SE Club: Students Database Project

ANALYIZING PHASE

I. IDENTIFIYING PROBLEM

The Club desires to have its own website database backend that contains dynamic data for a large number of students for different majors & minors based on their academic level, courses they have taken, and their academic GPA.

It is crucial to maintain & build a relational database instead of creating a static excel spreadsheet due to changes will be made in every other place that references that data. In addition, to benefit from the features provided by the website based on the student data.

II. IDENTIFIYING REQUIREMENTS

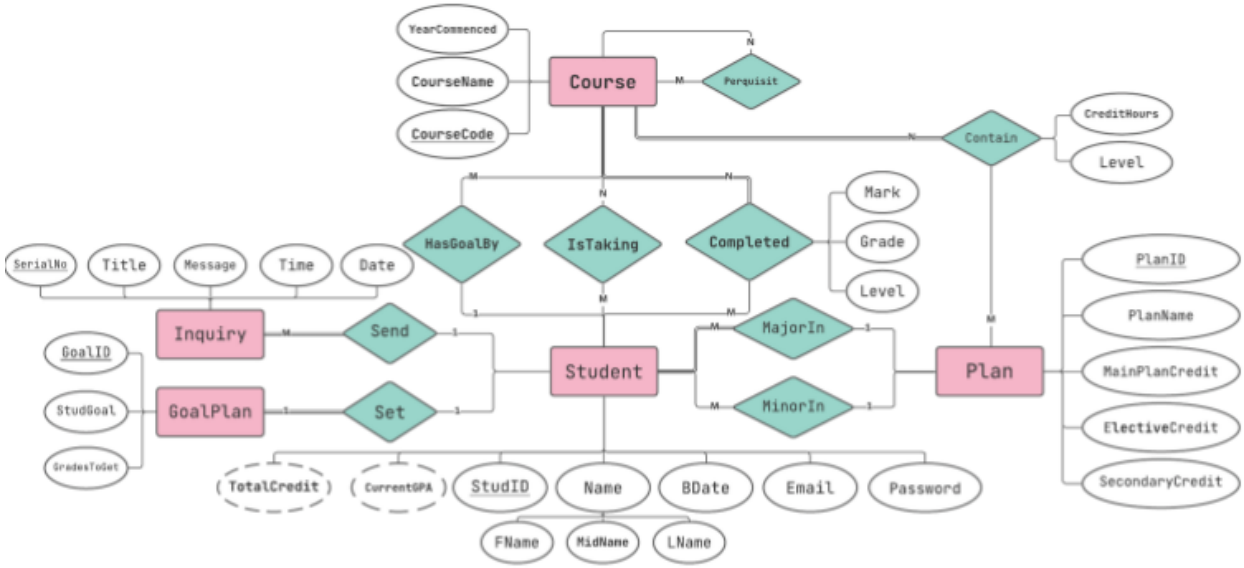
A. Main Requirements on the website are defined as the following:

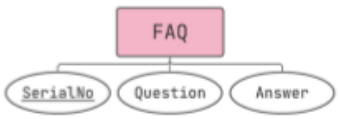
- Each Student That Signs Up should be having data Entry Such as (Email, Password, Major, Minor if any, Courses That have been taken, and total Credit.
- Based On the data that has been entered the user can view the upcoming courses smoothly according to their major & minor if any.
- Feature Set a Goal's aim is to let the student choose the GPA they want to get in the upcoming semester that needs to be achieved via taking specific courses that will increase their GPA according to their goal, according to their major & minor if any.
- The Students Can send inquires if they do have and it will show up with a specific date & Time.

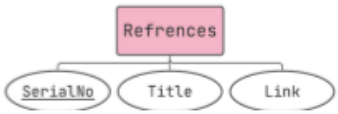
- In the worst-Case Scenario that occurred in xyzUni, a specific Course from a major, that is mutual in another major has different credit hours. The Database Should be differentiating credits for both of the majors based on the academic level.
- Each student is able to know their Current GPA.
- Each Student is able to view their grade/mark in their completed courses.
- The student will be notified if there is a prerequisite course for their upcoming courses.
- The Database has a FAQ that contains Questions and their answers.
- The Database has a Reference that contains several links stored in the DB

DESIGNIG PHASE

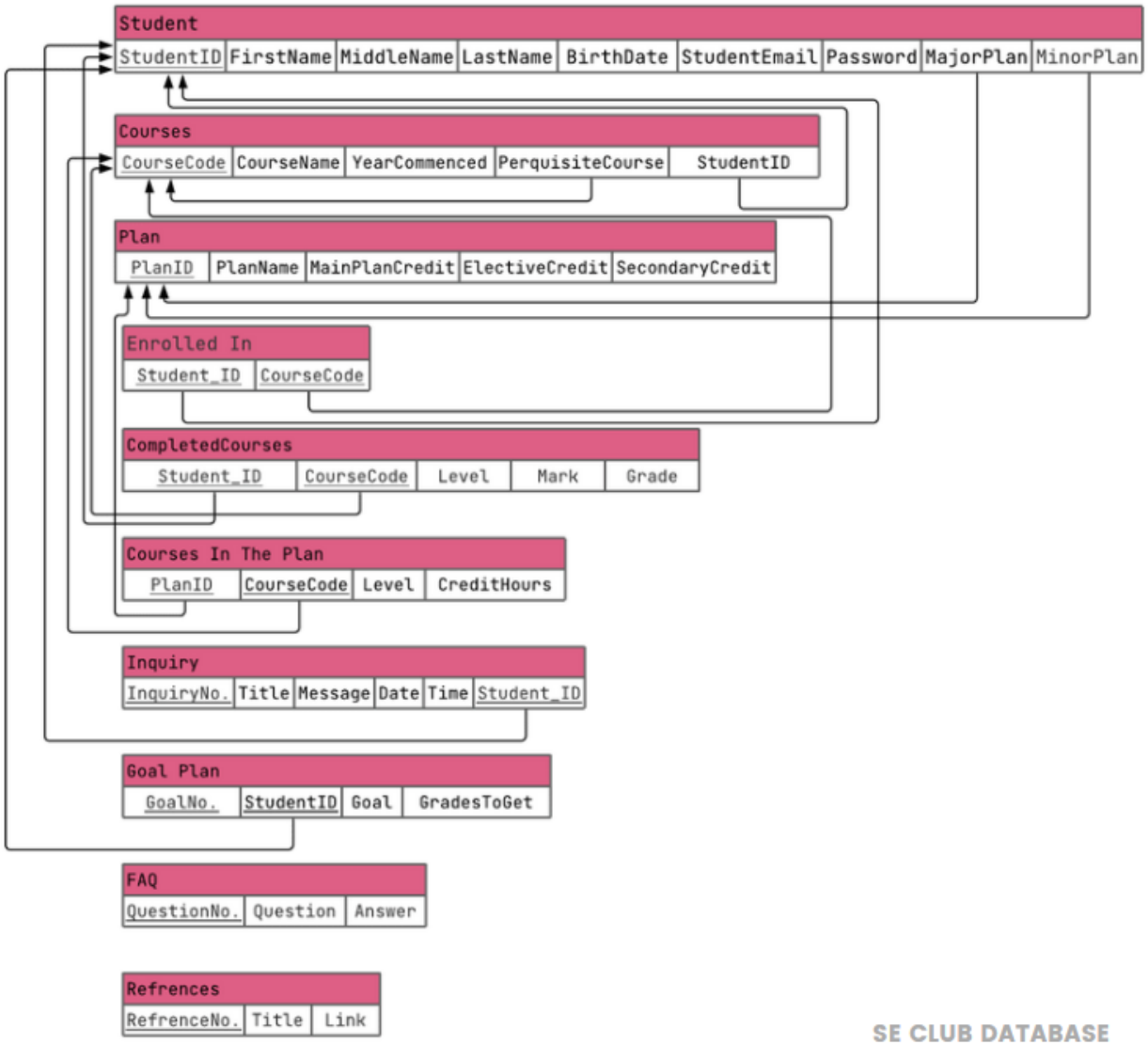
ER Diagram (Before Normalization).







Relational Schemas



III. DATABASE ENTITY DESCRIBTION

We created an entity called student so when the student signs up into the website we could keep track of the student's full name, date of birth, ID which is provided by xyz university also his university email, and password.

The entity has a relationship with course to indicate whether the student complete or currently taking a particular course. Regarding a feature on the website each course will need a goal to be associated with it if the student wants to benefit from this feature to set a goal plan for his GPA where we include information as Serial No., the goal that the student will set and what grades the student have to get in order to achieve this goal. We also want the information of each course such as its name, code. The plan majored in or minored in by each student also are being tracked including some basic information on the plan such as its name, ID, the main, elective and secondary credit of the plan. The plans contain the courses with level and their credit hours.

We will keep track of inquiries that student will send, and each inquiry will have serial No., the message and the title of that message, time and date when the inquiry is sent. Then regarding to the nature of the website we will need to keep track of links used on the website to make searching and using these links easier, so the actual link will be associated to a serial number and its title. Lastly, we have FAQ including some frequently asked questions with their answers.

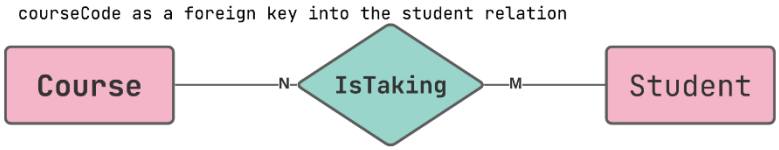
IV. ENTITY DESCRIBTION SLIGHT CHANGES (1NF)

After observation of the entites we found specific relations will not be as in it is prober relation. Therefore, it will occur specific errors in the scheme. For instance, the prequisite courses differs from program to another , it cannot be in the courses relation due to its dependacy on the program itself thereafter we added it to the relationship Courses in the plan.

In addition, we removed Grades to get from the Goal plan relation to be in a new relation called CoursesHasGoalByStudent that connects the student and the courses relations. The purpose of stating this change is to identify clarity to the courses that needs to be scored rather the than solely stating grades to goal which will not be concised to the end user.

Mentiong a case in specifics occurred during applying the N:M relation without seperating the PK’s in a different relation as follows

Since the student is the one who is taking the course, we took the courseCode as a foreign key into the student relation



So the schema for the student relation will be like this:

Student									
StudentID	FirstName	MiddleName	LastName	BirthDate	StudentEmail	Password	MajorPlan	MinorPlan	CourseCode

But When we try to fill values in the relation, we can see that the CourseCode is a multivalued Attribute so to take this relation to the first normal form we will take the unique values we need and create another relation Containing both CourseCode and StudentID

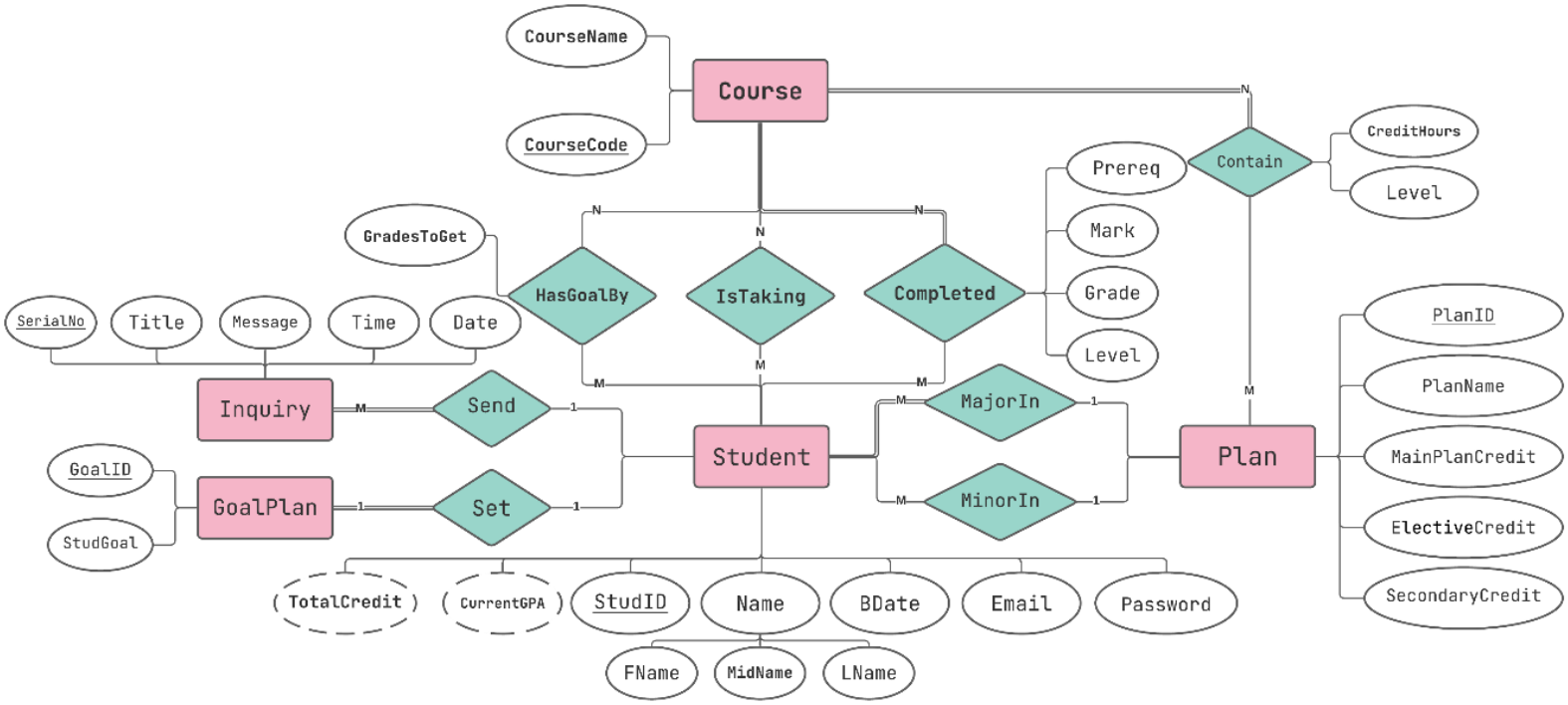
Student									
StudentID	FirstName	MiddleName	LastName	BirthDate	StudentEmail	Password	MajorPlan	MinorPlan	CourseCode
4111650	Sara	Mohammad	Aljuhani	25-03-2001	4111650@upm.edu.sa	*****	FC	Null	CS211 CS351 CS314
4107551	Reem	Ahmed	Alamri	23-12-2000	4107551@upm.edu.sa	*****	SE	FC	CS464 SE311 SE323

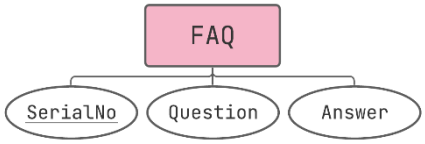
Now it will be in the FIRST NORMAL FORM:

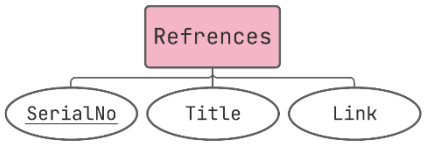
StudentCurrentlyTaking	
Student_ID	CourseCode
4111650	CS211
4111650	CS351
4111650	CS314
4107551	CS464
4107551	SE311
4107551	SE323

DESIGNIG PHASE AFTER APPLYING THE (1NF)

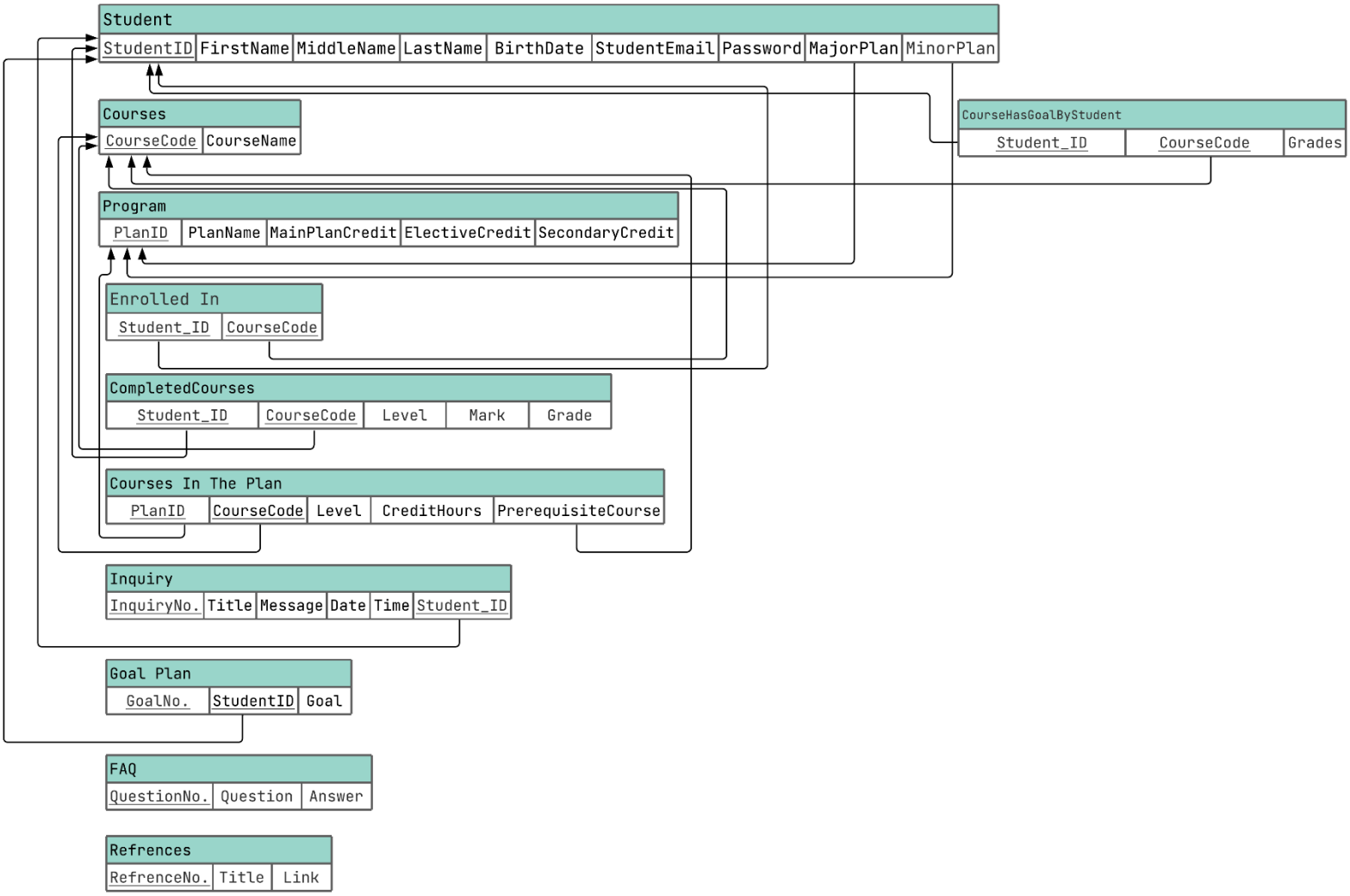
ER Diagram







Relational Schemas



DESCRIBTION PHASE

Program Course(3 rows)

We created a table for programs which contain information such Program Code, Program Name Main Credit which are courses that starts with the program code e.g (SE 262), Elective Credit, and Secondary Credit and this is the credit for courses that many programs could have e.g(MATH 101, GSOS xxx).

```
12 SELECT * FROM Program;
13
```

Results Explain Describe Saved SQL History

PROGRAMCODE	PROGRAMNAME	MAINCREDIT	ELECTIVECREDIT	SECONDARYCREDIT
NULL	NULL	0	0	0
SE	SOFTWARE ENGINEERING	126	120	50
FC	FORENSIC CYBER	123	118	49

3 rows returned in 0.00 seconds Download

Student Table(13 rows)

We created a table for student so when a student signs up into the website we keep track of the following information

```
12 SELECT * FROM Student;
13
```

Results Explain Describe Saved SQL History

STUDENTID	FIRSTNAME	MIDDLENAME	LASTNAME	STUDENTEMAIL	PASSWORD	MAJORPROGRAM	MINORPROGRAM	BIRTHDATE

Course Table (72 rows)

We created just two attributes for the course relation containing just the name and code. We know that it is more efficient to add more attributes and we are able to add more information such as level, credit hours, and prerequisite for each course but since there is a problem where some courses has same code as well as the name but they differ in the level or credit hours or their requisite based on the plan we created our table this way and added a table for the relationship between course and plan clarifying the course code, level, and credit hours for the course in a specific plan

```
SELECT * FROM Course;
```

Results | Explain | Describe | Saved SQL | History

FC 382	Defense Mechanisms
FC 304	Digital Forensic Tools and Techniques
FC 491	Capstone Project I
FC 411	Secure Network Design
FC xxx1	Professional Elective I
FC 421	Applied Cryptography
FC 492	Capstone Project II
FC 472	Security and Privacy Policies
FC xxx2	Professional Elective II
FC 462	Security Risk Management
SE 394	Summer Training
FC 394	Summer Training

4111674@upm.edu.sa somayaupm en Copyright © 1999, 2021, Oracle and/or its affiliates.

CourseInProgram Table(70 Rows)

This table is created to clarifying the program code, course code, level, and credit hours for the courses.

```
12 SELECT * FROM COURSESINPROGRAM;
13
```

Results Explain Describe Saved SQL History

PROGRAMCODE	COURSECODEINPROGRAM	PROGRAMLEVEL	CREDITHOURS	PREREQUISITE
SE	ENGL 102	2	3	ENGL 101
SE	ENGL 201	3	3	ENGL 102
SE	STAT 232	4	3	MATH 101
FC	CS 111	1	4	PCS 001
FC	PHYS 101	1	4	MATH 002
FC	ENGL 101	1	3	ENGL 005
FC	CS 112	2	4	CS 111
FC	ENGL 102	2	3	ENGL 101
FC	CS 351	3	4	CS 112
FC	ENGL 201	5	3	ENGL 102

StudentCompleted Table(36 Rows)

We created this table to track the courses that the student completed with the level when it completed and the marks the student has got

```
12 SELECT * FROM STUDENTCOMPLETED;
13
```

Results Explain Describe Saved SQL History

STUDENTID	COMPLETEDCOURSECODE	COURSEMARK	COMPINLEVEL	COURSEGRADE
4010024	MATH 001	92	0	A
4010024	MATH 002	93	0	A
4010024	MATH 101	80	1	B
4010024	MATH 102	96	2	A+
4010024	PCS 001	85	0	B+
4010024	CS 111	93	1	A
4010024	CS 112	98	2	A+
4010024	ENGL 101	96	1	A+
4010024	ENGL 102	88	2	B+
3910221	MATH 001	92	0	A

CoursesStudentsEnrolledInto Table(12 Rows)

We created this table to track the courses that the student is currently taking

```
SELECT * FROM CoursesStudentEnrolledInto;
```

Results Explain Describe Saved SQL History

STUDENTID	CURRENTCOURSES
3910221	CS 201
3910221	CS 211
3910221	CS 351
3910221	MATH 202
4010024	CS 201
4010024	CS 211
4010024	CS 351
4010024	MATH 202
4010064	CS 201
4010064	CS 211

StudentGoal Table(3 Rows)

This table is created to store the data of the student's goal whenever the student decide to create a goal for his/her grades.

```
SELECT * FROM STUDENTGOAL;
```

Results Explain Describe Saved SQL History

GOALNO	STUDENTID	STUDENTGOAL
1	4010064	3.8
2	3910221	2.5
3	3940022	4

3 rows returned in 0.01 seconds Download

CourseGoalByStudent Table(11 rows)

This table is created to store the data of the courses of the student and clarifying the grades he/she needs to get in order to achieve his/her goal.

```
SELECT * FROM COURSEGOALBYSTUDENT;
```

Results | Explain | Describe | Saved SQL | History

STUDENTID	GOALEDCOURSES	GRADESTOGET
4010064	SE 262	B+
4010064	STAT 232	A
3910221	MATH 202	B
4010064	MATH 202	A+
4010064	CS 224	A
3910221	CS 351	B
3910221	CS 201	C+

Inquiry Table

This table is created to store the data of the messages that student will send and the response of their inquiries.

```
SELECT * FROM Inquiry;
```

Results | Explain | Describe | Saved SQL | History

INQUIRYNO	STUDENTID	TITLE	MESSAGE	INQUIRYDATE	INQUIRYTIME
2	4010024	Studying Math 205	DEAR ALL WE DESIRE TO study math 205 as student, please inform us if there is a chance	03/28/2012	05-DEC-21 05.26.47.090781 PM
3	3910221	Complaint letters	Good Morning, where I could find Complaint letters, please inform us if there is a chance	09/28/2020	05-DEC-21 05.26.47.090781 PM
1	4010064	TRAVEL ABROAD	DEAR ALL WE DESIRE TOO TRavel abroad as student, please inform us if there is a chance	03/28/2012	05-DEC-21 04.26.47.090781 PM

3 rows returned in 0.01 seconds Download

FAQ Table

This table is created to store all FAQ with their answers to display them into the website and to allow searching for some common questions

```
SELECT * FROM FAQ;
```

Results | Explain | Describe | Saved SQL | History

QUESTIONNO	QUESTION	ANSWER
1	DEAR ALL WE DESIRE TOO Know how many departments u have?	Hey! tHEY ARE FOUR DEPARTMENTS
2	HEY EVERYONE, MAY WE KNOW IF YOU HAVE OTHER UNI BRANCHES?	We apologizize we only have in Madinah
3	Good Morning, How much should i pay to get accepted in the CS field	Hey, There!, Its prob less than a million ;)

RETRIEVAL EXPRESSIONS PHASE

I. Program Table Queries

1. *View all SE program information:*

SELECT * FROM Program

WHERE ProgramCode='SE';

```
--Queries of Program table --
SELECT * FROM Program
WHERE ProgramCode='SE';
```

Results | Explain | Describe | Saved SQL | History

PROGRAMCODE	PROGRAMNAME	MAINCREDIT	ELECTIVECREDIT	SECONDARYCREDIT
SE	SOFTWARE ENGINEERING	126	120	50

1 rows returned in 0.01 seconds Download

2. *View all FC program information:*

SELECT * FROM Program

WHERE ProgramCode='FC' AND MainCredit=123;

```
SELECT * FROM Program
WHERE ProgramCode='FC' AND MainCredit=123;
```

Results | Explain | Describe | Saved SQL | History

PROGRAMCODE	PROGRAMNAME	MAINCREDIT	ELECTIVECREDIT	SECONDARYCREDIT
FC	FORENSIC CYBER	123	118	49

1 rows returned in 0.01 seconds Download

3. *View Information about FC and any program that have "Engineer" on its name:*

SELECT * FROM Program

WHERE ProgramCode='FC' OR ProgramName Like '%ENGINEER%';

```
SELECT * FROM Program
WHERE  ProgramCode='FC' OR ProgramName Like '%ENGINEER%';
```

Results | Explain | Describe | Saved SQL | History

PROGRAMCODE	PROGRAMNAME	MAINCREDIT	ELECTIVECREDIT	SECONDARYCREDIT
SE	SOFTWARE ENGINEERING	126	120	50
FC	FORENSIC CYBER	123	118	49

2 rows returned in 0.01 seconds Download

4. ***View Information about FC or the program with name SE:***

SELECT * FROM Program

WHERE ProgramCode='FC' OR ProgramName='SE';

Results Explain Describe Saved SQL History

PROGRAMCODE	PROGRAMNAME	MAINCREDIT	ELECTIVECREDIT	SECONDARYCREDIT
NULL	NULL	0	0	0
FC	FORENSIC CYBER	125	118	49

5. ***View Information about the program where its code starts with F:***

SELECT * FROM Program

WHERE ProgramCode LIKE 'F%';

Results Explain Describe Saved SQL History

PROGRAMCODE	PROGRAMNAME	MAINCREDIT	ELECTIVECREDIT	SECONDARYCREDIT
FC	FORENSIC CYBER	125	118	49

1 rows returned in 0.01 seconds Download

II. Student Table Queries

1. ***View Information about students on FC program:***

SELECT * FROM Student

WHERE MajorProgram='FC';

2. ***View Information about student with ID 4010064 and Major on SE:***

SELECT * FROM Student

WHERE StudentID = xxxxxxx AND MajorProgram='SE';

III. StudentCompleted Table Queries

1. ***View Information about students who completed CS 211:***

SELECT * FROM StudentCompleted

WHERE CompletedCourseCode = 'CS 112';

STUDENTID	COMPLETEDCOURSECODE	COURSEMARK	COMPINLEVEL	COURSEGRADE
4010024	CS 112	98	2	A+
3910221	CS 112	98	2	A+
3910022	CS 112	98	2	A+
4010064	CS 112	100	2	A+

4 rows returned in 0.01 seconds Download

2. View Information about completed courses by students with ID 4010064, 4010024 :

SELECT CompletedCourseCode FROM

StudentCompleted WHERE

StudentID IN (4010064, 4010024);

```
SELECT CompletedCourseCode FROM StudentCompleted WHERE StudentID IN (4010064,4010024);
```

Results | Explain | Describe | Saved SQL | History

COMPLETEDCOURSECODE
CS 111
CS 112
ENGL 101
ENGL 102
MATH 001
MATH 002
MATH 101
MATH 102
PCS 001

3. View all Information about student's who got A+ , A :

SELECT * FROM StudentCompleted WHERE CourseGrade = 'A+' OR CourseGrade = 'A';

STUDENTID	COMPLETEDCOURSECODE	COURSEMARK	COMPINLEVEL	COURSEGRADE
4010024	MATH 001	92	0	A
4010024	MATH 002	93	0	A
4010024	MATH 102	96	2	A+
4010024	CS 111	93	1	A
4010024	CS 112	98	2	A+
4010024	ENGL 101	96	1	A+
3910221	MATH 001	92	0	A
3910221	MATH 002	93	0	A

4. View Information about all students except the student with ID 3910022 :

SELECT * FROM StudentCompleted

WHERE NOT StudentID = 3910022;

STUDENTID	COMPLETEDCOURSECODE	COURSEMARK	COMPINLEVEL	COURSEGRADE
4010024	MATH 001	92	0	A
4010024	MATH 002	93	0	A
4010024	MATH 101	80	1	B
4010024	MATH 102	96	2	A+
4010024	PCS 001	85	0	B+
4010024	CS 111	93	1	A
4010024	CS 112	98	2	A+

5. View all Information about student's who got grades from 90 to 95 :

SELECT * FROM StudentCompleted WHERE CourseMark BETWEEN 90 AND 95;

STUDENTID	COMPLETEDCOURSECODE	COURSEMARK	COMPINLEVEL	COURSEGRADE
4010024	MATH 001	92	0	A
4010024	MATH 002	93	0	A
4010024	CS 111	93	1	A
3910221	MATH 001	92	0	A
3910221	MATH 002	93	0	A
3910221	CS 111	93	1	A
3910022	MATH 001	95	0	A+
3910022	MATH 002	93	0	A

IV. CoursesStudentEnrolledInto Table Queries

1. *View Information about student's who are currently taking CS 211 :*

SELECT * FROM CoursesStudentEnrolledInto

WHERE CurrentCourses= 'CS 211';

STUDENTID	CURRENTCOURSES
3910221	CS 211
4010024	CS 211
4010064	CS 211

2. *View Information about current courses of the student with ID 1010064:*

SELECT CurrentCourses FROM CoursesStudentEnrolledInto WHERE StudentID IN (4010064);

CURRENTCOURSES
CS 201
CS 211
CS 351
MATH 202

3. *View current courses of the student with ID 4010064 or 4010024:*

SELECT * FROM CoursesStudentEnrolledInto

WHERE StudentID = 4010024 OR StudentID = 4010064;

STUDENTID	CURRENTCOURSES
4010024	CS 201
4010024	CS 211
4010024	CS 351
4010024	MATH 202
4010064	CS 201
4010064	CS 211
4010064	CS 351
4010064	MATH 202

4. *View current courses of all students except the student with ID 4010064:*

SELECT * FROM CoursesStudentEnrolledInto

WHERE NOT StudentID = 4010064;

STUDENTID	CURRENTCOURSES
3910221	CS 201
3910221	CS 211
3910221	CS 351
3910221	MATH 202
4010024	CS 201
4010024	CS 211
4010024	CS 351

5. ***View current courses of all students ordered by their ID:***

SELECT * FROM CoursesStudentEnrolledInto

ORDER BY StudentID;

STUDENTID	CURRENTCOURSES
3910221	CS 201
3910221	CS 211
3910221	CS 351
3910221	MATH 202
4010024	CS 201
4010024	CS 211
4010024	CS 351
4010024	MATH 202

V. Inquiry Table Queries

1. ***View the first inquiry:***

SELECT * FROM Inquiry WHERE InquiryNo=1;

INQUIRYNO	STUDENTID	TITLE	MESSAGE	INQUIRYDATE	INQUIRYTIME
1	4010064	TRAVEL ABROAD	DEAR ALL WE DESIRE TOO TRavel abroad as student, please inform us if there is a chance	03/28/2012	05-DEC-21 04.26.47.090781 PM

2. ***View the title of inquiries sent by students with ID 4010024, 4010064:***

SELECT Title FROM Inquiry WHERE StudentID IN (4010024,4010064);

TITLE
Studying Math 205
TRAVEL ABROAD

3. ***View the information of inquiry 2 and 3:***

SELECT * FROM Inquiry

WHERE InquiryNo=3 OR InquiryNo=2;

Results Explain Describe Saved SQL History

INQUIRYNO	STUDENTID	TITLE	MESSAGE	INQUIRYDATE	INQUIRYTIME
2	4010024	Studying Math 205	DEAR ALL WE DESIRE TO study math 205 as student, please inform us if there is a chance	03/28/2012	05-DEC-21 05.26.47.090781 PM
3	3910221	Complaint letters	Good Morning, where I could find Complaint letters, please inform us if there is a chance	09/28/2020	05-DEC-21 05.26.47.090781 PM

4. ***View information of inquiries that does not have "TRAVEL ABROAD" as title:***

SELECT * FROM InquiryWHERE NOT Title = 'TRAVEL ABROAD';

INQUIRYNO	STUDENTID	TITLE	MESSAGE	INQUIRYDATE	INQUIRYTIME
2	4010024	Studying Math 205	DEAR ALL WE DESIRE TO study math 205 as student, please inform us if there is a chance	03/28/2012	05-DEC-21 05.26.47.090781 PM
3	3910221	Complaint letters	Good Morning, where I could find Complaint letters, please inform us if there is a chance	09/28/2020	05-DEC-21 05.26.47.090781 PM

5. ***View information of inquiries that its title starts with C:***

SELECT * FROM Inquiry WHERE Title LIKE 'C%';

INQUIRYNO	STUDENTID	TITLE	MESSAGE	INQUIRYDATE	INQUIRYTIME
3	3910221	Complaint letters	Good Morning, where I could find Complaint letters, please inform us if there is a chance	09/28/2020	05-DEC-21 05.26.47.090781 PM

VI. FAQ Table Queries

1. ***View the first question:*** SELECT * FROM FAQ WHERE QuestionNo=1;

QUESTIONNO	QUESTION	ANSWER
1	DEAR ALL WE DESIRE TOO Know how many departments u have?	Hey! tHEY ARE FOUR DEPARTMENTS

1 rows returned in 0.01 seconds Download

2. ***View the first and second question:***

SELECT * FROM FAQ WHERE QuestionNo IN (1, 2);

QUESTIONNO	QUESTION	ANSWER
1	DEAR ALL WE DESIRE TOO Know how many departments u have?	Hey! tHEY ARE FOUR DEPARTMENTS
2	HEY EVERYONE, MAY WE KNOW IF YOU HAVE OTHER UNI BRANCHES?	We apologizize we only have in Madinah

3. ***View the second and third question:***

SELECT * FROM FAQ WHERE QuestionNo=3 OR QuestionNo=2;

QUESTIONNO	QUESTION	ANSWER
2	HEY EVERYONE, MAY WE KNOW IF YOU HAVE OTHER UNI BRANCHES?	We apologizize we only have in Madinah
3	Good Morning, How much should i pay to get accepted in the CS field	Hey, There!, Its prob less than a million ;)

4. ***View all question except the first one:***

SELECT * FROM FAQ WHERE NOT QuestionNo=1;

QUESTIONNO	QUESTION	ANSWER
2	HEY EVERYONE, MAY WE KNOW IF YOU HAVE OTHER UNI BRANCHES?	We apologizize we only have in Madinah
3	Good Morning, How much should i pay to get accepted in the CS field	Hey, There!, Its prob less than a million ;)

5. ***View the answer that starts with W:***

SELECT * FROM FAQ WHERE Answer LIKE 'W%';

QUESTIONNO	QUESTION	ANSWER
2	HEY EVERYONE, MAY WE KNOW IF YOU HAVE OTHER UNI BRANCHES?	We apologizize we only have in Madinah

VII. Reference Table Queries

1. *View the link with Serial No = 1:*

SELECT * FROM ReferenceLinks WHERE ReferenceNo=1;

REFERENCENO	TITLE	REFLINK
1	HISTORY TUTORIAL	https://www.edarabia.com/ar/6-%D8%AD%D9%82%D8%A7%D8%A6%D9%82-%D8%B9%D9%86-%D8%A7%D9%84%D8%A5%D8%B3%D9%83%D9%86%D8%AF%D8%B1-%D8%A7%D9%84%D9%85%D9%82%D8%AF%D9%88%D9%86%D9%8A/

2. *View links with Serial No are 1, 2:*

SELECT * FROM ReferenceLinks WHERE ReferenceNo IN (1, 2);

REFERENCENO	TITLE	REFLINK
1	HISTORY TUTORIAL	https://www.edarabia.com/ar/6-%D8%AD%D9%82%D8%A7%D8%A6%D9%82-%D8%B9%D9%86-%D8%A7%D9%84%D8%A5%D8%B3%D9%83%D9%86%D8%AF%D8%B1-%D8%A7%D9%84%D9%85%D9%82%D8%AF%D9%88%D9%86%D9%8A/
2	CS351 TUTORIAL	https://www.w3schools.com/sql/sql_in.asp

3. *View information about links with Serial No = 2, 3:*

SELECT * FROM ReferenceLinks WHERE ReferenceNo=3 OR ReferenceNo=2;

REFERENCENO	TITLE	REFLINK
2	CS351 TUTORIAL	https://www.w3schools.com/sql/sql_in.asp
3	CS351	https://www.w3schools.com/sql/sql_wildcards.asp

4. *View all links except the link with Serial No = 1:*

SELECT * FROM ReferenceLinks WHERE NOT ReferenceNo=1;

REFERENCENO	TITLE	REFLINK
2	CS351 TUTORIAL	https://www.w3schools.com/sql/sql_in.asp
3	CS351	https://www.w3schools.com/sql/sql_wildcards.asp

2 rows returned in 0.00 seconds Download

5. *View the link where the title starts with H:*

SELECT * FROM ReferenceLinks WHERE Title LIKE 'H%';

REFERENCENO	TITLE	REFLINK
1	HISTORY TUTORIAL	https://www.edarabia.com/ar/6-%D8%AD%D9%82%D8%A7%D8%A6%D9%82-%D8%B9%D9%86-%D8%A7%D9%84%D8%A5%D8%B3%D9%83%D9%86%D8%AF%D8%B1-%D8%A7%D9%84%D9%85%D9%82%D8%AF%D9%88%D9%86%D9%8A/

VIII. Course Table Queries

1. *View all CS courses:*

SELECT * FROM Course WHERE CourseCode LIKE 'CS%';

COURSECODE	COURSENAME
CS 111	Introduction to Computing and Programming
CS 112	Object Oriented Programming
CS 351	Fundamentals of Database Systems
CS 201	Introduction to Discrete Systems
CS 211	Data Structures and Algorithms
CS 221	Fundamentals of Operating Systems
CS 224	Computer Architecture and Organization
CS 321	Operating Systems
CS 464	Software Project Management
CS 332	Computer Networks & Data Communications
CS 262	System Analysis and Design
CS 314	Web Application Development
CS 232	Computer Networks

2. *View Courses that have the word Network:*

SELECT * FROM Course WHERE CourseName LIKE '%Network%';

```
48  SELECT * FROM Course WHERE CourseName LIKE '%Network%';
```

Results Explain Describe Saved SQL History

Results

COURSECODE	COURSENAME
CS 332	Computer Networks & Data Communications
CS 232	Computer Networks
FC 411	Secure Network Design

3 rows returned in 0.00 seconds Download

3. *View the Course Code for the given course name:*

SELECT CourseCode FROM Course WHERE CourseName = 'Introduction to Computing and Programming'

```
48  SELECT CourseCode FROM Course WHERE CourseName = 'Introduction to Computing and Programming';
```

Results Explain Describe Saved SQL History

COURSECODE
CS 111

rows returned in 0.01 seconds Download

4. ***View all CS and FC courses:***

SELECT CourseName FROM Course WHERE CourseCode LIKE 'CS%' OR CourseCode LIKE 'FC%';

Introduction to Computing and Programming
Object Oriented Programming
Fundamentals of Database Systems
Introduction to Discrete Systems
Data Structures and Algorithms
Fundamentals of Operating Systems
Computer Architecture and Organization
Operating Systems
Software Project Management
Computer Networks & Data Communications
Ethics and Professionalism
System Analysis and Design
Web Application Development
Computer Networks
Ethical Hacking

Cyber Security
Operating System Security
Web Security
Computer Forensics and Investigations
Secure Software Design
Defense Mechanisms
Digital Forensic Tools and Techniques
Capstone Project I
Secure Network Design
Professional Elective I
Applied Cryptography
Capstone Project II
Security and Privacy Policies
Professional Elective II
Security Risk Management
Summer Training
31 rows returned in 0.01 seconds Download

5. ***View all courses that does not belong to computer science:***

SELECT * FROM Course WHERE CourseCode NOT LIKE 'CS%' AND CourseCode NOT LIKE 'FC%' AND CourseCode NOT LIKE 'SE%';

- ***Viewing the same result using keyword (intersect):***

SELECT * FROM Course WHERE CourseCode NOT LIKE 'CS%'

INTERSECT

SELECT * FROM Course WHERE CourseCode NOT LIKE 'FC%'

INTERSECT

SELECT * FROM Course WHERE CourseCode NOT LIKE 'SE%';

COURSECODE	
MATH 001	Preparatory Math for Science I
MATH 002	Preparatory Math for Science II
ENGL 000	English for Beginners
ENGL 001	Preparatory English I
ENGL 002	Preparatory English II
ENGL 003	Preparatory English III
ENGL 004	Preparatory English IV
ENGL 005	Preparatory English V
PCS 001	Preparatory Computer Skills
ENGL 101	First Year Composition
PHYS 101	General Physics I
MATH 101	Calculus I
GHAL xx1	General Chemistry I
GHAL xx2	Humanities, Arts, and Languages Elective
ENGL 102	Introduction to Report Writing
PHYS 102	General Physics II

IX. CoursesInProgram Table Queries

1. View all information about a course associated with its program:

SELECT * FROM Course, CoursesInProgram WHERE

CoursesInProgram.CourseCodeInProgram = Course.CourseCode; (Displays 70rows)

```
48  SELECT * FROM Course, CoursesInProgram WHERE CoursesInProgram.CourseCodeInProgram = Course.CourseCode;
```

Results | Explain | Describe | Saved SQL | History

COURSECODE	COURSENAME	PROGRAMCODE	COURSECODEINPROGRAM	PROGRAMLEVEL	CREDITHOURS	PREREQUISITE
ENGL 101	First Year Composition	FC	ENGL 101	1	3	ENGL 005
ENGL 101	First Year Composition	SE	ENGL 101	1	3	ENGL 005
PHYS 101	General Physics I	FC	PHYS 101	1	4	MATH 002
PHYS 101	General Physics I	SE	PHYS 101	1	4	MATH 002
MATH 101	Calculus I	SE	MATH 101	1	4	MATH 002
MATH 101	Calculus I	FC	MATH 101	1	4	MATH 002
CS 111	Introduction to Computing and Programming	FC	CS 111	1	4	PCS 001
CS 111	Introduction to Computing and Programming	SE	CS 111	1	4	PCS 001
ENGL 102	Introduction to Report Writing	SE	ENGL 102	2	3	ENGL 101
ENGL 102	Introduction to Report Writing	FC	ENGL 102	2	3	ENGL 101
PHYS 102	General Physics II	FC	PHYS 102	2	4	PHYS 101
PHYS 102	General Physics II	SE	PHYS 102	2	4	PHYS 101
MATH 102	Calculus II	FC	MATH 102	2	4	MATH 101
MATH 102	Calculus II	SE	MATH 102	2	4	MATH 101
CS 112	Object Oriented Programming	FC	CS 112	2	4	CS 111
CS 112	Object Oriented Programming	SE	CS 112	2	4	CS 111
CS 351	Fundamentals of Database Systems	FC	CS 351	3	4	CS 112

2. View the names and credit hours of courses that are associated to any program without duplicate data:

SELECT DISTINCT CourseName, CreditHours FROM Course, CoursesInProgram

WHERE CoursesInProgram.CourseCodeInProgram = Course.CourseCode ORDER BY

CourseName; (Displays 50rows)

COURSENAME	CREDITHOURS
Applied Cryptography	3
Calculus I	4
Calculus II	4
Calculus III	4
Capstone Project I	3
Capstone Project II	3
Computer Architecture and Organization	3
Computer Forensics and Investigations	3
Computer Networks	4
Computer Networks & Data Communications	4
Cyber Security	3
Data Structures and Algorithms	4
Defense Mechanisms	3
Digital Forensic Tools and Techniques	4
Ethical Hacking	3
Ethics and Professionalism	3
First Year Composition	3

3. View SE courses in level 3 with their credit hours:

SELECT CourseName, CREDITHOURS FROM Course, CoursesInProgram

WHERE CoursesInProgram.CourseCodeInProgram = Course.CourseCode AND

CoursesInProgram.PROGRAMLEVEL = 3 AND CoursesInProgram.ProgramCode = 'SE'

ORDER BY courseName;

```
SELECT CourseName, CREDITHOURS FROM Course, CoursesInProgram
WHERE CoursesInProgram.CourseCodeInProgram = Course.CourseCode
AND CoursesInProgram.PROGRAMLEVEL = 3
AND CoursesInProgram.ProgramCode = 'SE'
ORDER BY courseName;
```

Results Explain Describe Saved SQL History

COURSENAME	CREDITHOURS
Data Structures and Algorithms	4
Fundamentals of Database Systems	4
Introduction to Discrete Systems	3
Technical Writing	3

4 rows returned in 0.01 seconds Download

4. View courses that have math 102 as their prerequisite (no repeated values):

SELECT DISTINCT CourseCode, CourseName, PROGRAMLEVEL FROM Course, CoursesInProgram WHERE CoursesInProgram.CourseCodeInProgram = Course.CourseCode AND PREREQUISITE = 'MATH 102' ORDER BY CourseName;

COURSECODE	COURSENAME	PROGRAMLEVEL
MATH 202	Calculus III	4
CS 201	Introduction to Discrete Systems	3
MATH 204	Linear Algebra	5
STAT 232	Probability and Statistics	4

4 rows returned in 0.01 seconds Download

5. View courses that have no prerequisite:

SELECT CourseCode, CourseName FROM Course, CoursesInProgram WHERE CoursesInProgram.CourseCodeInProgram = Course.CourseCode AND PREREQUISITE IS NULL;

```
47  SELECT CourseCode, CourseName FROM Course, CoursesInProgram WHERE CoursesInProgram.CourseCodeInProgram = Course.CourseCode AND PREREQUISITE IS NULL;
```

Results Explain Describe Saved SQL History

COURSECODE	COURSENAME
CS 221	Fundamentals of Operating Systems
FC 372	Ethics and Professionalism
FC 372	Ethics and Professionalism
SE xxx1	Professional Elective I
SE xxx2	Professional Elective II
SE xxx3	Professional Elective III
SE xxx4	Professional Elective IV
FC xxx1	Professional Elective I
FC xxx2	Professional Elective II
SE 394	Summer Training
FC 394	Summer Training

11 rows returned in 0.01 seconds Download

X. studentGoal Table Queries

1. View the name of a student specifying her goal:

SELECT FIRSTNAME,LASTNAME FROM Student, StudentGoal WHERE Student.StudentID = StudentGoal.StudentID AND StudentGoal.STUDENTGOAL = 4;

```
47  SELECT FIRSTNAME,LASTNAME FROM Student, StudentGoal WHERE Student.StudentID = StudentGoal.StudentID AND StudentGoal.STUDENTGOAL = 4;
```

Results Explain Describe Saved SQL History

FIRSTNAME	LASTNAME
Nourah	Saud

1 rows returned in 0.02 seconds Download

2. ***View the goals and ID of students who has major in SE program:***

SELECT StudentGoal.STUDENTID, STUDENTGOAL FROM StudentGoal, Program, Student WHERE Student.StudentID = StudentGoal.StudentID AND Program.ProgramCode = 'SE';

```
47  SELECT StudentGoal.STUDENTID, STUDENTGOAL FROM StudentGoal, Program, Student WHERE Student.StudentID = StudentGoal.StudentID AND Program.ProgramCode = 'SE';
```

Results | Explain | Describe | Saved SQL | History

STUDENTID	STUDENTGOAL
4010064	3.8
3910221	2.5
3940022	4

3 rows returned in 0.01 seconds Download

3. ***View the name and major student who has the goal 2.5:***

SELECT FIRSTNAME, MAJORPROGRAM FROM Student, StudentGoal WHERE Student.StudentID = StudentGoal.StudentID AND StudentGoal.STUDENTGOAL = 2.5;

```
47  SELECT FIRSTNAME, MAJORPROGRAM FROM Student, StudentGoal WHERE Student.StudentID = StudentGoal.StudentID AND StudentGoal.STUDENTGOAL = 2.5;
```

Results | Explain | Describe | Saved SQL | History

FIRSTNAME	MAJORPROGRAM
YASSER	SE

1 rows returned in 0.02 seconds Download

4. ***View the courses and grades for the goal of the student with ID 4010064:***

SELECT GOALEDCOURSES, GRADESTOGET FROM COURSEGOALBYSTUDENT WHERE COURSEGOALBYSTUDENT.StudentID = 4010064;

```
47  SELECT GOALEDCOURSES, GRADESTOGET FROM COURSEGOALBYSTUDENT WHERE COURSEGOALBYSTUDENT.StudentID = 4010064;
```

Results | Explain | Describe | Saved SQL | History

GOALEDCOURSES	GRADESTOGET
SE 262	B+
STAT 232	A
MATH 202	A+
CS 224	A

4 rows returned in 0.01 seconds Download

5. ***View the courses with goals along with their credit hours (without repeated data):***

SELECT DISTINCT GOALEDCOURSES, CreditHours FROM COURSEGOALBYSTUDENT, CoursesInProgram WHERE CoursesInProgram.CourseCodeInProgram = courseGoalByStudent.GoaledCourses;

```
47  SELECT DISTINCT GOALEDCOURSES, CreditHours FROM COURSEGOALBYSTUDENT, CoursesInProgram WHERE CoursesInProgram.CourseCodeInProgram = courseGoalByStudent.GoaledCourses;
```

Results | Explain | Describe | Saved SQL | History

GOALEDCOURSES	CREDITHOURS
CS 111	4
SE 262	3
CS 201	3
ENGL 101	3
STAT 232	3
MATH 202	4
CS 211	4
CS 224	3
CS 351	4
MATH 101	4

0 rows returned in 0.02 seconds Download

XI. CourseGoalByStudent Table Queries

1. ***View student ID with the grade which he/she should get in CS 351 to achieve the goal:***

SELECT StudentID, GradesToGet FROM CourseGoalByStudent WHERE GoaledCourses = 'CS 351';

```
47  SELECT StudentID, GradesToGet FROM CourseGoalByStudent WHERE GoaledCourses = 'CS 351';
```

Results | Explain | Describe | Saved SQL | History

STUDENTID	GRADESTOGET
3910221	B

1 rows returned in 0.00 seconds Download

2. View the courses student (3910221) has goal for:

SELECT GoaledCourses FROM CourseGoalByStudent WHERE StudentID = 3910221;

```
47  SELECT GoaledCourses FROM CourseGoalByStudent WHERE StudentID = 3910221;
```

Results | Explain | Describe | Saved SQL | History

GOALEDCOURSES
CS 201
CS 211
CS 351
MATH 202

4 rows returned in 0.01 seconds Download

3. View the credit for all courses that have a goal:

SELECT DISTINCT GoaledCourses, CreditHours FROM CourseGoalByStudent,

CoursesInProgram WHERE CourseGoalByStudent.GoaledCourses =

CoursesInProgram.CourseCodeInProgram;

```
47  SELECT DISTINCT GoaledCourses, CreditHours FROM CourseGoalByStudent, CoursesInProgram
48      WHERE CourseGoalByStudent.GoaledCourses = CoursesInProgram.CourseCodeInProgram;
```

Results | Explain | Describe | Saved SQL | History

GOALEDCOURSES	CREDITHOURS
CS 111	4
SE 262	3
CS 201	3
ENGL 101	3
STAT 232	3
MATH 202	4
CS 211	4
CS 224	3
CS 351	4
MATH 101	4

10 rows returned in 0.02 seconds Download

4. ***View the Name of the student who has a goal for CS 111:***

SELECT FirstName FROM Student, CourseGoalByStudent WHERE Student.studentID = CourseGoalByStudent.StudentID AND CourseGoalByStudent.GoaledCourses = 'CS 111';

```
47  SELECT FirstName FROM Student, CourseGoalByStudent WHERE Student.studentID = CourseGoalByStudent.StudentID AND CourseGoalByStudent.GoaledCourses =  'CS 111';
```

Results | Explain | Describe | Saved SQL | History

FIRSTNAME
Nourah

1 rows returned in 0.01 seconds Download

5. ***View the GPA goal for the student who has a grade goal for CS 111:***

SELECT StudentGoal FROM StudentGoal, CourseGoalByStudent WHERE StudentGoal.StudentID = CourseGoalByStudent.StudentID AND CourseGoalByStudent.GoaledCourses = 'CS 111' ;

```
47  SELECT StudentGoal FROM StudentGoal, CourseGoalByStudent WHERE StudentGoal.StudentID = CourseGoalByStudent.StudentID AND CourseGoalByStudent.GoaledCourses = 'CS 111' ;
```

Results | Explain | Describe | Saved SQL | History

STUDENTGOAL
4

1 rows returned in 0.01 seconds Download

6. ***View the goaled courses with their credit hours for the student who has 3.8 as goal:***

SELECT GOALEDCOURSES, CreditHours FROM StudentGoal, CourseGoalByStudent, CoursesInProgram WHERE StudentGoal.StudentID = CourseGoalByStudent.StudentID AND StudentGoal.StudentGoal = 3.8 AND CourseGoalByStudent.GoaledCourses = CoursesInProgram.CourseCodeInProgram;

Results | Explain | Describe | Saved SQL | History

GOALEDCOURSES	CREDITHOURS
STAT 232	3
SE 262	3
CS 224	3
MATH 202	4
STAT 232	3

5 rows returned in 0.00 seconds Download

SQL aggregate functions for all tables:

Using Functions For Program Table:

1. ***Using SUM function to calculate how much credit hours secondary courses has in all programs:***

```
SELECT SUM(SecondaryCredit) FROM Program;
```

Results | Explain | Describe | Saved SQL | History

SUM(SECONDARYCREDIT)
99

1 rows returned in 0.01 seconds Download

2. ***Using SUM function to calculate how much credit hours are in each program:***

```
SELECT ProgramCode, SUM(MainCredit + ElectiveCredit + SecondaryCredit) AS Credit FROM Program GROUP BY ProgramCode;
```

Results | Explain | Describe | Saved SQL | History

PROGRAMCODE	CREDIT
NULL	0
SE	296
FC	290

3 rows returned in 0.01 seconds Download

3. ***Using MAX function to find the maximum credit of all programs:***

```
SELECT MAX(MainCredit + ElectiveCredit + SecondaryCredit) FROM Program;
```

Results | Explain | Describe | Saved SQL | History

MAX(MAINCREDIT+ELECTIVECREDIT+SECONDARYCREDIT)
296

1 rows returned in 0.01 seconds Download

Using Functions For Student Table:

1. ***Using count function to know how many students are members on the website:***

```
SELECT COUNT(StudentId) FROM STUDENT;
```

Results | Explain | Describe | Saved SQL | History

COUNT(STUDENTID)
13

1 rows returned in 0.00 seconds Download

2. ***Using count function to count how many students majored in FC program are on the website:***

```
48    SELECT COUNT(StudentId) FROM STUDENT WHERE MajorProgram = 'FC';
```

Results | Explain | Describe | Saved SQL | History

COUNT(STUDENTID)
5

1 rows returned in 0.01 seconds Download

3. ***Using min, max functions to display youngest and oldest student:***

```
48    SELECT MIN(BIRTHDATE), MAX(BIRTHDATE) FROM STUDENT;
```

Results | Explain | Describe | Saved SQL | History

MIN(BIRTHDATE)	MAX(BIRTHDATE)
12/20/1995	05/20/2002

1 rows returned in 0.01 seconds Download

Using Functions For Course Table:

1. ***Counting how many math courses:***

```
48    SELECT COUNT(CourseCode) FROM Course WHERE CourseCode LIKE 'MATH%';
```

Results | Explain | Describe | Saved SQL | History

COUNT(COURSECODE)
6

1 rows returned in 0.01 seconds Download

2. ***Counting how many Math & English courses:***

```
47
48    SELECT COUNT(*) FROM Course WHERE CourseCode LIKE 'MATH%' OR CourseCode LIKE 'ENGL%';
```

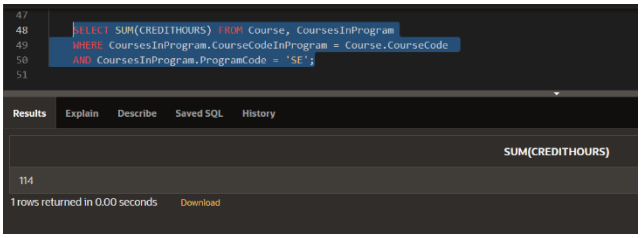
Results | Explain | Describe | Saved SQL | History

COUNT(*)
15

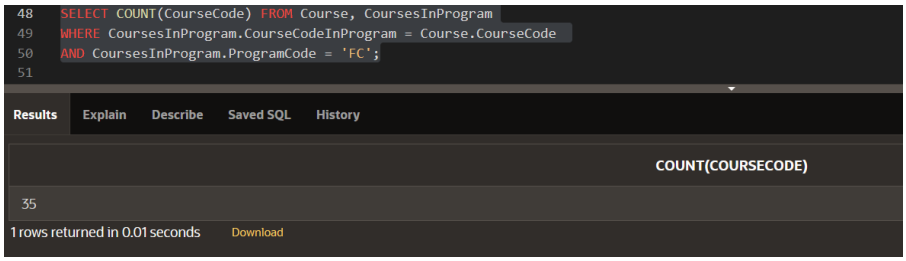
1 rows returned in 0.01 seconds Download

Using Functions For Courses In Program Table:

1. *Finding the sum of credit hours for courses in SE program*

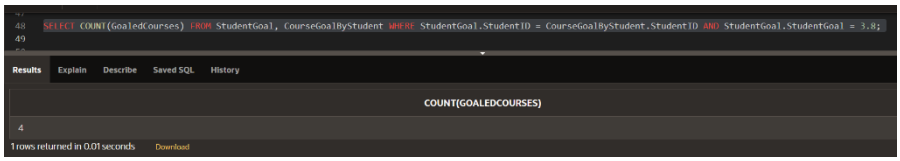


2. *Finding how many courses in FC program*

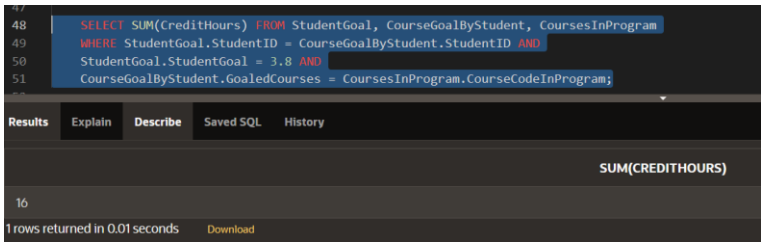


Using Functions For Course Goal By Student Table:

1. *Using COUNT to know how many courses this student has goal for*



2. *Using SUM to display the total of credit hours a student has a goal for*



Using Functions For Student Goal Table:

- ***Finding AVG, MAX, MIN of students' goals:***

```
SELECT AVG(StudentGoal) FROM STUDENTGOAL;
```

Results | Explain | Describe | Saved SQL | History

3.433333333333333333333333333333333333

```
SELECT MAX(StudentGoal) FROM STUDENTGOAL;
```

Results | Explain | Describe | Saved SQL | History

4

1 rows returned in 0.00 seconds Download

```
SELECT MIN(StudentGoal) FROM STUDENTGOAL;
```

Results | Explain | Describe | Saved SQL | History

2.5

1 rows returned in 0.00 seconds Download

Using Functions For Student Completed Into Table:

- ***Counting how many courses this student completed, and the sum of his marks and the minimum mark he got:***

```
SELECT COUNT(COMPLETEDCOURSECODE) FROM STUDENTCOMPLETED WHERE STUDENTID = 3910022;
```

Results | Explain | Describe | Saved SQL | History

COUNT(COMPLETEDCOURSECODE)
9

1 rows returned in 0.01 seconds Download

```
SELECT SUM(COURSEMARK) FROM STUDENTCOMPLETED WHERE STUDENTID = 3910022;
```

Results | Explain | Describe | Saved SQL | History

SUM(COURSEMARK)
829

1 rows returned in 0.02 seconds Download

```
SELECT MIN(COURSEMARK) FROM STUDENTCOMPLETED WHERE STUDENTID = 3910022;
```

Results | Explain | Describe | Saved SQL | History

80

1 rows returned in 0.01 seconds Download

Using Functions For Courses Student Enrolled Into Table:

1. ***Counting how many courses this student(3910221) is currently taking***

```
SELECT COUNT(CURRENTCOURSES) FROM COURSESSTUDENTENROLLEDINTO WHERE STUDENTID = 3910221;
```

Results | Explain | Describe | Saved SQL | History

COUNT(CURRENTCOURSES)
4

1 rows returned in 0.01 seconds Download

2. ***Counting how many students enrolled in this course (CS 201)***

```
SELECT COUNT(StudentID) FROM CoursesSTUDENTENROLLEDINTO WHERE CurrentCourses = 'CS 201';
```

Results | Explain | Describe | Saved SQL | History

COUNT(STUDENTID)
3

1 rows returned in 0.01 seconds Download

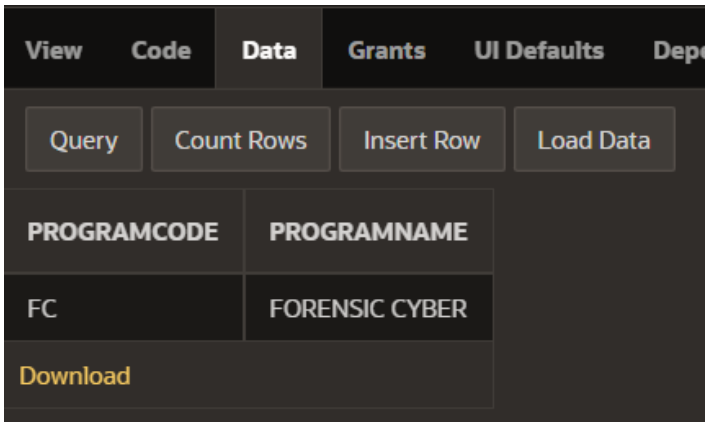
XII.Creating inner Join Code

select student.studentid, program.programcode from student inner join program on student.MAJORPROGRAM = program.programcode;

STUDENTID	PROGRAMCODE
4010064	SE
4010024	SE
4010545	FC
3910022	FC
3710022	SE
3622000	FC
3940021	SE

XIII. Creating Different Views

```
CREATE VIEW ProgramtView AS
SELECT ProgramCode,
ProgramName
FROM Program
 WHERE ProgramCode = 'FC';
```



```
CREATE VIEW
CoursesStudentEnrolledIntoView AS
SELECT StudentID, CurrentCourses
FROM CoursesStudentEnrolledInto
WHERE CurrentCourses= 'CS 201';
```