# Shopify - The Online Grocery Store Database Transactions

Name: Nishant Singh (solo project)

Roll Number: 2022328

## Non Conflicting Transactions:

**Transaction 1: Update Product Price**

```
-- Before Transaction --
SELECT * FROM Product WHERE ProductID = 1;

-- Transaction --
START TRANSACTION;
UPDATE Product SET Price = 25.99 WHERE ProductID = 1;
COMMIT;

-- After Transaction --
SELECT * FROM Product WHERE ProductID = 1;
```

**Transaction 2: Add New Product**

```
-- Before Transaction --
SELECT COUNT(*) FROM Product;

-- Transaction --
START TRANSACTION;
INSERT INTO Product (Name, Brand, Description, Price, QuantityInStock,
Category, SubCategory)
VALUES ('New Product', 'Brand X', 'Description of New Product', 15.99, 100,
'Electronics', 'Smartphones');
COMMIT;

-- After Transaction --
SELECT COUNT(*) FROM Product;
```

### Transaction 3: Update User Password

```
-- Before Transaction --
SELECT * FROM User WHERE UserID = 1;

-- Transaction --
START TRANSACTION;
UPDATE User SET Password = 'NewPassword123' WHERE UserID = 1;
COMMIT;

-- After Transaction --
SELECT * FROM User WHERE UserID = 1;
```

### Transaction 4: Delete Payment Record

```
-- Before Transaction --
SELECT * FROM Payment;

-- Transaction --
START TRANSACTION;
DELETE FROM Payment WHERE PaymentID = 1;
COMMIT;

-- After Transaction --
SELECT * FROM Payment;
```

## Conflicting Transactions:

### Conflicting Transaction 1: Update Product Price (Conflicting), First establish 2 connections to db

**In Connection 1:**

```
-- Before Transaction --
SELECT * FROM Product WHERE ProductID = 1;

-- Transaction 1 --
START TRANSACTION;
UPDATE Product SET Price = 30.99 WHERE ProductID = 1;
```

**Switch to Connection 2:**

```
-- Transaction 2 --
START TRANSACTION;
UPDATE Product SET Price = 35.99 WHERE ProductID = 1;
```

**Observe waiting time in Connection 2 because transaction 1 is still uncommitted and conflicting with transaction 2**

**Switch to Connection 1:**

```
-- Commit Transaction 1 --
Commit;
```

**We observe Transaction 2 is now complete and Transaction 1 is committed (changes done)**

```
SELECT * FROM Product WHERE ProductID = 1;
```

**Switch to Connection 2:**

```
-- Commit Transaction 2 --
Commit;
```

**Transaction 2 is committed (changes done)**

```
SELECT * FROM Product WHERE ProductID = 1;
```

**Conflicting Transaction 2: Update User Password (Conflicting), First establish 2 connections to db**

**In Connection 1:**

```
-- Before Transaction --
SELECT * FROM User WHERE UserID = 1;

-- Transaction 1 --
START TRANSACTION;
UPDATE User SET Password = 'NewPass123' WHERE UserID = 1;
```

**Switch to Connection 2:**

```
-- Transaction 2 --
START TRANSACTION;
UPDATE User SET Password = 'AnotherNewPass123' WHERE UserID = 1;
```

**Observe waiting time in Connection 2 because transaction 1 is still uncommitted and conflicting with transaction 2**

**Switch to Connection 1:**

```
-- Commit Transaction 1 --
Commit;
```

**We observe Transaction 2 is now complete and Transaction 1 is committed (changes done)**

```
SELECT * FROM User WHERE UserID = 1;
```

**Switch to Connection 2:**

```
-- Commit Transaction 2 --
Commit;
```

**Transaction 2 is committed (changes done)**

```
SELECT * FROM User WHERE UserID = 1;
```