# CS 325 - Week 6 Lab Exercise

## Deadline

Due by the end of lab on 2021-10-01.

## Purpose

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

## 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 **86**

## Important notes

- **I have included an example** `325lab6-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 3 - More where clause options and aggregate functions
- 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 `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`, 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 **`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 hiredates, and the results of adding 3 to every employee's hiredate, 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 hiredate + 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.txt` 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 hiredate and the latest hiredate 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.txt` 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.txt` 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 `empl` 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.txt` 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 part 13`**, then think of at least one question you could ask about employees, departments, and/or customers, that you think you can answer using either **computed columns**, **aggregate functions**, or both. (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 `325lab6.sql` and `325lab6-out.txt` files using `~st10/325submit` with a homework number of **86**.
  - (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 homework, etc.)