**1. Database Creation**

-- Create the database

CREATE DATABASE academic\_management;

USE academic\_management;

-- Create StudentInfo table

CREATE TABLE StudentInfo (

STU\_ID VARCHAR(10) PRIMARY KEY,

STU\_NAME VARCHAR(50) NOT NULL,

DOB DATE NOT NULL,

PHONE\_NO VARCHAR(15),

EMAIL\_ID VARCHAR(50) UNIQUE,

ADDRESS TEXT

);

-- Create CoursesInfo table

CREATE TABLE CoursesInfo (

COURSE\_ID VARCHAR(10) PRIMARY KEY,

COURSE\_NAME VARCHAR(100) NOT NULL,

COURSE\_INSTRUCTOR\_NAME VARCHAR(50) NOT NULL

);

-- Create EnrollmentInfo table

CREATE TABLE EnrollmentInfo (

ENROLLMENT\_ID VARCHAR(10) PRIMARY KEY,

STU\_ID VARCHAR(10),

COURSE\_ID VARCHAR(10),

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

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

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

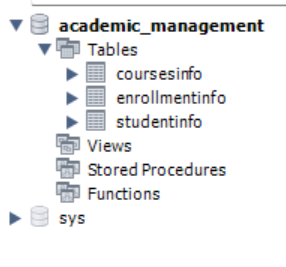
);

-- Add indexes for better performance

CREATE INDEX idx\_student\_name ON StudentInfo(STU\_NAME);

CREATE INDEX idx\_course\_name ON CoursesInfo(COURSE\_NAME);

CREATE INDEX idx\_enrollment\_status ON EnrollmentInfo(ENROLL\_STATUS);



**2. Data Creation:**

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

('S1001', 'Karthikeyan Subramanian', '2000-05-15', '+91-9876543201', 'karthik.s@email.com', '45, Gandhi Street, Adyar, Chennai - 600020, Tamil Nadu'),

('S1002', 'Lakshmi Priya Ramanathan', '2001-07-22', '+91-9876543202', 'lakshmi.r@email.com', '12/5, Kamarajar Salai, T.Nagar, Chennai - 600017, Tamil Nadu'),

('S1003', 'Muthu Krishnan', '2000-03-30', '+91-9876543203', 'muthuk@email.com', '78, East Coast Road, Besant Nagar, Chennai - 600090, Tamil Nadu'),

('S1004', 'Deepa Venkatesh', '2001-11-12', '+91-9876543204', 'deepa.v@email.com', '23, Raja Street, Mylapore, Chennai - 600004, Tamil Nadu'),

('S1005', 'Arun Prasad', '2000-09-25', '+91-9876543205', 'arun.p@email.com', '156, Anna Nagar West, Chennai - 600040, Tamil Nadu'),

('S1006', 'Meenakshi Sundaram', '2001-04-18', '+91-9876543206', 'meena.s@email.com', '89, Pondy Bazaar, T.Nagar, Chennai - 600017, Tamil Nadu'),

('S1007', 'Senthil Kumar Rajan', '2000-08-09', '+91-9876543207', 'senthil.r@email.com', '34, West Mambalam, Chennai - 600033, Tamil Nadu'),

('S1008', 'Anandhi Thirumalai', '2001-01-30', '+91-9876543208', 'anandhi.t@email.com', '67, Ashok Nagar, Chennai - 600083, Tamil Nadu');

-- Insert data into CoursesInfo table

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

('CS101', 'Introduction to Computer Science', 'Dr. Rajagopalan Venkatesan'),

('MT201', 'Advanced Mathematics', 'Prof. Saraswathi Krishnamoorthy'),

('PH301', 'Physics Fundamentals', 'Dr. Shanmugam Palaniappan'),

('EN401', 'English Literature', 'Prof. Padmavathi Ranganathan'),

('DB501', 'Database Management', 'Dr. Balakrishnan Narayanan'),

('TM101', 'Tamil Literature', 'Prof. Tamilselvan Murugesh'),

('CH201', 'Chemistry Basics', 'Dr. Vasudevan Sundaram');

-- Insert data into EnrollmentInfo table

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

('E10001', 'S1001', 'CS101', 'Enrolled'),

('E10002', 'S1001', 'MT201', 'Enrolled'),

('E10003', 'S1002', 'PH301', 'Enrolled'),

('E10004004', 'S1003', 'CS101', 'Enrolled'),

('E10005', 'S1004', 'EN401', 'Not Enrolled'),

('E10006', 'S1005', 'DB501', 'Enrolled'),

('E10007', 'S1002', 'TM101', 'Enrolled'),

('E10008', 'S1006', 'MT201', 'Enrolled'),

('E10009', 'S1007', 'DB501', 'Enrolled'),

('E10010', 'S1008', 'CH201', 'Enrolled'),

('E10011', 'S1003', 'TM101', 'Not Enrolled'),

('E10012', 'S1004', 'CH201', 'Enrolled');

**3) Retrieve the Student Information**

SELECT

s.STU\_ID as StudentID,

s.STU\_NAME as StudentName,

s.PHONE\_NO as PhoneNo,

s.EMAIL\_ID as EmailID,

GROUP\_CONCAT(

CONCAT(c.COURSE\_NAME, ' (', e.ENROLL\_STATUS, ')')

ORDER BY c.COURSE\_NAME

SEPARATOR '; '

) as Course\_Details

FROM

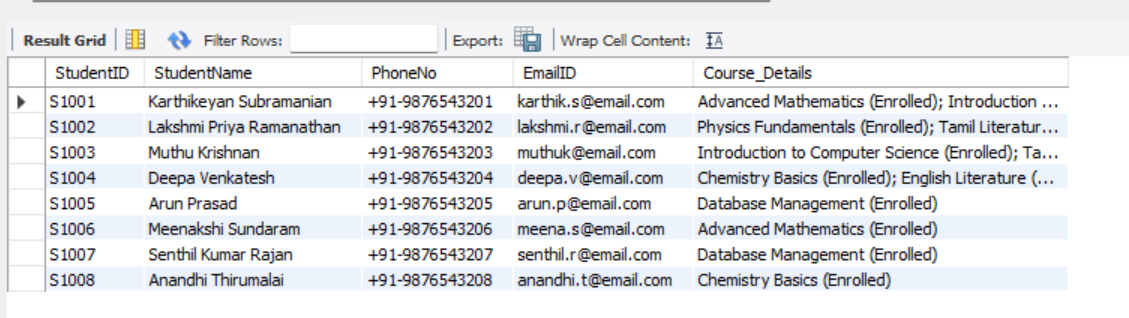
StudentInfo s

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

LEFT JOIN CoursesInfo c ON e.COURSE\_ID = c.COURSE\_ID

GROUP BY

s.STU\_ID, s.STU\_NAME, s.PHONE\_NO, s.EMAIL\_ID



**Write a query to retrieve a list of courses in which a specific student is enrolled**

SELECT

s.STU\_NAME,

s.PHONE\_NO,

s.EMAIL\_ID,

s.ADDRESS,

c.COURSE\_NAME,

e.ENROLL\_STATUS

FROM

StudentInfo s

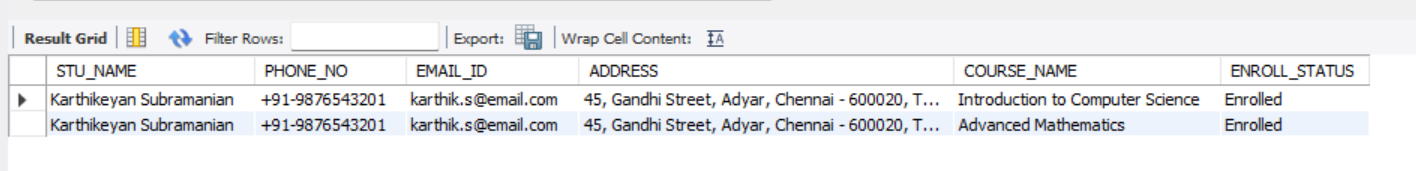
LEFT JOIN EnrollmentInfo e ON s.STU\_ID = e.STU\_ID

LEFT JOIN CoursesInfo c ON e.COURSE\_ID = c.COURSE\_ID

WHERE

s.STU\_NAME LIKE '%Karthik%'

;



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

SELECT

COURSE\_ID,

COURSE\_NAME,

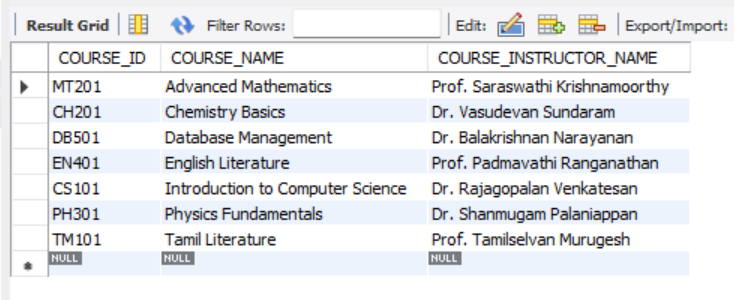
COURSE\_INSTRUCTOR\_NAME

FROM

CoursesInfo

ORDER BY

COURSE\_NAME;



**Write a query to retrieve course information for a specific course.**

SELECT

COURSE\_ID,

COURSE\_NAME,

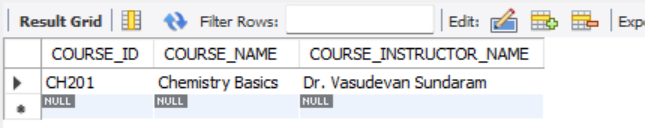
COURSE\_INSTRUCTOR\_NAME

FROM

CoursesInfo

WHERE

COURSE\_NAME = 'Chemistry Basics';



**Write a query to retrieve course information for multiple courses.**

SELECT

COURSE\_ID,

COURSE\_NAME,

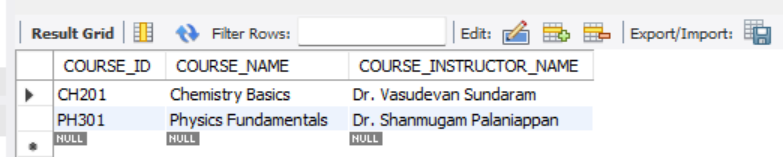
COURSE\_INSTRUCTOR\_NAME

FROM

CoursesInfo

WHERE

COURSE\_NAME in ('Chemistry Basics', 'Physics Fundamentals');



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

SELECT

COURSE\_NAME,

COUNT(ei.STU\_ID) as count\_students

FROM

CoursesInfo as ci

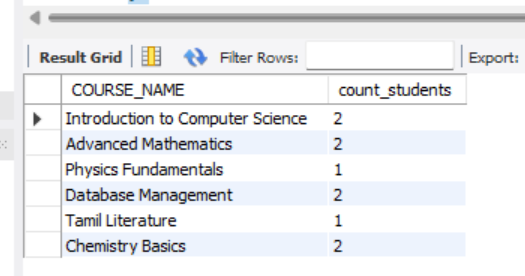
INNER JOIN EnrollmentInfo as ei ON ci.COURSE\_ID = ei.COURSE\_ID

WHERE

ei.ENROLL\_STATUS = 'Enrolled'

GROUP BY 1

;



**Write a query to retrieve the list of students enrolled in a specific course. please answer with one query.**

SELECT

c.COURSE\_NAME as 'Course',

c.COURSE\_INSTRUCTOR\_NAME as 'Instructor',

s.STU\_ID as 'Student ID',

s.STU\_NAME as 'Student Name',

s.PHONE\_NO as 'Contact Number',

s.EMAIL\_ID as 'Email',

e.ENROLL\_STATUS as 'Status',

s.ADDRESS as 'Address'

FROM

CoursesInfo c

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

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

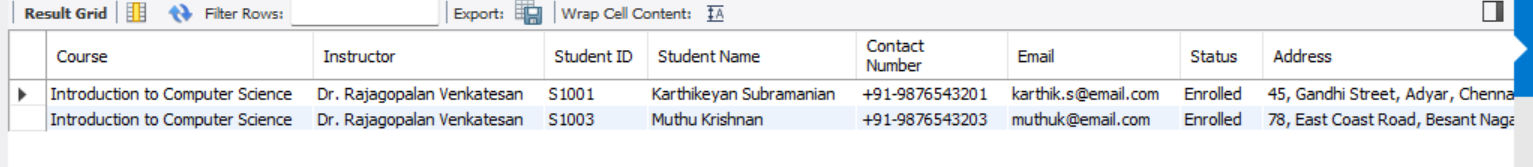
WHERE

c.COURSE\_ID = 'CS101' -- Change course ID as needed

AND e.ENROLL\_STATUS = 'Enrolled'

ORDER BY

s.STU\_NAME;



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

SELECT

c.COURSE\_INSTRUCTOR\_NAME,

COUNT(DISTINCT CASE WHEN e.ENROLL\_STATUS = 'Enrolled' THEN s.STU\_ID END) as Enrolled\_Student\_Count

FROM

CoursesInfo c

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

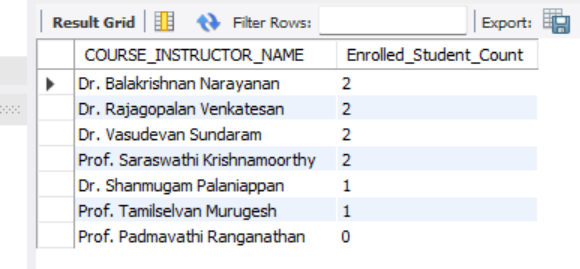
LEFT JOIN StudentInfo s ON e.STU\_ID = s.STU\_ID

GROUP BY

c.COURSE\_INSTRUCTOR\_NAME

ORDER BY

Enrolled\_Student\_Count DESC;



**Write a query to retrieve the list of students who are enrolled in multiple courses.**

SELECT

s.STU\_ID,

s.STU\_NAME,

COUNT(DISTINCT e.COURSE\_ID) as Course\_Count

FROM

StudentInfo s

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

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

WHERE

e.ENROLL\_STATUS = 'Enrolled'

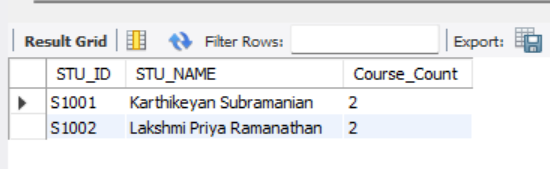
GROUP BY

1, 2

HAVING

COUNT(DISTINCT e.COURSE\_ID) > 1

;



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

**students(arranging from highest to lowest).**

SELECT

c.COURSE\_ID,

c.COURSE\_NAME,

c.COURSE\_INSTRUCTOR\_NAME,

COUNT(CASE WHEN e.ENROLL\_STATUS = 'Enrolled' THEN e.STU\_ID END) as Enrolled\_Students

FROM

CoursesInfo c

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

LEFT JOIN StudentInfo s ON e.STU\_ID = s.STU\_ID

GROUP BY

c.COURSE\_ID, c.COURSE\_NAME, c.COURSE\_INSTRUCTOR\_NAME

ORDER BY

Enrolled\_Students DESC;

