Ademir Vila

05/22/2021

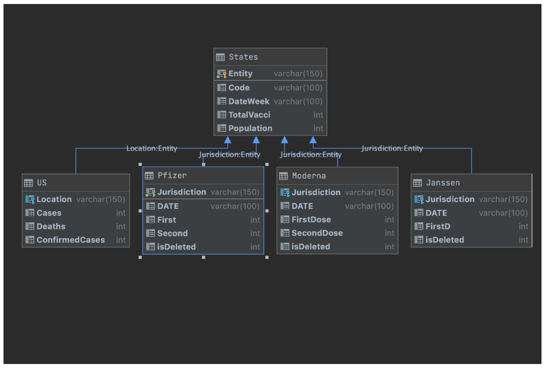
CPSC 408 Database Management

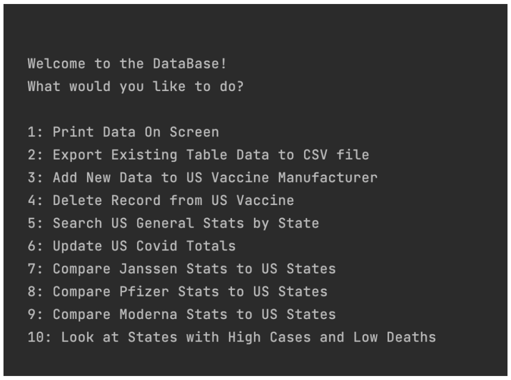
Final Project Write Up

For the final assignment for CPSC 408 Database Management, we were tasked to create our own database with a front-end application, similar to assignments four and five. In this case, I decided to tackle the issue of Covid-19 Vaccinations in terms of distribution, as well as comparing various statistics between states. I wanted to tackle this issue because thousands of vaccines each day are going to waste and most end up in the garbage, as vaccines such as Pfizer and Moderna are temperature controlled and places cannot keep them for more than 24 hours.

In terms of related works in the field, there are many Covid-19 trackers of all sorts, with the main one being the Center for Disease Control and Prevention (CDC). On their website, they are tracking all sorts of Covid data, from cases to vaccinations to statistics based on race. With that being said, a lot of their data is on a more broad, general basis. In addition, a few groups in the 408 Database class are also doing projects on Covid-19 stats. However, the focuses they have include the effects of mask use, as well as setting up appointments for vaccinations.

For my schema for this assignment, I have created 5 tables. Table ‘states’ is the primary table for the other four tables, as the primary key in this case in the name of the state, respectively defined as “State,” “Entity, ” and “Jurisdiction.” The schema is depicted below. As for my results, the main focus is on Vaccine Manufacturers, and how they compare to each other, as well as comparing it to General US statistics. With the startup of the python application, it will give you a set of options on what you can do in the database. Also, on startup, each csv file is read using pandas and then written to the table before the user enters a choice.

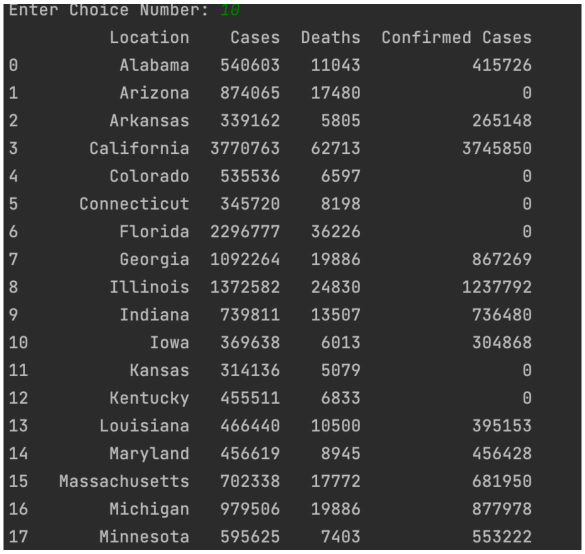




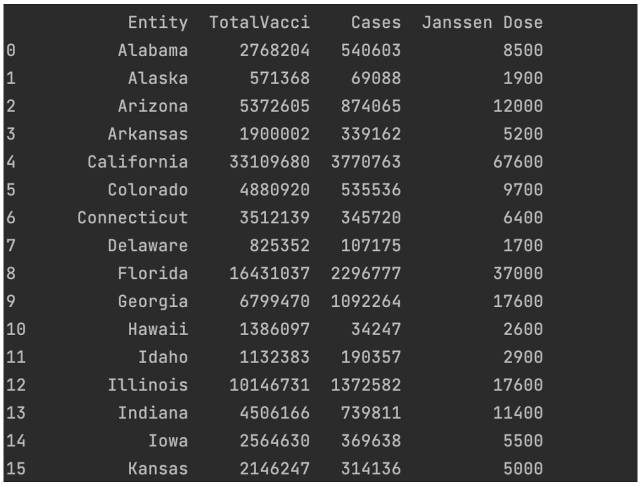
```
Welcome to the DataBase!
What would you like to do?

1: Print Data On Screen
2: Export Existing Table Data to CSV file
3: Add New Data to US Vaccine Manufacturer
4: Delete Record from US Vaccine
5: Search US General Stats by State
6: Update US Covid Totals
7: Compare Janssen Stats to US States
8: Compare Pfizer Stats to US States
9: Compare Moderna Stats to US States
10: Look at States with High Cases and Low Deaths
```

One of the main functions I wanted to focus more on is display of High cases and Low death rate, as well as comparing each manufacturer allocation of doses and comparing it to us general stats. For the display of High Cases and Low Death Rate, I used a subquery of select statements in order to generate the wanted query. The reason I wanted to use this query for my database is because it shows states that need to be focused on for pandemic prevention, as well as showing states may need more help in vaccine allocation. With that being said, this goes into my next main function: comparing each vaccine stat to US general stats. The reason why I separated these two is because I wanted to have the user first check each manufacturer stat(option 7-9) and then check which of these states have a high rate of cases and deaths(option 10). For these queries(option 7-9), I did a join across three tables: US, States and each respective manufacturer.



```
Enter Choice Number: 10
         Location    Cases  Deaths  Confirmed Cases
0         Alabama   540603   11043           415726
1         Arizona   874065   17480                0
2        Arkansas   339162    5805           265148
3      California  3770763   62713          3745850
4        Colorado   535536    6597                0
5     Connecticut   345720    8198                0
6         Florida  2296777   36226                0
7         Georgia  1092264   19886           867269
8        Illinois  1372582   24830          1237792
9         Indiana   739811   13507           736480
10           Iowa   369638    6013           304868
11         Kansas   314136    5079                0
12       Kentucky   455511    6833                0
13      Louisiana   466440   10500           395153
14       Maryland   456619    8945           456428
15  Massachusetts   702338   17772           681950
16       Michigan   979506   19886           877978
17      Minnesota   595625    7403           553222
```



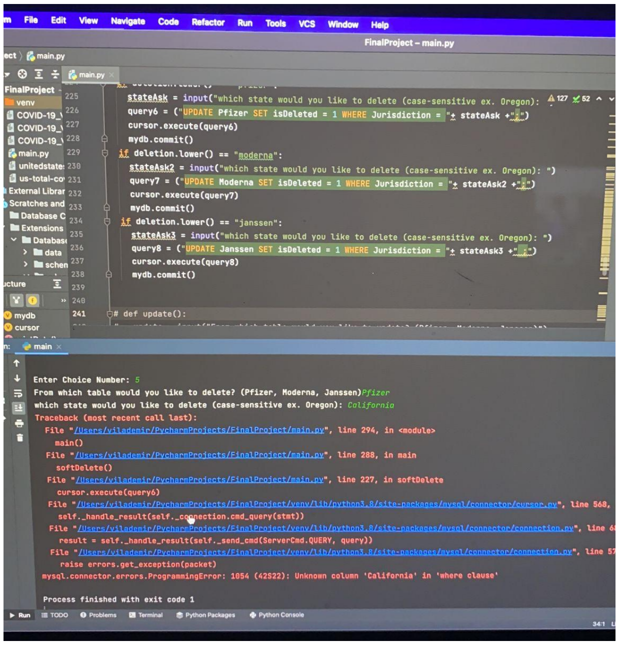
```
          Entity  TotalVacci    Cases  Janssen Dose
0        Alabama     2768204   540603          8500
1         Alaska      571368    69088          1900
2        Arizona     5372605   874065         12000
3       Arkansas     1900002   339162          5200
4     California    33109680  3770763         67600
5       Colorado     4880920   535536          9700
6    Connecticut     3512139   345720          6400
7       Delaware      825352   107175          1700
8        Florida    16431037  2296777         37000
9        Georgia     6799470  1092264         17600
10        Hawaii     1386097    34247          2600
11         Idaho     1132383   190357          2900
12      Illinois    10146731  1372582         17600
13       Indiana     4506166   739811         11400
14          Iowa     2564630   369638          5500
15        Kansas     2146247   314136          5000
```

As for my other functions, I wanted to touch on my delete and create a new record. These two functions are primarily for each Vaccine manufacturer table. The reason why only these two functions are used and update is not a part of this because each Vaccine table can have multiple entries of the same state. I wanted these tables to be more of a weekly update-type table so the user of the database can see how allocations are going each week, and if more or less need to be sent to each state.

For my update function, I utilized this for my US table, where the user can update Cases, Deaths and Confirmed Cases. I used Update in this case because I did not want to mess with the primary key, as adding a new entry creates duplicates to appear. The Update call keeps the same row in the table, but just updates each value for integrity purposes. My export function exports each table into a csv file, based on what actions were performed before. For example, if a user created a new record in one of the Vaccine Tables and then exported into a csv, that entry would be included in the report. This was done so the user can have a choice when to export the table, versus automatically generating a report that may not be needed.

My other functions within this database include printing and displaying the full tables, as well as displaying tables that have been queried (ie. options 7-10). Also, there is a soft delete function that sets the deleted value from null to 1 if the user decides to delete an entry. Lastly, once each task is performed, the user will be prompted whether or not they want to continue the database. This was done so the application will not quit after a single task is performed, and the user has to restart it each time.

Some problems I ran into during this project were syntax errors, as many times I would forget an apostrophe here or a set of parentheses there. In addition, another problem I ran into was exporting to a csv file. I figured out this problem by just using a pandas import function rather than sending the table to the csv from the query itself(using Outfile…).

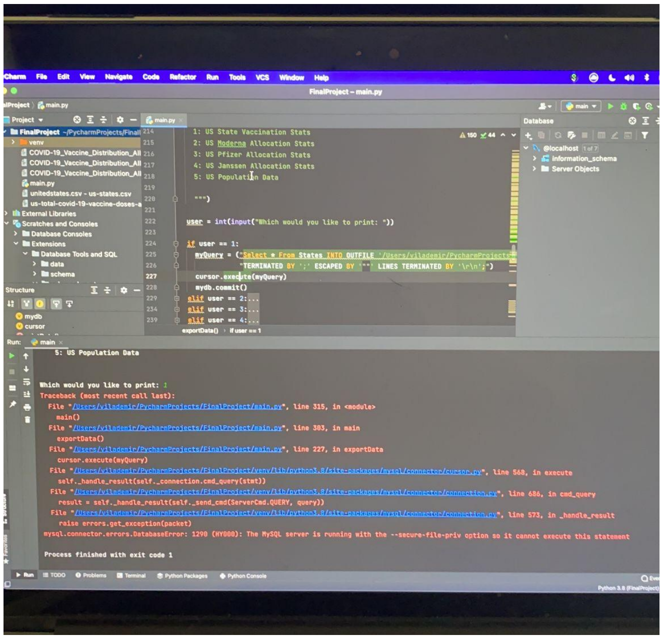


```
    stateAsk = input("which state would you like to delete (case-sensitive ex. Oregon):
    query6 = ("UPDATE Pfizer SET isDeleted = 1 WHERE Jurisdiction = "+ stateAsk +";")
    cursor.execute(query6)
    mydb.commit()
if deletion.lower() == "moderna":
    stateAsk2 = input("which state would you like to delete (case-sensitive ex. Oregon): ")
    query7 = ("UPDATE Moderna SET isDeleted = 1 WHERE Jurisdiction = "+ stateAsk2 +";")
    cursor.execute(query7)
    mydb.commit()
if deletion.lower() == "janssen":
    stateAsk3 = input("which state would you like to delete (case-sensitive ex. Oregon): ")
    query8 = ("UPDATE Janssen SET isDeleted = 1 WHERE Jurisdiction = "+ stateAsk3 +";")
    cursor.execute(query8)
    mydb.commit()


# def update():
```

```
Enter Choice Number: 5
From which table would you like to delete? (Pfizer, Moderna, Janssen)Pfizer
which state would you like to delete (case-sensitive ex. Oregon): California
Traceback (most recent call last):
  File "/Users/vilademir/PycharmProjects/FinalProject/main.py", line 294, in <module>
    main()
  File "/Users/vilademir/PycharmProjects/FinalProject/main.py", line 288, in main
    softDelete()
  File "/Users/vilademir/PycharmProjects/FinalProject/main.py", line 227, in softDelete
    cursor.execute(query6)
  File "/Users/vilademir/PycharmProjects/FinalProject/venv/lib/python3.8/site-packages/mysql/connector/cursor.py", line 568,
    self._handle_result(self._connection.cmd_query(stmt))
  File "/Users/vilademir/PycharmProjects/FinalProject/venv/lib/python3.8/site-packages/mysql/connector/connection.py", line 6
    result = self._handle_result(self._send_cmd(ServerCmd.QUERY, query))
  File "/Users/vilademir/PycharmProjects/FinalProject/venv/lib/python3.8/site-packages/mysql/connector/connection.py", line 5
    raise errors.get_exception(packet)
mysql.connector.errors.ProgrammingError: 1054 (42S22): Unknown column 'California' in 'where clause'

Process finished with exit code 1
```



```
    1: US State Vaccination Stats
    2: US Moderna Allocation Stats
    3: US Pfizer Allocation Stats
    4: US Janssen Allocation Stats
    5: US Population Data

    """)

user = int(input("Which would you like to print: "))

if user == 1:
    myQuery = ("Select * From States INTO OUTFILE '/Users/vilademir/PycharmProjects
                "TERMINATED BY ';' ESCAPED BY '"' LINES TERMINATED BY '\r\n';")
    cursor.execute(myQuery)
    mydb.commit()
elif user == 2:...
elif user == 3:...
elif user == 4:...
```

```
    5: US Population Data


Which would you like to print: 1
Traceback (most recent call last):
  File "/Users/vilademir/PycharmProjects/FinalProject/main.py", line 315, in <module>
    main()
  File "/Users/vilademir/PycharmProjects/FinalProject/main.py", line 303, in main
    exportData()
  File "/Users/vilademir/PycharmProjects/FinalProject/main.py", line 227, in exportData
    cursor.execute(myQuery)
  File "/Users/vilademir/PycharmProjects/FinalProject/venv/lib/python3.8/site-packages/mysql/connector/cursor.py", line 568, in execute
    self._handle_result(self._connection.cmd_query(stmt))
  File "/Users/vilademir/PycharmProjects/FinalProject/venv/lib/python3.8/site-packages/mysql/connector/connection.py", line 686, in cmd_query
    result = self._handle_result(self._send_cmd(ServerCmd.QUERY, query))
  File "/Users/vilademir/PycharmProjects/FinalProject/venv/lib/python3.8/site-packages/mysql/connector/connection.py", line 573, in _handle_result
    raise errors.get_exception(packet)
mysql.connector.errors.DatabaseError: 1290 (HY000): The MySQL server is running with the --secure-file-priv option so it cannot execute this statement

Process finished with exit code 1
```