|  |
| --- |
| Highline University’s Mentor Program (HUMP) Database System |
| **Project Report** |
| Capstone Project |

|  |
| --- |
|  |
| **Name:** Sayem Shahrier & Justin Beach  **Professor:** IIknur Aydin  **Semester:** Spring 2017  **Date:** 05/12/17  **Class:** BCS 260 |
|  |

**Table of Contents**

Data Model...................................................................................................................................2

Database Design..........................................................................................................................3

*Denormilization Reasoning*.................................................................................................4

*Properties of Columns*.........................................................................................................7

*Referential Integrity Constraints*.......................................................................................10

Database Implementation.......................................................................................................11

*SQL Scripts: Create Tables*...............................................................................................11

*SQL Scripts: Insert Data*...................................................................................................14

*Screenshot of Our Created Database*................................................................................16

**Data Model**

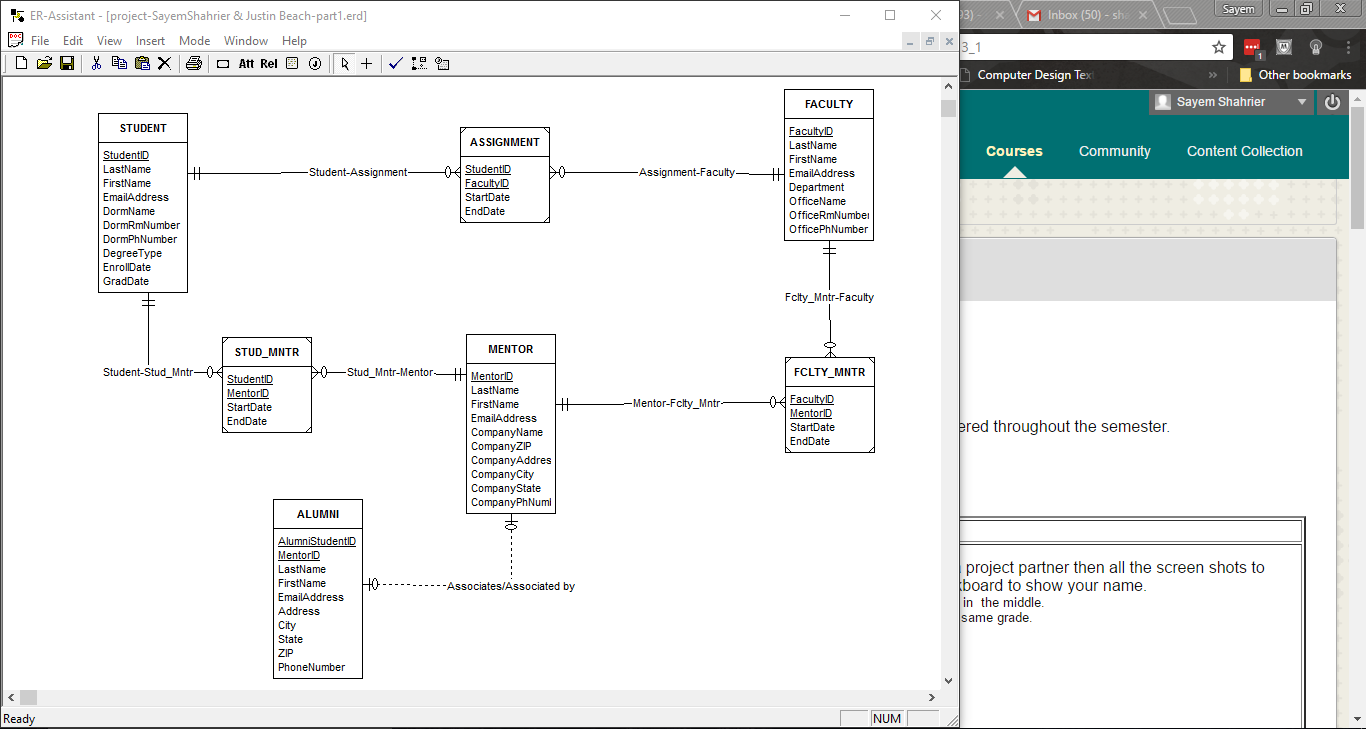


Image Description: Screenshot of Part 1.

**Database Design**

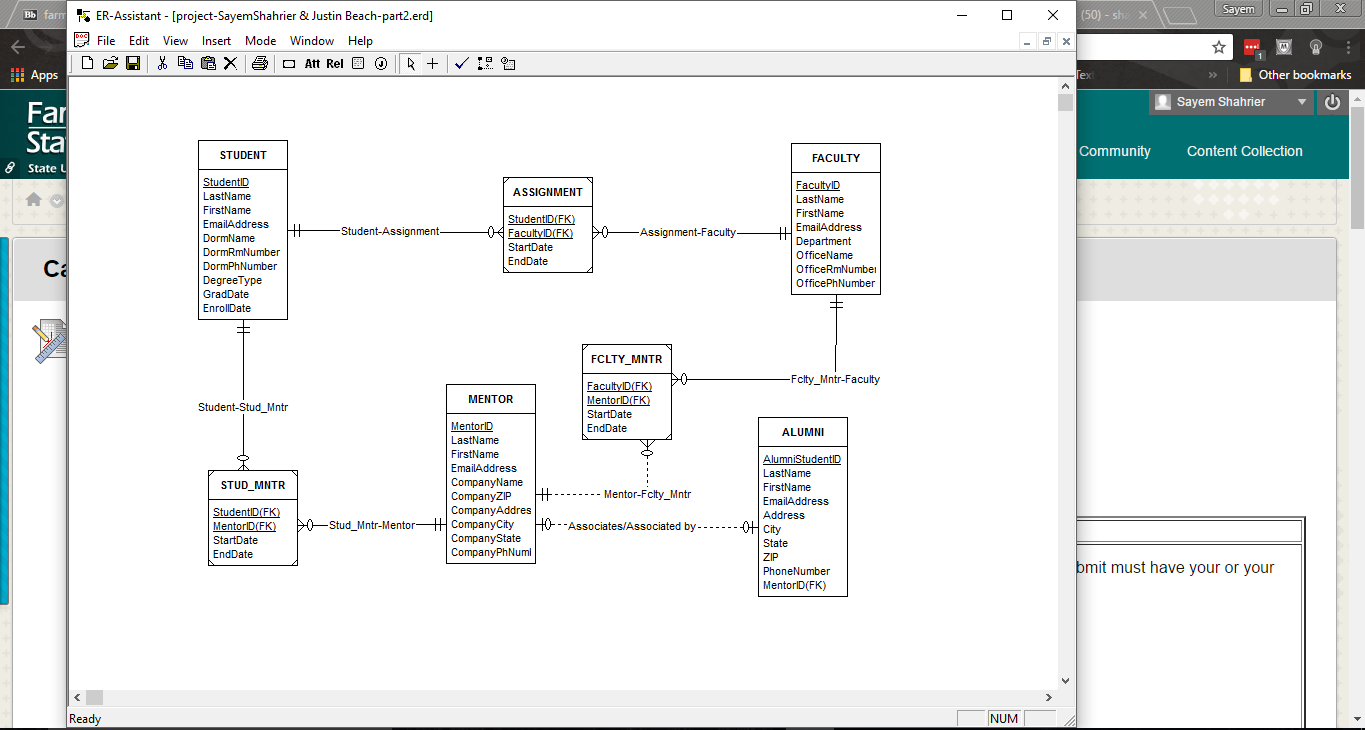


Image Description: Screenshot of Part 2.

***Denormilization***

Reasoning:

**MENTOR** table consisting of company information that was denormalized due to the assumptions that the attributes such as Address, ZIP and State would rarely change for the company itself. Another assumption was that an individual mentor has a possibility of changing companies at any given time, and the fact that multiple mentors could be associated with a single company. This design is more focus on the individual mentor over company data based on client’s request.

**STUDENT** table consists of the individual student and dorms that were denormalized due to the reasoning that every student is required to live on the campus. The dorm information for a given student is associated with the student’s ID number, and that the ID number will provide the rest of the student’s available details. Overall the design is based on the idea of associating the student with their dorm information assuming the dorm is a shared living space of two or more students with one dorm phone number as discussed with the client.

Department located within the **FACULTY** table was denormalized due to the assumption that faculty members could share a given work space such as the OfficeName, OfficeRmNumber and the Departmenttheyworkin. Despite such assumptions given towards shared space, a second assumption was made in reference towards each Faculty member having their own unique OfficePnNumber.

Based on our current design:

**MentorID 🡪** ( LastName, FirstName, EmailAddress, CompanyName, CompanyZip, CompanyAddress, CompanyCity, CompanyState, CompanyPnNumber)

**StudentID 🡪** (LastName, FirstName, EmailAddress, DormName, DormRmNumber, DormRmPhone, EnrollDate, GradDate, DegreeType)

**FacultyID** 🡪 (LastName, FirstName, EmailAddress, Department, OfficeName, OfficeRmNumber, OfficePnNumber)

Additional Remarks:

Most of these details are assumption because it was not explicitly mentioned in our user requirements. But we verified with our client that this database design would be best and appropriate for their need. Overall this will simplify and be the efficient way to implement the design for the database and provide the ideal solution for our client.

***Properties of Columns***

**STUDENT**

**Column Name Data Type Key Required Default Value Remark**

StudentID Integer Primary Key Yes None Surrogate Key:

Initial Value = 1000

Auto Increment=1

LastName Char(25) No Yes None

FirstName Char(25) No Yes None

EmailAddress VarChar(25) No Yes None

DormName Char(25) No Yes None

DormRmNumber Char(10) No Yes None

DormRmPhone Char((10) No Yes None Format:###-###-####

EnrollDate Date No Yes None Format: yyyy-mm-dd

GradDate Date No Yes None Format: yyyy-mm-dd

DegreeType Char(25) No Yes None

**STUDENT** (**StudentID,** LastName, FirstName, EmailAddress, DormName, DormRmNumber, DormRmPhone, EnrollDate, GradDate, DegreeType)

**FACULTY**

**Column Data Type Key Required Default Value Remark**

FacultyID Integer Primary Key Yes None Surrogate Key:

Initial Value = 2000

Auto Increment=1

LastName Char(25) No Yes None

FirstName Char(25) No Yes None

EmailAddress VarChar(25) No Yes None

Department Char(25) No Yes None

OfficeName Char(25) No Yes None

OfficeRmNumber Char(10) No No None

OfficePnNumber Char(25) No Yes None Format:###-###-####

**FACULTY** (**FacultyID**, LastName, FirstName, EmailAddress, Department, OfficeName, OfficeRmNumber, OfficePnNumber)

**MENTOR**

**Column Data Type Key Required Default Value Remark**

MentorID Integer Primary Key Yes None Surrogate Key:

Initial Value = 3000

Auto Increment=1

LastName Char(25) No Yes None

FirstName Char(25) No Yes None

EmailAddress VarChar(25) No Yes None

CompanyName Char(25) No Yes None

CompanyZip Char(5) No No None Format: #####

CompanyAddress Char(25) No No None

CompanyCity Char(25) No No None

CompanyState Char(2) No No None Format: AA

CompnayPnNumber Char(25) No Yes None Format:###-###-####

**MENTOR** (**MentorID**, LastName, FirstName, EmailAddress, CompanyName, CompanyZip, CompanyAddress, CompanyCity, CompanyState, CompanyPnNumber)

**ALUMNI**

**Column Data Type Key Required Default Value Remark**

AlumniStudentID Integer Primary Key Yes None Surrogate Key:

Initial Value = 4000

Auto Increment=1

MentorID Integer Foreign Key No None REF: MENTOR

Must be UNIQUE

LastName Char(25) No Yes None

FirstName Char(25) No Yes None

EmailAddress VarChar(25) No Yes None

Address Char(25) No No None

City Char(25) No No None

State Char(2) No No None Format: AA

Zip Char(5) No No None Format: #####

PhoneNumber Char(25) No Yes None Format:###-###-####

**ALUMNI** (**AlumniStudentID**, *MentorID*, LastName, FirstName, EmailAddress, Address, City, State, Zip, PhoneNumber)

Intersecting Tables:

**ASSIGNMENT**

**Column Data Type Key Required Default Value Remark**

StudentID Integer Primary Key, Foreign Key Yes None REF: STUDENT

FacultyID Integer Primary Key, Foreign Key Yes None REF: FACULTY

StartDate Date No Yes None Format: yyyy-mm-dd

EndDate Date No Yes None Format: yyyy-mm-dd

**ASSIGNMENT** (*StudentID*, *FacultyID*, StartDate, EndDate)

**STUD\_MNTR**

**Column Data Type Key Required Default Remark**

StudentID Integer Primary Key, Foreign Key Yes None REF: STUDENT

MentorID Integer Primary Key, Foreign Key Yes None REF: MENTOR

StartDate Date No Yes None Format: yyyy-mm-dd

EndDate Date No Yes None Format: yyyy-mm-dd

**STUD\_MNTR** (*StudentID, MentorID*, StartDate, EndDate)

**FCTLY MNTR**

**Column Data Type Key Required Default Remark**

FacultyID Integer Primary Key, Foreign Key Yes None REF: FACULTY

MentorID Integer Primary Key, Foreign Key Yes None REF: MENTOR

StartDate Date No Yes None Format: yyyy-mm-dd

EndDate Date No Yes None Format: yyyy-mm-dd

**FCTLY\_MNTR** (*FacultyID, MentorID*, StartDate, EndDate)

***Referential Integrity Constraints***

**Relationship Referential Integrity Constraint Cascading Behavior**

**Parent Child On Update On Delete**

Student ASSIGNMENT StudentID in ASSIGNMENT must exist in No No

StudentID in STUDENT.

Faculty ASSIGNMENT FacultyID in ASSIGNMENT must exist in No No

FacultyID in FACUTLY.

Student STUD\_MNTR StudentID in STUD\_MNTR must exist in No No

StudentID in STUDENT.

Mentor STUD\_MNTR MentorID in STUD\_MNTR must exist in No No

MentorID in MENTOR.

Faculty FCTLY\_MNTR FacultyID in FCTLY\_MNTR must exist in No No

FacultyID in FACULTY.

Mentor FCTLY\_MNTR MentorID in FCTLY\_MNTR must exist in No No

MentorID in MENTOR.

**Database Implementation**

***SQL Scripts: Create Tables***

CREATE TABLE STUDENT (

Student\_ID INT NOT NULL AUTO\_INCREMENT,

LastName CHAR(25) NOT NULL,

FirstName CHAR(25) NOT NULL,

EmailAddress VARCHAR(100) NOT NULL,

DormName CHAR(25) NOT NULL,

DormRmNumber CHAR(10) NOT NULL,

DormRmPhone CHAR(14) NOT NULL,

EnrollDate DATE NOT NULL,

GradDate DATE NOT NULL,

DegreeType CHAR(25) NOT NULL,

CONSTRAINT Student\_PK PRIMARY KEY (Student\_ID)

);

ALTER TABLE STUDENT AUTO\_INCREMENT = 1000;

CREATE TABLE FACULTY (

Faculty\_ID INT NOT NULL AUTO\_INCREMENT,

LastName CHAR(25) NOT NULL,

FirstName CHAR(25) NOT NULL,

EmailAddress VARCHAR(100) NOT NULL,

Department CHAR(25) NOT NULL,

OfficeName CHAR(25) NOT NULL,

OfficeRmNumber CHAR(10) NOT NULL,

OfficePnNumber CHAR(14) NOT NULL,

CONSTRAINT Faculty\_PK PRIMARY KEY (Faculty\_ID)

);

ALTER TABLE FACULTY AUTO\_INCREMENT = 2000;

CREATE TABLE MENTOR (

Mentor\_ID INT NOT NULL AUTO\_INCREMENT,

LastName CHAR(25) NOT NULL,

FirstName CHAR(25) NOT NULL,

EmailAddress VARCHAR(100) NOT NULL,

CompanyName CHAR(25) NOT NULL,

CompanyZip CHAR(5) NULL,

CompanyAddress CHAR(50) NULL,

CompanyCity CHAR(25) NULL,

CompanyState CHAR(2) NULL,

CompanyPnNumber CHAR(14) NOT NULL,

CONSTRAINT Mentor\_PK PRIMARY KEY (Mentor\_ID)

);

ALTER TABLE MENTOR AUTO\_INCREMENT = 3000;

CREATE TABLE ALUMNI (

AlumniStudent\_ID INT NOT NULL AUTO\_INCREMENT,

Mentor\_ID INT NULL,

LastName CHAR(25) NOT NULL,

FirstName CHAR(25) NOT NULL,

EmailAddress VARCHAR(100) NOT NULL,

Address CHAR(50) NOT NULL,

City CHAR(25) NOT NULL,

State CHAR(2) NOT NULL,

Zip CHAR(5) NOT NULL,

PhoneNumber CHAR(14) NOT NULL,

CONSTRAINT ALUMNI PRIMARY KEY (AlumniStudent\_ID),

CONSTRAINT MENTOR FOREIGN KEY (Mentor\_ID)

REFERENCES MENTOR (Mentor\_ID),

CONSTRAINT ALUMNI\_MENTOR UNIQUE (Mentor\_ID)

);

ALTER TABLE ALUMNI AUTO\_INCREMENT = 4000;

CREATE TABLE ASSIGNMENT (

Student\_ID INT NOT NULL,

Faculty\_ID INT NOT NULL,

StartDate DATE NOT NULL,

EndDate DATE NOT NULL,

CONSTRAINT S\_F\_PK PRIMARY KEY (Student\_ID , Faculty\_ID),

CONSTRAINT S\_F\_Student\_FK FOREIGN KEY (Student\_ID)

REFERENCES STUDENT (Student\_ID)

ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT S\_F\_Faculty\_FK FOREIGN KEY (Faculty\_ID)

REFERENCES FACULTY (Faculty\_ID)

ON DELETE NO ACTION ON UPDATE NO ACTION

);

CREATE TABLE STUD\_MNTR (

Student\_ID INT NOT NULL,

Mentor\_ID INT NOT NULL,

StartDate DATE NOT NULL,

EndDate DATE NOT NULL,

CONSTRAINT S\_M\_PK PRIMARY KEY (Student\_ID , Mentor\_ID),

CONSTRAINT S\_M\_Student\_FK FOREIGN KEY (Student\_ID)

REFERENCES STUDENT (Student\_ID)

ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT S\_M\_Mentor\_FK FOREIGN KEY (Mentor\_ID)

REFERENCES MENTOR (Mentor\_ID)

ON DELETE NO ACTION ON UPDATE NO ACTION

);

CREATE TABLE FCTLY\_MNTR (

Faculty\_ID INT NOT NULL,

Mentor\_ID INT NOT NULL,

StartDate DATE NOT NULL,

EndDate DATE NOT NULL,

CONSTRAINT F\_M\_PK PRIMARY KEY (Faculty\_ID , Mentor\_ID),

CONSTRAINT F\_M\_Faculty\_FK FOREIGN KEY (Faculty\_ID)

REFERENCES FACULTY (Faculty\_ID)

ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT F\_M\_Mentor\_FK FOREIGN KEY (Mentor\_ID)

REFERENCES MENTOR (Mentor\_ID)

ON DELETE NO ACTION ON UPDATE NO ACTION

);

***SQL Scripts: Insert Data***

############################ STUDENT ########################################

INSERT INTO STUDENT VALUES(

NULL, 'Doe', 'John', 'John.Doe@students.hu.edu', 'Dorm1', '100',

'1-631-867-5309', '2017-09-05', '2021-05-19', 'Liberal Arts');

INSERT INTO STUDENT VALUES(

NULL, 'Johnson', 'Dina', 'Dina.Johnson@student.hu.edu', 'Dorm2', '251',

'1-631-878-1122', '2015-08-31', '2019-05-21', 'Pre-Med');

INSERT INTO STUDENT VALUES(

NULL, 'Ambrose', 'Dean', 'Dean.Ambrose@students.hu.edu', 'Dorm3', '305',

'1-631-998-5544', '2016-01-25', '2020-05-17', 'Business');

############################################################################

############################ FACULTY #########################################

INSERT INTO FACULTY VALUES(

NULL, 'Smith', 'Jane', 'Jane.Smith@hu.edu', 'Computer Dept', 'Roosevelt',

'340', '1-631-999-5421');

INSERT INTO FACULTY VALUES(

NULL, 'Baker', 'Jill', 'Jill.Baker@hu.edu', 'BIO-Science Dept', 'Jefferson',

'120', '1-631-555-4444');

INSERT INTO FACULTY VALUES(

NULL, 'Jacobs', 'Michael', 'Michael.Jacobs@hu.edu', 'Business Dept', 'Clinton',

'400', '1-631-554-4488');

############################################################################

############################### MENTOR ######################################

INSERT INTO MENTOR VALUES(

NULL, 'Gates', 'William', 'Will.Gates@microsoft.com', 'Microsoft',

'10043', '34TH West Street', 'New York', 'NY', '1-202-555-5555');

INSERT INTO MENTOR VALUES(

NULL, 'Jefferson', 'Ronald', 'Ron.Jefferson@lehmanbros.com', 'Lehman Brothers',

'10011', '37TH South Street', 'New York', 'NY', '1-202-444-4444');

INSERT INTO MENTOR VALUES(

NULL, 'Green', 'Brian', 'Brian.Green@chase.com', 'Chase Manhattan',

'10021', '49TH East Street', 'New York', 'NY', '1-202-888-9911');

############################################################################

############################# ALUMNI ########################################

INSERT INTO ALUMNI VALUES(

NULL, 3000, 'Gates', 'William', 'Will.Gates@microsoft.com', '16 Main Street',

'White Plains', 'NY', '16499', '1-917-554-4455');

INSERT INTO ALUMNI VALUES(

NULL, 3001, 'Jefferson', 'Ronald', 'Ron.Jefferson@lehmanbros.com', '458 Sunset Drive',

'Queens', 'NY', '10234', '1-202-644-6677');

INSERT INTO ALUMNI VALUES(

NULL, 3002, 'Green', 'Brian', 'Brian.Green@chase.com', '87 Habor Lane',

'Brooklyn', 'NY', '10577', '1-914-444-3331');

############################################################################

-- Intersection Tables

############################ ASSIGNMENT ######################################

INSERT INTO ASSIGNMENT VALUES(1000, 2000, '2015-12-06', '2015-12-07');

INSERT INTO ASSIGNMENT VALUES(1001, 2001, '2013-05-16', '2013-05-18');

INSERT INTO ASSIGNMENT VALUES(1002, 2002, '2017-05-10', '2017-05-11');

############################################################################

############################ FCTLY\_MNTR ######################################

INSERT INTO FCTLY\_MNTR VALUES(2000, 3000, '2016-02-07', '2016-2-10');

INSERT INTO FCTLY\_MNTR VALUES(2001, 3001, '2017-02-16', '2017-02-18');

INSERT INTO FCTLY\_MNTR VALUES(2002, 3002, '2015-01-10', '2017-01-11');

############################################################################

############################ STUD\_MNTR ######################################

INSERT INTO STUD\_MNTR VALUES(1000, 3000, '2016-03-04', '2016-03-10');

INSERT INTO STUD\_MNTR VALUES(1001, 3001, '2015-02-08', '2015-02-18');

INSERT INTO STUD\_MNTR VALUES(1002, 3002, '2015-01-05', '2015-01-11');

############################################################################

***Screenshots* *of Our Created Database***

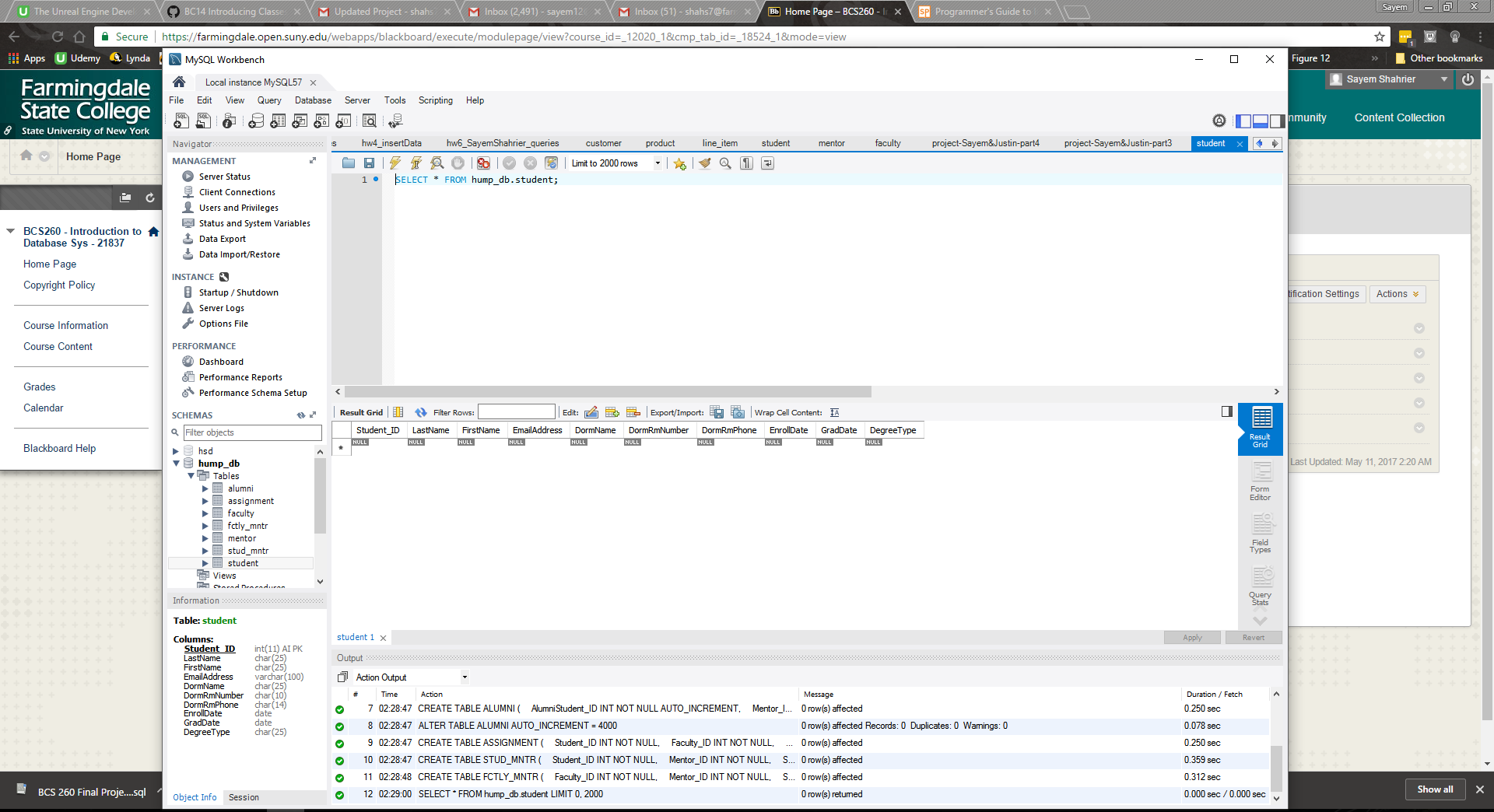


Image Description: Screenshot of Part 3 with STUDENT table and specified columns.

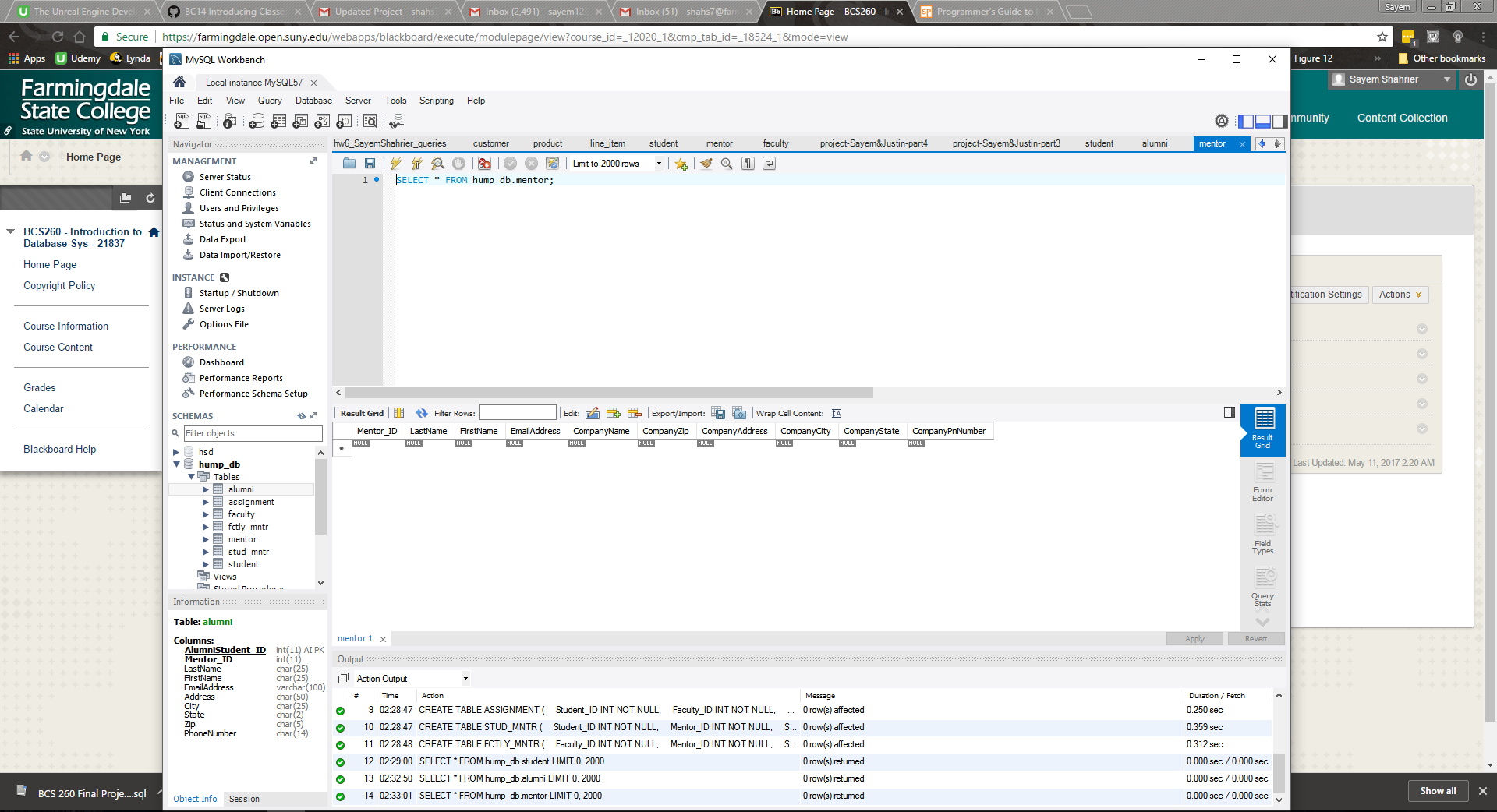
****

Image Description: Screenshot of Part 3 with MENTOR table followed by its columns.

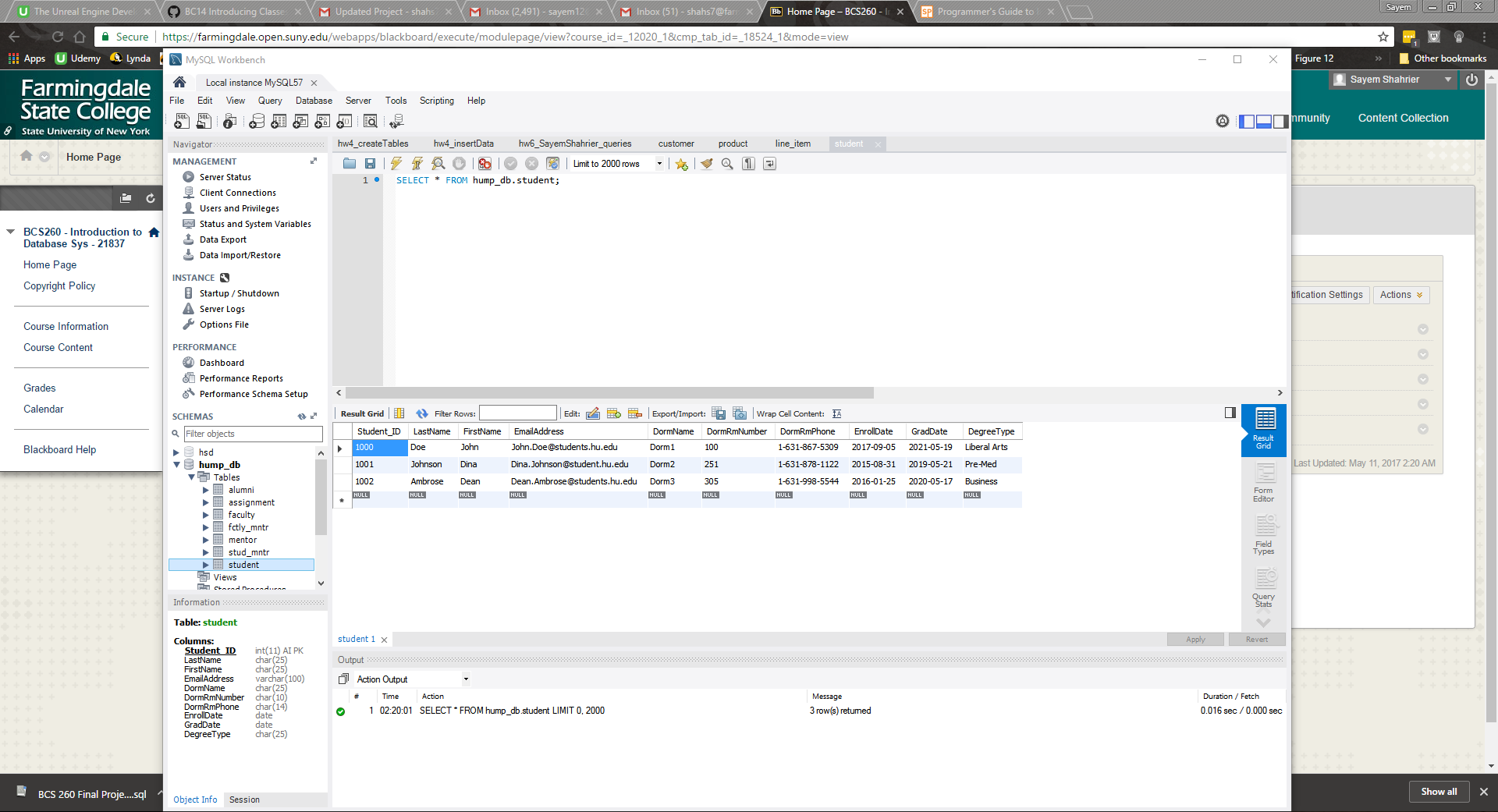


Image Description: Screenshot of Part 4 with STUDENT table and its populated data of columns.

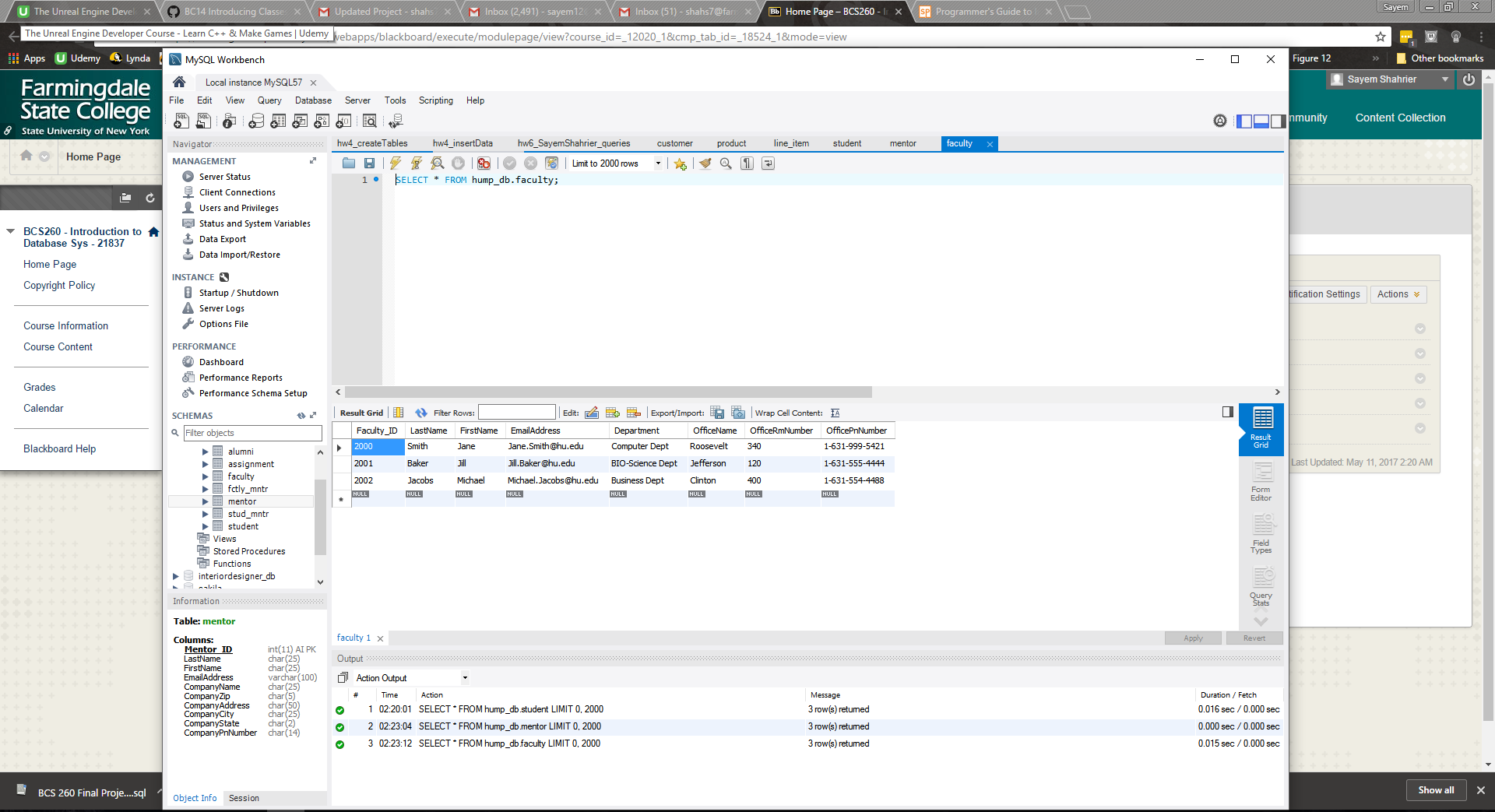


Image Description: Screenshot of Part 4 with FACULTY table and its populated data of columns.