|  |
| --- |
| ***Ahsanullah University of Science & Technology***  **Description: logo.jpg**Department of Computer Science & Engineering |
| University Management System |
| Database Lab  (CSE 3104) |

**Submitted To**

**Mr. Mohammad Moinul Hoque**

**Mr. Mir Tafseer Nayeem**

**Prepared By:**

**Md. Tanvir Rouf Shawon  
ID: 16.01.04.138**

**Jeba Maliha  
ID: 16.01.04.143**

**Project Report**

**Name of the Project:**

**University Management System**

**Project Objectives**

* The Main target of this project is to make a organized database of a university.
* To insert new student or Instructor data in the database and retrieve them for various purpose.
* Another objective of the system is to maintain the university information like student’s result, admission etc.
* It can also maintain the Instructor’s information for administrative purpose.

**Features of the project**

* Show the Instructor’s name according to their department.
* Show the Course title, course credit and department name for all department.
* Show the existing departments of the university.
* Show all the courses of CSE department
* Show all the courses of EEE department
* Show all the 3 credit courses of CSE department
* Login of normal user
* Changing the password of a registered user.
* Show the total credit of CSE department
* Show the total credit of all department.
* Show the result of student.
* Show the result of a student in a particular semester
* Show the semester wise Courses they have taken
* Show the course Teacher students have in their course
* Student’s login
* Student’s password change
* Show all the information of Students
* Show all the information of Department
* Show all the information of Instructors.
* Show all the information of Instructors of EEE department.
* Show all the information of Courses
* Show all the information of Sections
* Show all the information of Enrollment.
* Show the name and salary of the Instructors whose salary is 50000 or greater.
* Show the name and salary of the Instructors whose salary is 50000 or greater and department is CSE.
* Show all the information of Courses which are taken by a particular instructor.
* Show the name of the Instructor who are currently in service.
* Show the name of the Instructor who have left the university.
* Show the serving time and name of a particular Instructor who have left.
* Increase Instructor's salary by 10 percent whose salary is lower or equal to 30000
* Show the courses taken by a student.
* Show the courses of CSE department.
* Show the highest Salary among the Instructors
* Show the lowest Salary among the Instructors
* Show the courses that were offered in Spring 16 but not Fall 15.
* Show the department wise Maximum Salary
* Show the total credit hour of each department
* Show the section wise student in Spring 16 semester
* Show the students with name starting with B
* Show the student whose address is Sylhet
* Show the students who have age between 18 and 22
* Show the student who use Robi Sim
* Show the student whose phone number ends with 2
* Show the student who have got A+
* Show the instructor who take more than 2 courses
* Admin’s login in the database.
* Show the number of registered User
* Show the number of registered Student
* Show all the information about the registered User
* Show the registered User
* Show the student who attend class at a particular room

**Types of Users**

**# Normal Users**

In this university management system a normal user can on ly see some limited information like Instructor’s list, department list, Courses according to the departments etc.

**# Students (Private User)**

Students are the private user of this system. They can know their result of different semester, their marks, their course teacher etc. We can say that they are the users who only can see their own information and all the information a general user can get.

**# ADMIN**

Admin is the controller of the university management system. An administrative person will act as an admin. He can perform a lot of action on the database. Everything is under control of him. He can enter new Instructor, new Students, New sections and what not.

**Feature grouping according to the Users**

**# Normal User**

* Show the Instructor’s name according to their department.
* Show the Course title ,course credit and department name for all department.
* Show the existing departments of the university.
* Show all the courses of CSE department
* Show all the courses of EEE department
* Show all the 3 credit courses of CSE department
* Login of normal user
* Changing the password of a registered user.
* Show the total credit of CSE department
* Show the total credit of all department.

**# Students (Private Users)**

* Show the result of student.
* Show the result of a student in a particular semester
* Show the semester wise Courses they have taken
* Show the course Teacher students have in their course
* Student’s login
* Student’s password change

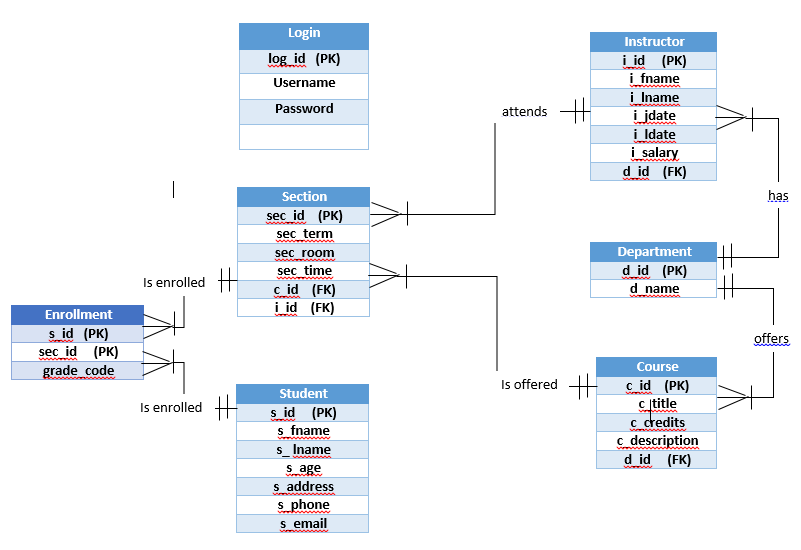
**# Admin**

* Show all the information of Students
* Show all the information of Department
* Show all the information of Instructors.
* Show all the information of Instructors of EEE department.
* Show all the information of Courses
* Show all the information of Sections
* Show all the information of Enrollment.
* Show the name and salary of the Instructors whose salary is 50000 or greater.
* Show the name and salary of the Instructors whose salary is 50000 or greater and department is CSE.
* Show all the information of Courses which are taken by a particular instructor.
* Show the name of the Instructor who are currently in service.
* Show the name of the Instructor who have left the university.
* Show the serving time and name of a particular Instructor who have left.
* Increase Instructor's salary by 10 percent whose salary is lower or equal to 30000
* Show the courses taken by a student.
* Show the courses of CSE department.
* Show the highest Salary among the Instructors
* Show the lowest Salary among the Instructors
* Show the courses that were offered in Spring 16 but not Fall 15.
* Show the department wise Maximum Salary
* Show the total credit hour of each department
* Show the section wise student in Spring 16 semester
* Show the students with name starting with B
* Show the student whose address is Sylhet
* Show the students who have age between 18 and 22
* Show the student who use Robi Sim
* Show the student whose phone number ends with 2
* Show the student who have got A+
* Show the instructor who take more than 2 courses
* Admin’s login in the database.
* Show the number of registered User
* Show the number of registered Student
* Show all the information about the registered User
* Show the registered User
* Show the student who attend class at a particular room.

**Names of the Entities with primary key**

|  |  |
| --- | --- |
| **Table Name** | **Primary Key** |
| Login | log\_id |
| Instructor | i\_id |
| Student | s\_id |
| Section | sec\_id |
| Department | d\_id |
| Course | c\_id |
| Enrollment | s\_id, c\_id |

**Entity Relationship Diagram (ERD)**



**Relational Model**

CREATE TABLE Student (

s\_id int IDENTITY(10001,1) PRIMARY KEY,

s\_fname varchar(20) NOT NULL,

s\_lname varchar(20) NOT NULL,

s\_age int NOT NULL,

s\_address varchar(100) NOT NULL,

s\_phone varchar(20) NOT NULL,

s\_email varchar(50) NOT NULL,

);

CREATE TABLE Department (

d\_id int IDENTITY(20001,1) PRIMARY KEY,

d\_name varchar(40) NOT NULL UNIQUE

);

CREATE TABLE Instructor (

i\_id int IDENTITY(30001,1) PRIMARY KEY,

i\_fname varchar(20) NOT NULL,

i\_lname varchar(20) NOT NULL,

i\_jdate date NOT NULL,

i\_ldate date NULL,

i\_salary int NOT NULL,

d\_id int NOT NULL REFERENCES Department(d\_id)

);

CREATE TABLE Course (

c\_id int IDENTITY(40001,1) PRIMARY KEY,

c\_title varchar(100) NOT NULL,

c\_credits tinyint NOT NULL,

d\_id int NOT NULL REFERENCES Department(d\_id),

c\_description varchar(255) NOT NULL

);

CREATE TABLE Section (

sec\_id int IDENTITY(50001,1) PRIMARY KEY,

sec\_term varchar(8) NOT NULL,

sec\_room varchar(4),

sec\_time varchar(10),

c\_id int NOT NULL REFERENCES Course(c\_id),

i\_id int REFERENCES Instructor(i\_id)

);

CREATE TABLE Enrollment (

s\_id int REFERENCES Student(s\_id),

sec\_id int REFERENCES Section(sec\_id),

grade\_code varchar(2),

PRIMARY KEY (s\_id, sec\_id)

);

CREATE TABLE Login (

log\_id int IDENTITY(60001,1) PRIMARY KEY,

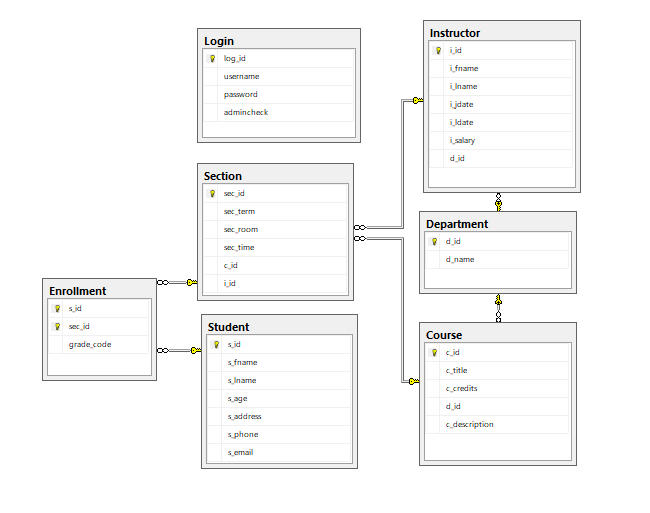
username varchar(50),

password varchar(100),

admincheck int not null

);

**Database Diagram**



**SQL Queries Grouped Under Different Types Of Users:**

Insert Into Login

Values ('Jamal','12345',1),

('Kamal','123',0),

('Chapel','123',0),

('Rahim','1234',2);

Insert Into Student

Values ('Jack', 'Johnson',21,'Dhaka', '01723423423', 'jack@gmail.com'),

('Dolly', 'Denver',22,'Dhaka', '01712345678', 'dolly@gmail.com'),

('Helmut', 'Ziegler',21,'Comilla','01898765432', 'helmut@yahoo.com'),

('Robert', 'Thompson',24,'Khulna','01808975643', 'robert@gmail.com'),

('Jeffrey', 'Petersen',23,'Sylhet', '01773439315', 'petersen@yahoo.com'),

('Jack', 'Pirate',22,'Chittagong','01712321124', 'pirate@gmail.com'),

('Barb', 'Wire',21, 'Comilla','01709876789','barb@yahoo.com'),

('Roberta', 'Strong',24, 'Sylhet','01827645238', 'strong@gmail.com'),

('Heather', 'Black',22,'Dhaka', '01921674523','black@yahoo.com'),

('Erik', 'Bjornsen',23,'Khulna', '01990675634', 'erik@yahoo.com');

Insert Into Department

Values ('CSE'),

('CIVIL'),

('EEE'),

('MECHA'),

('IPE'),

('BBA'),

('TEX');

Insert Into Instructor

Values ('Bill', 'Smith', '1995-12-21','2014-07-02',50000,20001),

('James', 'Peterson', '1997-03-12','2018-06-12',35000,20001),

('Adam', 'Thompson', '1998-04-15',null,60000,20002),

('Janetta', 'Oakley', '2001-02-13',null,50000,20003),

('Robin', 'Dexter', '2003-05-23',null,45000,20002),

('Annie', 'Jackson', '2004-06-02',null,30000,20004),

('Philip', 'Petrovsky', '2006-02-27',null,50000,20005),

('Anastasia', 'Scott', '2013-03-12',null,60000,20001);

Insert Into Course

Values ('EEE201',3, 20003, 'Microeconomics'),

('CIVIL301',3, 20002, 'Building Materials'),

('EEE302',3,20003, 'Power Electronics'),

('EEE303',4,20003 , 'Circuit Setup'),

('CSE201', 4,20001, 'Algorithms I'),

('CSE202', 3,20001, 'Data Stucture'),

('MECHA203',3,20004, 'Fluid Mechanics'),

('MECHA302',2,20004, 'Statistical Mathematics');

Insert Into Section

Values ('Spring16','7A05','10.30', 40001,30005),

('Fall16','7A05','01.00',40002,30004),

('Fall17','6A03', '12.30',40007,30007),

('Spring18','6A02', '12.30',40008,30007),

('Fall17','5A04', '11.00',40006,30002),

('Spring16','5A04', '11.00',40003,30005),

('Spring18','5A02', '11.00',40005,30003);

Insert Into Enrollment

Values (10001,50001,'A+'),

(10001,50006,'A'),

(10002,50001,'B-'),

(10002,50006,'A'),

(10003,50003,'A+'),

(10003,50005,'A+'),

(10004,50003,'B+'),

(10004,50005,'B'),

(10005,50002,'A-'),

(10005,50004,'B-'),

(10006,50004,'C'),

(10006,50007,'F'),

(10008,50003,'D'),

(10009,50003,'F'),

(10007,50001,'A-');

**# Normal User**

--> 1

SELECT i\_fname +' '+ i\_lname as Name\_of\_the\_Faculties,d\_name as Department\_Name

from Instructor I,Department D where I.d\_id=D.d\_id

-->2

SELECT c\_title as Course\_Title,c\_description as Course\_Name, c\_credits as Course\_credit, d\_name as Department\_Name

from Course C,Department D where C.d\_id=D.d\_id

-->3

SELECT d\_name as Department\_Name FROM Department

-->4

SELECT c\_title as Course\_Name,c\_credits as Credit

FROM Course C,Department D

where C.d\_id=D.d\_id and d\_name='CSE'

-->5

SELECT c\_title as Course\_Name,c\_credits as Credit

FROM Course C,Department D

where C.d\_id=D.d\_id and d\_name='EEE'

-->6

SELECT c\_title as Course\_Name,c\_credits as Credit

FROM Course C,Department D

where C.d\_id=D.d\_id and d\_name='CSE' and c\_credits=3

-->7

Select username from Login where username='Kamal' and password ='123' and admincheck=0

-->8

Update Login set password='1234' where username='Chapel'

-->9

Select sum(C.c\_credits) as Total\_Credit\_Of\_CSE from Course C,Department D

where C.d\_id=D.d\_id and D.d\_name='CSE'

-->10

Select D.d\_name as Departpent\_Name, sum(C.c\_credits) as Total\_Credit

from Course C,Department D

where C.d\_id=D.d\_id group by D.d\_name

**# Students (Private Users)**

-->1 Result of a student

SELECT st.s\_id as Student\_ID, c\_description as Course\_Name,grade\_code as Grade

FROM Enrollment E,Section S,Course C,Student St

where E.sec\_id=S.sec\_id and S.c\_id=C.c\_id and E.s\_id=St.s\_id and st.s\_id = 10001

-->2 Result of a student in a particuler semester

SELECT st.s\_id as Student\_ID, c\_description as Course\_Name,grade\_code as Grade

FROM Enrollment E,Section S,Course C,Student St

where E.sec\_id=S.sec\_id and S.c\_id=C.c\_id and E.s\_id=St.s\_id and st.s\_id = 10007 and S.sec\_term='Spring16'

-->3 Semester wise Courses they have taken

SELECT c\_description as Course\_Name,S.sec\_term as Semester

FROM Enrollment E,Section S,Course C,Student St

where E.sec\_id=S.sec\_id and S.c\_id=C.c\_id and E.s\_id=St.s\_id and st.s\_id = 10001

-->4 Course Teacher they have in their course

SELECT c\_description as Course\_Name,I.i\_fname as Instructor

FROM Enrollment E,Section S,Course C,Student St,Instructor I,Department D

where E.sec\_id=S.sec\_id and S.c\_id=C.c\_id and E.s\_id=St.s\_id and I.d\_id=D.d\_id and D.d\_id=C.d\_id and st.s\_id = 10001

-->5 Student Login

Select username from Login where username='Jamal' and password ='12345' and admincheck=1

-->6

Update Login set password='1234' where username='Chapel'

**# ADMIN**

-->1

SELECT \* from Student

-->2

SELECT \* from Department

-->3

SELECT \* from Instructor

-->5

SELECT \* from Instructor I,Department D where I.d\_id=D.d\_id and D.d\_name='EEE'

-->6

SELECT \* from Course

-->7

SELECT \* from Section

-->8

SELECT \* from Enrollment

-->9

SELECT i\_fname+' '+ i\_lname AS Name\_Of\_The\_Instructor,i\_salary AS Salary FROM Instructor WHERE i\_salary>=50000

-->10

SELECT i\_fname+' '+ i\_lname AS Name\_Of\_The\_Instructor,i\_salary AS Salary,d\_name As Department\_name

FROM Instructor I,Department D

WHERE i\_salary>=50000 and I.d\_id=D.d\_id and d\_name='CSE'

-->11

SELECT c\_description as Course\_Name FROM Course C,Section S,Instructor I

where S.c\_id=C.c\_id and S.i\_id=I.i\_id and i\_fname='Bill'

-->12 Current Instructor

SELECT i\_fname+' '+ i\_lname AS Name\_Of\_The\_Instructor FROM Instructor where i\_ldate is NULL

-->13 Instructors who have left

SELECT i\_fname+' '+ i\_lname AS Name\_Of\_The\_Instructor FROM Instructor where i\_ldate is not NULL

-->14 Serving Time of a teacher

SELECT DATEDIFF(YEAR,i\_jdate,i\_ldate) FROM Instructor where i\_fname='Bill'

-->15 Increse Instructor's salary by 10 percent whose salary is lower or equal to 30000

UPDATE Instructor set i\_salary=i\_salary+i\_salary\*.10 where i\_salary<=30000

-->16 Course That are taken by a Student

SELECT c\_description as Course\_Name FROM Enrollment E,Section S,Course C,Student St

where E.sec\_id=S.sec\_id and S.c\_id=C.c\_id and E.s\_id=St.s\_id and st.s\_fname='Jack'

-->17 Course of CSE

SELECT c\_description FROM Course where c\_title like 'CSE%'

-->18 Highest Salary among the Instructors

SELECT max(i\_salary) from Instructor

-->19 Lowest Salary among the Instructors

SELECT min(i\_salary) from Instructor

-->20 Courses that were offered in Spring 16 but not Fall 15\

SELECT c\_description as Course\_Name FROM Section S,Course C

where S.c\_id=C.c\_id and S.sec\_term='Spring16' and S.sec\_term !='Fall15'

-->21 Department wise Maximum Salary

SELECT D.d\_name as Department, max(i\_salary) from Instructor I,Department D

where I.d\_id=D.d\_id Group by D.d\_name

-->22 Total credit hour of each department

SELECT D.d\_name as Department, sum(c\_credits) from Course C,Department D

where C.d\_id=D.d\_id Group by D.d\_name

-->23 Section wise student in Spring 16 semister

SELECT S.sec\_id,count(St.s\_id) from Student St,Enrollment E, Section S

where S.sec\_term='Spring16' and St.s\_id=E.s\_id and E.sec\_id=S.sec\_id

group by S.sec\_id

-->24 Student with name starting with B

Select s\_fname+' '+s\_lname as Name from Student where s\_fname like 'B%'

-->25 Student whose address is Sylhet

Select s\_fname+' '+s\_lname as Name from Student where s\_address='sylhet'

-->26 Student who have age between 18 and 22

Select s\_fname+' '+s\_lname as Name from Student where s\_age between 18 and 22

-->27 Student who use Robi Sim

Select s\_fname+' '+s\_lname as Name from Student where s\_phone like '018%'

-->26 Student whose phone number ends with 2

Select s\_fname+' '+s\_lname as Name from Student where s\_phone like '%2'

-->27 Student who have got A+

Select distinct s\_fname+' '+s\_lname as Name from Student St, Enrollment E, Section S

where St.s\_id=E.s\_id and E.sec\_id=S.sec\_id and E.grade\_code='A+'

-->28 Instructor who take more than 2 courses

SELECT I.i\_fname as Instructor\_Name, count(S.c\_id) from Instructor I,Section S

where I.i\_id=S.i\_id Group by I.i\_fname having count(S.c\_id)>=2

-->29 Admin Login

Select username from Login where username='Rahim' and password ='1234' and admincheck=2

-->30 Number of registered User

Select count(log\_id) from Login where admincheck=0 or admincheck=1

-->31 Number of registered Student

Select count(log\_id) from Login where admincheck=1

--> 32 Know all the information about the registered User

Select \* from Login

-->33 Show the registered User

Select username from Login where admincheck=1 or admincheck=0

-->34 Student who attend class at a particular room

select s\_fname from Student where s\_id in

(select s\_id from Enrollment where sec\_id in

(Select sec\_id from Section where sec\_room='7A05'))

**Project Limitations**

* The system is not perfect for an university. It needs some improvement to use perfectly.
* The employee table is not added in this database.
* Instead of date of birth age is given in Student table. So we have to increase this column by one every year.
* Result is only given in grade system but marks should be included for better understanding.

**Conclusion & Future Work**

**Future Work :**

We can add some table like employee, hall etc. in this database and implement it completely. This project has no User Interface. So we can implement a User interface with Java or PHP. We can use it as a website database for an university also.

**Conclusion:**

As this project is a MSSql Database project. It can be used in different types of User interfaced project like Java. This is a portable project as it is only created by some sql code. This project is also efficient as we have attached all the query possible for an university management system. We have used a low amount of resources as we have not used anything without SQL server management studio. This is also a user friendly system for all types of user. At the end we can say that it is a complete system for a university.