**Business Data Management Lab Exercise**

**Index:**

**In the zip file, you can see that the lab.py is going to generate the mysql database and queries, the bdm.py and the bdm.html is for the html displaced.**

**A description of the system you used. This is how your report should start. The version of the operating system you are using, MYSQL version, APACHE version, version of Python or whatever other software you used.**

**Mac: Mac OS Sierra 10.12.4**

**MySQL: 5.7.17**

**Apache: 2.4.25**

**Python: 2.7.10**

**a brief explanation of you set up the system (how did you install all the servers, how you did the configuration, etc). Keep this brief.**

**In this part, first I install Python 2.7 and install MySQL and MySQL workbrench.**

**Then for the Apache configuration, I use the default folder/cgi-bin/ and default html path both in /Library/WebServer/ also uncomment some module and put some command in httpd.conf and I also do install MySQL-python connector.**

**Part 1.**

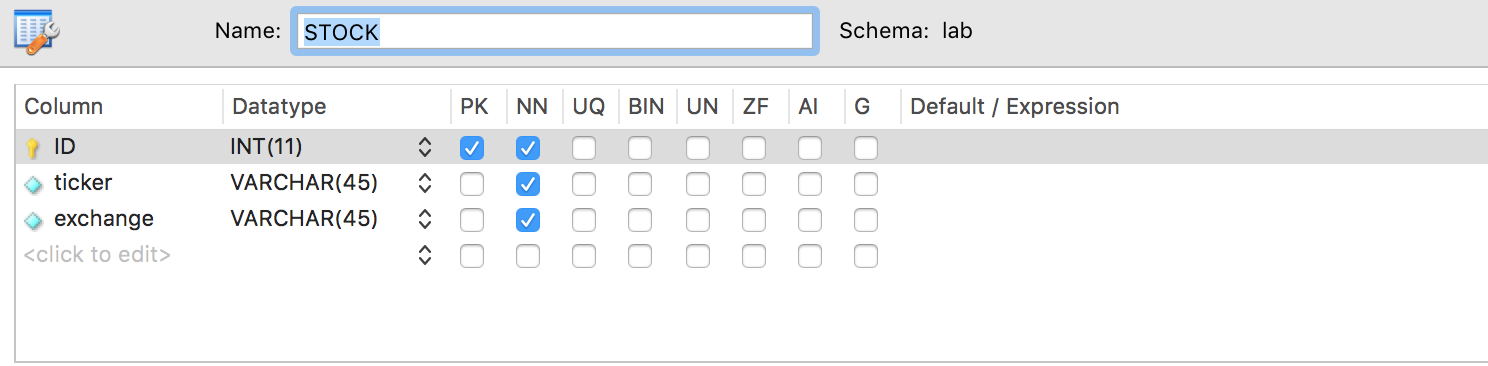
Devise relational algebra queries for the following schema of a database storing information about daily stock prices and basic transactions made by a trading firm. You should define domains so as to make the schema complete.

**Define Domain as follows, for each table, I all generate a ID as Primary Key**

STOCK (ticker, exchange)

ticker: the stock’s ticker symbol; e.g. GOOG, AAPL, GE

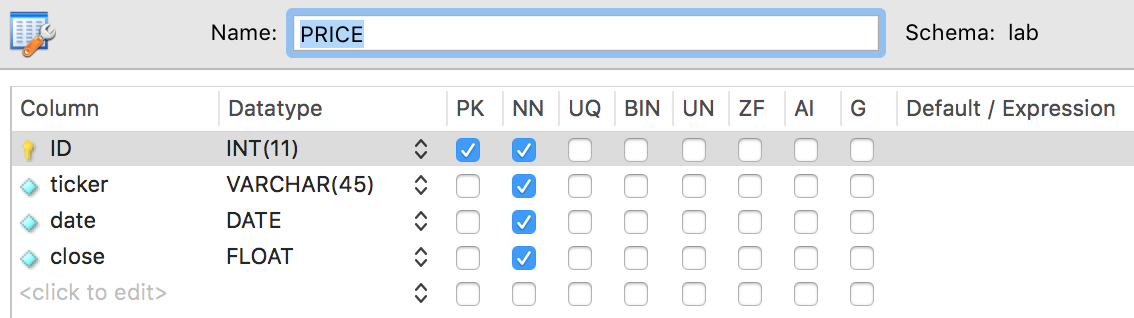
exchange: the exchange where the ticker is listed; e.g. NYSE, NASDAQ



PRICE (ticker, date, close)

ticker: the stock’s ticker symbol

date: the date of the price information close: the closing price of the stock



BUYnSELL (buy\_or\_sell,ticker, date, timestamp, value, num\_of\_shares)

buy or sell: ‘BUY’ or ‘SELL’

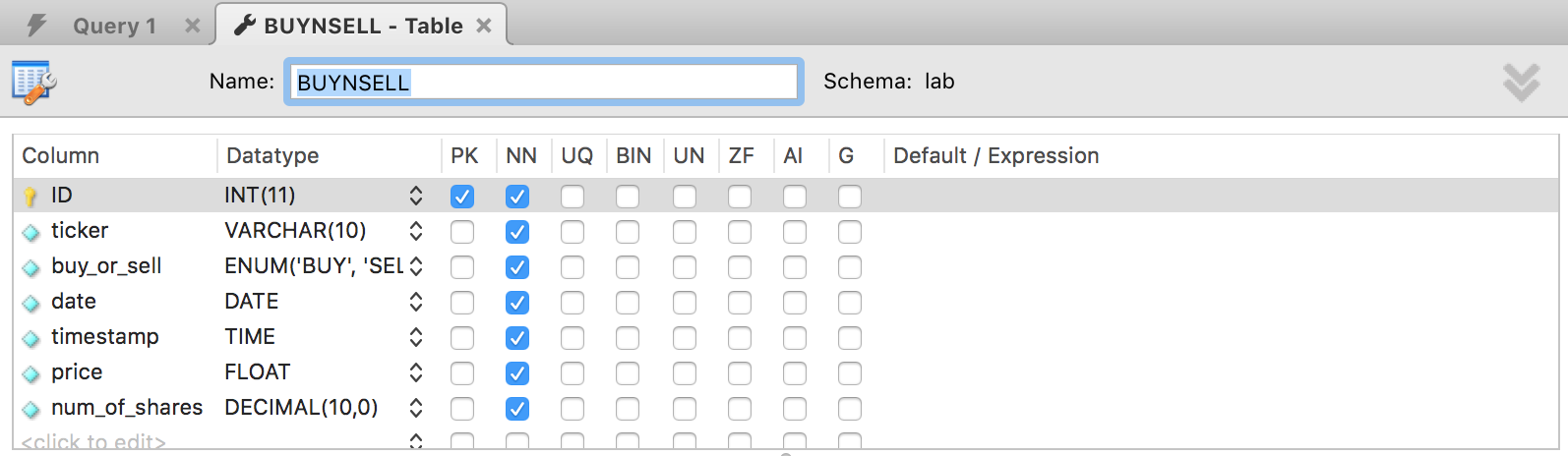
ticker: the stock’s ticker symbol

date: the date of the price information

timestamp: time of the transaction

value: the price of a single share

num\_of\_shares: number of shares (bought or sold)



Express the following as relational algebra expressions.

1. Find the tickers and all closing prices of all stocks exchanged in 2017.
2. Find all tickers (i.e. for all dates) whose closing price is both higher than ‘IBM’ on

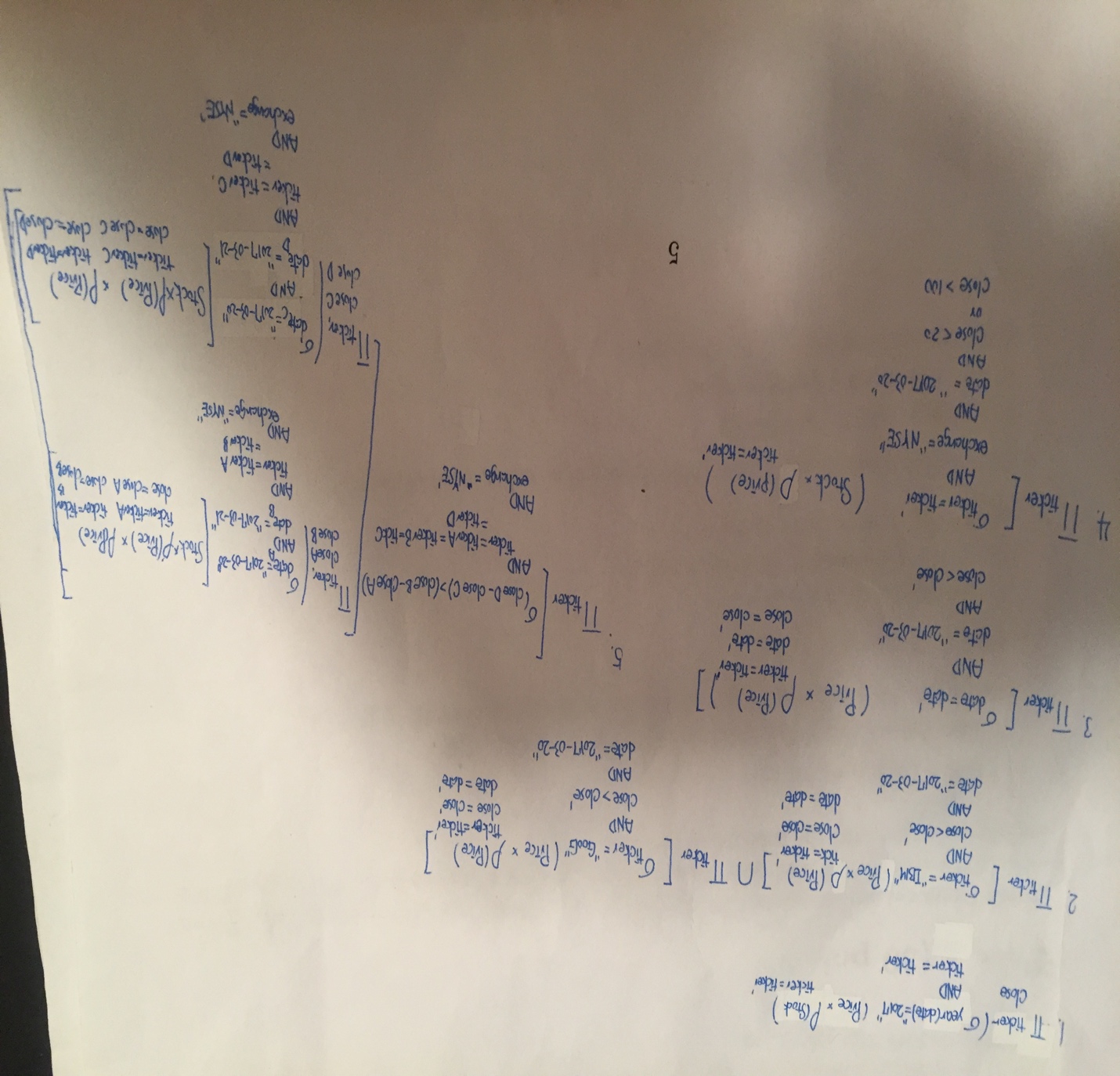
‘3/20/2017’ and no higher than ‘GOOG’ on ‘3/20/2017’.

1. Find the tickers of all stocks that closed at the highest price on ‘3/20/2017’.

(we are asking for “all stocks” since there may be more than one with the same “highest price”)

1. Find the tickers of all stocks in ‘NYSE’ whose closing price on ‘3/20/2017’ was either strictly below $20 or strictly above $100
2. Find all tickers in ‘NYSE’ of the stocks whose closing price showed the highest increase between ‘3/20/2017’ and ‘3/21/2017’ in ‘NYSE’ and whose closing price was (in ‘NYSE’) strictly above $100 for the entire 2017

(we are asking for “all stocks” since there may be more than one with the same increase. Recall that Relational Algebra does NOT support MAX, MIN, AVG operations.)



**Part 2.**

(40 points) Realize the database schema of Part 1 in MYSQL. Specify appropriate domains, keys, ICs, and all types of constraints. Then, implement as SQL queries all the relational algebra queries you devised in Part 1 (note that SQL queries can be “different/simpler” than the corresponding Relational Algebra ones).

In addition, to the above queries, also do in SQL the following one:

vi. Find the dates where the total price (i.e. price times num of shares) of ‘AAPL’ the firm (i.e. the trading firm which is using this database) sold was higher than what the firm bought in ‘NASDAQ’.

(1) What is the main reason for not asking to do the above query in Relational Algebra? Justify your answer. Use a detailed explanation and back it up with examples.

(2) Report the code that creates the database. For all the queries you wrote you should report the results of the queries on the example DB instance listed at the end of this lab exercise, but this example DB instance should be modified as follows. Whenever the answer to the query is empty (return nothing) you should add your own typical instances to the provide DB instance.

The main reason we can’t do the vi. Relational algebra may comes from that it’s hard to present the sum 1 > sum 2 in the relational algebra.

**For creating databases and mysql queries, you can run lab.py to generate the database and the query:**

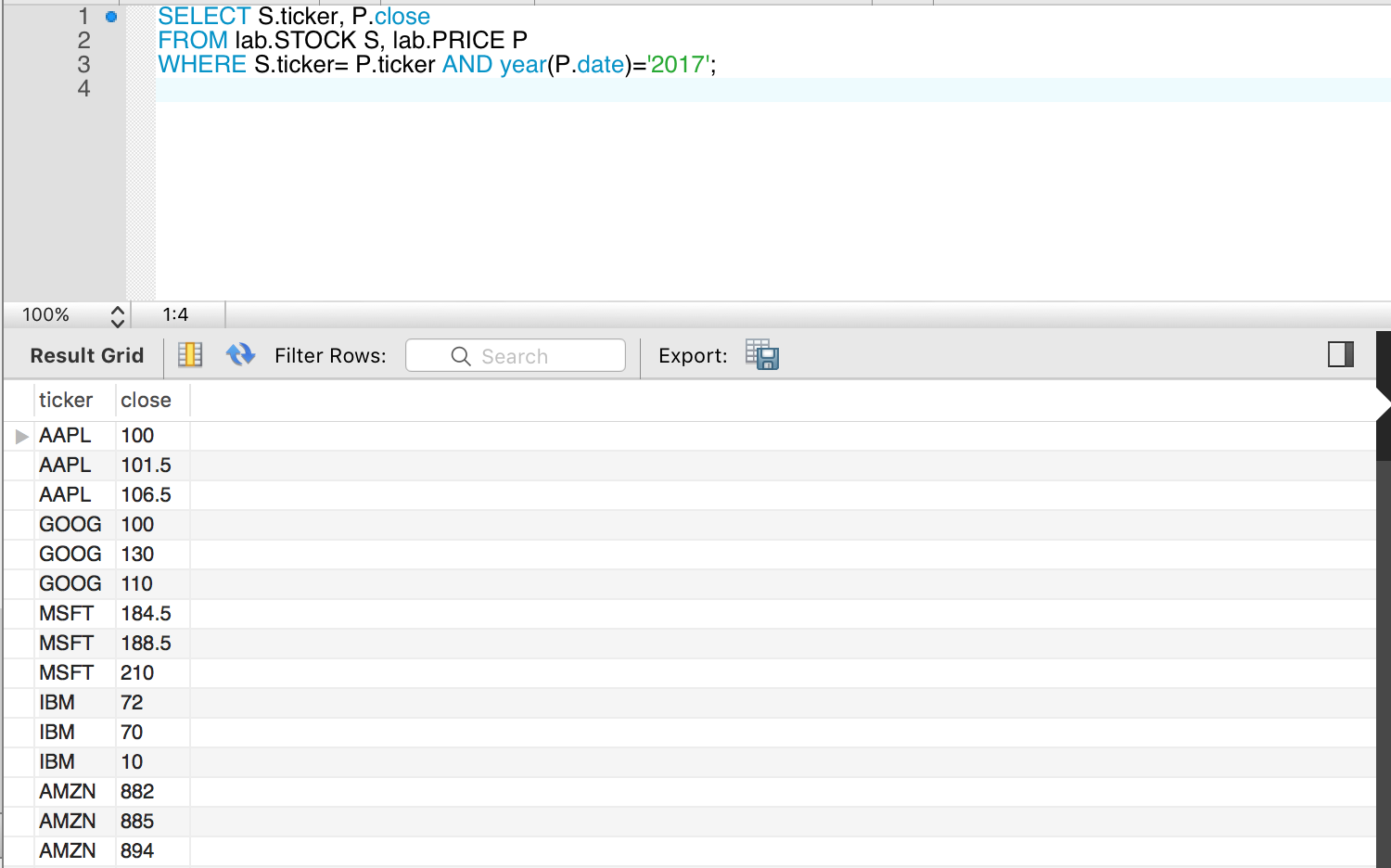
For the mysql query

1.

SELECT S.ticker, P.close

FROM lab.STOCK S, lab.PRICE P

WHERE S.ticker= P.ticker AND year(P.date)='2017';



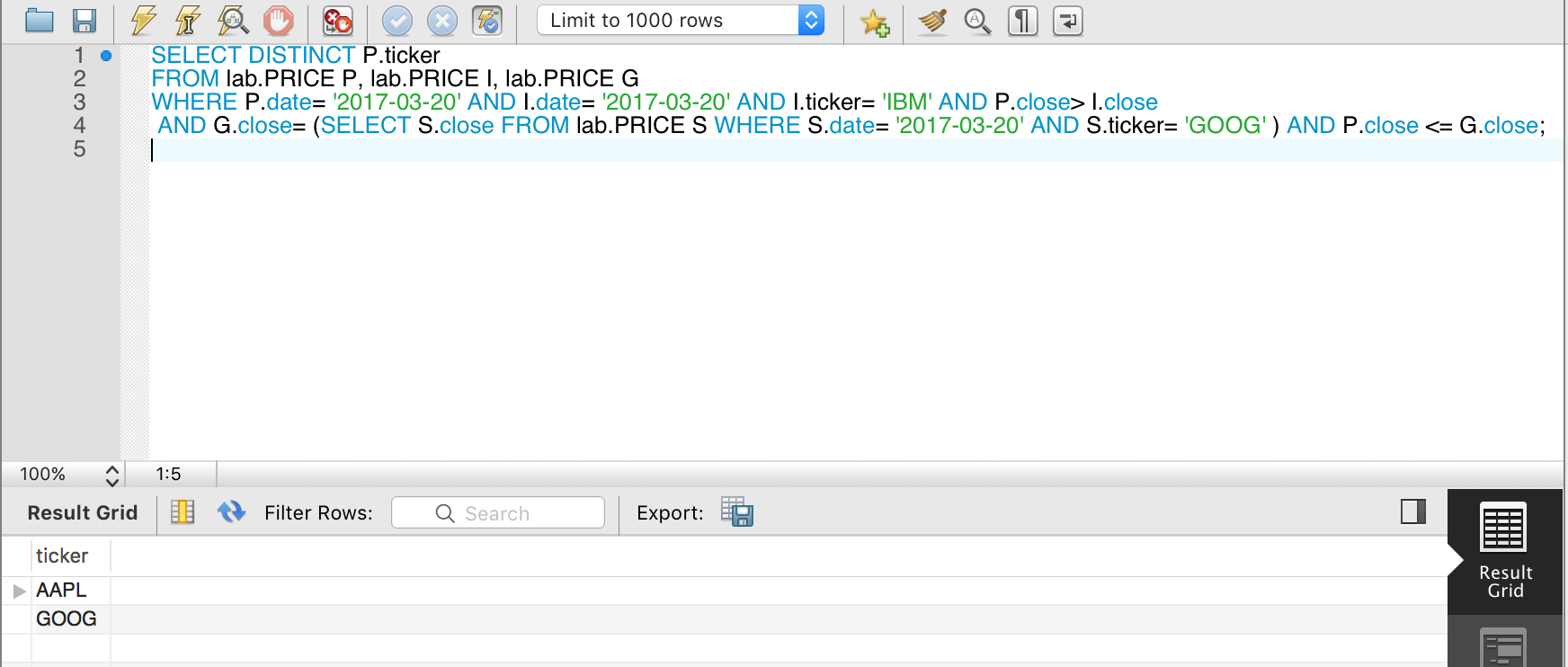
2.

SELECT DISTINCT P.ticker

FROM lab.PRICE P, lab.PRICE I, lab.PRICE G

WHERE P.date= '2017-03-20' AND I.date= '2017-03-20' AND I.ticker= 'IBM' AND P.close> I.close

AND G.close= (SELECT S.close FROM lab.PRICE S WHERE S.date= '2017-03-20' AND S.ticker= 'GOOG' ) AND P.close <= G.close;

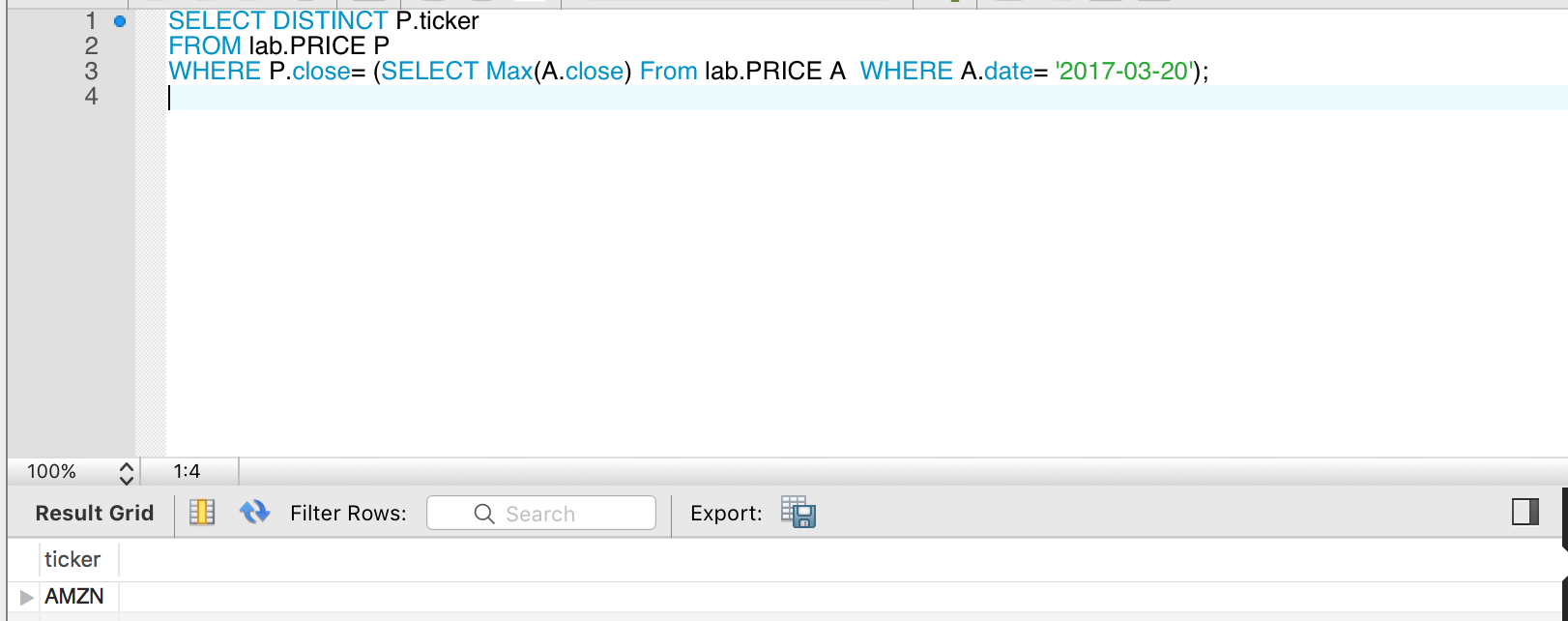


3.

SELECT DISTINCT P.ticker

FROM lab.PRICE P

WHERE P.close= (SELECT Max(A.close) From lab.PRICE A WHERE A.date= '2017-03-20');

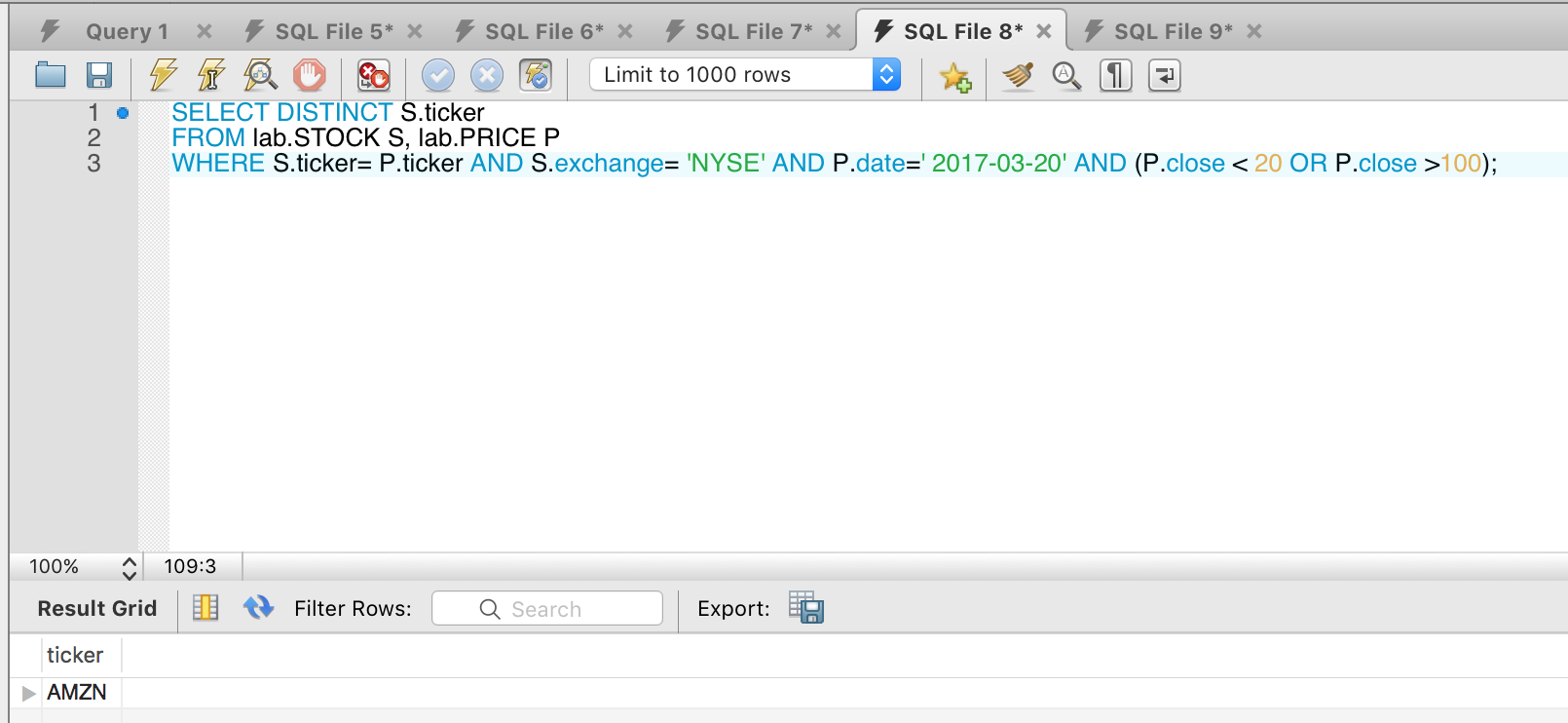


4.

SELECT DISTINCT S.ticker

FROM lab.STOCK S, lab.PRICE P

WHERE S.ticker= P.ticker AND S.exchange= 'NYSE' AND P.date=' 2017-03-20' AND (P.close < 20 OR P.close >100);



5.

SELECT DISTINCT P.ticker

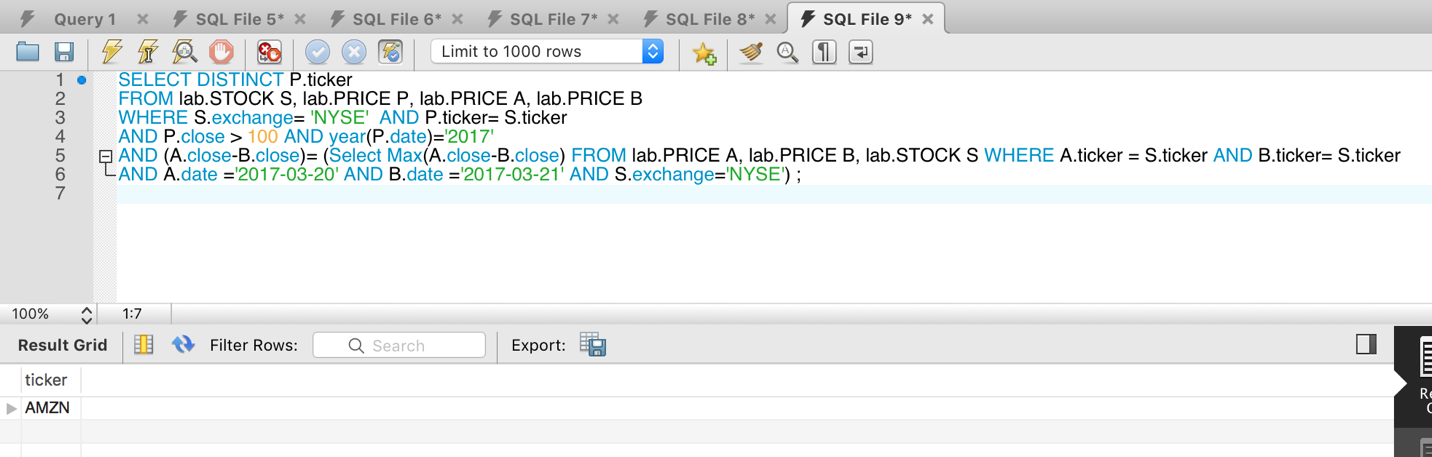
FROM lab.STOCK S, lab.PRICE P, lab.PRICE A, lab.PRICE B

WHERE S.exchange= 'NYSE' AND P.ticker= S.ticker

AND P.close > 100 AND year(P.date)='2017'

AND (A.close-B.close)= (Select Max(A.close-B.close) FROM lab.PRICE A, lab.PRICE B, lab.STOCK S WHERE A.ticker = S.ticker AND B.ticker= S.ticker

AND A.date ='2017-03-20' AND B.date ='2017-03-21' AND S.exchange='NYSE') ;



6.

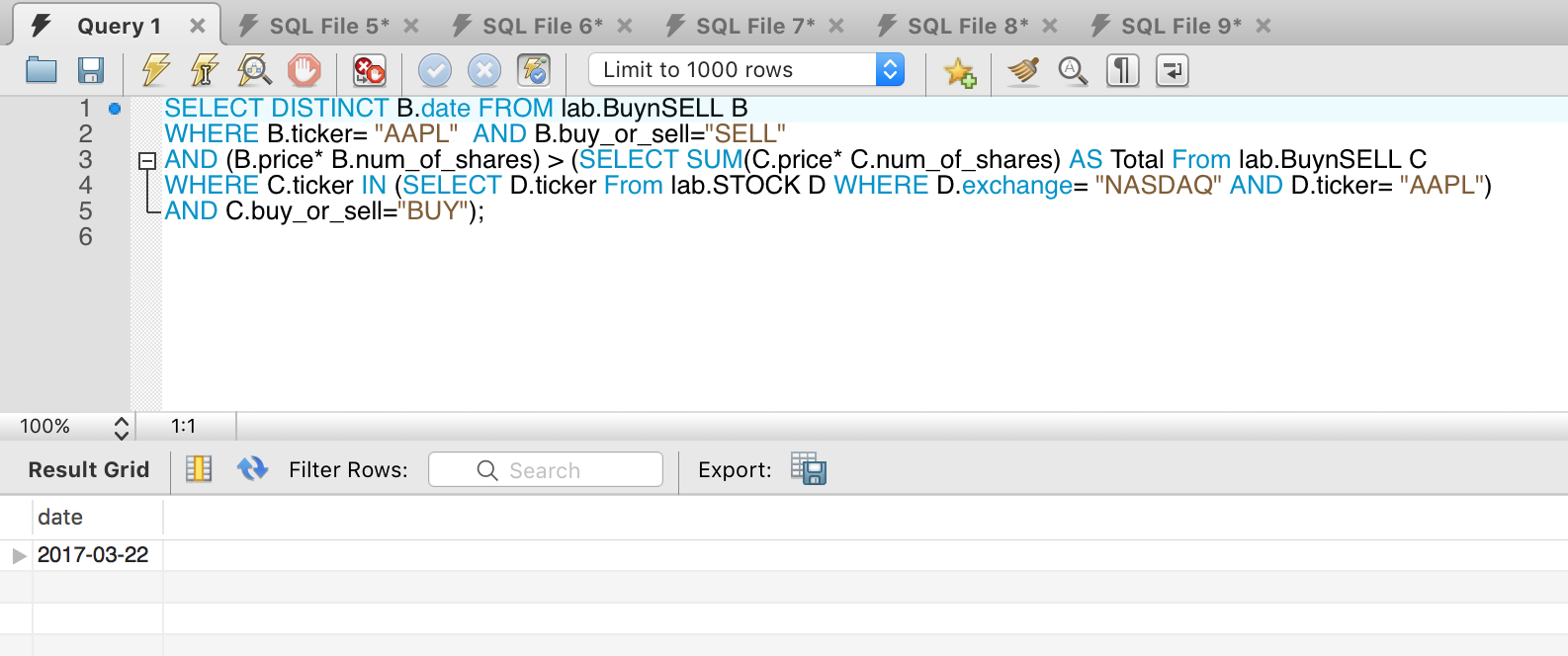
SELECT DISTINCT B.date FROM lab.BuynSELL B

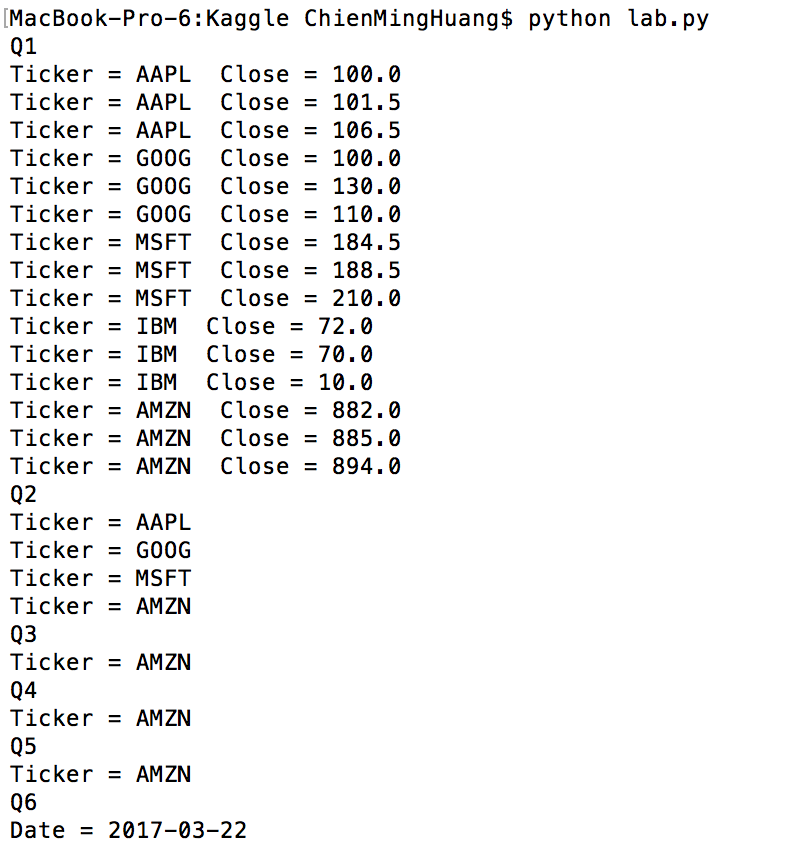
WHERE B.ticker= "AAPL" AND B.buy\_or\_sell="SELL"

AND (B.price\* B.num\_of\_shares) > (SELECT SUM(C.price\* C.num\_of\_shares) AS Total From lab.BuynSELL C

WHERE C.ticker IN (SELECT D.ticker From lab.STOCK D WHERE D.exchange= "NASDAQ" AND D.ticker= "AAPL")

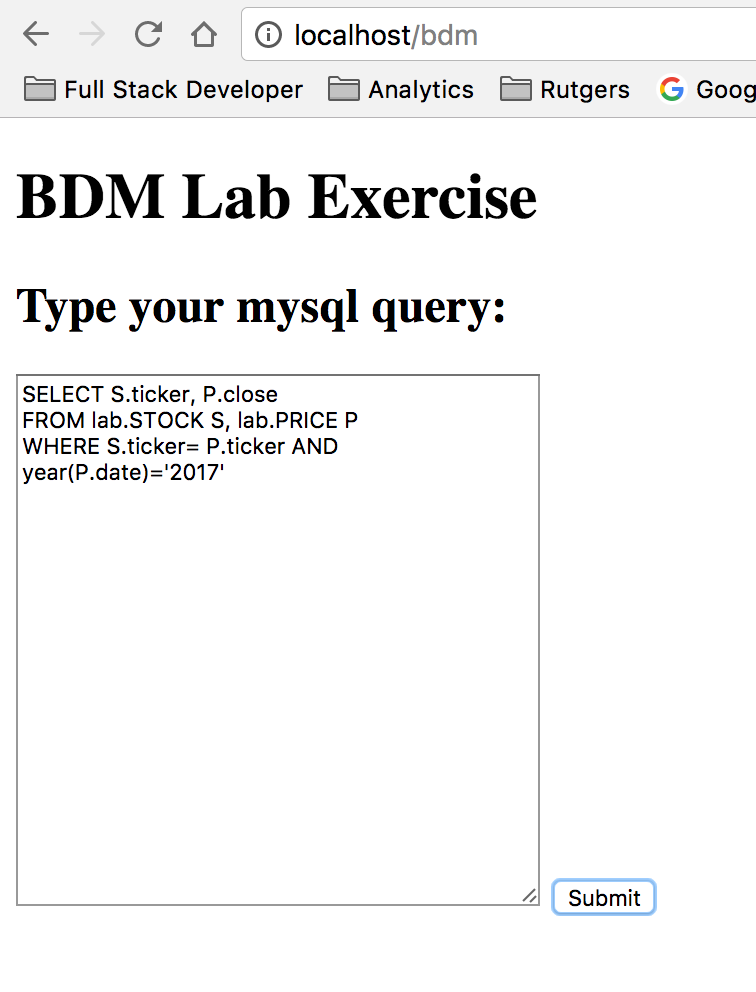
AND C.buy\_or\_sell="BUY")

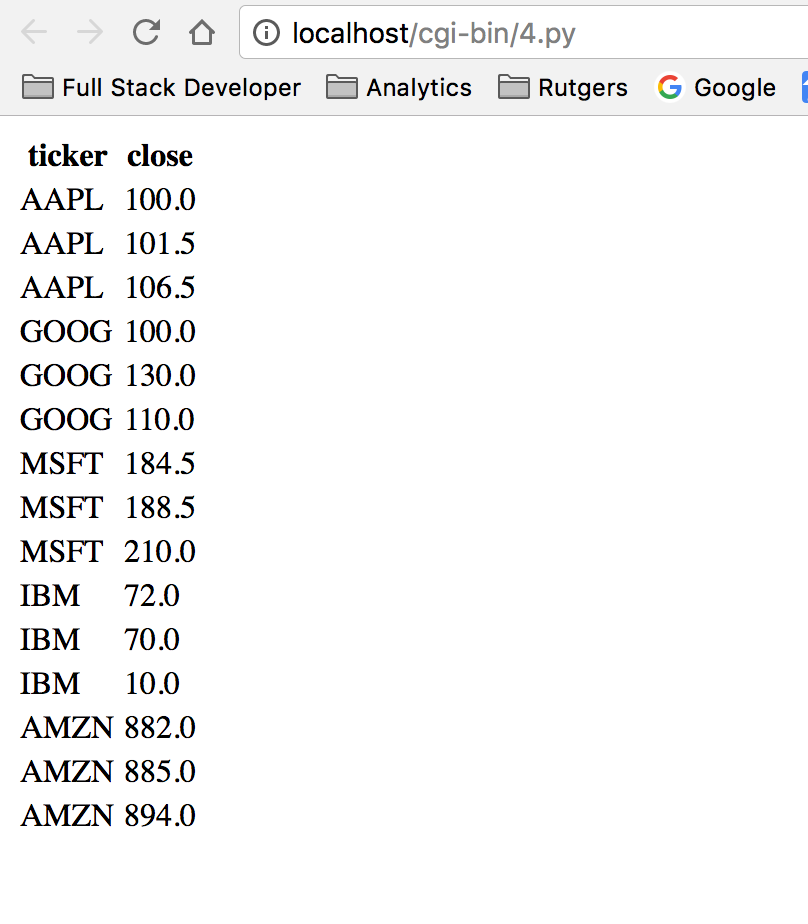




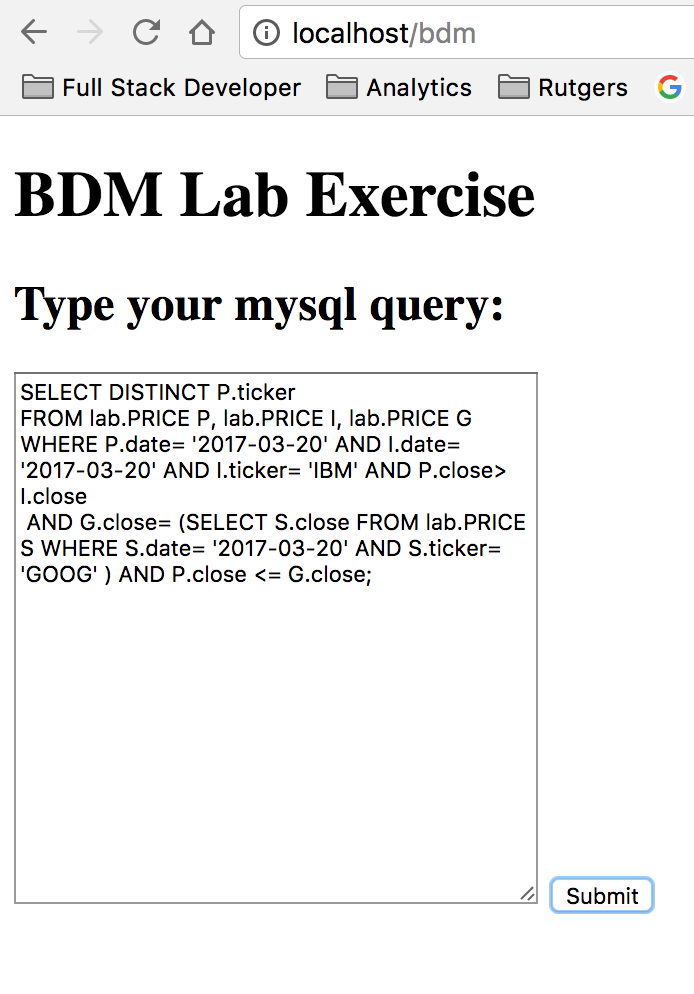
ii. (20 points) Make an HTML with a single textbox and a “submit button” where the user can enter a MYSQL query on the database you made in (i). Then, a CGI in Python (or in any other programming language you prefer) will submit this query to a MYSQL database and it will return an HTML file showing a simple table with the result of the query. The first row of the table should list the names of the attributes in the database and the remaining rows should list the tuples the query returned. Report the results of the HTML outputs on the example DB instance listed at the end of this document.

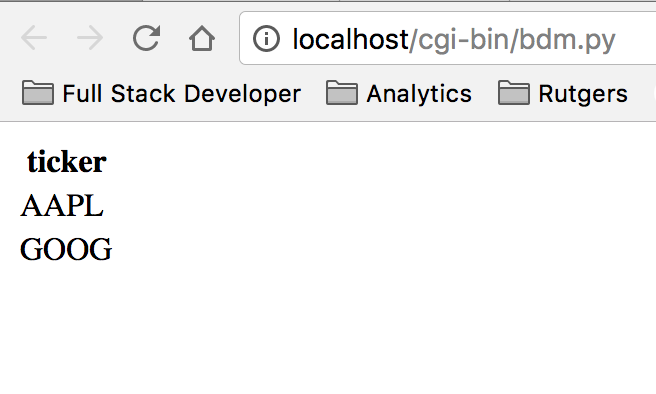
1.



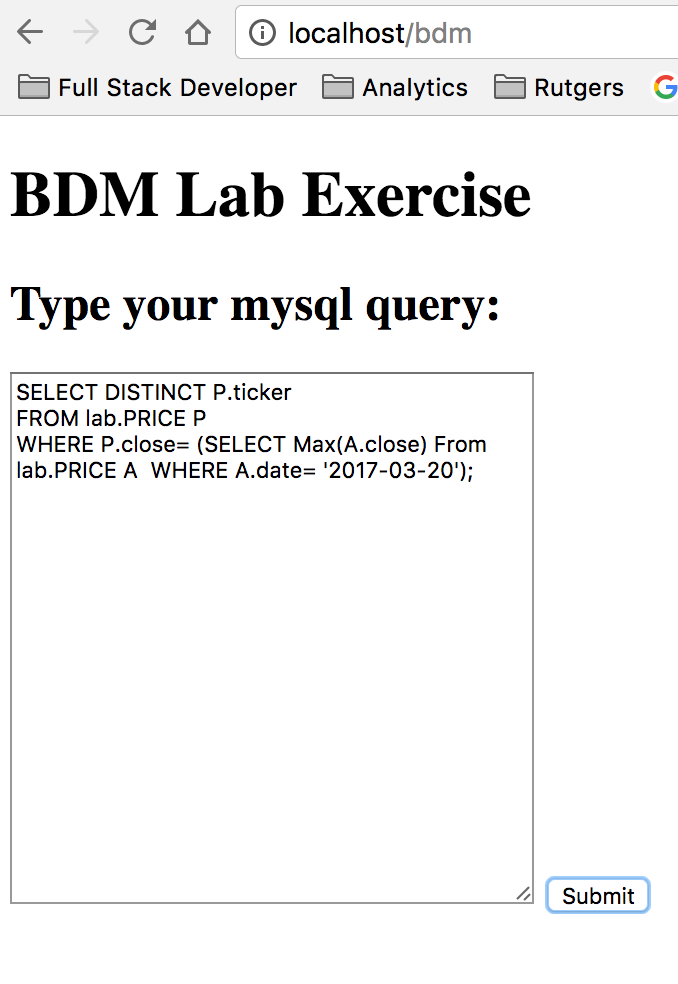


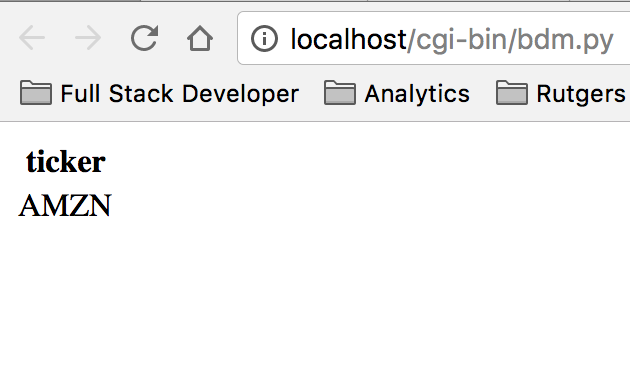
2.



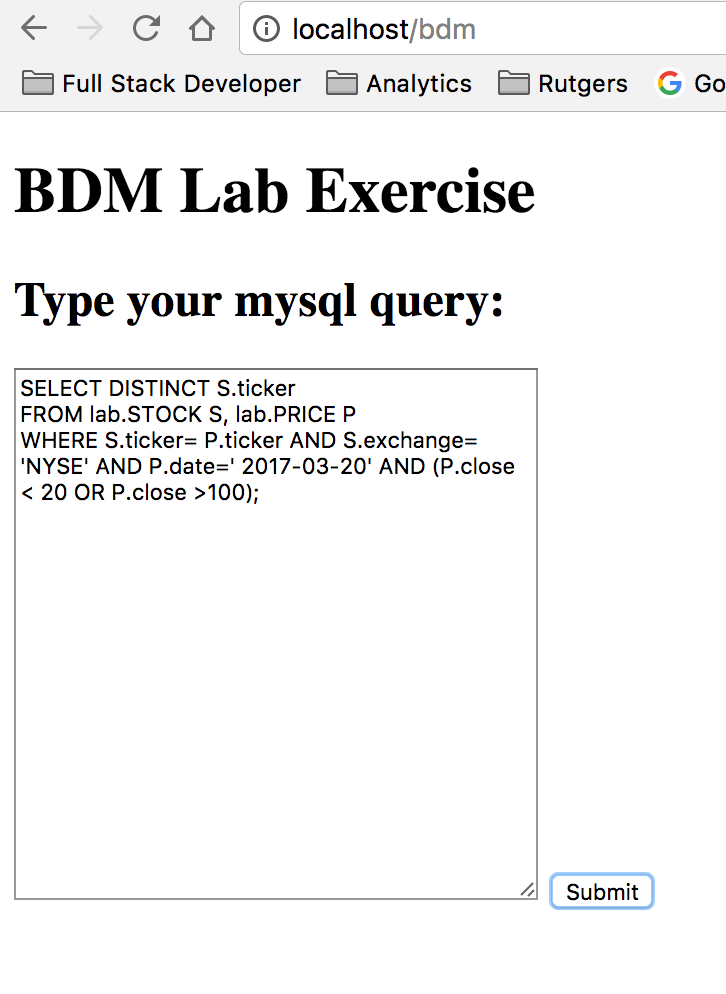


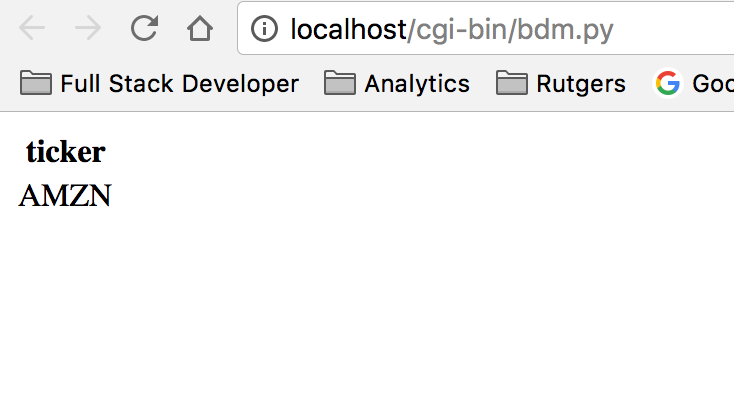
3.



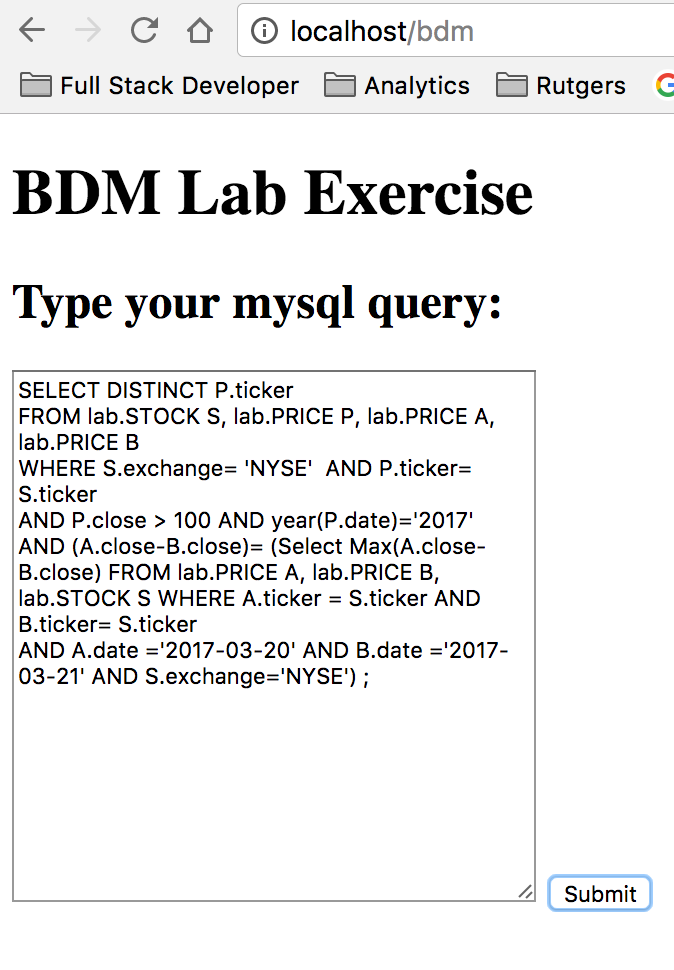


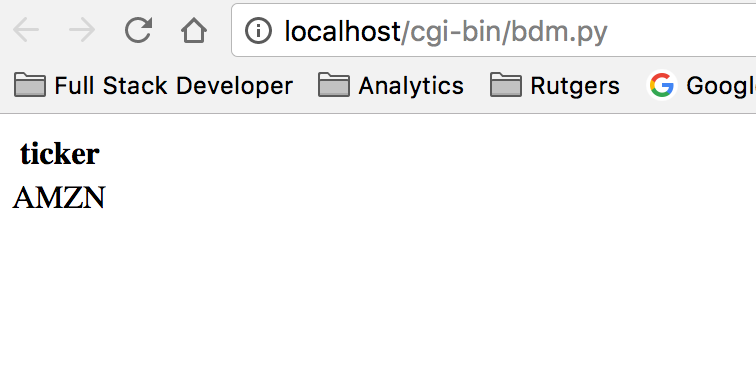
4.



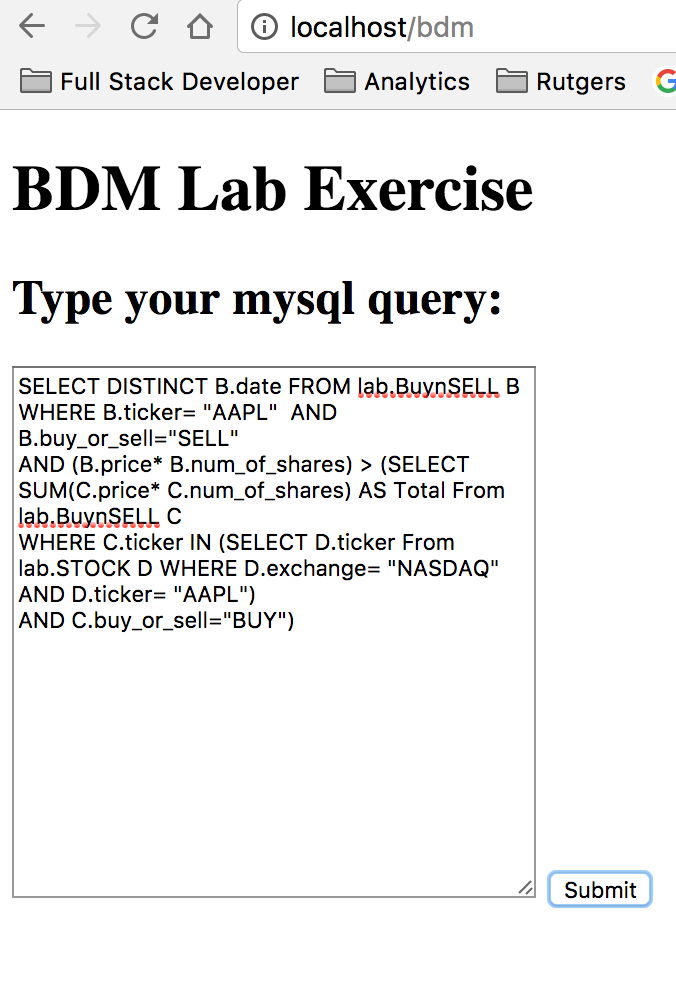


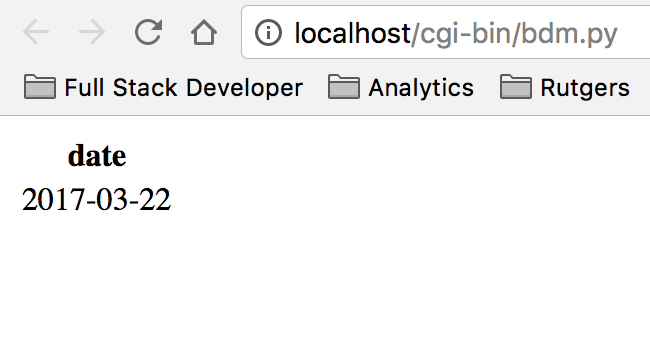
5.



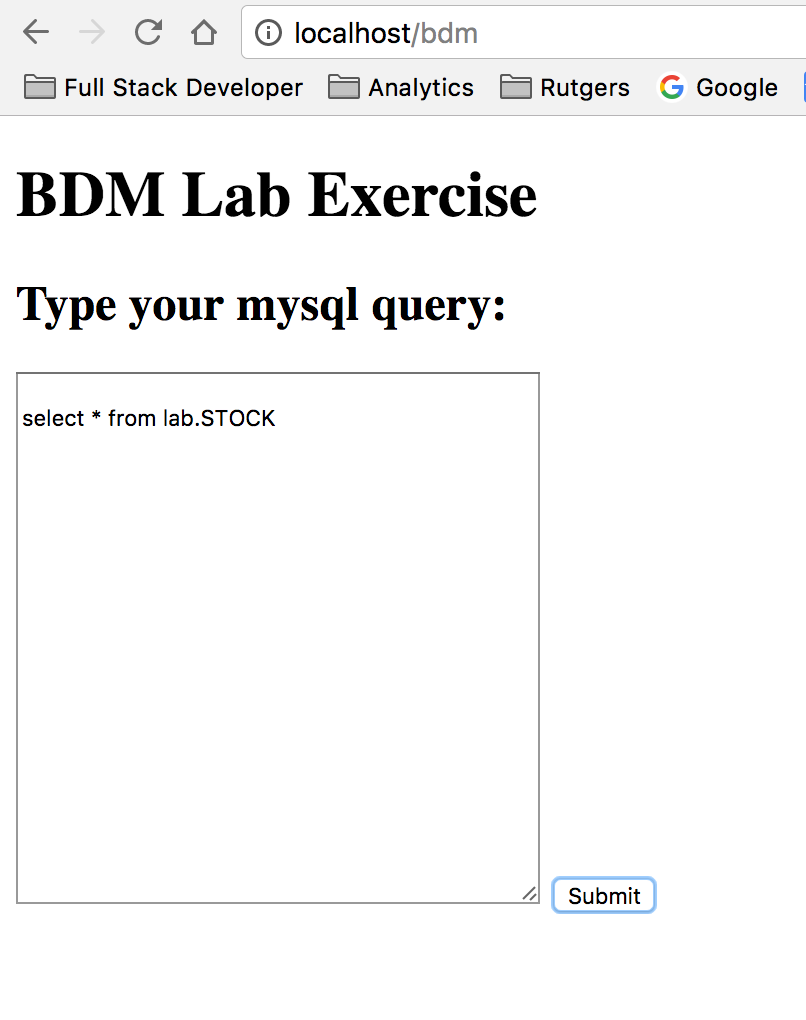


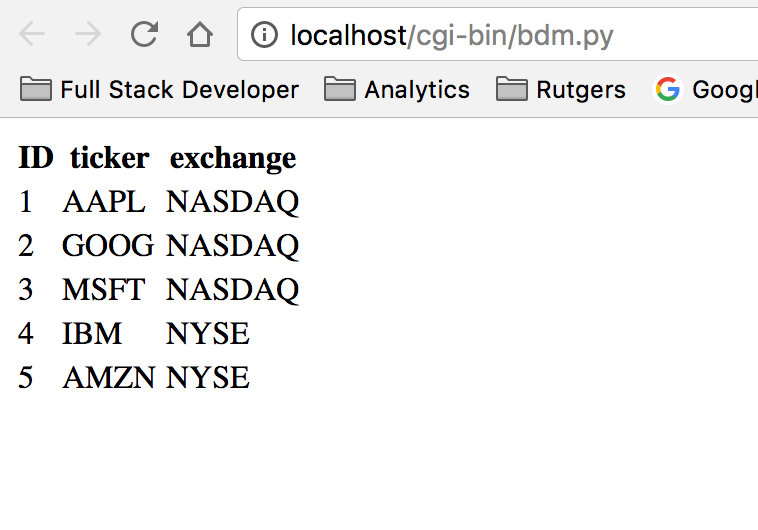
6.

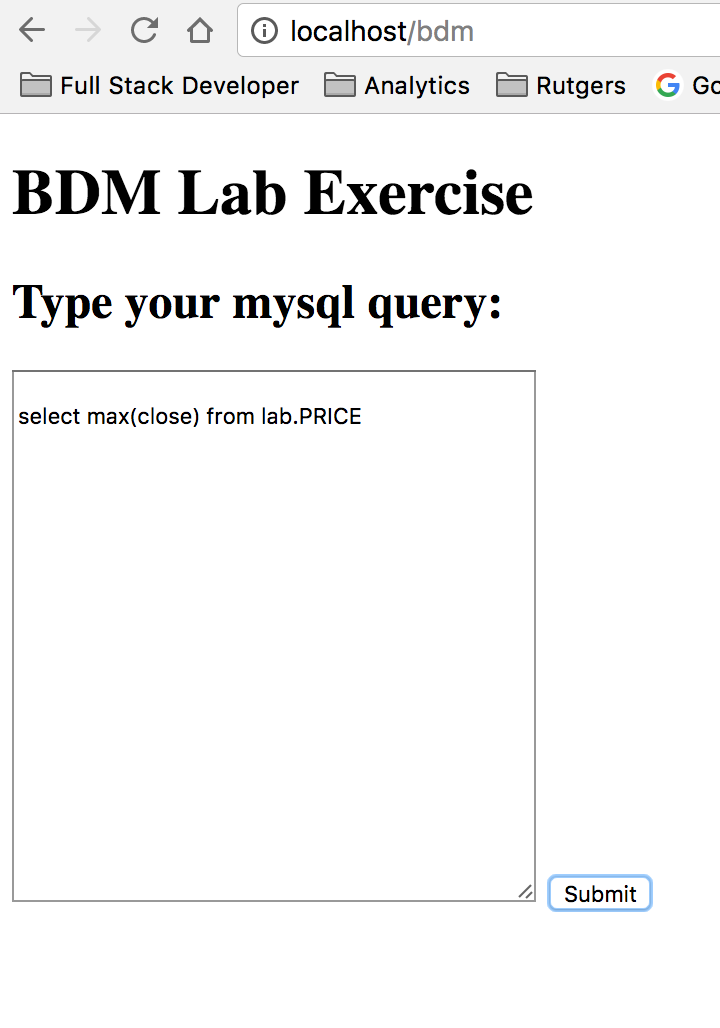


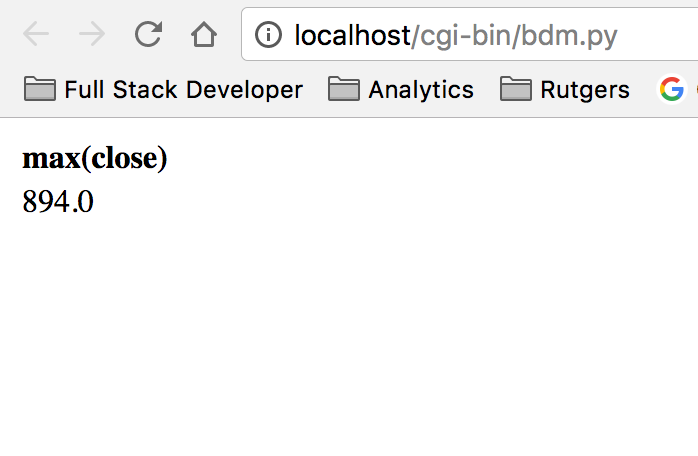


other indicative queries









**Part 3**

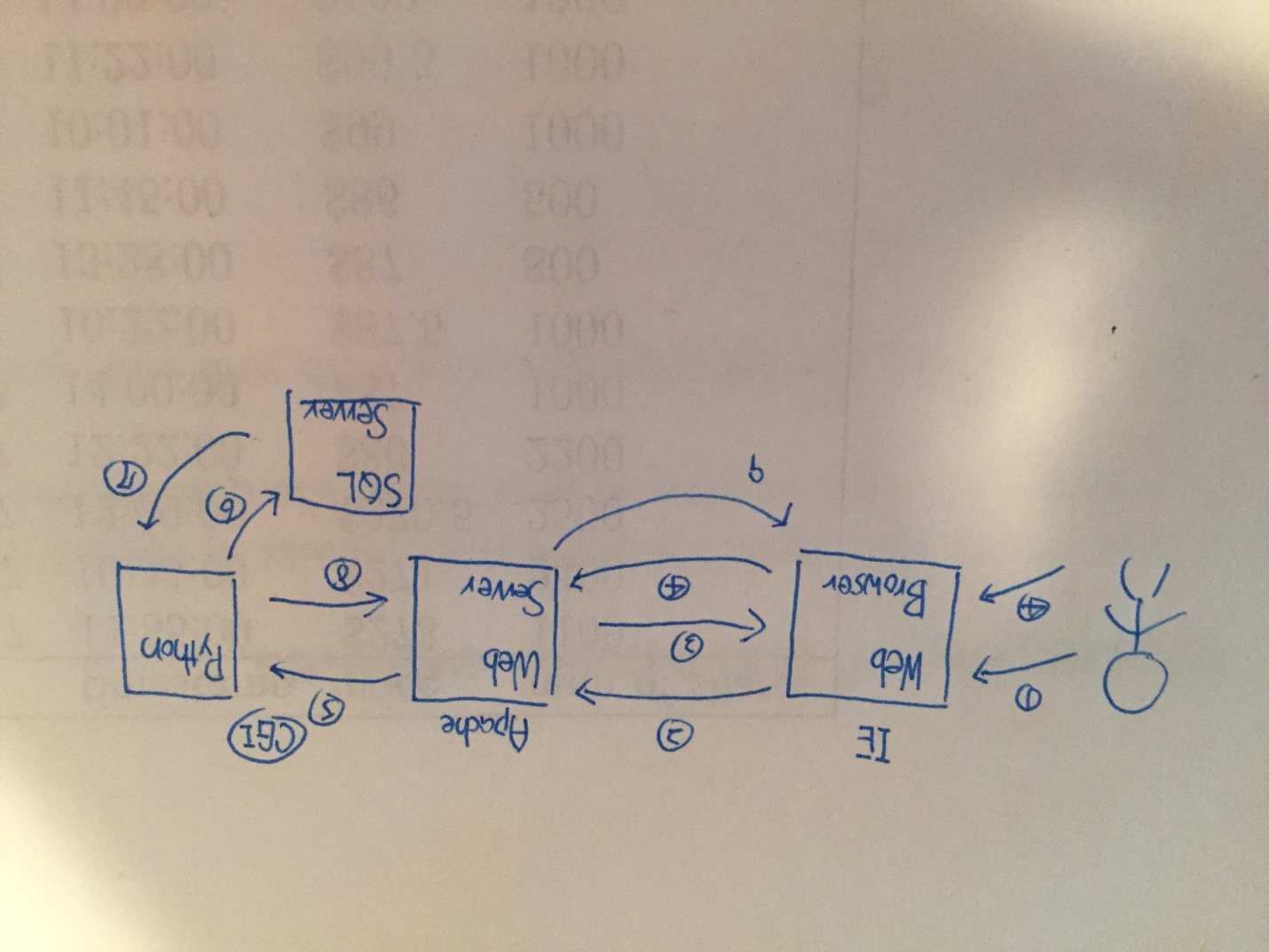
Research the literature about what is a Socket and how Socket programming is used. Explain in no more than 100 words how network programming and sockets come together. In another 100 words explain what is a “server”. Finally, write an essay of at most 300 words (in addition to the previous 200) and give a detailed diagram explaining which network services are related to your “Part 2” (above) That is, explain how many and what kind of servers (and what do they do) are involved in the system, how your program gets involved in the process, and what kinds of messages the system is exchanging, and what do we mean by “message”.

Ans:

A network socket is an internal endpoint for sending or receiving data at a single node in a computer network. A protocol stack, today usually provided by the operating system), is a set of programs that allows processes to communicate over a network using the protocols that the stack implements. The application programming interface (API) that programs use to communicate with the protocol stack, using network sockets, is called a socket API. Development of application programs that utilize this API is called socket programming or network programming.

A server is a computer program that provides services to other computer in the same or other computers. The computer that a server program runs in is also frequently referred to as a server. That machine may be a dedicated server or used for other purposes as well. In the client/server programming model, a server program awaits and fulfills requests from client programs, which may be running in the same or other computers. A given application in a computer may function as a client with requests for services from other programs and also as a server of requests from other programs.

For the Web Browser IE, it gets the html and html involves the CGI. The CGI can be a python program. For the SQL server, it has the database which can be individual. The Apache is always waiting for the request. The person browses the browser, so the web browsers get in touch with Apache. And the Apache will give a respond. Then here IE will send the message(file) to run the message to run the python program. The python program would connect mysql database to get data. Then mysql gives back the message through the python and Apache to the IE.



**Bonus points**

Extra marks for realizing a non-primitive HTML user interface.