Assignment 2: 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 of a transaction:

Atomicity: Atomicity ensures that a transaction is treated as a single unit of work. This means that either all the operations within the transaction are successfully completed and committed, or none of them are. There is no halfway point; it's all or nothing.

Consistency: Consistency ensures that the database remains in a valid state before and after the transaction. In other words, transactions must preserve the integrity constraints and business rules of the database. If the database is consistent before the transaction, it must remain consistent after the transaction, regardless of any failures that may occur during the transaction.

Isolation: Isolation ensures that the concurrent execution of transactions produces a result that is equivalent to some serial execution of the transactions. In simpler terms, transactions should appear to be executed in isolation from each other, even though they may be executing concurrently. Isolation prevents interference between transactions and ensures that each transaction sees a consistent snapshot of the database.

Durability: Durability ensures that once a transaction has been committed, its effects are permanent and will not be lost, even in the event of a system failure. This typically involves writing the changes made by the transaction to non-volatile storage (such as disk) so that they can be recovered if the system crashes.

SQL statements to simulate a transaction:

Start Transaction and Perform Updates:

```
mysql>
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> UPDATE account SET balance = balance - 500.00 WHERE account_id = 1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE account SET balance = balance + 500.00 WHERE account_id = 2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)
```

Isolation levels:

1. Read Uncommitted:

```
mysql>
mysql> SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
Query OK, 0 rows affected (0.00 sec)

mysql> BEGIN;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM account;
+------------+---------+
| account_id | balance |
+------------+---------+
|          1 |  500.00 |
|          2 | 2500.00 |
+------------+---------+
2 rows in set (0.00 sec)

mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)
```

2. Read Committed:

```
mysql>
mysql> SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
Query OK, 0 rows affected (0.00 sec)

mysql> BEGIN;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM account;
+------------+---------+
| account_id | balance |
+------------+---------+
|          1 |  500.00 |
|          2 | 2500.00 |
+------------+---------+
2 rows in set (0.00 sec)

mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)
```

Activate Windows

3. Repeatable Read:

```
mysql>
mysql> SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
Query OK, 0 rows affected (0.00 sec)

mysql> BEGIN;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM account;
+------------+---------+
| account_id | balance |
+------------+---------+
|          1 |  500.00 |
|          2 | 2500.00 |
+------------+---------+
2 rows in set (0.00 sec)

mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)
```

4. Serializable:

```
mysql>
mysql> SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
Query OK, 0 rows affected (0.00 sec)

mysql> BEGIN;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM account;
+------------+---------+
| account_id | balance |
+------------+---------+
|          1 |  500.00 |
|          2 | 2500.00 |
+------------+---------+
2 rows in set (0.00 sec)

mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)
```

Activate Windows