

Assignment 4:

Compose SQL statements to BEGIN a transaction, INSERT a new record into the 'orders' table, COMMIT the transaction, then UPDATE the 'products' table, and ROLLBACK the transaction

Program:

❖ -- Begin the transaction

BEGIN TRANSACTION;

❖ --Insert a new record into the 'orders' table

INSERT INTO orders (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);

❖ --Commit the transaction

COMMIT;

❖ --Update the 'products' table

UPDATE products
SET column1 = value1, column2 = value2, ...
WHERE condition;

❖ --Rollback the transaction in case of an error

ROLLBACK;

Assignment 5:

Begin a transaction, perform a series of INSERTs into 'orders', setting a SAVEPOINT after each, rollback to the second SAVEPOINT, and COMMIT the overall transaction.

Program:

❖ --Begin the transaction

BEGIN TRANSACTION;

❖ --Perform the first INSERT operation

INSERT INTO orders (column1, column2, ...) VALUES (value1, value2, ...);

❖ --Set the first SAVEPOINT

SAVEPOINT SP1;

❖ --Perform the second INSERT operation

INSERT INTO orders (column1, column2, ...) VALUES (value3, value4, ...);

❖ --Set the second SAVEPOINT

SAVEPOINT SP2;

❖ --Perform the third INSERT operation

INSERT INTO orders (column1, column2, ...) VALUES (value5, value6, ...);

❖ --Set the third SAVEPOINT

SAVEPOINT SP3;

❖ --Rollback to the second SAVEPOINT

ROLLBACK TO SAVEPOINT SP2;

❖ --Commit the overall transaction

COMMIT;

Assignment 6:

Draft a brief report on the use of transaction logs for data recovery and create a hypothetical scenario where a transaction log is instrumental in data recovery after an unexpected shutdown.

Title: The Role of Transaction Logs in Data Recovery

Introduction: Transaction logs are a critical component in database management systems, acting as a record of all transactions that have been executed by the system. They play a vital role in ensuring data integrity and recovery processes.

Transaction Logs for Data Recovery: Transaction logs maintain a sequential record of all changes made to the data within a database. In the event of a system failure or unexpected shutdown, these logs are instrumental in recovering lost or corrupted data. By replaying the transactions recorded in the log, the database can be restored to its last consistent state before the failure occurred.

Hypothetical Scenario: Imagine a financial services company, FinCorp, that processes thousands of transactions per hour. One day, due to an unexpected power outage, their main database server shuts down abruptly. The active transactions at the time of the shutdown were not committed to the database, leading to potential data loss and inconsistencies.

Fortunately, FinCorp has a robust database management system with transaction logging enabled. Here's how the transaction log aids in recovery:

1. **Initial Assessment:** Upon system restart, the database management system checks the transaction log. It identifies the last transaction that was committed before the shutdown.
2. **Redo Phase:** The system begins the 'redo' process, where all transactions from the log that were committed before the shutdown are reapplied to the database to ensure that no committed data is lost.
3. **Undo Phase:** Next, the 'undo' process starts. Any transactions that were not committed at the time of the shutdown are rolled back, ensuring data consistency and integrity.
4. **System Back Online:** After the redo and undo phases, the database is consistent and reflects the state it was in just before the shutdown. FinCorp's operations resume without any data loss or corruption.

Conclusion: Transaction logs serve as a safety net for databases, providing a means to recover from unforeseen failures. They ensure that a database can be brought back to a consistent state, maintaining the integrity and reliability of the data.

Recommendations:

- Regularly back up transaction logs along with the database.
- Monitor transaction log size to prevent it from affecting system performance.
- Test recovery procedures periodically to ensure data can be effectively restored.