**1.Introduction**

**\*Background:**

The objective of the project is to create one database called student management in this we have created tables related to students .Use this Database we can easily get the details of students using Sql querys.This helps everyone who are need to know thw details of every single person indetail with easy way using SQL querys.

Objectives:

1.evreyone can access and retrive the data easily

2.only admin can changes the databases

3.we can select,insert,update,delete,drop,join the data

**2.SOFTWARE SPECIFICATION** :

\***Language** : MySQL

\*  **Software** : Mysql work bench software

**MySQl:**

**1.**[**Data**](https://www.linkedin.com/feed/hashtag/?keywords=data&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7230288537009487872)**-raw fact  
2.**[**Database**](https://www.linkedin.com/feed/hashtag/?keywords=database&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7230288537009487872)**-structered file we can retreive,access,modify,delete,changes  
3.**[**DBMS**](https://www.linkedin.com/feed/hashtag/?keywords=dbms&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7230288537009487872)**-software  
4.**[**SQLcommands**](https://www.linkedin.com/feed/hashtag/?keywords=sqlcommands&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7230288537009487872)**:**[**DDL**](https://www.linkedin.com/feed/hashtag/?keywords=ddl&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7230288537009487872) **(Data Definition Language): Used to define and manage database structures.  
CREATE, ALTER, DROP, TRUNCATE, RENAME**[**DML**](https://www.linkedin.com/feed/hashtag/?keywords=dml&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7230288537009487872) **(Data Manipulation Language): Used to manipulate data in tables.  
INSERT, UPDATE, DELETE**[**DQL**](https://www.linkedin.com/feed/hashtag/?keywords=dql&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7230288537009487872) **(Data Query Language): Used to query data from the database.  
SELECT**[**DCL**](https://www.linkedin.com/feed/hashtag/?keywords=dcl&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7230288537009487872) **(Data Control Language): Used to control access to data.  
GRANT, REVOKE**[**TCL**](https://www.linkedin.com/feed/hashtag/?keywords=tcl&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7230288537009487872) **(Transaction Control Language): Used to manage transactions.  
COMMIT, ROLLBACK, SAVEPOINT, RELEASE SAVEPOINT  
  
5.Constraints -not null, primary key,foreign key,unique,check,default  
6.where clause**[**7.group**](http://7.group/) **by,order by ,having  
8.aggregate functions  
9.string functions and date functions and numeric functions  
10.joins ]  
11.views  
12.CTEs**

**Project:**

**create database StudentManagement;**

**use** **StudentManagement**;

**STUDENTS TABLE**

**create table Students( student\_id int primary key auto\_increment,first\_name varchar(50) not null,lastname varchar(50) not null,date\_of\_birth date not null,gender enum('male','female','others') not null,email varchar(50) unique not null,phone\_number varchar(50) unique ,address text);**

**select\*from** **Students**;

**insert into Students values**(null,'vasu','gadam','2004-06-10','Male','gadamsrinivasulu789@gmail.com',9392726310,'sanikvaram 523331'),

(null,'mani','kamisetty','2004-06-02','Male','manikanta9@gmail.com',9346093985,'sanikavaram 523331'),

(NULL, 'John', 'Doe', '2000-01-01', 'Male', 'john.doe1@example.com', '1234567890', '123 Main St'),

(NULL, 'Jane', 'Smith', '2001-02-02', 'Female', 'jane.smith@example.com', '2345678901', '456 Oak St'),

(NULL, 'Alice', 'Johnson', '1999-03-03', 'Female', 'alice.johnson@example.com', '3456789012', '789 Pine St'),

(NULL, 'Bob', 'Brown', '2002-04-04', 'Male', 'bob.brown@example.com', '4567890123', '101 Maple St'),

(NULL, 'Charlie', 'Davis', '2000-05-05', 'Male', 'charlie.davis@example.com', '5678901234', '202 Elm St'),

(NULL, 'Emily', 'Wilson', '2001-06-06', 'Female', 'emily.wilson@example.com', '6789012345', '303 Cedar St'),

(NULL, 'David', 'Miller', '1998-07-07', 'Male', 'david.miller@example.com', '7890123456', '404 Birch St'),

(NULL, 'Sophia', 'Moore', '1997-08-08', 'Female', 'sophia.moore@example.com', '8901234567', '505 Walnut St'),

(NULL, 'James', 'Taylor', '1999-09-09', 'Male', 'james.taylor@example.com', '9012345678', '606 Ash St'),

(NULL, 'Olivia', 'Anderson', '2000-10-10', 'Female', 'olivia.anderson@example.com', '0123456789', '707 Redwood St'),

(NULL, 'Mason', 'Thomas', '2001-11-11', 'Male', 'mason.thomas@example.com', '1123456789', '808 Cypress St'),

(NULL, 'Ella', 'Jackson', '1998-12-12', 'Female', 'ella.jackson@example.com', '2123456789', '909 Fir St'),

(NULL, 'Aiden', 'White', '1997-01-01', 'Male', 'aiden.white@example.com', '3123456789', '1010 Willow St'),

(NULL, 'Mia', 'Harris', '1999-02-02', 'Female', 'mia.harris@example.com', '4123456789', '1111 Spruce St'),

(NULL, 'Lucas', 'Martin', '2002-03-03', 'Male', 'lucas.martin@example.com', '5123456789', '1212 Chestnut St'),

(NULL, 'Charlotte', 'Thompson', '2001-04-04', 'Female', 'charlotte.thompson@example.com', '6123456789', '1313 Poplar St'),

(NULL, 'Noah', 'Garcia', '2000-05-05', 'Male', 'noah.garcia@example.com', '7123456789', '1414 Sycamore St'),

(NULL, 'Amelia', 'Martinez', '1998-06-06', 'Female', 'amelia.martinez@example.com', '8123456789', '1515 Dogwood St'),

(NULL, 'Liam', 'Rodriguez', '1997-07-07', 'Male', 'liam.rodriguez@example.com', '9123456789', '1616 Alder St'),

(NULL, 'Harper', 'Martinez', '1999-08-08', 'Female', 'harper.martinez@example.com', '0123456790', '1717 Hickory St'),

(NULL, 'William', 'Hernandez', '2002-09-09', 'Male', 'william.hernandez@example.com', '1123456790', '1818 Juniper St'),

(NULL, 'Avery', 'Lopez', '2001-10-10', 'Female', 'avery.lopez@example.com', '2123456790', '1919 Olive St'),

(NULL, 'Ethan', 'Gonzalez', '1998-11-11', 'Male', 'ethan.gonzalez@example.com', '3123456790', '2020 Magnolia St'),

(NULL, 'Sofia', 'Wilson', '1997-12-12', 'Female', 'sofia.wilson@example.com', '4123456790', '2121 Palm St'),

(NULL, 'Benjamin', 'Perez', '1999-01-01', 'Male', 'benjamin.perez@example.com', '5123456790', '2222 Bay St'),

(NULL, 'Lily', 'Taylor', '2002-02-02', 'Female', 'lily.taylor@example.com', '6123456790', '2323 Pinecone St'),

(NULL, 'Michael', 'Anderson', '2001-03-03', 'Male', 'michael.anderson@example.com', '7123456790', '2424 Oakleaf St'),

(NULL, 'Aubrey', 'Wilson', '1998-04-04', 'Female', 'aubrey.wilson@example.com', '8123456790', '2525 Treelawn St'),

(NULL, 'Alexander', 'Lewis', '1997-05-05', 'Male', 'alexander.lewis@example.com', '9123456790', '2626 Leafwood St'),

(NULL, 'Zoey', 'Clark', '1999-06-06', 'Female', 'zoey.clark@example.com', '0123456791', '2727 Glenwood St'),

(NULL, 'Jacob', 'Walker', '2002-07-07', 'Male', 'jacob.walker@example.com', '1123456791', '2828 Greenwood St'),

(NULL, 'Grace', 'Hall', '2001-08-08', 'Female', 'grace.hall@example.com', '2123456791', '2929 Meadow St'),

(NULL, 'Daniel', 'Allen', '1998-09-09', 'Male', 'daniel.allen@example.com', '3123456791', '3030 Woodside St'),

(NULL, 'Layla', 'Young', '1997-10-10', 'Female', 'layla.young@example.com', '4123456791', '3131 Wildwood St'),

(NULL, 'Matthew', 'King', '1999-11-11', 'Male', 'matthew.king@example.com', '5123456791', '3232 Creekside St'),

(NULL, 'Hannah', 'Wright', '2002-12-12', 'Female', 'hannah.wright@example.com', '6123456791', '3333 Riverwood St'),

(NULL, 'Joshua', 'Scott', '2001-01-01', 'Male', 'joshua.scott@example.com', '7123456791', '3434 Ridgewood St'),

(NULL, 'Scarlett', 'Torres', '1998-02-02', 'Female', 'scarlett.torres@example.com', '8123456791', '3535 Fernwood St'),

(NULL, 'Ryan', 'Nguyen', '1997-03-03', 'Male', 'ryan.nguyen@example.com', '9123456791', '3636 Birchwood St'),

(NULL, 'Ella', 'Rodriguez', '1999-04-04', 'Female', 'ella.rodriguez@example.com', '0123456792', '3737 Palmwood St'),

(NULL, 'Henry', 'Gonzalez', '2002-05-05', 'Male', 'henry.gonzalez@example.com', '1123456792', '3838 Elmwood St'),

(NULL, 'Aria', 'Lee', '2001-06-06', 'Female', 'aria.lee@example.com', '2123456792', '3939 Cottonwood St'),

(NULL, 'Samuel', 'Clark', '1998-07-07', 'Male', 'samuel.clark@example.com', '3123456792', '4040 Pinewood St'),

(NULL, 'Victoria', 'Allen', '1997-08-08', 'Female', 'victoria.allen@example.com', '4123456792', '4141 Cedarwood St'),

(NULL, 'Andrew', 'Young', '1999-09-09', 'Male', 'andrew.young@example.com', '5123456792', '4242 Ashwood St'),

(NULL, 'Abigail', 'Hall', '2002-10-10', 'Female', 'abigail.hall@example.com', '6123456792', '4343 Redwood St'),

(NULL, 'Joseph', 'Wright', '2001-11-11', 'Male', 'joseph.wright@example.com', '7123456792', '4444 Cypresswood St'),

(NULL, 'Luna', 'Harris', '1998-12-12', 'Female', 'luna.harris@example.com', '8123456792', '4545 Birchwood St'),

(NULL, 'Jackson', 'Martinez', '1997-01-01', 'Male', 'jackson.martinez@example.com', '9123456792', '4646 Maplewood St');

**DELETE FROM Students WHERE student\_id = 1;**

**drop table Students;**

**select\*from Students;**

**COURSES TABLE**

**USE StudentManagement;**

**create table Courses(course\_id int primary key auto\_increment ,course\_name varchar(50) not null,course\_description text,credit\_hours int not null);**

**INSERT INTO Courses (course\_id, course\_name, course\_description, credit\_hours)**

**VALUES**

(NULL, ' Computer Science', 'Basic principles of computer science, including programming and problem-solving.', 3),

(NULL, 'Calculus I', 'An introduction to differential and integral calculus of one variable.', 4),

(NULL, 'Physics I', 'Fundamentals of mechanics, waves, and thermodynamics.', 4),

(NULL, 'English Literature', 'Study of major works in English literature from various periods.', 3),

(NULL, ' Computer Science', 'Basic principles of computer science, including programming and problem-solving.', 3),

(NULL, 'Calculus I', 'An introduction to differential and integral calculus of one variable.', 4),

(NULL, 'Physics I', 'Fundamentals of mechanics, waves, and thermodynamics.', 4),

(NULL, 'English Literature', 'Study of major works in English literature from various periods.', 3),

(NULL, 'Introduction to Psychology', 'Overview of psychological theories, principles, and practices.', 3),

(NULL, 'Data Structures', 'In-depth study of data structures and algorithms.', 4),

(NULL, 'Database Systems', 'Principles of database design, implementation, and management.', 3),

(NULL, 'Operating Systems', 'Study of operating system concepts and design.', 4),

(NULL, 'Discrete Mathematics', 'Introduction to discrete mathematical structures and their applications.', 3),

(NULL, 'Microeconomics', 'Basic concepts and principles of microeconomic theory.', 3),

(NULL, 'Macroeconomics', 'Introduction to macroeconomic theory, including inflation, unemployment, and monetary policy.', 3),

(NULL, 'History of Western Civilization', 'A survey of major events, figures, and movements in Western history.', 3),

(NULL, 'Introduction to Sociology', 'Exploration of human society, social behaviors, and institutions.', 3),

(NULL, 'General Chemistry I', 'Basic principles of chemistry including atomic structure, bonding, and reactions.', 4),

(NULL, 'Biology I', 'Introduction to the principles of biology, including cell structure, genetics, and evolution.', 4),

(NULL, 'Linear Algebra', 'Study of vector spaces, linear transformations, and matrices.', 3),

(NULL, 'Calculus I', 'An introduction to differential and integral calculus of one variable.', 4),

(NULL, 'Physics I', 'Fundamentals of mechanics, waves, and thermodynamics.', 4),

(NULL, 'English Literature', 'Study of major works in English literature from various periods.', 3),

(NULL, 'Introduction to Psychology', 'Overview of psychological theories, principles, and practices.', 3),

(NULL, 'Discrete Mathematics', 'Introduction to discrete mathematical structures and their applications.', 3),

(NULL, 'Microeconomics', 'Basic concepts and principles of microeconomic theory.', 3),

(NULL, 'Macroeconomics', 'Introduction to macroeconomic theory, including inflation, unemployment, and monetary policy.', 3),

(NULL, 'History of Western Civilization', 'A survey of major events, figures, and movements in Western history.', 3),

(NULL, 'Introduction to Sociology', 'Exploration of human society, social behaviors, and institutions.', 3);

**select\*from Courses;**

**drop table courses;**

**ENROLLMENT TABLE**

**use StudentManagement;**

**create table enrollment(enrollment\_id int primary key auto\_increment,student\_id int not null,course\_id int not null,FOREIGN KEY (student\_id) REFERENCES Students(student\_id)**

**,FOREIGN KEY (course\_id) REFERENCES Courses(course\_id));**

**select \*from enrollment;**

**INSERT INTO Enrollment (student\_id, course\_id)**

**VALUES**

(1, 1), -- John Doe is enrolled in Introduction to SQL

(2, 2), -- Jane Smith is enrolled in Advanced SQL

(3, 3), -- Emily Jones is enrolled in Database Design

(4, 4), -- Michael Brown is enrolled in Data Warehousing

(5, 5), -- Sarah Davis is enrolled in NoSQL Databases

(6, 6), -- James Wilson is enrolled in Web Development

(7, 7), -- Olivia Taylor is enrolled in Machine Learning

(8, 8), -- Daniel Anderson is enrolled in Cloud Computing

(9, 1), -- Sophia Thomas is also enrolled in Introduction to SQL

(10, 2), -- Benjamin Jackson is also enrolled in Advanced SQL

(1, 3), -- John Doe is also enrolled in Database Design

(2, 4), -- Jane Smith is also enrolled in Data Warehousing

(3, 5), -- Emily Jones is also enrolled in NoSQL Databases

(4, 6), -- Michael Brown is also enrolled in Web Development

(5, 7), -- Sarah Davis is also enrolled in Machine Learning

(6, 8), -- James Wilson is also enrolled in Cloud Computing

(7, 1), -- Olivia Taylor is also enrolled in Introduction to SQL

(8, 2), -- Daniel Anderson is also enrolled in Advanced SQL

(9, 3), -- Sophia Thomas is also enrolled in Database Design

(10, 4); -- Benjamin Jackson is also enrolled in Data Warehousing

**INSTRUCTEORS TABLE**

**use StudentManagement;**

**create table Instructors**

**(instructer\_id int primary key auto\_increment,**

**first\_name varchar(40) not null,**

**last\_name varchar(50) not null,**

**email varchar(50) unique not null,**

**phone\_number varchar(40) unique not null);**

**INSERT INTO Instructors (first\_name, last\_name, email, phone\_number)**

**VALUES(**'vasu','gadam','gadamsrinivasulu789@gmail.com',9392726310),

('mani','kamisetty','kamisettimanikanta@gmail.com',9346093985),

('ravi','vadde','ravitejavadde@gmail.com',9389389292),

('sai','nuthalapati','saibabau@gmail.com',9292937373),

('vamsi','rayapati','rayapativamsi@gmail.com',9080888989),

('John', 'Doe', 'john.doe@example.com', 1234567890),

('Jane', 'Smith', 'jane.smith@example.com', 2345678901),

('Emily', 'Jones', 'emily.jones@example.com', 3456789012),

('Michael', 'Brown', 'michael.brown@example.com', 4567890123),

('Sarah', 'Davis', 'sarah.davis@example.com', 5678901234),

('James', 'Wilson', 'james.wilson@example.com', 6789012345),

('Olivia', 'Taylor', 'olivia.taylor@example.com', 7890123456),

('Daniel', 'Anderson', 'daniel.anderson@example.com', 8901234567),

('Sophia', 'Thomas', 'sophia.thomas@example.com', 9012345678),

('Benjamin', 'Jackson', 'benjamin.jackson@example.com', 1023456789),

('Alice', 'Martinez', 'alice.martinez@example.com', 2134567890),

('Chris', 'Garcia', 'chris.garcia@example.com', 3245678901),

('Emma', 'Martinez', 'emma.martinez@example.com', 4356789012),

('David', 'Rodriguez', 'david.rodriguez@example.com', 5467890123),

('Ava', 'Lopez', 'ava.lopez@example.com', 6578901234),

('Liam', 'Gonzalez', 'liam.gonzalez@example.com', 7689012345),

('Mia', 'Wilson', 'mia.wilson@example.com', 8790123456),

('Noah', 'Anderson', 'noah.anderson@example.com', 9801234567),

('Isabella', 'Taylor', 'isabella.taylor@example.com', 8912345678),

('Ethan', 'Jackson', 'ethan.jackson@example.com', 9023456789);

**select\*from instructors;**

**drop table instructors;**

**COURSEASSIGNMENT TABLE**

**use Studentmanagement;**

**create table CourseAssignments(assignment\_id int primary key auto\_increment,**

**course\_id int not null, instructer\_id int not null,**

**FOREIGN KEY (course\_id) REFERENCES Courses(course\_id),**

**FOREIGN KEY (instructer\_id) REFERENCES Instructors(instructer\_id));**

**INSERT INTO CourseAssignments (course\_id, instructer\_id)**

**VALUES**

(1, 1), -- Course 1 assigned to Instructor 1

(2, 2), -- Course 2 assigned to Instructor 2

(3, 3), -- Course 3 assigned to Instructor 3

(4, 4), -- Course 4 assigned to Instructor 4

(5, 5), -- Course 5 assigned to Instructor 5

(1, 2), -- Course 1 assigned to Instructor 2

(2, 3), -- Course 2 assigned to Instructor 3

(3, 4), -- Course 3 assigned to Instructor 4

(4, 5), -- Course 4 assigned to Instructor 5

(5, 1), -- Course 5 assigned to Instructor 1

(1, 3), -- Course 1 assigned to Instructor 3

(2, 4), -- Course 2 assigned to Instructor 4

(3, 5), -- Course 3 assigned to Instructor 5

(4, 1), -- Course 4 assigned to Instructor 1

(5, 2), -- Course 5 assigned to Instructor 2

(1, 4), -- Course 1 assigned to Instructor 4

(2, 5), -- Course 2 assigned to Instructor 5

(3, 1), -- Course 3 assigned to Instructor 1

(4, 2), -- Course 4 assigned to Instructor 2

(5, 3); -- Course 5 assigned to Instructor 3

**select\*from courseassignments**

**SQL QUERY :**

1. Retrieve the full names and email addresses of all students enrolled in the Computer Science course.
2. List all courses along with the number of students enrolled in each course.
3. Find the instructors who are teaching more than 2 courses.
4. Retrieve the details of students (name, email) who are enrolled in a course taught by Instructor Name.
5. List all students who have not enrolled in any course.
6. Get the list of courses with no assigned instructor.
7. Retrieve the full details of the Students table, sorted by last name in alphabetical order.
8. Find the total number of credit hours for a student enrolled in the Database Systems and Web Development courses.
9. Identify the student with the maximum number of enrollments.
10. Get the details of students whose phone numbers are missing.

**Anwers:**

**use studentmanagement;**

**select\*from students;**

**select\*from courses;**

**select\*from enrollment;**

**select\*from instructors;**

**select\*from courseassignments;**

**1.** Retrieve the full names and email addresses of all students enrolled in the Computer Science course.

**create view query1 as**

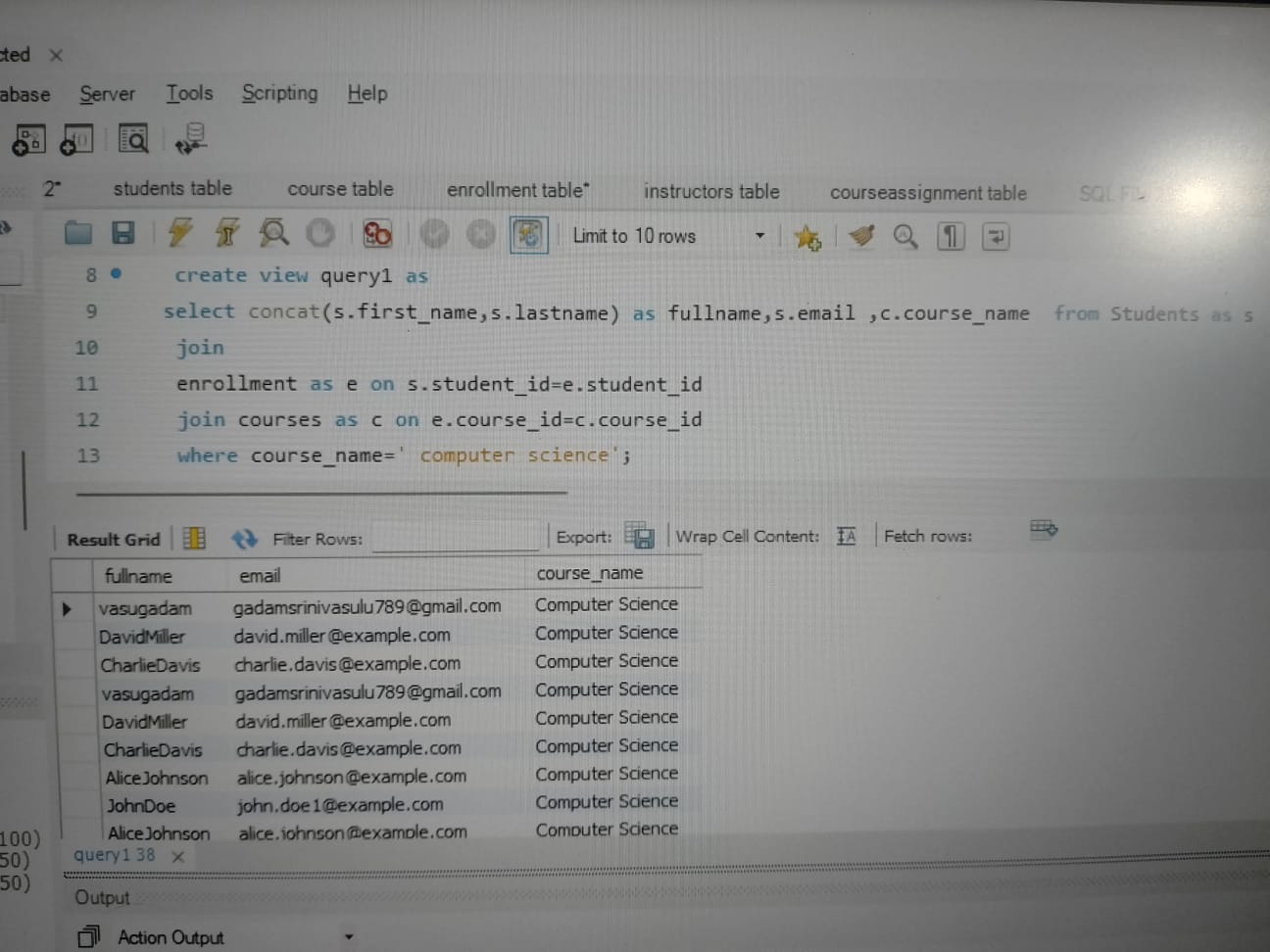
**select concat(s.first\_name,s.lastname) as fullname,s.email ,c.course\_name from Students as s**

**join**

**enrollment as e on s.student\_id=e.student\_id**

**join courses as c on e.course\_id=c.course\_id**

**where course\_name=' computer science';**

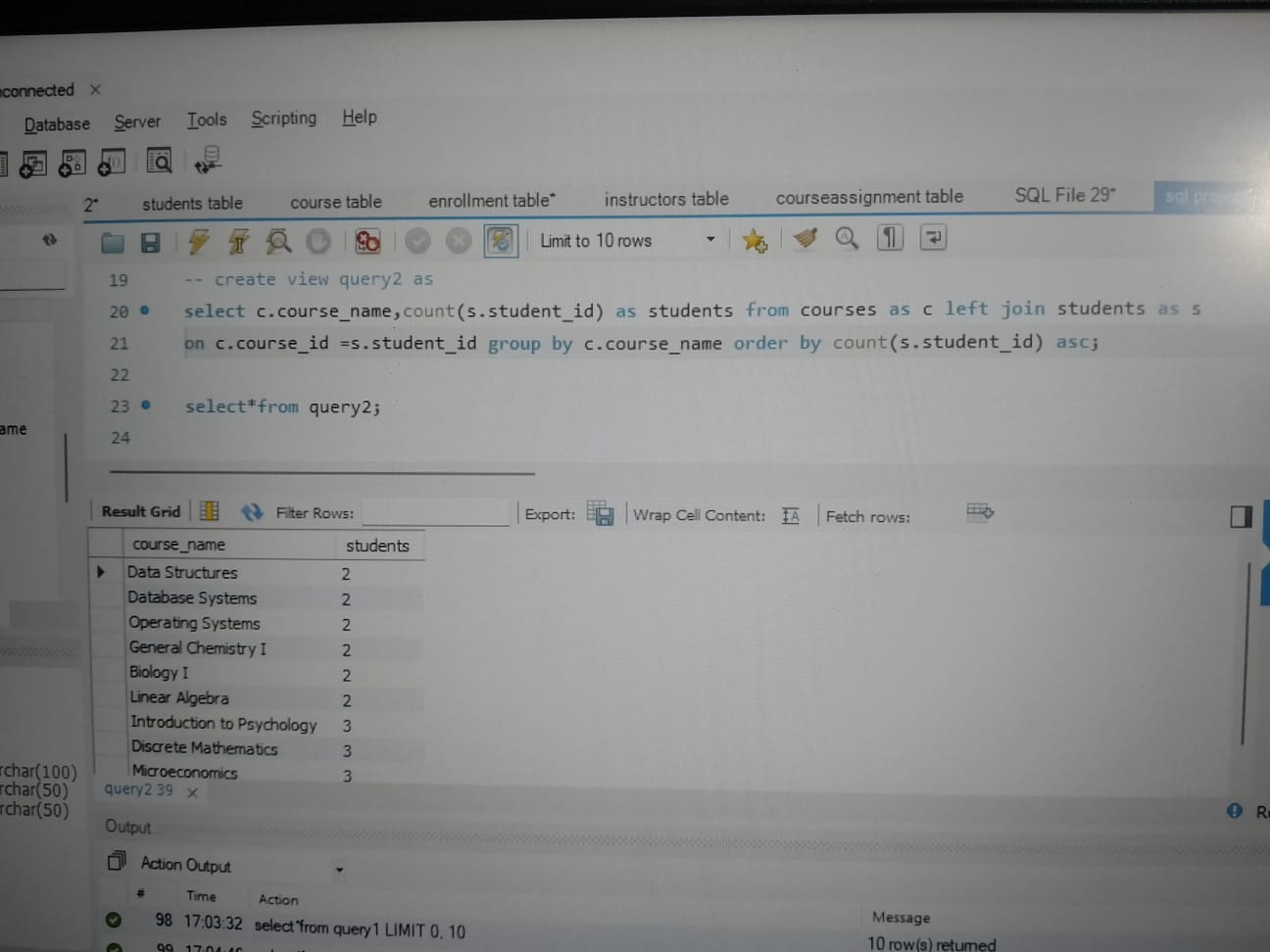
****

**select\*from query1 ;**

2.List all courses along with the number of students enrolled in each course.

**create view query2 as**

**select c.course\_name,count(e.student\_id) as students from courses as c left join enrollment as e on c.course\_id =e.course\_id group by course\_name order by count(e.student\_id) asc;**

****

**select\*from query2;**

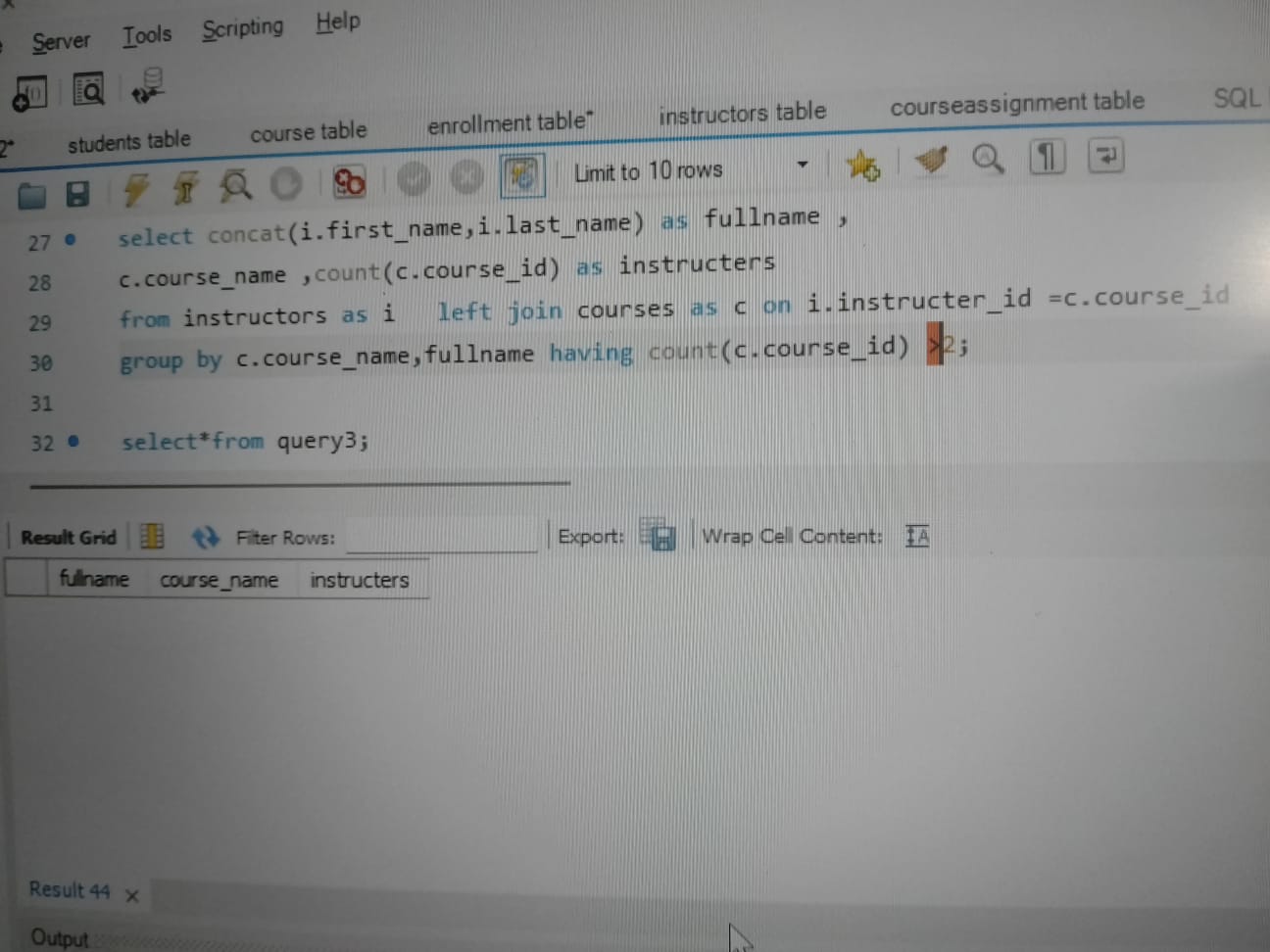
3.Find the instructors who are teaching more than 2 courses.

**create view query3 as**

**select concat(i.first\_name,i.last\_name) as fullname ,**

**c.course\_name ,count(c.course\_id) as instructers**

**from instructors as i left join courses as c on i.instructer\_id =c.course\_id group by c.course\_name,fullname having count(c.course\_id) >2;**

****

**select\*from query3;**

4.Retrieve the details of students (name, email) who are enrolled in a course taught by Instructor Name.

**create view query4 as**

**select concat(s.first\_name,s.lastname) as fullaname,**

**s.email,c.course\_name from students as s**

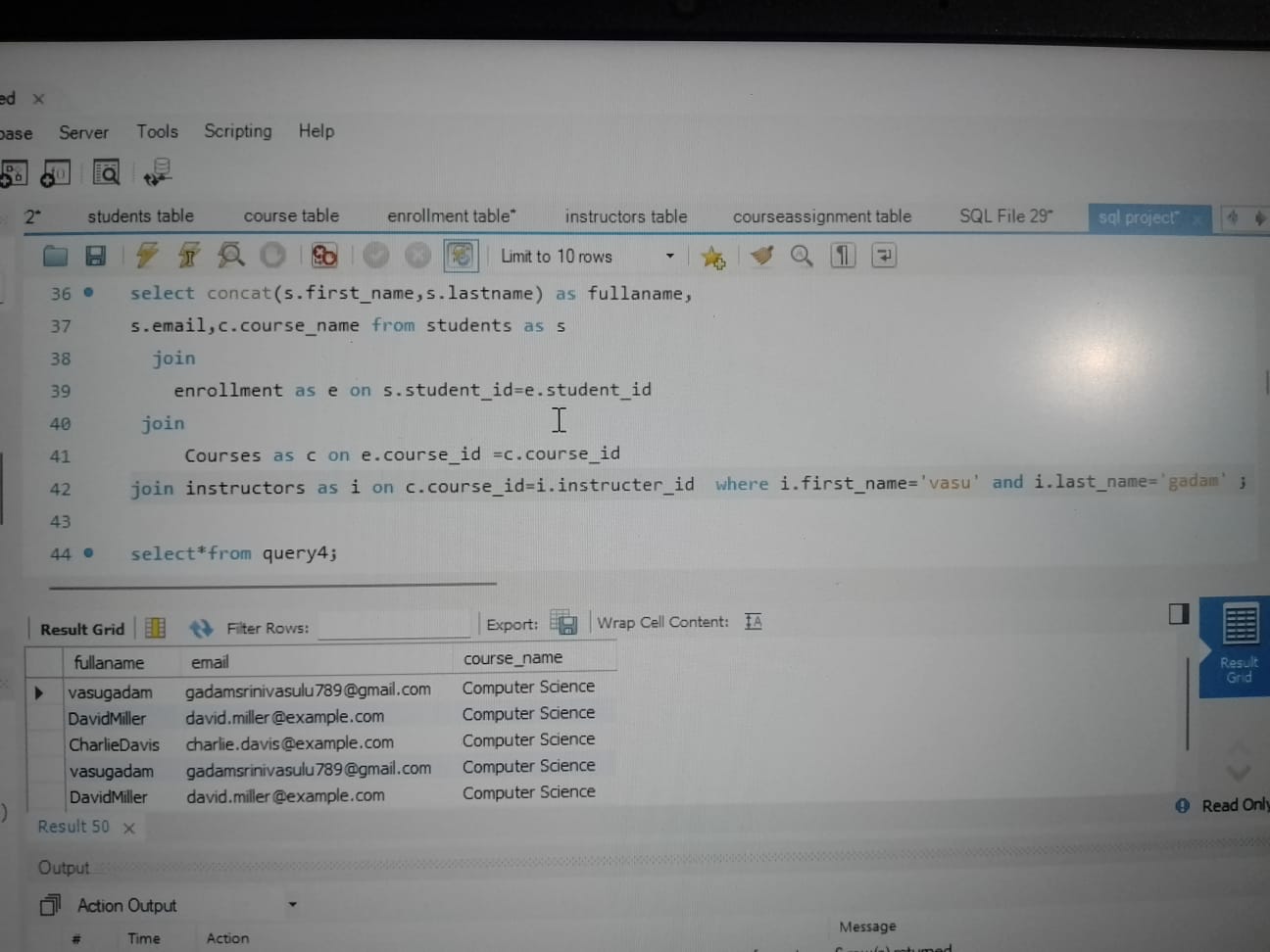
**join**

**enrollment as e on s.student\_id=e.student\_id**

**join**

**Courses as c on e.course\_id =c.course\_id**

**join instructors as i on c.course\_id=i.instructer\_id where i.first\_name='vasu' and i.last\_name='gadam' ;**

****

**select\*from query4;**

5.List all students who have not enrolled in any course.

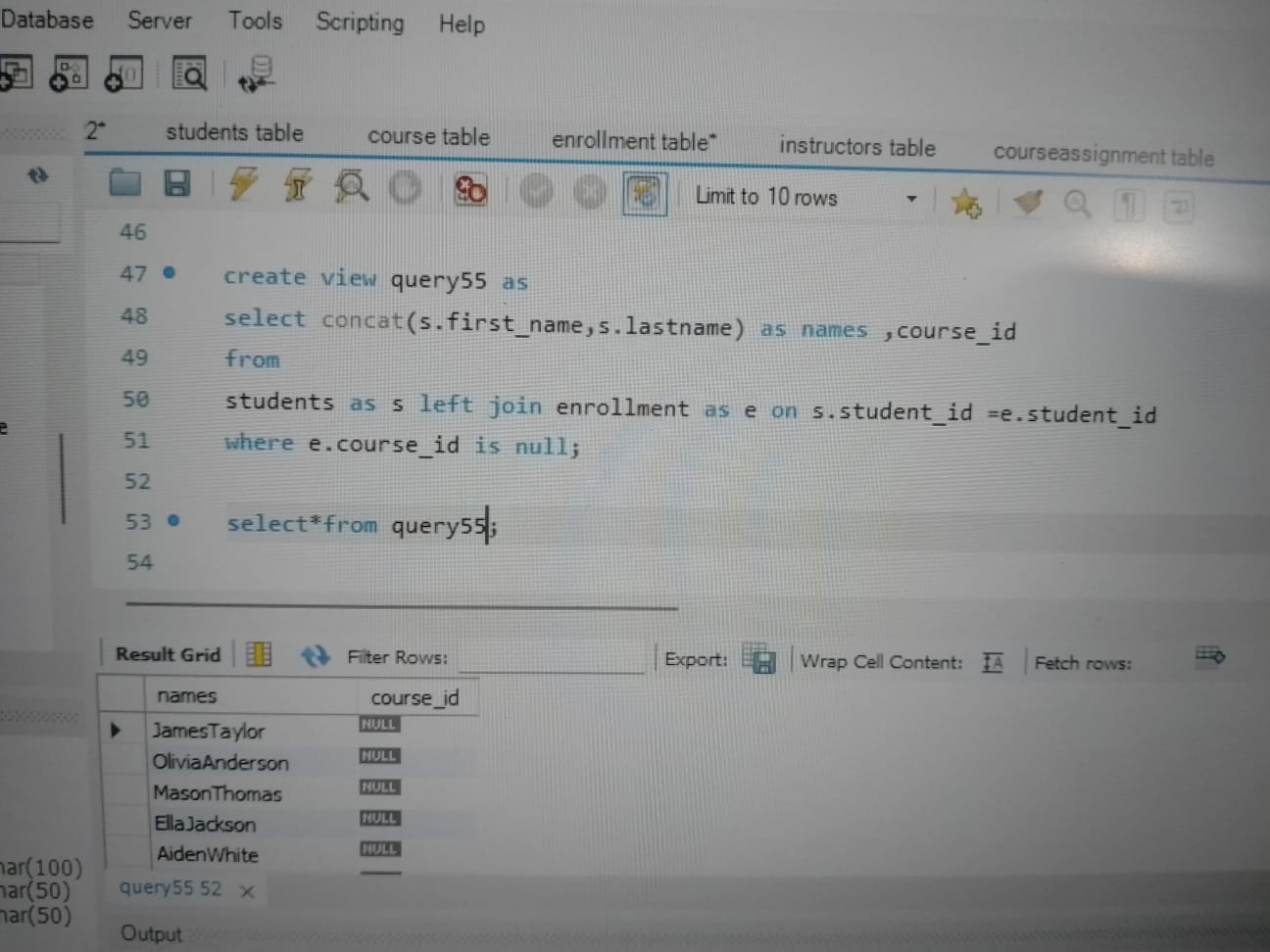
**create view query5 as**

**select concat(s.first\_name,s.lastname) as names ,course\_id**

**from**

**students as s left join enrollment as e on s.student\_id =e.student\_id**

**where e.course\_id is null;**

****

**select\*from query5;**

6.Get the list of courses with no assigned instructor.

**create view query6 as**

**select c.course\_name,Ca.instructer\_id**

**from courses as c**

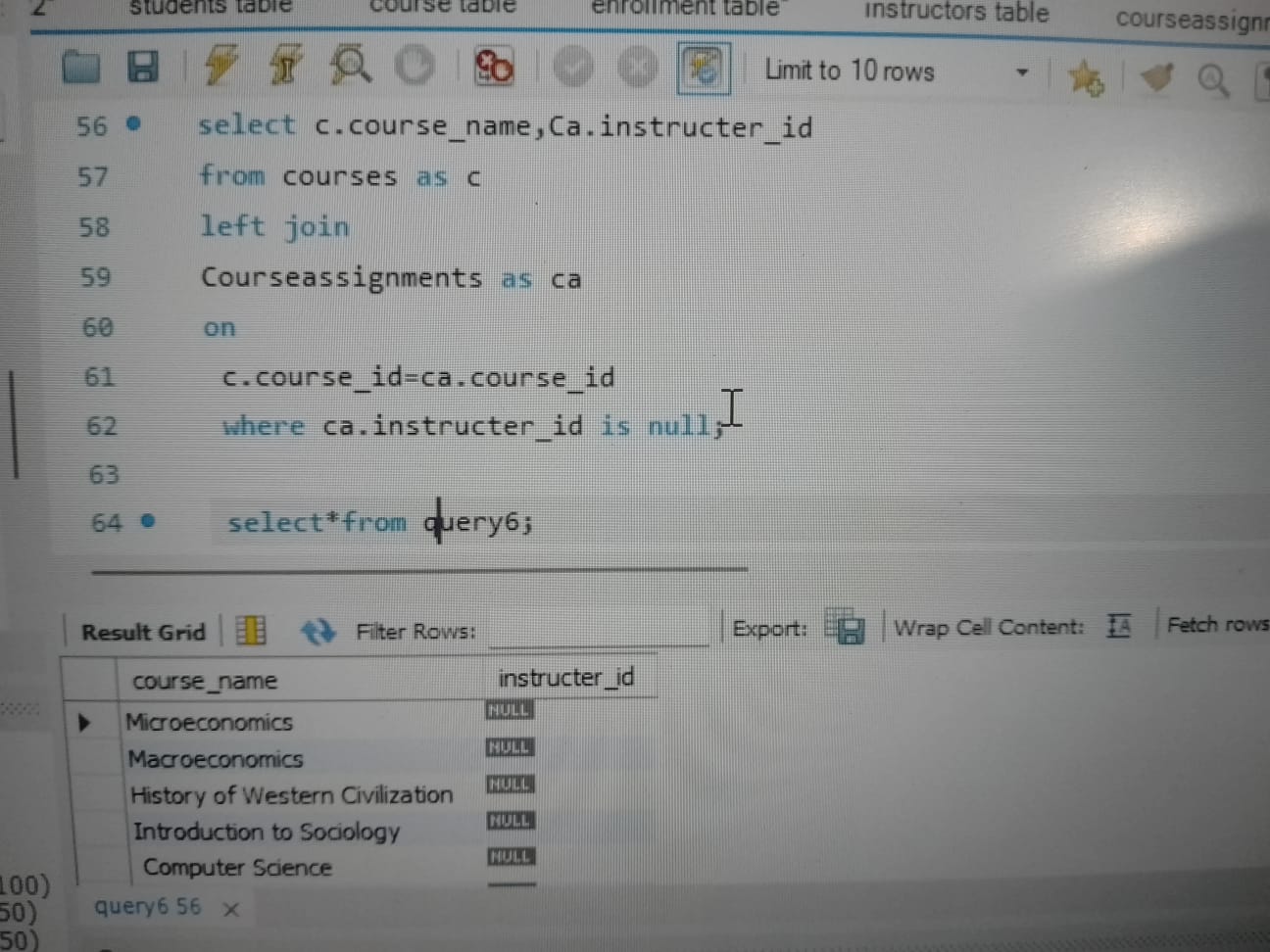
**left join**

**CourseAssignments as C**

**on**

**c.course\_id=Ca.course\_id**

**where Ca.instructer\_id is null;**

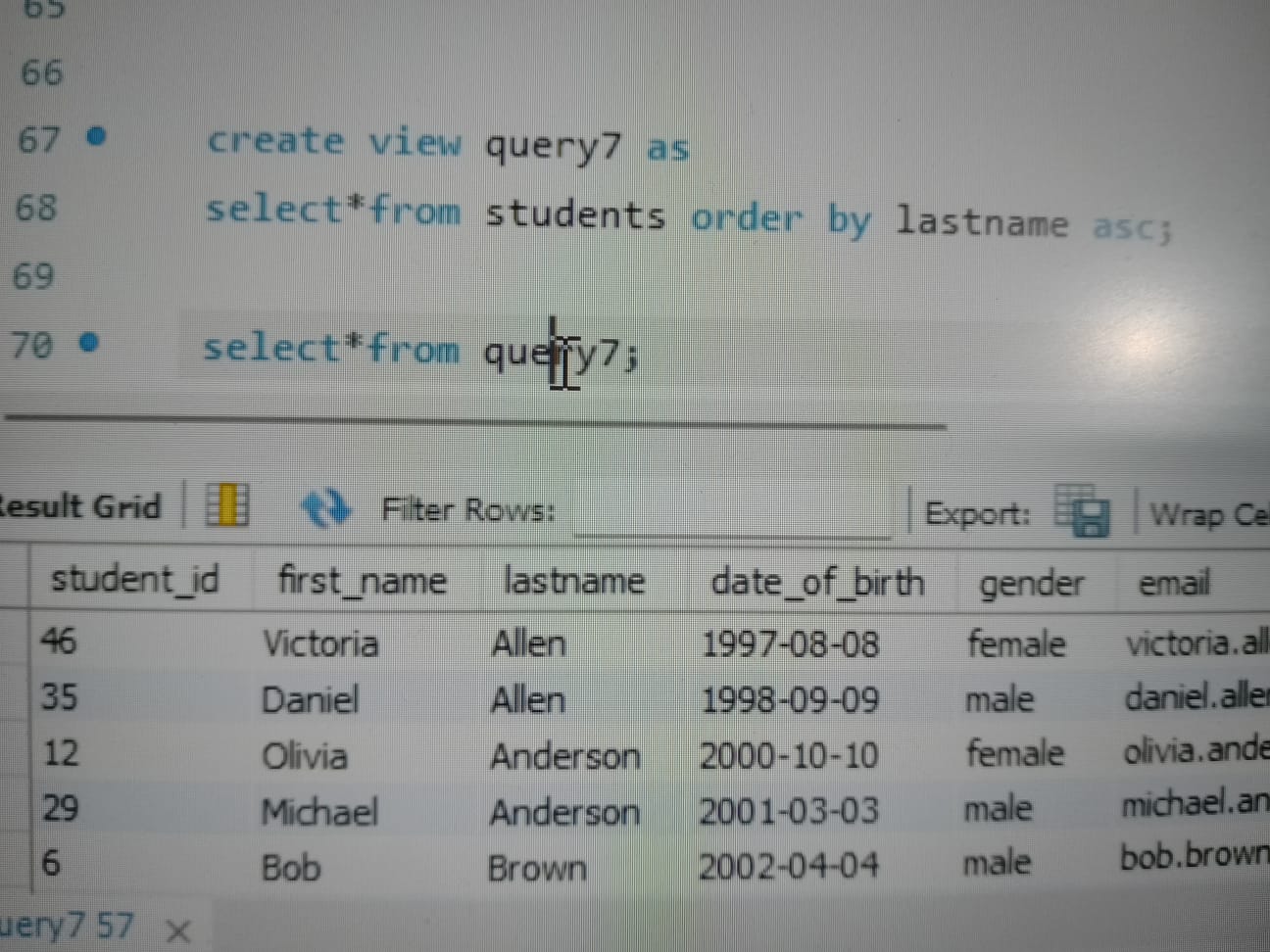
****

**select\*from query6;**

**7.** Retrieve the full details of the Students table, sorted by last name in alphabetical order.

**create view query7 as**

**select\*from students order by lastname asc;**

****

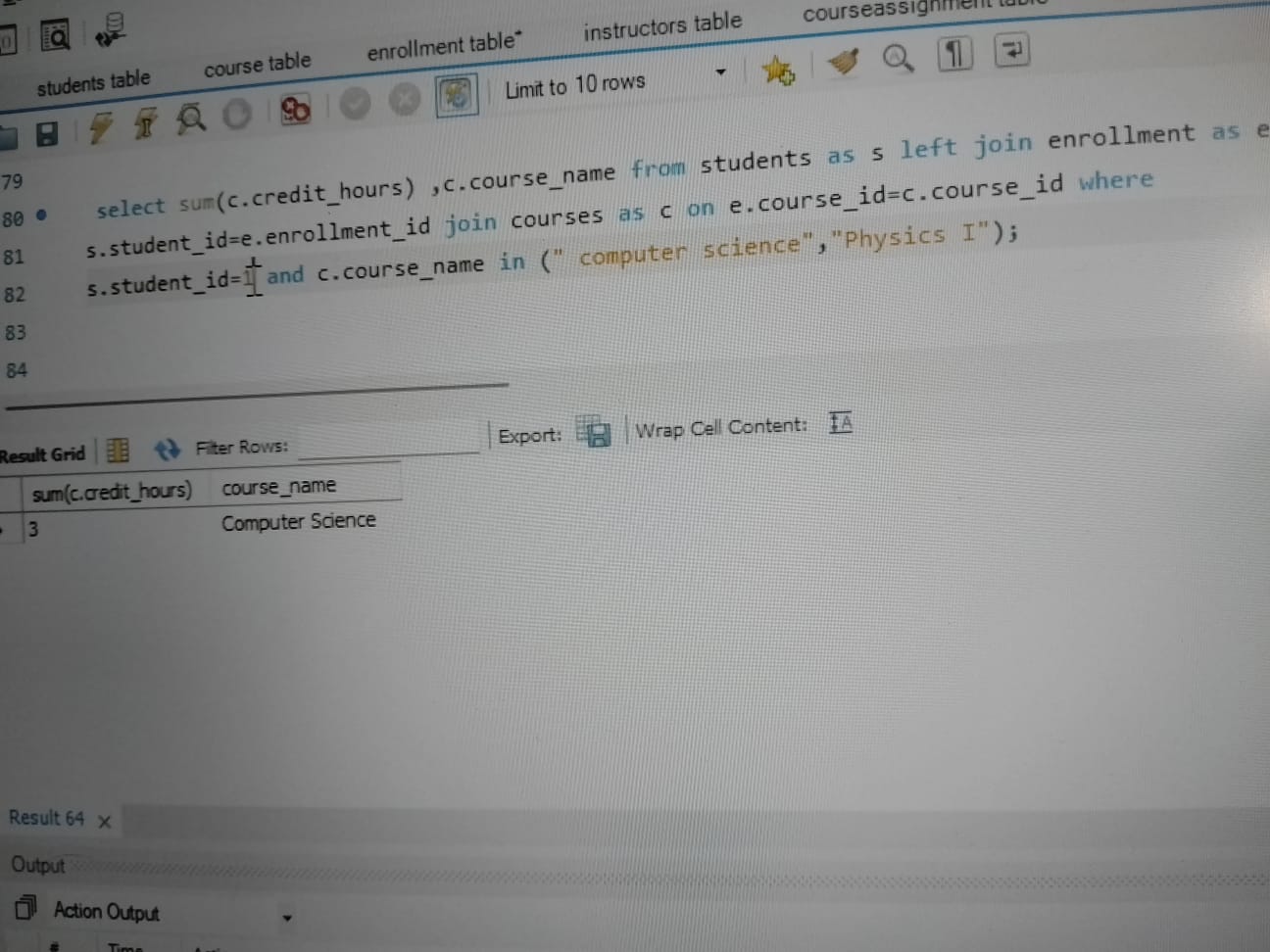
**select\*from query7;**

**8.** Find the total number of credit hours for a student enrolled in the Database Systems and Web Development courses.

**select sum(c.credit\_hours) from students as s left join enrollment as e on**

**s.student\_id=e.enrollment\_id join courses as c on e.course\_id=c.course\_id where**

**s.student\_id=3 and c.course\_name in (" computer science","Physics I");**

****

**9.** Identify the student with the maximum number of enrollments.

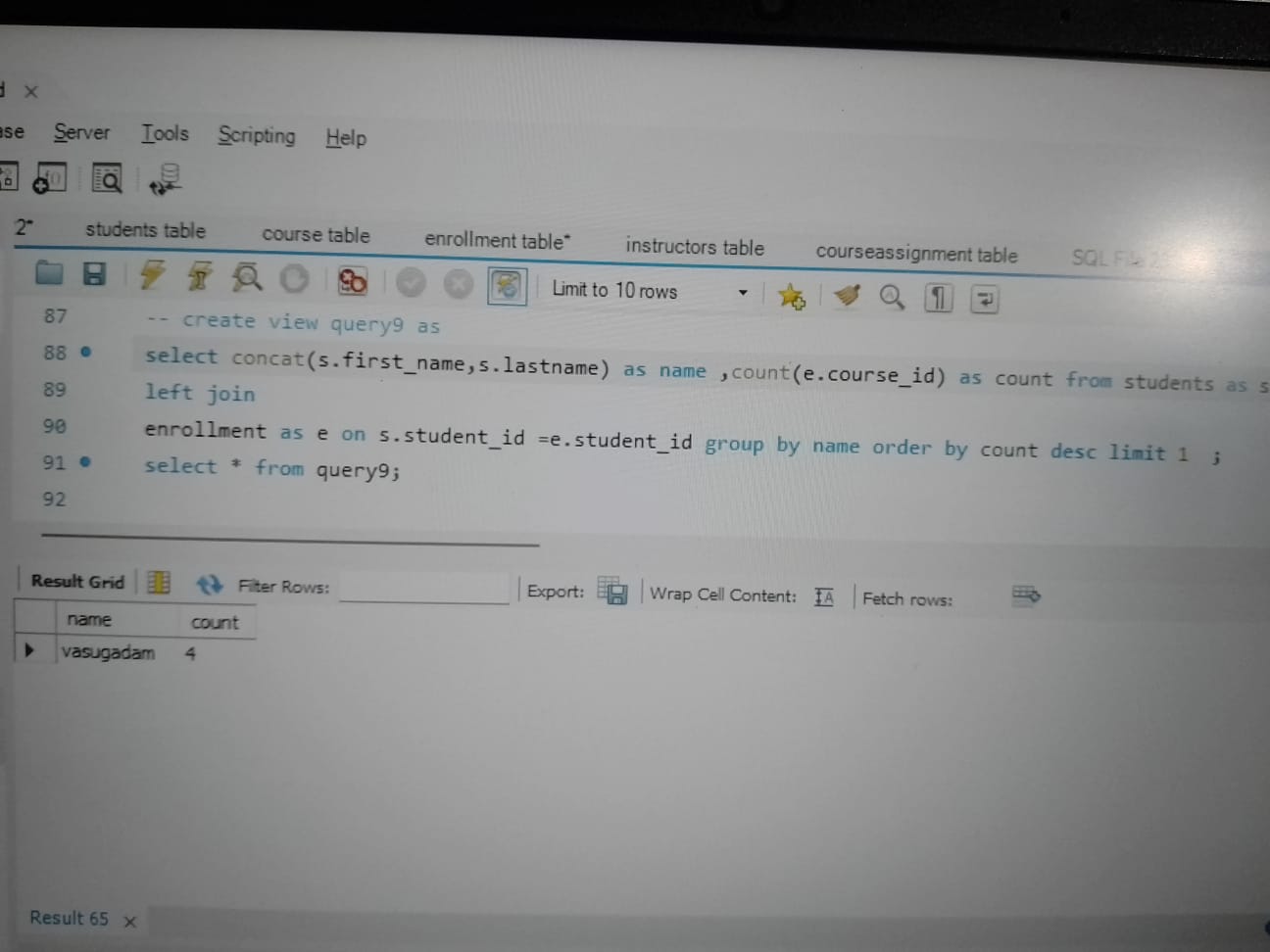
**create view query9 as**

**select concat(s.first\_name,s.lastname) as name ,count(e.course\_id) as count from students as s**

**left join**

**enrollment as e on s.student\_id =e.student\_id group by name order by count desc limit 1 ;**

**select \* from query9;**

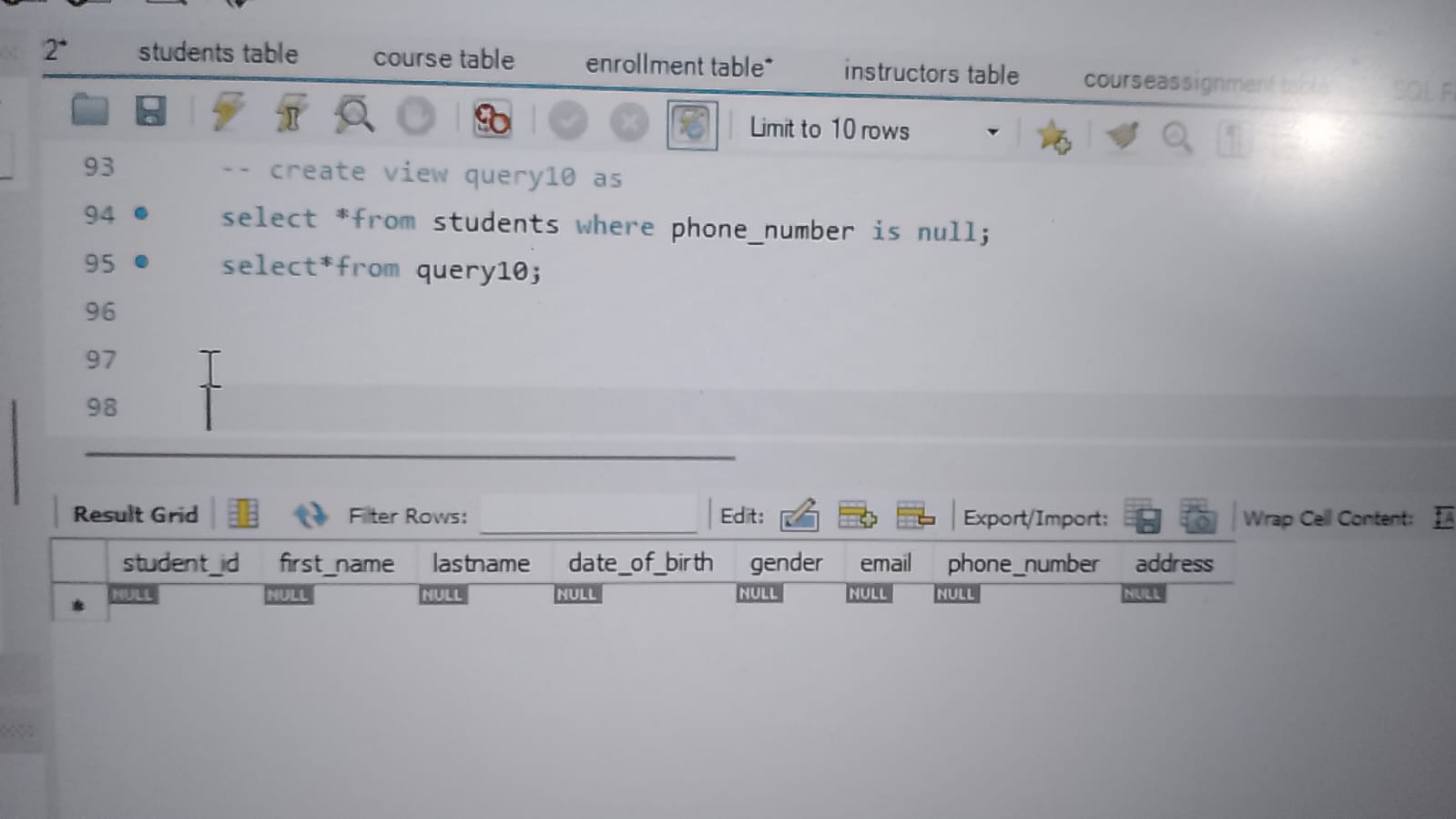
****

**10.** Get the details of students whose phone numbers are missing.

**create view query10 as**

**select \*from students where phone\_number is null;**

**select\*from query10;**

****

**Conclusion:**

From this data we can easily get any kind of information related to students.It is very usefull in colleges ,schools whenever we need know some particular data.It secure because only admin ca modifys and delete the data .

**Online Tutorial and Reference’s:**

• W3School HTML/CSS Tutorials, References and Examples @ http://www.w3schools.com/. (W3School is not related to W3C**)**

**.**MySQL Mother Site @ [www.mysql.com](http://www.mysql.com).

• MySQL 5.7 "Reference Manual" @ http://dev.mysql.com/doc/.

• MySQL 5.7 "SQL Statement Syntax" @ http://dev.mysql.com/doc/refman/5.7/en/sql-syntax.html.