CS 325 - Week 1 Lab Exercise

Deadline

Due by the end of lab on 2021-08-27.

How to submit

- For PART 1, **EACH person** in the pair should submit the file `325part1.txt` using `~st10/325submit` with a lab number of **81**
- For PART 2, **JUST the "driver"** should use `~st10/325submit` to submit the pair's copy of `325lab1.sql` and `325lab1-out.txt`, with a lab number of **81**

Important notes

- You may find the following useful for this lab exercise:
  - "Useful Linux Commands" handout
    - (posted on the public course web site and course Canvas site)
  - "Useful details - `ssh`, `~st10/325submit`, and `sftp`" handout
    - (posted on the public course web site and course Canvas site)
  - SQL Reading Packet 1 - Intro to Oracle SQL at HSU
    - (posted on the course Canvas site, in the "Modules" section, in the "Week 1 Asynchronous Materials" section and in the "SQL-related CS 325 Reading Packets" section)
  - `demo.sql`
    - (posted on the course Canvas site, in the "Modules" section, in the "Week 1 Asynchronous Materials" section
- **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.
  - save your script-file-in-progress `325lab1.sql`
  - run it in `sqlplus` using:

    ```
    SQL> start 325lab1.sql
    ```

    (or using one of the variants of this discussed in SQL Reading Packet 1)
- You are required to work in **pairs** for PART 2 of 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.
  - **TYPE BOTH (all) OF YOUR NAMES** in the beginning comment of PART 2's `325lab1.sql` file.

PART 1

This is a rare, NON-pair-programmed part -- each person should complete this individually while you are in the lab breakout rooms, but PLEASE discuss and help each other through it!

- You can take turns sharing the screen if you would like.
- Each person should use `ssh` (for example, via PuTTY on `vlab.humboldt.edu` or via command-line `ssh` in a terminal) to connect and log into `nrs-projects.humboldt.edu`
  - IF someone CANNOT successfully connect and log into nrs-projects, TELL THE LAB INSTRUCTOR.
  - (If the lab instructor can't figure out the problem during lab, you will still receive credit for Part 1 if the lab instructor has noted you were not able to do so during lab!)
- Each person should type `sqlplus` at the nrs-projects prompt, and make sure they can log into their HSU SQL*Plus account on the HSU `student` database.
  - You can then exit `sqlplus` by typing `exit` at the `SQL>` prompt.
  - IF someone CANNOT successfully log into `sqlplus`, TELL THE LAB INSTRUCTOR.
  - (If the lab instructor can't figure out the problem during lab, you will still receive credit for Part 1 if the lab instructor has noted you were not able to do so during lab!)
- Each person should:
  - make and protect a directory `325lab1` using the commands:

    ```
    mkdir 325lab1
    chmod 700 325lab1
    ```
  - go into that directory using:

    ```
    cd 325lab1
    ```
  - use `nano` to create a file `325part1.txt` that contains **their name** and **today's date**

    ```
    nano 325part1.txt
    ```

    (remember, type the control key and o (^O) to write out, or save, your file,

    and then type the control key and x (^X) to exit `nano`)
  - submit their resulting file using:

    ```
    ~st10/325submit
    ```

    ...using a lab number of **81**

And that's all of PART 1.

PART 2

For this part, now you should be pair-programming, Zoom style, in your breakout room:

- ONE student, student A, SHARES their screen in the Zoom breakout room

- the OTHER student(s), student B (and if necessary student C), says what to type (or take turns saying what to type), and student A types it
- BOTH (or ALL) the students should be looking at the shared screen, and discussing concepts/issues along the way

Using this approach:

- Begin a SQL script **`325lab1.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 **`325lab1-out.txt`**.

  ```
  spool 325lab1-out.txt
  ```

- **THINK ABOUT FIRST**: Decide on a table for a type of thing of your choice, that someone might like to **borrow**, with an appropriate name, and at least **three** appropriate columns such that:
  - each column can have a data type of your choice
  - give each column a descriptive name
  - make sure one of the columns should be unique for each item and make it the primary key using:

    `primary key (`*chosen_col_name*`)`

  CREATE this in SQL:
  - Write a `drop table` statement for your table, including a `cascade constraints` clause.
  - Write a `create table` statement for your table.
  - Be sure to include an appropriate `primary key` clause for your table!
- Insert at least **6** appropriate rows into your table.
- Write a `select` statement that will display your table's contents.
- Turn **off** spooling.

  ```
  spool off
  ```

- When you believe your SQL script is working properly (or at the end of lab, whichever comes first), the "driver" submit the pair's/trio's `325lab1.sql` and `325lab1-out.txt` files using

  `~st10/325submit` with a homework number of **81**.
  - (Once your pair's/trio's lab exercise files have been submitted, you may leave lab if you wish. Or, you can ask questions, read SQL Reading Packet 1 and/or DB Reading Packet 1, etc. But note that questions about today's lab exercise will get first priority.)