For week 8, I continued working on my SQL queries. I had written them the week before, but I began to modify them and gain better understanding of how they worked. For instance, at first I was not sure how to navigate between two tables in the SQL or write a query that got them both to work. However, I was able to understand the use of JOIN clause which is what I used for the tables. There is the join outer left, join outer right, join inner and join outer and depending on what one needed to do, they would use it as they seemed fit. There were also other clauses and functions I learned like the SELECT, which would select what one needed from the table. The SELECT * would select everything and usually you would need other conditions with the SELECT. There was also the WHERE which we use after the FROM and which is followed by a condition. There was the COUNT function which would return a number depending on the SELECT statement and the conditions it was given.

As I learned all these I was able to modify my queries from the questions I had written earlier. I was also able to understand them and know how to combine them. For instance, the query 'SELECT org_name, post_writer FROM science_organization FULL OUTER JOIN statements ON (science_organization.org_name = statements.post_writer);' would be used to know the organizations in the database that had post writers.

The breakdown of the query is SELECT org_name, post_writer: these two are the columns that I need to use from the different tables.

FROM science_organization: This was selecting from table one

FULL OUTER JOIN statements: This was joining with table two

ON (science_organization.org_name = statements.post_writer): This was putting them together to give us our representation.

I also checked that the query did what I wanted it to do in the database. This was correct and I went further to modify the other queries as many of them were not returning the result I wanted.

After this I went on to start creating an object relational model. This is what I would use to link my database. First I had to instal python virtual environment (pipenv), navigate to my project directory (Database-DrivenApp) and install peewee. Then I created a file in atom which I called correctmodels.py which is where I would develop all my models. I went through the skeleton I was given and had to understand each part as I created my model. For instance, the Class Statements in the python file would be mapped to the table Statements in the database, and the Class science_org would be mapped to the science_org in the database. The fields in these classes would be mapped to the columns in the database. I also learned about the

```
def __str__(self):
        return str(self.ID)
```

Which would return results when we print the instances of the object. At first I set those both to self.org_id but then after I spoke to Dr Jones I realized that it would be much better having something like org_name which would showcase things more clearly.