CS 325 - Week 10 Lab Exercise

Deadline

Due by the end of lab on 2021-10-29.

Purpose

To practice writing more SQL `select` statements, including practice with `union`, `intersect`, and `minus` operations, and to practice writing an `update` and a `delete` statement.

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 **90**

Important notes

- **I have included an example `325lab10-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 6 - Set-theoretic Operations, More on Modifying Data, and Sequences
  - `325lect10-2.sql` - the SQL script "built" during the Week 10 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 `325lab10`, protect it, and go to it:

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

- **NOTE**: You will be changing the contents of the `empl` and `dept` tables in this script, so it will be convenient to start with a "restored" original version of their contents each time you run this script.

  SO, make sure there is a copy of `set-up-ex-tbls.sql` in the current directory:

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

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.
- Put this command **IN** your script:

  `start set-up-ex-tbls.sql`

  ...so that your script will **always** start with a "clean" complete version of the `empl` and `dept` tables.
- **AFTER** that `start` command, **THEN** start spooling to a file **`325lab10-out.txt`**.
  - (I **don't** want the output from deleting the rows from, re-creating, and re-populating these tables in the 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 query 1`**, then write a query that performs the **union** of `job_title` and `mgr` values of employees whose salary is less than 2000 and `job_title` and `mgr` values of employees with a non-`NULL` commission.
- Write a `prompt` command outputting **`lab query 2`**, then write a query that performs the **intersection** of `job_title` and `mgr` values of employees whose salary is less than 2000 and `job_title` and `mgr` values of employees with a non-`NULL` commission.
- Write a `prompt` command outputting **`lab query 3`**, then write a query that performs the **difference** of `job_title` and `mgr` values of employees whose salary is less than 2000 and `job_title` and `mgr` values of employees with a non-`NULL` commission.
- Write a `prompt` command outputting **`lab query 4`**, then write a query that performs the **difference** of the employee last names and two times the salary of employees who are Clerks and the employee last names and two times the salary of employees whose hiredate is before January 1, 2015; but give the projected two-times-the-salary the column alias `TWICE_SALARY` and order the resulting rows in decreasing order of that two-times-the-salary.
- Write a `prompt` command outputting **`lab query-set 5`**, then:
  - write a query projecting JUST the employee last names, and the commissions, for employees whose commission is NOT `null`
  - write a query projecting JUST the employee last names, and the number `0`, for employees whose commission IS `null`
  - then, write a query projecting the **union** of the two above queries, giving the second column the heading `COMM_VALUE` and ordering the resulting rows in descending order of the second column's values, and for those with the same second column values, in ascending order of last name.
- Write a `prompt` command outputting **`lab query-set 6`**, then:
  - write a query projecting employee last names, job titles, and salaries of employees who do not have the job title of 'Manager' and make more than the lowest-paid employee with job title of 'Manager'

  - write a query projecting employee last names, job titles, and salaries of employees who work in a department whose location is New York
  - then, write a query projecting the **union** of the above two queries, displaying the resulting rows in order of salary.
- Write a `prompt` command outputting **`lab problem 7`**, then:
  - write a query projecting JUST the last names, job titles, salaries, and commissions JUST for employees with non-null commissions
  - write an **`UPDATE`** command that will increase the commission attribute by 100 for all Sales employees who make less than the average salary for a Sales employee
  - write a query projecting just the last names, job titles, salaries, and commissions for ALL employees
- Write a `prompt` command outputting **`lab problem 8`**, then:
  - write a **`DELETE`** command that will delete employees of any job title who make less than the lowest-paid Sales employee
  - write a query projecting just the last names, job titles, and salaries for all employees
- Write a `prompt` command outputting **`lab problem 9`**,

  do a SQL `rollback;` command, (to undo the database contents changes from lab problems 6 and 7),

  and then think of at least one question you could ask about employees, departments, and/or customers, that you think you can answer using **at least one of**:
  - `union`
  - `intersect`
  - `minus`

  ...(although it is fine if it could use more than one of these!) (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 `325lab10.sql` and `325lab10-out.txt` files using `~st10/325submit` with a homework number of **90**.
  - (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 7, etc.)