# UNIT VI

## Crash Recovery & Indexing

### 1. Failure Classification

There are various types of failure that may occur in a system, each of which needs to be dealt with in a different manner. Broadly failures are classified into the following types.

- **Transaction failure.** There are two types of errors that may cause a transaction to fail:
  1. **Logical error.** The transaction can no longer continue with its normal execution because of some internal condition, such as bad input, data not found, overflow, or resource limit exceeded.
  2. **System error.** The system has entered an undesirable state (deadlock), so the transaction cannot continue with its normal execution. The transaction, however, can be re executed at a later time.
- **System crash.** There is a hardware malfunction, or a bug in the database software or the operating system, that causes the loss of the content of volatile storage, and brings transaction processing to a halt. The content of nonvolatile storage remains intact, and is not corrupted.
  - The assumption that hardware errors and bugs in the software bring the system to a halt, but do not corrupt the nonvolatile storage contents, is known as the **fail-stop assumption**.
- **Disk failure.** A disk block loses its content as a result of either a head crash or failure during a data transfer operation. Copies of the data on other disks, or archival backups on tertiary media, such as tapes, are used to recover from the failure.

**Storage Types**

- **Volatile storage.** Information stored in volatile storage does not usually survive system crashes. Examples volatile storage: main memory and cache memory. Access to volatile storage is extremely fast.
- **Nonvolatile storage.** Information residing in nonvolatile storage survives system crashes. Example nonvolatile storage is disk and magnetic tapes. Disks are used for online storage, whereas tapes are used for archival storage. Nonvolatile storage is slower than volatile storage. In database systems, disks are used for most nonvolatile storage. Other nonvolatile media are normally used only for backup data.
- **Stable storage.** Information residing in stable storage is never lost.

## 2. Different Types of Recovery Techniques

### 2.1. Log-Based Recovery

- The most widely used structure for recording database modifications is the **log**. The log is a sequence of **log records**, recording all the update activities in the database. There are several types of log records.
- We denote various types of log records as:
    1. <Ti start>. Transaction Ti has started (start log record).
    2. <Ti, Xj, V1, V2>. Transaction Ti has performed a write on data item Xj. Xj had value V1 before the write, and will have value V2 after the write (update log record).
    3. <Ti commit>. Transaction Ti has committed (commit log record).
    4. <Ti abort>. Transaction Ti has aborted (abort log record).

#### 2.1.1. Deferred Database Modification

- In this, the modifications done by the transaction are not recorded in the database but **deferred (postponed)** until the transaction commits. However, these modifications are recorded in the log file.

- When a transaction commits, the modifications are applied to the database using the log file. This process is called as re-doing.
- If the system crashes before the transaction completes its execution, or if the transaction aborts, then the information on the log is simply ignored.
- As an example, consider the following; Assume initial values of A=50 and B=100

| **Transaction(T1)** | **Database Before** | **Database After** | **Log** |
|---|---|---|---|
| Start | A=50, B=100 | A=50, B=100 | <T1, start> |
| Read(A) | 50 | 50 | -- |
| Update A to 500 | 50 | 50 | <T1,A,50,500> |
| Read(B) | 100 | 100 | --- |
| Update B to 200 | 100 | 100 | <T1,B,100,200> |
| Commit | | A=500, B=200 | <T1, Commit> |

| **Transaction(T1)** | **Database Before** | **Database After** | **Log** |
|---|---|---|---|
| Start | A=50, B=100 | A=50, B=100 | <T1, start> |
| Read(A) | 50 | 50 | -- |
| Update A to 500 | 50 | 50 | <T1,A,50,500> |
| Read(B) | 100 | 100 | --- |
| Update B to 200 | 100 | 100 | <T1,B,100,200> |
| Abort | | A=50, B=100 | <T1, Abort> |

### 2.1.2.Immediate Database Modification

- In this, the modifications done by the transaction are immediately applied to the database. However, these modifications are first recorded in the log file before applying to the database.
- When a transaction commits, the modifications are made permanent and the log contents are discarded.
- If the system crashes before the transaction completes its execution, or if the transaction aborts, all the modifications done to the database are discarded. This process is called as Un-doing.
- As an example, consider the following; Assume initial values of A=50 and B=100

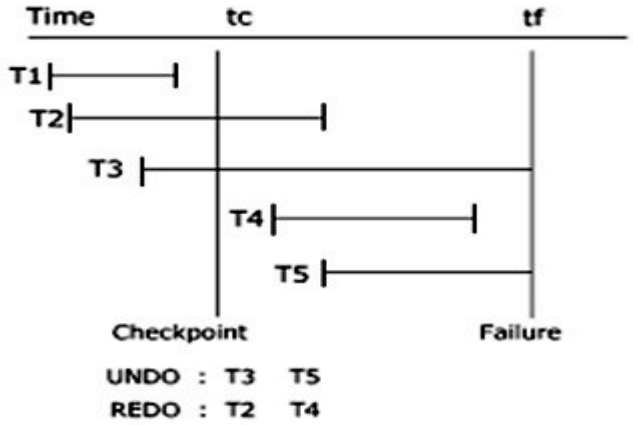
| Transaction(T1) | Database Before | Database After | Log |
|---|---|---|---|
| Start | A=50, B=100 | A=50, B=100 | <T1, start> |
| Read(A) | 50 | 50 | -- |
| Update A to 500 | 50 | 500 | <T1,A,50,500> |
| Read(B) | 100 | 100 | --- |
| Update B to 200 | 100 | 200 | <T1,B,100,200> |
| Commit | | A=500, B=200 | <T1, Commit> |

| Transaction(T1) | Database Before | Database After | Log |
|---|---|---|---|
| Start | A=50, B=100 | A=50, B=100 | <T1, start> |
| Read(A) | 50 | 50 | -- |
| Update A to 500 | 50 | 500 | <T1,A,50,500> |
| Read(B) | 100 | 100 | --- |
| Update B to 200 | 100 | 200 | <T1,B,100,200> |
| Abort | | A=50, B=100 | <T1, Abort> |

- Both *redo* and *undo* operations are idempotent; that is, executing it several times must be equivalent to executing it once.

### 2.1.3.Check-pointing

- When a system failure occurs, we must use the log to determine those transactions that need to be redone and those that need to be undone. In principle, we need to search the entire log to determine this information.
- There are two major difficulties with this approach:
  1. The search process is time consuming.
  2. Most of the transactions that, according to our algorithm, need to be redone have already written their updates into the database. Although redoing them will cause no harm, it will nevertheless cause recovery to take longer.
- To reduce these types of overhead, the system periodically performs **checkpoints.**
- Updating the database at fixed intervals of time is called as check-pointing.
- As an example, consider the following figure.



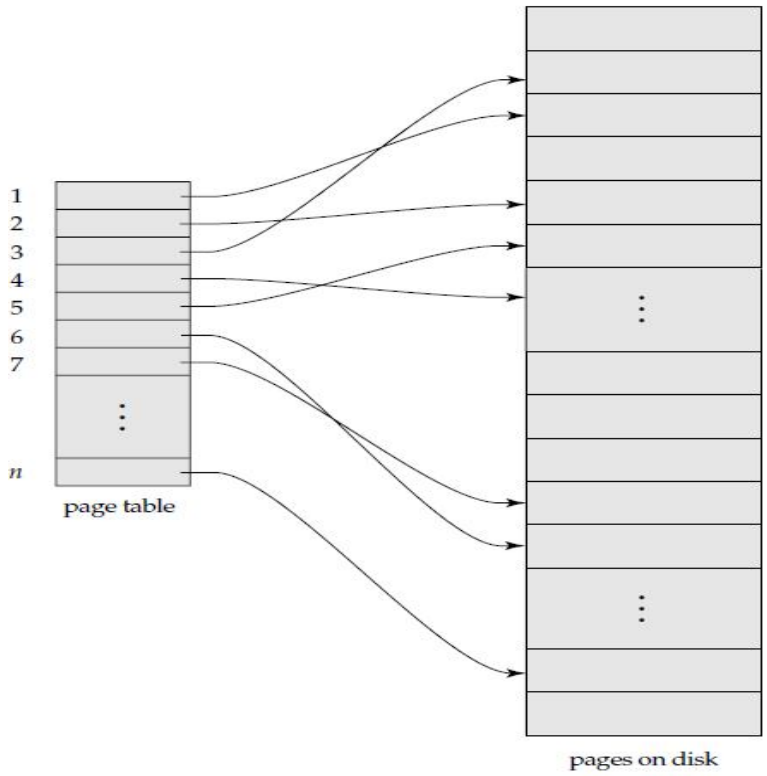

In check pointing, recovery process proceeds as follows.

- As shown in the figure, T1 has committed before check point hence, nothing is done to T1.

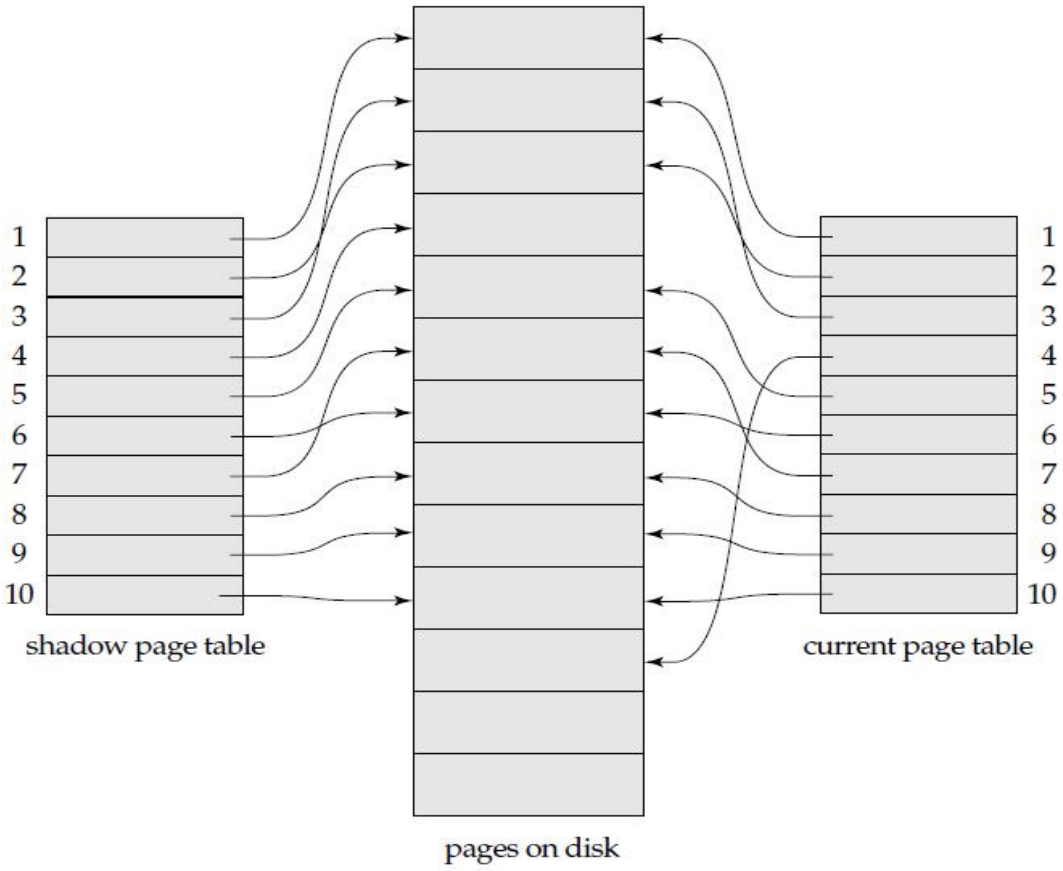
- T2 and T4 have not committed before check point but committed before system failure hence, T2 and T4 are Redone.
- T3 and T5 have not committed before check point as well as before system failure hence, T3 and T5 are Undone.

### 3. Shadow Paging

- An alternative to log-based crash-recovery techniques is **shadow paging**.
- Shadow paging may require fewer disk accesses than do the log-based methods discussed previously.
- The database is partitioned into some number of fixed-length blocks, which are referred to as **pages** as shown in the figure below.
- The key idea behind the shadow-paging technique is to maintain *two* page tables during the life of a transaction: the **current page table** and the **shadow page table**.



- When the transaction starts, both page tables are identical. The shadow page table is never changed over the duration of the transaction. The current page table may be changed when a transaction performs a write operation.
- All modifications done by the transaction are applied to the current page table only.
- Suppose that the transaction $T_j$ performs a write($X$) operation, it modifies the current page table so that the $i^{th}$ entry points to the modified page.




- The above figure shows the shadow and current page tables of some transaction $T_j$. It also shows that transaction $T_j$ has modified the data item present in $4^{th}$ page table.

- When a transaction commits before a system crash, the shadow page table is replaced with the current page table, since we may need it for recovery from a crash.
- When a transaction does not commit before a system crash, the current page table is simply ignored and the shadow page table is used as it is.
- The main disadvantage of shadow paging is that it requires the transactions be executed serially.

## UNIT VI

### Assignment-Cum-Tutorial Questions

### SECTION-A

***Objective Questions***

1. Which of the following is under failure classification? [ ]
   i. Transaction failure
   ii. System crash
   iii. Disk Failure
   iv. Storage failure

A) i and ii B) i , ii and iii

C) i, ii, iii and iv D) ii and iii

2. The most widely used structure for recording database modifications is called _____.

3. Identify crash recovery techniques from the following. [ ]

A) Log-Based B) Check-pointing

C) Shadow- Paging D) All of the above

4. An update log record contains which of the following [ ]

A) Transaction identifier B) Data item identifier

C) Old value and new value D) All of the above

5. What is the effect of the UNDO operation corresponding to a log record where Ti is the transaction, and V1 and V2 are the old and new values respectively of a data location X? [ ]

A) No change to X C) Writes the value V1 to X

B) Writes the value V2 to X D) Sets X to 0

6. Consider the log of transactions below. [ ]

< T0 start >

< T0, S, 100, 120 >

< T0, H, 1, 3 >

Identify the correct actions, which are part of the Undo(T0)

A) H is restored to 3 C) S is set to 120

B) H is restored to 1 D) < T0, S, 100 > log record is written out

7. Identify the options which are true for Immediate-modification scheme.

A) Allows updates of an uncommitted transaction to be made to the buffer, or the disk itself, before the transaction commits. [ ]

B) Update log record must be written after a database item is written.

C) Output of updated blocks to disk storage can take place at any time before or after transaction commit.

D) Performs updates to buffer/disk only at the time of transaction commit

8. Consider the following log sequence: [ ]

```
<T0, start>
<T0, A, 1000, 950>
<T0, B, 2000, 2050>
<T0, commit>
<T1, start>
<T1, C, 700, 600>
```

What will be the recovery action by immediate modification recovery?

A) undo T0, redo T1 C) redo T0, redo T1

B) redo T0, undo T1 D) undo T0, undo T1

9. Identify the incorrect statement based on checkpointing. [ ]

A) All updates are stopped while doing check pointing.

B) Continue scanning backwards till a record < Ti start > is found for every transaction Ti in L.

C) Scan backwards from end of log to find the most recent < checkpoint L > record.

D) Scan from starting of log to find the most recent < checkpoint L > record.

10. What is the effect of the UNDO operation corresponding to a log record < Ti , Y, S, K > where Ti is the transaction, and S and K are the old and new values respectively of a data location Y? [ ]

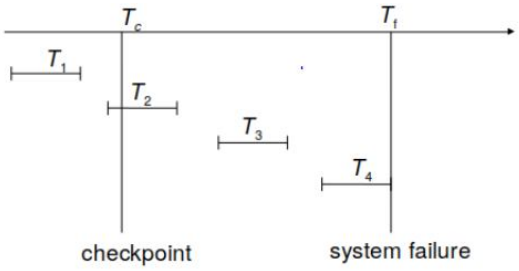
A) No change to Y C) Writes the value S to Y

B) Writes the value K to Y D) Sets Y to 0.

## *SECTION-B*

### *Descriptive Questions*

1. Explain the concept of failure classification.
2. Distinguish between different Storage Mechanisms
3. Explain different types of Recovery Techniques.
4. Distinguish between immediate and deferred database modification (update).
5. Illustrate check pointing with an example.
6. Summarize the importance of shadow paging.
7. Consider the following state of transactions:




and the statements below:

1. T1 can be ignored.
2. T2 and T3 redone
3. T4 undone
4. T4 redone

Mark the correct group of statements from the options below.

A) 1), 2), 4)

B) 1), 2), 3)

C) only 1) and 2) but not 3)

D) only 2) and 3) but not 1)

8. Consider the log of transactions given below and answer the Q. No-6 and Q. No-7:

< T2 start >

< T2, H, 18, 20 >

< T3 start >

< checkpoint {T2, T3} >

< T3 commit >

< T4 start >

< T4, G, 6, 7 >

< T2, Y, 12 >

< T2 abort >

Suppose there is a crash after the record < T2 abort >.

Identify the correct statement(s) from the Redo phase

A) The undo list initially contains T2, T3

B) The undo list initially contains T2, T3, T4

C) T3 is removed from undo list after some steps

D) T2 is removed from undo list after some steps.

9. Identify the incorrect statement(s) based on the Undo phase

A) The undo list at the start of the undo phase contains T2, T4

B) < T4, G, 6 > log record is written out

C) The undo list at the start of the undo phase contains T2

D) < T4, abort > log record is written out