CS 325 - Week 11 Lab Exercise

Deadline

You can find the due date and time for Lab 11 in the Assignment tab on Canvas, listed under your specific lab section.

Purpose

To practice with some SQL*Plus features that can be useful for creating reports

How to submit

EACH person in the pair/trio should submit a copy of `325lab11.zip` on Canvas.

Important notes

- **I have included an example** `325lab11-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 8 - Simple reports, parts 1 and 2
- 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 `325lab11`, protect it, and go to it:

```
mkdir 325lab11
chmod 700 325lab11
cd 325lab11
```

- **IF** you do not already have tables `empl`, `dept`, and `customer`, **PUT** the SQL script (`set-up-ex-tbls.sql` under week 11 lab module on canvas) to your 325lab11 directory on nrs-projects through **sftp** (with a **put** command):

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

Lab Exercise tasks

- Then, begin a SQL script **`325lab11.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 **`325lab11-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 write a query that does a join of `dept` and `empl`, BUT it only projects the columns `dept_name` and `salary`, ORDERING the results by `dept_name`.
- Write a `prompt` command outputting **`lab problem 2`**, then...
  - set the page size to 50 lines, and
  - set feedback to off.
  - use `/` to re-run the previous query
- Write a `prompt` command outputting **`lab problem 3`**, then...
  - write a `column` command that gives the `dept_name` column a noticeably-different heading *and* format of your choice, and
  - use `/` to re-run the previous query, to show this change in action.
- Write a `prompt` command outputting **`lab problem 4`**, then...
  - write a `column` command that gives the `salary` column a noticeably-different heading *and* format of your choice, and
  - use `/` to re-run the previous query, to show this change in action.
- Write a `prompt` command outputting **`lab problem 5`**, then...
  - write a `break` command to break on column `dept_name`, skipping 1 row after each different value
  - use `/` to re-run the previous query, to show this change in action.
- Write a `prompt` command outputting **`lab problem 6`**, then...
  - write a `compute` command to print the average of the salaries for each `dept_name`, and
  - use `/` to re-run the previous query, to show the `compute`'s effects.
- Write a `prompt` command outputting **`lab problem 7`**, then...
  - write a `ttitle` to add a top title of your choice,
  - use `/` to re-run the previous query, and show the resulting top title.
- Write a `prompt` command outputting **`lab problem 8`**, then:
  - Write a second query of your choice (at least somewhat **different** from that in `lab problem 1`) that projects at least 3 different columns and/or computations, and contains at least 4 rows, and uses `order by` to specify the order its rows are displayed.
  - Use `column` commands to give each column a noticeably-different heading and format of your choice.
  - Use at least one other of the SQL*Plus features we have discussed, or from SQL Reading Packet 8, to make one other noticeable change (you could change the top title; you could add a bottom title; you could use a `break`, or a `break` and a `compute`; you could change the `linesize` or `pagesize` in some noticeable way; etc.)

- (For this problem, it is fine to put your `column` commands and other SQL*Plus commands *before* your query, rather than putting your query, then these commands, then a `/`.

We played around a lot with `/` in the earlier problems, but it really is more typical in a SQL script building a report to put the SQL*Plus commands setting what you want and *then* the query whose display is to make use of them!)

- Write a `prompt` command outputting **`lab problem 9`**, then...
  - be polite, and write the command(s) to clear breaks, columns, and computes,
  - and turn top titles off,
  - and set pagesize back to 14,
  - and set feedback back to 6.
  - (And if you reset any other SQL*Plus defaults for `lab problem 8`, restore those to their default values, also.)
- Turn off spooling.

When you believe your SQL script is working properly, zip the `325lab11` folder, submit the `325lab11.zip` on Canvas, your folder should contain at least the following files:

- `325lab11.sql`
- `325lab11-out.txt`