

Module 5

Transaction Processing

Single user versus multi-user systems

A DBMS is said to be single user, if at most one user at a time can use the system and it is multiuser if many users can use the system at a time and hence access the database concurrently. Multi user systems use the concept of multi-programming

Transactions

A transaction is an executing program that forms a logical unit of database processing. A transaction includes one or more database access operations-these include insertion, deletion, modification or retrieval operations.

One way of specifying the transaction boundaries is by using explicit begin transaction and end transaction statements. Between these two statements constitutes the transaction.

If the database operation in a transaction does not update the database but only retrieve the data, then the transaction is called read-only transaction

If the database operation in a transaction updates the database, then the transaction is called read-write transaction

Database Items

A database is basically represented as a collection of named data items. A data item can be a database record but it can also be a larger unit such as a whole disk block or even a small unit such as an attribute.

The size of the data item is called its granularity

Database access operations

read-item(X) : Reads a database item named X in to a program variable

write-item(X) : write the value of variable X into the database

Why concurrency control is needed

When two or more transactions are executed concurrently, there may be a chance of happening certain problems. So we should need concurrency control software to resolve these problems. Some of the problems are

1. Lost update problem

This problem occurs when two transactions that access the same database items and their operations interleaved in a way , that make the value of some database items incorrect.

Suppose the transactions T1 and T2 are executed at the same time and their operations are interleaved in a way that the final value of X is incorrect. That is, because T2 reads the value of X before T1 write it to the database

T1	T2
read-item(X) $X=X-N$	read-item(X) $X=X+M$
Write -item(X)	

2. Temporary update problem (Dirty read problem)

This problem occurs when one transaction updates a database item and then the transaction fails for some reason, Meanwhile the updated item is accessed by another transaction before it is changed back to the original value

T1	T2
read-item(X) $X=X-N$ Write -item(X)	read-item(X) $X=X+M$ Write -item(X)

3. The incorrect summary problem

If one transaction is calculating an aggregate summary function on a number of database items while other transactions are updating some of these items, **then the aggregate function may calculate some values before they are updated and others after they are updated. This problem is called incorrect summary problem**

4. **The Unrepeatable Read Problem**

This problem occurs when a transaction T1 reads the same item twice and the item is changed by another transaction T2 between the two reads. **Hence T receive different values for its two reads of same item**

Why Recovery is needed?

Whenever a transaction is submitted to a DBMS for execution, the system is responsible for making sure that all the operations in the transactions are completed successfully and their effect is recorded permanently in the database. Then that transaction is called committed transaction

If the operations in the transactions are completed unsuccessfully and is not recorded in the database, then that transaction is aborted. These types of transactions are called aborted transactions.

Following are some of the reason for aborting transaction

1. **Computer Failure(System crash)** – Hardware, software or network error occurs in the computer system during transaction execution
2. **Transaction or System error** – some operation in the transaction may cause it to fail such as integer overflow or division by zero etc.
3. **Local errors or exception condition detected by the transaction**-During transaction execution, certain condition may occur that leads to the cancellation of the transaction. For example: data for the transaction may not be found
4. **Concurrency control enforcement**:-The concurrency control method may decide to abort a transaction because it violates serializability or cause some deadlock problems
5. **Disk failure**:-Transaction execution fails because of the disk read/write head crash
6. **Physical problems**-This refers to the problems such as power or air conditioning failure, fire, theft or flood etc.

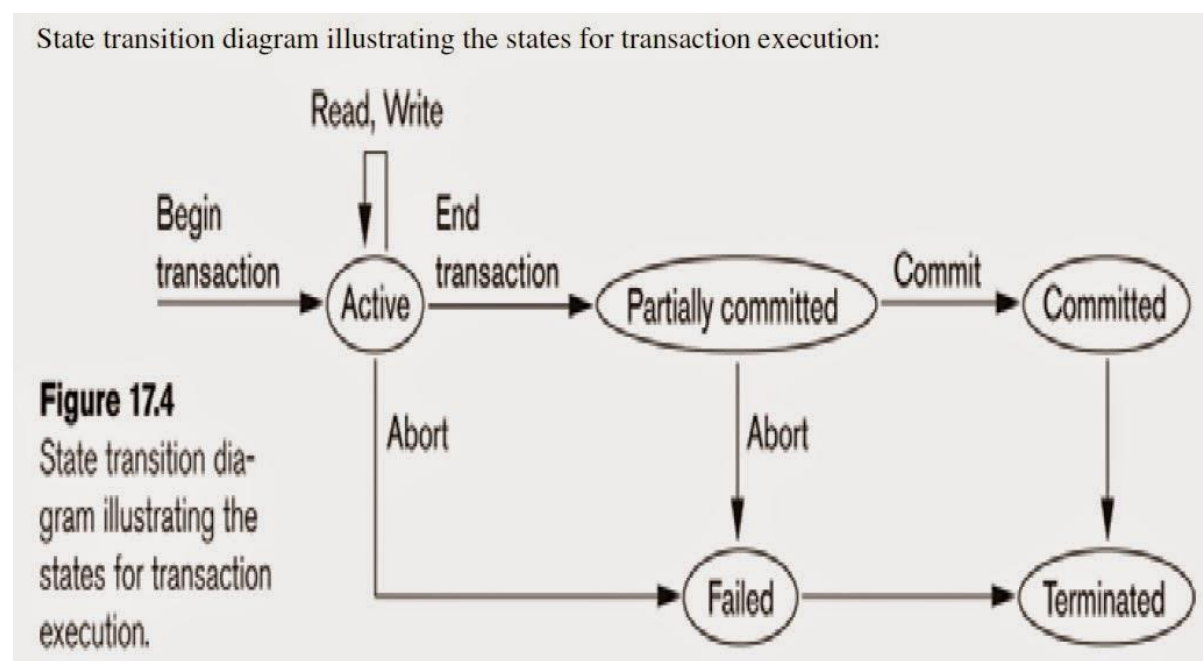
Transaction and System Concepts

1. Operations of transaction

Recovery manager of the DBMS keep track the following operations

- a. **BEGIN-TRANSACTION**-This marks the beginning of transaction execution
- b. **READ or WRITE**- it specify the read or write operations on the database items that are executed as a part of a transaction
- c. **END-TRANSACTION**- This marks the end of transaction execution
- d. **COMMIT-TRANSACTION**-This signals a successful end of a transaction so that any changes executed by the transaction can be safely committed to the database
- e. **ROLLBACK or ABORT**- This signals an unsuccessful end of a transaction so that any changes executed by the transaction will not be applied to the database

2. Transaction States



A transaction enters into **active state** immediately after it starts execution, where it can execute READ or WRITE operation. When the transaction ends, it moves to the **partially committed state**. At this point, the system checks whether the operations are executed successfully. If the transaction is executed successfully, it is reached to the **commit point** and enters in **committed state**. When the transaction is committed, it concludes its execution successfully and enters in **terminated state**.

However a transaction can go to the **failed state**, if its operations are executed unsuccessfully or if the transaction is aborted during its active state. When the transaction leaves the system, it enters in **terminated state**.

3. **System Log**

To be able to recover from failures, the system maintains a log to keep track of all transaction operations that affect the values of database items. The log is a sequential append-only file, which is kept on a disk.

Log entries are called log records. Following are the type of log entries. Here T refers to unique transaction-id, which is automatically generated by the system to identify each transaction

1. **[start-transaction, T]** – indicates that transaction T has started execution
2. **[write-item, T, X, old-value, new-value]**-indicates that the transaction T has changed the value of database item X from old-value to new-value
3. **[read-item, T, X]**- indicates that the transaction T read the value of database item X
4. **[commit, T]**-indicates that the transaction T has completed successfully
5. **[abort, T]**- indicates that the transaction T has completed unsuccessfully

4. **Commit Point**

A transaction T reaches its commit point, when all of its operations that access the database have been executed successfully and the effect of all the transaction operations have been recorded in the system log and database
After the commit point, the transaction enters in committed state

Properties of Transaction

Transaction should possess certain properties. They are called ACID properties.

ACID Properties

Atomicity:

A transaction is an atomic unit of processing. It should either be performed in its entirety or not performed at all.

It is the responsibility of the transaction recovery subsystem of DBMS to ensure atomicity

Consistency Preservation:

A transaction should be consistency preserving. That is, it should be completely executed from beginning to end without interference from other transactions

A transaction should take the database from one consistent state to another. It is the responsibility of the database programmers who write programmes which should maintain the database in consistent state

Isolation:

A transaction should be isolated from other transactions, even though many transactions are executing concurrently. That is, the execution of a transaction should not be interfered with any other transactions executing concurrently. This property is enforced by the concurrency control subsystem of the DBMS

Following are the levels of isolation of a transaction:

Level 0: it does not overwrite the dirty reads

Level 1: it has no lost updates

Level 2: no lost updates and no dirty reads

Level 3: in addition to level 2 properties it has repeatable reads. This isolation is called true isolation

Durability or Permanency:

It states that the changes applied to the database by a committed transaction must persist in the database. This property is the responsibility of the transaction recovery subsystem of DBMS

Database Security

Control Measures

There are 4 control measures used to provide the security of data in databases

- Access control
- Inference control
- Flow control
- Data encryption

Access Control:

The security mechanism of a DBMS must include provisions for restricting access to the database system. This function is called access control. Access control is done by creating user accounts and passwords to control the login process by the DBMS.

DBA is responsible for creating new user accounts and passwords for various users

Inference Control

It is a control measure used in statistical databases. Statistical databases are used to provide statistical information or summaries of values based on various criteria. These

types of databases are used in market research firms or governmental firms to get statistical information about the population

It also ensure that information about the individuals cannot be accessed

Flow control:

It prevents the information from flowing in such a way that it cannot reaches to unauthorized users. That is, the information must flow through right channels

Data Encryption:

It is used to protect sensitive or confident data. The data is encoded using some coding algorithms, so that only authorized user can access that coded data through decryption

Database security and DBA

The DBA has a DBA account in the DBMS. It is called system or super user account, which provides powerful capabilities that are not available to the ordinary user accounts.

DBA can perform the following types of actions to the user accounts

1. **Account creation**: This action creates a new account and password for each user who need database access
2. **Privilege granting**: this action grant certain privileges to user accounts
3. **Privilege Revocation**: this action cancels certain privileges that were previously given to certain user accounts
4. **Security level assignment**: assigning certain security levels to user accounts

Database Audits

To keep a record of all updates applied to the databases by a particular user, we keep the system log.

We expand the log entries so that they also include account number of the user and the device Id of the user are recorded in the log. **If any tampering with the database is suspected, a database audit is performed.**

It consists of reviewing the log entries to examine all database access operations during a certain period. When an illegal or unauthorized operation found, the DBA can determine the account number and device ID which is used to perform that operation.

A database log, that is mainly used for the security purpose is called an audit trail

Granting and Revoking of Privileges

Granting of Privileges

'GRANT' option is used by the DBA to grant privileges to the user account.

Syntax: GRANT privilege name ON object name TO user name

For example, to grant table creation permission to the user 'amen', use the following code

GRANT CREATETAB TO amen

To grant insert, delete and update privileges on the table employee to the user amen, use the code

GRANT INSERT, DELETE, UPDATE ON employee TO amen

We can also use the GRANT command with 'WITH GRANT' option

GRANT SELECT ON employee TO amen WITH GRANT

Now amen can assign privileges to another user on employee table

Revoking of Privileges

'REVOKE' command is used by the DBA to cancel the privileges from user account.

Syntax: REVOKE privilege name ON object name FROM user name

To cancel the SELECT privilege from amen on employee table use the following code

REVOKE select ON employee_name FROM amen

To cancel insert, delete and update privileges on the table employee from the user amen, use the code

REVOKE INSERT, DELETE, UPDATE ON employee FROM amen