**ST. XAVIER’S COLLEGE**

**(Affiliated to Tribhuvan University)**

Maitighar, Kathmandu



**Database Management System**

**Theory Assignment#10**

**Submitted by:**

**Shreesha Pokharel**

013BSCCSIT038

**Submitted to:**

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

Lecturer

Department of Computer Science

Date of submission: 30th October, 2015

**Transaction management**

1. **Introduction**

A transaction can be defined as a group of tasks. A single task is the minimum processing unit which cannot be divided further.

A transaction is a very small unit of a program and it may contain several low level tasks. A transaction in a database system must maintain **A**tomicity, **C**onsistency, **I**solation, and **D**urability − commonly known as ACID properties − in order to ensure accuracy, completeness, and data integrity.

* **Atomicity**: This property states that a transaction must be treated as an atomic unit, that is, either all of its operations are executed or none. There must be no state in a database where a transaction is left partially completed. States should be defined either before the execution of the transaction or after the execution/abortion/failure of the transaction.
* **Consistency**: The database must remain in a consistent state after any transaction. No transaction should have any adverse effect on the data residing in the database. If the database was in a consistent state before the execution of a transaction, it must remain consistent after the execution of the transaction as well.
* **Durability**: The database should be durable enough to hold all its latest updates even if the system fails or restarts. If a transaction updates a chunk of data in a database and commits, then the database will hold the modified data. If a transaction commits but the system fails before the data could be written on to the disk, then that data will be updated once the system springs back into action.
* **Isolation**: In a database system where more than one transaction are being executed simultaneously and in parallel, the property of isolation states that all the transactions will be carried out and executed as if it is the only transaction in the system. No transaction will affect the existence of any other transaction.

1. **Transactions**

A transaction is an event which occurs on the database. Generally a transaction reads a value from the database or writes a value to the database.

Although a transaction can both read and write on the database, there are some fundamental differences between these two classes of operations. A read operation does not change the image of the database in any way. But a write operation, whether performed with the intention of inserting, updating or deleting data from the database, changes the image of the database.

Example A transfer of money from one bank account to another requires 2 changes to the database i.e. both must succeed or Fail together. For e.g. ATM money transfer (UPDATE commands to increase/decrease Balance)

1. **Transaction recovery**

The recovery system ensures the database contains exactly those updates produced by committed transactions

-I.e. atomicity and durability, despite failures.  
Modifying the database without ensuring that the transaction will commit may leave the database in an inconsistent state. Consider transaction *Ti* that transfers $50 from account *A* to account *B*; goal is either to perform all database modifications made by *Ti* or none at all. Several output operations may be required for *Ti* (to output *A* and *B*). A failure may occur after one of these modifications have been made but before all of them are made. To ensure atomicity despite failures, we first output information describing the modifications to stable storage without modifying the database itself.

1. **System recovery**
   * The system takes checkpoints automatically
   * Upon system restart after a crash, transactions that finished successfully prior to the crash are redone, and those that were not complete prior to the crash are undone
   * REDO and UNDO logs
   * ARIES: Algorithms for Recovery and Isolation Exploiting Semantics – recovery by repeating history – REDO first, then UNDO
2. **Media recovery**

The failures in this category are caused because the media that hold the data, such as the hard disk fails. This affects the database, the media, and the transition that were being processed at the time of media failure. Media recovery, thus, is an attempt to recover from those problems.

1. **Two-phase commit**

Unlike a transaction on a local database, a distributed transaction involves altering data on multiple databases. Consequently, distributed transaction processing is more complicated, because the database must coordinate the committing or rolling back of the changes in a transaction as a self-contained unit. In other words, the entire transaction commits, or the entire transaction rolls back.

The database ensures the integrity of data in a distributed transaction using the **two-phase commit mechanism**. In the **prepare phase**, the initiating node in the transaction asks the other participating nodes to promise to commit or roll back the transaction. During the **commit phase**, the initiating node asks all participating nodes to commit the transaction. If this outcome is not possible, then all nodes are asked to roll back.

1. **SQL facilities**

In addition to the advanced facilities noted above, SQL is rich in the type of ease of use capabilities that are necessary to support relational databases from the simple to the complex.

**Table Facility:** First and foremost, SQL provides a table facility that enables a prompted, intuitive interface for the following functions:

* Defining databases
* Populating databases with rows
* Manipulating databases

**Table Editor:** SQL also provides a table editor that makes it easy for you to perform the following functions against rows in table data that is structured in row and column format:

* Access
* Insert
* Update
* Delete

**Query Facility:** With the Query facility, SQL permits you to interactively define queries and have results displayed in a variety of report formats including the following:

* Tabular
* Matrix
* Free format