**Technical Report: 3 Fast 3 Furious**

**Topic/DataSources:** State-level Covid data and notable variables all obtained via Kaggle

**Extract, Transform & Load**:

1. We created our first Jupyter Notebook <Obesity.ipynb> and used:
   1. Pandas
   2. Sqlalchemy
   3. numpy
2. We imported the following <csvData.csv> file into our notebook
3. We executed a quick read of this file to determine what columns and data were present
4. Next we pulled some basic statistical details of the data within the file (.describe)
5. We then renamed columns for ‘obesityRate’ and ‘Pop’ to ‘Obesity Rate” and “Population” into a new data frame for a cleaner presentation.
6. We then imported the <raw\_data.csv> file into our notebook
7. Again, we executed a quick read of the file to determine what columns and data were present
8. We then opened PGAdmin and created our Database <project\_db>
9. Within the database we created tables:
   1. First created table: <obesity\_state> with the following columns
      1. ‘State’ – Varchar (Primary Key)
      2. ‘Obesity Rate’ – Decimal
      3. ‘Population’ INT
   2. Second created table: <health\_state> with the following columns
      1. ‘State’ Varchar (Primary Key)
      2. ‘Health Expen’ INT
10. Next in our notebook, we connected to the database where our tables were created
11. We then ran a quick validation to identify the table names for the newly created tables above
12. We had to reformat our ‘Health Expen’ field to remove the “$” character from the field values
13. We then appended the newly formatted values to the <health\_state> SQL table
14. We executed a quick read validation of that field to confirm the formatting was completed correctly
15. Back in our PGAgmin, we then joined the <obesity\_state> and <health\_state> tables on the ‘State’ key from each table
16. We then created a combined table <combined\_df> of the first 2 tables with the following field/attributes:
    1. ‘State’ – Varchar (Primary Key)
    2. ‘Obesity Rate’ – Decimal
    3. ‘Population’ – INT
    4. ‘Health Expen’ – INT
17. We exported the data from the <combined\_df> table to the <State,ObesityRate, Population,Health Expenryan.csv> file
18. Next we created the following table:
    1. <final\_state> with the following columns:
       1. ‘State’ – Varchar (Primary Key)
       2. ‘Tested’ – Varchar
       3. ‘Infected’ – INT
       4. ‘Deaths’ – INT
       5. ‘ICU Beds’ – INT
       6. ‘Age 0-25’ – Decimal
       7. ‘Age 26-54’ – Decimal
       8. ‘Age 55+’ – Decimal
    2. We then joined the <final\_state> and <combined\_df> tables on the ‘State’ key from each table
19. We then created a second notebook <adding\_covid.ipynb> and used:
    1. Pandas
    2. Sqlalchemy
    3. Numpy
20. We imported the following <covid.csv> file into our notebook
21. We executed a quick read of this file to determine what columns and data were present
22. Next we performed a validation to identify all of the Unique values for the ‘State’ field
23. We executed a count of the unique values in the State field to determine if there were 50 states as expected
24. We noticed an extra row for the District of Columbia and we executed a drop command to remove this value (we noted this as Row 7)
25. We ran another read to confirm we were left with 50 rows for the 50 expected States
26. We then recreated a new data frame only pulling in the columns we were interested in:
    1. State
    2. Tested
    3. Infected
    4. Deaths
    5. ICU Beds
    6. Age 0-25
    7. Age 26-54
    8. Age 55+
27. We then imported the <State,ObesityRate,Population,Health Expenryan.csv> file
28. Next we again connected to the SQL <project\_db> database where our tables were created
29. We ran a quick validation to determine what tables existed within the db
30. We appended the values from the newly formatted data frame to the <combined\_df> table
31. We then ran an export from the <final\_state> table of all the data into the <Ryan’s Table.csv> file.
32. Our final production database containing our tables above are all considered a relational database.