CSE 111 – DATABASE SYSTEMS
Lab 8

In this lab session, you will learn how to create indexes for a query workload by using the recommendations of a database auto-tuner. Specifically, you have to create indexes for the queries in Lab 3 based on the recommendations provided by the `SQLite Expert`. To achieve this, you have to use the `.expert` command from `SQLite`. When applied to a query, `.expert` provides index suggestions to make the query run optimally. `.expert` does not create the suggested indexes. This is the responsibility of the user.

In order to complete the lab you have to perform the following tasks:

1. Execute the queries from Lab 3, whose `SQL` statements are provided in the files `test/x.sql`, where `x` is the number of the query. In addition to the `SQL` statement, these files activate the query analyzer `.eqp`, which displays the query execution plan. Since there are no indexes in the database, all the queries require table scan and/or automatic index creation.

2. For every query going from 1 to 15, invoke the `.expert` command to get the optimal index recommendation. Then, create the first suggested index with the name pattern `table_idx_attribute1_attribute2`, e.g., `lineitem_idx_l_quantity`.

3. Execute the queries from Lab 3 again. This time, the query execution plans have to include the newly created index.

4. The important part here is that you create one index at a time and then rerun all the queries to see how their plans change and what new indexes are recommended. As long as there are still indexes recommended, you go back to step 2.

5. Add every `CREATE INDEX` statement you use to the `create-index.sql` file, in the order in which you determine them. This is the only file you are required to edit. The file has to include all the index creation statements recommended by the auto-tuner.

6. You can check the correctness of your index creation by executing the command `./test.sh` in the terminal. The expected output is available in `results/x.res`, where `x` is the number of the query. The output produced by your code is available in `output/x.out`. They have to match exactly for every query, e.g., `1.res` has to match with `1.out`. Notice that the match has to be in the query execution plan since you do not write the `SQL` statement. The expected plan uses indexes, which have to have exactly the same name as in the `res` file. There may be formatting differences between the query plan printed on your local machine and the expected plan. The final check is done on our machine, so follow that format.

You have to submit only the `create-index.sql` file with the `CREATE INDEX` statements. We will run this file against a database that does not have any indexes.