the most recent consistent state just before the time of failure to do this, the system must keep information about the changes that were applied to the data items by the various transactions.

This information is typically kept in the system log.

Why recovery is needed?

Whenever a transaction is submitted to a DBMS for execution, the system is responsible for making sure that either

i. All the operations in the transaction are completed successfully and their effect is recorded permanently in the database, or

ii. The transaction has no effect whatsoever on the database or on any other transactions.

The DBMS must not permit some operations of a transaction T to be applied to the database while other operations of T are not.

This may happen if a transaction *fails* after executing some of its operations, but before executing all of them.

4. Checkpoints

Oct. 2018 – *4M*
What are checkpoints? How are they useful in crash recovery?

Checkpoint is another type of entry in the log.

A [Checkpoint] log record is written into the log periodically at that point when the system writes out to the database on the disk, the effect of all write operations of all committed transactions.

Hence, all transactions that have their [commit, T] entries in the log before a [Checkpoint] entry, do not have to redo their write operations in case of a system crash.

The periodic intervals after which a checkpoint is to be taken is decided by the recovery manager component of the DBMS.

Taking a Checkpoint consists of the following actions

i. Suspend execution of transaction temporarily.

ii. Force write all update operations of committed transactions from main memory buffers to disk.

iii. Write a [Checkpoint] record to the log and force write the log to the disk.

iv. Resume executing transactions.

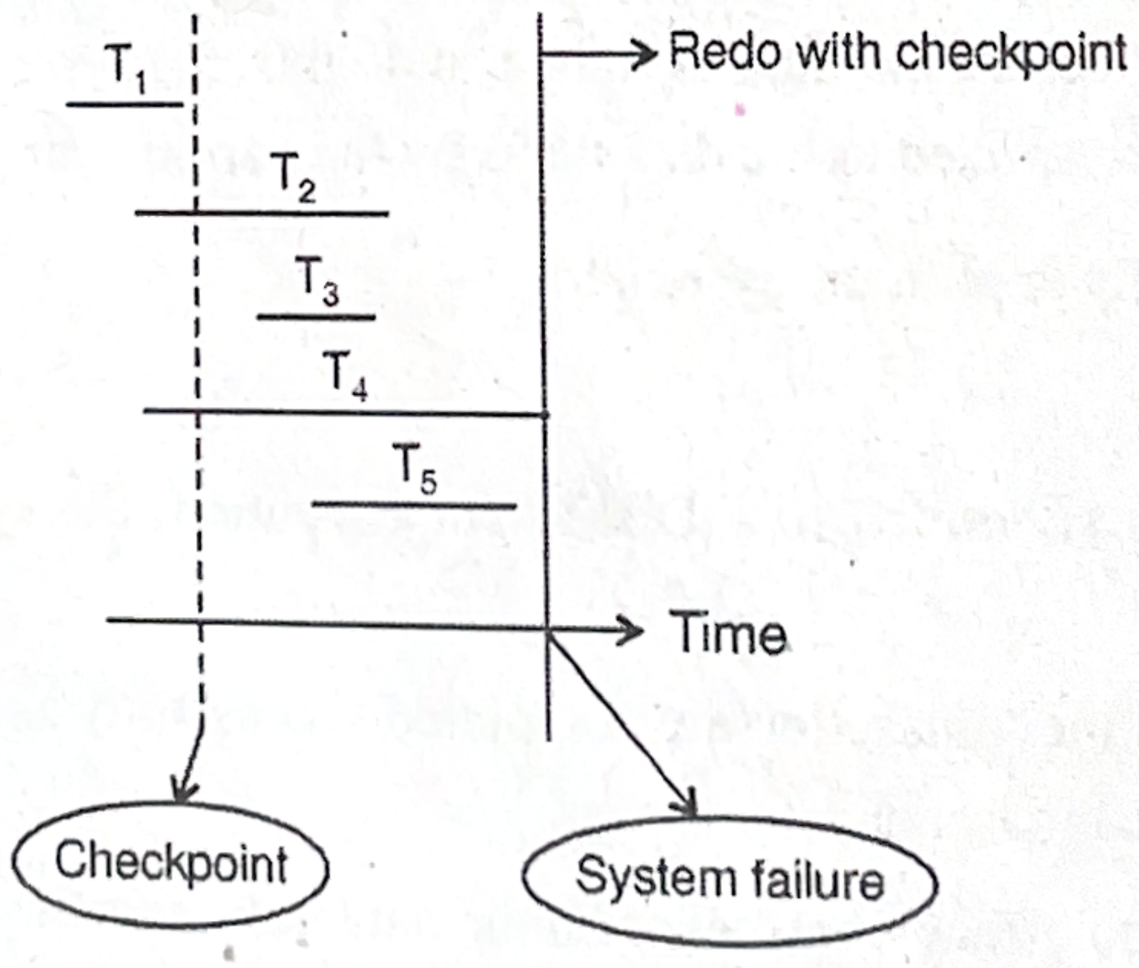

Figure 4.1

Changes made by T_1 are stored in the database. So, there is nothing to do with them.

DBMS will redo transactions T_2 and T_3 (their log is already on disk).

DBMS or the user has to rerun T_4 and T_5.