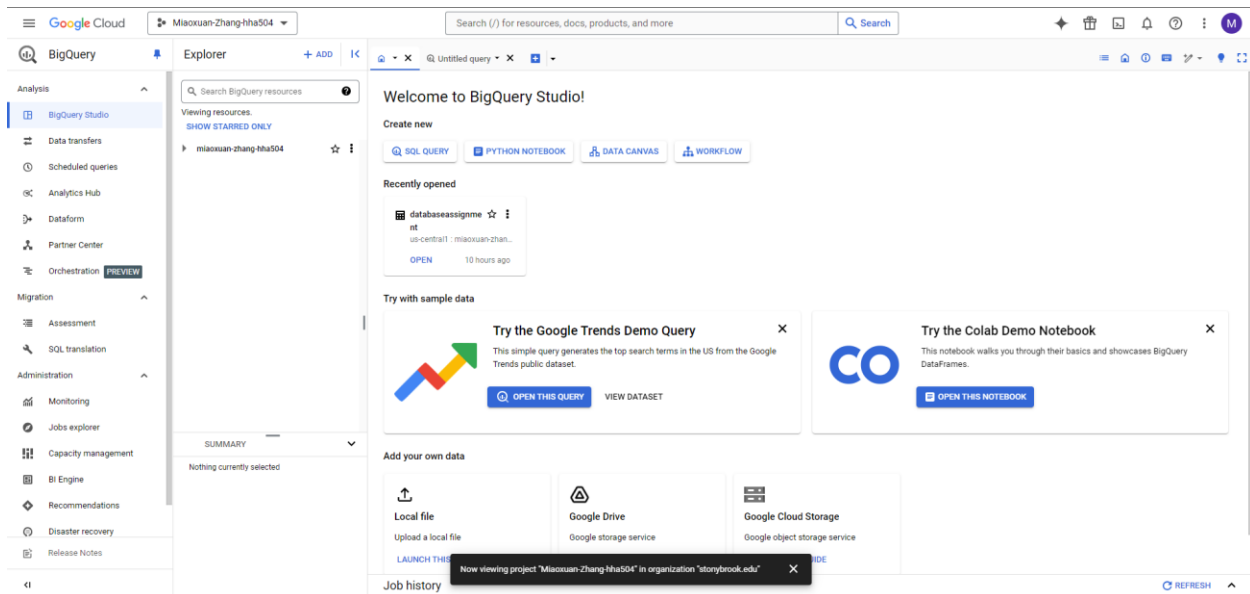


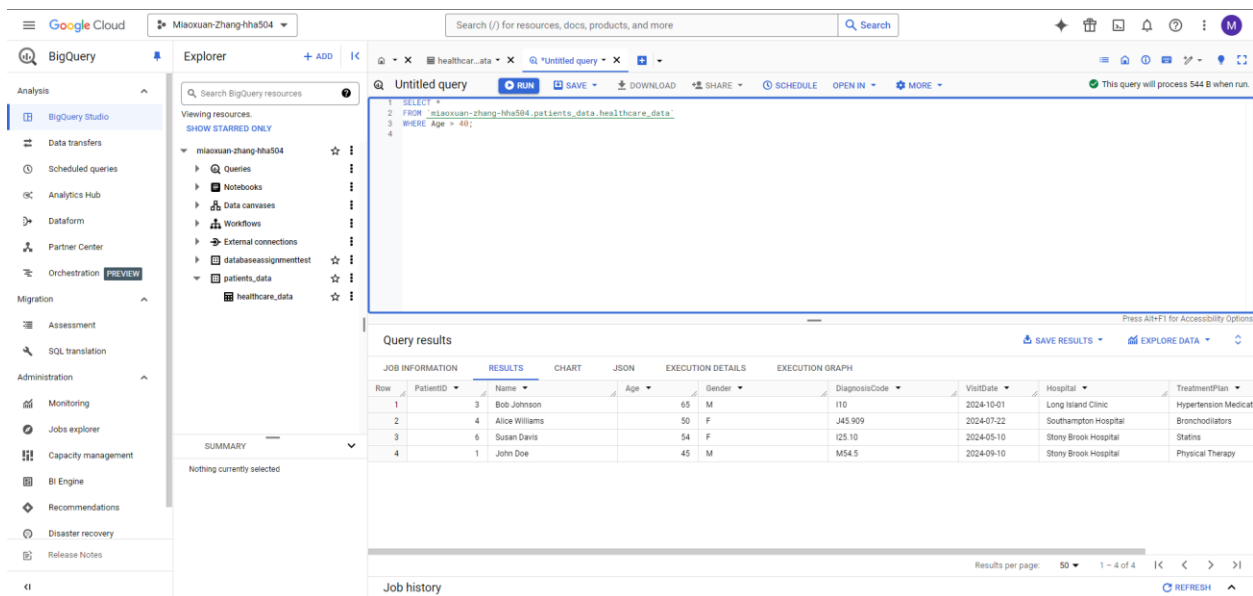
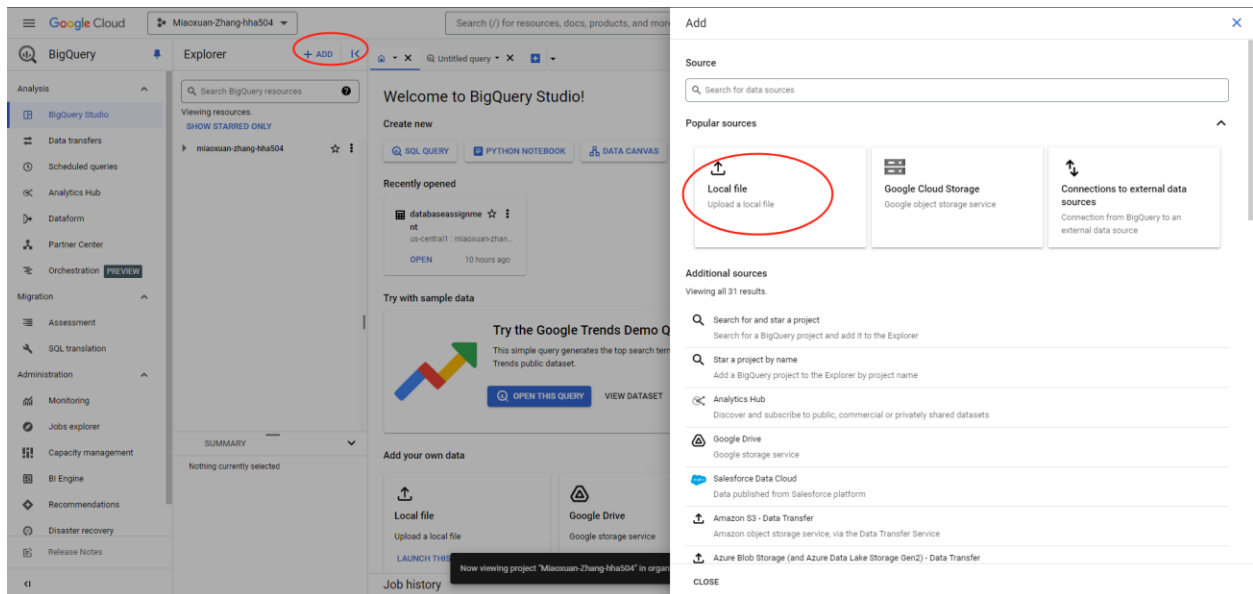
Assignment: Working with Managed No-SQL Databases

Instructions

1. Start and Configure Databases

- **Google BigQuery (GCP):**
 - Navigate to BigQuery in the Google Cloud Console.
 - Use your student account project to create a new dataset in BigQuery.
 - Upload the provided healthcare dataset (CSV) into a table within your dataset.
 - Note the connection details and the query editor interface.

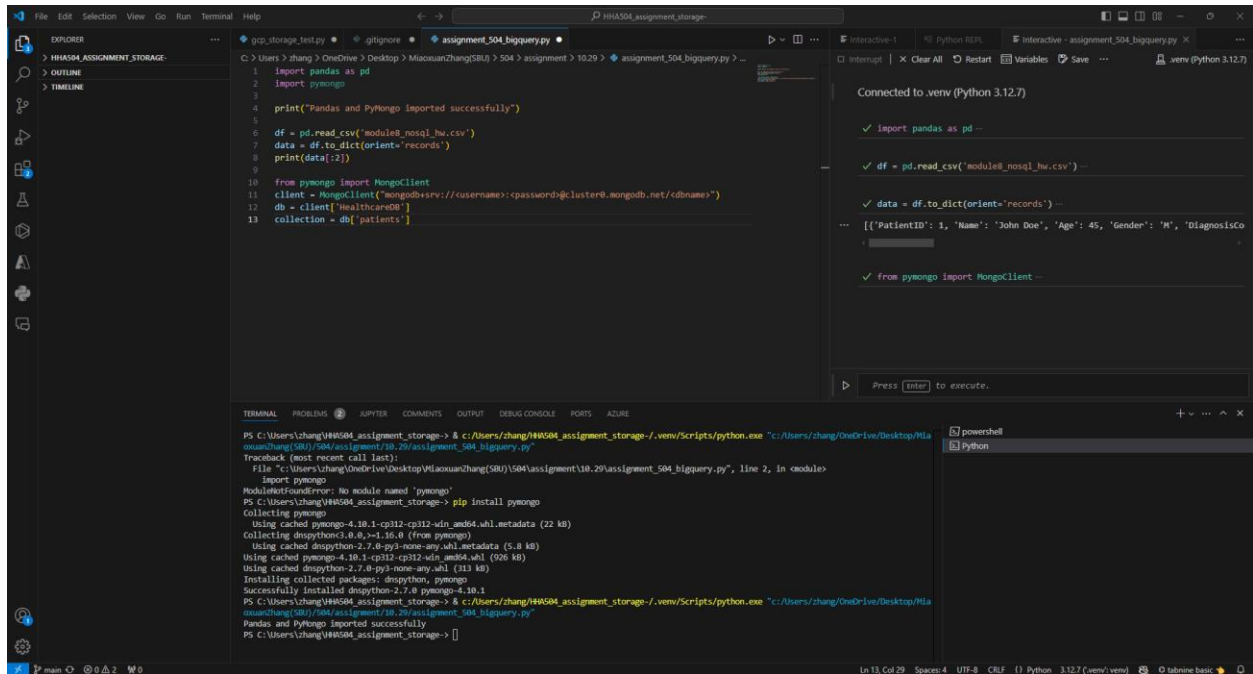




• MongoDB Atlas (Cloud):

- Go to [MongoDB Atlas](#) and register for the free tier using your Stony Brook email.
- Create a new database instance and configure it with basic settings.
- Insert the provided healthcare dataset into a collection, ensuring each row is converted to a JSON document.

- Document the steps and connection details.



```
1 import pandas as pd
2 import pymongo
3
4 print("Pandas and PyMongo Imported successfully")
5
6 df = pd.read_csv('module8_nosql_hu.csv')
7 data = df.to_dict(orient='records')
8 print(data[:2])
9
10 from pymongo import MongoClient
11 client = MongoClient('mongodb://username:password@cluster0.mongodb.net/dbname')
12 db = client['HealthcareDB']
13 collection = db['patients']
```

Connected to .venv (Python 3.12.7)

```
✓ import pandas as pd --
✓ df = pd.read_csv('module8_nosql_hu.csv') --
✓ data = df.to_dict(orient='records') --
... [{"PatientID": 1, "Name": "John Doe", "Age": 45, "Gender": "M", "Diagnosis": "Hypertension"}]
```

Press **Enter** to execute.

PS C:\Users\zhang\HH4504_assignment_storage> & c:\Users\zhang\HH4504_assignment_storage\.venv\Scripts\python.exe "c:\Users\zhang\OneDrive\Desktop\H4504\assignment\10.20\assignment_504_bigquery.py"

Traceback (most recent call last):

File "c:\Users\zhang\OneDrive\Desktop\H4504\assignment\10.20\assignment_504_bigquery.py", line 2, in <module>

import pymongo

ModuleNotFoundError: No module named 'pymongo'

PS C:\Users\zhang\HH4504_assignment_storage> pip install pymongo

Collecting pymongo

Using cached pymongo-4.10.1-cp312-cp312-win_amd64.whl.metadata (22 kB)

Collecting dnspython<2.0.0,=>1.16.0 (from pymongo)

Using cached dnspython-2.7.0-py3-none-any.whl.metadata (5.8 kB)

Using cached dnspython-2.7.0-py3-none-any.whl (131 kB)

Using cached dnspython-2.7.0-py3-none-any.whl (131 kB)

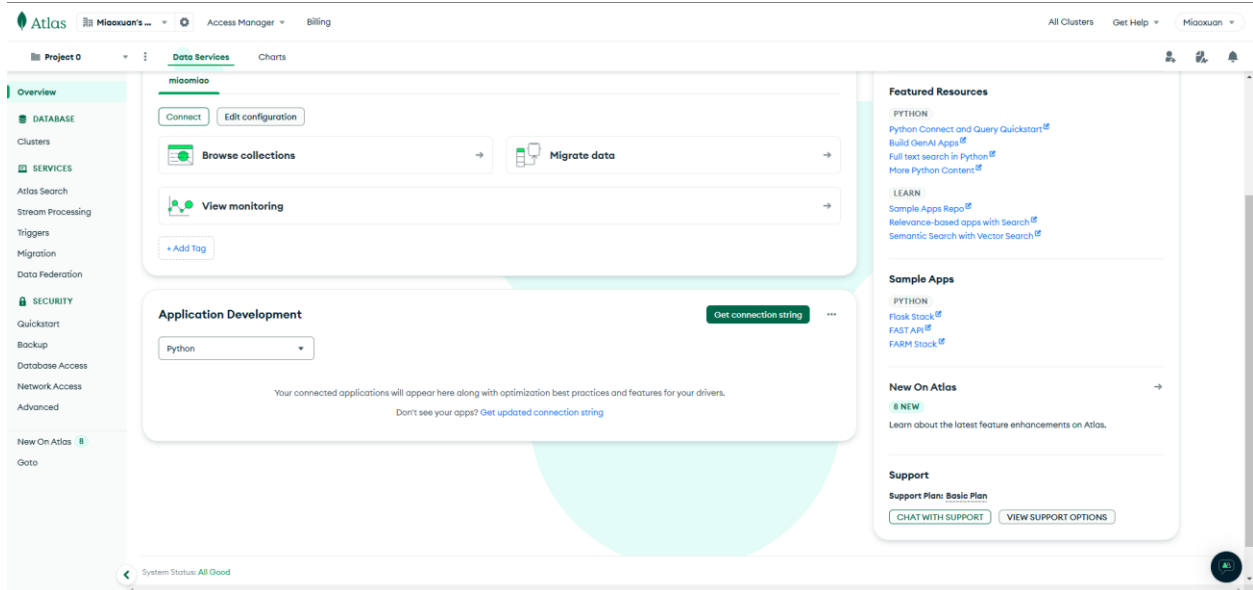
Installing collected packages: dnspython, pymongo

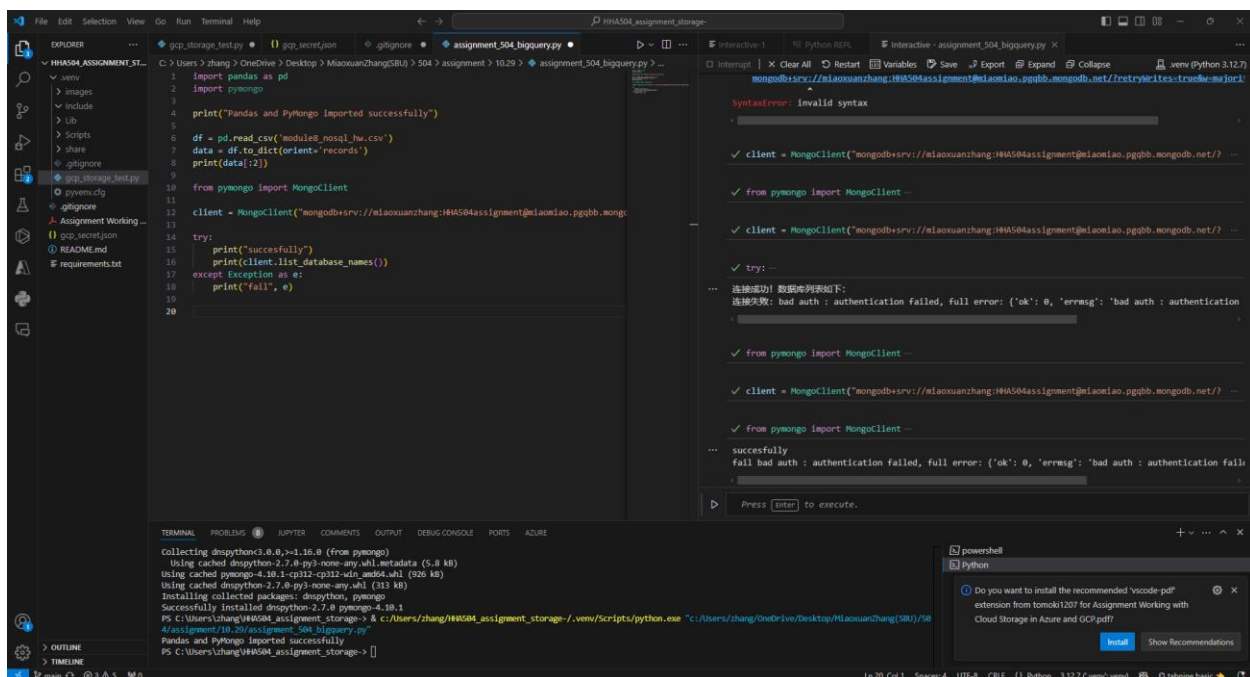
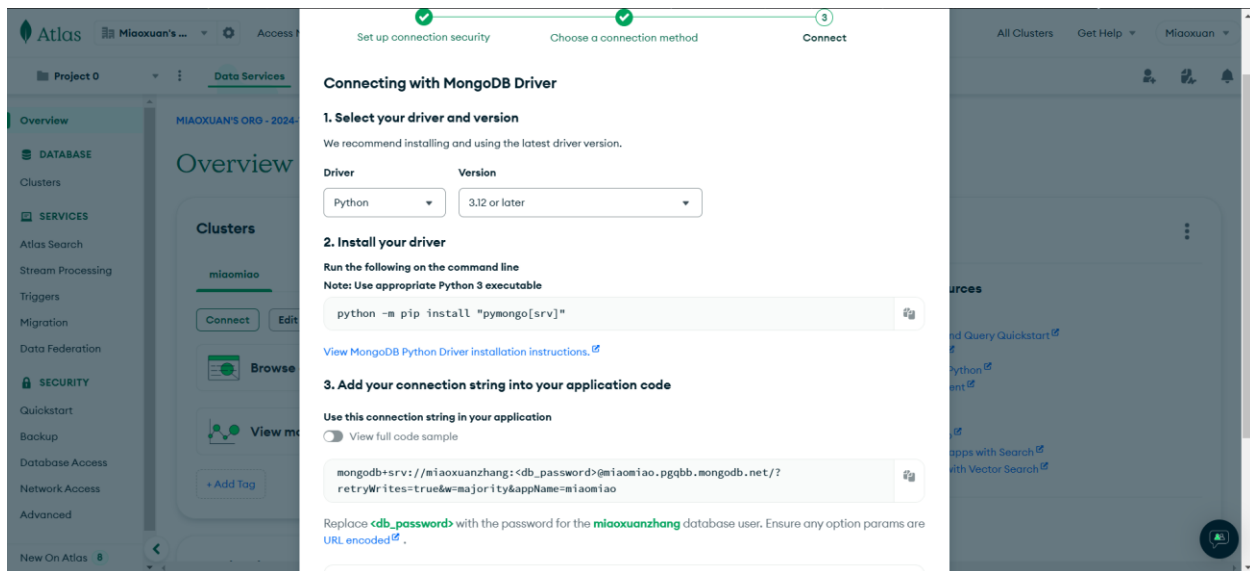
Successfully installed dnspython-2.7.0 pymongo-4.10.1

PS C:\Users\zhang\HH4504_assignment_storage> & c:\Users\zhang\HH4504_assignment_storage\.venv\Scripts\python.exe "c:\Users\zhang\OneDrive\Desktop\H4504\assignment\10.20\assignment_504_bigquery.py"

Pandas and PyMongo Imported successfully

PS C:\Users\zhang\HH4504_assignment_storage> []





I can't connect to the cloud.

• Redis Cloud:

- Go to [Redis Cloud](#) and sign up for a free tier account using your Stony Brook email.
- Set up a new Redis database instance.

- Use PatientID as the key and the rest of the patient data as the value (either as a serialized JSON string or separate fields).
- Document the process and connection details.

The screenshot shows the Redis Cloud interface. The left sidebar contains navigation links: Databases, Subscriptions, Data Access Control, Access Management, Logs, Account Settings, Usage Report, Billing & Payments, Redis Copilot, Download Center, Support, and Documentation. The main content area is titled "Databases" and features a search bar and filter tabs (Status, Subscription, Capabilities). A table lists the databases:

Status	Database name	Subscription	Endpoint	Memory usage	Capabilities
●	Miaoxuan-free-db	#2464288 Miaoxuan-free-db	Connect	3MB / 30MB (10.1%)	...

Below the table, it says "Showing 1 out of 1 rows". A modal titled "Connect to Miaoxuan-free-db" is open on the right, showing the RedisInsight connection options. It includes a "Download" button, a "Try it now" button, and a "Try the Node.js starter app in Github Codespaces" button.

The screenshot shows the Redis Cloud "Data Access Control" page. The left sidebar is the same as the previous screenshot. The main content area is titled "Data Access Control" and has tabs for Users, Roles, and Redis ACLs. The "Users" tab is selected, showing a table with the following data:

Username	Role	Password	Status
miaomiao	miaomiao	*****	●

At the bottom of the table, it says "Items per page: 10" and "Page 1 of 1".

```

19
20
21 import pandas as pd
22 import redis
23 import json
24 df = pd.read_csv('module8_nosql_hw.csv')
25 r = redis.StrictRedis(
26     host='redis-13712.c52.us-east-1-4.ec2.redns.redis-cloud.com',
27     port=13712,
28     password='KJ3@zHcwmjU!bNZ', # 替换为正确的 Redis 密码
29     decode_responses=True
30 )
31
32 for _, row in df.iterrows():
33     patient_data = row.to_dict()
34     r.set(patient_data['PatientID'], json.dumps(patient_data))
35
36 print("Successfully store in Redis!")

```

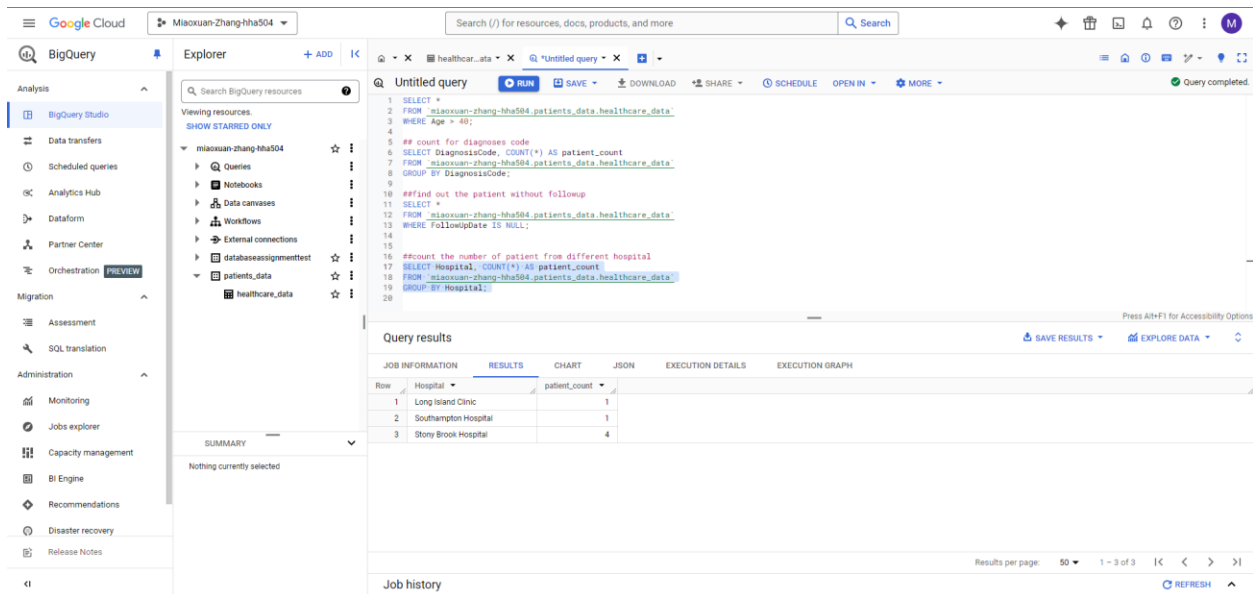
2. Explore BigQuery (GCP)

- **BigQuery:**

- In the Google Cloud Console, run a simple SQL query against the dataset you uploaded:

```
SELECT * FROM `your_project_id.your_dataset_id.your_table_id` WHERE Age > 40
```

- Monitor the usage and cost associated with running the query.



3. Modify and Explore the Data in MongoDB Atlas and Redis Cloud

- **MongoDB Atlas:**

- Insert the dataset into MongoDB Atlas, converting each row into a JSON-like document.
- Ensure fields like PatientID and VisitDate are treated appropriately (i.e., unique identifiers and date types).
- Run a simple query to retrieve patient data based on a condition (e.g., Age > 40).

- **Redis Cloud:**

- Insert key-value pairs where the PatientID is the key, and the rest of the patient data is the value.
- For example, retrieve the data for PatientID=1, then update the TreatmentPlan value.
- Explore Redis's capabilities to update and query the dataset, e.g., retrieving all data for PatientID=1.

4. Describe Your Experience

- For each of the three services (BigQuery, MongoDB Atlas, Redis Cloud), document your experience creating and working with the healthcare dataset:
 - Describe the setup process and any configuration steps.

- Share your reflections on the interface and usability of each platform.

5. Submit Your Work

- Create a Markdown document that includes:
 - Screenshots of the database creation and configuration process in BigQuery, MongoDB Atlas, and Redis Cloud.
 - The SQL query run in BigQuery and the results.
 - Documentation of your experience and reflections on working with each platform (BigQuery, MongoDB Atlas, Redis Cloud).
- Commit and push this Markdown document, along with the screenshots and query results, to your GitHub repository.

Deliverables

- A Markdown document in a GitHub repository called HHA504_assignment_nosql_dbs that includes:
 - Screenshots of database creation and configuration for BigQuery, MongoDB Atlas, and Redis Cloud.
 - BigQuery dataset creation and query results.
 - Reflections on working with each of the three platforms.