Data Pipeline Documentation: Gans Scooter company data collection

Dear future Gans colleague,

we are Gans Project Group 5 and have prepared the “Gans Scooter company data collection pipeline” for you. This documentation is meant for you to ease in quickly when taking over the project. For a quick overview we have enclosed diagrams of the data pipeline and the schema. Input and Output points of the pipeline are just one click away.

We as a team made great efforts to review the pipeline’s code thoroughly and to design the best possible data flow. The output of this pipeline is a database which will update every night. It will allow you to query the data you need for your decision making process concerning the placement and movement of e-scooters in various European cities with airports.

To sum up what we did: We collected data about various European Cities regarding weather and incoming flights at their respective airports. The output of the pipeline is a relational mysql data base hosted in the AWS RDS cloud which automatically requests updated data from its API data sources.

Abstract

Who requested the data base?: Ali El-Kassas, Guillem Perdigò data science instructors at WBS Coding School

Purpose: Data engineering group work assignment

Task: Collect weather and flight data to strategically place and move e-scooters in various cities with airports.

Data pipeline name: Gans Scooter company data collection

Owner: Gans project Team 5 of DS#006 - Ali WBS Coding School

Used since: 15 june 2022

Input data:

- The whole document for API data input: https://colab.research.google.com/drive/1vvCGEVI5-X4sakoBbDcDbKhUEmbxHfaD?usp=sharing

1) Cities: https://colab.research.google.com/drive/1vvCGEVI5-X4sakoBbDcDbKhUEmbxHfaD#scrollTo=twrTKJlCAVjI&line=26&uniqifier=1
2) Weather: https://colab.research.google.com/drive/1vvCGEVI5-X4sakoBbDcDbKhUEmbxHfaD#scrollTo=HZCKdEDD3hei&line=21&uniqifier=1
3) Airports: https://colab.research.google.com/drive/1vvCGEVI5-X4sakoBbDcDbKhUEmbxHfaD#scrollTo=xJW2uuDMWmGv&line=1&uniqifier=1
4) Flights: https://colab.research.google.com/drive/1vvCGEVI5-X4sakoBbDcDbKhUEmbxHfaD#scrollTo=SDCMvYNpZsMN&line=5&uniqifier=1

Output at the end of the pipeline: gans-database.cl1f05rvfink.us-east-1.rds.amazonaws.com

Runtime environment:
https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#database:id=gans-database;is-cluster=false

Short Process Recap:

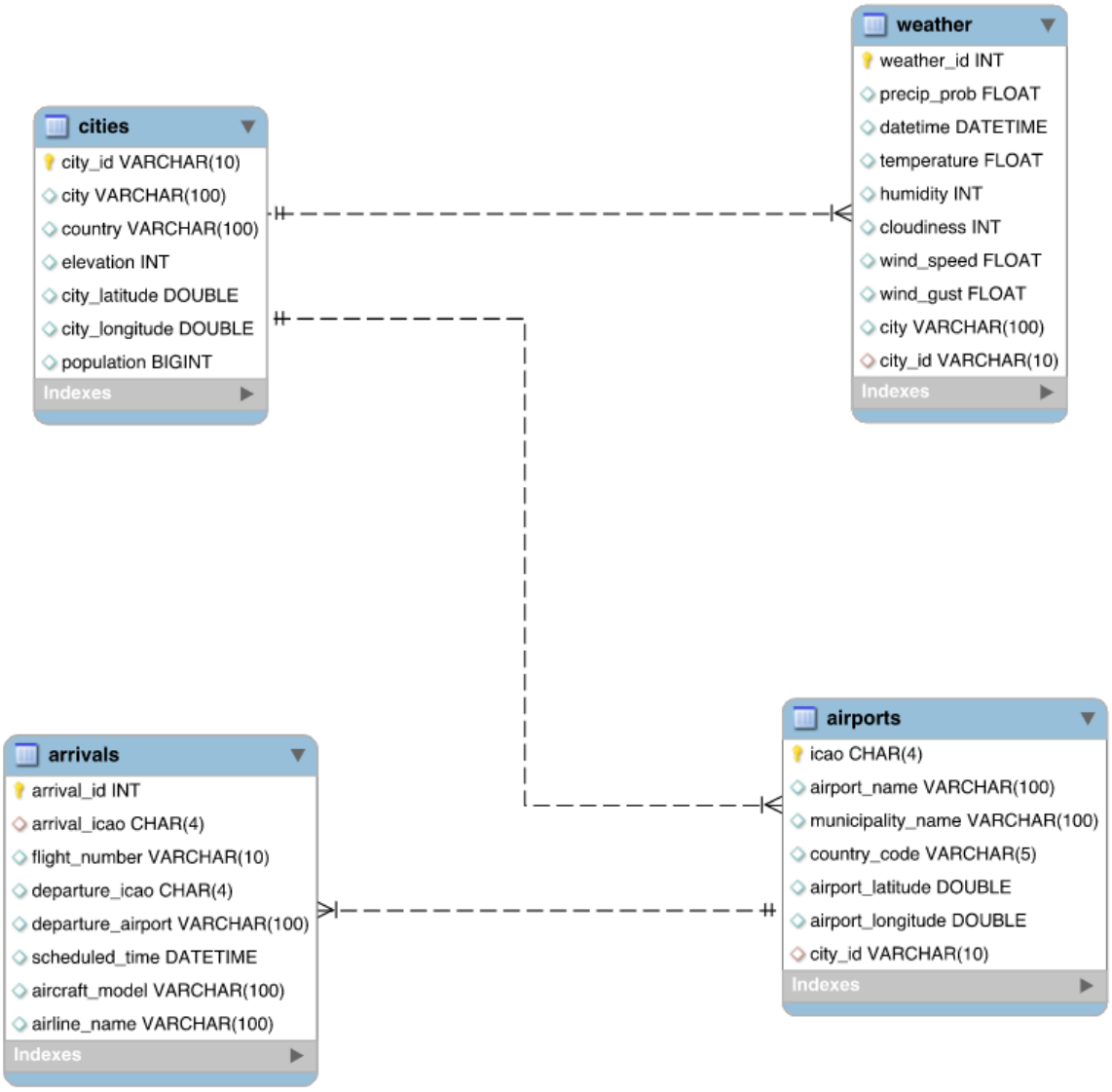
We collected data about weather in and flights to various European cities. For the sake of consistent data and code quality we opted to solely use data from APIs and to refrain from web scraping. We used Rapid API and OpenWeather as sources. Here are links to the input data within our python script. You can also find these links in the abstract of this documentation.

- Cities: https://colab.research.google.com/drive/1vvCGEVI5-X4sakoBbDcDbKhUEmbxHfaD#scrollTo=twrTKJICAVjI&line=26&uniqifier=1
- Weather: https://colab.research.google.com/drive/1vvCGEVI5-X4sakoBbDcDbKhUEmbxHfaD#scrollTo=HZCKdEDD3hei&line=21&uniqifier=1
- Airports: https://colab.research.google.com/drive/1vvCGEVI5-X4sakoBbDcDbKhUEmbxHfaD#scrollTo=xJW2uuDMWmGv&line=1&uniqifier=1
- Flights: https://colab.research.google.com/drive/1vvCGEVI5-X4sakoBbDcDbKhUEmbxHfaD#scrollTo=SDCMvYNpZsMN&line=5&uniqifier=1

Once we collected the necessary data for the pipeline we created four new tables within our python script. Here are the links to the new tables:

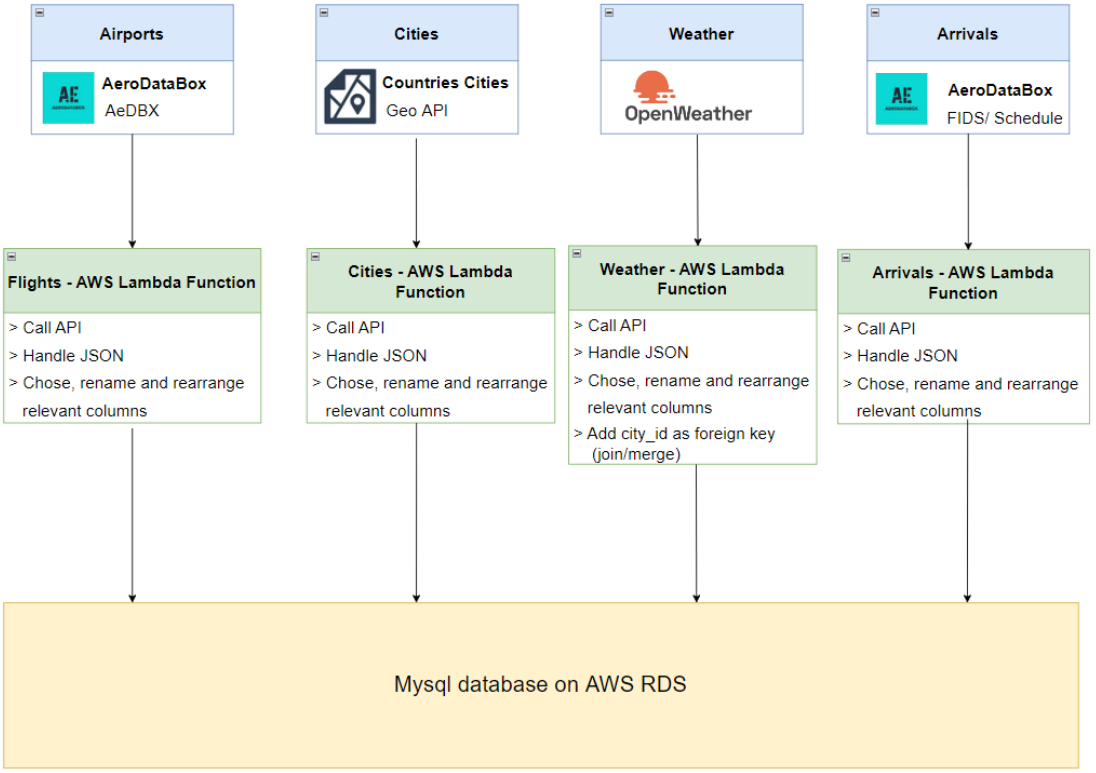
- Cities: https://colab.research.google.com/drive/1vvCGEVI5-X4sakoBbDcDbKhUEmbxHfaD#scrollTo=q41xnU8V-dSr&line=4&uniqifier=1
- Weather: https://colab.research.google.com/drive/1vvCGEVI5-X4sakoBbDcDbKhUEmbxHfaD#scrollTo=9YwFgjDsZROe&line=1&uniqifier=1
- Airports: https://colab.research.google.com/drive/1vvCGEVI5-X4sakoBbDcDbKhUEmbxHfaD#scrollTo=pXeic2qe5K5u&line=4&uniqifier=1
- Flights: https://colab.research.google.com/drive/1vvCGEVI5-X4sakoBbDcDbKhUEmbxHfaD#scrollTo=V0tHPvXLuT2n&line=1&uniqifier=1

Once the tables were designed we created a locally hosted, relational database for you to query. Below you can find the respective schema.



We assumed that you would like the data to update automatically while not having your computer run around the clock. Therefore we moved the local database into Amazons' AWS RDS cloud. We did this by creating Lambda functions and giving these functions access to the AWS RDS cloud. The function connects the database to its API sources and replicates the relationship between the tables as defined in the schema. Within AWS we also set up a trigger via eventbridge to update the API data on weather and flights every day at 11:59 pm.

This is a visual overview of the data pipeline. Remember the relationship between the tables is depicted in the schema above:



And here is the link to the final database which unfortunately will not work once you click it. We had to shut down the database to avoid costs while your position is still vaccant:

Gans-database.cl1f05rvfink.us-east-1.rds.amazonaws.com