

Assignment: SQL #0

Avraham Leff

COM3580: Fall 2020

1 Assignment-Specific Packaging

The general packaging is unchanged from the basic “Homework Requirements” (see slides from first lecture and “Homework Policies for COM 3563” on Piazza).

This assignment’s “DIR” **must be named** *SQL0*. Your report **must be named** *\$DIR/SQL0.pdf*.

That last character is a zero, not an “o” ☺

2 Motivation

Yes, this should be an easy drill: but we need to get you started with

- Formulating SQL queries from natural language specifications
- Executing those queries from the POSTGRES CLI

3 Advice

Because of the large number of SQL topics, and especially because they don’t always have a natural progression that builds from one to another, I will be assigning drill exercises on material that we may not have fully

covered in lecture. I **expect that you've read (or at least skimmed)** the textbook (or equivalent material), so that you know where to look for the knowledge to solve these exercises.

This exercise relates to “early” Chapter 3 material.

4 Requirements

Exercise 3.16 references the *employee* relational database below, where the primary keys are underlined.

employee (ID, *person_name*, *street*, *city*)
works (ID, *company_name*, *salary*)
company (*company_name*, city)
manages (ID, manager_id)

- Download the SQL “population” script [SQL1_Populate.sql](#). Use that script to create and populate the database tables per the above schema.

Note: in all such exercises, you are responsible for (re)initializing the database to this initial state. All subsequent interactions with the databases must execute the specified steps, **in the specified order!**

- Note: it is perfectly ok if your query returns the empty relation!
- For each of the specified *natural language* queries, you must
 1. Show me a formatted SQL “translation” of that query
 2. Show a **clear snapshot** of your POSTGRESQL executing that query and the results therefrom.

3. If the result-set contains *more than three tuples*, you **must** append the clause `ORDER BY ID ASC LIMIT 3` to your query to show only the first three tuples.
- Note: most, if not all, your grade depends on “correctness”: i.e., the result of your query!

4.1 The Problems To Be Solved

1. The DDL used in the “*populate employee*” script is minimal. As you can see, the only integrity constraint used is a *primary key* constraint.

You’re already aware of the importance of *foreign key* constraints. For all relevant table schema, list the enhanced executable-ready DDL, including your new foreign key constraints. For brevity, omit the non-foreign-key DDL: show only the table name and the foreign key constraints. You will be graded on both too many, and too few, constraints.

Note: this exercise is about understanding the purpose of a foreign key constraint, and then using some imagination as to the scenario that Figure 3.19 is modeling.

2. Exercise 3.16(a)
3. Exercise 3.16(b)
4. *Using the university schema, write an SQL query to find the name and ID of Physics students advised by an instructor in the Physics department. Order the results by ascending student name.*
5. *Using the university schema, write an SQL query to find the names of those departments whose budget is higher than that of History. List them in alphabetic order.*
6. Exercise 3.33: order the results by ascending *course id*.