CS 325 - Week 9 Lab Exercise

Deadline

Due by the end of lab on 2021-10-22.

Purpose

To practice writing more SQL `select` statements, including practice with `order by`, `group by`, and `having` clauses.

How to submit

JUST this "driver" for each pair should use `~st10/325submit` to submit the pair's copy of this lab exercise's files, with a lab number of **89**

Important notes

- **I have included an example** `325lab9-out.txt` **along with this lab exercise handout, for comparison purposes.**
  - This is both to let you know if you are on the right track, AND to hopefully encourage DEBUGGING of your SQL `select` statements if you see significant differences.
- You may find the following useful for this lab exercise:
  - NOTE that on the course Canvas site, under Modules, in the "Class Recordings" section, the FIRST link in that section leads to the public course web site's "In-class Examples" section.
  - SQL Reading Packet 5 - Order by Group by, and Having
  - `325lect09-2.sql` - the SQL script "built" during the Week 9 Asynchronous Material
- You are required to work in **pairs** for this lab exercise. If you are not pair-programming, then you may not receive full credit for your lab exercise.
  - If there are an odd number of students attending lab, or too many students with connectivity issues, some teams may have 3 students.
- **RECOMMENDATION:** RUN your script-in-progress FREQUENTLY as you are developing it -- do not create the entire script before running it for the first time.

Lab Exercise set-up

- On nrs-projects, **CREATE** a directory `325lab9`, protect it, and go to it:

```
mkdir 325lab9
chmod 700 325lab9
cd 325lab9
```

- **IF** you do not already have tables `empl`, `dept`, and `customer`, COPY the following script to your directory:

```
cp ~st10/set-up-ex-tbls.sql .  # remember the space and period
```

...and **run** it in `sqlplus` to get your own versions of these tables.

Lab Exercise tasks

- Then, begin a SQL script **`325lab9.sql`** with comment(s) including at least **BOTH (all)** of your **names** and **today's date**. Add commands for the following into this SQL script.
- Start spooling to a file **`325lab9-out.txt`**.
- Write a `prompt` command to print a message to the screen containing **both** of your names.
- Write a `prompt` command outputting **`lab query 1`**, then write a query performing a relational selection of all of the rows of the `dept` table, displaying the resulting rows in order of department **location**.
- Write a `prompt` command outputting **`lab query 2`**, then write a query performing a relational selection of all of the rows of the `dept` table, displaying the resulting rows in order of department **name**.
- Write a `prompt` command outputting **`lab query 3`**, then write a query projecting just the `empl_last_name`, `dept_name`, `salary`, and `hiredate` from the join of the `empl` and `dept` tables, displaying the resulting rows in primary order by `dept_name` and in secondary order by `hiredate`.
- Write a `prompt` command outputting **`lab query 4`**, then write a query projecting just the `empl_last_name`, `dept_name`, `salary`, and `hiredate` from the join of the `empl` and `dept` tables, displaying the resulting rows in primary order of **decreasing** `salary` and in secondary order by **increasing** `hiredate`.
- Write a `prompt` command outputting **`lab query 5`**, then write a query projecting, for each value of `mgr`, the earliest `hiredate` for those `empl` rows with that value of `mgr`. (That is, the result has just two columns: the `mgr` value, and the earliest `hiredate` of employees who have that `mgr`.)
- Write a `prompt` command outputting **`lab query 6`**, then write a query projecting, for each value of `mgr`, the earliest `hiredate` for those `empl` rows with that value of `mgr`, BUT only for `mgr` values for which the earliest `hiredate` is after January 1, 2015. (The result has the same two columns as lab query 5's result, buts it should have fewer rows.)
- Write a `prompt` command outputting **`lab query 7`**, then write a query projecting, from the join of `empl` and `dept`, for each value of `dept_name`, the number of employees who work in that department, displaying the resulting rows in order of **increasing** number of employees.
- Write a `prompt` command outputting **`lab query 8`**, then write a query projecting, from the join of `empl` and `dept`, for each value of `dept_name`, the number of employees who work in that department, BUT only for departments whose average salary of less than $2000, displaying the resulting rows in order of **increasing** number of employees.
- Write a `prompt` command outputting **`lab part 9`**, then think of at least one question you could ask about employees, departments, and/or customers, that you think you can answer using **at least one of**:
  - `order by`

  - `group by` (used with at least one aggregate function call)
  - `having`

  ...(although it is fine if it could use more than one of these!) (It should ask something **different** than is answered by any of the queries above.)

  Then:

  - Write a `prompt` command printing at least one such question you decided on.
  - Then write a query answering each such question you give. (For lab exercise purposes, make sure the result has at least one row in it.)

- Turn off spooling.
- When you believe your SQL script is working properly, submit your `325lab9.sql` and `325lab9-out.txt` files using `~st10/325submit` with a homework number of **89**.
  - (Once you have submitted your lab exercise files, you may leave lab if you wish. Or, you can ask questions, (noting that lab-exercise-related questions need to receive 1st priority), work on the CS 325 Project Model Draft milestone, work on finishing touches on Homework 6, etc.)