Database Recovery Techniques

2022-11-13

DATABASE RECOVERY SCENARIO

There are different types of failures encoutered by the database that is stored. A few of them are as follows

Logical Error : These are errors that can be caused due to wrong data, issues in programming or issue like items out of stock in the database of a grocery store or insufficient funds in the bank account of a customer etc. These error should be handled by the transaction.

Computer Failure : Software or hardware issues of a system can lead the main memory of the device being lost.

Physical problems or Environmental Disasters : natural disasters such as floods, earthquake etc can cause loss of data.

SCENARIO TO RECOVER THE DATABASE

Let us take an example of a “GROCERY STORE” Suppose a CUSTOMER ‘A’ goes to a grocery store and purchases all the items required and comes to the cash counter to pay his bills. The cashier scans all his items and the number of each item purchased gets deducted from the total number of items present in stock. As soon as the cashier is about to sum up total amount and finalize the the order and print out the receipt the power goes off and all the computer devices are shut down. It takes around a minute or two for the inverter to start and then all the devices are restarted. But what about the data? What about the number of items that got deducted from the items present in the stock?

There are two types of recovery techniques :

Log based recovery technique - In this technique the all the changes made in the transactions are maintained in logs.

Shadow paging - In this technique bot the data fter cmodification as well as the data after modification is stored on the disk.

LOG BASED RECOVERY TECHNIQUE WILL COME INTO PICTURE FOR THE GROCERY STORE SCENARIO

The log based recovery has two types :

Deferred Update - The transaction needs to be in a partial committed state for it to be updated in the database. If the transaction is not in a partially committed state then it cannot be updated in the databse.

Immediate Update - In this recovery the database is updated while the transaction is active.

SOLUTION

The Immediate update log based recovery technique is the one that should be the case for this particular scenario as the transaction is allowed to update the database in its active state. Hence the items that have been deducted from the stock will be updated as soon as the item gets scanned at the cash counter. The data modifications made while the transactions are active is known as uncommitted modifications.

Immediate update ensures that all the updates made by a transaction are recorded in the database on the disk before the transaction commits therefore the redo operation is not required. As in this scenario all the item values are updated in the database of the grocery store before the person finalized the order(commits) and prints out a receipt. Therefore, this recovery algorithm is known as UNDO/NO-REDO recovery algorithm. For instance if the customer decides that he wants to cancel a few items then the partial undo operation is required. If the customer decides to cancel his entire order then the undo operation can be used.

Immediate update can also allow a transaction to commit before all the changes are written to the database. This requires both the undo and redo operation and hence known as UNDO/REDO recovery algorithm. The log records both the after image as well as the before image of the database.

EXAMPLE - GROCERY STORE SCENARIO

T1 - Transaction for Customer A

T2 - Transaction for Customer B

[TA, start] – Transaction A starts

[TA, Biscuits, 1000,998] – Customer A buys 2 packs of biscuits items in the stock are reduced by 2 items

[TA, Milk, 3000,2999] – Customer A buys 1 cartoon of milk and the stock quantity afterwards is reduced by 1 item

[TA, commit] – Transaction A committed

[TB, start] – Transaction B starts

[TB, Chocolates, 500,490] – Customer B buys 10 bars of chocolates and stock quantity is reduced by 10

[TB, Chips, 100,97] – Customer B buys 3 packets of chips and stock quantity gets reduced by 3 items

[TB, commit] – Transaction B is committed

If the system fails (in this case power failure) before writing the log record [TA, commit]. Undo (TA) has to be performed. If it fails after writing the the record [TA, commit] but prior to the commit operation of [TB, commit] then redo (TA) and undo (TB) will have to be performed. If it fails after writing the [TB, commit] then redo(TA) and redo(TB) should have to be done.

The UNDO/REDO algorithm explained above is the one of the most commonly used algorithms.

REFERANCE: Chapter 11: Database Recovery System