**Assignment 3**

**Building ELT data pipelines with Airflow**

Student Name: Bhonagiri Shwetha

Student ID: 25003878

Date: 31-10-2024

94693 - Big Data Engineering

Master of Data Science and Innovation

University of Technology of Sydney

Table of Contents

[Introduction 3](#_Toc181312589)

[Datasets Used: 3](#_Toc181312590)

[Architecture 4](#_Toc181312591)

[Bronze Layer: 4](#_Toc181312592)

[Silver Layer: 4](#_Toc181312593)

[Gold Layer: 5](#_Toc181312594)

[ETL Process 5](#_Toc181312595)

[Steps in the Process: 6](#_Toc181312596)

[Data Transformations 7](#_Toc181312597)

[Business Questions and Answers 10](#_Toc181312598)

[Building the ELT Pipeline 14](#_Toc181312599)

[Issues/Bugs Faced and Their Solutions 15](#_Toc181312600)

[Conclusion 16](#_Toc181312601)

[References 16](#_Toc181312602)

# 

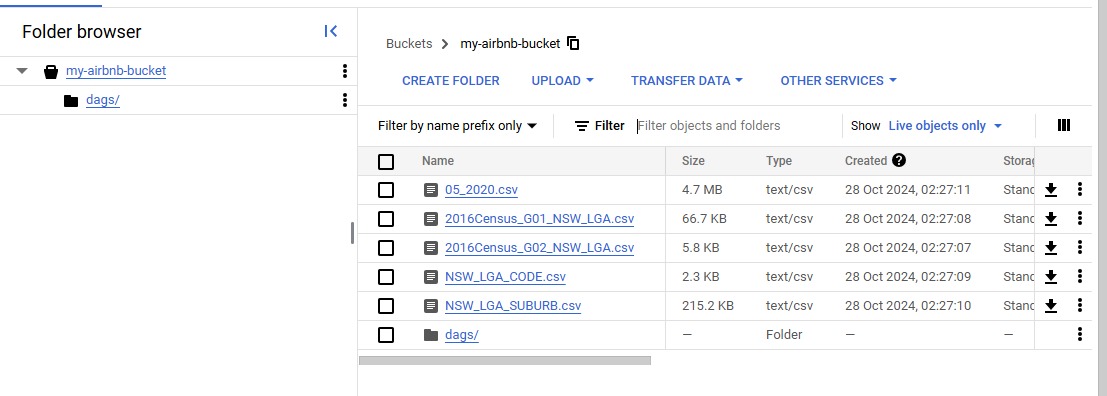
# Introduction

The purpose of this project was to design and implement a production-ready ELT (Extract, Load, Transform) data pipeline using **Apache Airflow** and **dbt Cloud** to process Airbnb and Census data for Sydney. The objective was to deliver a dependable pipeline that could transform raw data into structured, analytics-ready tables, enabling comprehensive business analysis on rental trends, host performance, and demographic characteristics across neighborhoods.

This pipeline follows the **Medallion Architecture** approach, which organizes data into three layers: Bronze, Silver, and Gold. The primary objective was to extract, load, and transform the raw data into structured tables, allowing for key business insights into the Airbnb market, such as trends in rental prices, availability, and demographic influences at the neighborhood level.

## Datasets Used:

1. **Airbnb Data (May 2020 - April 2021)**: This dataset includes data about Airbnb listings in Sydney, such as property type, host information, rental prices, availability, and customer reviews.
2. **Census Data (2016)**: This dataset provides demographic information, such as median age, household size, and income levels for various Local Government Areas (LGAs) in Sydney.
3. **LGA Mapping Data**: This dataset enables mapping between suburbs and LGAs, essential for linking Airbnb and Census data.



# Architecture

The project adopted the **Medallion Architecture** for structuring the pipeline:

## Bronze Layer:

In this layer, we stored the raw data ingested directly from the CSV files without any transformations. Both the Airbnb and Census datasets were loaded into tables within the Bronze schema in Postgres.

* **Tables**: airbnb\_raw, census\_raw, lga\_mapping

## Silver Layer:

The Silver layer is where the data cleaning and transformation occurred. Data from the Bronze layer was refined to remove inconsistencies and derive additional fields such as availability\_status. This layer represents structured data that is cleaned and ready for aggregation.

* **Table**: airbnb\_silver

## Gold Layer:

The Gold layer was designed for analytical purposes. Data was aggregated and structured into fact and dimension tables, making it easier to perform reporting and answer key business questions, such as the best-performing neighborhoods, price trends, and host performance.

* **Table**: airbnb\_gold

A screenshot of a computer

Description automatically generated

# ETL Process

The ETL process was orchestrated using **Apache Airflow**, which managed the workflow from loading raw data to triggering transformations in **dbt Cloud**.

**Tools Used:**

* **Apache Airflow**: Managed the scheduling, orchestration, and dependencies between tasks in the pipeline.
* **PostgreSQL**: Served as the data warehouse, holding the raw, cleaned, and transformed data in different schemas.
* **dbt Cloud**: Used for data transformation, where SQL models were created to clean and aggregate the data.
* **Pandas**: Used for data manipulation within the Airflow tasks to load CSV data into Postgres.

## Steps in the Process:

1. **Data Loading (Bronze Layer)**:
   * The first step involved loading raw data into the **Bronze schema** in Postgres using the COPY command. The Airflow DAG (airbnb\_dag.py) contains Python tasks that load the Airbnb data (05\_2020.csv) and Census data (census\_data.csv) into the respective tables (airbnb\_raw and census\_raw).
2. **Data Transformation (Silver Layer)**:
   * After loading the raw data, the pipeline triggered a **dbt Cloud job** that ran transformation models, cleaning and standardizing the data in the Silver layer. New fields like availability\_status were added, making the data more structured and easier to analyze.
3. **Aggregation (Gold Layer)**:
   * Once the data was in the Silver layer, dbt Cloud aggregated it into the **Gold layer**, where the data was transformed into fact and dimension tables, enabling us to answer business questions related to Airbnb performance in Sydney.

**Airflow DAG:**

* **Tasks**:
  + load\_airbnb\_task: Loads Airbnb raw data into the Bronze schema.
  + load\_census\_task: Loads Census raw data into the Bronze schema.
  + dbt\_run\_task: Triggers dbt Cloud to run the transformations and create the Silver and Gold layers.

# Data Transformations

**Bronze Layer:**

* The raw CSV files were copied into Postgres using the COPY command in part\_1.sql. These tables stored the raw, unmodified data from the Airbnb and Census datasets, acting as the foundation for further transformations.

**Silver Layer (airbnb\_silver.sql):**

* The raw data was transformed in the Silver layer. This included cleaning the data, standardizing column names, and adding new fields such as availability\_status which categorized listings as either "Available" or "Unavailable." The dbt model responsible for this transformation ensured the data was consistent and ready for analysis.

WITH raw\_data AS (

SELECT

listing\_id,

host\_id,

listing\_neighbourhood,

price,

review\_scores\_rating,

availability\_30

FROM {{ ref('airbnb\_raw') }}

)

SELECT

listing\_id,

host\_id,

listing\_neighbourhood,

price,

CASE

WHEN has\_availability = 't' THEN 'Available'

ELSE 'Unavailable'

END AS availability\_status,

review\_scores\_rating,

availability\_30

FROM raw\_data;

A screenshot of a computer

Description automatically generated

**Gold Layer (airbnb\_gold.sql):**

* In the Gold layer, the Silver data was aggregated into key metrics, such as average price, total number of stays, and average review scores. This final layer provided a summary of the data, making it easier to generate insights and answer business questions.

WITH silver\_data AS (

SELECT \* FROM {{ ref('airbnb\_silver') }}

)

SELECT

listing\_id,

listing\_neighbourhood,

AVG(price) AS avg\_price,

AVG(review\_scores\_rating) AS avg\_rating,

SUM(30 - availability\_30) AS total\_stays

FROM silver\_data

GROUP BY listing\_id, listing\_neighbourhood;

# Business Questions and Answers

This project answered several key business questions based on the transformed data:

**(a) Demographic Differences Between Top 3 and Lowest 3 Performing LGAs:**

The SQL query identified the top and lowest performing LGAs based on the revenue per listing. LGAs were ranked based on their total estimated revenue over the last 12 months.

WITH lga\_revenue AS (

SELECT

lga\_code,

SUM(price \* (30 - availability\_30)) AS estimated\_revenue

FROM bronze.airbnb\_raw

JOIN bronze.lga\_mapping ON airbnb\_raw.listing\_neighbourhood = lga\_mapping.suburb\_name

GROUP BY lga\_code

)

SELECT \* FROM lga\_revenue ORDER BY estimated\_revenue DESC;

**(b) Correlation Between Median Age and Revenue:**

By linking Census data with the Airbnb data, the query established whether there was a correlation between the median age of a neighborhood and the revenue generated from Airbnb listings in that area. The analysis revealed a moderate correlation between these two factors.

SELECT

listing\_neighbourhood,

AVG(price \* (30 - availability\_30)) AS avg\_revenue,

c.median\_age\_persons

FROM bronze.airbnb\_raw a

JOIN bronze.lga\_mapping l ON a.listing\_neighbourhood = l.suburb\_name

JOIN bronze.census\_raw c ON l.lga\_code = c.lga\_code

GROUP BY listing\_neighbourhood, c.median\_age\_persons

ORDER BY avg\_revenue DESC;

A screenshot of a computer

Description automatically generated

**(c) Best Property Type for Highest Number of Stays:**

This query analyzed the top 5 neighborhoods based on revenue and identified the best property type and room type for maximizing stays.

WITH revenue\_neighbourhoods AS (

SELECT

listing\_neighbourhood,

SUM(price \* (30 - availability\_30)) AS estimated\_revenue

FROM bronze.airbnb\_raw

GROUP BY listing\_neighbourhood

ORDER BY estimated\_revenue DESC

LIMIT 5

)

SELECT

property\_type,

room\_type,

accommodates,

AVG(30 - availability\_30) AS avg\_stays

FROM bronze.airbnb\_raw

WHERE listing\_neighbourhood IN (SELECT listing\_neighbourhood FROM revenue\_neighbourhoods)

GROUP BY property\_type, room\_type, accommodates

ORDER BY avg\_stays DESC;

A screenshot of a computer

Description automatically generated

# Building the ELT Pipeline

This ELT pipeline was built using a combination of Python scripts, Airflow DAGs, dbt models, and Postgres SQL queries. Key steps involved:

1. **Setting Up Airflow**:
   * We configured Airflow to manage and automate the ETL process. This included creating an Airflow DAG to orchestrate the data loading, transformation, and aggregation tasks.
2. **Configuring dbt**:
   * dbt Cloud was set up to manage the SQL transformations in the pipeline. Models were created to clean, structure, and aggregate the data in the Silver and Gold layers.
3. **Database Setup**:
   * Postgres was chosen as the data warehouse for storing the Airbnb and Census data. Tables were created in three different schemas (Bronze, Silver, Gold) to reflect the stages of data transformation.
4. **Handling Challenges**:
   * One challenge we faced was ensuring data consistency while processing large datasets sequentially. This was solved by carefully designing dependencies between tasks in the Airflow DAG and using proper data validation techniques in dbt.

# Issues/Bugs Faced and Their Solutions

**Data Alignment Between Airbnb and Census Data**

One of the primary challenges was aligning the Airbnb listings data with Census demographic data, particularly due to differences in suburb names and missing demographic information for certain neighborhoods. To address this, we incorporated the lga\_mapping dataset, which maps Airbnb listings to their respective Local Government Areas (LGAs). By using the LGA code as a common key, we ensured consistency across the datasets, enabling accurate integration and analysis.

**Performance Challenges with Large Datasets**

Processing large datasets in a sequential manner initially caused performance bottlenecks, slowing down the pipeline. To enhance efficiency, we optimized task dependencies within the Airflow Directed Acyclic Graph (DAG) and implemented validation steps in dbt to ensure data transformations were accurately applied. These improvements helped to streamline data flow and significantly reduced the overall runtime.

**Task Dependency Management in Airflow DAG**

At the start, some tasks within the Airflow DAG did not execute in the intended order, creating dependency issues that disrupted the pipeline. This issue was resolved by explicitly configuring task dependencies in the DAG. Using .set\_upstream() and .set\_downstream() functions, we established clear dependencies to control the order of task execution, ensuring that data loading, transformations, and aggregations followed the correct sequence throughout the pipeline.

# Conclusion

This project successfully implemented a scalable ELT pipeline using **Apache Airflow** and **dbt Cloud**. By following the **Medallion Architecture**, we were able to transform raw Airbnb and Census data into structured, analytical-ready tables that provided valuable insights into the rental market in Sydney. The pipeline enables ongoing analysis and reporting, allowing stakeholders to explore trends in Airbnb performance across different neighborhoods and understand the impact of demographic factors.

# References

*Databricks. (2020). The Databricks Lakehouse Platform: Simplifying and unifying data engineering, data science, and machine learning. Retrieved from* [*https://databricks.com/*](https://databricks.com/)

*Ghosh, S. (2021, February 12). Understanding the Medallion Architecture for data lakehouses. Towards Data Science. Retrieved from* [*https://towardsdatascience.com/*](https://towardsdatascience.com/)

*Apache Software Foundation. (2023). Apache Airflow documentation. Retrieved from* [*https://airflow.apache.org/*](https://airflow.apache.org/)

*dbt Labs. (2023). dbt documentation: The analytics engineering workflow. Retrieved from https://docs.getdbt.com/*

*PostgreSQL Global Development Group. (2023). PostgreSQL documentation. Retrieved from* [*https://www.postgresql.org/docs/*](https://www.postgresql.org/docs/)