## Instructions

Read these instructions all the way through before you start.

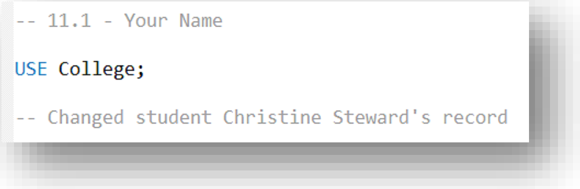
Download the answer document. Be sure to put your name at the top of the answer document where that's indicated. Answer the questions below in that document and when you are done, upload it into Blackboard.

## Overview

In this unit, for the first time, we work with SQL statements that modify data in the database. We'll change existing records (UPDATE), add new records (INSERT) and delete record (DELETE).

## Preparing your SQL

Make sure you include the comment and USE statement at the top of your SQL



```
-- 11.1 - Your Name

USE College;

-- Changed student Christine Steward's record
```

and don't forget to beautify your SQL, by hand if needed. In your screen shot of the Result Grid you need to show only first 6-10 rows. If you're not sure about these, see Unit 02 instructions for details on how to do that.

## Important Tips

- For the first time with this unit you're modifying data in the College database. This is going to cause some changes in the way your work. For example, while working on a SELECT statement you could run it over and over again while you're tweaking it. That may not be the case with statements that change the data. In one of the steps you'll add a new student to the student table with the INSERT statement. You may not be able to run the INSERT again because you would cause a duplicate record in the table. So, just rerun the CollegeCreate.sql script to recreate the database.

- It's also possible for you do accidently mess up some of the data, for example updating or deleting the wrong record. If you've messed up data or even think you might have, you can always rerun the CollegeCreate.sql script and in a few seconds you can put the database right back to the way it was.

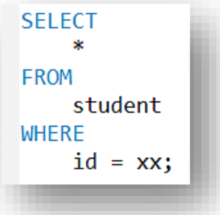
- Since you can always recreate the database, be sure to not be afraid to experiment with these statements.

- MySQL may not properly beautify UPDATE, INSERT and DELETE statements automatically. Please make them neat by hand if needed.

## Steps

1. [10] Student Deborah Reid has just married and is changing her last name to Nelson. Also, she's now majoring in Information Systems. Write an UPDATE statement to change her last name, email address and MajorID. Do this in a single statement. Note that email addresses use the student's first and last name. Then write a SELECT SQL statement that shows all the fields of her Student record after you've change it.

   In your UPDATE and SELECT statements, use Deborah's student ID, not her name, in your WHERE clause.

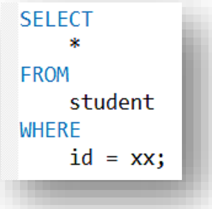


```
SELECT
    *
FROM
    student
WHERE
    id = xx;
```

2. [10] Add a new student to the student table. Here's the student's information.

| Field | Values |
|---|---|
| **LastName** | Robert |
| **FirstName** | Biggs |
| **Email** | RobertBiggs@college.edu |
| **Sex** | Male |
| **DateOfBirth** | 12/15/1997 |
| **EnrolledDate** | <use the date you're writing the SQL> |
| **MajorID** | Mathematics <use the id> |
| **Scholarship** | $3,350 |

Then write a SELECT statement lists all the fields from just the new record.



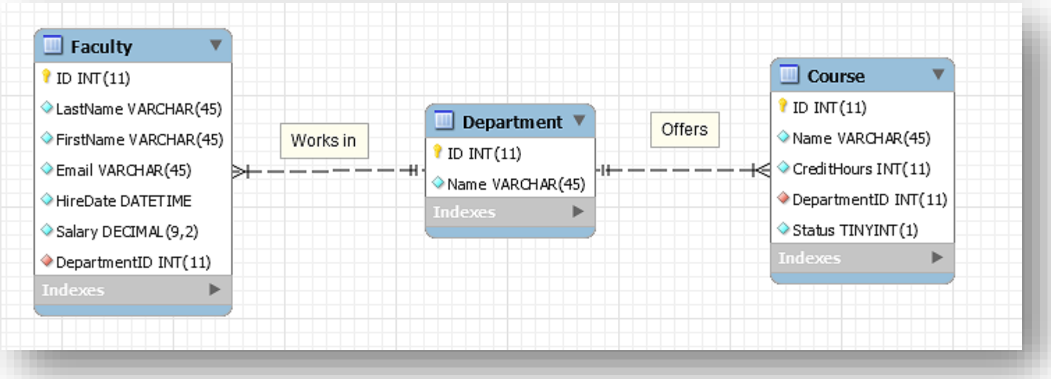
```
SELECT
    *
FROM
    student
WHERE
    id = xx;
```

3. [5] Run the INSERT statement from the previous step a second time in an attempt to add the student a second time. Then write one or two sentences explaining what happened and why.

4. [5] The Nursing department has decided to no longer offer the course “NS 1020 - Blood Letting”. Write a SQL Statement to remove this course from the system. Use the ID, not the name of the course, to select the record to delete. Run a SELECT statement listing all the Nursing courses to show that Blood Letting is no longer in the system.

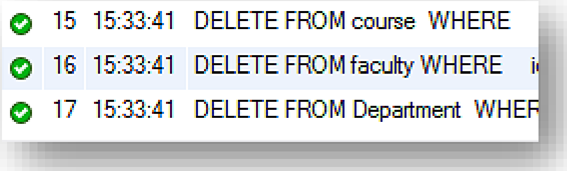
5. [10] The Miss Albright, the last professor in the Romance Languages department, has died and the college is closing the Romance Languages department. You’ll need to remove the department, any faculty that still works in the department, and any course the department offers. Write the <u>three</u> SQL DELETE statements necessary to remove everything in the system about the Romance Languages department.

   This step requires an understanding of Referential Integrity and of the College database. It helps to remember that the Department table has direct relationships with the Faculty and Course tables. For example, you can’t delete a department record if there’s still a faculty member in it.



Run these three statement and provide a screen shot the Output window showing the that three statements <u>successfully completed</u>. It should look something like this, but be sure to <u>show the</u>

whole message.



```
✓ 15  15:33:41  DELETE FROM course  WHERE
✓ 16  15:33:41  DELETE FROM faculty WHERE    i
✓ 17  15:33:41  DELETE FROM Department  WHER
```

6. [10] Add a new course (IT 2014 - C# Programming) into the system, chose the appropriate department. Then add a new section for the course in a semester of your choosing. Add yourself as a new student in the system and then register yourself for the section. Altogether, you'll have four INSERT statements. Paste your insert statements into the answer document.

   Once you have all four records in system, run these SELECT statements and paste the four results into the answer document.

```
SELECT * FROM Course WHERE ID = (SELECT MAX(ID) FROM Course) ;
SELECT * FROM Section WHERE ID = (SELECT MAX(ID) FROM Section) ;
SELECT * FROM Student WHERE ID = (SELECT MAX(ID) FROM Student) ;
SELECT * FROM Registration WHERE ID = (SELECT MAX(ID) FROM Registration) ;
```