|  |  |
| --- | --- |
| Name: | Harverinder Singh |
| UID: | 23BCS13630 |
| Subject: | ADBMS |
| Section: | 622-B |

Ans 4.3:

**Code:**

CREATE TABLE StudentEnrollments ( student\_id INT PRIMARY KEY, student\_name VARCHAR(100), course\_id VARCHAR(10), enrollment\_date DATE

);

INSERT INTO StudentEnrollments (student\_id, student\_name, course\_id, enrollment\_date)

VALUES

(1, 'Ashish', 'CSE101', '2024-06-01'),

(2, 'Smaran', 'CSE102', '2024-06-01'),

(3, 'Vaibhav', 'CSE103', '2024-06-01');

#Part A

START TRANSACTION;

-- Step 1: Lock row with student\_id = 1

UPDATE StudentEnrollments

SET course\_id = 'CSE201'

WHERE student\_id = 1;

-- Step 2: Later tries to lock student\_id = 2

UPDATE StudentEnrollments

SET course\_id = 'CSE301'

WHERE student\_id = 2;

#Part B

START TRANSACTION;

-- Step 1: Lock row with student\_id = 2

UPDATE StudentEnrollments

SET course\_id = 'CSE202'

WHERE student\_id = 2;

-- Step 2: Later tries to lock student\_id = 1

UPDATE StudentEnrollments

SET course\_id = 'CSE302'

WHERE student\_id = 1;

#Part B: Applying MVCC to Prevent Conflicts

--Transaction 1 (User A - Reader)

START TRANSACTION ISOLATION LEVEL REPEATABLE READ;

-- Reads snapshot data

SELECT student\_id, student\_name, course\_id, enrollment\_date

FROM StudentEnrollments

WHERE student\_id = 1;

--**Transaction 2 (User B - Writer)**

START TRANSACTION;

-- Updates same row

UPDATE StudentEnrollments

SET enrollment\_date = '2024-07-10'

WHERE student\_id = 1;

COMMIT;

# Part C: Comparing Locking vs MVCC

START TRANSACTION;

SELECT \* FROM StudentEnrollments WHERE student\_id = 1 FOR UPDATE;

UPDATE StudentEnrollments

SET course\_id = 'CSE401'

WHERE student\_id = 1;

START TRANSACTION;

SELECT \* FROM StudentEnrollments WHERE student\_id = 1;

-- This is BLOCKED until T1 commits

**Scenario 2: MVCC (Snapshot Isolation)**

**Transaction 1 (Writer):**

START TRANSACTION;

UPDATE StudentEnrollments

SET course\_id = 'CSE402'

WHERE student\_id = 1;

-- Not committed yet

**Transaction 2 (Reader):**

START TRANSACTION ISOLATION LEVEL REPEATABLE READ;

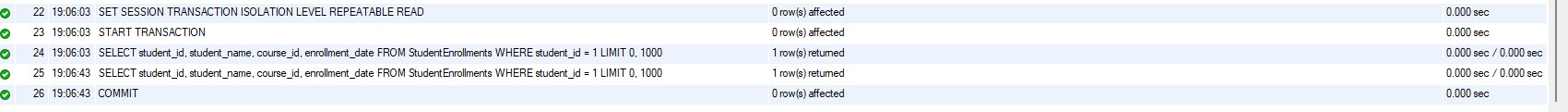
SELECT \* FROM StudentEnrollments WHERE student\_id = 1;

Output:

(A)

 **The error “Lost connection to MySQL server during query” occurs because Tab 1 tried to update a row that was already locked by Tab 2.**

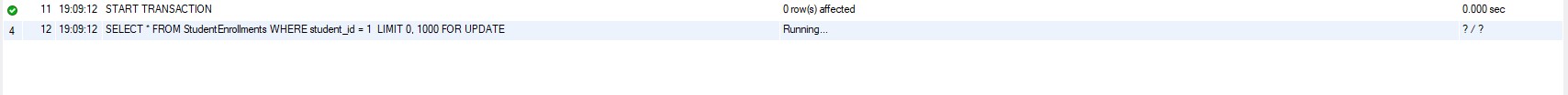
**PART B:**

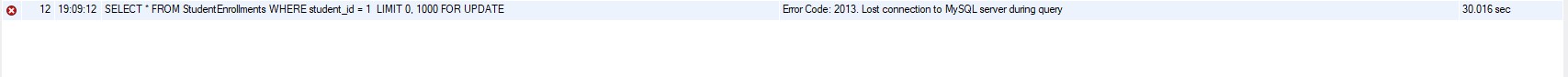


PART C:

**Scenario 1**

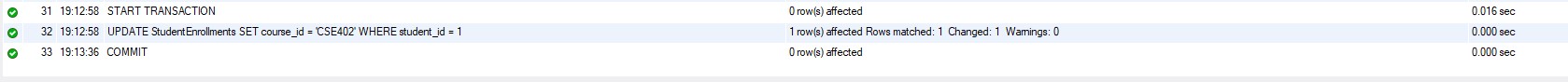
Session A (Transaction 1 - Locks row)





**Scenario 2**

Session A(Writer)



Session B(reader)

