**ST. XAVIER’S COLLEGE**

**Maitighar, Kathmandu**



**DATABASE MANAGEMENT SYSTEM**

**Theory Assignment #9**

**Submitted by**

Sushant Gautam

013BSCCSIT043

**Submitted to**

|  |  |
| --- | --- |
| Er. Sanjay Kumar Yadav  (Lecturer, St. Xavier’s College ) |  |

**Date of Submission:**

**Thursday, October 01, 2015**

**Database Recovery:**

**Database Recovery** is the process of restoring the database and the data to a consistent state. This may include restoring lost data up to the point of the event (e.g. system crash).

**Purpose of Database Recovery**

There are many situations in which a transaction may not reach a commit or abort point. Some of them are given below:

* To bring the database into the last consistent state which existed prior to the failure.
* To preserve transaction properties (Atomicity, Consistency, Isolation and Durability).
* There are many situations in which a transaction may not reach a commit or abort point.
* An operating system crash can terminate the DBMS processes
* The DBMS can crash
* The system might lose power
* A disk may fail or other hardware may fail.
* Human error can result in deletion of critical data.

Example:

If the system crashes before a fund transfer transaction completes its execution, then either one or both accounts may have incorrect value. Thus, the database must be restored to the state before the transaction modified any of the accounts.

**Types of Failure**

The database may become unavailable for use due to

* **Transaction failure**: Transactions may fail because of incorrect input, deadlock, incorrect synchronization.
* **System failure**: System may fail because of addressing error, application error, operating system fault, RAM failure, etc.
* **Media failure**: Disk head crash, power disruption, etc.

Transaction Log:

* For recovery from any type of failure data values prior to modification (BFIM - BeFore Image) and the new value after modification (AFIM – AFter Image) are required.
* These values and other information is stored in a sequential file called Transaction log. A sample log is given below. Back P and Next P point to the previous and next log records of the same transaction.

Data Update

* **Immediate Update**: As soon as a data item is modified in cache, the disk copy is updated.
* **Deferred Update**: All modified data items in the cache is written either after a transaction ends its execution or after a fixed number of transactions have completed their execution.
* **Shadow update**: The modified version of a data item does not overwrite its disk copy but is written at a separate disk location.
* **In-place update**: The disk version of the data item is overwritten by the cache version.

Data Caching

* Data items to be modified are first stored into database cache by the Cache Manager (CM) and after modification they are flushed (written) to the disk.
* The flushing is controlled by **Modified** and **Pin-Unpin** bits.
* **Pin-Unpin**: Instructs the operating system not to flush the data item.
* **Modified**: Indicates the AFIM of the data item.

Transaction **Roll-back (Undo)** and **Roll-Forward (Redo)**

* To maintain atomicity, a transaction’s operations are redone or undone.
* **Undo**: Restore all BFIMs on to disk (Remove all AFIMs).
* **Redo**: Restore all AFIMs on to disk.
* Database recovery is achieved either by performing only Undos or only Redos or by a combination of the two. These operations are recorded in the log as they happen.