

Name:	Aditi Tyagi
UID:	23BCS10641
Subject:	ADBMS
Section:	622-A

### Part A: Prevent Duplicate Enrollments Using Locking

```

CREATE TABLE StudentEnrollments (
    enrollment_id INT IDENTITY(1,1) PRIMARY KEY,
    student_name VARCHAR(100) NOT NULL,
    course_id VARCHAR(10) NOT NULL,
    enrollment_date DATE NOT NULL,
    CONSTRAINT UQ_StudentCourse UNIQUE (student_name, course_id) -- Prevent
duplicate enrollments
);

INSERT INTO StudentEnrollments (student_name, course_id, enrollment_date)
VALUES
    ('Ashish', 'CSE101', '2024-07-01'),
    ('Smaran', 'CSE102', '2024-07-01'),
    ('Vaibhav', 'CSE101', '2024-07-01');

BEGIN TRANSACTION;

BEGIN TRY
    INSERT INTO StudentEnrollments (student_name, course_id, enrollment_date)
    VALUES ('Ashish', 'CSE101', '2024-07-01');

```

```
COMMIT TRANSACTION;

PRINT 'User A: Enrollment succeeded.';

END TRY

BEGIN CATCH

    ROLLBACK TRANSACTION;

    PRINT 'User A: Enrollment failed due to duplicate entry.';

    PRINT ERROR_MESSAGE();

END CATCH
```

```
BEGIN TRANSACTION;

BEGIN TRY

    INSERT INTO StudentEnrollments (student_name, course_id, enrollment_date)
    VALUES ('Ashish', 'CSE101', '2024-07-01');

    COMMIT TRANSACTION;

    PRINT 'User B: Enrollment succeeded.';

END TRY

BEGIN CATCH

    ROLLBACK TRANSACTION;

    PRINT 'User B: Enrollment failed due to duplicate entry.';

    PRINT ERROR_MESSAGE();

END CATCH

SELECT * FROM StudentEnrollments;
```

Output:

User A: Enrollment failed due to duplicate entry.

Violation of UNIQUE KEY constraint 'UQ\_StudentCourse', Cannot insert duplicate key in object 'dbo.StudentEnrollments'. The duplicate key value is (Ashish, CSE101).

User B: Enrollment failed due to duplicate entry.

Violation of UNIQUE KEY constraint 'UQ\_StudentCourse', Cannot insert duplicate key in object 'dbo.StudentEnrollments'. The duplicate key value is (Ashish, CSE101).

enrollment_id	student_name	course_id	enrollment_date
1	Ashish	CSE101	2024-07-01
2	Smaran	CSE102	2024-07-01
3	Vaibhav	CSE101	2024-07-01

## Part B: Use SELECT FOR UPDATE to Lock Student Record

```
IF OBJECT_ID('StudentEnrollments', 'U') IS NOT NULL
    DROP TABLE StudentEnrollments;
```

```
CREATE TABLE StudentEnrollments (
    enrollment_id INT IDENTITY(1,1) PRIMARY KEY,
    student_name VARCHAR(100) NOT NULL,
    course_id VARCHAR(10) NOT NULL,
    enrollment_date DATE NOT NULL,
    CONSTRAINT UQ_StudentCourse UNIQUE (student_name, course_id)
);
```

```
INSERT INTO StudentEnrollments (student_name, course_id, enrollment_date)
VALUES
    ('Ashish', 'CSE101', '2024-07-01'),
    ('Smaran', 'CSE102', '2024-07-01'),
    ('Vaibhav', 'CSE101', '2024-07-01');
BEGIN TRANSACTION;
```

```
SELECT * FROM StudentEnrollments WITH (ROWLOCK, UPDLOCK)
WHERE student_name = 'Ashish' AND course_id = 'CSE101';
```

```
BEGIN TRANSACTION;
UPDATE StudentEnrollments
SET enrollment_date = '2024-08-15'
WHERE student_name = 'Ashish' AND course_id = 'CSE101';
```

```
COMMIT;
```

Output:

enrollment_id	student_name	course_id	enrollment_date
1	Ashish	CSE101	2024-07-01

## Part C: Demonstrate Locking Preserving Consistency in Concurrent Transactions

```
IF OBJECT_ID('StudentEnrollments', 'U') IS NOT NULL
    DROP TABLE StudentEnrollments;
```

```
CREATE TABLE StudentEnrollments (
    enrollment_id INT IDENTITY(1,1) PRIMARY KEY,
    student_name VARCHAR(100) NOT NULL,
    course_id VARCHAR(10) NOT NULL,
    enrollment_date DATE NOT NULL,
    CONSTRAINT UQ_StudentCourse UNIQUE (student_name, course_id)
);
```

```
INSERT INTO StudentEnrollments (student_name, course_id, enrollment_date)
VALUES ('Ashish', 'CSE101', '2024-07-01');
```

```
BEGIN TRANSACTION;
UPDATE StudentEnrollments
SET enrollment_date = '2024-08-01'
WHERE student_name = 'Ashish' AND course_id = 'CSE101';
COMMIT;

BEGIN TRANSACTION;
UPDATE StudentEnrollments
SET enrollment_date = '2024-09-01'
WHERE student_name = 'Ashish' AND course_id = 'CSE101';
COMMIT;
```

```
SELECT * FROM StudentEnrollments;
```

Output:

enrollment_id	student_name	course_id	enrollment_date
1	Ashish	CSE101	2024-09-01