1.Introduction

Background:

The objective of the project is to create one database called student management in this project 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 the details of every single person indetail with easy way using SQL/MYSQL 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-**count(),sum(),min(),max(),avg() **9.string functions and date functions and numeric functions  
10.joins-(inner join,left join,right join,full join)  
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(first\_name, last\_name, date\_of\_birth, gender, email, phone\_number, address)

**VALUES**

('John', 'Doe', '2000-01-15', 'Male', 'john.doe1@example.com', 1234567890, '123 Main St'),

('Jane', 'Smith', '1999-02-20', 'Female', 'jane.smith2@example.com', 1234567891, '456 Elm St'),

('Alice', 'Johnson', '1998-03-25', 'Female', 'alice.johnson3@example.com', 1234567892, '789 Oak St'),

('Bob', 'Brown', '2001-04-30', 'Male', 'bob.brown4@example.com', 1234567893, '101 Pine St'),

('Charlie', 'Davis', '2002-05-05', 'Male', 'charlie.davis5@example.com', 1234567894, '202 Cedar St'),

('Diana', 'Evans', '2003-06-10', 'Female', 'diana.evans6@example.com', 1234567895, '303 Maple St'),

('Eve', 'Foster', '2000-07-15', 'Female', 'eve.foster7@example.com', 1234567896, '404 Birch St'),

('Frank', 'Green', '1999-08-20', 'Male', 'frank.green8@example.com', 1234567897, '505 Cherry St'),

('Grace', 'Harris', '1998-09-25', 'Female', 'grace.harris9@example.com', 1234567898, '606 Walnut St'),

('Henry', 'Ivy', '2001-10-30', 'Male', 'henry.ivy10@example.com', 1234567899, '707 Ash St'),

('Isabella', 'Jones', '2002-11-05', 'Female', 'isabella.jones11@example.com', 1234567800, '808 Beech St'),

('Jack', 'King', '2003-12-10', 'Male', 'jack.king12@example.com', 1234567801, '909 Fir St'),

('Karen', 'Lewis', '2000-01-15', 'Female', 'karen.lewis13@example.com', 1234567802, '1010 Spruce St'),

('Liam', 'Miller', '1999-02-20', 'Male', 'liam.miller14@example.com', 1234567803, '1111 Poplar St'),

('Mia', 'Nelson', '1998-03-25', 'Female', 'mia.nelson15@example.com', 1234567804, '1212 Dogwood St'),

('Noah', 'Owens', '2001-04-30', 'Male', 'noah.owens16@example.com', 1234567805, '1313 Willow St'),

('Olivia', 'Perez', '2002-05-05', 'Female', 'olivia.perez17@example.com', 1234567806, '1414 Pine St'),

('Paul', 'Quinn', '2003-06-10', 'Male', 'paul.quinn18@example.com', 1234567807, '1515 Elm St'),

('Quincy', 'Reed', '2000-07-15', 'Male', 'quincy.reed19@example.com', 1234567808, '1616 Oak St'),

('Rachel', 'Stone', '1999-08-20', 'Female', 'rachel.stone20@example.com', 1234567809, '1717 Cedar St'),

('Sam', 'Turner', '1998-09-25', 'Male', 'sam.turner21@example.com', 1234567810, '1818 Maple St'),

('Tina', 'Underwood', '2001-10-30', 'Female', 'tina.underwood22@example.com', 1234567811, '1919 Birch St'),

('Uma', 'Vasquez', '2002-11-05', 'Female', 'uma.vasquez23@example.com', 1234567812, '2020 Cherry St'),

('Victor', 'White', '2003-12-10', 'Male', 'victor.white24@example.com', 1234567813, '2121 Walnut St'),

('Wendy', 'Xavier', '2000-01-15', 'Female', 'wendy.xavier25@example.com', 1234567814, '2222 Ash St'),

('Xander', 'Young', '1999-02-20', 'Male', 'xander.young26@example.com', 1234567815, '2323 Beech St'),

('Yara', 'Zimmerman', '1998-03-25', 'Female', 'yara.zimmerman27@example.com', 1234567816, '2424 Fir St'),

('Zack', 'Adams', '2001-04-30', 'Male', 'zack.adams28@example.com', 1234567817, '2525 Spruce St'),

('Amy', 'Baker', '2002-05-05', 'Female', 'amy.baker29@example.com', 1234567818, '2626 Poplar St'),

('Brian', 'Carter', '2003-06-10', 'Male', 'brian.carter30@example.com', 1234567819, '2727 Dogwood St'),

('Clara', 'Diaz', '2000-07-15', 'Female', 'clara.diaz31@example.com', 1234567820, '2828 Willow St'),

('David', 'Edwards', '1999-08-20', 'Male', 'david.edwards32@example.com', 1234567821, '2929 Pine St'),

('Ella', 'Fisher', '1998-09-25', 'Female', 'ella.fisher33@example.com', 1234567822, '3030 Elm St'),

('George', 'Garcia', '2001-10-30', 'Male', 'george.garcia34@example.com', 1234567823, '3131 Oak St'),

('Hannah', 'Hernandez', '2002-11-05', 'Female', 'hannah.hernandez35@example.com', 1234567824, '3232 Cedar St'),

('Ian', 'Jackson', '2003-12-10', 'Male', 'ian.jackson36@example.com', 1234567825, '3333 Maple St'),

('Julia', 'Kennedy', '2000-01-15', 'Female', 'julia.kennedy37@example.com', 1234567826, '3434 Birch St'),

('Kevin', 'Lopez', '1999-02-20', 'Male', 'kevin.lopez38@example.com', 1234567827, '3535 Cherry St'),

('Laura', 'Martinez', '1998-03-25', 'Female', 'laura.martinez39@example.com', 1234567828, '3636 Walnut St'),

('Michael', 'Nelson', '2001-04-30', 'Male', 'michael.nelson40@example.com', 1234567829, '3737 Ash St'),

('Natalie', 'Olsen', '2002-05-05', 'Female', 'natalie.olsen41@example.com', 1234567830, '3838 Beech St'),

('Oscar', 'Parker', '2003-06-10', 'Male', 'oscar.parker42@example.com', 1234567831, '3939 Fir St');

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**

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

('Engg Math I', 'Fundamentals of calculus, linear algebra, and differential equations.', 4),

('computer science', 'Basic principles of physics with applications in engineering.', 4),

('Engg Chemistry', 'Introduction to chemical principles and their applications in engineering.', 3),

('computer science', 'Fundamentals of electrical engineering, including circuit analysis and electromagnetism.', 4),

('os', 'Study of forces, motion, and energy in engineering systems.', 4),

('Comp Prog', 'Introduction to programming concepts using a high-level programming language.', 3),

('computer science', 'Concepts of data organization, algorithms, and their implementation.', 3),

('Digital Logic', 'Fundamentals of digital systems, logic gates, and circuits.', 3),

('os', 'Study of semiconductor devices and electronic circuits.', 4),

('engg math I', 'Introduction to continuous and discrete-time signals and systems.', 3),

('daa', 'Architecture and programming of microprocessors and microcontrollers.', 4),

('OS', 'Concepts of operating systems, including processes, memory management, and file systems.', 3),

('DBMS', 'Introduction to database concepts, models, and SQL.', 3),

('Comp Networks', 'Study of networking concepts, protocols, and architectures.', 3),

('Soft Engg', 'Principles of software development, project management, and quality assurance.', 3);

select\*from 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, enrollment\_date)**

**VALUES**

(1, 1, '2024-08-01'),

(2, 2, '2024-08-01'),

(3, 3, '2024-08-01'),

(4, 4, '2024-08-02'),

(5, 5, '2024-08-02'),

(6, 6, '2024-08-02'),

(7, 7, '2024-08-03'),

(8, 8, '2024-08-03'),

(9, 9, '2024-08-03'),

(10, 10, '2024-08-04');

select\*from enrollment;

**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**

('John', 'Doe', 'john.doe@example.com', '123-456-7890'),

('Jane', 'Smith', 'jane.smith@example.com', '123-456-7891'),

('Alice', 'Johnson', 'alice.johnson@example.com', '123-456-7892'),

('Bob', 'Brown', 'bob.brown@example.com', '123-456-7893'),

('Charlie', 'Davis', 'charlie.davis@example.com', '123-456-7894'),

('Diana', 'Evans', 'diana.evans@example.com', '123-456-7895'),

('Frank', 'Green', 'frank.green@example.com', '123-456-7896'),

('Grace', 'Harris', 'grace.harris@example.com', '123-456-7897'),

('Henry', 'Ivy', 'henry.ivy@example.com', '123-456-7898'),

('Isabella', 'Jones', 'isabella.jones@example.com', '123-456-7899');

select\*from 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),

(2, 2),

(3, 3),

(4, 4),

(5, 5),

(6, 6),

(7, 7),

(8, 8),

(9, 9),

(10, 10);

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**

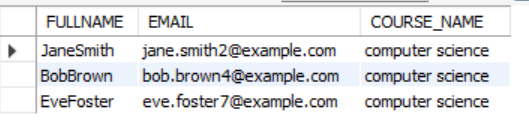
select concat( s.first\_name,s.last\_name)as FULLNAME,

s.EMAIL,c.COURSE\_NAME from students as s

inner 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';

****

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

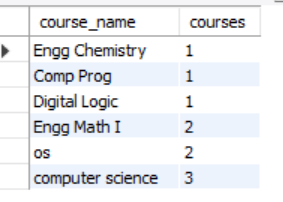
**Query:**

**select course\_name,count(c.course\_id) as courses from courses as c**

**left join enrollment as e on c.course\_id=e.course\_id**

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

**group by course\_name order by courses;**

****

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

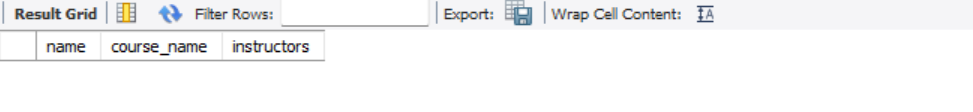
**Query:**

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

**c.course\_name ,count(course\_id) as instructors from instructors as i**

**left join courses as c on i.instructor\_Id=c.course\_id**

**group by name,c.course\_name having instructors >2;**

****

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

**Query:**

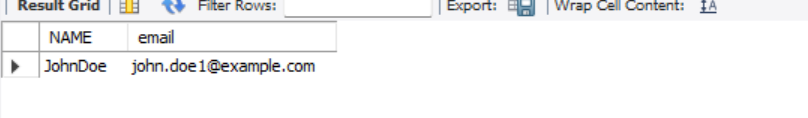
**select CONCAT(S.first\_name,S.LAST\_NAME)AS NAME,**

**S.email from STUDENTS as S join enrollment as e on s.student\_id=e.student\_id**

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

**join instructors as i on c.course\_id=i.instructor\_id**

**where i.first\_name='john' and i.last\_name='doe';**



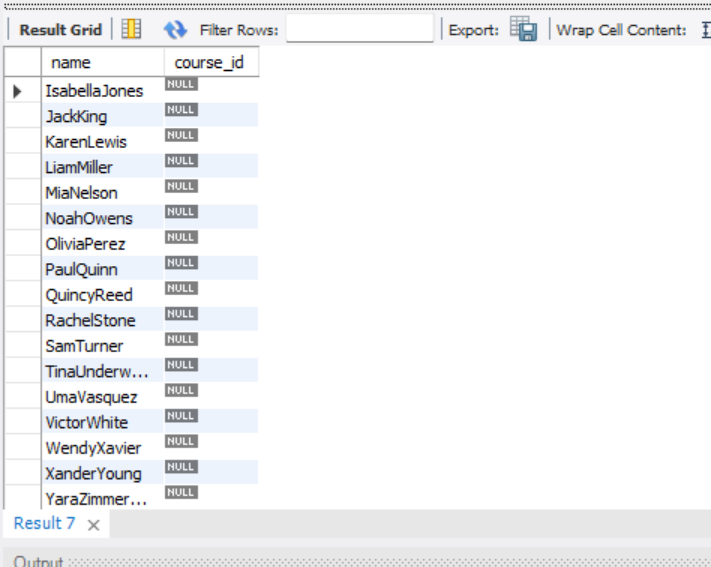
**5**.List all students who have not enrolled in any course.

**Query:**

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

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

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

****

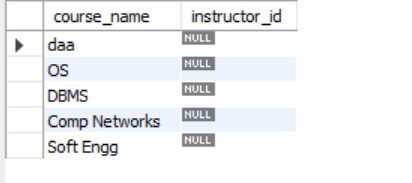
**6**.Get the list of courses with no assigned instructor.

**Query:**

**select c.course\_name,instructor\_id from courses as c**

**left join instructors as i on c.course\_id=i.instructor\_id**

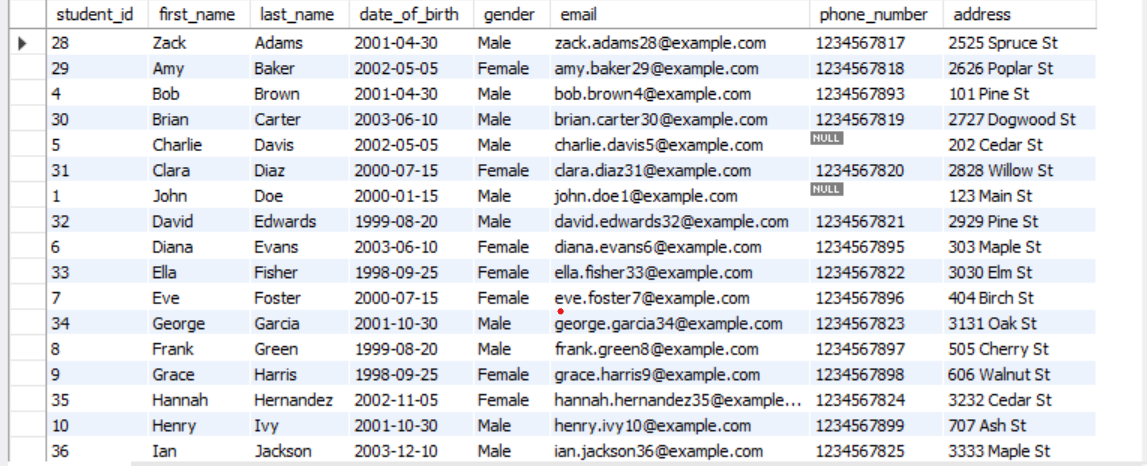
**where i.instructor\_id is null ;**

****

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

**Query:**

**select\*from students order by last\_name;**

****

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

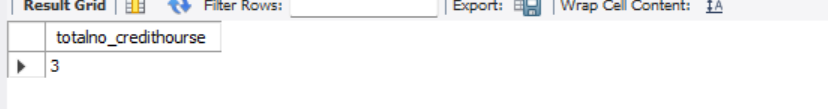
**Query:**

**select count(credit\_hours) as totalno\_credithourse from courses as c**

**left join students as s on course\_id=student\_id**

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

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

****

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

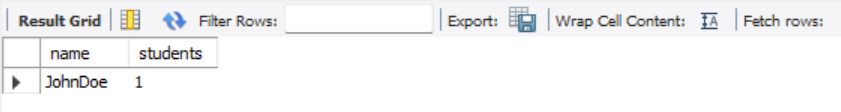
**Query:**

**select concat(s.first\_name,s.last\_name) as name,**

**count(s.student\_id) as students from students as s left join**

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

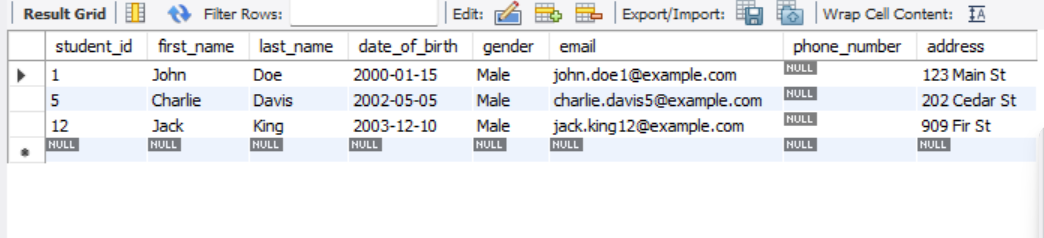
**group by name order by students desc limit 1 ;**

****

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

**Query:**

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

****

**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 to know some particular data.It secure because only admin can modifies and delete the data .

**Online Tutorial and Reference’s:**

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