**ST. XAVIER’S COLLEGE**

**MAITIGHAR, KATHMANDU**

****

**Database Management System**

**Theory Assignment #9**

**Submitted by:**

Rojesh Tamrakar

013BSCCSIT032

**Submitted to:**

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

Lecturer

Department of Computer Science

St. Xavier’s College, Maitighar

Database Recovery

In computing, **data recovery** is a process of salvaging inaccessible **data** from corrupted or damaged secondary storage, removable media or files, when the **data** they store cannot be accessed in a normal way.

1. Purpose of Data Recovery

The purpose of this policy is as follows:

* To provide secure storage for data assets critical to the work flow of official
* university business
* To prevent loss of data in the case of accidental deletion / corruption of data,
* system failure, or disaster
* To permit timely restoration of archived data in the event of a disaster or system failure

1. Types of failure

Failures may be

|  |  |
| --- | --- |
| Transaction | Caused by errors within the transaction processes. |
| System | Caused by failure of network or operating system or physical threats to the system as a whole. |
| Media | Failure of hard disk, out of memory errors, out of disk space errors. |

Failure may be caused by a number of things.

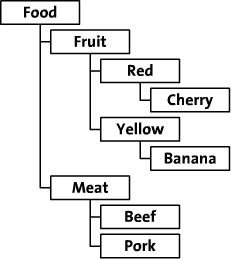
|  |  |
| --- | --- |
| A System Crash | A hardware, software or network error causes the transaction to fail. |
| Transaction or System error | Some operation in the transaction may cause the failure or the user may interrupt the transaction. |
| Local Errors or Exceptions | Conditions occur during the transaction that results in transaction cancellation. |
| Concurrency Control Enforcement | Several transactions may be in deadlock so the transaction may be aborted to be restarted later. |
| Disk Failure | Read Write error on the physical disk. |
| Physical Problems | This can be any range of physical problems, such as power failure, mounting wrong disk or tape by operator, wiring problems etc |
| Catastrophe Situations | Large scale threats to the system and the data for example fire, cyclone, security breaches etc. |

Transaction errors, system errors, system crashes, concurrency problems and local errors or exceptions are the more common causes of system failure.  The system must be able to recover from such failures without loss of data.

1. The storage Hierarchy

Storing trees is a common problem, with multiple solutions. There are two major approaches: the adjacency list model, and the modified preorder tree traversal algorithm.

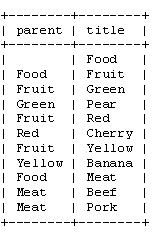
The two methods of saving hierarchical data using the tree from a fictional online food store as an example. This food store organizes its food by category, by colour and by type. The tree looks like this:



It contains a number of code examples that show how to save and retrieve data.

The Adjacency List Model

The first, and most elegant, approach we’ll try is called the ‘adjacency list model’ or the ‘recursion method’. It’s an elegant approach because you’ll need just one, simple function to iterate through your tree. In our food store, the table for an adjacency list looks like this:



1. Buffer Management

The primary purpose of a SQL Server database is to store and retrieve data, so intensive disk I/O is a core characteristic of the Database Engine. And because disk I/O operations can consume many resources and take a relatively long time to finish, SQL Server focuses on making I/O highly efficient. Buffer management is a key component in achieving this efficiency.

The buffer management component consists of two mechanisms: the buffer manager to access and update database pages, and the buffer cache (also called the buffer pool), to reduce database file I/O.

The subsystem responsible for the allocation of buffer space is called the buffer manager

The buffer manager must use some sophisticated techniques in order to provide good service:

* Pinned Blocks

For the database to be able to recover from crashes, we need to restrict times when a block maybe written back to disk. A block not allowed to be written is said to be pinned. Many operating systems do not provide support for pinned blocks, and such a feature is essential if a database is to be ``crash resistant''.

* Forced Output of Blocks

Sometimes it is necessary to write a block back to disk even though its buffer space is not needed. (Called the forced output of a block.) This is due to the fact that MM contents (and thus the buffer) are lost in a crash, while disk data usually survives.

* Replacement Strategy –

When there is no room left in the buffer, some block must be removed to make way for the new one. Typical operating system memory management schemes use a ``least recently used'' (LRU) method. (Simply remove the block least recently referenced.) This can be improved upon for database applications.

1. Transaction Log

Transactions

* A database is updated by processing *transactions* that result in changes to one or more records.
* A user’s program may carry out many operations on the data retrieved from the database, but the DBMS is only concerned with data read/written from/to the database.
* The DBMS’s abstract view of a user program is a sequence of transactions (reads and writes).

Log File

* Log file may be duplexed or triplexed.
* Log file sometimes split into two separate randomaccess files.
* Potential bottleneck; critical in determining overall performance.

1. Data Updates

**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.

1. Data Caching
2. Transaction Roll back (Undo) & Roll forward
3. Check Pointing, Shadow Paging
4. Recovery Schemes (WAL : Write Ahead Logging Protocol)
5. Failure with Loss of Non-volatile storage (General Concepts)
6. Recovery in Multi-database System