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.

The ACID properties describe the key characteristics that ensure the reliability and consistency of transactions in a database system. Here's a simplified explanation of each property:

Atomicity: Atomicity ensures that a transaction is treated as a single unit of work. It means that either all the operations within the transaction are successfully completed and committed, or none of them are. If any part of the transaction fails, the entire transaction is rolled back to its original state, ensuring that the database remains consistent.

Consistency: Consistency ensures that the database remains in a valid state before and after the transaction. In other words, the integrity constraints and rules defined in the database schema are maintained throughout the transaction. Even though the data might be temporarily in an inconsistent state during the execution of a transaction, it will be restored to a consistent state upon successful completion or rollback.

Isolation: Isolation ensures that the operations within a transaction are isolated from other concurrent transactions. Each transaction should appear to be executed in isolation from other transactions, even though they may be executed concurrently. Isolation prevents interference or data corruption caused by concurrent transactions by enforcing various levels of concurrency control.

Durability: Durability guarantees that once a transaction is committed, its changes are permanently saved and will not be lost, even in the event of a system failure or crash. The changes made by a committed transaction are stored in non-volatile storage (e.g., disk), ensuring that they are preserved even if the system shuts down unexpectedly.

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

In this example, we're performing fund transfer operations within a transaction. We set the isolation level to SERIALIZABLE to ensure that the transaction is isolated from other concurrent transactions. This prevents issues like dirty reads, non-repeatable reads, and phantom reads.

Different isolation levels (e.g., READ COMMITTED, REPEATABLE READ, SERIALIZABLE) provide varying levels of concurrency control and isolation. You can experiment with different isolation levels to observe their effects on concurrent transactions and data consistency.

```
-- Simulating a transaction that transfers funds from one account to another

-- Set the isolation level to SERIALIZABLE (highest level of isolation)
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;

-- Start the transaction
BEGIN TRANSACTION;

-- Deduct amount from source account
UPDATE Accounts
SET Balance = Balance - 100
WHERE AccountNumber = 'source_account';

-- Add amount to destination account
UPDATE Accounts
SET Balance = Balance + 100
WHERE AccountNumber = 'destination_account';

-- Commit the transaction
COMMIT;
```