

CS 325 - Week 6 Lab Exercise

Deadline

You can find the due date and time for Lab 6 in the Assignment tab on Canvas, listed under your specific lab section.

Purpose

To practice writing more SQL `select` statements, including some using column aliases, computed columns, table aliases, aggregate functions, and nested selects/sub-selects.

How to submit

EACH person in the pair/trio should submit a copy of `325lab6.zip` on Canvas.

Important notes

- **I have included an example `325lab6-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 3
 - SQL Reading Packet 4
- 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 `325lab6`, protect it, and go to it:

```
mkdir 325lab6
chmod 700 325lab6
cd 325lab6
```
- **IF** you do not already have tables `empl`, `dept`, and `customer`, **PUT** the SQL script (`set-up-ex-tb1s.sql` under week 6 lab module on canvas) to your `325lab6` directory on nrs-projects through **sftp** (with a **put** command):

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

Lab Exercise tasks

- Then, begin a SQL script `325lab6.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 **325lab6-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 that projects just department names and department locations, but uses a column alias of `LOCATION` for the department location.
- Write a `prompt` command outputting **lab query 2**, then write a query that projects just the employee last names, hire dates, and job titles, but uses column aliases to get exactly these column headings displayed in the result (including using the case shown):

```
Last Name   Hired      Job Title
```

- Write a `prompt` command outputting **lab query 3**, then write a query that projects the employee last names, their hire dates, and the results of adding 3 to every employee's hire date, using the column alias of `"H PLUS 3"` for the 3rd column in your result.
 - FUN FACT: yes, you can add to an Oracle SQL date! And it results in the date that many days from the given date. Check out the result for Martin's hire date + 3.
- Write a `prompt` command outputting **lab query 4**, then write a query that projects job titles, commissions, and the results of multiplying commissions by 1.2 (seeing the values of commissions if they were increased by 20%), using the column alias of `"PLUS 20%"` for the 3rd column in your result.
 - Hint: you should see what happens when you try to project a computed column for a selected row with a null value for that column... 8-)
- Write a `prompt` command outputting **lab query 5**, then write a query that selects just those employees with a non-null commission, and for those employees, projects job titles, commissions, and the results of multiplying commissions by 1.2 (seeing the values of commissions if they were increased by 20%), using the column alias of `"PLUS 20%"` for the 3rd column in your result.
- Write a `prompt` command outputting **lab query 6**, then write a query that projects just customer last names and the **last names** of the employees that serve as their employee rep.
- Write a `prompt` command outputting **lab query 7**, then write a query that projects just employee last names and the locations of the departments for which they work. For full credit, use **table aliases** for the `dept` and `empl` tables in this query.
- Write a `prompt` command outputting **lab query 8**, then write a query that projects just the employee last names, the department numbers, and the names of departments for which they work. For full credit, use **table aliases** for the `dept` and `empl` tables in your query.
- Write a `prompt` command outputting **lab query 9**, then write a query that projects the number of rows of `empl` that have salaries greater than 2000.
 - Note: I added a column alias for this in my version of `325lab6.sql`, so the posted `325lab6-out` file would not "give away" too much!
 - You may decide if you would like to use a column alias for this query or not.
- Write a `prompt` command outputting **lab query 10**, then write a query that projects the earliest hire date and the latest hire date for employees whose job title is 'Clerk'.
 - Note: I added column aliases for this in my version of `325lab6.sql`, so the posted

325lab6-out file would not "give away" too much!

- You may decide if you would like to use column aliases for this query or not.
- Write a `prompt` command outputting **lab query 11**, then write a query that projects, for employees whose job title is 'Sales', how many such rows were selected, the sum of the commissions for those rows, and the average commission for those rows.
 - Note: I added column aliases for this in my version of 325lab6.sql, so the posted 325lab6-out file would not "give away" too much!
 - You may decide if you would like to use column aliases for this query or not.
- Write a `prompt` command outputting **lab query 12**, then write a query that projects, for all the rows of the `emp1` table, how many rows were selected, how many of the selected rows have a non-null value for the `commission` column, and how many rows have a non-null value of the `mgr` column.
 - Note: I added column aliases for this in my version of 325lab6.sql, so the posted 325lab6-out file would not "give away" too much!
 - You may decide if you would like to use column aliases for this query or not.
- Write a `prompt` command outputting **lab query 13**, then write a query that projects the last name and hiredate of employees with the job title of 'Manager' who make more than the average salary of employees with the job title of 'Manager'.
 - (note: this involves `job_title`, it does **NOT** involve the `mgr` attribute)
- Write a `prompt` command outputting **lab query 14**, then write a query, **using an appropriate sub-select**, and **not** using a join, that projects the hiredates of the employees whose department location is New York.
- Write a `prompt` command outputting **lab query 15**, then write a query, **using an appropriate sub-select**, that projects the employee last names and hiredates for **all** employees that were hired after the latest-hired employee who has the job title of 'Manager'.
- (note: again, this involves `job_title`, it does **NOT** involve the `mgr` attribute)
- Turn off spooling.

When you believe your SQL script is working properly, zip the 325lab6 folder, submit the 325lab6.zip on Canvas, your folder should contain at least the following files:

- 325lab6.sql
- 325lab6-out.txt

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