Debakshmi Pan

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.

Let's imagine a bank transaction:

- **Atomicity**: Imagine you're withdrawing money from one account and depositing it into another. Atomicity ensures that either both the withdrawal and deposit happen successfully, or neither happens at all. It's like buying a combo meal at a fast-food restaurant - you get both the burger and the drink together, or you get nothing.

- **Consistency**: Imagine you have $100 in your account, and you transfer $50 to another account. Consistency ensures that your total balance remains accurate, even if the transfer fails halfway through. It's like keeping track of your total score in a game - it should always reflect the correct sum of your individual scores, regardless of any interruptions.

- **Isolation**: Imagine you and your friend are making separate transactions at an ATM simultaneously. Isolation ensures that your friend's transaction doesn't interfere with yours, and vice versa. It's like having separate lanes at a toll booth - each car (transaction) moves independently, without being affected by the cars in other lanes.

- **Durability**: Imagine you've successfully deposited money into your account. Durability ensures that this transaction is permanently recorded, even if there's a power outage or system crash. It's like saving a document on your computer - once it's saved, it's there to stay, regardless of any technical issues.

Now, let's write some SQL statements for a bank transaction:

```
-- Start a transaction

START TRANSACTION;

-- SQL statements within the transaction

UPDATE accounts SET balance = balance - 50 WHERE account_id = 1;

UPDATE accounts SET balance = balance + 50 WHERE account_id = 2;


-- Commit the transaction

COMMIT;
```

And to set different isolation levels:

```
-- Set isolation level to READ COMMITTED

SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

-- Start a transaction

START TRANSACTION;

-- SQL statements within the transaction

...

-- Commit the transaction

COMMIT;
```

Replace READ COMMITTED with any other isolation level for testing different levels of concurrency control.