ACID Properties:

- **Atomicity:** A transaction is either fully executed or not executed at all. It follows the "all or nothing" rule.
- **Consistency:** After a transaction, the database remains in a consistent state. Integrity constraints are maintained.
- **Isolation:** Concurrent transactions do not interfere with each other. Changes made by one transaction are not visible to others until committed.
- **Durability:** Once a transaction is committed, its changes are permanent even in case of system failure.

SQL Statements for Transaction Simulation:

Let's consider a simple example where we transfer money from one account to another. We'll use pseudocode for demonstration:

```
-- Assume we have two accounts: Account X and Account Y
BEGIN TRANSACTION;

-- Deduct 100 from Account X
UPDATE Accounts SET Balance = Balance - 100 WHERE AccountNumber = 'X';

-- Add 100 to Account Y
UPDATE Accounts SET Balance = Balance + 100 WHERE AccountNumber = 'Y';

COMMIT; -- If successful, commit changes; otherwise, rollback
```

Locking and Isolation Levels:

- **Shared Locks (S):** Allow multiple transactions to read the same resource but not write it.
- **Exclusive Locks (X):** Allow one transaction to read and write a resource, blocking others.
- **Isolation Levels:**
  - **READ UNCOMMITTED**: Allows dirty reads (reading uncommitted data).
  - **READ COMMITTED (locking)**: Reads committed data, but locks may cause contention.
  - **READ COMMITTED (snapshot)**: Uses row versioning to avoid locks.
  - **REPEATABLE READ**: Prevents phantom reads by locking ranges.
  - **SERIALIZATION**: Highest isolation level, ensures serial execution.