# CS 325 - Week 11 Lab Exercise

## Deadline

Due by the end of lab on 2021-11-05.

## Purpose

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

## 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 **91**

## Important notes

- **I have included an example** `325lab11-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 7 - Views, and Simple Reports - Part 1
    - `325lect11-2.sql` - the SQL script "built" during the Week 11 Asynchronous Material - part 2
- 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 `325lab11`, protect it, and go to it:  `mkdir 325lab11`
  `chmod 700 325lab11`
  `cd 325lab11`
- (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:

```
cp ~st10/set-up-ex-tbls.sql  .   # remember SPACE and PERIOD at end!!
```

# 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 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, submit your `325lab11.sql` and `325lab11-out.txt` files using `~st10/325submit` with a homework number of **91**.

  - – (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 Design Draft milestone, work on finishing touches on Homework 8, etc.)