

College Course Management Database  
Logan Howard and Krista Williston  
CSI 3450

## **Abstract**

Our Course Management Database is useful in many different ways. Primarily, our system can be utilized by universities or other institutes to display what courses they offer. Similarly, students can use our database to monitor their GPA, total credit hours taken, how many courses they've taken, how many seats are available per course, information about tutors for their classes, and much more. The main problem our Database System is trying to solve is the occasionally confusing system that many colleges employ. Information is often not straightforward and not complete, whereas our database strives to provide up-to-date accurate information along with additional details that most of our competitors aren't using. Our goal is to make our database as user-friendly as possible, catering towards students as our primary consumer. However, this expensive database can still be beneficial to faculty members, who can easily track down emails of colleagues, information about the heads of departments, and more. Our project gives a large scale view of the School system, from information on Universities to Departments to Professors, Students, and even Tutors.

## **Business Rules**

Writing Business Rules was an initial step used when creating our Course Management Database. These Rules allowed us to show a brief outline of what the relations between the entities should be. Utilizing Business Rules contributes to designing ER Diagrams and Relationship models. They are usually written to show each end of the connectivities in a diagram model. The Business Rules we wrote to build our database include the following:

A University employs many Employees  
Every Employee belong to 1 University  
A University has many Departments  
Every Department belongs to 1 University  
An Employee can be a Head of Department, a Professor, or both  
A Department has 1 Head of Department  
A Head of Department belongs to 1 Department  
A Professor teaches many Courses  
Many Courses are taught by 1 Professor  
Many Tutors are employed as Employee  
An Employee can be many types of Tutors  
A Tutor can Mentor many times  
Many Mentors sessions come from 1 Tutor  
A Course can be Mentored by many different employees

Many students are Mentored for a Course  
A Course is Registered For many times  
Many Registrations are for 1 Course  
A Student Registers many times  
Many Registrations are for 1 Student

## **Entities**

Entities were used to show the big ideas that are included in a database. They are the overall topics that an attribute describes. When looking for a specific attribute, relating it back to an entity makes the attribute easier to find in a database or data directory. Instead of looking for something specific, one can look under a general topic and narrow it down from there. When creating entities, entity integration needs to be considered to ensure each primary key is unique and not null. The entities mentioned in our database include the following:

University  
Department  
Employee  
Head\_of\_Dept  
Professor  
Tutor  
Mentored  
Course  
Registered  
Student

## **Attributes**

Attributes describe entities. They are what people are looking for when they use a database. They are the components of an entity that a database can hold information about. Our Course Management Database assigns numbers to Universities, Departments, Employees like Professors and Tutors, Students, and the Courses they take. Other attributes include descriptions about Employee and Student names and contact information. The attributes that describe the entities in our database include the following:

*University*  
PK: UNI\_NUM  
FK1: COURSE\_NUM

*Department*

PK: DEPT\_NUM  
FK1: EMP\_NUM  
FK2: UNI\_NUM

*Employee*

PK: EMP\_NUM  
FK1: UNI\_NUM  
EMP\_LAME  
EMP\_FNAME  
EMP\_INITIAL

*Head of Department*

PK: EMP\_NUM  
FK1: COURSE\_NUM  
FK2: DEPT\_NUM  
FK3: UNI\_NUM  
HEAD\_PHONE  
HEAD\_EMAIL

*Professor*

PK: EMP\_NUM  
PK, FK1: PROF\_NUM  
FK2: COURSE\_NUM  
FK3: DEPT\_NUM  
FK4: UNI\_NUM  
PROF\_RANK  
PROF\_PHONE  
PROF\_EMAIL

*Tutor*

PK: EMP\_NUM  
PK, FK1: TUTOR\_NUM  
FK2: COURSE\_NUM  
FK3: UNI\_NUM  
FK4: STU\_NUM

*Mentored*

PK, FK1: TUTOR\_NUM  
PK, FK2: COURSE\_NUM

### *Course*

PK: COURSE\_NUM  
FK1: PROF\_NUM  
FK2: STU\_NUM  
FK3: TUTOR\_NUM  
FK4: UNI\_NUM  
COURSE\_DESCRIPT  
SEATS\_AVL  
WAIT\_NUM

### *Registered*

PK, FK1: STU\_NUM  
PK, FK2: COURSE\_NUM

### *Student*

PK: STU\_NUM  
FK1: COURSE\_NUM  
FK2: UNI\_NUM  
STU\_LNAME  
STU\_FNAME  
STU\_INITIAL  
STU\_EMAIL  
STU\_GPA  
CRED\_HRS  
TOT\_COURSES

## **Relationships**

The relationships between entities are similar to that of business rules. They link entities together to show what kind of relationship is between them. These relationships are then used in ER Diagrams and Relationship Models to show what kind of lines and line endings should be used. Relationships can be optional or mandatory and can include 0, 1, or many relations. For example, HEAD\_OF\_DEPARTMENT and PROFESSOR are overlapping subtypes. Since HEAD\_OF\_DEPARTMENT and PROFESSOR are considered as subtypes for EMPLOYEE, the specialization hierarchy reflects a 1:1 relationship between EMPLOYEE and its subtypes. It can be described as “is-a” relationships. Once relationships are established, normalization needs to be considered. Reducing down to the third normal form reduces data duplication and anomalies, and ensures referential integrity. The Relationships used in our database include the following:

UNIVERSITY and DEPARTMENT relationship is (1:M)  
UNIVERSITY and EMPLOYEE relationship is (1:M)  
DEPARTMENT and HEAD\_OF\_DEPARTMENT relationship is (1:1)  
EMPLOYEE and HEAD\_OF\_DEPARTMENT relationship is (1:1)  
EMPLOYEE and PROFESSOR relationship is (1:1)  
EMPLOYEE and TUTOR relationship is (0:M)  
TUTOR and MENTORED relationship is (1:M)  
COURSE and MENTORED relationship is (1:M)  
PROFESSOR and COURSE relationship is (1:M)  
COURSE and REGISTERED\_FOR relationship is (1:M)  
STUDENT and REGISTERED\_FOR relationship is (1:M)

## **Data Directory**

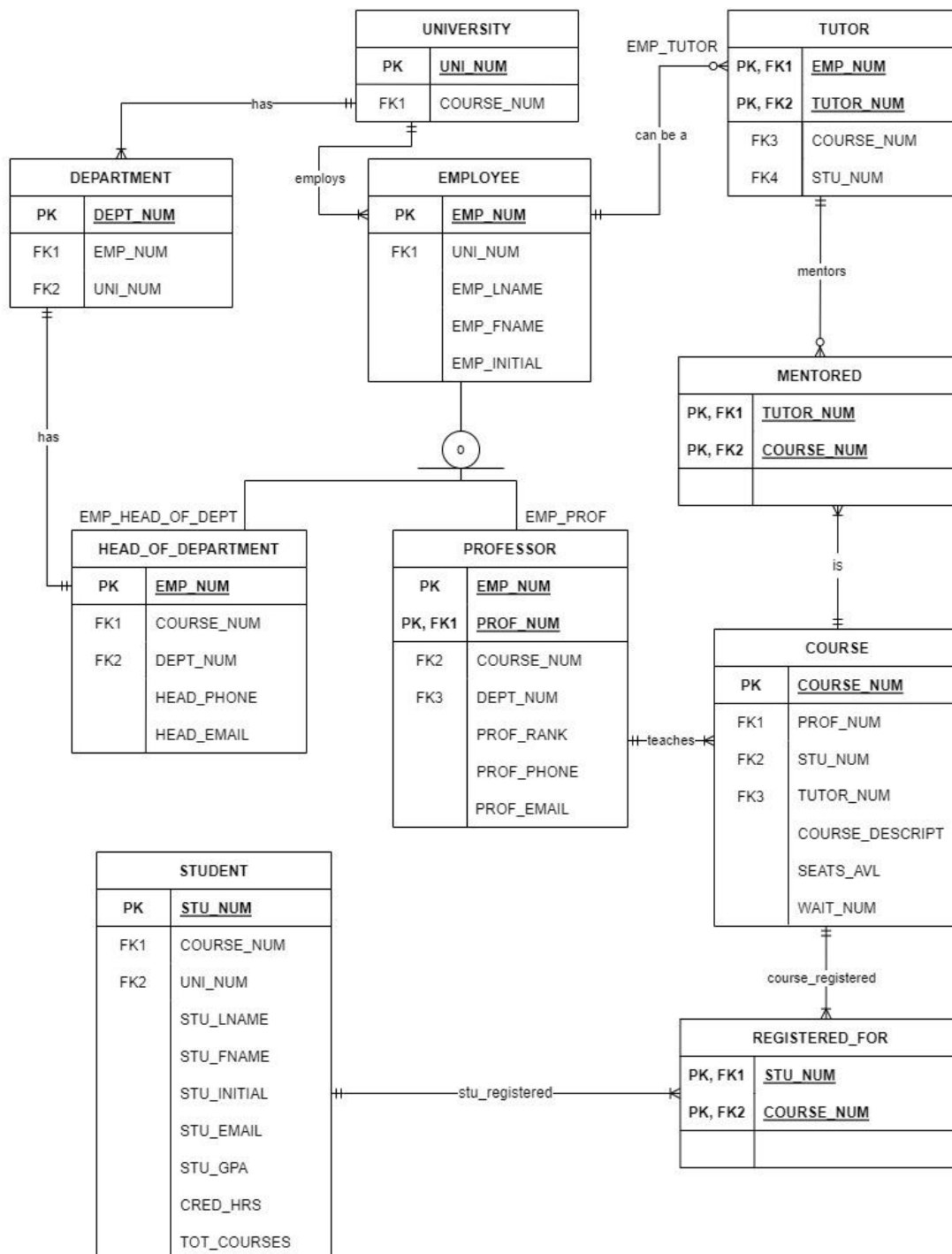
Our data directory provides a brief overview of the database and the information it contains. It is an inventory that specifies the source, location, ownership, usage, and destination of all of the data elements that are stored in a database. This makes it easy to find entities and their attributes.

Table Name	Attribute Name	Contents	Type	Format	Range	Required	PK or FK	FK Referenced Table
University	UNI_NUM	University Number	NUMBER (5)	99999		Y	PK	
	COURSE_NUM	Course Number	NUMBER (4)	9999	1000 - 9999		FK	Course
Department	DEPT_NUM	Department Number	NUMBER (3)	999	100 - 999	Y	PK	
	PROF_NUM	Professor Number	NUMBER (3)	999			FK	Professor
Employee	UNI_NUM	University Number	NUMBER (5)	99999			FK	University
	EMP_NUM	Employee Number	NUMBER (8)	99999999		Y	PK	
	UNI_NUM	University Number	NUMBER (5)	99999			FK	
	EMP_LNAME	Employee Last Name	VARCHAR (20)	Xxxxxxxx		Y		
	EMP_FNAME	Employee First Name	VARCHAR (20)	Xxxxxxxx		Y		
Head_of_Dept	EMP_INITIAL	Employee Initial	CHAR (1)	X				
	EMP_NUM	Employee Number	NUMBER (8)	99999999		Y		
	DEPT_NUM	Department Number	NUMBER (3)	999	100 - 999	Y	FK	Department
	COURSE_NUM	Course Number	NUMBER (4)	9999	1000 - 9999	Y	FK	Course
	HEAD_PHONE	Head of Department Phone	CHAR (8)	999-9999				
Professor	HEAD_EMAIL	Head of Department Email	VARCHAR (25)	xxx@xxx.edu				
	PROF_NUM	Professor Number	NUMBER (3)	999		Y	PK	
	EMP_NUM	Employee Number	NUMBER (8)	99999999		Y	PK	
	COURSE_NUM	Course Number	NUMBER (4)	9999	1000 - 9999	Y	FK	Course
	DEPT_NUM	Department Number	NUMBER (3)	999	100 - 999	Y	FK	Department
	PROF_RANK	Professor Rank	VARCHAR (20)	Xxxxx Xxxx				
Tutor	PROF_PHONE	Professor Phone	CHAR (8)	999-9999				
	PROF_EMAIL	Professor Email	VARCHAR (25)	xxx@xxx.edu				
	EMP_NUM	Employee Number	NUMBER (8)	99999999		Y		
	TUTOR_NUM	Tutor Number	NUMBER (3)	999		Y	PK	
Mentored	COURSE_NUM	Course Number	NUMBER (4)	9999	1000 - 9999	Y	FK	Course
	STU_NUM	Student Number	NUMBER (9)	xxxxxxxxx			FK	Student
Course	TUTOR_NUM	Tutor Number	NUMBER (3)	999			FK	Tutor
	COURSE_NUM	Course Number	NUMBER (4)	9999	1000 - 9999	Y	FK	Course
	COURSE_DESCRIPT	Course Description	VARCHAR (100)	Xxxx. Xxx.		Y		
	SEATS_AVL	Seats Available	NUMBER (3)	999				
	WAIT_NUM	Wait List Number	NUMBER (3)	999				
	STU_NUM + COURSE_NUM	Student Number + Course Number				Y	PK	
Student	STU_NUM	Student Number	NUMBER (9)	xxxxxxxxx		Y	PK	
	COURSE_NUM	Course Number	NUMBER (4)	9999	1000 - 9999	Y	FK	Course
	UNI_NUM	University Number	NUMBER (5)	99999			FK	University
	STU_LNAME	Student Last Name	VARCHAR (20)	Xxxxxxxx		Y		
	STU_FNAME	Student First Name	VARCHAR (20)	Xxxxxxxx		Y		
	STU_INITIAL	Student Initial	CHAR (1)	X				
	STU_EMAIL	Student Email	VARCHAR (25)	xxx@xxx.edu				
	STU_GPA	Student Grade Point Average	DECIMAL (3, 2)	9.99	0.00 - 5.00			
Registered	CRED_HRS	Total Credit Hours	NUMBER (5)	99.99				
	TOT_COURSES	Total Courses Taken	NUMBER (2)	99				

FK	Foreign Key
PK	Primary Key
CHAR	Fixed Character Length Data (1 - 255 characters)
VARCHAR	Variable Character Length Data (1 - 2,000 characters)
NUMBER	Numeric Data
DECIMAL	Decimal Number Data

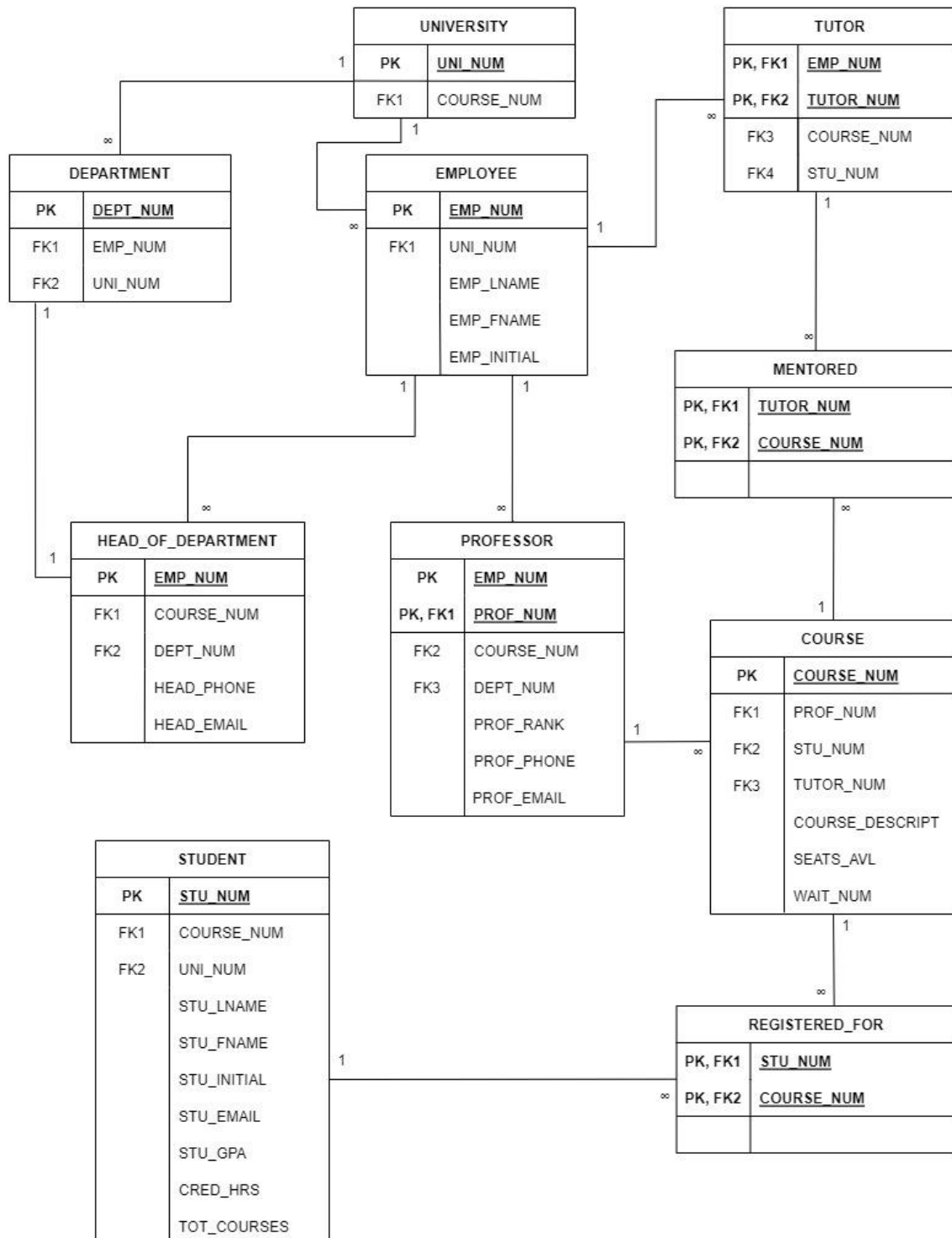
## Entity Relationship Model

An entity relationship model brings together business rules, entities, attributes, and their relationships with one another. This is a visual model of what the database covers. Below is the ER Diagram of our Course Management Database. Most relationships are 1:M, but some are different. The relationship between DEPARTMENT and HEAD\_OF\_DEPARTMENT shows a 1:1 relationship meaning there needs to be only one HEAD\_OF\_DEPARTMENT per DEPARTMENT. The relationship between EMPLOYEE and TUTOR is 1:M optional, meaning there has to be at least one but there could be many. HEAD\_OF\_DEPARTMENT and PROFESSOR are overlapping subtypes of EMPLOYEE so they show a 1:1 relationship with EMPLOYEE. STUDENT and REGISTERED\_FOR are also 1:M relationships. COURSE and REGISTERED\_FOR are 1:M relationships. TUTOR and MENTORED are 1:M relationships. DEPARTMENT and UNIVERSITY are 1:M relationships. EMPLOYEE and UNIVERSITY are 1:M relationships. COURSE and PROFESSOR are 1:M relationships. COURSE and STUDENT are 1:M relationships. COURSE and REGISTERED\_FOR are 1:M relationships. TUTOR and MENTORED are 1:M relationships. DEPARTMENT and HEAD\_OF\_DEPARTMENT are 1:1 relationships. EMPLOYEE and TUTOR are 1:M optional relationships. HEAD\_OF\_DEPARTMENT and PROFESSOR are overlapping subtypes of EMPLOYEE so they show a 1:1 relationship with EMPLOYEE.



## Relational Database Model

A relationship database model is another way to show business rules, entities, attributes, and their relationships. Similar to an Entity Relationship Model, it is another type of visual model of the database. The 1 and  $\infty$  symbols on separate ends of the same connective lines represent 1:M relationships while the 1 and 1 symbols represent 1:1 relationships. There is no overlapping relationship between PROFESSOR and HEAD\_OF\_DEPARTMENT because Relational Database Models are supposed to be simpler versions of ERD's. Below is the Relational Database Model of our Course Management Database.





## Implementation of Database Using Microsoft SQL Server

Using what we learned about designing databases throughout the semester, we implemented our database and its components into the Microsoft SQL Server. The SQL Server allowed us to create tables and their attributes, as well as assign values to them, to show an example of what the database could look like with real data. There are also example queries included of what it would look like if we were looking for specific data. Below is the implementation of our design process to create our Course Management Database.

/\*Create Tables\*/

```
CREATE TABLE COURSE (  
    COURSE_NUM NUMERIC(4),  
    PROF_NUM NUMERIC(3),  
    STU_NUM NUMERIC(9),  
    TUTOR_NUM NUMERIC(3),  
    COURSE_DESCRIPT VARCHAR(100),  
    SEATS_AVL NUMERIC(3),  
    WAIT_NUM NUMERIC(3),  
    PRIMARY KEY(COURSE_NUM));  
  
CREATE TABLE UNIVERSITY(  
    UNI_NUM NUMERIC(5),  
    COURSE_NUM NUMERIC(4),  
    PRIMARY KEY(UNI_NUM),  
    FOREIGN KEY(COURSE_NUM) REFERENCES COURSE);  
  
CREATE TABLE EMPLOYEE(  
    EMP_NUM NUMERIC(8),  
    UNI_NUM NUMERIC(5),  
    EMP_LNAME VARCHAR(20),  
    EMP_FNAME VARCHAR(20),  
    EMP_INITIAL CHAR(1),  
    PRIMARY KEY(EMP_NUM),  
    FOREIGN KEY(UNI_NUM) REFERENCES UNIVERSITY);  
  
CREATE TABLE DEPARTMENT(  
    DEPT_NUM NUMERIC(3),  
    PROF_NUM NUMERIC(3),  
    UNI_NUM NUMERIC(5),  
    PRIMARY KEY(DEPT_NUM),  
    FOREIGN KEY(UNI_NUM) REFERENCES UNIVERSITY);
```

```
CREATE TABLE PROFESSOR(  
    PROF_NUM NUMERIC(3),  
    EMP_NUM NUMERIC(8),  
    COURSE_NUM NUMERIC(4),  
    DEPT_NUM NUMERIC(3),  
    PROF_RANK VARCHAR(20),  
    PROF_PHONE CHAR(8),  
    PROF_EMAIL VARCHAR(25),  
PRIMARY KEY (PROF_NUM),  
FOREIGN KEY (EMP_NUM) REFERENCES EMPLOYEE(EMP_NUM),  
FOREIGN KEY (COURSE_NUM) REFERENCES COURSE,  
FOREIGN KEY (DEPT_NUM) REFERENCES DEPARTMENT);
```

```
CREATE TABLE HEAD_OF_DEPT(  
    EMP_NUM NUMERIC(8),  
    DEPT_NUM NUMERIC(3),  
    COURSE_NUM NUMERIC(4),  
    HEAD_PHONE CHAR(8),  
    HEAD_EMAIL VARCHAR(25),  
FOREIGN KEY (DEPT_NUM) REFERENCES DEPARTMENT,  
FOREIGN KEY (COURSE_NUM) REFERENCES COURSE);
```

```
CREATE TABLE STUDENT (  
    STU_NUM NUMERIC(9),  
    COURSE_NUM NUMERIC(4),  
    UNI_NUM NUMERIC(5),  
    STU_LNAME VARCHAR(20),  
    STU_FNAME VARCHAR(20),  
    STU_INITIAL CHAR(1),  
    STU_EMAIL VARCHAR(25),  
    STU_GPA DECIMAL(3, 2),  
    CRED_HOURS NUMERIC(5),  
    TOT_COURSES NUMERIC(2),  
PRIMARY KEY (STU_NUM),  
FOREIGN KEY (COURSE_NUM) REFERENCES COURSE,  
FOREIGN KEY (UNI_NUM) REFERENCES UNIVERSITY);
```

```
CREATE TABLE TUTOR (  
    EMP_NUM NUMERIC(8),
```

```
TUTOR_NUM NUMERIC(3),
COURSE_NUM NUMERIC(4),
STU_NUM NUMERIC(9),
PRIMARY KEY(EMP_NUM, TUTOR_NUM),
FOREIGN KEY(COURSE_NUM) REFERENCES COURSE,
FOREIGN KEY(STU_NUM) REFERENCES STUDENT);
```

```
CREATE TABLE REGISTERED_FOR (
    STU_NUM NUMERIC(9),
    COURSE_NUM NUMERIC(4),
    PRIMARY KEY(STU_NUM, COURSE_NUM),
    FOREIGN KEY(STU_NUM) REFERENCES STUDENT,
    FOREIGN KEY(COURSE_NUM) REFERENCES COURSE);
```

```
CREATE TABLE MENTORED (
    TUTOR_NUM NUMERIC(3),
    COURSE_NUM NUMERIC (4),
    PRIMARY KEY(TUTOR_NUM, COURSE_NUM),
    FOREIGN KEY(COURSE_NUM) REFERENCES COURSE);
```

/\*Data Rows\*/

```
INSERT INTO UNIVERSITY VALUES(12121, 4350);
INSERT INTO UNIVERSITY VALUES(31415, 1170);
INSERT INTO UNIVERSITY VALUES(10340, 4500);
```

```
INSERT INTO DEPARTMENT VALUES(100, 535, 10340);
INSERT INTO DEPARTMENT VALUES(314, 135, 31415);
INSERT INTO DEPARTMENT VALUES(111, 998, 12121);
```

```
INSERT INTO EMPLOYEE VALUES(12309753, 10340, 'Cunningham', 'Cade', 'P');
INSERT INTO EMPLOYEE VALUES(31415926, 31415, 'Patel', 'Pi', 'F');
INSERT INTO EMPLOYEE VALUES(48201450, 12121, 'Baddoo', 'Akil', 'N');
```

```
INSERT INTO HEAD_OF_DEPT VALUES(48201450, 119, 4500, 354-1189, 'abdet@ou.edu');
INSERT INTO HEAD_OF_DEPT VALUES(12309753, 458, 2370, 482-0133,
'ccchamp@det.edu');
INSERT INTO HEAD_OF_DEPT VALUES(12121210, 120, 4480, 012-2101, 'bin@oak.edu');
```

```
INSERT INTO PROFESSOR VALUES(535, 31415926, 1030, 10340, 'Associate Professor',
132-9867, 'lifeop@sdsu.edu');
```

INSERT INTO PROFESSOR VALUES(144, 14320067, 3370, 11011, 'Professor', 681-3370, 'osman@casewest.edu');

INSERT INTO PROFESSOR VALUES(109, 00339917, 1080, 40011, 'Assistant Professor', 033-7485, 'overit@duke.edu');

INSERT INTO PROFESSOR VALUES(333, 12309753, 2370, 458, 'Associate Professor', 482-0133, 'ccchamp@det.edu');

INSERT INTO PROFESSOR VALUES(225, 48201450, 4500, 119, 'Associate Professor', 384-1189, 'abdet@ou.edu');

INSERT INTO COURSE VALUES(3540, 535, 001243587, 103, 'Electronics', 10, 2);

INSERT INTO COURSE VALUES(1600, 144, 004556890, 211, 'Physics', 20, 25);

INSERT INTO COURSE VALUES(2663, 109, 003752893, 458, 'Math', 5, 5);

INSERT INTO COURSE VALUES(2370, 333, 000000000, 000, 'Chemistry', 2, 1);

INSERT INTO COURSE VALUES(4500, 225, 000000000, 000, 'History', 3, 8);

INSERT INTO STUDENT VALUES(001243587, 3450, 12121, 'Jordan', 'Poole', 'A', 'japoole@gsw.edu', 3.33, 100, 30);

INSERT INTO STUDENT VALUES(004556890, 1600, 31415, 'Breanna', 'Stewart', 'M', 'brestewart@sstorm.edu', 2.5, 31, 7);

INSERT INTO STUDENT VALUES(003752893, 2663, 10340, 'Swin', 'Cash', 'M', 'swincash@detshock.edu', 3.0, 73, 18);

INSERT INTO STUDENT VALUES(005334801, 3450, 10340, 'Doe', 'Jane', 'G', 'jdoe@detshock.edu', 3.7, 95, 28);

INSERT INTO STUDENT VALUES(002238974, 4500, 31415, 'Joe', 'Dane', 'L', 'sstorm.edu', 3.6, 22, 4);

INSERT INTO TUTOR VALUES(39482010, 103, 3450, 008752117);

INSERT INTO TUTOR VALUES(48201450, 211, 1600, 003498216);

INSERT INTO TUTOR VALUES(53869235, 458, 2663, 003984019);

INSERT INTO REGISTERED\_FOR VALUES(001243587, 3450);

INSERT INTO REGISTERED\_FOR VALUES(004556890, 1600);

INSERT INTO REGISTERED\_FOR VALUES(003752893, 2663);

INSERT INTO REGISTERED\_FOR VALUES(005334801, 2370);

INSERT INTO REGISTERED\_FOR VALUES(002238974, 4500);

INSERT INTO MENTORED VALUES(103, 3450);

INSERT INTO MENTORED VALUES(211, 1600);

INSERT INTO MENTORED VALUES(458, 2663);

Queries:

1 - This query selects different attributes from the STUDENT table for students with a GPA between 3.0 and 4.0.

```
SELECT STU_LNAME, STU_FNAME, STU_INITIAL, STU_EMAIL, STU_GPA,  
CRED_HOURS, TOT_COURSES  
FROM STUDENT WHERE STU_GPA BETWEEN 3.0 AND 4.0;
```

	STU_LNA...	STU_FNA...	STU_INITI...	STU_EMAIL	STU_G...	CRED_HOU...	TOT_COURS...
1	Jordan	Poole	A	japoole@gsw.edu	3.33	100	30
2	Swin	Cash	M	swincash@detshock.edu	3.00	73	18

2 - This query selects different attributes from the COURSE table for courses where there are more than eight seats available for registration.

```
SELECT COURSE_NUM, COURSE_DESCRIPT, SEATS_AVL, WAIT_NUM  
FROM COURSE WHERE SEATS_AVL > 8;
```

	COURSE_NUM	COURSE_DESCRI...	SEATS_A...	WAIT_NUM
1	1600	Physics	20	25
2	3540	Electronics	10	2

3 - This query selects a specific employee from the EMPLOYEE table using the employee's first name.

```
SELECT * FROM EMPLOYEE WHERE EMP_FNAME = 'Cade';
```

	EMP_NUM	UNI_NUM	EMP_LNA...	EMP_FNAME	EMP_INITIAL
1	12309753	10340	Cunningham	Cade	P

4 - This query averages student's GPAs, then orders them by course number

```
SELECT COURSE_NUM, AVG(STU_GPA) AS AVG_GPA  
FROM STUDENT  
GROUP BY COURSE_NUM;
```

	COURSE_NUM	AVG_GPA
1	1600	2.500000
2	2663	3.000000
3	3450	3.330000

Other Queries used to print every Table's data points:

```

SELECT * FROM UNIVERSITY;
SELECT * FROM DEPARTMENT;
SELECT * FROM EMPLOYEE;
SELECT * FROM HEAD_OF_DEPT;
SELECT * FROM PROFESSOR;
SELECT * FROM COURSE;
SELECT * FROM STUDENT;
SELECT * FROM TUTOR;
SELECT * FROM REGISTERED_FOR;
SELECT * FROM MENTORED;

```

Output for all Tables and their information (using the compiler.io):

### Output

```

12121|4350
31415|1170
10340|4500
100|535|10340
314|135|31415
111|998|12121
12309753|10340|Cunningham|Cade|P
31415926|31415|Patel|Pi|F
48201450|12121|Baddoo|Akil|N
48201450|119|4500|354-1189|abdet@ou.edu
12309753|458|2370|482-0133|ccchamp@det.edu
12121210|120|4480|012-2101|bin@oak.edu
535|31415926|1030|10340|Associate Professor|132-986
7|lifeop@sdsu.edu
144|14320067|3370|11011|Professor|681-3370|osman@ca
sewest.edu
109|339917|1080|40011|Assistant Professor|033-7485|
overit@duke.edu

```

```
109|339917|1080|40011|Assistant Professor|033-7485|
overit@duke.edu
3540|535|1243587|103|Electronics|10|2
1600|144|4556890|211|Physics|20|25
2663|109|3752893|458|Math|5|5
1243587|3450|12121|Jordan|Poole|A|japoole@gsw.edu|
3.33|100|30
4556890|1600|31415|Breanna|Stewart|M|brestewart@sst
orm.edu|2.5|31|7
3752893|2663|10340|Swin|Cash|M|swincash@detshock.ed
u|3|73|18
39482010|103|3450|8752117
48201450|211|1600|3498216
53869235|458|2663|3984019
1243587|3450
4556890|1600
3752893|2663
103|3450
```

```
103|3450
211|1600
458|2663
```

```
[Execution complete with exit code 0]
```