CS 325 - Week 4 Lab Exercise

Deadline

Due by the end of lab on 2021-09-17.

Purpose

To practice writing some of the "classic" relational operations in SQL.

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 **84**

Important notes

- 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.
  - DB Reading Packet 3 - Intro to the Relational Model
  - SQL Reading Packet 2 - Writing Relational Operations using SQL
- 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 `325lab4`, protect it, and go to it:

```
mkdir 325lab4
chmod 700 325lab4
cd 325lab4
```

- **COPY** the following SQL script to your `325lab4` directory by typing this Linux command at the nrs-projects prompt:

```
cp ~st10/set-up-ex-tbls.sql .  # remember the space and period
```

  ...and run the SQL script in `sqlplus` to get your own tables `empl`, `dept`, and `customer`

- Then, begin a SQL script **`325lab4.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 **`325lab4-out.txt`**.

- Use one or more **`prompt`** commands to print a message to the screen containing **both (all)** of your names
- **IMPORTANT NOTE:** We will definitely be writing SQL `select` statements that do combinations of relational operations -- but for today's lab, I want to see if you can write SQL `select` statements that specifically do *just* a particular relational operation, to help you remember the underlying basic relational operations.

Problem 1

Remember: a **relational selection** operation selects **just the specified rows** from a table (just the specified tuples from a relation). It should include *all* of the attributes for each of the rows meeting the stated criteria.

- Write a `prompt` command printing to the screen: **`true relational selections`**
  - Then, write **at least TWO different** SQL `select` statements that each result in a **true relational selection** of your choice from one of the tables `empl`, `dept`, or `customer` .
  - Make sure each has at least one row in its results.

Problem 2

Remember: a **relational projection** operation grabs **just the specified columns/attributes** from a relation, and eliminates any duplicate rows in what results, so that the result will still be a relation.

- Write a `prompt` command printing to the screen: **`true relational projections`**
  - Then, write **at least TWO different** SQL `select` statements that each result in a **true relational projection** of your choice from one of the tables `empl`, `dept`, or `customer`
  - (The SQL `select` statement actually supports both "true" relational projection, and projection that can result in duplicate rows.
    - For this problem's `select` statements, you are expected to **include the SQL keyword that eliminates any duplicate rows in the results**, even if your particular projection might not happen to have any duplicate rows.)
  - Make sure each has at least one row in its results.

Problem 3

Remember: a **Cartesian product** operation on two relations concatenates every tuple/row from one relation to every tuple/row from another relation, forming a third relation. That is, it produces a relation consisting of all possible pairs of rows from two relations.

(No, by itself, this is not particularly useful! But it is, *conceptually* at least, part of the process of an equi-join operation, so it is good to know about. Plus, it is *far* too easy to write one of these accidentally, when you do not mean it, if you are not aware of the difference between this operation and the equi-join operation.) So...

- Write a `prompt` command printing to the screen: **`true Cartesian product`**
  - Then, write at least one SQL `select` statement that results in a **Cartesian product** of your choice of two of the tables `empl`, `dept`, and `customer`

Problem 4

Remember: an **equi-join** operation, at least conceptually, performs a Cartesian product of two tables, and then from that product selects only those rows in which a column or columns from one of those tables has the same value as a column or columns from the other of those tables. (This **join condition** is an equality for an **equi-**join -- get it?)

(Typically, it is expected that the column/columns being joined upon have the same domain(s) -- most frequently, the column/columns are a foreign key in one of the tables and the primary key being referenced in the other of the tables.)

- Write a `prompt` command printing to the screen: **`true equi-join of customer and empl`**
  - Then, write a SQL `select` statement resulting in a **true equi-join** of the `customer` and `empl` tables, such that the resulting rows have, for each customer, all of the customer's attributes, along with all of the attributes of that customer's employee rep.
  - (Hint 1: be careful -- what attributes have the **same domain** in these two tables, and will give this desired result? What should be join condition be, to get this result?)
  - (Hint 2: if you have done this correctly, your result will have three rows and seventeen (!!) attributes per row!)

- Now, turn **off** spooling.
- When you believe your SQL script is working properly, submit your `325lab4.sql` and `325lab4-out.txt` files using `~st10/325submit` with a homework number of `84`.
  - (Once you have submitted your lab exercise files, you may leave lab if you wish. Or, you can ask me questions, (noting that lab-exercise-related questions need to receive 1st priority), work on the CS 325 homework, etc.)

When you are done with the four problems above (or at the end of lab, whichever comes first), submit your files using `~st10/325submit` with a lab number of **84**:

- **`325lab4.sql`**
- **`325lab4-out.txt`**
- (and note that `~st10/325submit` will also grab your copy of `set-up-ex-tbls.sql`, and that's fine!)
  - (Once your pair's/trio's lab exercise files have been submitted, you may leave lab if you wish. Or, you can ask questions about the reading packets, etc. But note that questions about today's lab exercise will get first priority.)