Lab 2: Operators

Release Date: Tuesday September 24, 2024

Checkpoint Due date: To Be Announced

Due Date: Monday Oct 21, 2024 by 11:59 pm ET

1 Introduction

In this lab assignment, you will write a set of operators for GoDB to implement table modifications (e.g., insert and delete records), filters, joins, aggregates, etc. These will build on top of the foundation that you wrote in Lab 1 to provide you with a database system that can perform simple queries over multiple tables.

You do not need to implement transactions or locking in this lab. You will find detailed instructions on specific tasks you need to complete in `lab2.md` in the course GoDB repository.

2 Getting Started

In your local repository, first add and commit your code for lab 1.

```
git add --all
```

```
git commit "Lab 1 complete"
```

Then you can pull the new lab with

```
git pull upstream main
```

The files we provided to you in lab 1 that are updated include `catalog.go`, `parser_test.go`, `table_stats_lab1.go` in `godb`, and `go.mod` in the root directory of the repo. Furthermore, some `.dat` and `.txt` files are added, deleted, or moved around. **You should not see any merge conflicts in any lab code.** If you made changes to them and do see merge conflicts, please accept the new changes we shipped for lab2 when you are resolving the merge conflict. There may be merge conflicts on the `.sum` and `.mod` files. Please accept the change you just pulled and fetch necessary libraries with `go get` as instructed in the command line when you run `go test`. Reach out to the course staff if you are confused about this step. Now, you can sync the changes to your own remote repository if applicable:

```
git push origin main
```

3 Collaboration

If you are working with a partner, lease try to stay with the same person you worked with for lab 1 for this lab and future labs. Since you share a codebase for lab 1, it is nuanced to switch partners since later labs are supposed to build on earlier ones. If there is an exceptional circumstance and you must switch the partnership arrangement, please come talk to the course staff so we can figure out the arrangement together.

4 Submitting Your Work

After you finish all the exercises, you will need to submit

1. All the code of the lab.
2. A short lab report writeup.

We will be using Gradescope to autograde the code of the labs. Check Piazza for the access code for Gradescope. If you are still unable to join the course in Gradescope, please reach out the course staff. You may submit your code multiple times before the deadline; we will grade the latest version as determined by Gradescope. Place the write-up in a file called **lab2-writeup.txt** with your submission.

If you are working with a partner, only one person needs to submit to Gradescope. However, make sure to add the other person to your group. Also note that each member must have their own lab report and work on it individually. Please add your Kerberos username to the file name and in the writeup itself (e.g., `lab2-writeup-username1.txt` and `lab2-writeup-username2.txt`).

The easiest way to submit to Gradescope is uploading `.zip` files containing both your code and the lab report writeups. On Linux/MacOS, you can do so by running the following command from the parent directory of your local repository:

```
zip -r submission.zip godb/ lab2-writeup.txt
```

If you are working with a partner:

```
zip -r submission.zip godb/ lab2-writeup-username1.txt lab2-writeup-username2.txt
```

This year, we also ask you to make a *checkpoint submission*, detailed in subsection 4.2.

4.1 Lab Report

Your lab report should be brief (maximum of 2 pages) and contain the collowing

- Describe any design decisions you made.
- Discuss and justify any changes you made to the API.
- Describe any missing or incomplete elements of your code.
- Describe how long you spent on the lab, and whether there was anything you found particularly difficult or confusing.

4.2 Checkpoint submission

This year, to encourage you to start the lab early, we are setting up a checkpoint submission. To complete the checkpoint submission, upload your progress to the same Gradescope assignment. Your submission should pass 15 points worth of the testcases on the autograder. Moreover, the checkpoint submission should contain a preliminary lab report writeup. You only need to report if there is anything you find particulatly difficult or confusing so far in this preliminary writeup (or just say ”everything is good so far”). A few days after the checkpoint submission, we will host a lab bootcamp to review relevant concepts and address issues that people commonly have trouble with. This checkpoint submission is so that you can get the most from the bootcamp and we have plenty of time to help you with the lab.

We will announce the due date for the checkpoint submission very soon, once we determine the timing of the bootcamp and other class activities.

4.3 Submitting a bug

Please submit bug reports to 6.5830-staff@mit.edu or via a private Piazza post. When you do, please try to include:

- A description of the bug.
- A `.go` file with test functions that we can drop into the ‘godb‘ directory, compile, and run.
- A `.txt` file with the data that reproduces the bug.

If you are the first person to report a particular bug in GoDB, we will give you a (large) sweet treat!

5 Grading

75% of your grade will be based on whether or not your code passes the system test suite on autograder. These tests will be a superset of the tests we have provided. Before handing in your code, you should make sure it produces no errors (passes all of the local tests) when you run ‘go test‘ in the ‘godb‘ directory.

Before testing, Gradescope will replace the go test files with our version of these files. This means you should make sure that your code passes the unmodified tests.

You should get immediate feedback and error outputs for failed visible tests (if any) from Gradescope after submission. There may exist several hidden tests (a small percentage) that will not be visible until after the deadline. The score given will be your grade for the autograded portion of the assignment. An additional 25% of your grade will be based on the quality of your writeup and our subjective evaluation of your code. This part will also be published on Gradescope after we finish grading your assignment.

6 Lab 2

You will need to fill in pieces of code that are not implemented. For this lab, you will need to implement operators for this lab. You may find the lecture and reading material on database operators particularly useful. `lab2.md` in the course repository walks you through the code and the tasks you will need to complete.

7 Getting Help

If at any point you need help with with lab, feel free to post on Piazza or reach out to one of the TAs or the instructor. Their contact information can be found on the course homepage. There will be office hours and a bootcamp session where we review the concepts covered in the lab and common problems people run into together. Look out on Piazza and the course website for these plans.