Team 29
Swati Sharma(2021568), Siddhant Rai Viksit(2021565)

Alldeez

Schedule 1: Conflict Serializable

Transaction 1:

A user adds a new product to the products table and updates their profile in the users table.

1. Start transaction
2. INSERT INTO products (categoryid, sellerid, productname, Price, productdesc) VALUES (1, 1, 'New Product', 50, 'This is a new product');
3. UPDATE users SET fullname='New Name', useraddress='New Address' WHERE userid=1;
4. COMMIT transaction

Operation Type	Operation
START TRANSACTION;	
Write	INSERT INTO products (categoryid, sellerid, productname, Price, productdesc) VALUES (1, 1, 'New Product', 50, 'This is a new product');
Write	UPDATE users SET fullname='New Name', useraddress='New Address' WHERE userid=1;
COMMIT TRANSACTION;	

Transaction 2:

Another user purchases a product and leaves a review for the product they purchased.

1. Start transaction
2. INSERT INTO carts (productid, orderid) VALUES (1, 1);

3. UPDATE orders SET haspaid=1, delivererid=1 WHERE orderid=1;
4. INSERT INTO reviews (productid, userid, Rating) VALUES (1, 2, 4);
5. COMMIT transaction

Operation Type	Operation
START TRANSACTION;	
Write	INSERT INTO carts (productid, orderid) VALUES (1, 1);
Write	UPDATE orders SET haspaid=1, delivererid=1 WHERE orderid=1;
Write	INSERT INTO reviews (productid, userid, Rating) VALUES (1, 2, 4);
COMMIT TRANSACTION;	

In this schedule, the operations are conflict serializable because transaction 2 waits for transaction 1 to finish before it starts, and the order of the operations within each transaction does not matter.

Schedule 2: Non-Conflict Serializable

Transaction 1:

A user adds a new product to the products table and updates their profile in the users table.

1. Start transaction
2. INSERT INTO products (categoryid, sellerid, productname, Price, productdesc) VALUES (1, 1, 'New Product', 50, 'This is a new product');
3. UPDATE users SET fullname='New Name', useraddress='New Address' WHERE userid=1;
4. COMMIT transaction

Operation Type	Operation
START TRANSACTION;	
Write	INSERT INTO products (categoryid, sellerid, productname, Price, productdesc) VALUES (1, 1, 'New Product', 50, 'This is a new product');
Write	UPDATE users SET fullname='New Name', useraddress='New Address' WHERE userid=1;
COMMIT TRANSACTION;	

Transaction 2:

Another user purchases a product and leaves a review for the product they purchased.

1. Start transaction
2. INSERT INTO carts (productid, orderid) VALUES (1, 1);
3. SELECT * FROM products WHERE productid=1;
4. UPDATE orders SET haspaid=1, delivererid=1 WHERE orderid=1;
5. INSERT INTO reviews (productid, userid, Rating) VALUES (1, 2, 4);
6. COMMIT transaction

Operation Type	Operation
START TRANSACTION;	
Write	INSERT INTO carts (productid, orderid) VALUES (1, 1);
Read	SELECT * FROM products WHERE productid=1;
Write	UPDATE orders SET haspaid=1, delivererid=1 WHERE orderid=1;
Write	INSERT INTO reviews (productid, userid, Rating) VALUES (1, 2, 4);

COMMIT TRANSACTION;	

In this schedule, transaction 2 reads from the products table before transaction 1 commits, which means that the order of the transactions cannot be switched. Therefore, this schedule is non-conflict serializable.