Transactions, Locks, and Constraints in MySQL

Aim

To demonstrate the behavior of UNIQUE constraints, row-level locking, and transaction blocking in MySQL using the StudentEnrollments table.

SQL Code

```
-- Step 1: Create Table with UNIQUE Constraint
CREATE TABLE StudentEnrollments (
    enrollment_id INT PRIMARY KEY,
    student_name VARCHAR(100) NOT NULL,
    course_id VARCHAR(10) NOT NULL,
    enrollment_date DATE NOT NULL,
    UNIQUE (student_name, course_id)
);

-- Step 2: Insert a valid enrollment
INSERT INTO StudentEnrollments (enrollment_id, student_name, course_id, enrollment_date)
VALUES (1, 'Ashish', 'CSE101', '2024-07-01');

-- Step 3: Try inserting the same student in same course again (fails)
INSERT INTO StudentEnrollments (enrollment_id, student_name, course_id, enrollment_date)
VALUES (2, 'Ashish', 'CSE101', '2024-07-01');
-- UNIQUE constraint violation

-- Step 4: Transaction with row lock
START TRANSACTION;
SELECT * FROM StudentEnrollments WHERE student_name = 'Ashish' AND course_id = 'CSE101'
FOR UPDATE;

-- Another session (User B) trying to update same row will wait
UPDATE StudentEnrollments SET enrollment_date = '2024-07-02'
WHERE student_name = 'Ashish' AND course_id = 'CSE101';

-- Step 5: Concurrent locking scenario
START TRANSACTION;
SELECT * FROM StudentEnrollments WHERE enrollment_id = 1 FOR UPDATE;
UPDATE StudentEnrollments SET enrollment_date = '2024-07-03' WHERE enrollment_id = 1;
-- Do not commit yet

-- Another session tries to lock same row
START TRANSACTION;
```

```
SELECT * FROM StudentEnrollments WHERE enrollment_id = 1 FOR UPDATE;
UPDATE StudentEnrollments SET enrollment_date = '2024-07-04' WHERE enrollment_id = 1;
-- This will wait until User A commits or rollbacks

-- Step 6: Final commit for both sessions
COMMIT;  -- User A
COMMIT;  -- User B

-- Step 7: Verify final result
SELECT * FROM StudentEnrollments WHERE enrollment_id = 1;
```

Expected Output

enrollment_id | student_name | course_id | enrollment_date

---

1 | Ashish | CSE101 | 2024-07-01 -> 2024-07-03 -> 2024-07-04

Observations:

1. The UNIQUE constraint prevents duplicate enrollment of the same student in the same course.
2. FOR UPDATE locks a row until the transaction commits/rolls back.
3. Other sessions attempting updates must wait (block) until the lock is released.
4. This ensures consistency and prevents dirty writes.