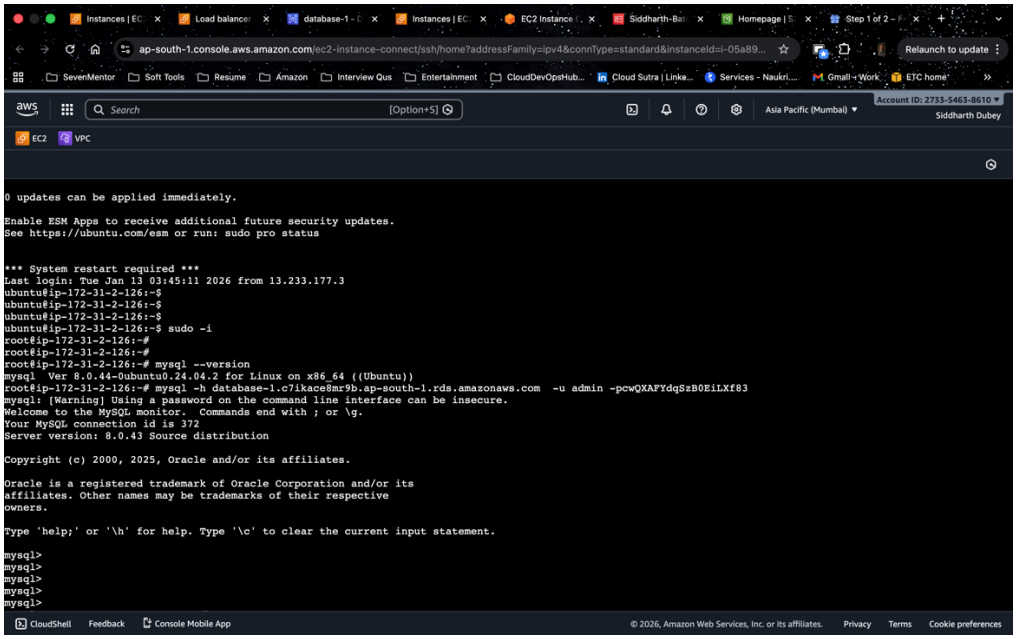
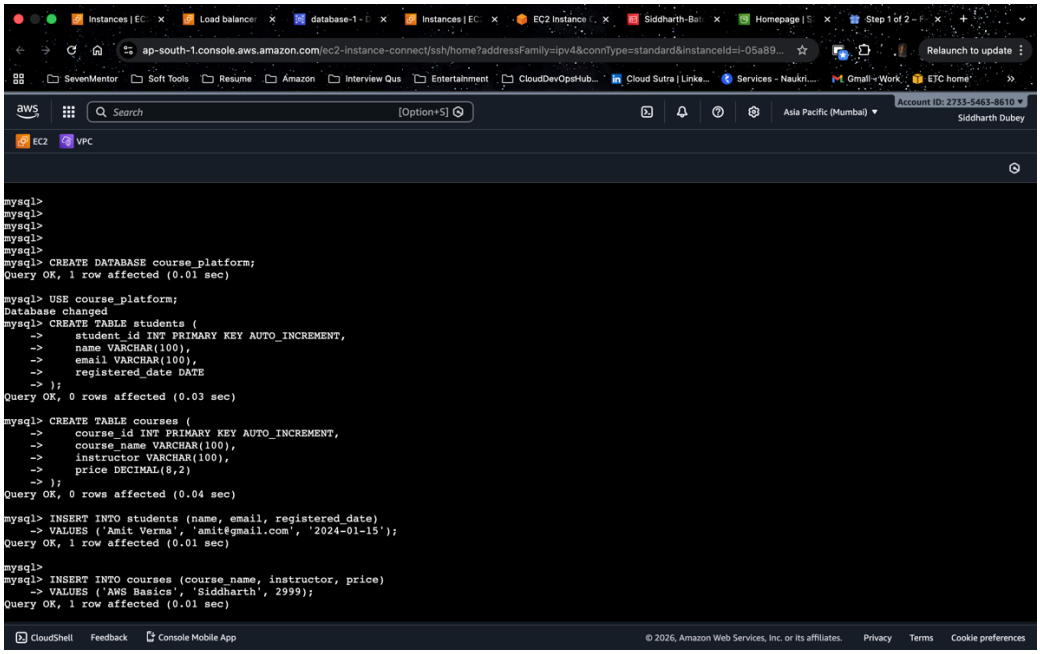
Practical 7: EC2 SQL Proxy + RDS MySQL Connection via Endpoint

Minor Project 1: Online Course Management System

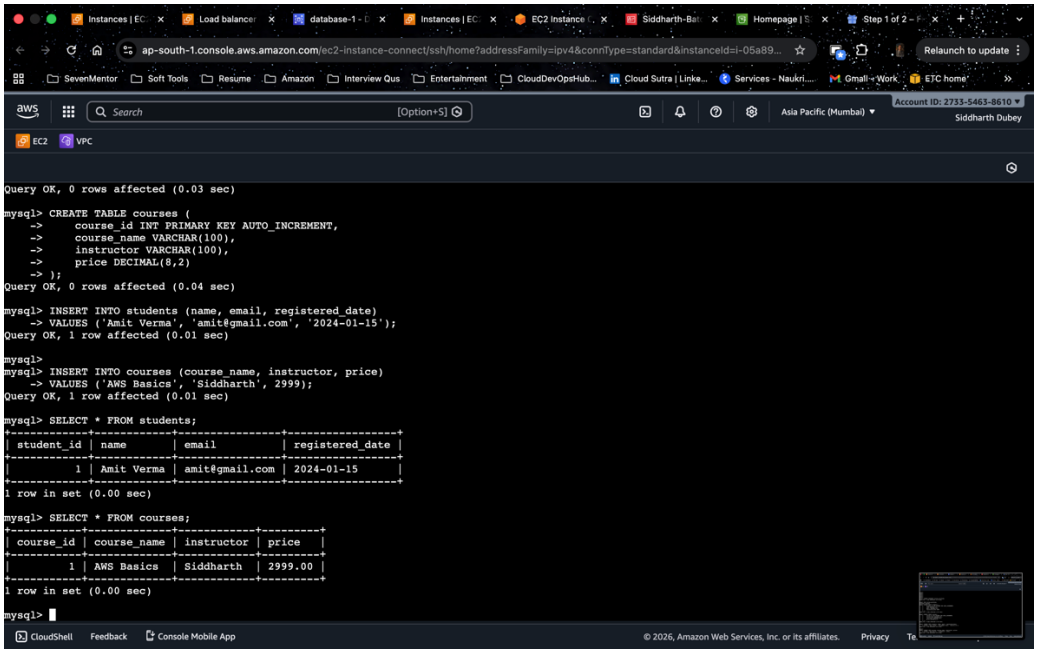
0. Setup RDS and MySQL Database



1. CREATE DATABASE
2. USE
3. CREATE TABLE – Students
4. CREATE TABLE – Courses
5. INSERT

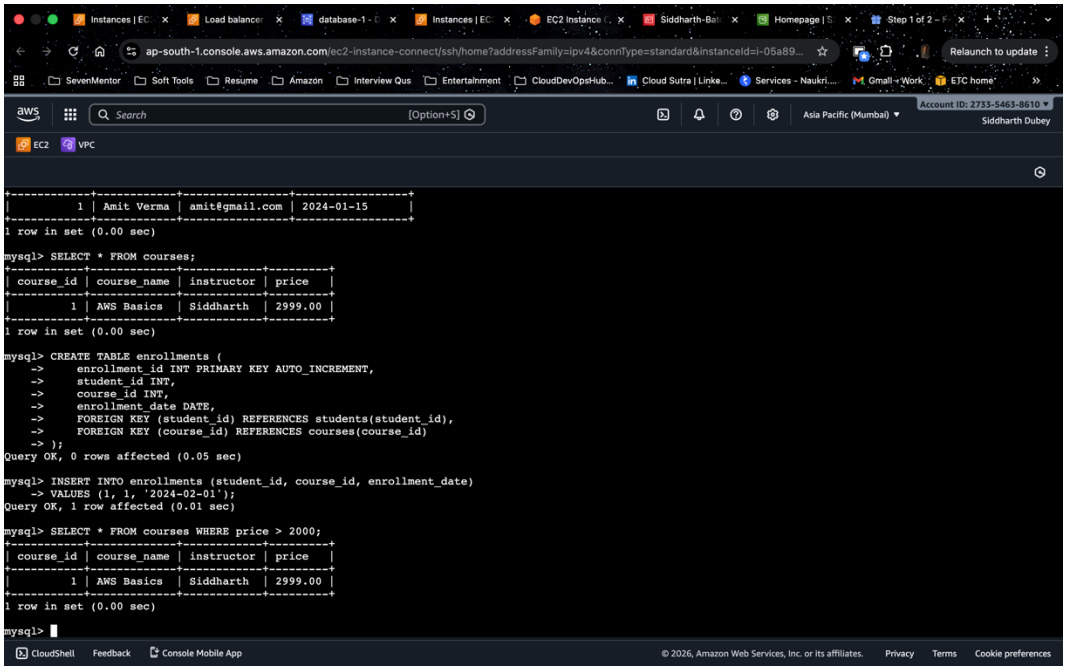


6. SELECT



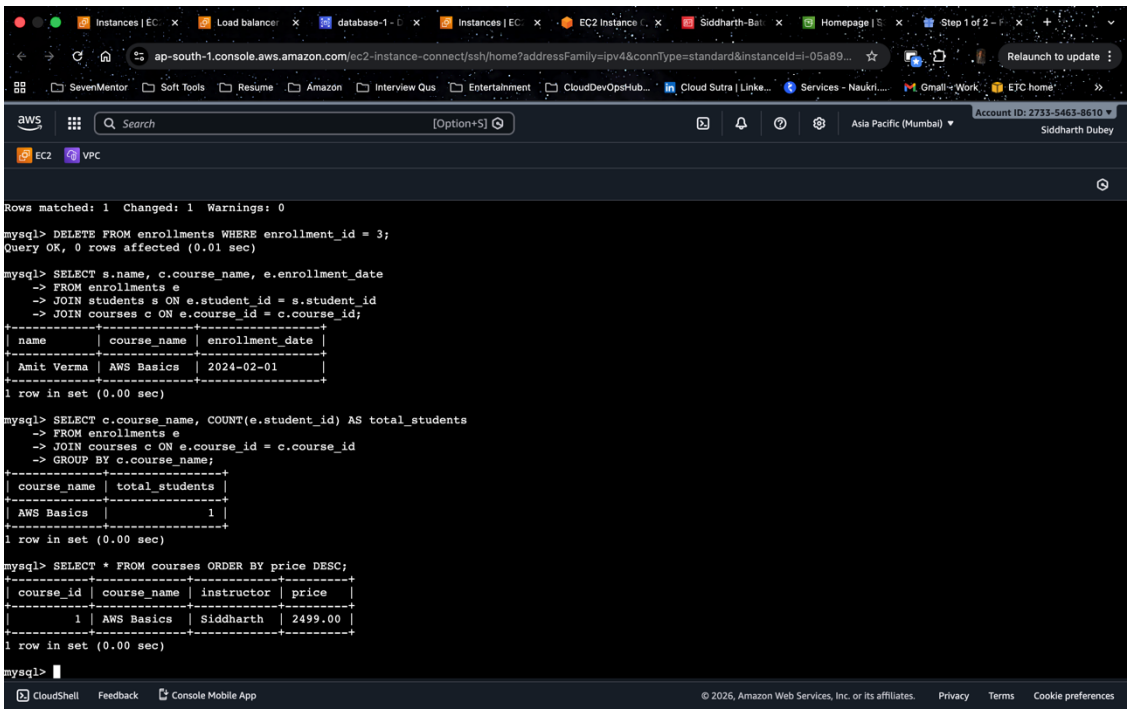
```
Query OK, 0 rows affected (0.03 sec)

mysql> CREATE TABLE courses (
    ->     course_id INT PRIMARY KEY AUTO_INCREMENT,
    ->     course_name VARCHAR(100),
    ->     instructor VARCHAR(100),
    ->     price DECIMAL(8,2)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> INSERT INTO students (name, email, registered_date)
    -> VALUES ('Amit Verma', 'amit@gmail.com', '2024-01-15');
Query OK, 1 row affected (0.01 sec)

mysql>
mysql> INSERT INTO courses (course_name, instructor, price)
    -> VALUES ('AWS Basics', 'Siddharth', 2999);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM students;
+------------+------------+----------------+-----------------+
| student_id | name       | email          | registered_date |
+------------+------------+----------------+-----------------+
|          1 | Amit Verma | amit@gmail.com | 2024-01-15      |
+------------+------------+----------------+-----------------+
1 row in set (0.00 sec)

mysql> SELECT * FROM courses;
+-----------+-------------+------------+---------+
| course_id | course_name | instructor | price   |
+-----------+-------------+------------+---------+
|         1 | AWS Basics  | Siddharth  | 2999.00 |
+-----------+-------------+------------+---------+
1 row in set (0.00 sec)

mysql>
```

7. CREATE TABLE – Enrollments (Relationship)
8. INSERT with Foreign Key
9. WHERE
10. UPDATE



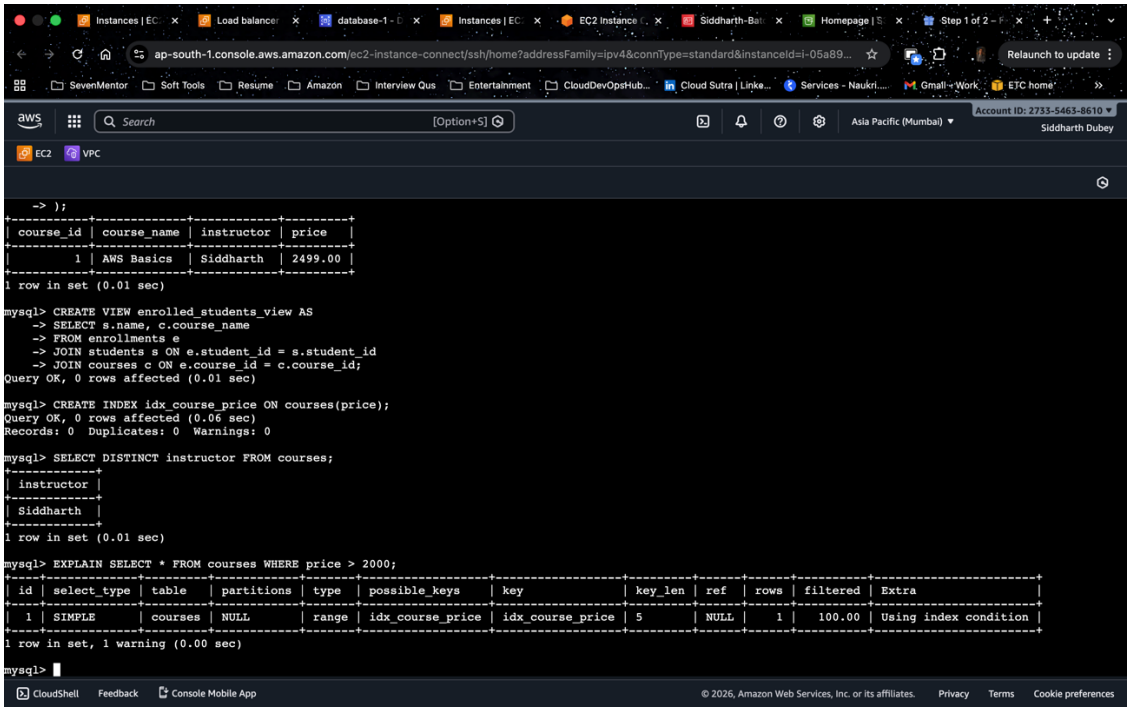
```
+------------+------------+----------------+-----------------+
|          1 | Amit Verma | amit@gmail.com | 2024-01-15      |
+------------+------------+----------------+-----------------+
1 row in set (0.00 sec)

mysql> SELECT * FROM courses;
+-----------+-------------+------------+---------+
| course_id | course_name | instructor | price   |
+-----------+-------------+------------+---------+
|         1 | AWS Basics  | Siddharth  | 2999.00 |
+-----------+-------------+------------+---------+
1 row in set (0.00 sec)

mysql> CREATE TABLE enrollments (
    ->     enrollment_id INT PRIMARY KEY AUTO_INCREMENT,
    ->     student_id INT,
    ->     course_id INT,
    ->     enrollment_date DATE,
    ->     FOREIGN KEY (student_id) REFERENCES students(student_id),
    ->     FOREIGN KEY (course_id) REFERENCES courses(course_id)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> INSERT INTO enrollments (student_id, course_id, enrollment_date)
    -> VALUES (1, 1, '2024-02-01');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM courses WHERE price > 2000;
+-----------+-------------+------------+---------+
| course_id | course_name | instructor | price   |
+-----------+-------------+------------+---------+
|         1 | AWS Basics  | Siddharth  | 2999.00 |
+-----------+-------------+------------+---------+
1 row in set (0.00 sec)

mysql>
```

11. DELETE
12. JOIN
13. COUNT
14. ORDER BY



```
Rows matched: 1  Changed: 1  Warnings: 0

mysql> DELETE FROM enrollments WHERE enrollment_id = 3;
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT s.name, c.course_name, e.enrollment_date
    -> FROM enrollments e
    -> JOIN students s ON e.student_id = s.student_id
    -> JOIN courses c ON e.course_id = c.course_id;
+-------------+-------------+-----------------+
| name        | course_name | enrollment_date |
+-------------+-------------+-----------------+
| Amit Verma  | AWS Basics  | 2024-02-01      |
+-------------+-------------+-----------------+
1 row in set (0.00 sec)

mysql> SELECT c.course_name, COUNT(e.student_id) AS total_students
    -> FROM enrollments e
    -> JOIN courses c ON e.course_id = c.course_id
    -> GROUP BY c.course_name;
+-------------+----------------+
| course_name | total_students |
+-------------+----------------+
| AWS Basics  |              1 |
+-------------+----------------+
1 row in set (0.00 sec)

mysql> SELECT * FROM courses ORDER BY price DESC;
+-----------+-------------+------------+---------+
| course_id | course_name | instructor | price   |
+-----------+-------------+------------+---------+
|         1 | AWS Basics  | Siddharth  | 2499.00 |
+-----------+-------------+------------+---------+
1 row in set (0.00 sec)

mysql>
```

15. LIMIT
16. SUBQUERY
17. VIEW
18. INDEX
19. DISTINCT
20. EXPLAIN



```
    -> );
+-----------+-------------+------------+---------+
| course_id | course_name | instructor | price   |
+-----------+-------------+------------+---------+
|         1 | AWS Basics  | Siddharth  | 2499.00 |
+-----------+-------------+------------+---------+
1 row in set (0.01 sec)

mysql> CREATE VIEW enrolled_students_view AS
    -> SELECT s.name, c.course_name
    -> FROM enrollments e
    -> JOIN students s ON e.student_id = s.student_id
    -> JOIN courses c ON e.course_id = c.course_id;
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE INDEX idx_course_price ON courses(price);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> SELECT DISTINCT instructor FROM courses;
+------------+
| instructor |
+------------+
| Siddharth  |
+------------+
1 row in set (0.01 sec)

mysql> EXPLAIN SELECT * FROM courses WHERE price > 2000;
+----+-------------+---------+------------+-------+-----------------+-----------------+---------+------+------+----------+-----------------------+
| id | select_type | table   | partitions | type  | possible_keys   | key             | key_len | ref  | rows | filtered | Extra                 |
+----+-------------+---------+------------+-------+-----------------+-----------------+---------+------+------+----------+-----------------------+
|  1 | SIMPLE      | courses | NULL       | range | idx_course_price | idx_course_price | 5       | NULL |    1 |   100.00 | Using index condition |
+----+-------------+---------+------------+-------+-----------------+-----------------+---------+------+------+----------+-----------------------+
1 row in set, 1 warning (0.00 sec)

mysql>
```

- Designed relational database schema for students, courses, and enrollments
- Implemented CRUD operations and relational joins
- Generated reports using GROUP BY, COUNT, and subqueries
- Optimized queries using indexing and EXPLAIN
- Created views for simplified data access
- **Tech Used:** MySQL | SQL | Relational Database Design