Exercise Week 1: Relational Databases

Introduction to Database Systems
Eleni Tzirita Zacharatou and Omar Shahbaz Khan

Spring 2024

PostgreSQL Installation

To run today's exercise, as well as many of the other exercises and assignments of the course, you must install the PostgreSQL database system (postgresql.org) on your laptop. Installation guides that should work for 90+% of the students have been posted on Piazza/LearnIT. If you have problems, ask your TAs during the exercise session or on Piazza.

1 Using PostgreSQL

Download the Games database scripts from LearnIT within Lecture 1. Start by following the instructions in the 01-commands.txt file. Create a database, run the database install script using psql, and then run the queries, both using pgAdmin and psql. Read through the queries and see whether you can make sense of them.
Note: Depending on your system and installation, you might need to adjust the commands in the 01-commands.txt file. You can consult the Useful Commands Cheatsheet on LearnIT for help.

2 Your First Relational Database

Note 1: All the code needed for this exercise is in the ***Exercise Week 1: Examples*** *slides that you can find on LearnIT within Lecture 1. The emphasis here is on working with the database software. You should use this exercise to play with the SQL statements as much as you can.*

Consider the sample Coffee database presented in the above slides. Your task is to create part of this database. Write SQL commands to CREATE the following tables:

Coffees(<u>name</u>, manufacturer)

Coffeehouses(<u>name</u>, address, license)

Sells(<u>coffeehouse</u>, <u>coffee</u>, price)

The underlined columns should form the primary keys of the relations. Note that there should also be foreign key relationships in the definition of the last table to the corresponding columns in the first two tables. Slides 11-12 have commands for two of the three tables.

Maintain all the SQL commands in a script file, which you can run repeatedly using psql. Before creating a table, always drop the table if it exists (see slide 2). With foreign keys, you need to be careful about the order in which you drop tables (try dropping tables in the incorrect order!). Run this script from the command prompt. In the final exam, you must be able to do this, so you might as well start early. Ask the TAs for help, if needed.

Once you have the basic database, try the following:

1. Write SQL commands to INSERT some data into the tables (see slide 17). Maintain these commands in a separate SQL file.

2. Try creating some simple SQL queries to SELECT data. You can look at the queries in the Games database query script from Part 1 for inspiration. Maintain these commands in a separate SQL file.

3. Modify the database with some more advanced constraints (see slide 15). Then modify your INSERT and SELECT statements to test whether these constraints work.

Note 2: Feel free to be creative with the last part, even to the point of creating some strange tables and test data. The more features you play with, the better.

3 A Simple Database—With and Without a DBMS

Note 1: If you do many experiments with the coffee database above, you may not have much time for this last exercise. It is still a worthwhile thought experiment to do when you have time. You can do this exercise in pairs or small groups.

Suppose that you must implement a system that contains information about food in a canteen. The following information should be stored:

- Information on the dishes that can be bought. Each dish has a name and a price.

- Every day, the dishes available for sale are recorded, along with the quantity of each dish sold.

- There is a list of ingredients, each having a name and a supplier. For each dish, the ingredients used and their respective quantities are recorded.

a) How would you store this information *without using a DBMS?* Sketch an implementation in a language such as Java (or any other object-oriented, imperative language). Consider that the implementation should be flexible and e.g. allow changing the milk supplier. You are not expected to write any code, just give an overall design.

You can imagine many queries on such a database, e.g.:

- Find all dishes that contain eggs.
- Find the total sales amount (in DKK) of today.
- Find the most sold dish today.

b) How would you implement these queries? Sketch a solution. Especially for students who have taken the algorithms class: Think about how the solution scales to large data sets. Can you avoid repeated data traversals?

c) Create a relational data model for the database. Write down the database schema.

Note 2: Over the next two weeks, we will explore how to perform the queries above with little effort using a DBMS and achieve high performance. You may revisit this exercise towards the end of the class to evaluate what you have learned.