Mini Project 1: Automation for Query Development and Execution

Project Start: Tuesday, 29.10.2024

Project End: Thursday, 21.11.2024

Task 0: Setup

This mini project focuses on query development for a relational database. For your convenience, we have created a virtual machine containing a sample database that you can use to test your queries. It is available under `https://nextcloud.hpi.de/s/zPLkDdLHeoaALtr` (password: "ase2024-p1").

To run the virtual machine:

1. Install Oracle VirtualBox (`https://www.virtualbox.org/`)
2. Run VirtualBox and select "File" → "Import Appliance..."
3. Select the file "ASE2024-P1.ova" from the Nextcloud share as the source file and hit "Finish"
4. After the import has finished, select the virtual machine from the list on the left and hit "Start"
5. Use the user "ase2024" (password: "ase2024") for login

The virtual machine has an installation of PostgreSQL (see `https://www.postgresql.org/`). A database named "ase2024db" has already been created and populated with synthetic data created by the data generator of a social-network-themed querying benchmark (documentation available under `https://ldbcouncil.org/ldbc_snb_docs/ldbc-snb-specification.pdf`). To access the database and start querying, you can execute the command "psql ase2024db" on the terminal. The database schema is available under "/home/ase2024/Project-1/schema.sql" (in the virtual machine).

If you prefer to set up a PostgreSQL database yourself instead of using the virtual machine, you can use the files provided in the archive "ase2024-p1-files.zip" from the Nextcloud share. The folder "social_network-sf0.1-CsvMergeForeign-StringDateFormatter" contains some test data, which can be loaded into a database via the scripts "create_tables.sql" and "load.sql". Note that this may require adjusting the paths to the test data in "load.sql". The archive also contains the database schema in "schema.sql" (only for inspection, does not need to be run).

Task 1: Query Development

The following query for the schema in "/home/ase2024/Project-1/schema.sql" on the virtual machine shall be implemented in SQL:

> "Find all messages longer than 100 characters that have been liked by at least 20 people. Only consider messages where at least half of the likes are from people that are foreign to the message's creator, with a foreigner being a person that neither the creator nor any of their friends know. For each message, return the message's id, the total number of likes and the number of foreign likes. Sort the returned messages by the ratio of foreign likes from highest to lowest."

Subtasks:

1. Implement the query *without* consulting an LLM (such as ChatGPT) for assistance. Usage of other tools/resources is allowed. You can test your query over a sample database containing a small, synthetically generated social network by executing the command "`psql ase2024db -f <your-query-file>`" from the terminal in the provided virtual machine.

2. Use an LLM (such as ChatGPT, `https://chatgpt.com/`) to generate an implementation of the above query from the natural language description. Check the result for correctness and try to guide the LLM to produce a correct implementation via prompting if necessary.

3. Reflect on the process of query development from subtasks 1 and 2. Questions you may ask yourselves include (but are obviously not limited to): How much experience in query development did the team have? Which steps did the query development involve in each subtask? How time-consuming and difficult was the development? How was the quality of the resulting implementation? How helpful was the LLM in general and the additional explanations it provided (if any)? How often did you have to prompt the LLM and were there any misunderstandings (on your side or the LLM's)? What degree of automation did the tools you used for query development achieve on their own and in combination?

Task 2: Query Extension

The query from Task 1 shall now be extended as follows:

> "For each message, also return the full name of the first foreign person who liked the message and the time when they created the like."

Subtasks:

1. Manually adjust both query implementations developed in Task 1 to accommodate the extension above, *without* consulting an LLM for assistance. Usage of other tools/resources is allowed.
2. Use an LLM to automatically perform the extension for both query implementations from Task 1. Check the result for correctness and try to guide the LLM to produce a correct implementation via prompting if necessary.
3. Reflect on the process of query development from subtasks 1 and 2. In addition to the questions from Task 1, you may think about how automation as provided by the LLM could be integrated holistically into an iterative query development process.

Task 3: Query Execution

The execution of the query developed in Tasks 1 and 2 via PostgreSQL shall now be investigated in some detail. The `EXPLAIN` keyword can be used to instruct PostgreSQL to output an execution plan for an SQL statement instead of executing it directly (see `https://www.postgresql.org/docs/current/sql-explain.html`). Several web-based tools offer capabilities for visualizing such query plans, e.g. `https://explain.dalibo.com/`.

Subtasks:

1. Select one of the query implementations from Task 2. Generate an execution plan for the query using PostgreSQL's `EXPLAIN` keyword and try to develop a basic understanding of the generated plan, potentially using the visualization provided by another tool. In addition, ask an LLM to generate an execution plan in the same format.
2. Reflect on the output of the LLM in comparison to the plan created by PostgreSQL. Questions you may ask yourselves include (but are obviously not limited to): Is the execution plan generated by the LLM plausible? How helpful were additional explanations provided by the LLM (if any)? To what degree is query execution already automated in PostgreSQL? Are there any fundamental differences between the query development tasks from Tasks 1 and 2 and the creation of a query execution plan that may have implications regarding automation?