Project Deadline 6: DBMS Winter-2024

Amartya Singh (2022062) Kartik Prasad (2022240) Anisha Bopardikar (2022078)

April 20, 2024

1 Non-Conflicting Transactions

1.1 Transaction 1: Add a new supplier, add a new product, and add a new drug from that supplier to the inventory

This transaction is adding a new supplier, a new product, and a new inventory record for that product supplied by the new supplier. It's a series of `INSERT` operations that add new records to the `Suppliers`, `Products`, and `Inventory` tables.

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
LOCK TABLES Suppliers WRITE, Products WRITE, Inventory WRITE;
START TRANSACTION;
INSERT INTO Suppliers (SupplierID, Name, Address, City, Province, PostalCode)
VALUES (101, 'MediSupply', '123 Health St', 'New York', 'NY', '10001');
INSERT INTO Products (ProductID, Product_Name, Average_Price, Total_Quantity, Recent_date_of_expiry)
VALUES (201, 'Aspirin', 10.99, 1000, '2025-12-31');
INSERT INTO Inventory (Inv_ID, ProductID, Drug_Name, Drug_Description, Quantity_on_Hand,
Price, Supplier_ID, Recent_date_of_expiry)
VALUES (301, 201, 'Aspirin', 'Pain reliever', 1000, 10.99, 101, '2025-12-31');
COMMIT;
UNLOCK TABLES;
```

1.2 Transaction 2: Add a new product and add it to the inventory with the price set by the supplier

This transaction is adding a new product and a new inventory record for that product with the price set by the supplier. It's a series of `INSERT` operations that add new records to the `Products` and `Inventory` tables.

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
LOCK TABLES Products WRITE, Inventory WRITE;
START TRANSACTION;
INSERT INTO Products (ProductID, Product_Name, Average_Price, Total_Quantity, Recent_date_of_expiry)
VALUES (202, 'Ibuprofen', 1, 1, '2025-12-31');
INSERT INTO Inventory (Inv_ID, ProductID, Drug_Name, Drug_Description, Quantity_on_Hand,
Price, Supplier_ID, Recent_date_of_expiry)
VALUES (302, 202, 'Ibuprofen', 'Pain reliever', 1000, 9.99, 101, '2025-12-31');
COMMIT;
UNLOCK TABLES;
```

1.3 Transaction 3: Add a new prescription, update inventory, add a new order, and place the order with the supplier

This transaction is adding a new prescription, updating the quantity on hand in the inventory for a specific product, and adding a new order for that prescription. It involves `INSERT` operations to add new records to the `Prescription` and `Orders` tables, an `UPDATE` operation to modify a record in the `Inventory` table, and an `INSERT` operation to add a record to the `PlaceOrder` table linking the order to the supplier.

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
LOCK TABLES Prescription WRITE, Inventory WRITE, Orders WRITE, PlaceOrder WRITE;
START TRANSACTION;
INSERT INTO Prescription (Prescription_No, Date_of_Prescription, Patient_ID, Doctor_ID,
Drug_Name, Dosage)
VALUES (401, NOW(), 5, 6, 'Aspirin', 'Take one pill daily');
UPDATE Inventory SET Quantity_on_Hand = 400 WHERE ProductID = 201;
INSERT INTO Orders (OrderID, Prescription_ID, CustomerNumber, Date_of_order)
VALUES (301, 401, 5, NOW());
INSERT INTO PlaceOrder (OrderID, SupplierID)
VALUES (301, 101);
COMMIT;
UNLOCK TABLES;
```

1.4 Transaction 4: Add a new order, place the order with the supplier, and update the related customer's address

This transaction is adding a new order, placing the order with the supplier, and updating the address of the related customer. It involves `INSERT` operations to add new records to the `Orders` and `PlaceOrder` tables, and an `UPDATE` operation to modify a record in the `Customers` table.

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
LOCK TABLES Orders WRITE, PlaceOrder WRITE, Customers WRITE;
START TRANSACTION;
INSERT INTO Orders (OrderID, Prescription_ID, CustomerNumber, Date_of_order)
VALUES (302, 4, 5, NOW());
INSERT INTO PlaceOrder (OrderID, SupplierID)
VALUES (302, 101);
UPDATE Customers SET AddressLine1 = '456 Wellness Ave' WHERE CustomerNumber = 5;
COMMIT;
UNLOCK TABLES;
```

2 Conflicting Transactions

2.1 Transaction 5: Attempt to delete a supplier while updating their related inventory

This transaction is attempting to delete a supplier while another transaction is trying to update the inventory of the deleted supplier. It involves a `DELETE` operation to remove a record from the `Suppliers` table and an `UPDATE` operation to modify a record in the `Inventory` table.

```
START TRANSACTION;
DELETE FROM Suppliers WHERE SupplierID = 101;
-- Meanwhile, another transaction tries to update the inventory of the deleted supplier
START TRANSACTION;
UPDATE Inventory SET Quantity_on_Hand = 200 WHERE Supplier_ID = 101;
-- This will cause a conflict as one transaction is trying to delete a record while another is trying to
COMMIT;
```

2.2 Transaction 6: Attempt to read a supplier while another transaction is updating the same supplier

This transaction is attempting to read a supplier while another transaction is trying to update the same supplier. It involves a `SELECT` operation to read a record from the `Suppliers` table and an `UPDATE` operation to modify the same record.

```
START TRANSACTION;
SELECT * FROM Suppliers WHERE SupplierID = 101;
-- Meanwhile, another transaction tries to update the same supplier
START TRANSACTION;
UPDATE Suppliers SET Name = ’NewSupplierName’ WHERE SupplierID = 101;
-- This will cause a conflict as one transaction is trying to read a record while another is trying to up
COMMIT;
```