**ST. XAVIER’S COLLEGE**

**(Affiliated to Tribhuvan University)**

Maitighar, Kathmandu



**DATABASE MANAGEMENT SYSTEM**

**THEORY ASSIGNMENT #11\_01**

**Submitted by:**

Rojesh Tamrakar

013BSCCSIT032

**Submitted to:**

|  |  |
| --- | --- |
| **Er. Sanjay Kumar Yadav** |  |

Lecturer

Department of Computer Science

Date of submission: 30th August, 2015

**Transaction Management**

1. **Introduction**

A database transaction is a sequence of actions that are treated as a single unit of work. These actions should either complete entirely or take no effect at all. Transaction management is an important part of and RDBMS oriented enterprise applications to ensure data integrity and consistency.

1. **Transaction**

A transaction is a logical unit of work that contains one or more SQL statements. A transaction is an atomic unit. The effects of all the SQL statements in a transaction can be either all committed (applied to the database) or all rolled back (undone from the database).

A transaction begins with the first executable SQL statement. A transaction ends when it is committed or rolled back, either explicitly with a COMMIT or ROLLBACK statement or implicitly when a DDL statement is issued.

To illustrate the concept of a transaction, consider a banking database. When a bank customer transfers money from a savings account to a checking account, the transaction can consist of three separate operations:

* Decrement the savings account
* Increment the checking account
* Record the transaction in the transaction journal

1. **Transaction Recovery**

* There are many situations in which a transaction may not reach a commit or abort point.
  1. An operating system crash can terminate the DBMS processes
  2. The DBMS can crash
  3. The system might lose power
  4. A disk may fail or other hardware may fail.
  5. Human error can result in deletion of critical data.
* In any of these situations, data in the database may become inconsistent or lost.
* For example, if a transaction has completed 30 out of 40 scheduled writes to the database when the DBMS crashes, then the database may be in an inconsistent state as only part of the transaction’s work was completed.
* 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).
* Two approaches are discussed here: Manual Reprocessing and Automated Recovery.

1. **System Recovery**

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.

1. **Media Recovery**

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 data base recovery procedures, the potential failures must be identified and the reliability of the hardware and software must be determined.

1. **Two-phase Commit**

An error can arise when trying to write or read a file that is required to operate an Oracle database. This occurrence is called media failure because there is a physical problem reading or writing to files on the storage medium. A common example of media failure is a disk head crash, which causes the loss of all files on a disk drive. All files associated with a database are vulnerable to a disk crash, including data files, online redo log files, and control files. The appropriate recovery from a media failure depends on the files affected. However, following steps can be useful:

* Restore the database from the last backup
* Use the transaction log to redo any changes made since the last backup.
* If the transaction log is damaged then you can’t do the above step. We need to store the log on a separate physical device to the database. The risk of losing both is then reduced.

1. **SQL Facilities**

A transaction is a sequence of operations performed as a single logical unit of work. A logical unit of work must exhibit four properties, called the ACID (Atomicity, Consistency, Isolation, and Durability) properties, to qualify as a transaction:

* **Atomicity**: A transaction should be treated as a single unit of operation which means either the entire sequence of operations is successful or unsuccessful.
* **Consistency**: This represents the consistency of the referential integrity of the database, unique primary keys in tables etc.
* **Isolation**: There may be many transactions processing with the same data set at the same time, each transaction should be isolated from others to prevent data corruption.
* **Durability**: Once a transaction has completed, the results of this transaction have to be made permanent and cannot be erased from the database due to system failure.