Week 4 2024 Homework.
SQL Experience Part 2.
Due before Lecture February 20th, 2024

Author Names:

This assignment can be done in pairs or individually. If you collaborate on a solution, both members should submit a copy of the solutions. Also, if you are working in pairs, you should still both strive to understand the solutions you submit.

For each of the following data requests, create the SQL command and run it against the company database provided on the Moodle. Paste the command and result into this document. Once complete, submit the modified document to the Moodle submission assignment. Each question will indicate if a join, nested query, or both should be used. We have been working with natural joins right along, while information about nested queries is provided in the slides from February 15th that are posted on the Moodle for Week 4.

To see the column names and make the displayed data nicer use the following dot-commands.

```
.header on
.mode columns
```

If you are cutting and pasting anything from the **SQLite** window change the font to either "Courier New" or "Monaco". These are both fixed width fonts and will display things more nicely.

As you walk through this assignment, if you have any questions please ask by email or office hours.

1. Using a nested query, provide all employee names (**Lname**) for project "Computerization".

    Hint 1 – the solution will look similar to the second nested query on the slides.

2. Add five new employees to the employee table. They should have unique **Ssn** from the employees currently there, also they will have the same boss "Jennifer Wallace" (whose **Ssn** is 987654321) and will be assigned to department 5. Show both the commands you ran and the output of a *select* to demonstrate you correctly updated the information.

    Hint 2 – Sometimes mistakes are made where incomplete or incorrect information is added to the database, you can run the following commands from the directory where you ran your "make" command. The first command destroys your database and the second rebuilds it from the provided CSV files in the data directory.

```
make drop
make build
```

3. Recently, the supervisor Jennifer Wallace left the company. You do not need to remove Mrs Wallace from the employee table, but you do need to change the **Super_ssn** of all the people that reported to Mrs Wallace with the **Ssn** of their new supervisor James Borg. Use the update command to realign all the individuals who worked for Mrs Wallace and will now work for Mr Borg. Then display the employee table to show the changes

were made correctly. If you make a mistake, you can reset the database as per hint 2, but you will also need to re-run the changes in question 2.

4. After the arrival of the new employees and recent departure of Mrs Wallace, the owner of the company needs to see who is now working for each supervisor and what projects these individuals are assigned. The owner would also like to see the output with meaningful header names. This can be done with a three table join.

5. After reviewing the report from question 4, the owner would now like to see the same report but for a specific supervisor identified by their last name without the supervisor's name listed. Your boss see's the three table join as computationally expensive and wants you to create this new report with only a two table join and a nested query to get the supervisor's **Ssn** using their last name. To demonstrate that your query works, use James Borg's last name in this query.