In-class activities: Sep 25, 2017

Activities and group work this week function the same way as our previous activity. We recommend that you continue working with the same 3-person group. We suggest that you switch who within your group operates the laptop throughout class time, so you can all get hands-on practice.

We know that you may not have completed some of the activity steps from last week and that’s OK. You will complete some of those in conjunction with today’s activities. Today’s activity sheet refers back to last week's document and prompts you to complete a particular step or steps from there before you continue with certain steps here.

Pacing:

In a flipped classroom, the activities give you some structure, but the objective is that everyone can adjust the pace according to their needs. That means that you are in charge of your pace. However, we urge you to maintain focus and efficiency through the activities to ensure that you don’t fall behind.

We do not expect you to finish all the activities on this sheet today, as the queries in steps 5 and 7 are quite hard. Next week, we will have work wrap-up, which means that you will all get a chance to work through the problems you didn’t get to.

Today, we expect all of you to finish at least steps 1, 2, and 3 from this sheet, and steps 6 and 7 from last week’s sheet for those who have not completed them already. However, we hope you will get farther, so that you are not left with too much work to cover next week.

Quiz:

There will be a short end-of-class quiz. As before, you should indicate your 3-person team on the quiz. We keep track of that based on table and laptop numbers. Note your table number, indicated on the whiteboard and monitor assigned to your table. There are two laptop numbers, one on yellow sticker and one on white sticker. Indicate both if possible, but at least one is necessary. If you cannot find these numbers, please talk to the instructors or TAs before you turn in your quiz.
Step 1: Load the data for the activity

This week, we will work on our practicedb database, but we have modified the data slightly. We will drop the previous database, re-create it, and load the new data. Don’t skip this step, as someone may have modified the database as was created last week on the machine you are working on, and some of the exercises may not work as intended.

Start postgres and connect to psql (instructions on Moodle). Type \c to check what database you are connected to. You should be connected to a database other than practicedb. If you are connected to practicedb, connect to any other database with \c <database_name>

Drop the previous practicedb database with the following command:

DROP DATABASE practicedb;

Then recreate it:

CREATE DATABASE practicedb;

Now connect to the practicedb database:

\c practicedb

Open the file practiceData2.sql, from this week’s schedule and copy-paste the entire contents into your prompt. Check that you didn’t get any errors and the proper tables are created.
Step 2: Advanced join practice

*If you have not completed (or you feel you need to practice again) steps 6 and 7 of last week’s activity, do so now.*

Joins are very important operations in databases and you need a lot of practice with them. Writing join queries can get complicated, and some of these may take you some time to figure out. Don’t hesitate to ask other teams for help and feedback, and if you feel stuck, talk to the TAs and instructors.

Write a query to answer each of the following questions:

1. Which country or countries have sold products during the month of June?
   *Hint: Note that there is a new table in your database (Sales) that records the number of each product item sold each month. How many tables do you need to join for this query?*

2. Return a list of names of all employees who work on two distinct projects.
   *Hint: This query is similar to the 2nd query from last week's step 6: team-to-table.*

3. List all employee names and the projects that each is involved in. Do not omit any employees even if they don’t have a project.
   *Hint: This query is similar to a query from last week’s step 6, but not exactly the same. This needs a different type of join.*

4. Return a list of all employee names and the names of their managers. Do not omit any employees even if they have no manager.
   *Hint: This query is similar to a query from last week’s step 6, but not exactly the same. This needs a different type of join.*

***Team-to-table***

Reflect on how you determine how to write these joins. How do you determine how many tables you need to join? How do you determine if do you need a self-join? When do you need an outer join?
Step 3: Aggregation

We will practice aggregate queries. These aggregates are over a single table (i.e., they don't involve joins).

Write the following queries:

1. Compute the average price of all products.

2. Compute the average product price for each category of products. (You should write a single query that does that, not a separate query for each category!)

3. Compute the average price of all products cheaper than $150, for each category of products.

4. Compute the average price of products cheaper than $150, for each category, but only keep the categories where the maximum price of any product in this category is greater than $80.

5. Compute the average price of products cheaper than $150, for each category, but only keep the categories that start with the letter ‘G’.

*** Team-to-table ***

Check your understanding of the HAVING clause. Can you explain which conditions should go to the HAVING clause vs which conditions should go to the WHERE clause? Think about queries 4 and 5 above. Do your queries have a HAVING clause? If they do, can the condition(s) of the HAVING clause move to the WHERE clause? If they do not have a HAVING clause, can a WHERE condition move to the HAVING clause?
Step 4: More complex queries – Aggregations and joins

In this step, our queries will get more complex, as you need to combine aggregation and joins.

1. How many products have been sold from the category ‘Gadgets’? (Sales are recorded in the Sales table.)

2. What is the total revenue made by sales of ‘Gadgets’ during the month of December?

3. For each employee, list how many of projects they are involved in. Check that you are reporting the number of projects for all employees and that the counts are correct.

*** Team-to-table ***

While some aggregates are simple (e.g., finding the price of the cheapest product), finding witnesses for these aggregates (e.g., which is the cheapest product) requires more complicated, typically nested queries. Work with the teams at your table to solve these harder queries:

1. For each product, find the month during which the product had the most sales out of the year. For example, if out of all the months of the year, Gizmo had the most sales in February, then the tuple (Gizmo, February) should be in the result.

2. Which company makes the most products?

3. Find the country that produces the cheapest product.

4. Find the employee with the most projects.
**Step 5: Advanced practice on real-world data**

Connect to the IMDB database on your system. You can see all available datasets with the command `\l`.

You can see the tables in this dataset with the `\d` command. This dataset has the following tables:

- **ACTOR** (id, fname, lname, gender)
- **MOVIE** (id, name, year)
- **DIRECTORS** (id, fname, lname)
- **CASTS** (pid, mid, role)
- **MOVIE_DIRECTORS** (did, mid)
- **GENRE** (mid, genre)

You will be working on queries of varying difficulty on this dataset.

*If you have not completed (or you feel you need to practice again) step 9 of last week’s activity, do so now. Once you are done, continue with the queries on the following page.*

These are all challenging queries, so we suggest checking your work with the other teams at your table for all of them.
1. List all directors who directed 30 or more thrillers, in descending order of the number of thrillers they directed. Return the directors’ names and the number of thriller movies each of them directed.

2. We want to find actors that played five or more roles in the same movie during the year 2015. Notice that CASTS may have occasional duplicates, but we are not interested in these: we want actors that had five or more distinct roles in the same movie in the year 2015.
   a. Write a query that returns the actors’ names, the movie name, and the number of distinct roles that they played in that movie (which will be ≥ 5).
   b. Write a query that returns the actors’ names, the movie name, and all the distinct roles (five or more) that the actor played in that movie.

3. [Hard query] For simplicity, the IMDB dataset lists an actor’s gender as M or F. (The dataset doesn’t provision for non-binary actors, and the questions below are based on this assumption.)
   a. For each year, count the number of movies in that year that had only female actors. For movies where no one was casted, you can decide whether to consider them female-only or male-only.
   b. Now make a small change: for each year, report the percentage of movies with only female actors made that year, and also the total number of movies made that year. For example, one answer may be:

   1990 31.81 13522

   meaning that in 1990 there were 13,522 movies, and 31.81% had only female actors. You do not need to round your answer.
Step 6: Wrapping up

Complete steps 8 and 10, and any other steps you may have omitted from the previous week’s activity sheet. If you have additional time, work on the harder queries on step 7 of the following page.
Step 7: Harder queries [OPTIONAL]

This step is optional. We suggest that you work on it only after you complete all other steps.

First, note that these queries are quite complex, and may take a while to run. Think together with your teammates how you can work through them. One possibility is to create a smaller, sample dataset, which can help you write and debug the queries. This can be a bit of a challenge in itself!

1. Which is the film (or films) with the largest cast? Return the movie title and the size of the cast. By "cast size" we mean the number of distinct actors that played in that movie: if an actor played multiple roles, or if the actor is simply listed more than once in CASTS, we still count her/him only once.

2. Which decade has the largest number of films? We consider that a decade is a sequence of any 10 consecutive years. For example 1965, 1966, ..., 1974 is a decade, and so is 1967, 1968, ..., 1976.

3. How many actors have Bacon number 2? The Bacon number of an actor is the length of the shortest path between the actor and Kevin Bacon in the "co-acting" graph. That is, Kevin Bacon has Bacon number 0; all actors who acted in the same film as KB have Bacon number 1; all actors who acted in the same film as some actor with Bacon number 1 have Bacon number 2, etc.

To ponder: how can you compute the Bacon number of each actor in the database?