TRANSACTIONS:-

NON-CONFLICTING TRANSACTIONS:-

1.

START TRANSACTION;

UPDATE user_details SET Contact_Number = '9998887776' WHERE UserName = 'Sneha Sharma';

UPDATE inventory SET Quantity = Quantity - 2 WHERE ProductID = 2;

UPDATE order_user SET Status_Order = 'D' WHERE OrderID = 2 ;

COMMIT;

2.

START TRANSACTION;

UPDATE inventory SET Quantity = Quantity + 10 WHERE ProductID = 3;

UPDATE order_user SET Status_Order = 'RD' WHERE ProductID = 3 ;

COMMIT;

3.

Start Transaction;

INSERT INTO User_Details (UserName, Passwords, Contact_Number, Email_ID, Address, City, State, Pincode)
VALUES ('Aditri', 'password1', '8098787654', 'aditri@iiitd.ac.in', 'dwarka sector 10', 'Delhi', 'Delhi', '110075');
commit;

4.

Given below in Conflicting to show the difference.

CONFLICTING TRANSACTIONS:

```
import threading
import pymysql
import time;

def transaction_one():
    connection = pymysql.connect(host='localhost', user='root',
password='root', db='styleu')
    try:
        with connection.cursor() as cursor:
            cursor.execute("START TRANSACTION;")
            cursor.execute("UPDATE inventory SET Quantity = Quantity - 1
WHERE ProductID = 2;")
            cursor.execute("COMMIT;")
    finally:
        connection.close()
```

```
    print("Transaction One Completed")

def transaction_two():
    connection = pymysql.connect(host='localhost', user='root',
password='root', db='styleu')
    try:
        with connection.cursor() as cursor:
            cursor.execute("START TRANSACTION;")
            cursor.execute("UPDATE inventory SET Quantity = Quantity - 2
WHERE ProductID = 2;")
            cursor.execute("COMMIT;")
    finally:
        connection.close()
    print("Transaction Two Completed")


def transaction_three():
    connection = pymysql.connect(host='localhost', user='root',
password='root', db='styleu')
    try:
        with connection.cursor() as cursor:
            cursor.execute("START TRANSACTION;")
            cursor.execute("UPDATE user_details SET Email_ID =
'sneha@update.com' WHERE UserName = 'Sneha Sharma';")
            cursor.execute("COMMIT;")
    finally:
        connection.close()
    print("Transaction three Completed")


thread_one = threading.Thread(target=transaction_one)
thread_two = threading.Thread(target=transaction_two)
thread_three = threading.Thread(target=transaction_three)


thread_one.start()
time.sleep(2)
thread_two.start()
thread_three.start()
```

```
thread_one.join()
thread_two.join()
thread_three.join()
```

Explanation :-

Here transaction_one and transaction_two are conflicting while three is not.

We run the transaction_one first after which a wait time of two seconds is introduced. Now in this wait time transaction three always finishes before transaction 2 cause transaction 2 waits for the lock to be released by transaction 1. (or however the database ensures the locking of conflicting transactions.)

This shows that non conflicting transactions can proceed uninterrupted while the conflicting transactions wait for the previous transaction to finish.