CSP 586 Assignment 4

Atharva Kadam

Requirement 1: Create a table that shows the dataset name and the URL listed on the City of Chicago data portal and the requirements in the requirements specification document of the Chicago Business Intelligence for Strategic Planning that needs that data source.

	Dataset Name	URL
Requirement 1	Taxi Trips COVID-19	https://data.cityofchicago.org/Transportation/Taxi-Trips/wrvz-psew https://data.cityofchicago.org/Health-Human-Services/COVID-19-Cases-Tests-and-Deaths-by-ZIP-Code/yhhz-zm2v
Requirement 2	Taxi Trips Neighborhood COVID-19	https://data.cityofchicago.org/Transportation/Taxi-Trips/wrvz-psew https://www.chicagotribune.com/chi-community-areas-htmlstory.html https://data.cityofchicago.org/Health-Human-Services/COVID-19-Cases-Tests-and-Deaths-by-ZIP-Code/yhhz-zm2v
Requirement 3	CCV Taxi Trips	https://data.cityofchicago.org/Health-Human-Services/Chicago-COVID-19-Community-Vulnerability-Index-CCV/xhc6-88s9 https://data.cityofchicago.org/Transportation/Taxi-Trips/wrvz-psew
Requirement 4	Taxi Trips	https://data.cityofchicago.org/Transportation/Taxi-Trips/wrvz-psew
Requirement 5	Unemployment Building Permit	https://data.cityofchicago.org/Health-Human-Services/Public-Health-Statistics-Selected-public-health-in/iqnk-2tcu/data https://www.chicago.gov/city/en/depts/bldgs/dataset/building_permits.html
Requirement 6	Unemployment Building Permit	https://data.cityofchicago.org/Health-Human-Services/Public-Health-Statistics-Selected-public-health-in/iqnk-2tcu/data https://www.chicago.gov/city/en/depts/bldgs/dataset/building_permits.html

Requirement 2: Does every data source (dataset) have all attributes needed for every report/query required for Chicago Business Intelligence for Strategic Planning? For example, does Building Permits (https://www.chicago.gov/city/en/depts/bldgs/dataset/building_permits.html) dataset have Zip Code? does the Health Stat dataset (https://data.cityofchicago.org/Health-Human-Services/Public-HealthStatistics-Selected-public-health-in/iqnk-2tcu/data) and (https://data.cityofchicago.org/resource/iqnk-2tcu.json) to find the unemployment and poverty level data for the different community areas have the Zip Code?

Every dataset doesn't possess all the attributes needed for every query required for Chicago Business Intelligence. Zip Code is not present in building permits, health stat, unemployment dataset.

Requirement 3: What are the tables and their attributes, data types that you created for the Data Lake. What is the database engine that you used for your Data Lake? Explain in detail if you need to create a table for every dataset and if you need to create a new table to store results of merging data from different datasets.

Following are the databases and their respective attributes:-

1. Taxi_trip-

```
trip_id,
trip_start_timestamp,
trip_end_timestamp,
pickup_centroid_latitude,
pickup_centroid_longitude,
dropoff_centroid_latitude,
dropoff_centroid_longitude,
pickup_zip_code,
dropoff_zip_code
```

2. Unemployment-

```
community_area,
community_area_name,
per_capita_income,
unemployment_rate
```

3. Building Permit-

```
permit_id,
permit_name,
community_area,
ZIP_code,
```

```
            centroid_longitude,
            centroid_latitude
```

I have used PostgresSQL for my data lake. I have created a new table for all my data sets, so each individual dataset has its own table.

Requirement 4: Create a table that shows the dataset name and the URL listed and whether it has Neighborhood Names, Community Areas, and Zip Codes.

Dataset	URL	
Taxi Trips	https://data.cityofchicago.org/Transportation/Taxi-Trips/wrvz-psew	No
COVID-19	https://data.cityofchicago.org/Health-Human-Services/COVID-19-Cases-Tests-and-Deaths-by-ZIP-Code/yhhz-zm2v	Only Zip code exists
Neighborhood	https://www.chicagotribune.com/chi-community-areas-htmlstory.html	Yes
CCVI	https://data.cityofchicago.org/Health-Human-Services/Chicago-COVID-19-Community-Vulnerability-Index-CCV/xhc6-88s9	yes
Unemployment	https://data.cityofchicago.org/Health-Human-Services/Public-Health-Statistics-Selected-public-health-in/iqnk-2tcu/data	Only Zip code does not exists
Building Permit	https://www.chicago.gov/city/en/depts/bldgs/dataset/building_permits.html	Only Zip code exists

Requirement 5: Explain how you cross-reference Neighborhood Names, Community Areas, and Zip Codes.

Most datasets have latitude and longitude values which can be converted to ZIP codes with the help of geocode. Tables like building permit and unemployment have community areas and names mentioned which can be used to cross reference those tables. Either of the two can be used as foreign key to address table which has zip codes.

Requirement 6: Explain why there are two data sources for the transportation: (https://data.cityofchicago.org/Transportation/TaxiTrips/wrvz-psew) and (https://data.cityofchicago.org/Transportation/Transportation-NetworkProviders-Trips/m6dm-c72p)

Taxi journeys were reported to the City of Chicago as part of its regulatory function. The Taxi ID is consistent for every given taxi medallion number but does not show the number, Census Tracts are concealed in some circumstances, and times are rounded to the nearest 15 minutes to safeguard privacy while allowing for aggregate studies.
In certain cases, census tracts are suppressed, and times are rounded to the nearest 15 minutes. Fares and tips are rounded to the nearest $2.50 and $1.00, respectively.
Since November 2018, Transportation Network Providers (also known as ridesharing firms) have been compelled by statute to report all trips to the City of Chicago.

Requirement 7: List all reports that are needed to meet the requirements for the Chicago Business Intelligence for Strategic Planning project. For example, a report is needed to provide all information of the taxi trips from O'Hare and Midway airports to the different zip codes.

- *Business Intelligence Report* - tracking and forecasting events that have direct or indirect negative or positive impacts on businesses and neighborhoods in different zip codes within the city of Chicago. Forecast daily, weekly, and monthly traffic patterns.
- *Airport Report* - track trips from these airports to the different zip codes and the reported COVID-19 positive test cases.
- *CCVI Report* - track the number of taxi trips from/to the neighborhoods that have CCVI Category with value HIGH.
- *Top 5 neighborhood Report* - Find top 5 neighborhoods with highest unemployment rate and poverty rate.
- *"Little Guys" Business Report* - Grant loans for applicants with the lowest number of permits - new construction and income less than 30,000.

Requirement 8: Create a table that has the name of every report you identified in the prior requirement, and document for every report the following:

Report Name	Data Source	Data API
Business Intelligence	Taxi Trips COVID-19	https://data.cityofchicago.org/resource/m6dm-c72p.json https://data.cityofchicago.org/resource/yhhz-zm2v.json
Airport	Taxi Trips	https://data.cityofchicago.org/resource/m6dm-c72p.json

	COVID-19	https://data.cityofchicago.org/resource/yhhz-zm2v.json
CCVI	CCVI Taxi Trips	https://data.cityofchicago.org/resource/xhc6-88s9.json https://data.cityofchicago.org/resource/m6dm-c72p.json
Top 5 neighborhood	Unemployment Building Permit	https://data.cityofchicago.org/api/views/INLINE/rows.json?accessType=DOWNLOAD https://www.chicago.gov/city/en/depts/bldgs/dataset/building_permits.json
"Little Guys" Business	Unemployment Building Permit	https://data.cityofchicago.org/api/views/INLINE/rows.json?accessType=DOWNLOAD https://www.chicago.gov/city/en/depts/bldgs/dataset/building_permits.json

Dataset	Json
Taxi Trips	https://data.cityofchicago.org/resource/m6dm-c72p.json
COVID-19	https://data.cityofchicago.org/resource/yhhz-zm2v.json
CCVI	https://data.cityofchicago.org/resource/xhc6-88s9.json
Unemployment	https://data.cityofchicago.org/api/views/INLINE/rows.json?accessType=DOWNLOAD
Building Permit	

For taxi dataset:

1. Trip_id
2. Trip_start_timestamp
3. Trip_end_timestamp
4. Pickup_centroid_latitude
5. Pickup_centroid_longitude

6. Dropoff_centroid_latitude
7. Dropoff_centroid_longitude
8. Pickup_zip_code
9. dropoff_zip_code

For COVID dataset:

1. ZIP_code
2. caseID
3. caseRate weekly

For CCVI:

1. Zip
2. CCVI_score
3. Community_name
4. ZIP_code

For Unemployment :

1. commuinty_area
2. commuinty_area_name
3. per_capita_income
4. unemployment_rate

For Building Permit:

1. permit_id
2. permit_number
3. permit_type
4. Community_area
5. ZIP_code
6. centroid_latitude
7. centroid_longitude

We are using defensive coding to prepare and preprocess data. We are avoiding any null or wrong data.

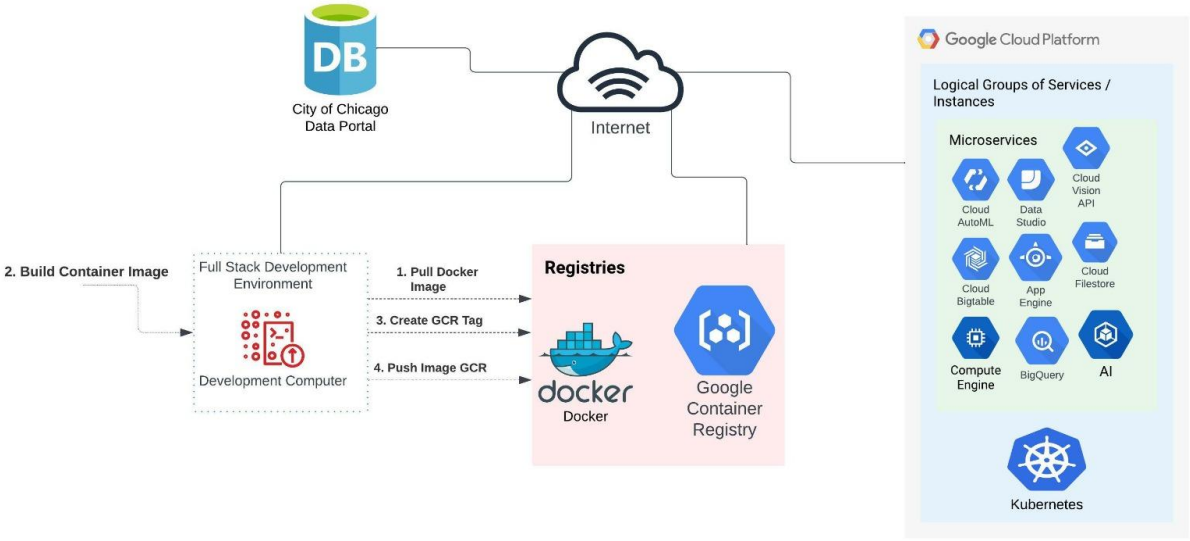
Requirement 9:

Requirement 10:

Requirement 11:

Consider the following architecture and design diagram to build the cloud-native microservice for the Chicago Business Intelligence for Strategic Planning project. Identify and document the steps and workflows for data collection, data preprocessing, and name the microservices needed for deployment; your documentation and annotation of the workflows and steps should be similar to the annotation that you see on the diagram below.

Following are the services of Google Cloud will be used as micro services:
App Engine: To deploy react app on google cloud, Build, deploy, debug, and monitor Node.js applications, Flask, Go App Server
Compute Engine: NGINX
Fire store: Persist your data
Cloud Storage: Store file uploads
Google Cloud Operation Suite: Monitor your app
Big Query & AI Platform: StatModel, Prophet
AI Platform Training, AI Platform Prediction and Cloud Storage: Keras
Cloud Natural Language: NLP Sentiment



Requirement 12:
List the names of the microservices and their purposes that you decided are needed to implement the Chicago Business Intelligence for Strategic Planning project. For example, you might state that you need to pull Postgres image from Docker to be your database engine.

- Pull Postgres image from Docker to be your database engine
- To deploy react app on google cloud in App Engine
- Build, deploy, debug, and monitor Node.js applications in App Engine
- Deploy Flask on Google App Engine
- Run Go App Server on Google App Engine
- Run NGINX on Google Compute Engine
- Persist your data from google firestore
- Use google cloud store to store upload files
- For monitoring the app use google cloud operation suite
- For Prophet and StatModel we are using Big Query and AI Platform
- For NLP sentiment we are using cloud natural language

- Build Postgres image on container

Requirement 13:
Compare and contrast your design considering the following options:
1. Use the personal development computer native operating system (OS) to build, deploy, and run loosely-coupled programs (microservices)
2. Use Docker build, deploy, and run containers for the difference microservices on the development computer native operating system (OS)
3. Use Google Cloud to build, deploy, and run the needed microservices

I have implemented the project with option 1 because I am using PostgreSQL on my local workstation and have also installed Go server. I am building business logic to fetch data from the Chicago data portal into my system and perform analysis on the local environment. Physically, I'm utilizing my own operating system and hardware for this. If changes to one system's design, implementation, or behavior do not induce changes in the other, the systems are loosely linked. Coupling can occur in microservices when a change to one microservice triggers an almost immediate change to all other microservices that work with it directly or indirectly.
Kubernetes, an open source platform for automating the deployment, scaling, and management of containerized applications, is used to deploy cloud-native apps. Kubernetes, which was created at Google, has become the operating system for launching cloud-native apps.
When a microservice requires persistent data, such as files, GCS plays a crucial role in a Google microservices architecture. The service can stay stateless by storing the files outside of the microservice's own file system. This means that while retrieving and putting files to GCS, multiple instances of the same microservice can run simultaneously. Google Cloud Storage is a wonderful solution for enabling scalability and decoupling without much technical work because of its near-infinite scalability, low operational overhead, and low cost per GB.
However, it should be emphasized that this strategy has flaws of its own. The only way to interface with GCS is through an API, CLI, or SDK, so there is no way to mount a server.