الجامعة الإسلامية العالمية شيتاغونغ
International Islamic University Chittagong

Department of Computer Science and Engineering

Database System Sessional

CSE 2408

Submitted by :

Abdullah Al Tajbir	Md. Shahin Uddin
Matric No : C161054	Matric No : C161050
Section :4BM	

Submitted to :

Mohammad Safiullah
Assistant Professor
International Islamic University Chittagong

Mr. A.F.M. Soyeb Chowdhury
Lecturer
International Islamic University Chittagong

REMARK

Project on University Department Management System

Project Overview:

The 'University Management System' is designed to keep track of the information of the, Students, Teachers, Employees, Course, Fees .

Project Description:

The management system allows authorized members to access the record of academically registered students & teachers. It can be used in various educational institutes across the globe and simplifies working of institutes.

Purpose Of The Project:

The system maintains records of students & teachers, the courses and modules on which they are registered and the outcome of their studies & teaches, students fess and teachers & employee salary. The application is distributed application because the data is stored centrally by the university but accessed by academic staff and administrators across the university. The real system would be service oriented and would expose some of its functionality as a set of services which could be consumed by other applications.

Problems In The Existing System:

The future of the existing system has to be seen as one of continuous change, where the information that is stored is of increasing complexity and quantity. The existing system is required to provide a useable and well managed interface for student, teacher, academic and administrator users to view and manipulate the data for which it is responsible. For each it must allow the rapid formulation and resolution of queries related to the student, teacher & Employee information. There is also a requirement for the system to interact with other information sources as required, both as an information source and as a consumer of related information during the resolution of queries.

Solutions of This Problems:

- The development of this new system contains the following activities, which try to automate the entire process keeping in the view of database integration approach.
- User Friendliness is provided in the application with various controls provided by system Rich User Interface.
- The system makes the overall project management much easier and flexible.
- It can be accessed over the Intranet.
- The student information can be stored in centralized database which can be maintained by the system.
- This can give the good security for user information because data is not in client machine.
- Authentication is provided for this application only registered students can access.

There is no risk of data management at any level while the project development is under process.

Database Schema:

Department

DepartmentId	*Department*

Student

StudentName	*StudentId*	*Department.*	*Semester*	*Address*	*ContactNo*	*Advisor*	*Email*

Teacher

TeacherName	*TeacherId*	*Department*	*ContactNo*	*Address*	*Designation*	*Salary*

Course

CourseId	*CourseName*	*CreditHour*

Employee

EmployeeName	*EmployeeId*	*Department*	*ContactNo*	*Address*	*Salary*

Fees

StudentId	*ReceiptNo*	*FeesAmount*	*DateofReceipt*

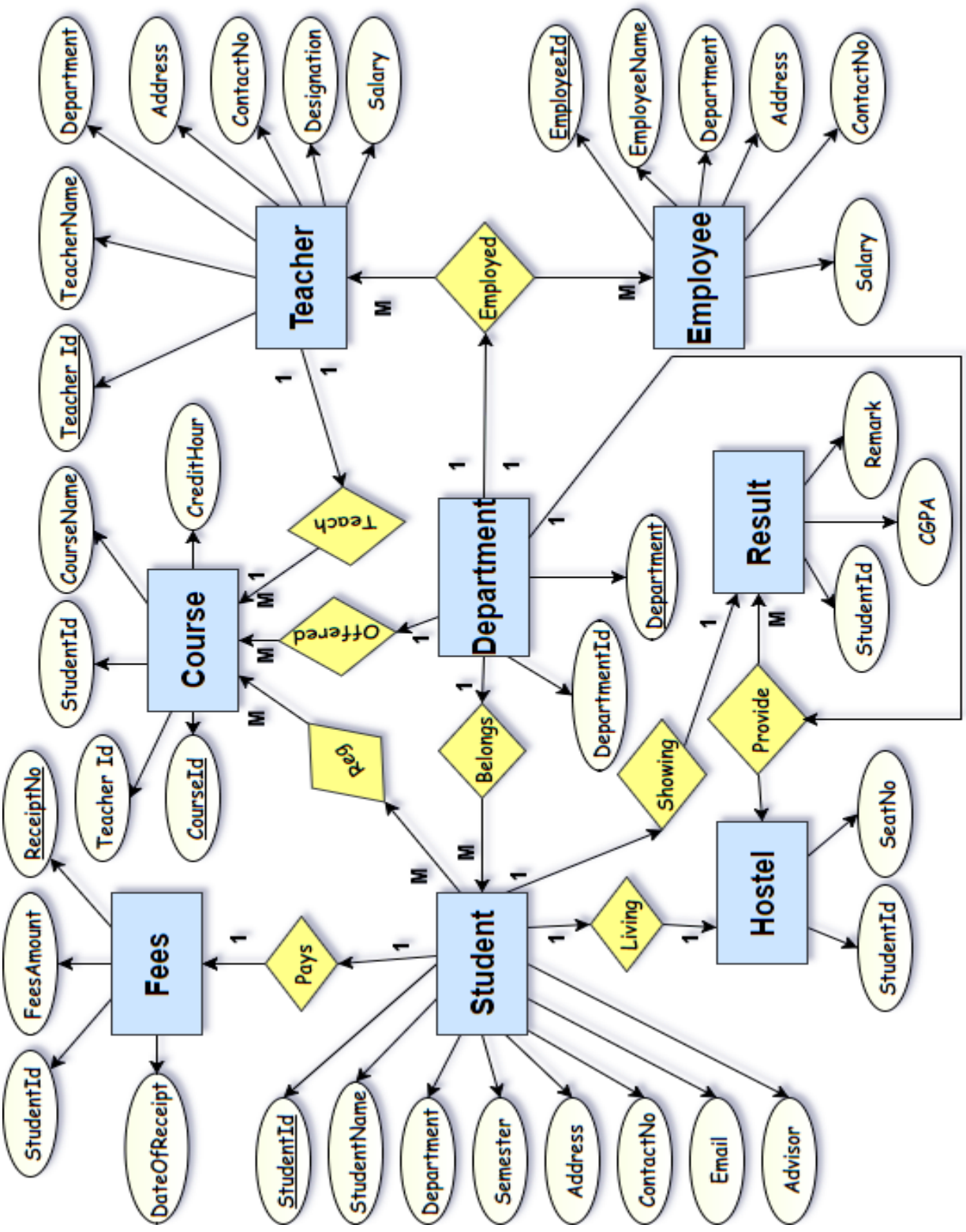
Hostel

Student Id	*Seat No.*

Result

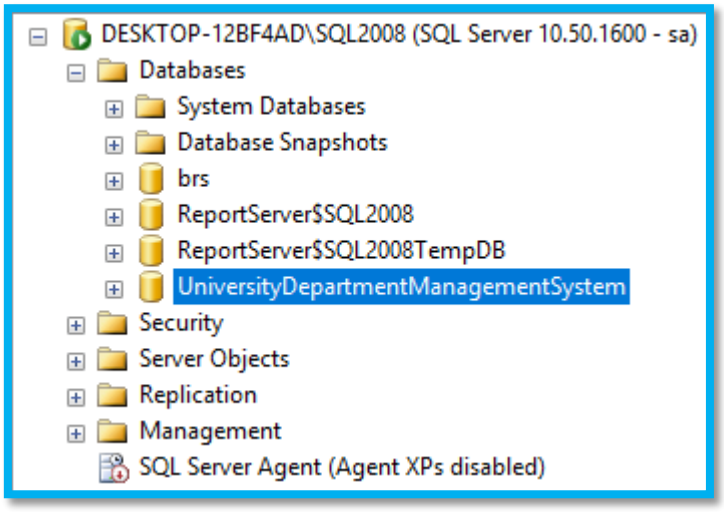
StudentId	*CGPA*	*Remark*

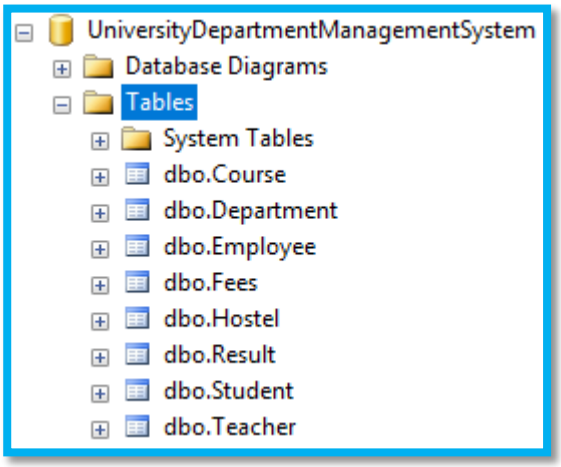
E-R Diagram:


Teacher
Employee
Employed
Department
Course
Offered
Teach
Student
Belongs
Pays
Fees
Living
Hostel
Provide
Result
Showing
Department
TeacherName
Address
ContactNo
Designation
Salary
Teacher Id
EmployeeId
EmployeeName
Department
Address
ContactNo
Salary
CourseName
CreditHour
StudentId
Teacher Id
CourseId
DepartmentId
Department
Remark
CGPA
StudentId
ReceiptNo
FeesAmount
StudentId
DateOfReceipt
StudentId
StudentName
Department
Semester
Address
ContactNo
Email
Advisor
SeatNo
StudentId
1
M

Database query

```
1.create database UniversityDepartmentManagementSystem;
```





Department table

```
create table Department(

DepartmentID varchar(100) not null,
Department varchar(100) not null primary key

);
```

Data for Department Table

- `insert into Department( DepartmentId,Department) values ('CSE 111','CSE');`
- `insert into Department( DepartmentId,Department) values ('BBA 111','BBA');`

	DepartmentID	Department
1	BBA 111	BBA
2	CSE 111	CSE
3	EEE 111	EEE
4	ELL 111	ELL
5	ETE 111	ETE

STUDENT TABLE

```
create table Student(
StudentId varchar(100) not null primary key,
StudentName varchar(100) not null,
Department varchar(100) not null,
Semester varchar(100) not null,
Address varchar(100) not null,
ContactNo varchar(100) not null,
Email varchar(100) not null,
Advisor varchar(100) not null,
);
```

Data for Student Table

- ```
  insert into Student
  (StudentId,StudentName,Department,Semester,Address,ContactN
  o,Email,Advisor, CourseId) values
  ('C161054','Tasbir','CSE','8','Agrabad','01681','Tasbir@yah
  oo.com','Kamal Uddin');
  ```
- ```
  insert into Student
  (StudentId,StudentName,Department,Semester,Address,ContactN
  o,Email,Advisor, CourseId) values
  ('C161050','Shahin','CSE','4','Dewanhut','01898','Shahin@ya
  hoo.com','Jamal Uddin');
  ```

	StudentId	StudentName	Department	Semester	Address	ContactNo	Email	Advisor
1	B101156	Ishmam	BBA	6	AK Khan	01789	ishmam@gmail.com	Giash Uddin
2	B133066	Mizan	BBA	7	Dewanhat	01515	mizan@yahoo.com	Mir Zafar
3	C161050	Shahin	CSE	4	Dewanhut	01898	Shahin@yahoo.com	Jamal Uddin
4	C161054	Tasbir	CSE	8	Agrabad	01681	Tasbir@yahoo.com	Kamal Uddin
5	E161041	Hasan	EEE	2	Kathghor	01910	hasan@gmail.com	Najim Uddin
6	T161055	Ibnul	ETE	3	Newmarket	01745	mallu@hotmail.com	Shanin Elahi

TEACHER TABLE

```
create table Teacher(
TeacherId varchar(100) not null primary key,
TeacherName varchar(100) not null,
Department varchar(100) not null,
Address varchar(100) not null,
ContactNo varchar(100) not null,
Designation varchar(100) not null,
Salary varchar(100) not null,
);
```

Data for Teacher Table

- ```
  insert into
  Teacher(TeacherId,TeacherName,Department,Address,ContactNo,
  Designation,Salary, CourseId) values ('CS16123','Tanvir Ul
  Alam','CSE','Agrabad','01753','Assistant
  Professor','20000')
  ```
- ```
  insert into
  Teacher(TeacherId,TeacherName,Department,Address,ContactNo,
  Designation,Salary, CourseId) values ('BS15523','Hasan
  Ujjaman','BBA','Boropol','01663','Assistant
  Professor','15000');
  ```

	TeacherId	TeacherName	Department	Address	ContactNo	Designation	Salary
1	BS15523	Hasan Ujjaman	BBA	Boropol	01663	Assistant Professor	35000
2	CS15103	Monjur Alam	CSE	Foujdarhat	01619	Lecturer	30000
3	CS16105	MD Safiullah	CSE	Bohoddarhat	01717	Assosiate Professor	70000
4	CS16123	Tanvir Ul Alam	CSE	Agrabad	01753	Assistant Professor	32000
5	ES16112	Soyeb Chy	EEE	GEC	01919	Lecturer	25000
6	LS15105	Alam Chy	ELL	New Market	01717	Assosiate Professor	50000

Employee Table

```
create table Employee(
EmployeeId varchar(100) not null primary key,
EmployeeName varchar(100) not null,
Department varchar(100) not null,
Address varchar(100) not null,
ContactNo varchar(100) not null,
Salary varchar(100) not null,
);
```

Data for Employee Table

- ```
  insert into Employee(EmployeeId,EmployeeName,Department,Address,ContactNo,Salary) values ('CE123','Karim Hasan','CSE','Olongkar','01552','3000');
  ```
- ```
  insert into Employee(EmployeeId,EmployeeName,Department,Address,ContactNo,Salary) values ('BE126','Kamal Chowdhury','BBA','New Market','01772','4000');
  ```

	EmployeeId	EmployeeName	Department	Address	ContactNo	Salary
1	BE124	AB Mahmud	BBA	Agrabad	01478	6000
2	BE126	Kamal Chowdhury	BBA	New Market	01772	4000
3	CE123	Karim Hasan	CSE	Olongkar	01552	3000
4	CE124	Hasem	CSE	Boropol	01588	5000
5	EE123	Abdul	EEE	Agrabad	02589	9000

Course Table

```
create table Course(
CourseId varchar(100) not null primary key,
CourseName varchar(100) not null,
CreditHour varchar(100) not null,
TeacherId varchar(100),
StudentId varchar(100),
);
```

Data for Course Table

- `insert into Course(CourseId,CourseName,CreditHour) values ('CSE-2404','Pulse','3');`
- `insert into Course(CourseId,CourseName,CreditHour) values ('BBA-1102','Accounting','2');`

	CourseId	CourseName	CreditHour	StudentId	TeacherId
1	BBA-1102	Accounting	2	C161050	BS15523
2	CSE-1120	C	4	C161054	NULL
3	CSE-1203	Discrete Math	3	NULL	CS16105
4	CSE-2401	C++	2	C161054	CS15103
5	CSE-2404	Pulse	3	C161050	ES16112
6	CSE-2406	Java	3	C161054	NULL
7	CSE-2407	Database	3	C161050	CS16105

Fees Table

```
create table Fees(
ReceiptNo varchar(100) not null primary key,
StudentId varchar(100) not null,
FeesAmount varchar(100) not null,
DateOfReceipt varchar(100) not null
);
```

Data for Fees Table

- ```
  insert into
  Fees(ReceiptNo,StudentId,FeesAmount,DateOfReceipt) values
  ('1260','C161050','380000','12.01.2018');
  ```
- ```
  insert into
  Fees(ReceiptNo,StudentId,FeesAmount,DateOfReceipt) values
  ('1234','C161054','350000','01.01.2018');
  ```

	ReceiptNo	StudentId	FeesAmount	DateOfReceipt
1	1230	B101156	250000	10.01.2018
2	1234	C161054	350000	01.01.2018
3	1260	C161050	380000	12.01.2018
4	1266	T161055	300000	10.01.2018
5	1269	E161041	400000	10.01.2018

Hostel Table

```
create table Hostel(
StudentId varchar(100) not null,
SeatNo varchar(100) not null,

);
```

Data for Hostel Table

- `insert into Hostel(StudentId,SeatNo)values('C161054','102');`
- `insert into Hostel(StudentId,SeatNo)values('C161050','103');`
- `insert into Hostel(StudentId,SeatNo)values('B101156','104');`

	StudentId	SeatNo
1	C161054	102
2	C161050	103
3	B101156	104

Result Table

```
create table Result(
StudentId varchar(100) not null,
CGPA varchar(100) not null,
Remark varchar(100) not null
);
```

Data for Result Table

- ```
  insert into
  Result(StudentId,CGPA,Remark)values('C161054','3.50','Good'
  );
  ```
- ```
  insert into
  Result(StudentId,CGPA,Remark)values('C161050','3.80','Exece
  llent');
  ```
- ```
  insert into
  Result(StudentId,CGPA,Remark)values('B101156','3.60','Very
  Good');
  ```

	StudentId	CGPA	Remark
1	C161054	3.50	Good
2	C161050	3.80	Execellent
3	B101156	3.60	Very Good
4	E161041	3.55	Good
5	T161055	3.70	Good
6	B101156	3.90	Excellent
  ```

Foreign key relationship in Tables

```
alter table Student add constraint Dp_Student_FK
foreign key (Department) references Department (Department)

alter table Teacher add constraint Dp_Teacher_FK
foreign key (Department) references Department (Department)

alter table Employee add constraint Dp_Employee_FK
foreign key (Department) references Department (Department)

alter table Course add constraint Std_Cou_FK
foreign key (StudentId) references Student (StudentId)

alter table Fees add constraint Std_Fee_FK
foreign key (StudentId) references Student (StudentId)

alter table Fees add constraint Std_Fee_FK
foreign key (StudentId) references Student (StudentId)

alter table Result add constraint Std_Re_FK
foreign key (StudentId) references Student (StudentId)

alter table Hostel add constraint Std_Hos_FK
foreign key (StudentId) references Student (StudentId)
```

Stored Procedure From sql Server to DataBase project:

```
1.create proc AllStudentResultbyId
as
Begin
     SELECT Student.StudentId,Student.StudentName, Result.CGPA

     FROM Result
     inner join Student ON Student.StudentId = Result.StudentId
End
```

Execute- AllStudentResultbyId

	StudentId	StudentName	CGPA
1	C161054	Tasbir	3.50
2	C161050	Shahin	3.80
3	B101156	Ishmam	3.60
4	E161041	Hasan	3.55
5	T161055	Ibnul	3.70
6	B101156	Ishmam	3.90

```
2.create proc StudentResultbyId
@StudentId varchar(100)

as
Begin
     SELECT Student.StudentId, Result.CGPA

     FROM Result
     inner join Student ON Student.StudentId = Result.StudentId

     where Student.StudentId=@StudentId
End
```

Execute-StudentResultbyId 'C161050'

	StudentId	CGPA
1	C161050	3.80

```
3.create proc StudentCoursebyId
@StudentId varchar(100)
as
Begin
     SELECT Student.StudentId,Student.StudentName,
Course.CourseName

     FROM Course
     inner join Student ON Student.StudentId = Course.StudentId

     where Student.StudentId=@StudentId
End
Execute-StudentCoursebyId 'C161054'
```

	StudentId	StudentName	CourseName
1	C161054	Tasbir	C
2	C161054	Tasbir	C++
3	C161054	Tasbir	Java

```
4.alter proc AllStudentCoursebyId
as
Begin
     SELECT
Student.StudentId,Student.StudentName,Student.Semester,
Course.CourseName

     FROM Course
     inner join Student ON Student.StudentId = Course.StudentId
End

Execute-AllStudentCoursebyId
```

	StudentId	StudentName	Semester	CourseName
1	C161050	Shahin	4	Accounting
2	C161054	Tasbir	8	C
3	C161054	Tasbir	8	C++
4	C161050	Shahin	4	Pulse
5	C161054	Tasbir	8	Java
6	C161050	Shahin	4	Database

```
5.create proc AllTeacherTakeCoursebyId
as
Begin
     SELECT Teacher.TeacherId, Course.CourseName

     FROM Course
     inner join Teacher ON Teacher.TeacherId = Course.TeacherId
End
```

```
Execute-AllTeacherTakeCoursebyId
```

	TeacherId	CourseName
1	BS15523	Accounting
2	CS16105	Discrete Math
3	CS15103	C++
4	ES16112	Pulse
5	CS16105	Database

```
6.Create proc TeacherTakeCoursebyId
@TeacherId varchar(100)
as
Begin
     SELECT Teacher.TeacherId,Teacher.TeacherName,
Course.CourseName

     FROM Course
     inner join Teacher ON Teacher.TeacherId = Course.TeacherId
     where Teacher.TeacherId=@TeacherId
End
```

```
Execute-TeacherTakeCoursebyId 'CS16105'
```

	TeacherId	TeacherName	CourseName
1	CS16105	MD Safiullah	Discrete Math
2	CS16105	MD Safiullah	Database

7.
```
create proc StudentNameByDepartment
@Department varchar(100)

as
Begin
     SELECT Student.StudentId,Student.StudentName,
Student.Department

     FROM Student

     where Student.Department=@Department
End
```

Execute-StudentNameByDepartment 'CSE'

	StudentId	StudentName	Department
1	C161050	Shahin	CSE
2	C161054	Tasbir	CSE

8.
```
create proc TeacherNameByDepartment
@Department varchar(100)

as
Begin
     SELECT Teacher.TeacherId, Teacher.TeacherName,
Teacher.Department

     FROM Teacher

     where Teacher.Department=@Department
End
```

Execute- TeacherNameByDepartment 'CSE'

	TeacherId	TeacherName	Department
1	CS15103	Monjur Alam	CSE
2	CS16105	MD Safiullah	CSE
3	CS16123	Tanvir Ul Alam	CSE

Conclusion

- The University Department Management System which capable of storing university resources such as Student, Employee, Teacher, Course & other related factors and their relationship was implemented.

- The system supports SQL Language.

- The system can easily extended by introducing new modules.
