

**DEPARTMENT OF COMPUTER SCIENCE**

Course Name: Database Management Sys I

Course Code: CPS 542

**TERM PAPER**

*Submitted by:*

**ANUSH REDDY KUNDURU**

**SUMANTH KANUKUNTLA  
SIDDARTH GOPINATH  
REDHA ALMAHDI**

*Submitted to:*

**Dr.** [**BUCKLEY,**](http://www.udayton.edu/directory/artssciences/computerscience/courte_dale_e.php) **JAMES**

Associate Professor; Director of Graduate Program in Computer Science

University of Dayton

## Introduction:-

The goal of this project is to create universities database to centrally handle the information of all the ungraduated and graduate students in the universities, and to provide access to this information with an easy to use interface that can be accessed by any device with basic languages programs. In addition, the database is the only professional structure for organizing, manipulating and retrieving deferent information structures and huge amount of data. To get quick access, high security, integrity, and statistics, multiuser access etc. we have to use database management systems (DBMS) to fetch these advantages that I mentioned. To put it briefly, a university involves of a number of departments. Each department has several colleges. Students enroll in a particular course. Each module is taught by a professor from the appropriate department, and each professor teaches a group of students.

## Implementation

The database was written to work with the oracle database to store the data, and it is written in SQL to work with any other DBMS. The content of the database is divided into five table’s components. The first is the universities table which is includes all information that related to universities data which already shown in Figure 1.1. This table is used to look up the University ID, Name, Rank scores, contact, Location, Phone No, and Email. The second table used is for college’s data where each university has at least five colleges. This table holds the most of the significant information that describes college schema with details as shown below in Figure 1.1.

The third table used by the system is used to track the colleges of the department’s information through the degree program. This table holds foreign key which is to specify colleges in which department. Therefore, each college might have two departments which mean different location as well. The fourth table holds entire information of professors which is representing the professor id, name, specialization, schedule, office hours, and in which college is the professor as shown in Table 1.1. The other two tables are used by

the database to manage student’s records which are Course table and Student table.

|  |  |
| --- | --- |
|  |  |

Table 1.1 database tables structure

|  |
| --- |
|  |

# Figure 1.1 University E-R Diagram

College

Department

Student

Professor

consists of

Course/Section

has

attends

Works

1

N

M

Dean

College code

college name

college ranking

name

Block

class schedule

officehours

prof id

specialization

Contact

Dept name

Num\_of\_Faculty

Dept.Chair\_id

name

dob

Student id

Major

GPA

course title

num\_of\_

seats

Prof\_id

ClassLocation

course id

Advisor\_id

M name

F name

L name

University

consists of

Univ. ID

Univ. Name

Univ.Ranking

location

address line 1

address line 2

state

zipcode

Teaches

N

Course\_id(s)

1

N

1

1

1

N

N

1

Contact

Contact

Phone

E-mail

Contact

Dept\_id

Age

Resources

Univ\_id

College.code

F name

M name

L name

Dept\_id

Collegecode

Dept id

* **Relational DataBase Schema**

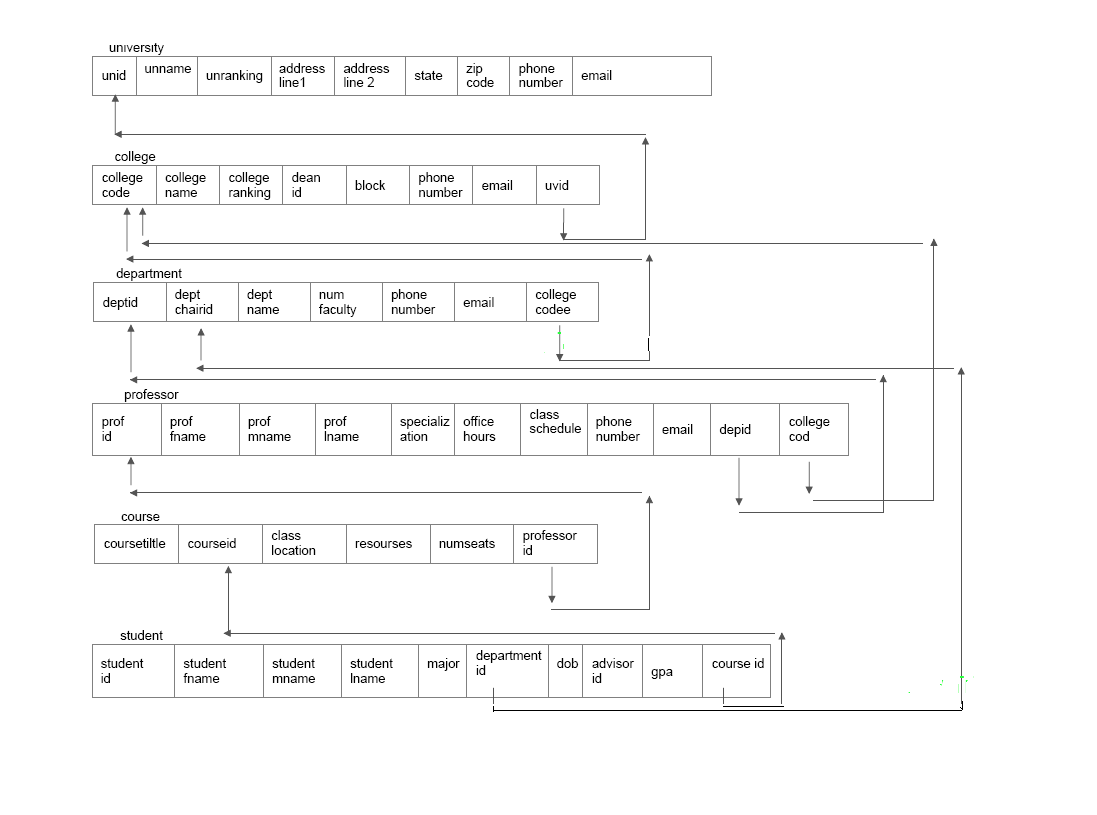
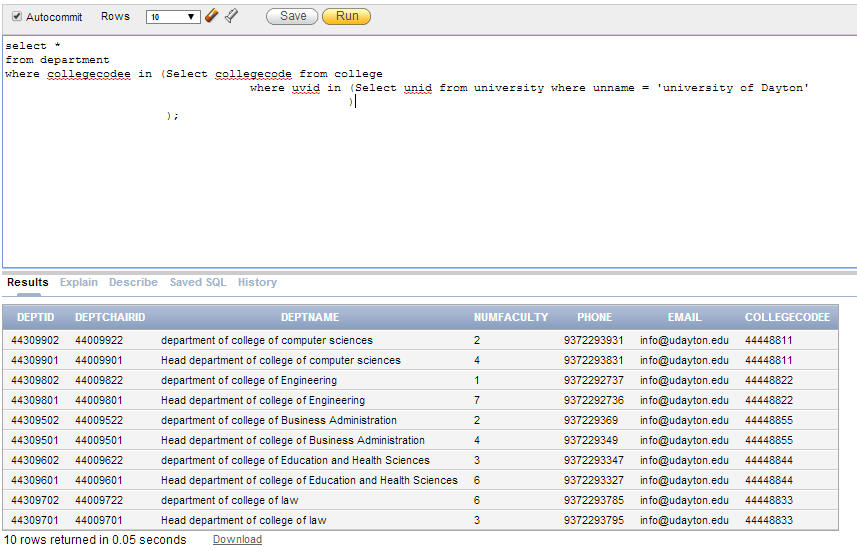
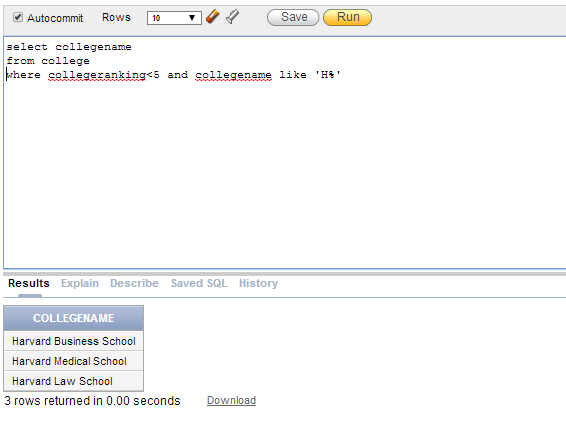


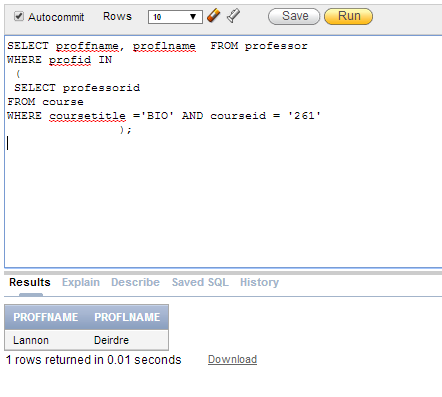
Figure 1.2 Schema diagram for entire database

* **SQL Queries:**

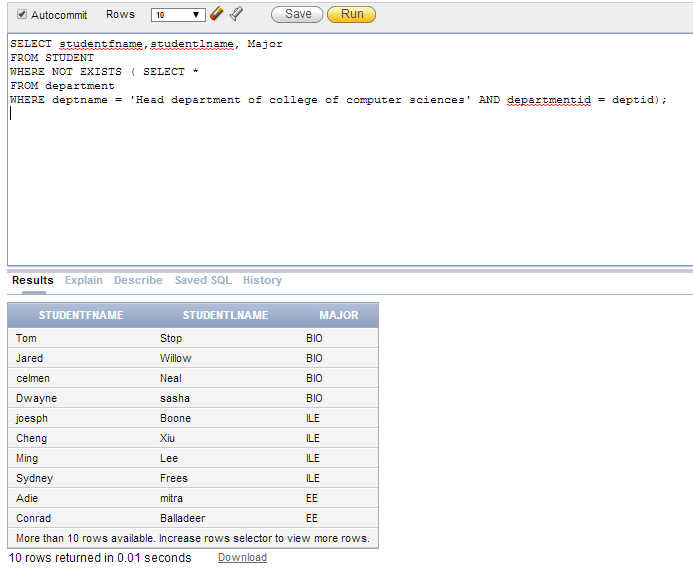
1. Retrieve all departments’ information from University of Dayton.



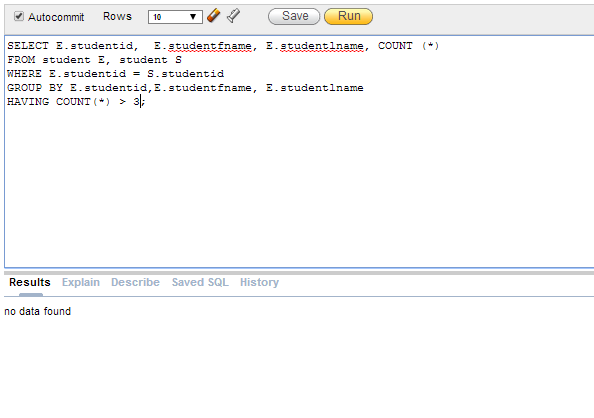
1. Retrieve all colleges that their rank less than 5 and name starting with ‘H’.
2. Retrieve the name of professor who are teaching BIO 261 course



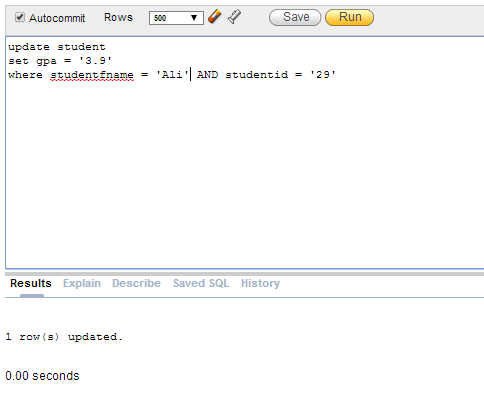
1. Retrieve all students who are in computer science department and their majors.



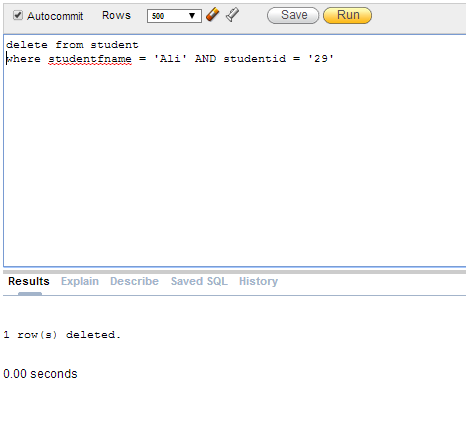
1. Retrieve the names of the students who enroll more than 3 courses.
2. Select student taking more than 3 courses.



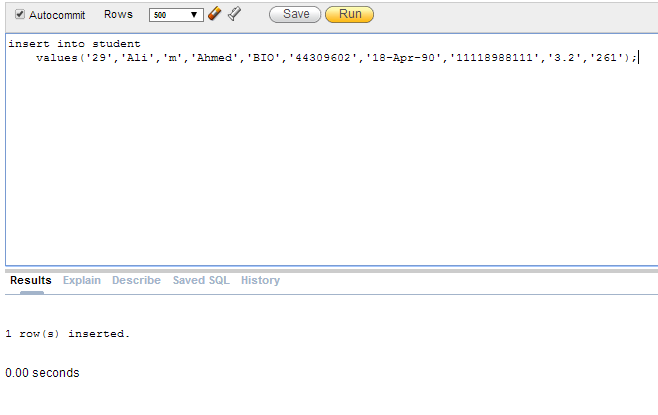
1. Modify the GPA attribute of the STUDENT tuple with student id= ‘29’ and studentfname = ‘ALI’ to ‘3.9’



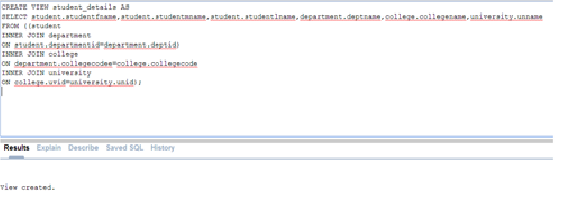
1. Delete the record for the student whose name is ‘Ali’ and student id is ‘29’.

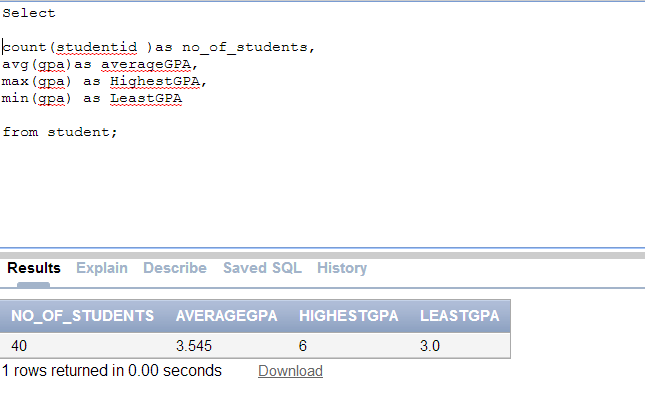


1. Insert a new student ('29','Ali','m','Ahmed','BIO','44309602','18-Apr-90', '11118988111', '3.2','261') in the database.



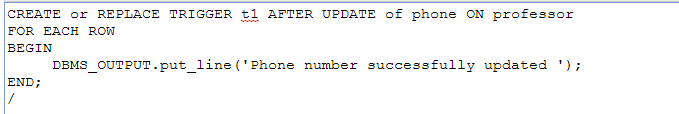
1. Create a view with student details like his name and department, college and university names

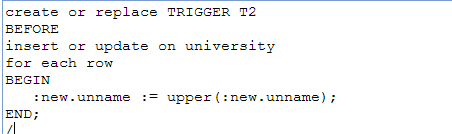


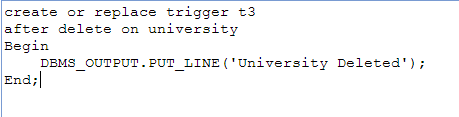
1. This query calculates the average gpa of all student and also shows the highest and least gpa  
   

* **TRIGGERS:**

1. A trigger is fired to display a message on the screen whenever a professor phone number Is updated



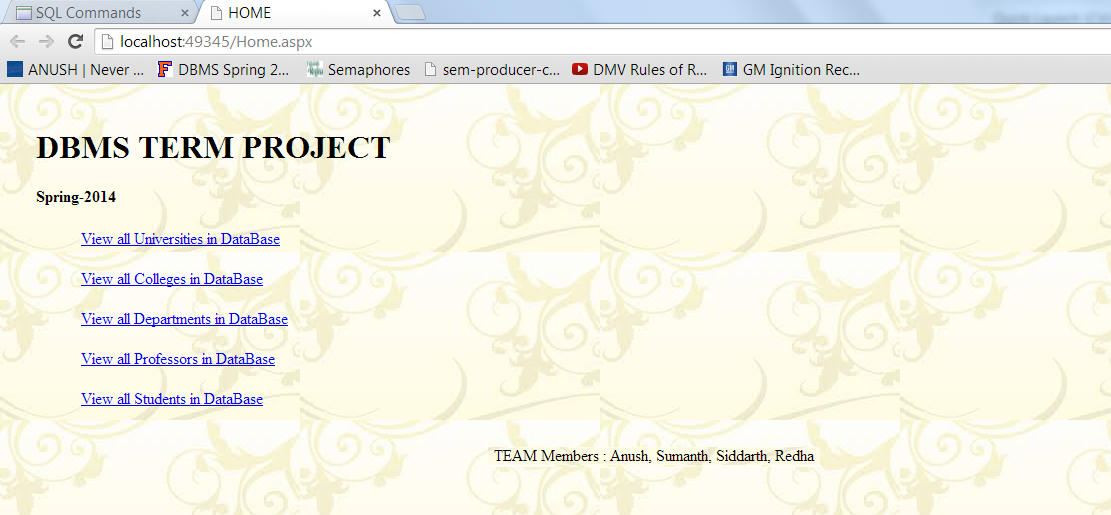
1. A trigger is fired to convert the university names to upper case letters whenever a new university name is inserted or updated 
2. A trigger is fired to display a message whenever a University tuple is deleted.



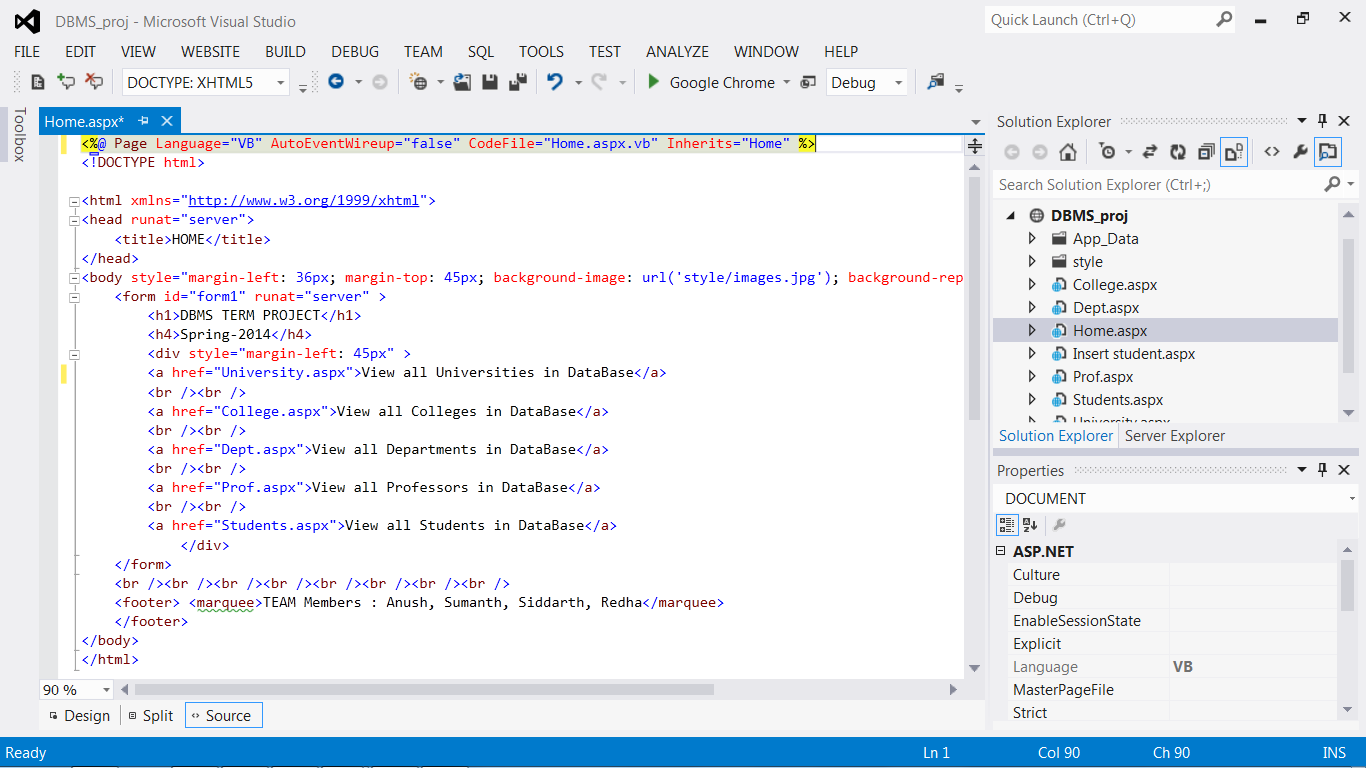
* **WEB INTERFACE:**

Web interface is developed using .Net framework version 4.0

The home page looks like this:

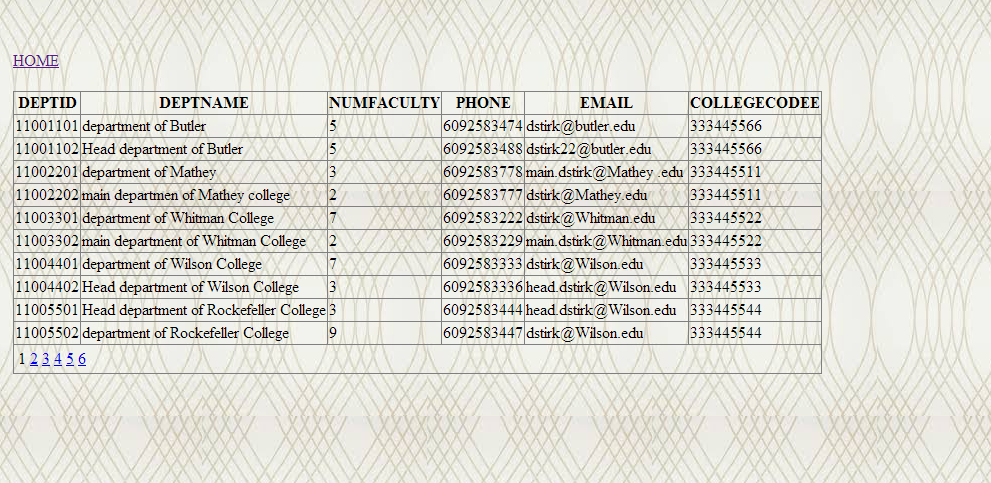


Sample code snippet for the above interface:



Screenshots of other pages in the interface:

1. This page shows the different universities present I the database and when a particular university is selected, the colleges present in that university are displayed in another below.   
   
2. This page shows all the department s in the Database



* **OBSERVATIONS:**

The inspiration for our is project our college. Looking at the structure of University of Dayton, different colleges under it and different departments under each college we designed our project. The got started very quickly with our group, we sat down together and we drew up a blue-print of what we thought our project needed and what information was critical. After we had this blue-print figured out we had a smooth transition in implementing our original mockup which included drawing the ER diagram, creating schema and populating the Database. A big problem that we encountered was dealing with triggers. When implementing triggers we encountered several errors like Table mutation error, Trigger revalidation error etc. To overcome this obstacle we decided to design basic triggers with only display messages and not include any DML triggers. In our queries we tried to include as many varieties as possible by including views, insert, update and delete statements. All four of our team members were eager on implementing them and we were all trying to learn them as we went along with project. Overall our group learned a great deal from this project and it really helped our understanding of how much meticulous work and planning is needed just to setup a basic Database.