



North South University

Course Title: Database Management System

Project Title: A management System for all university

Project Group-05

Section-05

Course Code: CSE-311

Semester- Fall-2020

Submitted To: Kamruddin Md. Nur

Date of Submission: 14-01-21

Name	Student ID	Group NO
Rofiqul Alam Shehab	1831185042	5
S M Gazzali Arafat Nishan	1831513642	5
Arfana Rahman	1831172042	5

Project Title: A management System for all university

Introduction: All University management system is a system that has all the information of universities in Bangladesh. This system can help a prospective student by providing all the necessary admission related information of different universities. This system stores information of all the universities of Bangladesh under one database, this database can help a prospective student find the information. That he/she is looking for his/her choice of program from different universities, the number of credits offered by different university for that program, the faculties of that university, the cost of the programs for different university etc. And more. This All University Database system will make it easier, convenient and time saving for a prospective student to choose his/her desired University to get admission into.

Description\Goals: A large number of students sat for a given university exam, and they have different choices. These choices are starting from their childhood. At that time, they usually try to visit different university's websites, but they don't get full information from a single university website. Then, they search for different university websites to look for vast information about that university. Sometimes, it's so time-consuming & they waste a lot of time on it. Hence, we are trying to create a database for all universities whether the examinee can get all the university information. Such as student facilities, exam & admission date, established year, vice-chancellor name, teacher quality, student facilities, publications, university ranking, etc.

Context: We all know that the traditional paper based system has some flaws. An organization can lose or damage the information or can't monitor all the university's information at a same time what is needed for an appeared examine. So, basically we tried to make a DBMS that will allow the organization to let the examine know about information for all university into a single application. They don't need to search in different university's website to look for basic and depth information. Using this application, they will be able to see all the information in a single application. The organization will be able to provide basic information about the university.

Administrator or an organization panel is in control of everything who will be able to access the database and can perform some actions like adding information in the database, editing or updating the data, deleting data from the database according to the organization's necessity. For example, the organization will be able to store university information, student's information, teacher's research & award information, rankings of university, necessary email addresses, course's list, financial aid, scholarship budget, research budget etc.

Technology Stacks:

1. **MySQL:** We have used “MySQL” server to build our project. Using SQL, we have created some tables.
2. **Draw.io :** E.R. Diagram is designed in the draw.io site.

Back-End Details:

Aggregate functions are used to store information in the database. For example:-

- i. Count Function –We have used count functions, then the organization will be able to determine the total number of the university in the database. From this, an admin will be able to how many university exists in the database.
- ii. Sum Function- Sum functions can easily find the total amount of research budget or admission fees yearly. it will help the organization to find out monthly research budget and yearly admission fee.
- iii. Join Clause- Full outer join, Inner join, Left outer join, Right outer join can be used to join multiple tables with each other. So, that the user can reduce data redundancy
- iv. In the all university management database. Using these clauses, a stable connection between tables can be created.
- v. Like\NULL\logical condition- Using these operation, the administrator will be able to search on the database with specified wildcard search.
- vi. Group by\Having clause- By using these clauses, the administrator will be able to aggregate using group functions.
- vii. Subqueries- Multiple-Row subqueries can be performed in the database.
- viii. Universities information display on the profile .For example, University name, it's id, rank, email, contact, admission.
- ix. Students & Teachers information also display on the profile. Such as , teachers publications,
- x. Departments & Courses information can be achieved easily. Such as Department ID, Number of courses, Credit cost etc.
- xi. Miscellaneous Information- it contains various type of information like accreditation, academic information, online portal system, start-up availability, research facility or budget etc.

Development:

The complete development time can be divided in three phases. In each phase, we focused on a particular target and completed it step by step.

Phase 1: In this phase, we designed the ER diagram and Database Schema. The first prototype was later scrapped and a new optimized schema was developed in the projected time.

Phase 2: This phase is focused on backend development. We used SQL to completely develop the backend side and MySQL to connect database and perform query.

Phase 3: This was the testing phase. The system was running fluently and performing desired functions.

Limitations: We have created a database called “A DBMS for all university” and populated data in the database perfectly. Basically, the project goal was to develop backend activities and drawing ER Diagram along appropriate information. While we were developing the project, we had faced some minor issues. However, minor errors were found and solved in the back end section. We think, we could develop this DBMS a little more if we have enough time to work for the frontend section. Since the project was basically focused on backend section that’s why front end section isn’t performed in this project. But, we hope that we could do that. On the other hand, triggers and stored procedure can’t be performed in this project.

Conclusion & Future plan: While working on the project, we found that the project can be extended with more functionality. It can be connected with frontend part. There can be more administrative controls. The search function can be used here for better controls.

E.R. Diagram:

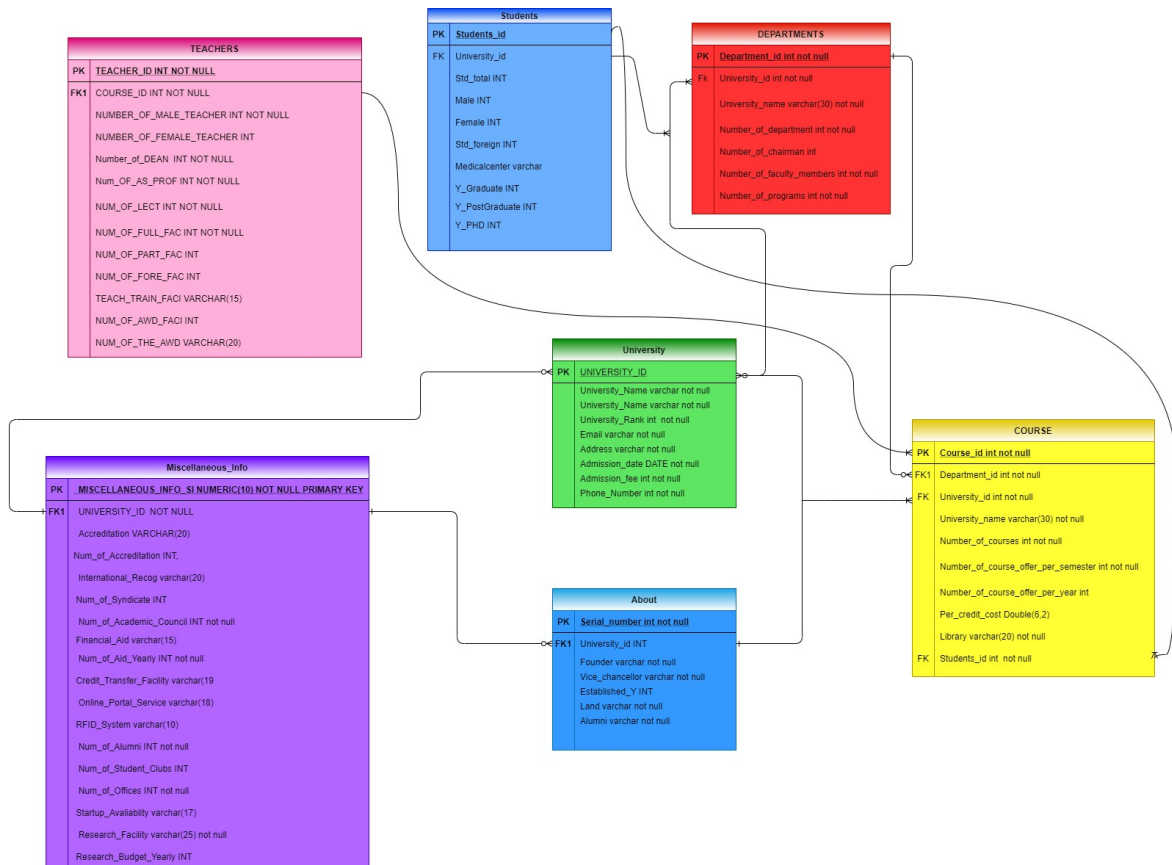
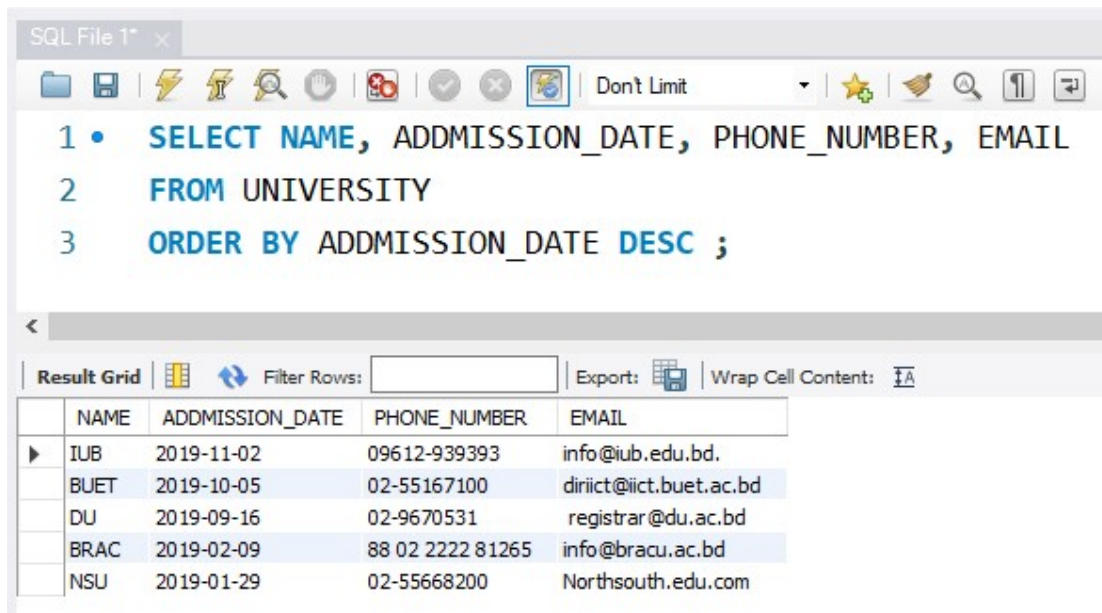


Fig-01-ER Diagram

Queries: We have included some queries & screenshot from MySQL below to fulfil our project goal.

1.



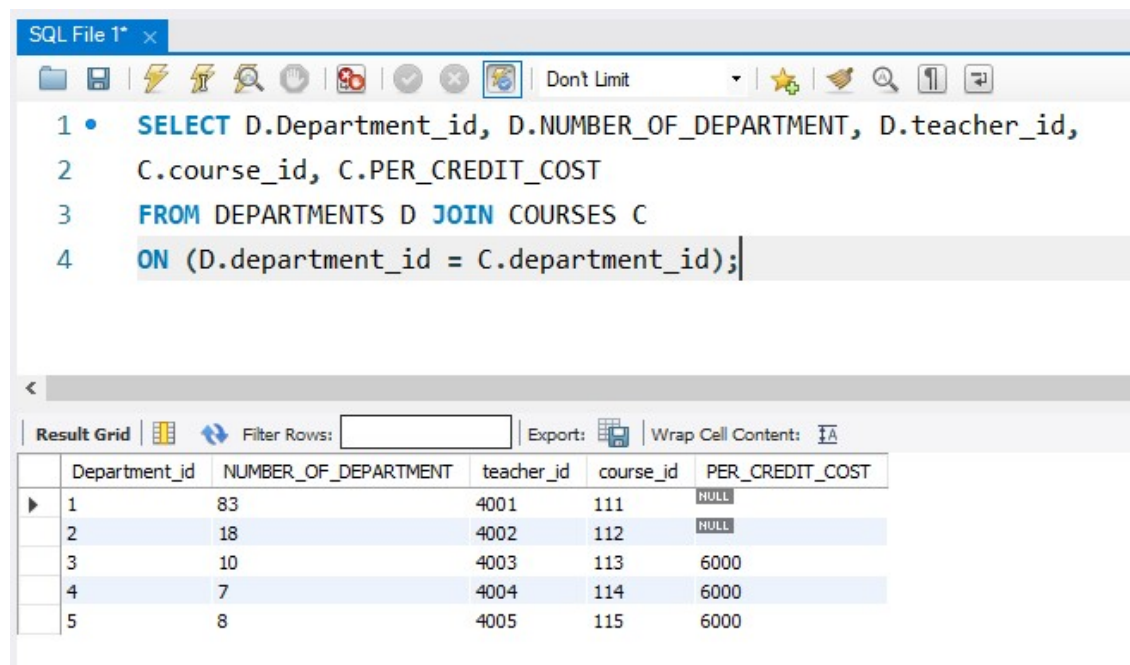
The screenshot shows the MySQL IDE interface. The query editor contains the following SQL query:

```
1 • SELECT NAME, ADMISSION_DATE, PHONE_NUMBER, EMAIL
2 FROM UNIVERSITY
3 ORDER BY ADMISSION_DATE DESC ;
```

Below the query editor, the 'Result Grid' tab is active, displaying the results of the query in a table format:

	NAME	ADMISSION_DATE	PHONE_NUMBER	EMAIL
▶	IUB	2019-11-02	09612-939393	info@iub.edu.bd.
	BUET	2019-10-05	02-55167100	dirict@iict.buet.ac.bd
	DU	2019-09-16	02-9670531	registrar@du.ac.bd
	BRAC	2019-02-09	88 02 2222 81265	info@bracu.ac.bd
	NSU	2019-01-29	02-55668200	Northsouth.edu.com

2. Using Join clause to make a connection between two tables.



The screenshot shows the MySQL IDE interface. The query editor contains the following SQL query:

```
1 • SELECT D.Department_id, D.NUMBER_OF_DEPARTMENT, D.teacher_id,
2 C.course_id, C.PER_CREDIT_COST
3 FROM DEPARTMENTS D JOIN COURSES C
4 ON (D.department_id = C.department_id);
```

Below the query editor, the 'Result Grid' tab is active, displaying the results of the query in a table format:

	Department_id	NUMBER_OF_DEPARTMENT	teacher_id	course_id	PER_CREDIT_COST
▶	1	83	4001	111	NULL
	2	18	4002	112	NULL
	3	10	4003	113	6000
	4	7	4004	114	6000
	5	8	4005	115	6000

3. Using Sub query to find the min number of dean.

The screenshot shows a SQL File 1 window with the following query:

```
1 • SELECT TEACHER_ID, NUM_OF_FULL_FAC, NUM_OF_THE_AWD, Number_of_DEAN
2 FROM TEACHERS
3 WHERE Number_of_DEAN =
4         (SELECT MIN(Number_of_DEAN)
5          FROM TEACHERS);
```

Below the query, the Result Grid shows the following data:

	TEACHER_ID	NUM_OF_FULL_FAC	NUM_OF_THE_AWD	Number_of_DEAN
▶	4001	550	UFO	4
	4004	570	FAO	4
*	NULL	NULL	NULL	NULL

4. Updating information in a row.

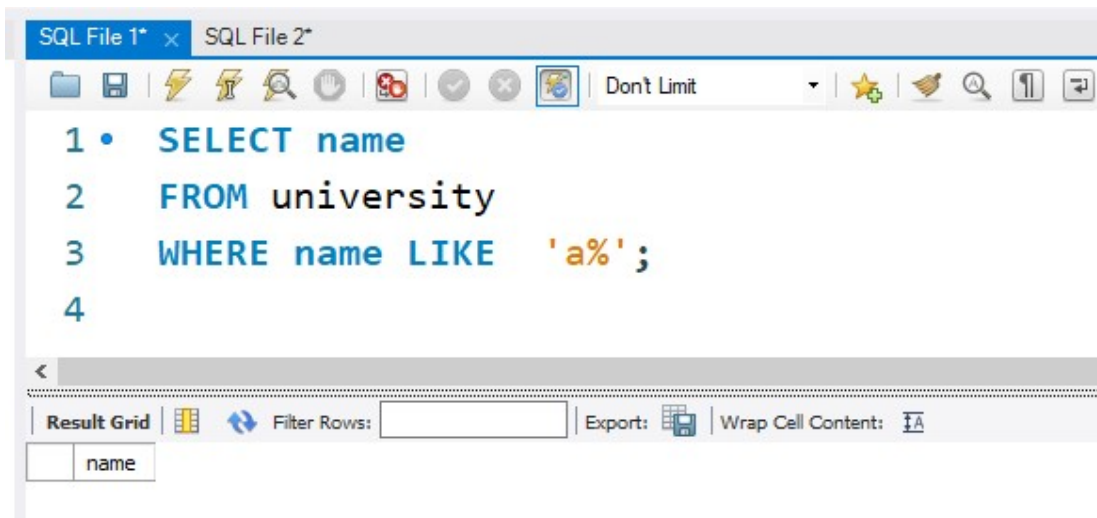
The screenshot shows a SQL File 1 window with the following queries:

```
1 • update teachers
2 set number_of_dean=6
3 where teacher_id=4001;
4
5 • select teacher_id, number_of_dean
6 from teachers
7 where teacher_id =4001;
```

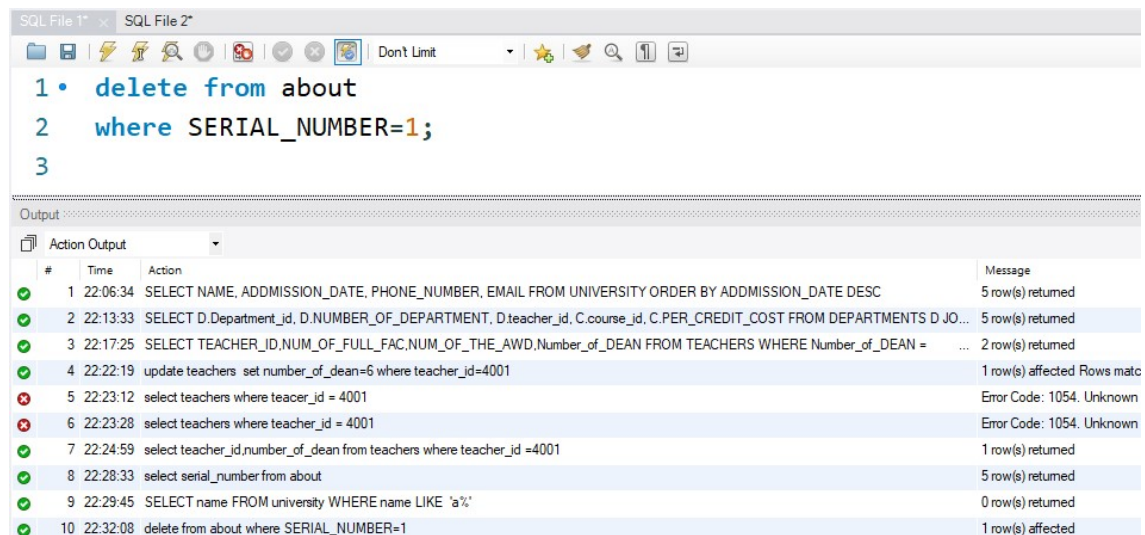
Below the queries, the Result Grid shows the following data:

	teacher_id	number_of_dean
▶	4001	6

5. Searching data using like.



6. Deleting data from about table.



END