Tony Nguyen

Dr. Shawn Bowers

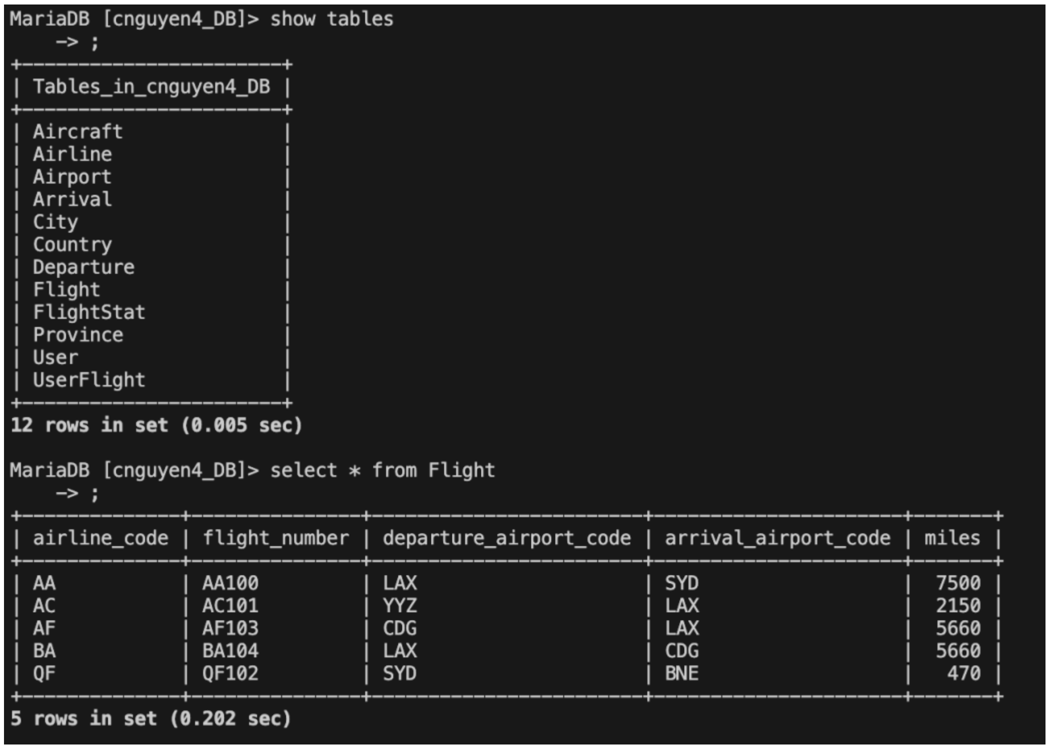
CPSC 321

5 December 2023

Project Part 4

1. I have been working on completing the schema design. Based on the original version, several changes have been made:

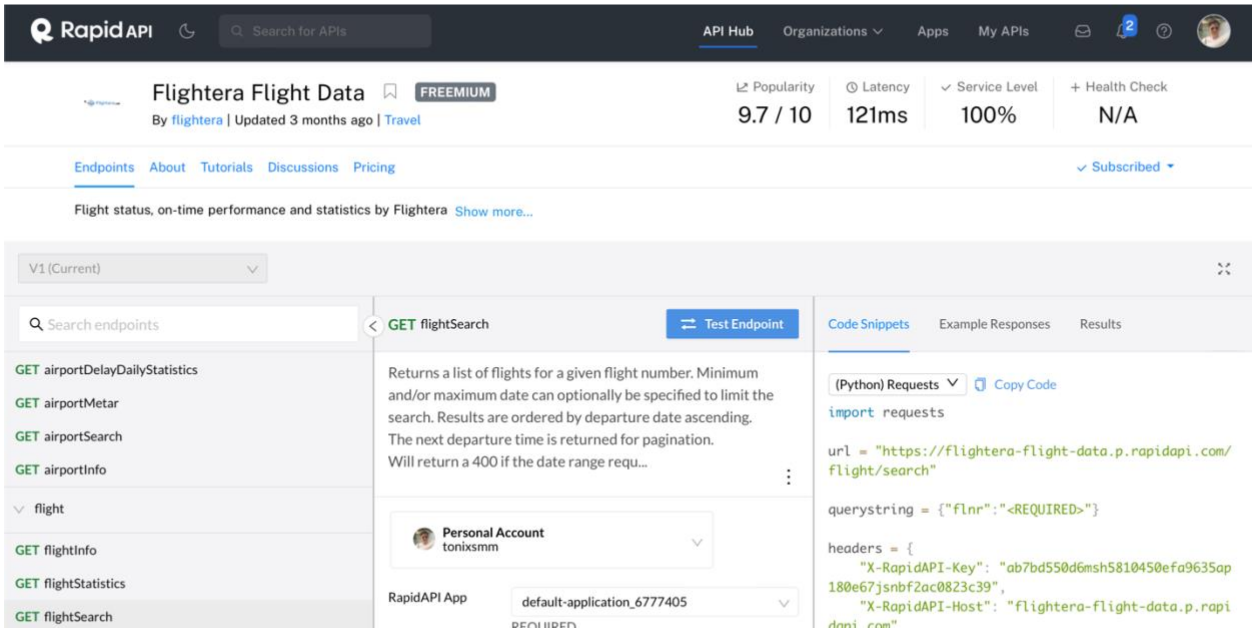
- Removing flight statistic schemas: Instead of creating a table to save them, I will use SQL aggregation functions to calculate them using SQL's VIEW dynamically.
- Adding foreign keys to complete the relationships based on the ER diagram. I also made some changes to the schemas to ensure their normalization constraints.
- Populating some data to test the schema design.



```
MariaDB [cnguyen4_DB]> show tables
    -> ;
+-----------------------+
| Tables_in_cnguyen4_DB |
+-----------------------+
| Aircraft              |
| Airline               |
| Airport               |
| Arrival               |
| City                  |
| Country               |
| Departure             |
| Flight                |
| FlightStat            |
| Province              |
| User                  |
| UserFlight            |
+-----------------------+
12 rows in set (0.005 sec)

MariaDB [cnguyen4_DB]> select * from Flight
    -> ;
+--------------+---------------+------------------------+----------------------+-------+
| airline_code | flight_number | departure_airport_code | arrival_airport_code | miles |
+--------------+---------------+------------------------+----------------------+-------+
| AA           | AA100         | LAX                    | SYD                  |  7500 |
| AC           | AC101         | YYZ                    | LAX                  |  2150 |
| AF           | AF103         | CDG                    | LAX                  |  5660 |
| BA           | BA104         | LAX                    | CDG                  |  5660 |
| QF           | QF102         | SYD                    | BNE                  |   470 |
+--------------+---------------+------------------------+----------------------+-------+
5 rows in set (0.202 sec)
```

2. On the backend side, I have been working on the API endpoints to support the features. I planned to use FlightRadar24 API at first, but then I realized they only offered it for enterprise use. With that, I looked for substitute options and found Flightera Flight Data as a perfect substitute. Since Flightera has a limit for the number of queries allowed to be made under the free tier, I will only use it when needed, meaning I will pre-populate the database with some dummy data to test the features.

- I am setting up a Flask app that runs on Python to handle the API endpoints.
- I also plan to complete the query functions to connect to the database and return the results to the front end within this week.
- I set up a Heroku app to host the backend API. I will update the API endpoints once I complete them.



Flightera Flight Data FREEMIUM

By flightera | Updated 3 months ago | Travel

Popularity 9.7 / 10 | Latency 121ms | Service Level 100% | Health Check N/A

Flight status, on-time performance and statistics by Flightera Show more...

GET airportDelayDailyStatistics
GET airportMetar
GET airportSearch
GET airportInfo
flight
GET flightInfo
GET flightStatistics
GET flightSearch

GET flightSearch

Returns a list of flights for a given flight number. Minimum and/or maximum date can optionally be specified to limit the search. Results are ordered by departure date ascending. The next departure time is returned for pagination. Will return a 400 if the date range requ...

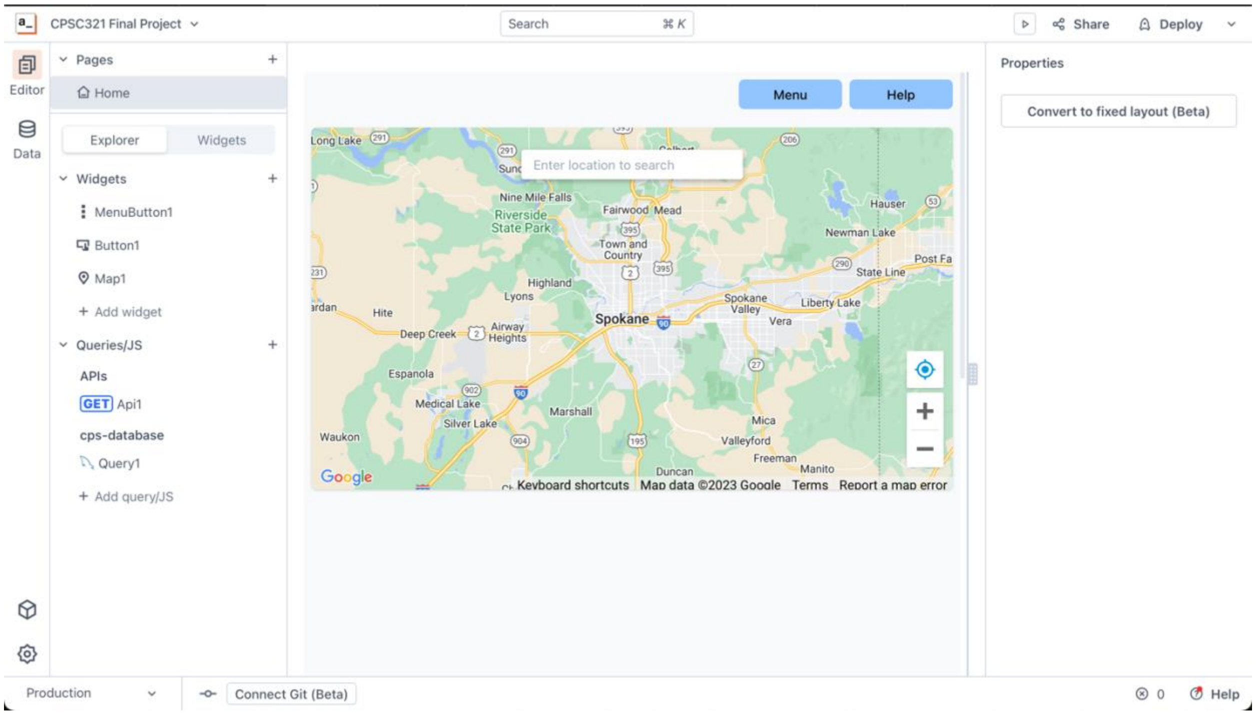
Personal Account
tonixsmm

RapidAPI App: default-application_6777405

```
import requests

url = "https://flightera-flight-data.p.rapidapi.com/
flight/search"

querystring = {"flnr":"<REQUIRED>"}

headers = {
    "X-RapidAPI-Key": "ab7bd550d6msh5810450efa9635ap
180e67jsnbf2ac0823c39",
    "X-RapidAPI-Host": "flightera-flight-data.p.rapi
dapi.com"
```

3. On the frontend side, I have been working on Appsmith to create the initial UI for the app. I have run into some issues with connecting it to the backend API, but I will try to resolve them within this week.

- I set up a Docker container to deploy the app. It is my first time setting up a Dockerfile from scratch, so I am still learning how to use it. I will update the Dockerfile once I complete it.


CPSC321 Final Project
Search
Share
Deploy
Editor
Data
Pages
Home
Explorer
Widgets
MenuButton1
Button1
Map1
Add widget
Queries/JS
APIs
GET Api1
cps-database
Query1
Add query/JS
Menu
Help
Enter location to search
Spokane
Properties
Convert to fixed layout (Beta)
Production
Connect Git (Beta)
Help