

Business Data Management

Lab Exercise

Instructor: Periklis A. Papakonstantinou (periklis.research@gmail.com)

Teaching assistant: Nikolas Melissaris-Papanikolaou

Due date: Friday, November 16 at 11.59pm. We will set up Blackboard to receive your assignment reports (check the Blackboard announcements).

The exercise consists of three parts.

What to include in your submission

Some students will be asked by the Teaching Assistant to show how their projects run on their computer and explain how everything was installed, how everything was setup, and all the details for configuring your system, and of course a detailed/in-depth technical description of how and why your programs work in the way they do. If you have used AWS (Amazon Web Services) then you have to login and explain to the TA what have you done there.

In addition to the above (which is true for some students), every student should submit the following in one ZIP file.

1. A report that contains all answers to the relational algebra queries. If you think that it helps your case you should add a brief explanation why you wrote the query the way you did. If you add such a short explanation you will be able to get part marks in case of a mistake.
2. The same report should contain all SQL queries. Again you can add a brief explanation if you want.
3. In the same report, you should list all the NEW entries you made to the database provided in the end of the Lab Exercise.
4. In the same report, you should show the result of each of your queries on the Database instance you used (i.e. the given one + the entries you added).

5. In the same report, you should provide screenshots for your web application and the tests you performed.
6. You should provide a script that generates the database that you used in your tests. The script should be in a form that the TA can run it on his MYSQL system. If you think it is necessary you should include a file with instructions of how to do this.
7. You should provide the code of all files you used (HTML form, Python CGI script, or anything else) in this exercise.
8. **Make sure that you change nothing in your computer system until you receive the grade in this assignment.** The reason is that you may be asked to demonstrate everything in your computer (and things must look identical to your report). If there is any discrepancy then you will get zero in your lab exercise and unfortunately you will be automatically excluded from the project, where you will also get zero marks. You also loose all bonus points. In this case the grade in this class will be the rest of things (final exam and the quizzes).

Part 1 (30 points)

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.

- 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.

- i. Find the tickers and all closing prices of all stocks exchanged in 2017.
- ii. Find all tickers (i.e. for all dates) whose closing price is both higher than 'IBM' on '9/20/2018' and *no higher* than 'GOOG' on '9/20/2018'.
- iii. Find the tickers of all stocks that closed at the highest price on '9/20/2018'.
(we are asking for "all stocks" since there may be more than one with the same "highest price")
- iv. Find the tickers of all stocks in 'NYSE' whose closing price on '9/20/2018' was either strictly below \$20 or strictly above \$100
- v. Find all tickers in 'NYSE' of the stocks whose closing price showed the highest increase between '9/20/2018' and '9/21/2018' in 'NYSE' and whose closing price was (in 'NYSE') strictly above \$100 for the entire 2018
(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.)

Remarks: In the above queries if there are no relevant tuples in the database to compare then you do not include these in the resulting relation (i.e. what your query returns). For example, if a ticker does not appear on both '9/20/2018' and '9/21/2018' in 'NYSE' then you should not include it in the result. Furthermore, we are also not really careful about datatypes. For example, for the BUY/SELL attribute we must create a new datatype, but to keep things simple I will allow you to use a string and furthermore for making things even simpler you won't have to check for consistency. Finally, the above database does not have any Integrity Constraints specified. If you want you can specify primary, foreign keys, etc, however it is not necessary to do so neither here nor in SQL in the next question. Of course, in reality you must always specify ICs, but for simplicity in this lab exercise you can optionally do this if you want.

Part 2 (60 points)

- i. (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. Wherever you had to calculate a MAX or MIN value you are allowed in SQL to use the primitives MAX or MIN if you find it easier this way (but you should be careful because these queries might need additional more advanced SQL commands).

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

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.
- 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.

Remark: No special HTML formatting is necessary. The most basic (4–5 commands) HTML commands suffice to do this part. *You do not loose any marks* for a very primitive looking webpage design. For example, when the rows are returned it is okay to just print it in the form “TICKER=GOOG”, instead of presenting the results in a nicely printed HTML table. Basic HTML and CGI handling can be found either in the online resources we provided or in the 7th Chapter of our textbook.

Important remark: *It is not required* to use Python for the CGI. Use any programming language you feel comfortable with. If you choose Python then you can receive TA help in how Python integrates with MySQL and Apache.

Part 3 (10 points)

Research the literature and write an essay explaining what are the Sockets, what is Socket Programming, and how these notions are related to the work that a “server” has to do. Also, briefly explain what do people mean when they use the term “server”. Give some examples if you think that they will help explaining what is going on. This should be an essay of at most 200 words.

You should also write a second essay of at most 300 words explaining the following (in addition to the essay you should provide a detailed diagram): 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”.

Bonus points (5)

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

Example of database entries

In your report you should present your results in the following database instance. You must also add at least 5 new entries that make the queries you wrote meaningful.

(remark: when you are experimenting with your code you should change the database)

STOCK

ticker	exchange
AAPL	NASDAQ
GOOG	NASDAQ
MSFT	NASDAQ
IBM	NYSE

Remark: If you wish you can modify the above table by *adding* some new entries.

PRICE

ticker	date	close
AAPL	9/20/2018	\$100
AAPL	9/21/2018	\$101.5
AAPL	9/22/2018	\$106.5
GOOG	9/20/2018	\$100
GOOG	9/21/2018	\$130
GOOG	9/22/2018	\$110
MSFT	9/20/2018	\$184.5
MSFT	9/21/2018	\$188.5
MSFT	9/22/2018	\$210
IBM	9/20/2018	\$72
IBM	9/21/2018	\$70
IBM	9/22/2018	\$10

Remark: You must modify the above table by *adding* indicative entries such that the answers to the queries of Part 1 and 2 appear to be somewhat typical. Do not add more than 10 new entries (unless it is absolutely necessary for demonstrating your answers). Present all added entries in your report.

BUYnSELL

ticker	buy_or_sell	date	timestamp	price	num_of_shares
IBM	BUY	9/20/2018	11:55:00	\$273	1100
IBM	BUY	9/21/2018	10:45:00	\$271	2400
IBM	SELL	9/22/2018	12:09:00	\$270.5	2500
GOOG	BUY	9/20/2018	12:22:00	\$86	2200
GOOG	SELL	9/20/2018	14:00:00	\$87	1000
GOOG	SELL	9/21/2018	10:22:00	\$87.5	1000
GOOG	BUY	9/21/2018	13:28:00	\$87	800
GOOG	SELL	9/22/2018	11:45:00	\$86	500
AAPL	BUY	9/20/2018	10:01:00	\$99	1000
AAPL	BUY	9/20/2018	11:22:00	\$99.5	1000
AAPL	BUY	9/21/2018	14:22:00	\$100	1000
AAPL	SELL	9/22/2018	14:42:00	\$103	3000
MSFT	BUY	9/20/2018	11:45:00	\$186	1500
MSFT	SELL	9/21/2018	10:45:00	\$188	1000
MSFT	BUY	9/22/2018	12:03:00	\$187	5000

Remark: You must modify the above table by *adding* indicative entries such that the answers to the queries of Part 1 and 2 appear to be somewhat typical. Do not add more than 5 new entries. Present all added entries in your report.