College Course Management Database

Logan Howard and Krista Williston
Group 3, 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 extensive 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, Departments, Professors, Students, and even Tutors.

Business Rules

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 an Employee

An Employee can be many types of Tutors

A Tutor can Mentor many times

Many Mentor sessions come from 1 Tutor

A Course can be Mentored for by a Tutor

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 & Attributes

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_LNAME
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

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

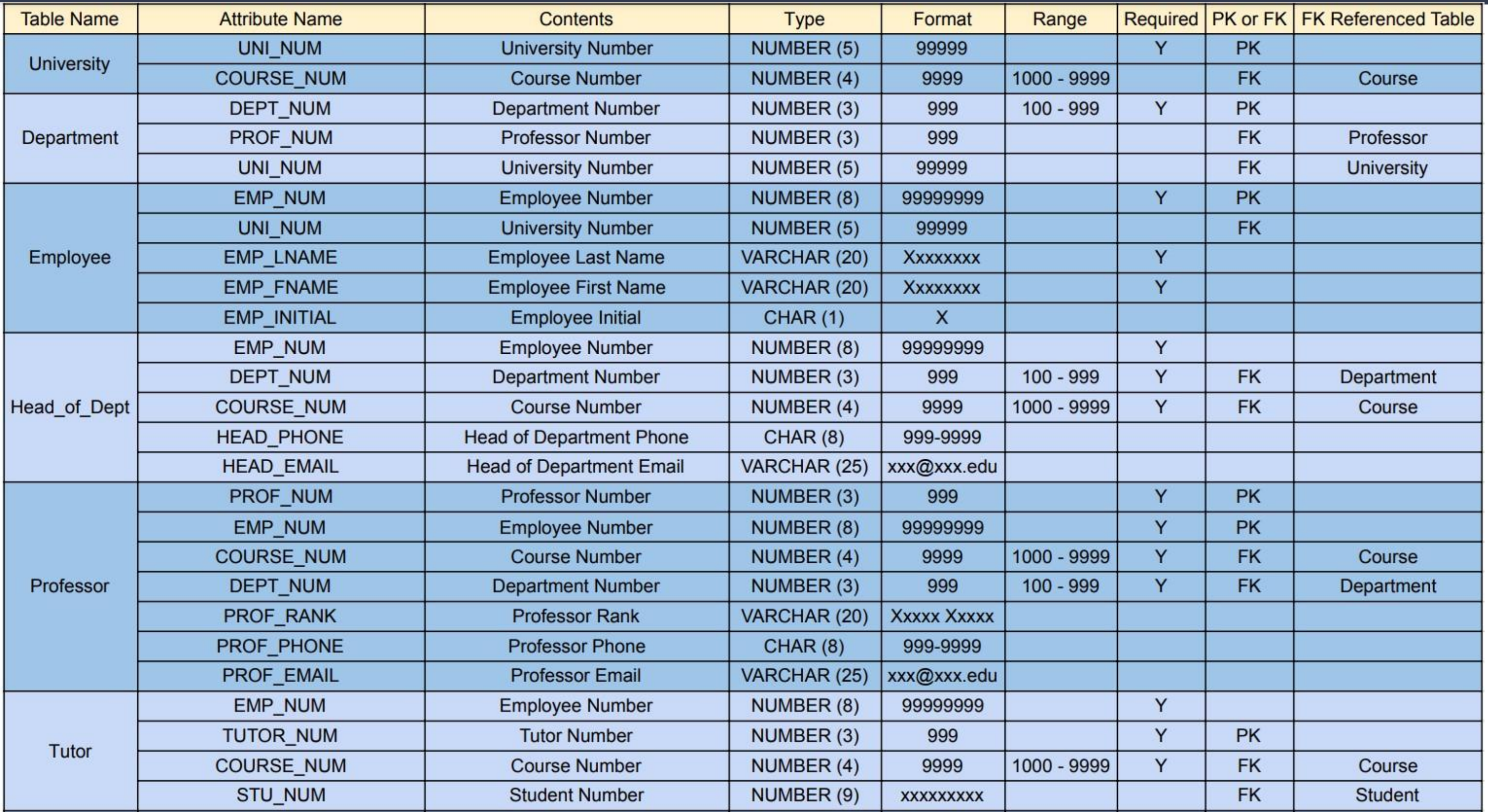
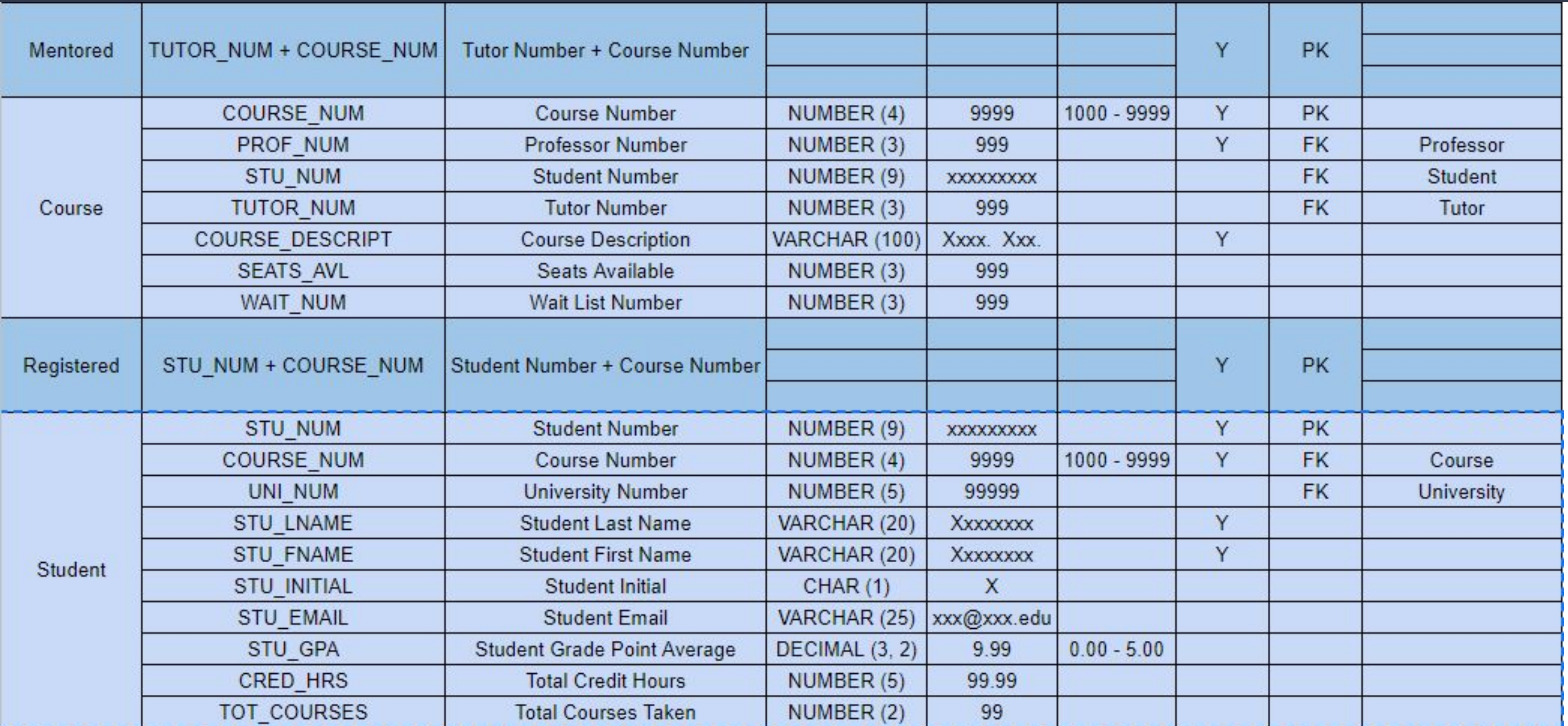


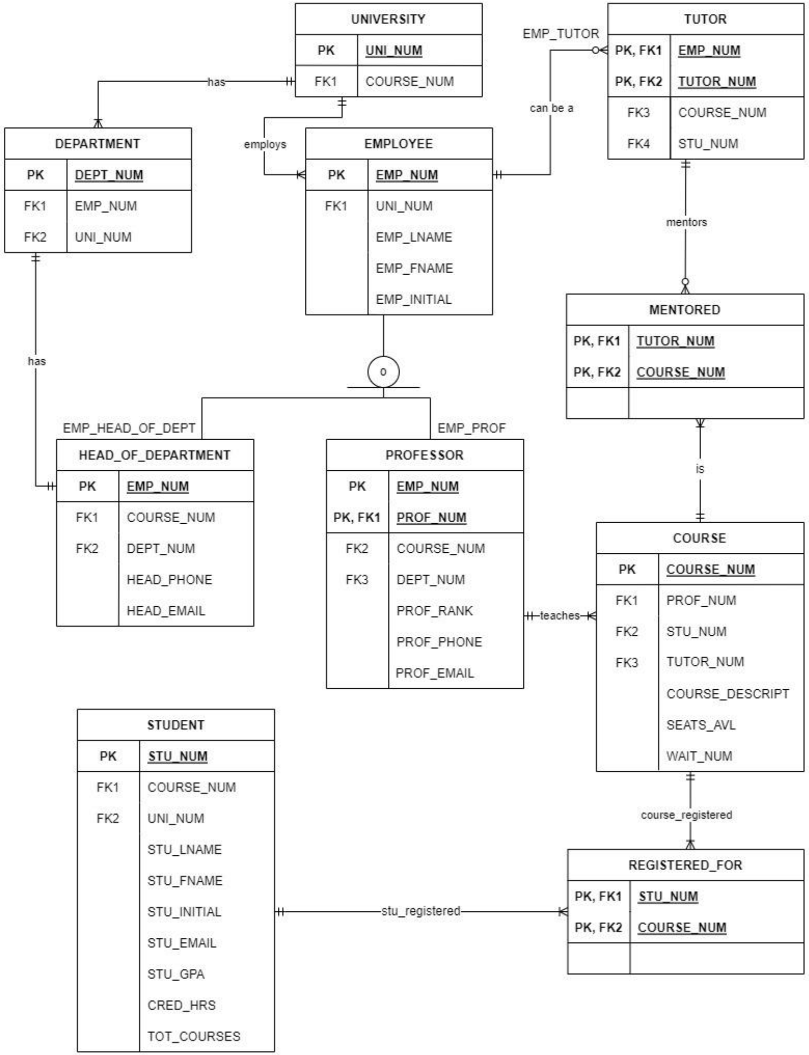
Table Name	Attribute Name	Contents	Type	Format	Range	Required	PK or FK	FK Referenced Table
University	UNI_NUM	University Number	NUMBER (5)	99999		Y	PK	
	COURSE_NUM	Course Number	NUMBER (4)	9999	1000 - 9999		FK	Course
Department	DEPT_NUM	Department Number	NUMBER (3)	999	100 - 999	Y	PK	
	PROF_NUM	Professor Number	NUMBER (3)	999			FK	Professor
	UNI_NUM	University Number	NUMBER (5)	99999			FK	University
Employee	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		
	EMP_INITIAL	Employee Initial	CHAR (1)	X				
Head_of_Dept	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				
	HEAD_EMAIL	Head of Department Email	VARCHAR (25)	xxx@xxx.edu				
Professor	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 Xxxxx				
	PROF_PHONE	Professor Phone	CHAR (8)	999-9999				
	PROF_EMAIL	Professor Email	VARCHAR (25)	xxx@xxx.edu				
Tutor	EMP_NUM	Employee Number	NUMBER (8)	99999999		Y		
	TUTOR_NUM	Tutor Number	NUMBER (3)	999		Y	PK	
	COURSE_NUM	Course Number	NUMBER (4)	9999	1000 - 9999	Y	FK	Course
	STU_NUM	Student Number	NUMBER (9)	xxxxxxxxx			FK	Student

Data Directory cont.



Mentored	TUTOR_NUM + COURSE_NUM	Tutor Number + Course Number				Y	PK	
Course	COURSE_NUM	Course Number	NUMBER (4)	9999	1000 - 9999	Y	PK	
	PROF_NUM	Professor Number	NUMBER (3)	999		Y	FK	Professor
	STU_NUM	Student Number	NUMBER (9)	xxxxxxxxx			FK	Student
	TUTOR_NUM	Tutor Number	NUMBER (3)	999			FK	Tutor
	COURSE_DESCRIPT	Course Description	VARCHAR (100)	Xxxx. Xxx.		Y		
	SEATS_AVL	Seats Available	NUMBER (3)	999				
	WAIT_NUM	Wait List Number	NUMBER (3)	999				
Registered	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)	Xxxxxxx		Y		
	STU_FNAME	Student First Name	VARCHAR (20)	Xxxxxxx		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			
	CRED_HRS	Total Credit Hours	NUMBER (5)	99.99				
	TOT_COURSES	Total Courses Taken	NUMBER (2)	99				

Entity Relationship Model



UNIVERSITY

PK	UNI_NUM
FK1	COURSE_NUM

TUTOR

PK, FK1	EMP_NUM
PK, FK2	TUTOR_NUM
FK3	COURSE_NUM
FK4	STU_NUM

DEPARTMENT

PK	DEPT_NUM
FK1	EMP_NUM
FK2	UNI_NUM

EMPLOYEE

PK	EMP_NUM
FK1	UNI_NUM
	EMP_LNAME
	EMP_FNAME
	EMP_INITIAL

MENTORED

PK, FK1	TUTOR_NUM
PK, FK2	COURSE_NUM

HEAD_OF_DEPARTMENT

PK	EMP_NUM
FK1	COURSE_NUM
FK2	DEPT_NUM
	HEAD_PHONE
	HEAD_EMAIL

PROFESSOR

PK	EMP_NUM
PK, FK1	PROF_NUM
FK2	COURSE_NUM
FK3	DEPT_NUM
	PROF_RANK
	PROF_PHONE
	PROF_EMAIL

COURSE

PK	COURSE_NUM
FK1	PROF_NUM
FK2	STU_NUM
FK3	TUTOR_NUM
	COURSE_DESCRIPT
	SEATS_AVL
	WAIT_NUM

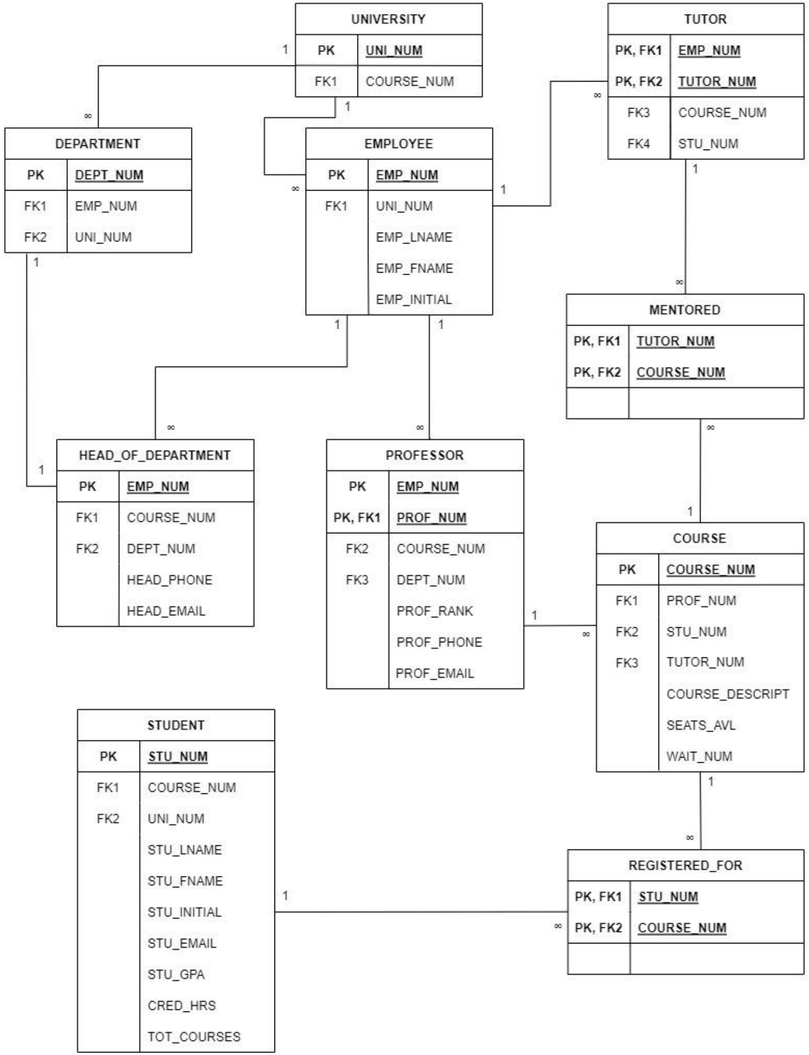
STUDENT

PK	STU_NUM
FK1	COURSE_NUM
FK2	UNI_NUM
	STU_LNAME
	STU_FNAME
	STU_INITIAL
	STU_EMAIL
	STU_GPA
	CRED_HRS
	TOT_COURSES

REGISTERED_FOR

PK, FK1	STU_NUM
PK, FK2	COURSE_NUM

Relational Database Model



UNIVERSITY	
PK	UNI_NUM
FK1	COURSE_NUM

TUTOR	
PK, FK1	EMP_NUM
PK, FK2	TUTOR_NUM
FK3	COURSE_NUM
FK4	STU_NUM

DEPARTMENT	
PK	DEPT_NUM
FK1	EMP_NUM
FK2	UNI_NUM

EMPLOYEE	
PK	EMP_NUM
FK1	UNI_NUM
	EMP_LNAME
	EMP_FNAME
	EMP_INITIAL

MENTORED	
PK, FK1	TUTOR_NUM
PK, FK2	COURSE_NUM

HEAD_OF_DEPARTMENT	
PK	EMP_NUM
FK1	COURSE_NUM
FK2	DEPT_NUM
	HEAD_PHONE
	HEAD_EMAIL

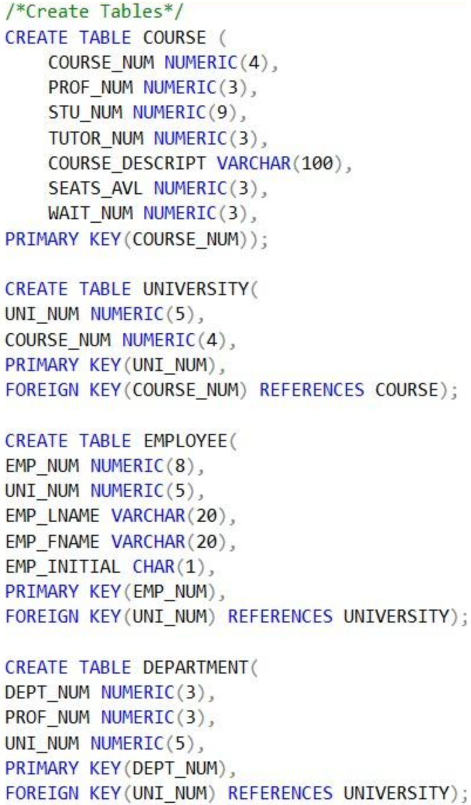
PROFESSOR	
PK	EMP_NUM
PK, FK1	PROF_NUM
FK2	COURSE_NUM
FK3	DEPT_NUM
	PROF_RANK
	PROF_PHONE
	PROF_EMAIL

COURSE	
PK	COURSE_NUM
FK1	PROF_NUM
FK2	STU_NUM
FK3	TUTOR_NUM
	COURSE_DESCRIPT
	SEATS_AVL
	WAIT_NUM

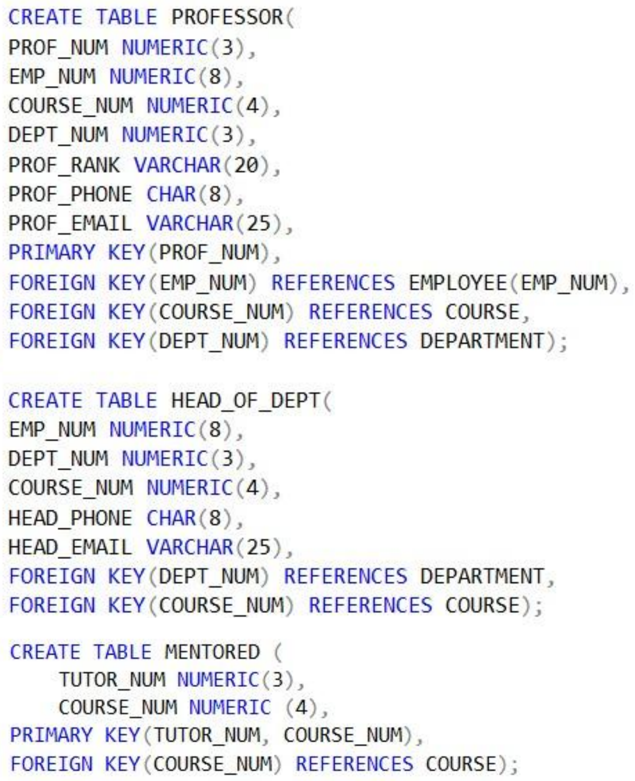
STUDENT	
PK	STU_NUM
FK1	COURSE_NUM
FK2	UNI_NUM
	STU_LNAME
	STU_FNAME
	STU_INITIAL
	STU_EMAIL
	STU_GPA
	CRED_HRS
	TOT_COURSES

REGISTERED_FOR	
PK, FK1	STU_NUM
PK, FK2	COURSE_NUM

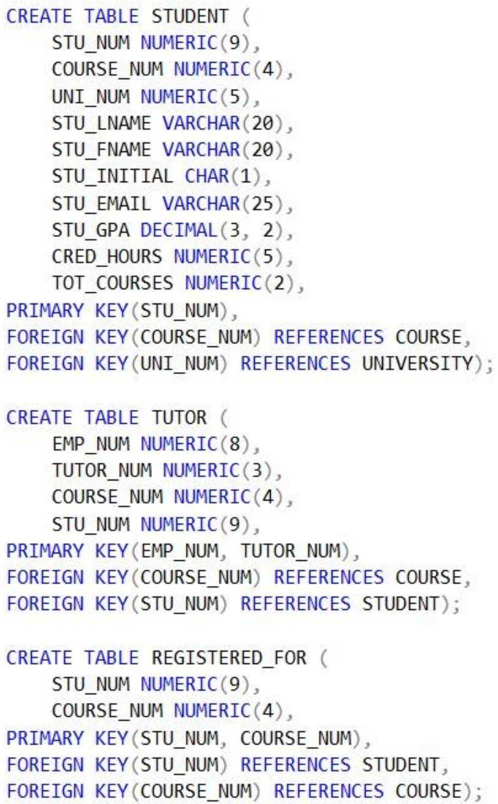
Implementation of Database Using Microsoft SQL Server



```
/*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);
```

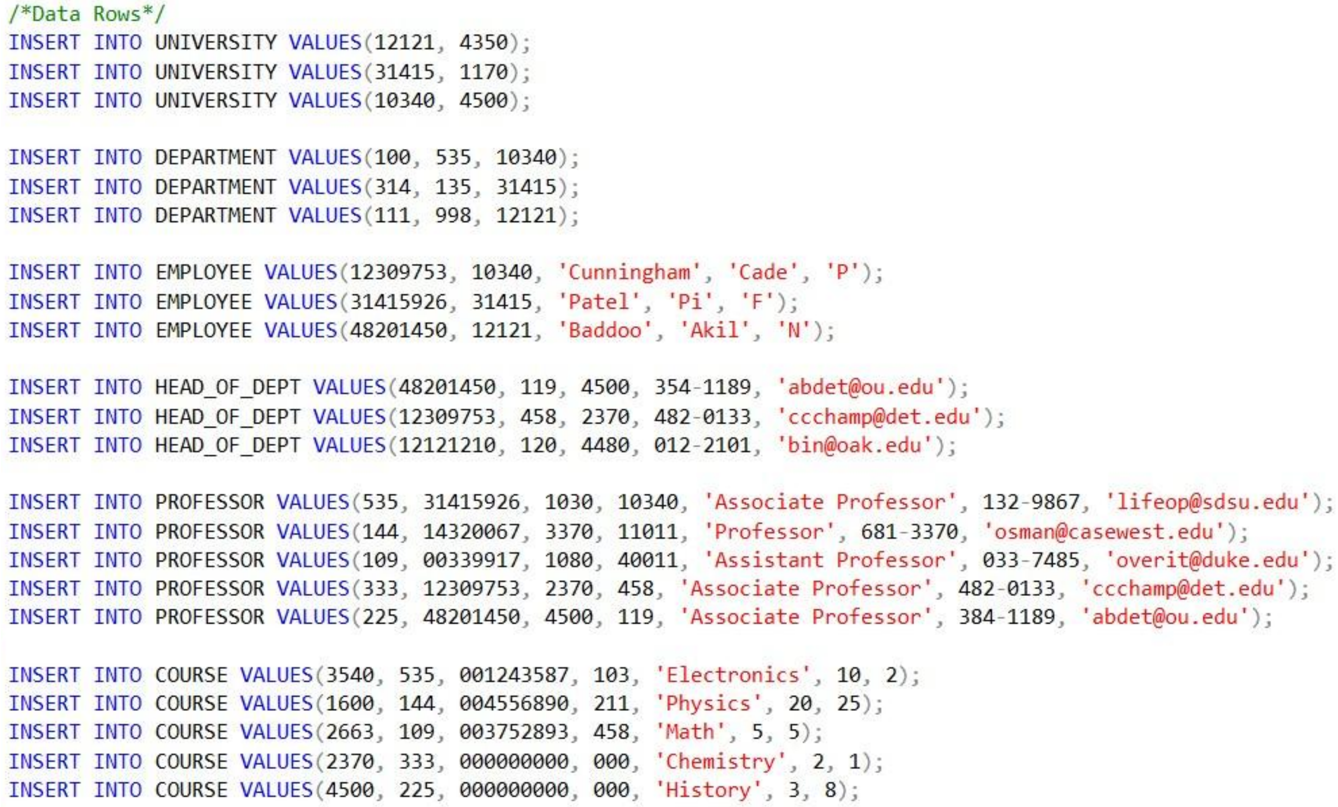


```
CREATE TABLE PROFESSOR(
PROF_NUM NUMERIC(3),
EMP_NUM NUMERIC(8),
COURSE_NUM NUMERIC(4),
DEPT_NUM NUMERIC(3),
PROF_RANK VARCHAR(20),
PROF_PHONE CHAR(8),
PROF_EMAIL VARCHAR(25),
PRIMARY KEY(PROF_NUM),
FOREIGN KEY(EMP_NUM) REFERENCES EMPLOYEE(EMP_NUM),
FOREIGN KEY(COURSE_NUM) REFERENCES COURSE,
FOREIGN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



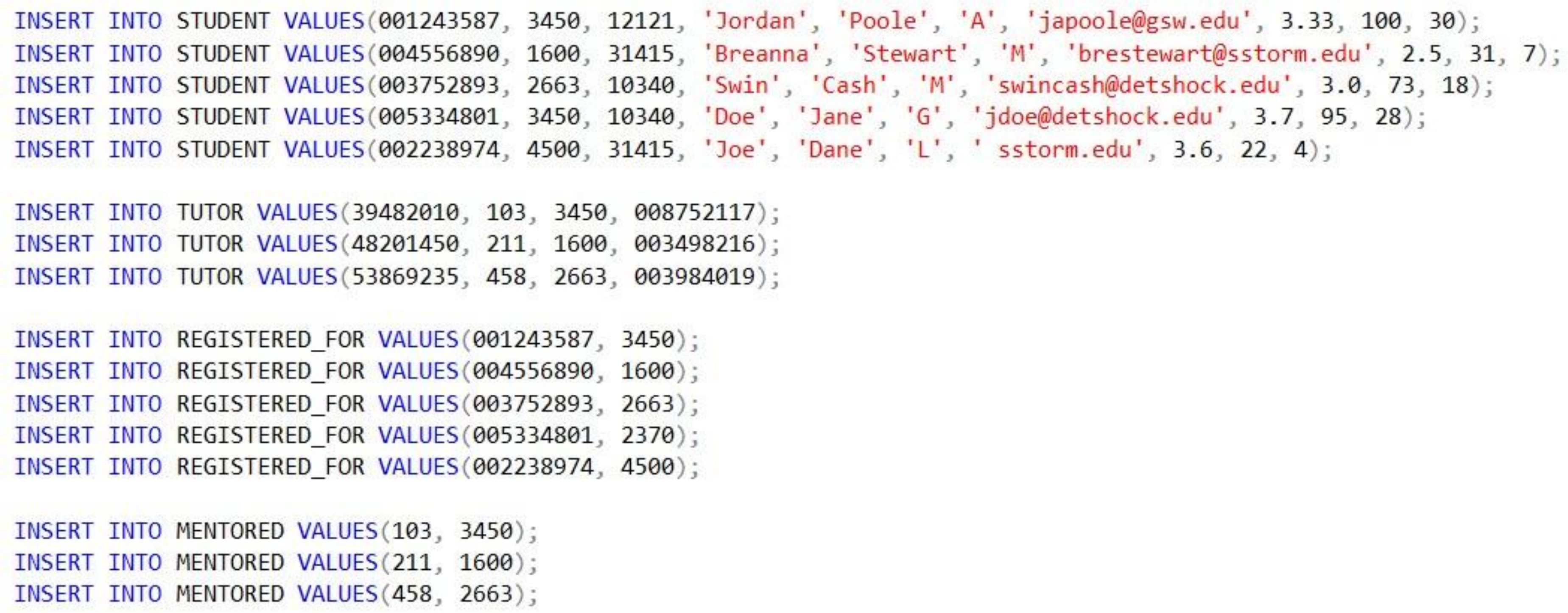
```
CREATE TABLE STUDENT (
    STU_NUM NUMERIC(9),
    COURSE_NUM NUMERIC(4),
    UNI_NUM NUMERIC(5),
    STU_LNAME VARCHAR(20),
    STU_FNAME VARCHAR(20),
    STU_INITIAL CHAR(1),
    STU_EMAIL VARCHAR(25),
    STU_GPA DECIMAL(3, 2),
    CRED_HOURS NUMERIC(5),
    TOT_COURSES NUMERIC(2),
PRIMARY KEY(STU_NUM),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

Implementation of Database Using Microsoft SQL Server



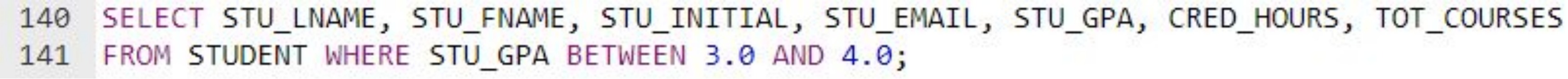
```
/*Data Rows*/
INSERT INTO UNIVERSITY VALUES(12121, 4350);
INSERT INTO UNIVERSITY VALUES(31415, 1170);
INSERT INTO UNIVERSITY VALUES(10340, 4500);

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);
```

Implementation of Database Using Microsoft SQL Server

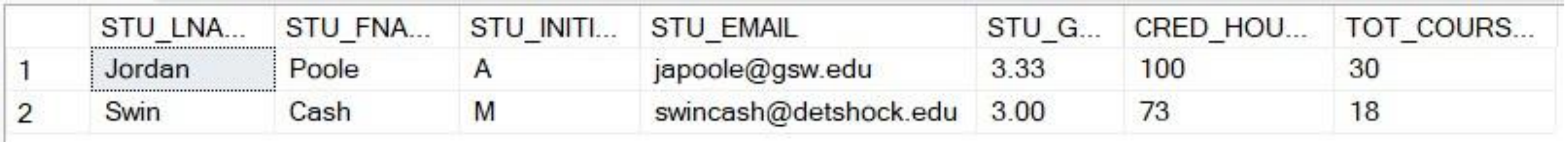


```
INSERT INTO STUDENT VALUES(001243587, 3450, 12121, 'Jordan', 'Poole', 'A', 'japoole@gsw.edu', 3.33, 100, 30);
INSERT INTO STUDENT VALUES(004556890, 1600, 31415, 'Breanna', 'Stewart', 'M', 'brestewart@sstorm.edu', 2.5, 31, 7);
INSERT INTO STUDENT VALUES(003752893, 2663, 10340, 'Swin', 'Cash', 'M', '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 - GPA between 3.0 and 4.0

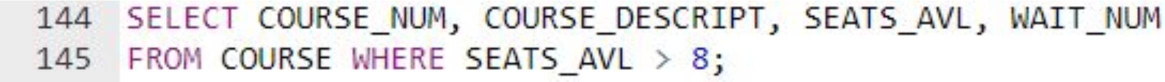


```sql
SELECT STU_LNAME, STU_FNAME, STU_INITIAL, STU_EMAIL, STU_GPA, CRED_HOURS, TOT_COURSES
FROM STUDENT WHERE STU_GPA BETWEEN 3.0 AND 4.0;
```

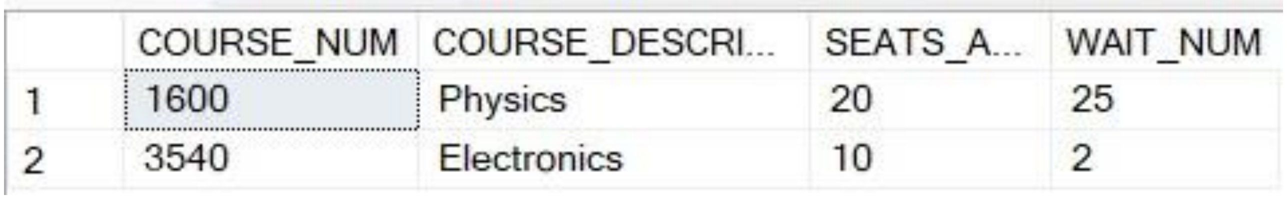


	STU_LNA...	STU_FNA...	STU_INITI...	STU_EMAIL	STU_G...	CRED_HOU...	TOT_COURS...
1	Jordan	Poole	A	japoole@gsw.edu	3.33	100	30
2	Swin	Cash	M	swincash@detshock.edu	3.00	73	18

Queries - Courses with more than 8 seats available

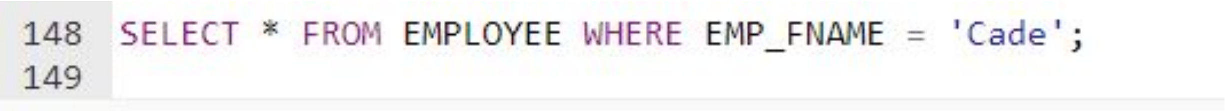


```
SELECT COURSE_NUM, COURSE_DESCRIPT, SEATS_AVL, WAIT_NUM
FROM COURSE WHERE SEATS_AVL > 8;
```

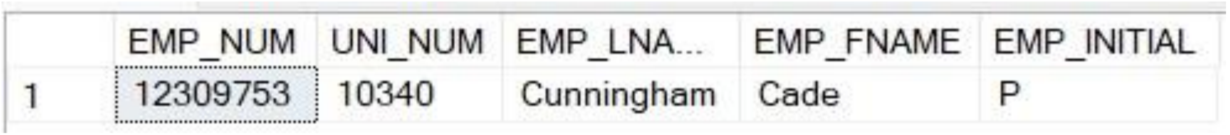


	COURSE_NUM	COURSE_DESCRI...	SEATS_A...	WAIT_NUM
1	1600	Physics	20	25
2	3540	Electronics	10	2

Queries - Finding specific employee using their first name

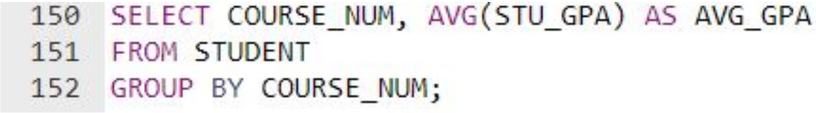


```sql
SELECT * FROM EMPLOYEE WHERE EMP_FNAME = 'Cade';

```

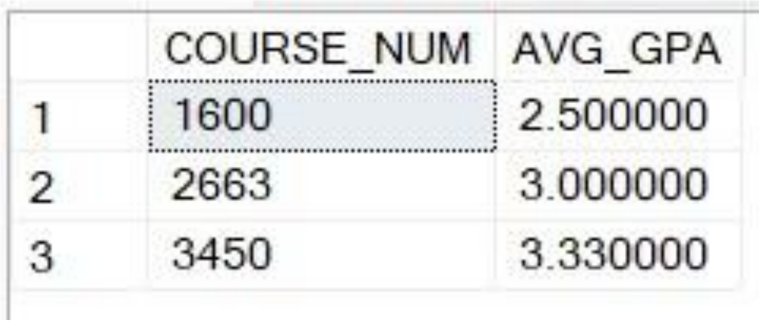


	EMP_NUM	UNI_NUM	EMP_LNA...	EMP_FNAME	EMP_INITIAL
1	12309753	10340	Cunningham	Cade	P

Queries – Average GPA



```
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

Challenges

- Microsoft SQL Server not cooperating
    - Not connecting
    - Not understanding errors given
- Finding a way we both could work on the SQL implementation at the same time
    - Not always having access to Microsoft SQL
- Determining relationships for overlapping entities