Assignment 3 :- Explain the ACID properties of a transaction in your own words. Write SQL statements to simulate a transaction that includes locking and demonstrate different isolation levels to show concurrency control.

ACID Properties:

Atomicity: Atomicity ensures that a transaction is treated as a single unit of work, meaning that either all the operations within the transaction are successfully completed, or none of them are. If any part of the transaction fails, the entire transaction is rolled back to its original state, maintaining data integrity.

Consistency: Consistency ensures that a transaction transitions the database from one consistent state to another consistent state. It ensures that the database constraints, such as foreign key constraints or unique constraints, are not violated during the transaction's execution.

Isolation: Isolation ensures that the concurrent execution of transactions does not result in interference or inconsistency. Each transaction should appear to be executed in isolation, regardless of other concurrent transactions. Isolation levels determine the degree to which transactions are isolated from each other.

Durability: Durability guarantees that once a transaction is committed, its effects persist even in the event of system failures such as power outages or crashes. The changes made by the committed transaction are stored permanently and cannot be lost.

Now, let's simulate a transaction in SQL and demonstrate different isolation levels:

SQL Statements:

-- Begin a transaction

BEGIN TRANSACTION;

-- SQL statements within the transaction

UPDATE Accounts

SET Balance = Balance - 100

WHERE AccountID = 123;

UPDATE Accounts

SET Balance = Balance + 100

WHERE AccountID = 456;

-- Commit the transaction

COMMIT;

This transaction transfers 100 units from account 123 to account 456.

Isolation Levels:

Read Uncommitted: Allows dirty reads, meaning a transaction can see changes made by other uncommitted transactions.

SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;

Read Committed: Prevents dirty reads, but allows non-repeatable reads, meaning a transaction can see changes committed by other transactions after it started.

SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

Repeatable Read: Prevents dirty reads and non-repeatable reads, but allows phantom reads, meaning a transaction may see new rows added by other transactions after it started.

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;

Serializable: Provides the highest level of isolation, preventing dirty reads, non-repeatable reads, and phantom reads by locking data aggressively.

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;

These SQL statements set different isolation levels for transactions, demonstrating how concurrency control can be managed in a database system. Depending on the isolation level, transactions may exhibit different behaviours regarding data consistency and concurrency.