Homework 4: Text-to-SQL

1. Design an SQL schema for a sqlite database for blocks and transactions. Include in the schema all the data elements in the *getblocks RPC call when verbosity=2* is specified.
    a. **Bonus Points**: Find a way to auto generate the SQL schema from the JSON schema returned by the RPC call.
        i. If it doesn't work out of the box fully automatically, there should be ways to get it to do 99% of the work and some small human-made adjustments.
        ii. Possible approaches:
            1. Put an example of getblock JSON object in the prompt and ask the LLM to directly generate a SQL schema to encode the JSON object:
                a. You'll likely need to play around with the prompt to get this to work right, if it's even possible. Read this entire prompt engineering guide carefully and write some small examples to build intuition for writing good prompts:
                    i. https://www.promptingguide.ai/
            2. Ask the LLM to write code to do this mapping.
                a. Instead of "Give me a SQL schema for this JSON object.", ask it the following:
                    i. "Write code to auto-generate a SQL schema from a JSON object".
2. Write a program that keeps the database updated with the latest blocks.
    a. The program uses RPC calls to extract the latest blocks and transactions and then
    b. Then, the program transforms them into SQL statements that are run on the database.
        i. **Bonus Points**: Find a way to use LLMs to do this mapping, but your approach must be guaranteed to be 100% correct all the time.
    c. This program should be scheduled to run every few minutes.
    d. Take necessary steps to ensure that the data stored in the database is perfectly correct, always up to date, and consistent.
3. Write a program to accept natural language questions from a user and give back the answer according to your SQL database.
    a. Take the OpenAI API python file you wrote as a starting point.
    b. Change it so that it accepts these inputs:
        i. a natural language questions
        ii. an absolute path to the sqlite database.
    c. Generate the prompt:
        i. Set the system prompt: "You are a SQL developer that is expert in Bitcoin and you answer natural language questions about the bitcoind database

in a sqlite database. You always only respond with SQL statements that are correct.”.

1. Automatically extract the SQL schema from the database and then concatenate it with the natural language question.

ii. Note: The purpose of the system vs user role designation is to separate out the description of the task from an instance of the task.

1. In our case:
    a. The task is text_to_sql on a specific database for bitcoind.
    b. The instance of the task is “how many blocks are there?”

4. Write a minimum 10 test cases of varying difficulty that show how your system is able to convert natural language questions bitcoin data into an exact answer from executing on the database.
    a. Ensure that your test cases are comprehensive enough to build confidence that your solution is correct**. Significant points deducted for insufficient test cases.**
    b. A test case is a triple:
        i. natural language question
        ii. a sql statement that generates the correct answer.
        iii. the actual answer from executing on the db.
5. Write three very hard test cases that are so hard that the system is not able to answer correctly. The purpose of this is to find the limit of what’s possible for this task.
    a. A test case is a triple:
        i. natural language question
        ii. an expected sql statement that generates the correct answer.
        iii. the expected answer from executing expected sql on the db.
        iv. the incorrect sql statement that your system currently generates.
        v. the incorrect answer from executing incorrect sql on the db.
    b. Prepare a slide to present these three test cases to the class
    c. Note: See the SpiderV2 leaderboard for text_to_sql for an example of SQL queries that are so complex and large that the state of the art (SOTA) text_to_sql is not able to solve:
        i. https://spider2-sql.github.io/