**All the databases are created using Mysql**

**Task1** :

**Project Title: Academic Management System**

Project Description:

Design and develop an Academic Management System using SQL. The projects should involve

three tables 1.StudentInfo 2. CoursesInfo 3.EnrollmentInfo. The Aim is to create a system that

allows for managing student information and course enrollment. The project will include the

following tasks:

**1.Database Creation:**

a)Create the StudentInfo table with columns STU\_ ID, STU\_NAME, DOB, PHONE\_NO,

EMAIL\_ID,ADDRESS.

- CREATE TABLE StudentInfo (

STU\_ID INT PRIMARY KEY,

STU\_NAME VARCHAR(100),

DOB DATE,

PHONE\_NO VARCHAR(15),

EMAIL\_ID VARCHAR(100),

ADDRESS VARCHAR(255)

);

b)Create the CoursesInfo table with columns COURSE\_ID,

COURSE\_NAME,COURSE\_INSTRUCTOR NAME.

CREATE TABLE CoursesInfo (

COURSE\_ID INT PRIMARY KEY,

COURSE\_NAME VARCHAR(100),

COURSE\_INSTRUCTOR\_NAME VARCHAR(100)

);

c)Create the EnrollmentInfo with columns ENROLLMENT\_ID, STU\_ ID, COURSE\_ID,

ENROLL\_STATUS(Enrolled/Not Enrolled). The FOREIGN KEY constraint in the EnrollmentInfo

table references the STU\_ID column in the StudentInfo table and the COURSE\_ID column in the CoursesInfo table.

CREATE TABLE EnrollmentInfo (

ENROLLMENT\_ID INT PRIMARY KEY,

STU\_ID INT,

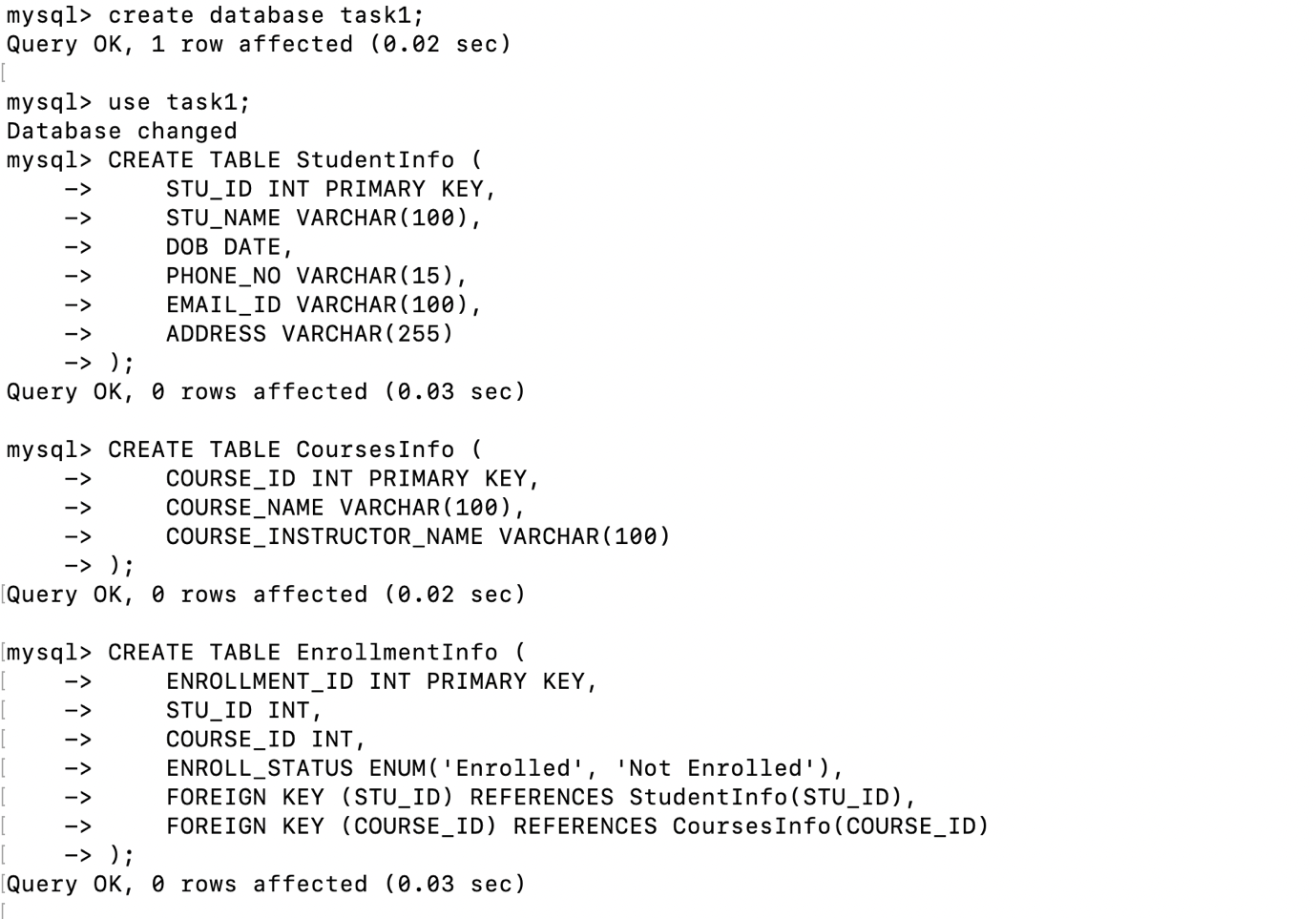
COURSE\_ID INT,

ENROLL\_STATUS ENUM('Enrolled', 'Not Enrolled'),

FOREIGN KEY (STU\_ID) REFERENCES StudentInfo(STU\_ID),

FOREIGN KEY (COURSE\_ID) REFERENCES CoursesInfo(COURSE\_ID)

);



2.Data Creation:

Insert some sample data for StudentInfo table , CoursesInfo table, EnrollmentInfo with

respective fields.

INSERT INTO StudentInfo (STU\_ID, STU\_NAME, DOB, PHONE\_NO, EMAIL\_ID, ADDRESS)

VALUES

(1, 'Ruchika Ajmera', '2000-01-15', '123-456-7890', 'ruchika.ajmera@example.com', '123 Street'),

(2, 'Ruchi ', '1999-02-22', '234-567-8901', 'Ruchi@example.com', 'whitefield'),

(3, 'Ruch Ajm', '2001-03-30', '345-678-9012', 'ruchajm@example.com', 'xyz road');



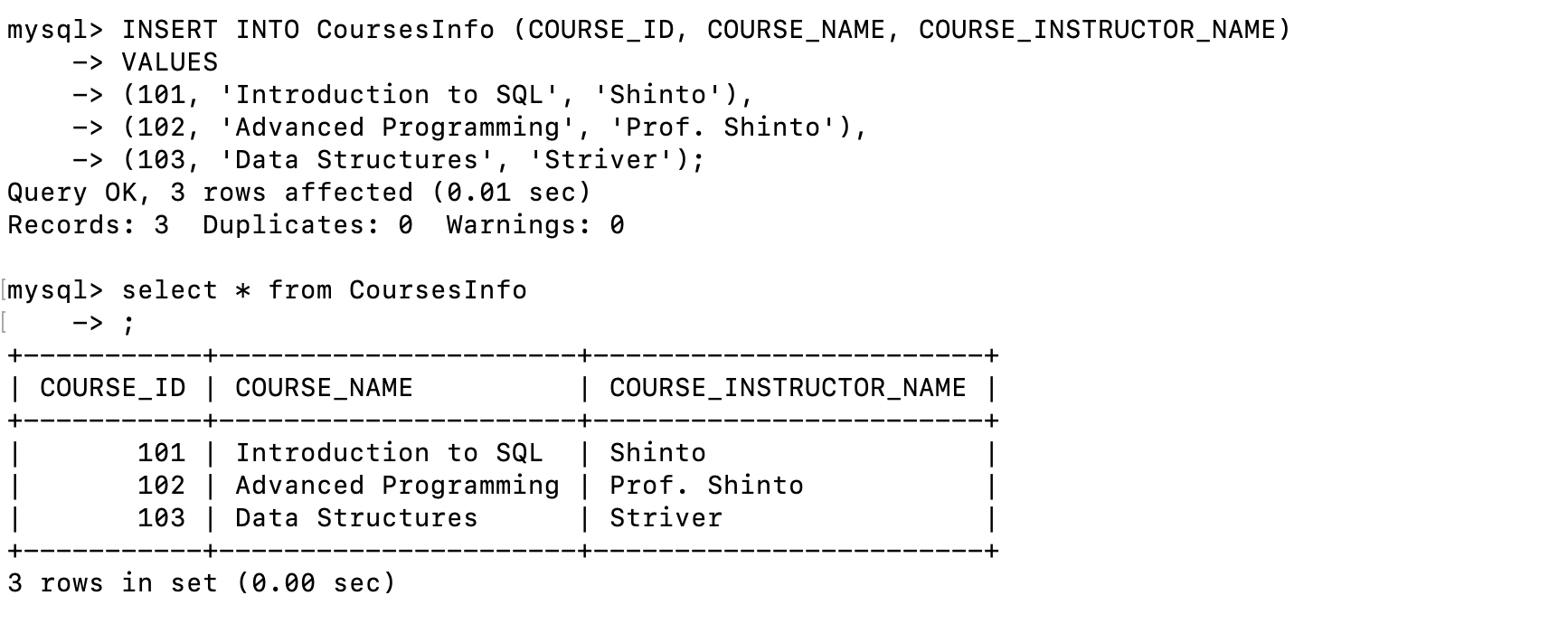
INSERT INTO CoursesInfo (COURSE\_ID, COURSE\_NAME, COURSE\_INSTRUCTOR\_NAME)

VALUES

(101, 'Introduction to SQL', 'Shinto'),

(102, 'Advanced Programming', 'Prof. Shinto'),

(103, 'Data Structures', 'Striver');



INSERT INTO EnrollmentInfo (ENROLLMENT\_ID, STU\_ID, COURSE\_ID, ENROLL\_STATUS)

VALUES

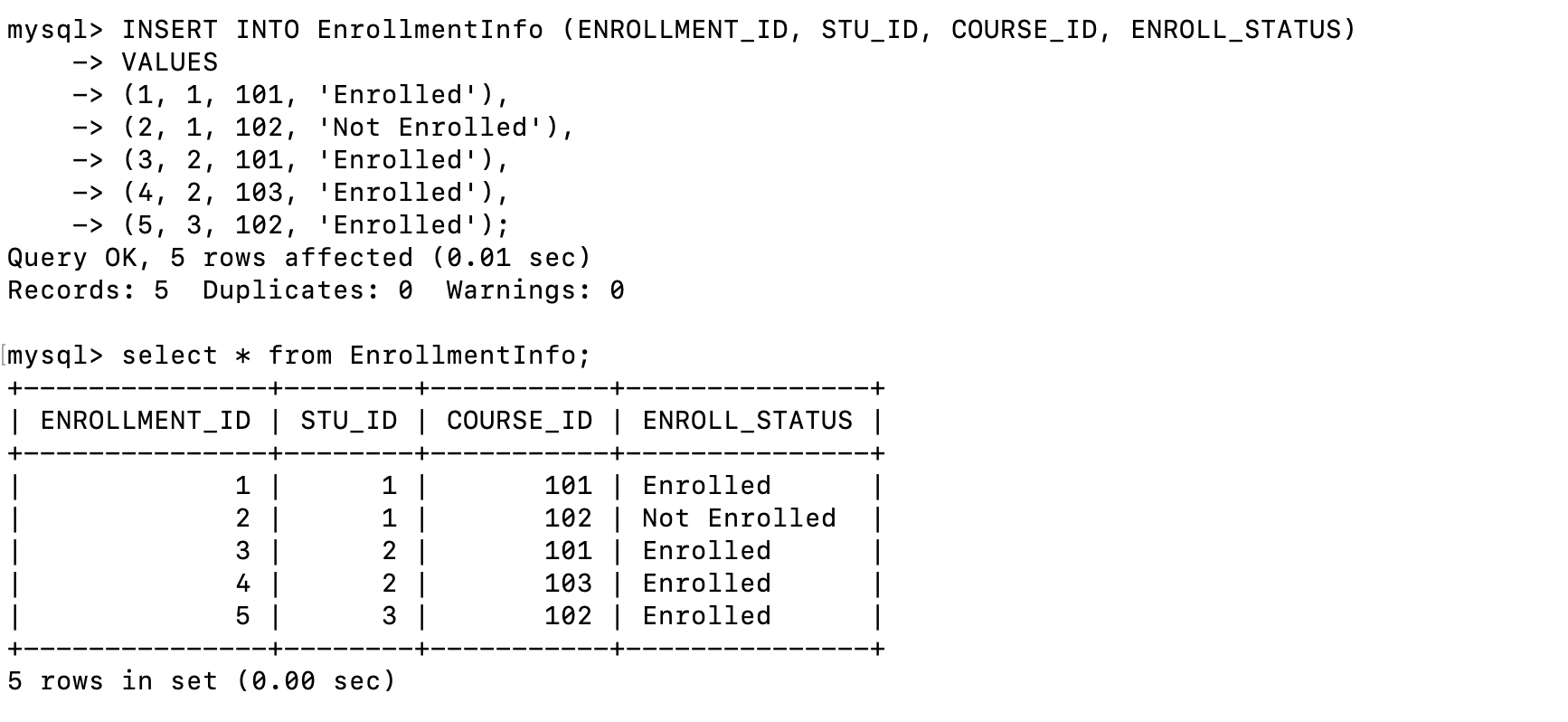
(1, 1, 101, 'Enrolled'),

(2, 1, 102, 'Not Enrolled'),

(3, 2, 101, 'Enrolled'),

(4, 2, 103, 'Enrolled'),

(5, 3, 102, 'Enrolled');



3) Retrieve the Student Information

a) Write a query to retrieve student details, such as student name, contact informations, and

Enrollment status.

SELECT s.STU\_NAME, s.PHONE\_NO, s.EMAIL\_ID, e.ENROLL\_STATUS

FROM StudentInfo s

JOIN EnrollmentInfo e ON s.STU\_ID = e.STU\_ID;

b) Write a query to retrieve a list of courses in which a specific student is enrolled.

SELECT c.COURSE\_NAME

FROM CoursesInfo c

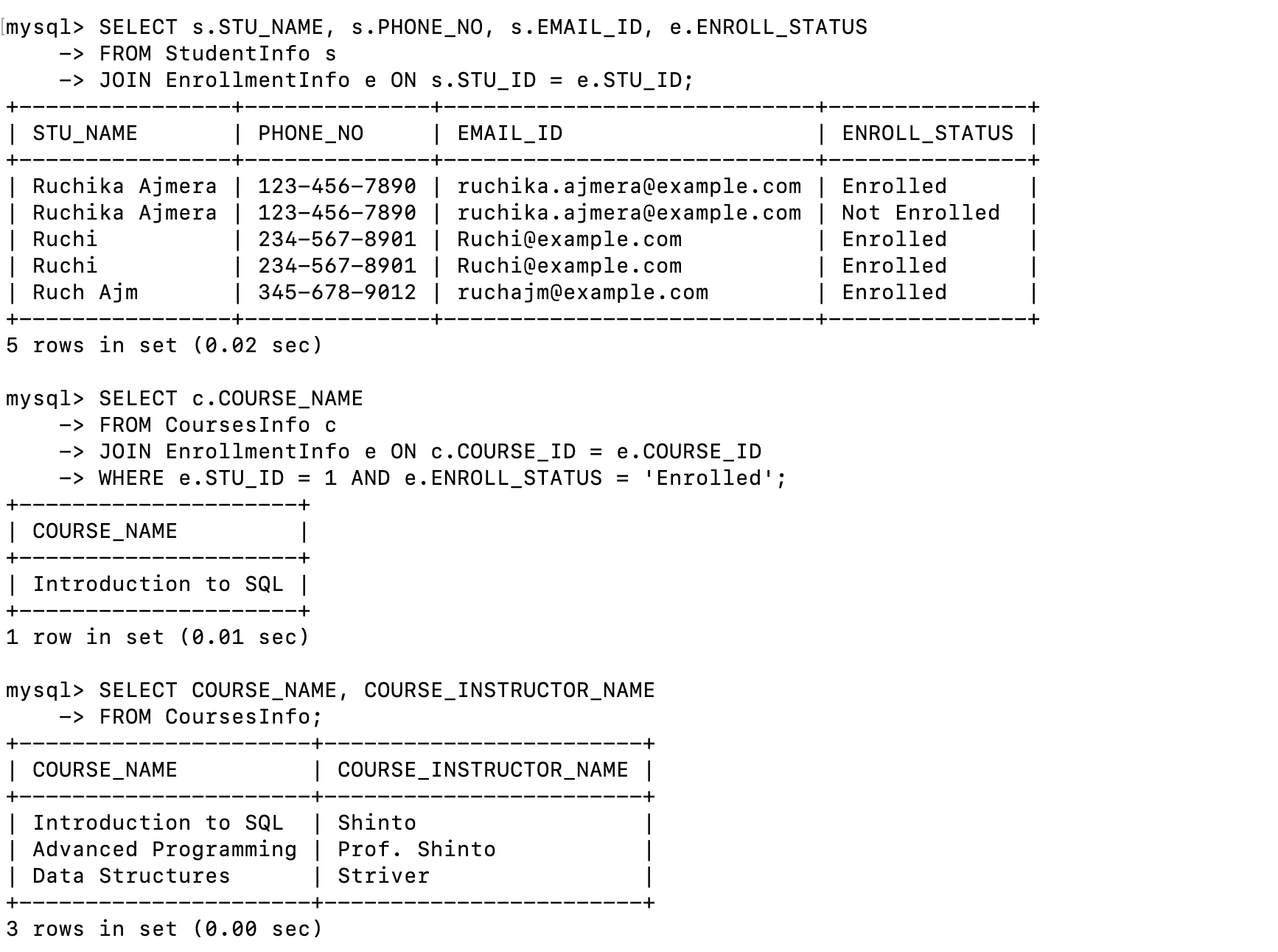
JOIN EnrollmentInfo e ON c.COURSE\_ID = e.COURSE\_ID

WHERE e.STU\_ID = 1 AND e.ENROLL\_STATUS = 'Enrolled';

c) Write a query to retrieve course information, including course name, instructor information.

SELECT COURSE\_NAME, COURSE\_INSTRUCTOR\_NAME

FROM CoursesInfo;



d) Write a query to retrieve course information for a specific course .

SELECT COURSE\_NAME, COURSE\_INSTRUCTOR\_NAME

FROM CoursesInfo

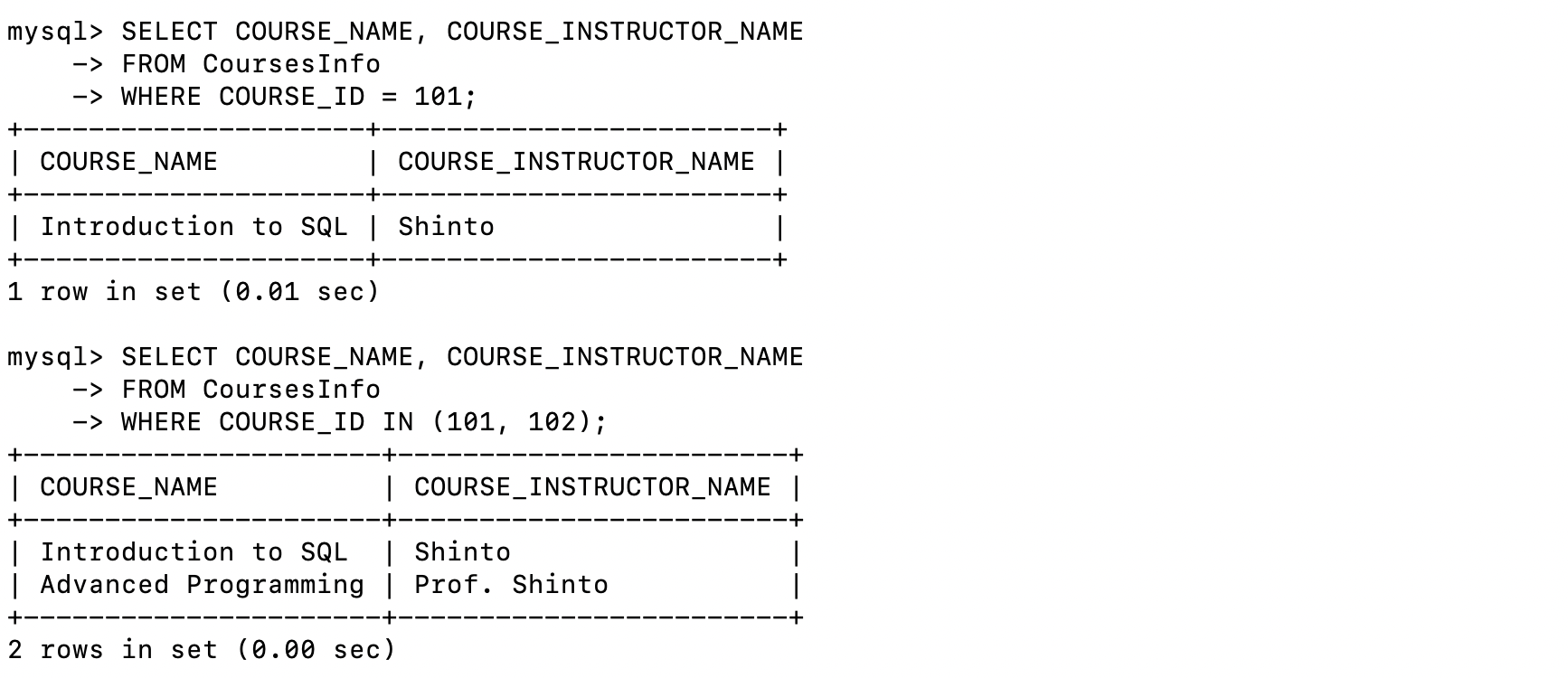
WHERE COURSE\_ID = 101;

e) Write a query to retrieve course information for multiple courses.

SELECT COURSE\_NAME, COURSE\_INSTRUCTOR\_NAME

FROM CoursesInfo

WHERE COURSE\_ID IN (101, 102);



4. Reporting and Analytics (Using joining queries)

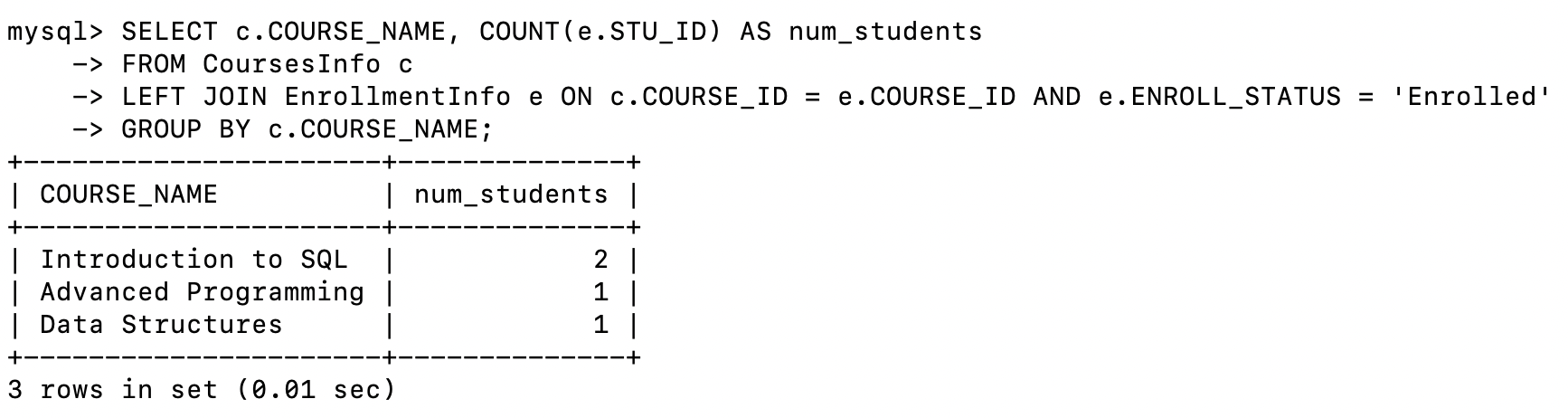
a) Write a query to retrieve the number of students enrolled in each course

SELECT c.COURSE\_NAME, COUNT(e.STU\_ID) AS num\_students

FROM CoursesInfo c

LEFT JOIN EnrollmentInfo e ON c.COURSE\_ID = e.COURSE\_ID AND e.ENROLL\_STATUS = 'Enrolled'

GROUP BY c.COURSE\_NAME;



b) Write a query to retrieve the list of students enrolled in a specific course

SELECT s.STU\_NAME

FROM StudentInfo s

JOIN EnrollmentInfo e ON s.STU\_ID = e.STU\_ID

WHERE e.COURSE\_ID = 101 AND e.ENROLL\_STATUS = 'Enrolled';



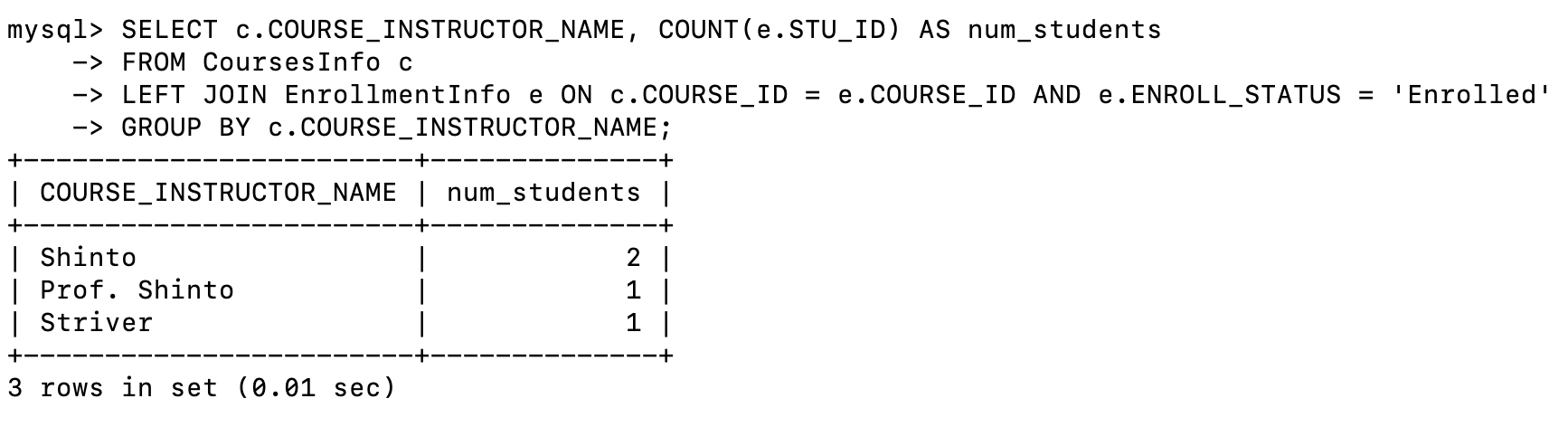
c) Write a query to retrieve the count of enrolled students for each instructor.

SELECT c.COURSE\_INSTRUCTOR\_NAME, COUNT(e.STU\_ID) AS num\_students

FROM CoursesInfo c

LEFT JOIN EnrollmentInfo e ON c.COURSE\_ID = e.COURSE\_ID AND e.ENROLL\_STATUS = 'Enrolled'

GROUP BY c.COURSE\_INSTRUCTOR\_NAME;



d) Write a query to retrieve the list of students who are enrolled in multiple courses

SELECT s.STU\_NAME

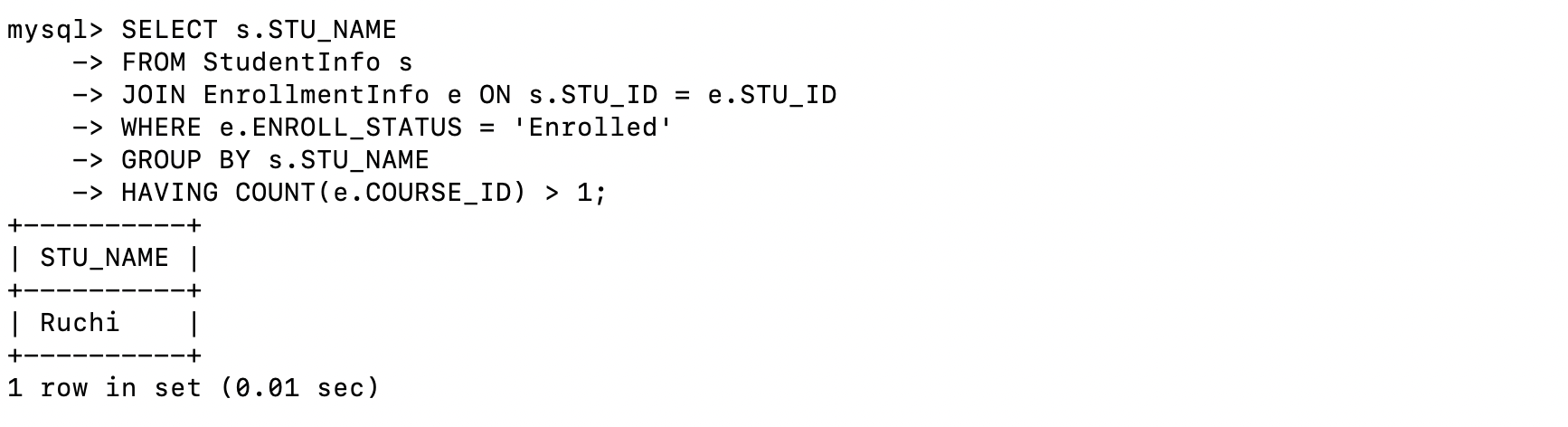
FROM StudentInfo s

JOIN EnrollmentInfo e ON s.STU\_ID = e.STU\_ID

WHERE e.ENROLL\_STATUS = 'Enrolled'

GROUP BY s.STU\_NAME

HAVING COUNT(e.COURSE\_ID) > 1;



e) Write a query to retrieve the courses that have the highest number of enrolled

students(arranging from highest to lowest)

SELECT c.COURSE\_NAME, COUNT(e.STU\_ID) AS num\_students

FROM CoursesInfo c

LEFT JOIN EnrollmentInfo e ON c.COURSE\_ID = e.COURSE\_ID AND e.ENROLL\_STATUS = 'Enrolled'

GROUP BY c.COURSE\_NAME

ORDER BY num\_students DESC;

