

Isaac Shoebottom, 3429069  
Lab 6

Question 1:

Query 1:

```
CREATE TABLE Students
(
    studentID INT AUTO_INCREMENT,
    studentName VARCHAR(30) NOT NULL,
    email VARCHAR(40),
    gpa FLOAT DEFAULT 0.0,
    PRIMARY KEY (studentID),
    CONSTRAINT CHECK_GPA
        CHECK (gpa BETWEEN 0 and 4.3)
);
```

Query 2:

```
ALTER TABLE Students ADD dateOfBirth DATETIME;
```

Query 3:

```
CREATE TABLE Enrollments
(
    studentID INT,
    courseID INT,
    letterGrade CHAR(2),
    PRIMARY KEY(studentID, courseID),
    FOREIGN KEY(studentID)
        REFERENCES Students(studentID)
        ON DELETE NO ACTION
        ON UPDATE CASCADE
);
```

Query 4:

```
ALTER TABLE Enrollments
ADD FOREIGN KEY(courseID)
REFERENCES Courses(courseID)
ON DELETE NO ACTION
ON UPDATE CASCADE;
```

Question 2:

Query:

```
CREATE TABLE Courses
(
courseID INT AUTO_INCREMENT,
courseNum VARCHAR(10),
courseName VARCHAR(40),
courseDescription VARCHAR(80),
creditHours FLOAT DEFAULT 3.0,
CHECK (creditHours BETWEEN 0.0 AND 6.0),
PRIMARY KEY(courseID)
);
```

Question 3:

Query 1:

```
INSERT INTO Courses(courseNum, courseName, courseDescription,
creditHours)
VALUES ('CS1103', 'Data & Information Management', 'Introduction
to Databases', 7);
```

Query 2:

```
INSERT INTO Courses(courseNum, courseName, courseDescription,
creditHours)
VALUES ('CS1303', 'Discrete Structures', 'Computer Science but
it\'s math actually', 4);
```

Question 4:

Query:

```
ALTER TABLE Courses ADD approvedDate DATETIME;
```

Question 5:

Query:

```
UPDATE Courses
SET approvedDate = ("2020-7-8")
where courseID = 1;
```

Question 6:

Rows added for example output:

```
INSERT INTO Students(studentName, email, gpa, dateOfBirth)
VALUES ('Isaac Shoebottom', 'ir.shoebottom@gmail.com', 3.5,
"2002-08-07");
```

```
INSERT INTO Enrollments(studentID, courseID, letterGrade)
VALUES(1, 1, 'A+');
```

```
INSERT INTO Enrollments(studentID, courseID, letterGrade)
VALUES(1, 2, 'B');
```

Query:

```
select studentName, courseName, courseDescription, creditHours,
letterGrade
from Students
natural join Enrollments
natural join Courses;
```

Output (Margins odd because the table is wide):

studentName	courseName	courseDescription	creditHours	letterGrade
Isaac Shoebottom	Data & Information Management	Introduction to Databases	7	B
Isaac Shoebottom	Discrete Structures	Computer Science but it's math actually	4	A+