Hw 4 Solution

```
-- Part 1
CREATE TABLE PROFESSOR (
ID int NOT NULL, -- the primary key, so not null
first varchar(50), -- make this big enough to hold most names
last varchar(50),
PRIMARY KEY (ID)
) CHARACTER SET 'utf8mb4'
COLLATE 'utf8mb4_unicode_520_ci';

CREATE TABLE STUDENT (
ID int NOT NULL, -- the primary key, so not null
first varchar(50), -- make this big enough to hold most names
last varchar(50),
major varchar(50),
PRIMARY KEY (ID)
) CHARACTER SET 'utf8mb4'
COLLATE 'utf8mb4_unicode_520_ci';

CREATE TABLE COURSE (
catnum varchar(20) NOT NULL, -- the primary key, so not null
description varchar(2000) NOT NULL, -- this was a candidate key, so we know it's not
 null
ge_area varchar(10),
units int,
PRIMARY KEY (catnum),
UNIQUE (description) -- this was a candidate key, so we know it's unique
) CHARACTER SET 'utf8mb4'
COLLATE 'utf8mb4_unicode_520_ci';

CREATE TABLE PROF_EMAIL (
ID int NOT NULL, -- part of the primary key, so not null
email varchar(255) NOT NULL, -- part of the primary key, so not null
PRIMARY KEY (ID, email),
FOREIGN KEY (ID) REFERENCES PROFESSOR(ID)
ON DELETE CASCADE
ON UPDATE CASCADE
) CHARACTER SET 'utf8mb4'
COLLATE 'utf8mb4_unicode_520_ci';

CREATE TABLE STUDENT_EMAIL (
ID int NOT NULL, -- part of the primary key, so not null
email varchar(255) NOT NULL, -- part of the primary key, so not null
PRIMARY KEY (ID, email),
FOREIGN KEY (ID) REFERENCES STUDENT(ID)
ON DELETE CASCADE
```

```
ON UPDATE CASCADE
) CHARACTER SET 'utf8mb4'
COLLATE 'utf8mb4_unicode_520_ci';

CREATE TABLE SECTION (
catnum varchar(20) NOT NULL, -- part of the primary key, so not null
sectnum int NOT NULL, -- part of the primary key, so not null
semester varchar(20) NOT NULL, -- part of the primary key, so not null
room_num varchar(20),
prof_ID int,
PRIMARY KEY (catnum, sectnum, semester),
FOREIGN KEY (catnum) REFERENCES COURSE(catnum)
ON DELETE CASCADE
ON UPDATE CASCADE
) CHARACTER SET 'utf8mb4'
COLLATE 'utf8mb4_unicode_520_ci';

CREATE TABLE ENROLLED (
studentid int NOT NULL, -- part of the primary key, so not null
catnum varchar(20) NOT NULL, -- part of the primary key, so not null
sectnum int NOT NULL, -- part of the primary key, so not null
semester varchar(20) NOT NULL, -- part of the primary key, so not null
grade varchar(2),
rating int,
PRIMARY KEY (studentid, catnum, sectnum, semester),
FOREIGN KEY (studentid) REFERENCES STUDENT(ID)
ON DELETE CASCADE
ON UPDATE CASCADE,
FOREIGN KEY (catnum, sectnum, semester) REFERENCES SECTION(catnum, sectnum, semester)
ON DELETE CASCADE
ON UPDATE CASCADE
) CHARACTER SET 'utf8mb4'
COLLATE 'utf8mb4_unicode_520_ci';

-- Part 2
INSERT INTO PROFESSOR(ID, first, last) VALUES (1, 'Amos', 'Burton');
INSERT INTO PROF_EMAIL(ID, email) VALUES (1, 'amos@legitimatesalvage.com');
INSERT INTO STUDENT(ID, first, last, major) VALUES (1, 'Naomi', 'Nagata', 'Physics');
INSERT INTO STUDENT(ID, first, last, major) VALUES (2, 'Chrisjen', 'Avasarala', 'Ling
uistics');
INSERT INTO STUDENT_EMAIL(ID, email) VALUES (1, 'naomi@opa.org');
INSERT INTO COURSE(catnum, description, ge_area, units)
VALUES ('CSC134', 'Database Management Systems', NULL, 3);
INSERT INTO SECTION(catnum, sectnum, semester, room_num, prof_ID)
VALUES ('CSC134', 5, 'Spring 2020', 'RVR2008', 1);
INSERT INTO ENROLLED(studentid, catnum, sectnum, semester, grade, rating)
VALUES (1, 'CSC134', 5, 'Spring 2020', 'A', 4);

-- We need to use the entire primary key in our WHERE clause to insure we've uniquely
identified what to update.
UPDATE SECTION SET room_num = 'RVR1002' WHERE catnum = 'CSC134' AND sectnum = 5 AND s
emester = 'Spring 2020';

DELETE FROM STUDENT WHERE ID = 2;
```

```
-- Part 3
/*
You are not required to use the IF EXISTS in your drop tables. Since I use this to
clean up after grading each student, and not everyone successfully creates all their
 tables,
it just helps me avoid errors in the script.
*/
DROP TABLE IF EXISTS ENROLLED;
DROP TABLE IF EXISTS STUDENT_EMAIL;
DROP TABLE IF EXISTS PROF_EMAIL;
DROP TABLE IF EXISTS SECTION;
DROP TABLE IF EXISTS COURSE;
DROP TABLE IF EXISTS STUDENT;
DROP TABLE IF EXISTS PROFESSOR;
```