Big Data Engineering

Assignment 3: Building ELT data pipelines with Airflow

Aim:

The aim of this assignment is to build production-ready ELT data pipelines using Apache Airflow and dbt Cloud. You will process and transform Airbnb and Census data for Sydney, load it into a data warehouse following a medallion architecture (Bronze, Silver, Gold), and create a data mart for analytical insights. The assignment also includes performing ad-hoc analyses to address key business questions.

Introduction to the datasets

1. Airbnb

Airbnb is an online-based marketing company that connects people looking for accommodation (Airbnb guests) to people looking to rent their properties (Airbnb hosts) on a short-term or long-term basis. The rental properties include apartments (dominant), homes, boats, and a whole lot more. As of 2019, there are 150 million users of Airbnb services in 191 countries, making it a major disruptor of the traditional hospitality industry (this is akin to how Uber and other emerging transportation services have disrupted the traditional intra-city transportation services). As a rental ecosystem, Airbnb generates tons of data including but not limited to: density of rentals across regions (cities and neighbourhoods), price variations across rentals, host-guest interactions in the form of reviews, and so forth.

We will focus on Sydney for this assignment, you can find the original data and more information on this link, however the website is purging the available data regularly.

The modified dataset used in this assignment is from May 2020 to April 2021.

2. Census

The Census of Population and Housing (Census) is Australia's largest statistical collection undertaken by the Australian Bureau of Statistics (ABS). For more than 100 years, the Census has provided a snapshot of Australia, showing how the country has changed over time, allowing it to plan for the future. The aim of the Census is to accurately collect data on the key characteristics of people in Australia on Census night and the dwellings in which they live. In 2016, the Census counted close to 10 million dwellings and approximately 24 million people, the largest number counted to date.

The information provided in the Census helps estimate Australia's population, which is used to distribute government funds and plan services for the community – housing, transport, education, industry, hospitals and the environment. Census data is also used by individuals and organisations in the public and private sectors to make informed decisions on policy and planning issues that impact the lives of all Australians.

You can find the original dataset and find more information on this link.

Tasks:

You will have to set up an Airflow and Postgres environment using GCP (Cloud Composer and SQL instance) and dbt Cloud.

Part 0: Download the datasets:

a. 12 months of Airbnb listing data for Sydney: link

b. The tables G01 ("Selected Person Characteristics by Sex") and G02 ("Selected Medians and Averages") of the General Community Profile Pack from the 2016 census at the LGA level: link.

c. A dataset to help you join both datasets based on LGAs code and a mapping between LGAs and Suburbs: link.

Part 1: Use Airflow to load the initial raw data into Postgres

1. **Upload the Dataset**: Upload the first month of Airbnb data (`05_2020.csv`) + the census dataset and the LGAs mapping into the Airflow storage bucket.

2. Using DBeaver, set up a Bronze schema in your Postgres instance and create the necessary raw tables to store the initial data.

3. Build an Airflow DAG with no set schedule interval (`schedule_interval=None`) that reads the data from the storage bucket and loads it into the raw tables within the Bronze schema on Postgres.

Part 2: Design a data warehouse with dbt

1. Create a data warehouse architecture on Postgres using the Medallion architecture (Bronze, Silver, Gold) with dbt. Include at least 4 dimension tables (e.g., listing, host, suburb, LGA, etc.) along with two Census tables as reference data in the Gold layer. The layers are defined as follows:

   a. Bronze: Stores the raw tables loaded from Airflow and any additional tables derived from this raw data, particularly focusing on the Airbnb dataset.

   b. Silver: Contains cleaned and transformed versions of the Bronze tables with consistent naming conventions. This layer includes snapshots for your dimensions using a timestamp strategy, addressing any issues with listing dates and LGAs.

   c. Gold:
      i. Implements a star schema consisting of dimension and fact tables, where fact tables contain only IDs and metrics (e.g., price).
      ii. Datamart : This is where the answers to the key business questions will be stored. It should be materialised as views created from the fact and dimension tables, taking into account Slowly Changing Dimensions Type 2 (SCD2).

2. **For the datamart, create the 3 following views:**

a. **dm_listing_neighbourhood**:

This view should provide insights per `listing_neighbourhood` and `month/year` with the following metrics:

- Active listings rate
- Minimum, maximum, median and average price for active listings
- Number of distinct hosts
- Superhost rate
- Average of review_scores_rating for active listings
- Percentage change for active listings
- Percentage change for inactive listings
- Total Number of stays
- Average Estimated revenue per active listings

The view should be ordered by `listing_neighbourhood` and `month/year`.

b. **dm_property_type**:

This view should present information per `property_type`, `room_type`, `accommodates`, and `month/year` including:

- Active listings rate
- Minimum, maximum, median and average price for active listings
- Number of distinct hosts
- Superhost rate
- Average of review_scores_rating for active listings
- Percentage change for active listings
- Percentage change for inactive listings
- Total Number of stays
- Average Estimated revenue per active listings

The view should be ordered by `property_type`, `room_type`, `accommodates`, and `month/year`.

c. Dm_host_neighbourhood

This view provides data per `host_neighbourhood_lga` (derived from transforming `host_neighbourhood` to the corresponding LGA) and `month/year` with the following metrics:

- Number of distinct host
- Estimated Revenue
- Estimated Revenue per host (distinct)

The view should be ordered by `host_neighbourhood_lga` and `month/year`.

Definitions:

- Active listings: Listings where "has_availability" = "t".
- Active Listing Rate = (total Active listings / total listing) * 100
- Superhost Rate = (total distinct hosts with "host_is_superhost" = 't' / total distinct hosts) * 100
- Percentage change (month to month) = ((final value - original value) / original value) * 100
- Number of stays (only for active listings) = 30 - availability_30
- Estimated revenue per active listings = for each active listing per period: number of stays * price
- Estimated revenue per host= Total Estimated revenue per active listings/ total distinct hosts

Advises:
Truncate all tables before running your dag for the first time.
Be careful of the order of operation, especially when loading dimension and fact data.
Be careful with SCD in the facts table (use the snapshots models).
Use “dbt run --full-refresh” if you want to recreate every model.

Part 3: End to end orchestration

1. **Modify the Existing Airflow DAG:**
   Update your Airflow DAG to include a task that triggers a dbt job. This will run your dbt models, transforming the data through the layers of the data warehouse.

2. **Load the Remaining Airbnb Data:**
   Extend your DAG to load the remaining Airbnb datasets month by month in chronological order. Ensure that each month’s data is processed sequentially to maintain the correct order and data integrity throughout the pipeline.

Part 4: Ad-hoc analysis

Answer the following questions with supporting results (write SQL on Postgres):

a. What are the demographic differences (e.g., age group distribution, household size) between the top 3 performing and lowest 3 performing LGAs based on estimated revenue per active listing over the last 12 months?

b. Is there a correlation between the median age of a neighbourhood (from Census data) and the revenue generated per active listing in that neighbourhood?

c. What will be the best type of listing (property type, room type and accommodates for) for the top 5 “listing_neighbourhood” (in terms of estimated revenue per active listing) to have the highest number of stays?

d. For hosts with multiple listings, are their properties concentrated within the same LGA, or are they distributed across different LGAs?

e. For hosts with a single Airbnb listing, does the estimated revenue over the last 12 months cover the annualised median mortgage repayment in the corresponding LGA? Which LGA has the highest percentage of hosts that can cover it?

Add the answers with screenshots in your report.

This is an individual assignment, each student will be marked individually.

Deliverables:

Each student will have to submit

1. **SQL Queries for Part 1**:
   a. Submit a single .sql file containing all SQL queries used for Part 1 in Postgres.
   b. File name: part_1.sql
2. **Airflow DAG Script for Parts 1 and 3**:
   a. Provide the complete Airflow DAG script that combines Parts 1 and 3 into a single DAG.
   b. **File**: One combined Airflow DAG file.
3. **SQL Queries for Part 4**:
   a. Submit a single .sql file containing all SQL queries used for Part 4 in Postgres.
   b. File name: part_4.sql
4. **dbt Cloud Files**:
   a. Include all the .sql files from your dbt Cloud repository for the following folders:
      i. **Models**
      ii. **Snapshots**
   b. Include the **dbt_project.yml** file from your dbt Cloud repository.
5. **Handover Report**:
6. **Additional Relevant Files**:

The report should not exceed 3000 words (figures and tables are not counted).

Compress all deliverables into a single zip file and use the following file naming format for the submission:
Assignment_3_FirstName_LastName.zip

A good "handover" report should contained:
1. High-level view of your project.
2. Explanation for the different steps of your project.
3. Any issues/bugs you faced and how you solved them.
4. Answers to the business questions with supporting evidence.
5. Relevant screenshots/images/diagrams/flows if necessary.

You can assume that the reader of your report will have a similar understanding and knowledge of any technical skills (Python/SQL).

Assessment Criteria:

- Quality of code (Python/SQL/dbt).
- Justification of data transformation, data formats, data storage, DAGs structure and accuracy of results with evidence supporting claims
- Quality of findings and recommendations for business questions.
- Clarity and quality of written report.

Criteria Details and weights:

Criteria	Further Details
Quality of code (Python/SQL/dbt)	1. Code can be executed without raising an error and is well commented 2. Data Warehouse is correctly set up (layers, fact, dimension, SCD).
Justification of data transformation, data formats, data storage, DAGs structure, accuracy of results with evidence supporting claims.	1. High level explanation of each major step and decision. 2. Follows the good “handover” report guidelines
Quality of findings and recommendations for business questions.	1. Correct answers to the business questions. 2. Recommendations to the business are relevant.
Clarity and quality of written report.	1. Complete and professionally formatted report (spelling, grammar, punctuation, layout). 2. Report is not exceeding the maximum length

Due Date:

All assignments need to be submitted before the **due date (28th October)** on Canvas.
Penalties will be applied for late submission

Late submission will be penalised 10 pts per day after the due date.