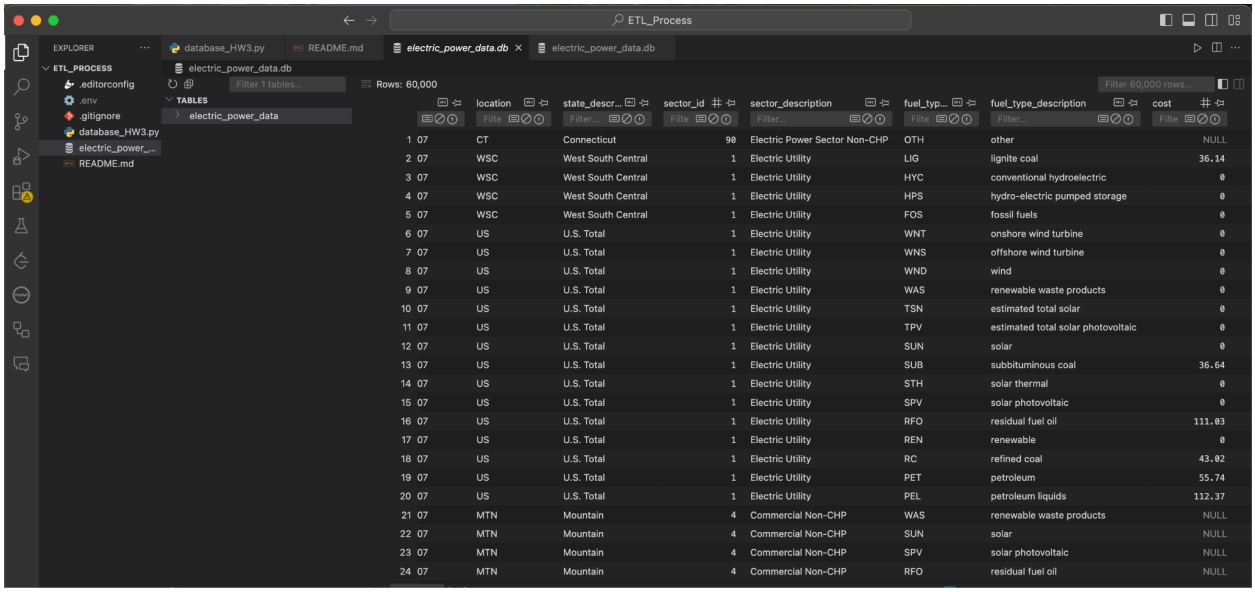
Mitchell Cootauco

10/8/24

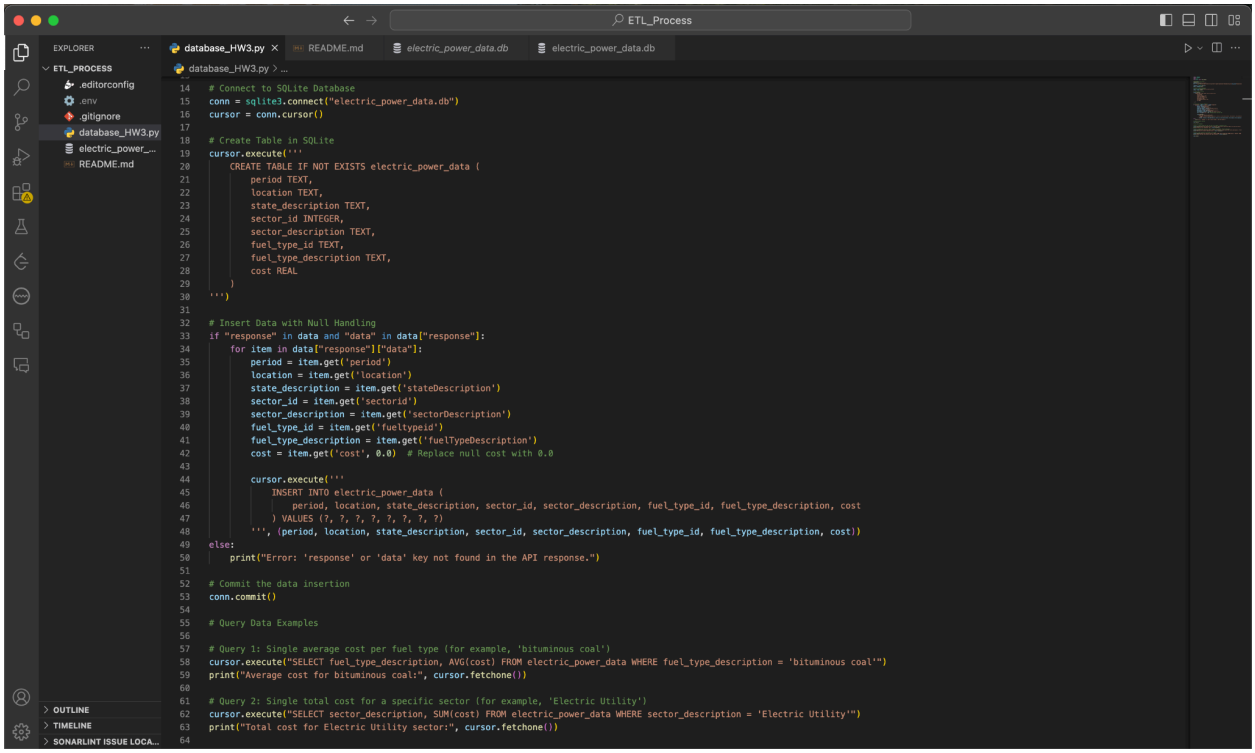
CMSI 620 HW #3

This was my first time using SQLite, so setting up and interacting with a local SQL database in Python was a great learning opportunity. I gained hands-on experience creating tables, structuring schemas, and handling data types, which clarified how relational databases operate. I learned how to define and develop tables directly from Python using sqlite3, setting up a schema to capture essential fields like period, location, fuel_type, and cost. I also implemented SQL commands in Python to insert records into the table, handle null values, and execute queries to analyze data.

- **Query 1**: SELECT fuel_type_description, AVG(cost) FROM electric_power_data WHERE fuel_type_description = 'bituminous coal'
    - Calculated the average cost for “bituminous coal” across records
- **Query 2**: SELECT sector_description, SUM(cost) FROM electric_power_data WHERE sector_description = 'Electric Utility'
    - Summed the costs for the “Electric Utility” sector
- **Query 3**: SELECT fuel_type_description, COUNT(*) FROM electric_power_data WHERE period = '2024-07' GROUP BY fuel_type_description LIMIT 3
    - Counted records per fuel type for July 2024



		location	state_descr...	sector_id	sector_description	fuel_typ...	fuel_type_description	cost
1	07	CT	Connecticut	90	Electric Power Sector Non-CHP	OTH	other	NULL
2	07	WSC	West South Central	1	Electric Utility	LIG	lignite coal	36.14
3	07	WSC	West South Central	1	Electric Utility	HYC	conventional hydroelectric	0
4	07	WSC	West South Central	1	Electric Utility	HPS	hydro-electric pumped storage	0
5	07	WSC	West South Central	1	Electric Utility	FOS	fossil fuels	0
6	07	US	U.S. Total	1	Electric Utility	WNT	onshore wind turbine	0
7	07	US	U.S. Total	1	Electric Utility	WNS	offshore wind turbine	0
8	07	US	U.S. Total	1	Electric Utility	WND	wind	0
9	07	US	U.S. Total	1	Electric Utility	WAS	renewable waste products	0
10	07	US	U.S. Total	1	Electric Utility	TSN	estimated total solar	0
11	07	US	U.S. Total	1	Electric Utility	TPV	estimated total solar photovoltaic	0
12	07	US	U.S. Total	1	Electric Utility	SUN	solar	0
13	07	US	U.S. Total	1	Electric Utility	SUB	subbituminous coal	36.64
14	07	US	U.S. Total	1	Electric Utility	STH	solar thermal	0
15	07	US	U.S. Total	1	Electric Utility	SPV	solar photovoltaic	0
16	07	US	U.S. Total	1	Electric Utility	RFO	residual fuel oil	111.03
17	07	US	U.S. Total	1	Electric Utility	REN	renewable	0
18	07	US	U.S. Total	1	Electric Utility	RC	refined coal	43.02
19	07	US	U.S. Total	1	Electric Utility	PET	petroleum	55.74
20	07	US	U.S. Total	1	Electric Utility	PEL	petroleum liquids	112.37
21	07	MTN	Mountain	4	Commercial Non-CHP	WAS	renewable waste products	NULL
22	07	MTN	Mountain	4	Commercial Non-CHP	SUN	solar	NULL
23	07	MTN	Mountain	4	Commercial Non-CHP	SPV	solar photovoltaic	NULL
24	07	MTN	Mountain	4	Commercial Non-CHP	RFO	residual fuel oil	NULL




```python
# Connect to SQLite Database
conn = sqlite3.connect("electric_power_data.db")
cursor = conn.cursor()

# Create Table in SQLite
cursor.execute('''
    CREATE TABLE IF NOT EXISTS electric_power_data (
        period TEXT,
        location TEXT,
        state_description TEXT,
        sector_id INTEGER,
        sector_description TEXT,
        fuel_type_id TEXT,
        fuel_type_description TEXT,
        cost REAL
    )
''')

# Insert Data with Null Handling
if "response" in data and "data" in data["response"]:
    for item in data["response"]["data"]:
        period = item.get('period')
        location = item.get('location')
        state_description = item.get('stateDescription')
        sector_id = item.get('sectorid')
        sector_description = item.get('sectorDescription')
        fuel_type_id = item.get('fueltypeid')
        fuel_type_description = item.get('fuelTypeDescription')
        cost = item.get('cost', 0.0)  # Replace null cost with 0.0

        cursor.execute('''
            INSERT INTO electric_power_data (
                period, location, state_description, sector_id, sector_description, fuel_type_id, fuel_type_description, cost
            ) VALUES (?, ?, ?, ?, ?, ?, ?, ?)
        ''', (period, location, state_description, sector_id, sector_description, fuel_type_id, fuel_type_description, cost))
else:
    print("Error: 'response' or 'data' key not found in the API response.")

# Commit the data insertion
conn.commit()

# Query Data Examples

# Query 1: Single average cost per fuel type (for example, 'bituminous coal')
cursor.execute("SELECT fuel_type_description, AVG(cost) FROM electric_power_data WHERE fuel_type_description = 'bituminous coal'")
print("Average cost for bituminous coal:", cursor.fetchone())

# Query 2: Single total cost for a specific sector (for example, 'Electric Utility')
cursor.execute("SELECT sector_description, SUM(cost) FROM electric_power_data WHERE sector_description = 'Electric Utility'")
print("Total cost for Electric Utility sector:", cursor.fetchone())
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  COMMENTS  PORTS

(base) mitchell_cootauco@Mitchells-MacBook-Air ETL_Process % python database_HW3.py
Average cost for bituminous coal: ('bituminous coal', 51.968378378378375)
Total cost for Electric Utility sector: ('Electric Utility', 93032.9)
Record count per fuel type in July 2024 (limited): [('all coal products', 1675), ('all fuels', 2908), ('all renewables', 2558)]
(base) mitchell_cootauco@Mitchells-MacBook-Air ETL_Process %
```