# Report Homework Assignment 2: COP 6930 Special topics: Blockchain

Naman Arora
*UFID: 3979-0439*
*University of Florida*
naman.arora@ufl.edu

***Abstract*—This document is indented to serve as the report for completed homework assignment that it is attached with. The report goes through the justification of some of the choices made to complete the assignment, then through the procedure of running the code attached to extract results similar to the ones provided in expected output section at the end.**

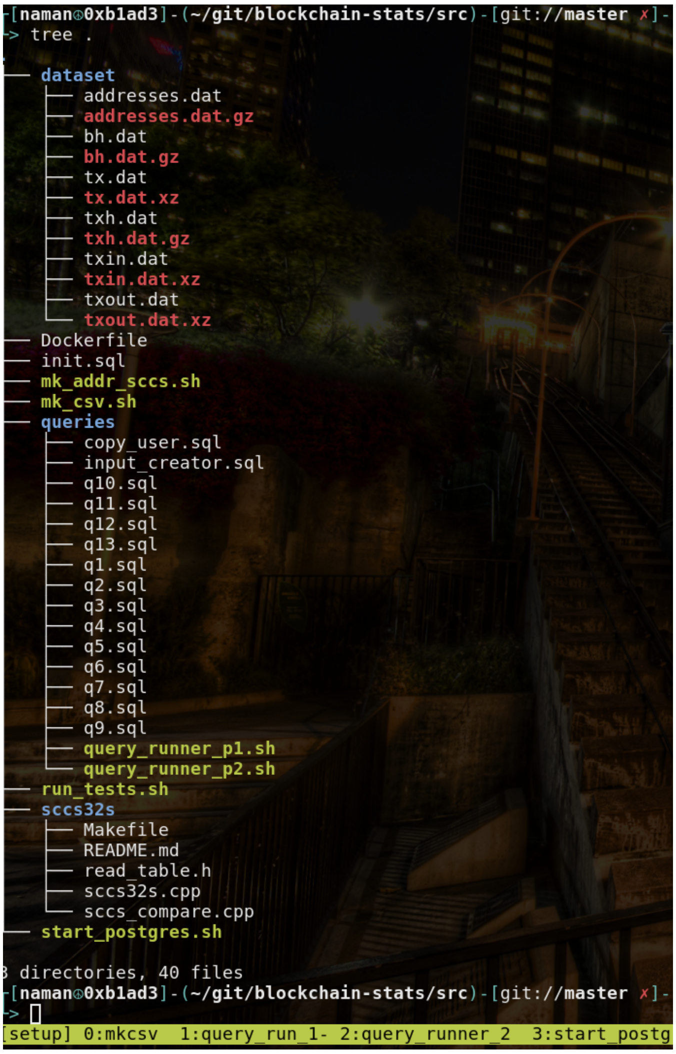
## I. Project Dependencies

- PostgreSQL: The postgreSQL database is a very well respected and high preforming open source database query server. The kind of database provided for analysis was a good fit for conventional SQL based analysis.
- Docker: In the event of unavailability of the postgreSQL database server at hand, the repository is designed to run in dockerized container mode. A nice workflow is created with the help of helper scripts and is described in the next section.
- sccs32s: This project is leveraged to create a new addr_sccs.dat file with joint+serial control.

## II. Directory Structure

Since the repository is designed to run in both native and dockerized mode, there are a lot of helper scripts which demand explanation. After the explanation of the directory structure, both types of work flows will be introduced.

- the src/dataset directory: The project expects all the *.dat files to be present in this directory.
- the src/queries direcotry: This directory contains all the queries, one for every corresponding question in the assignment document along with two runner scripts, one for each part. All the query files have explanations within them in the form of comments.



```
-[naman☮0xb1ad3]-(~/git/blockchain-stats/src)-[git://master ✗]-
-> tree .

── dataset
   ├── addresses.dat
   ├── addresses.dat.gz
   ├── bh.dat
   ├── bh.dat.gz
   ├── tx.dat
   ├── tx.dat.xz
   ├── txh.dat
   ├── txh.dat.gz
   ├── txin.dat
   ├── txin.dat.xz
   ├── txout.dat
   └── txout.dat.xz
── Dockerfile
── init.sql
── mk_addr_sccs.sh
── mk_csv.sh
── queries
   ├── copy_user.sql
   ├── input_creator.sql
   ├── q10.sql
   ├── q11.sql
   ├── q12.sql
   ├── q13.sql
   ├── q1.sql
   ├── q2.sql
   ├── q3.sql
   ├── q4.sql
   ├── q5.sql
   ├── q6.sql
   ├── q7.sql
   ├── q8.sql
   ├── q9.sql
   ├── query_runner_p1.sh
   └── query_runner_p2.sh
── run_tests.sh
── sccs32s
   ├── Makefile
   ├── README.md
   ├── read_table.h
   ├── sccs32s.cpp
   └── sccs_compare.cpp
── start_postgres.sh

3 directories, 40 files
-[naman☮0xb1ad3]-(~/git/blockchain-stats/src)-[git://master ✗]-
-> ▯
[setup] 0:mkcsv  1:query_run_1- 2:query_runner_2  3:start_postg
```

Fig. 1. Directory Structure

- The src/start_postgres.sh script This script starts up the postgres_naman container. Before starting, though, it checks if docker is available, if the user running has permission to run docker and if such an image is available. In the last case, it builds the image from the provided Dockerfile and then runs it. In all the other cases, it fails.

- The src/run_tests.sh script: This script assumes the container instance of postgresql_naman image is up and running and ready to accept connections. It then runs all the queries in order, both part one and two, along with the formation of the addr_sccs.dat file in between. All the outputs with their queries are printed on the stdout.
- The src/mk_addr_sccs.sh script: This helper script does as its name suggests, it requires that the dataset/out.csv file is present which is generated by the queries/input_creator.sql query.

## III. Instructions for Running

### A. *Dockerized workflow*

Start the docker container

$ ./start_postgresql.sh

Run all the tests

$ ./run_tests.sh

### B. *Native Workflow:*

Assuming the postgresql is started, with all the tables created and default user as postgres, run the part 1

$ ./queries/query_runner_p1.sh queries

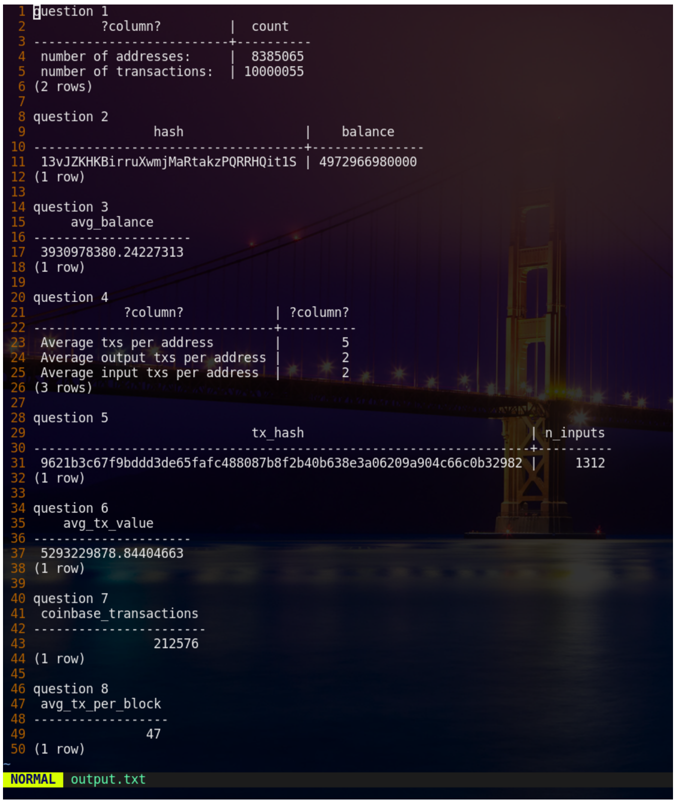
Make the addr_sccs.dat file

$./mk_addr_sccs.sh

Run part2

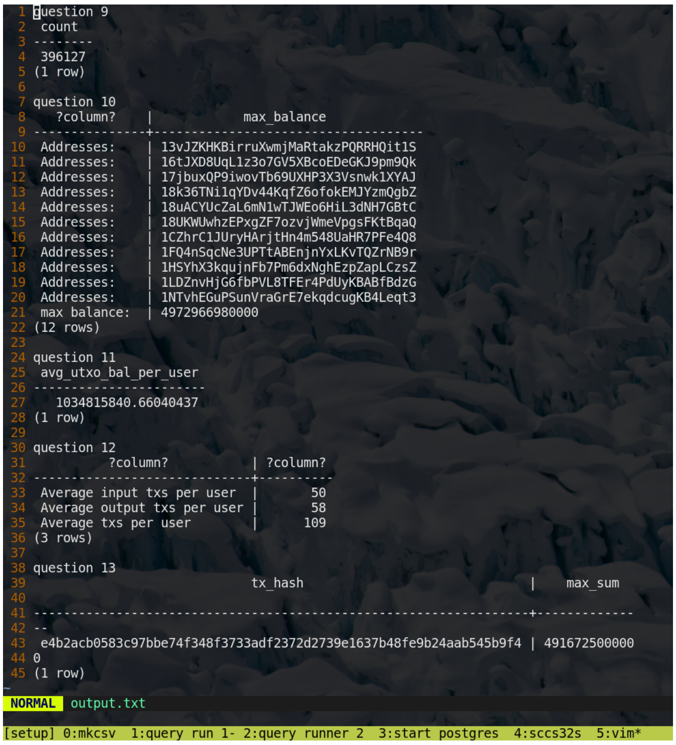
$ ./queries/query_runner_p2.sh queries

## IV. Output Screenshorts



```
question 1
         ?column?         |  count
--------------------------+----------
 number of addresses:     |  8385065
 number of transactions:  | 10000055
(2 rows)

question 2
               hash                |    balance
-----------------------------------+---------------
 13vJZKHKBirruXwmjMaRtakzPQRRHQit1S | 4972966980000
(1 row)

question 3
     avg_balance
---------------------
 3930978380.24227313
(1 row)

question 4
            ?column?              | ?column?
----------------------------------+----------
 Average txs per address          |        5
 Average output txs per address   |        2
 Average input txs per address    |        2
(3 rows)

question 5
                             tx_hash                              | n_inputs
------------------------------------------------------------------+----------
 9621b3c67f9bddd3de65fafc488087b8f2b40b638e3a06209a904c66c0b32982 |     1312
(1 row)

question 6
    avg_tx_value
---------------------
 5293229878.84404663
(1 row)

question 7
 coinbase_transactions
-----------------------
                212576
(1 row)

question 8
 avg_tx_per_block
------------------
               47
(1 row)
~
NORMAL  output.txt
```

Fig. 2. Output for Part 1



```
question 9
 count
--------
 396127
(1 row)

question 10
   ?column?    |             max_balance
---------------+------------------------------------
 Addresses:    | 13vJZKHKBirruXwmjMaRtakzPQRRHQit1S
 Addresses:    | 16tJXD8UqL1z3o7GV5XBcoEDeGKJ9pm9Qk
 Addresses:    | 17jbuxQP9iwovTb69UXHP3X3Vsnwk1XYAJ
 Addresses:    | 18k36TNi1qYDv44KqfZ6ofokEMJYzmQgbZ
 Addresses:    | 18uACYUcZaL6mN1wTJWEo6HiL3dNH7GBtC
 Addresses:    | 18UKWUwhzEPxgZF7ozvjWmeVpgsFKtBqaQ
 Addresses:    | 1CZhrC1JUryHArjtHn4m548UaHR7PFe4Q8
 Addresses:    | 1FQ4nSqcNe3UPTtABEnjnYxLKvTQZrNB9r
 Addresses:    | 1HSYhX3kqujnFb7Pm6dxNghEzpZapLCzsZ
 Addresses:    | 1LDZnvHjG6fbPVL8TFEr4PdUyKBABfBdzG
 Addresses:    | 1NTvhEGuPSunVraGrE7ekqdcugKB4Leqt3
 max balance:  | 4972966980000
(12 rows)

question 11
 avg_utxo_bal_per_user
-----------------------
   1034815840.66040437
(1 row)

question 12
          ?column?           | ?column?
-----------------------------+----------
 Average input txs per user  |       50
 Average output txs per user |       58
 Average txs per user        |      109
(3 rows)

question 13
                             tx_hash                              |    max_sum

------------------------------------------------------------------+-------------
--
 e4b2acb0583c97bbe74f348f3733adf2372d2739e1637b48fe9b24aab545b9f4 | 491672500000
0
(1 row)
~
NORMAL  output.txt
[setup] 0:mkcsv  1:query run 1- 2:query runner 2  3:start postgres  4:sccs32s  5:vim*
```

Fig. 3. Output for Part 2

## V. Conclusion

This concludes the homework assignment two. Even though considerable effort was made to write clever programs in C and parse the given database files, each effort tried to inevitably make a database implementation in some way or another. Hence, instead of reinventing the wheel, an established technology was used and effort was redirected to its deployment and execution. All the queries run a bit slower in the docker environment, taking up about 5-6 minutes in a hex core i7-8750H machine. Native workflow can easily make up for the time delay. All the code for the project is committed over at GitLab repository available at:

https://gitlab.com/r0ck3r008/blockchain-stats