CS 325 - Week 10 Lab Exercise

Deadline

You can find the due date and time for Lab 10 in the Assignment tab on Canvas, listed under your specific lab section.

Purpose

To practice writing more SQL `select` statements, including practice with creating and using SQL views.

How to submit

EACH person in the pair/trio should submit a copy of `325lab10.zip` on Canvas.

Important notes

- **I have included an example** `325lab10-out.pdf` **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:
  - Course Slides
  - SQL Reading Packet 7 - Views, and Simple Reports - Part 1
- For this lab exercise, you are required to work in pairs - 2-person (a single 3-person team will be allowed if there are an odd total number of students in the lab). The point here is to have teams discuss the answers to be given below – if nobody in the team knows the answer, please consult the lecture slides, posted readings, and online documentation for guidance. Ask the team next to you ONLY if you have exhausted these other options first!
- **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 `325lab10`, protect it, and go to it:

```
mkdir 325lab10
chmod 700 325lab10
cd 325lab10
```

- (Friendly tip: go ahead and run `set-up-ex-tbls.sql` in `sqlplus` before starting your script, since the Week 10 Lab Exercise might have left them in a non-standard state!)
- To make a copy of `set-up-ex-tbls.sql` in the current directory by:
  - **PUT** the SQL script (`set-up-ex-tbls.sql` under week 10 lab module on canvas) to your 325lab10 directory on nrs-projects through **sftp** (with a **put** command).

Lab Exercise tasks

- Then, begin a SQL script **`325lab10.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 **`325lab10-out.txt`**.
- Write a `prompt` command to print a message to the screen containing **both** of your names.
- Write a `prompt` command outputting **`lab problem 1`**, then drop and create a view **`empl_salaries`** that includes just employee last names and employee salaries.
- Write a `prompt` command outputting **`lab problem 2`**, then write a query that **uses JUST** the `empl_salaries` view to project all of the columns of all of the rows of the `empl_salaries` view.
- Write a `prompt` command outputting **`lab problem 3`**, then write a query that **uses JUST** the `empl_salaries` view to project just the highest salary.
- Write a `prompt` command outputting **`lab problem 4`**, then drop and create a view **`earliest_hires`** that will contain two columns: a job title, and the earliest (minimum) hire date for someone with that job title.
- Write a `prompt` command outputting **`lab problem 5`**, then write a query that **uses JUST** the `earliest_hires` view to project all of the columns of all of the rows of the `earliest_hires` view.
- Write a `prompt` command outputting **`lab problem 6`**, then write a query that **uses JUST** the `earliest_hires` view to project just the latest minimum hire date for any job title.
- Write a `prompt` command outputting **`lab problem 7`**, then drop and create a view **`cust_reps`** that includes three columns:
  - the customer last names concatenated with a comma and blank and the customer first names, somehow given a column name of `CUSTOMER`
  - the last name of the employee serving as their employee rep, somehow given the column name of `REPD_BY`
  - the location of the department of the employee serving as their employee rep
- Write a `prompt` command outputting **`lab problem 8`**, then write a query that **uses JUST** the `cust_reps` view to project all of the columns of all of the rows of the `cust_reps` view.
- Write a `prompt` command outputting **`lab problem 9`**, then think of at least one different view of the information about employees, departments, and/or customers, that you think would be useful or interesting to have.

(It should be a view at least somewhat **different** than those you have already created for this lab exercise.)

Then:

- Write a `prompt` command describing the view you decided to create.
- Then drop and create that view.
- Then write a query projecting all of the columns of all of the rows of your new view.
- Then write another query that uses your new view, but differently from the previous query --
  * it might project just some of its columns,
  * or it might just select just some of its rows,
  * or it might be joined or usefully used with another table or view (using sub-selects or set-theoretic operations),

    * or you might use group by with that view,
    * or some interesting combination of the above!
  - (For lab exercise purposes, make sure the results of both of these queries have at least one row in them.)
- Turn off spooling.

When you believe your SQL script is working properly, zip the `325lab10` folder, submit the `325lab10.zip` on Canvas, your folder should contain at least the following files:

- `325lab10.sql`
- `325lab10-out.txt`

Once your lab exercise files have been submitted, you may leave the lab if you wish.