Assignment no.3

Student Name: Shaqran Bin Saleh

Student ID: 25010238

94693 Big Data Engineering

TD School

University of Technology Sydney

Table of Contents

Introduction

This report presents a comprehensive approach to building and managing an Extract, Load, and Transform (ELT) pipeline utilizing Apache Airflow and dbt Cloud for Airbnb and Census data related to Sydney. The primary objective of this assignment is to create a production-ready data pipeline that supports analytical insights by implementing the Medallion architecture—Bronze, Silver, and Gold layers. The pipeline facilitates the integration of Airbnb listing data with demographic Census data, enabling the generation of a data mart designed to answer specific business questions.

The datasets for this project include 12 months of Airbnb data, which provides information on property types, pricing, availability, and host details, and Census data that captures demographic attributes at the Local Government Area (LGA) level. Through Airflow, the data is loaded into a PostgreSQL environment, transformed through dbt Cloud into structured layers, and organized for efficient analysis. The Gold layer adopts a star schema design, optimized for querying metrics on listings, property types, and host neighborhoods, among others.

This report outlines the development process, from data ingestion and transformation to the creation of data marts for analytical purposes. It addresses key business questions related to revenue generation, host property distribution, and demographic correlations, providing actionable insights supported by SQL queries and visual evidence. Additionally, the report discusses challenges encountered, solutions implemented, and recommendations based on the findings.

Methodology

The Data

In this assignment we had to deal with a lot of datasets which were all .csv files. Two of them had details of New South Wales's LGAs (Local Government Areas), two of them had census details of the LGAs (Local Government Areas), and there were several files containing information about property listings.

The structure of these files was as follows:

.csv file name	Columns
2016Census_G01_NSW_LGA	LGA_CODE_2016, Tot_P_M, Tot_P_F, Tot_P_P, Age_0_4_yr_M, Age_0_4_yr_F, Age_0_4_yr_P, Age_5_14_yr_M, Age_5_14_yr_F, Age_5_14_yr_P, Age_15_19_yr_M, Age_15_19_yr_F, Age_15_19_yr_P, Age_20_24_yr_M, Age_20_24_yr_F, Age_20_24_yr_P, Age_25_34_yr_M ,Age_25_34_yr_F, Age_25_34_yr_P, Age_35_44_yr_M, Age_35_44_yr_F, Age_35_44_yr_P, Age_45_54_yr_M, Age_45_54_yr_F, Age_45_54_yr_P, Age_55_64_yr_M, Age_55_64_yr_F, Age_55_64_yr_P, Age_65_74_yr_M, Age_65_74_yr_F, Age_65_74_yr_P, Age_75_84_yr_M, Age_75_84_yr_F, Age_75_84_yr_P, Age_85ov_M, Age_85ov_F, Age_85ov_P,

	Counted_Census_Night_home_M, Counted_Census_Night_home_F, Counted_Census_Night_home_P, Count_Census_Nt_Ewhere_Aust_M, Count_Census_Nt_Ewhere_Aust_F, Count_Census_Nt_Ewhere_Aust_P, Indigenous_psns_Aboriginal_M, Indigenous_psns_Aboriginal_F, Indigenous_psns_Aboriginal_P, Indig_psns_Torres_Strait_Is_M, Indig_psns_Torres_Strait_Is_F, Indig_psns_Torres_Strait_Is_P, Indig_Bth_Abor_Torres_St_Is_M, Indig_Bth_Abor_Torres_St_Is_F, Indig_Bth_Abor_Torres_St_Is_P, Indigenous_P_Tot_M, Indigenous_P_Tot_F, Indigenous_P_Tot_P, Birthplace_Australia_M, Birthplace_Australia_F, Birthplace_Australia_P, Birthplace_Elsewhere_M, Birthplace_Elsewhere_F, Birthplace_Elsewhere_P, Lang_spoken_home_Eng_only_M, Lang_spoken_home_Eng_only_F, Lang_spoken_home_Eng_only_P, Lang_spoken_home_Oth_Lang_M, Lang_spoken_home_Oth_Lang_F, Lang_spoken_home_Oth_Lang_P, Australian_citizen_M, Australian_citizen_F, Australian_citizen_P, Age_psns_att_educ_inst_0_4_M, Age_psns_att_educ_inst_0_4_F, Age_psns_att_educ_inst_0_4_P, Age_psns_att_educ_inst_5_14_M, Age_psns_att_educ_inst_5_14_F, Age_psns_att_educ_inst_5_14_P, Age_psns_att_edu_inst_15_19_M, Age_psns_att_edu_inst_15_19_F, Age_psns_att_edu_inst_15_19_P, Age_psns_att_edu_inst_20_24_M, Age_psns_att_edu_inst_20_24_F, Age_psns_att_edu_inst_20_24_P, Age_psns_att_edu_inst_25_ov_M, Age_psns_att_edu_inst_25_ov_F, Age_psns_att_edu_inst_25_ov_P, High_yr_schl_comp_Yr_12_eq_M, High_yr_schl_comp_Yr_12_eq_F, High_yr_schl_comp_Yr_12_eq_P, High_yr_schl_comp_Yr_11_eq_M, High_yr_schl_comp_Yr_11_eq_F, High_yr_schl_comp_Yr_11_eq_P, High_yr_schl_comp_Yr_10_eq_M, High_yr_schl_comp_Yr_10_eq_F, High_yr_schl_comp_Yr_10_eq_P, High_yr_schl_comp_Yr_9_eq_M, High_yr_schl_comp_Yr_9_eq_F, High_yr_schl_comp_Yr_9_eq_P, High_yr_schl_comp_Yr_8_belw_M, High_yr_schl_comp_Yr_8_belw_F, High_yr_schl_comp_Yr_8_belw_P, High_yr_schl_comp_D_n_g_sch_M, High_yr_schl_comp_D_n_g_sch_F, High_yr_schl_comp_D_n_g_sch_P, Count_psns_occ_priv_dwgs_M, Count_psns_occ_priv_dwgs_F, Count_psns_occ_priv_dwgs_P, Count_Persons_other_dwgs_M, Count_Persons_other_dwgs_F, Count_Persons_other_dwgs_P
2016Census_G02_NSW_LGA	LGA_CODE_2016, Median_age_persons, Median_mortgage_repay_monthly, Median_tot_prsnl_inc_weekly, Median_rent_weekly, Median_tot_fam_inc_weekly, Average_num_psns_per_bedroom, Median_tot_hhd_inc_weekly, Average_household_size

NSW_LGA_CODE	LGA_CODE, LGA_NAME
NSW_LGA_SUBURB	LGA_NAME, SUBURB_NAME
Listing files	LISTING_ID, SCRAPE_ID, SCRAPED_DATE, HOST_ID, HOST_NAME, HOST_SINCE, HOST_IS_SUPERHOST, HOST_NEIGHBOURHOOD, LISTING_NEIGHBOURHOOD, PROPERTY_TYPE, ROOM_TYPE, ACCOMMODATES, PRICE, HAS_AVAILABILITY, AVAILABILITY_30, NUMBER_OF_REVIEWS, REVIEW_SCORES_RATING ,REVIEW_SCORES_ACCURACY, REVIEW_SCORES_CLEANLINESS, REVIEW_SCORES_CHECKIN, REVIEW_SCORES_COMMUNICATION, REVIEW_SCORES_VALUE

Uploading to Airflow Storage 'Buckets'

We were tasked to upload 5 files into the airflow cloud storage, the files were the census files, LGA files and the first month of Airbnb data (05_2020.csv).

So in data/raw directory we uploaded these files.

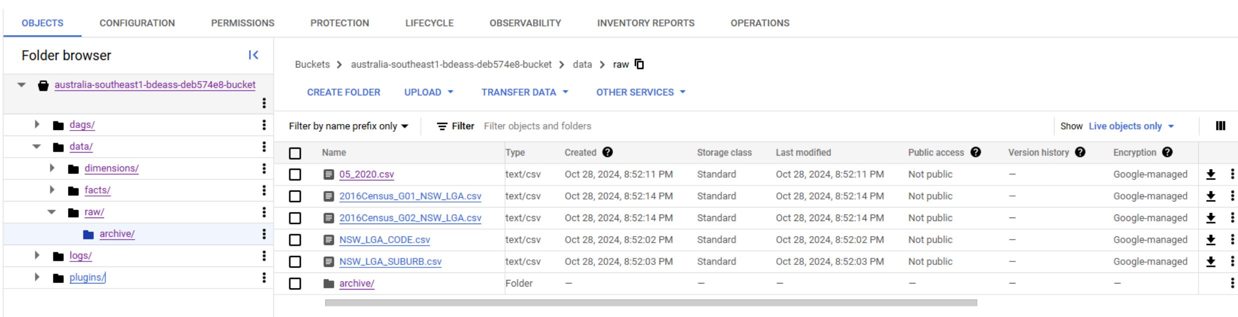


Figure 1: Files uploaded to airflow storage buckets

Creating Schema on Dbeaver

After uploading the files in the buckets the next step was to make a schema called "Bronze" that would hold the data in separate tables.

To progress with this step we setup our Dbeaver with Google Cloud Platform's SQL service through our private IP. And after having a successful connection, we created a new worksheet where we made our query of creating schema and all the tables.

```
CREATE SCHEMA IF NOT EXISTS BRONZE
```

Figure 2: Creating Bronze Schema

```sql
CREATE TABLE bronze.raw_listing (
    LISTING_ID INT,
    SCRAPE_ID BIGINT,
    SCRAPED_DATE VARCHAR,
    HOST_ID INT,
    HOST_NAME VARCHAR,
    HOST_SINCE DATE,
    HOST_IS_SUPERHOST VARCHAR,
    HOST_NEIGHBOURHOOD VARCHAR,
    LISTING_NEIGHBOURHOOD VARCHAR,
    PROPERTY_TYPE VARCHAR,
    ROOM_TYPE VARCHAR,
    ACCOMMODATES INT,
    PRICE INT,
    HAS_AVAILABILITY VARCHAR,
    AVAILABILITY_30 INT,
    NUMBER_OF_REVIEWS INT,
    REVIEW_SCORES_RATING INT,
    REVIEW_SCORES_ACCURACY INT,
    REVIEW_SCORES_CLEANLINESS INT,
    REVIEW_SCORES_CHECKIN INT,
    REVIEW_SCORES_COMMUNICATION INT,
    REVIEW_SCORES_VALUE INT
);
```

Figure 3: Creating listing table

```sql
create TABLE BRONZE.RAW_LGACODE (
    LGA_CODE INT,
    LGA_NAME  VARCHAR

);
```

Figure 4: Creating LGACODE table

```sql
create  TABLE BRONZE.RAW_LGASUBURB (
    LGA_NAME  VARCHAR,
    SUBURB_NAME VARCHAR

);
```

Figure 5: Creating LGASUBURB table

```sql
CREATE  TABLE BRONZE.RAW_CENSUSG1 (
    LGA_CODE_2016 VARCHAR,Tot_P_M INT,Tot_P_F INT, Tot_P_P INT,Age_0_4_yr_M INT,
    Age_0_4_yr_F INT, Age_0_4_yr_P INT,Age_5_14_yr_M INT, Age_5_14_yr_F INT,
    Age_5_14_yr_P INT,Age_15_19_yr_M INT, Age_15_19_yr_F INT, Age_15_19_yr_P INT,
    Age_20_24_yr_M INT, Age_20_24_yr_F INT, Age_20_24_yr_P INT, Age_25_34_yr_M INT,
    Age_25_34_yr_F INT,Age_25_34_yr_P INT,Age_35_44_yr_M INT,
    Age_35_44_yr_F INT,Age_35_44_yr_P INT,Age_45_54_yr_M INT,
    Age_45_54_yr_F INT,Age_45_54_yr_P INT,Age_55_64_yr_M INT,Age_55_64_yr_F INT,
    Age_55_64_yr_P INT, Age_65_74_yr_M INT,Age_65_74_yr_F INT,Age_65_74_yr_P INT,
    Age_75_84_yr_M INT,Age_75_84_yr_F INT,Age_75_84_yr_P INT,Age_85ov_M INT,
    Age_85ov_F INT,Age_85ov_P INT,Counted_Census_Night_home_M INT,
    Counted_Census_Night_home_F INT,Counted_Census_Night_home_P INT,
    Count_Census_Nt_Ewhere_Aust_M INT,Count_Census_Nt_Ewhere_Aust_F INT,
    Count_Census_Nt_Ewhere_Aust_P INT,Indigenous_psns_Aboriginal_M INT,
    Indigenous_psns_Aboriginal_F INT,Indigenous_psns_Aboriginal_P INT,
    Indig_psns_Torres_Strait_Is_M INT,Indig_psns_Torres_Strait_Is_F INT,
    Indig_psns_Torres_Strait_Is_P INT,Indig_Bth_Abor_Torres_St_Is_M INT,
    Indig_Bth_Abor_Torres_St_Is_F INT,Indig_Bth_Abor_Torres_St_Is_P INT,Indigenous_P_Tot_M INT,
    Indigenous_P_Tot_F INT,Indigenous_P_Tot_P INT,Birthplace_Australia_M INT,Birthplace_Australia_F INT,
    Birthplace_Australia_P INT,Birthplace_Elsewhere_M INT,Birthplace_Elsewhere_F INT,Birthplace_Elsewhere_P INT,
    Lang_spoken_home_Eng_only_M INT,Lang_spoken_home_Eng_only_F INT,Lang_spoken_home_Eng_only_P INT,
    Lang_spoken_home_Oth_Lang_M INT,Lang_spoken_home_Oth_Lang_F INT,Lang_spoken_home_Oth_Lang_P INT,
    Australian_citizen_M INT,Australian_citizen_F INT,Australian_citizen_P INT,
    Age_psns_att_educ_inst_0_4_M INT,Age_psns_att_educ_inst_0_4_F INT,Age_psns_att_educ_inst_0_4_P INT,
    Age_psns_att_educ_inst_5_14_M INT,Age_psns_att_educ_inst_5_14_F INT,Age_psns_att_educ_inst_5_14_P INT,
    Age_psns_att_edu_inst_15_19_M INT,Age_psns_att_edu_inst_15_19_F INT,Age_psns_att_edu_inst_15_19_P INT,
    Age_psns_att_edu_inst_20_24_M INT,Age_psns_att_edu_inst_20_24_F INT,Age_psns_att_edu_inst_20_24_P INT,
    Age_psns_att_edu_inst_25_ov_M INT,Age_psns_att_edu_inst_25_ov_F INT,Age_psns_att_edu_inst_25_ov_P INT,
    High_yr_schl_comp_Yr_12_eq_M INT,High_yr_schl_comp_Yr_12_eq_F INT,High_yr_schl_comp_Yr_12_eq_P INT,High_yr_schl_comp_Yr_11_eq_M INT,
    High_yr_schl_comp_Yr_11_eq_F INT,High_yr_schl_comp_Yr_11_eq_P INT,High_yr_schl_comp_Yr_10_eq_M INT,High_yr_schl_comp_Yr_10_eq_F INT,
    High_yr_schl_comp_Yr_10_eq_P INT,High_yr_schl_comp_Yr_9_eq_M INT,High_yr_schl_comp_Yr_9_eq_F INT,High_yr_schl_comp_Yr_9_eq_P INT,
    High_yr_schl_comp_Yr_8_belw_M INT,High_yr_schl_comp_Yr_8_belw_F INT,High_yr_schl_comp_Yr_8_belw_P INT,
    High_yr_schl_comp_D_n_g_sch_M INT,High_yr_schl_comp_D_n_g_sch_F INT,High_yr_schl_comp_D_n_g_sch_P INT,Count_psns_occ_priv_dwgs_M INT,
    Count_psns_occ_priv_dwgs_F INT,Count_psns_occ_priv_dwgs_P INT,
    Count_Persons_other_dwgs_M INT,Count_Persons_other_dwgs_F INT,Count_Persons_other_dwgs_P INT
);
```

Figure 6: Creating census G1 table

```sql
CREATE TABLE bronze.raw_censusg2 (
    LGA_CODE_2016 VARCHAR,
    Median_age_persons INT,
    Median_mortgage_repay_monthly INT,
    Median_tot_prsnl_inc_weekly INT,
    Median_rent_weekly INT,
    Median_tot_fam_inc_weekly INT,
    Average_num_psns_per_bedroom FLOAT,
    Median_tot_hhd_inc_weekly INT,
    Average_household_size FLOAT
);
```

Figure 7: Creating census G2 table

DAG Creation and Trigger

After uploading the files to the buckets and building up a schema and tables for the datasets, the next step is to make a DAG.py file which would hold all the necessary steps trigger the data transfer from the buckets to the Dbeaver.

We created a DAG file with **schedule_interval=None** such that it does not trigger in a scheduled manner, rather manual trigger is needed. After we have made our DAG file we upload it to the Airflow Buckets in the dags/ directory.

After that we went into the Airflow UI where we see the list of DAGs we have in the system. We then go inside of our newly created DAG and trigger it.

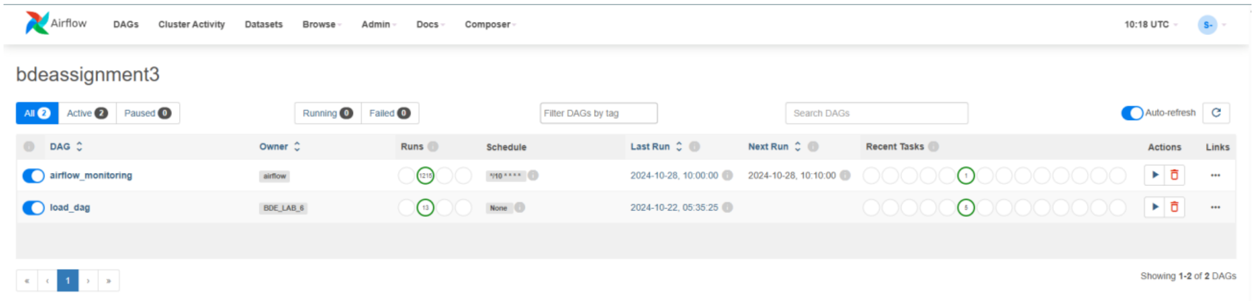


Figure 8: List of DAG files in the system

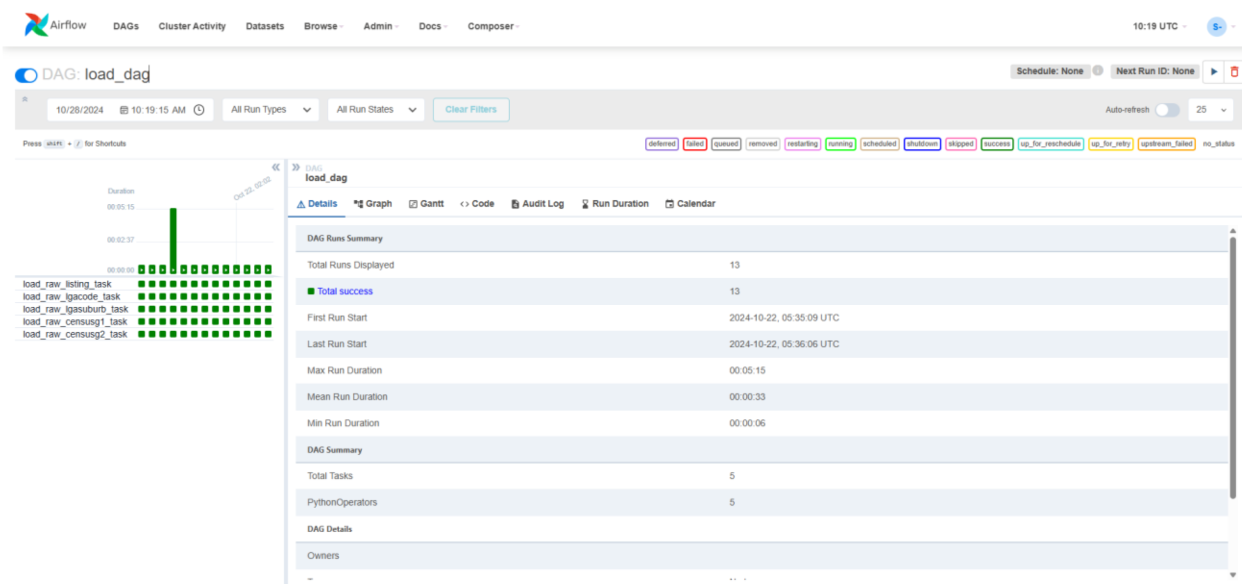


Figure 9: Successful Trigger of DAG file

dbt Data Warehousing (Setting up Medallion Architecture)

The data warehouse architecture was built using a medallion approach, which segments the data into three distinct layers—Bronze, Silver, and Gold. This structure enhances data quality and supports efficient querying, as each layer serves a specific purpose in the data transformation pipeline.

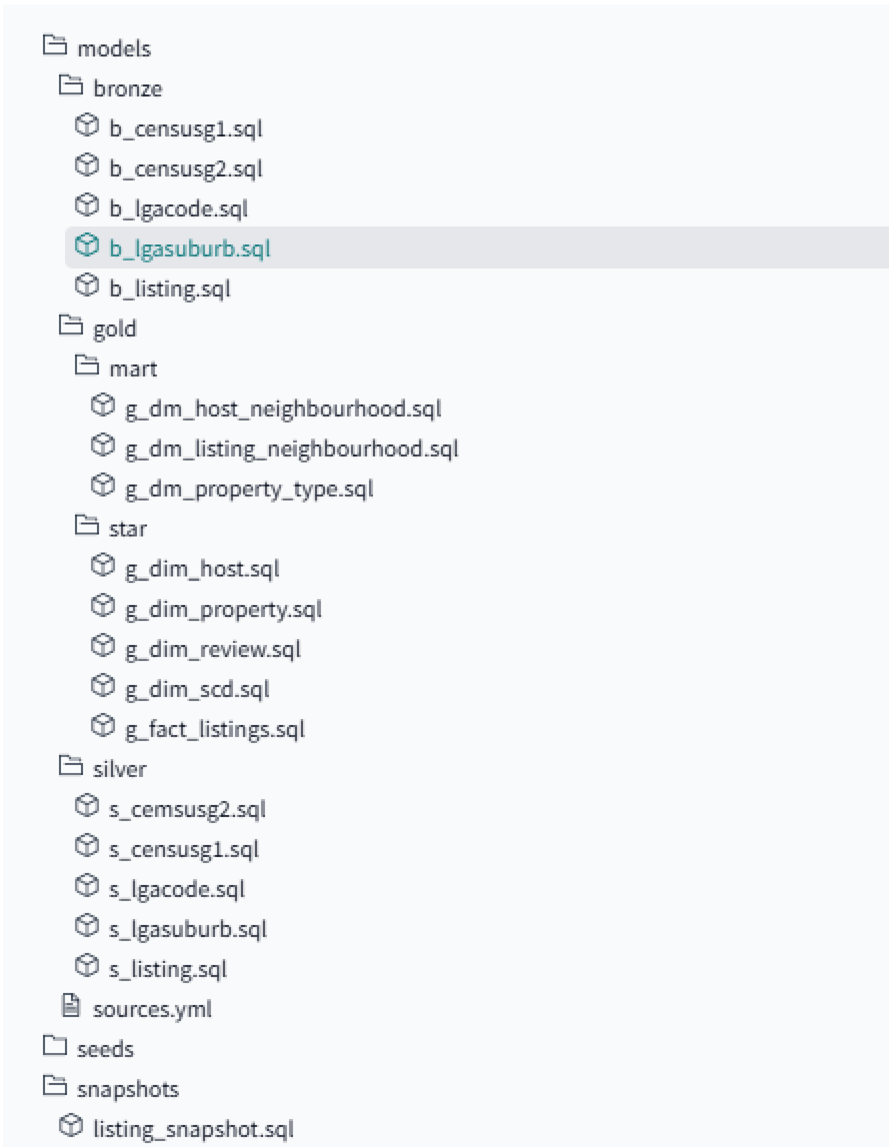


Figure 10: Successful Medallion Architecture

Bronze Layer

Raw data from Airbnb and Census datasets, initially loaded by Airflow into the Dbeaver, was stored in this layer. The Bronze layer includes tables in their unaltered form, with additional tables created for specific raw data, particularly focusing on Airbnb listings. This layer acts as a foundation for further transformations.

Silver Layer

This layer cleans and standardizes data from created tables from the Bronze schema, ensuring consistency in data formats and naming conventions. Transformations included handling null values, correcting data types, and aligning naming conventions across tables. To support Slowly Changing Dimensions (SCD) requirements, we implemented snapshots for the dimension tables, using timestamps to track changes in listing data and Local Government Area (LGA) mappings.

Gold Layer

The Gold layer was designed as a star schema, featuring dimension and fact tables. Fact tables in this layer contain only essential IDs and metrics, such as price, while dimension tables house descriptive data. This schema structure facilitates efficient data retrieval for analytical purposes. Additionally, the Gold layer includes a datamart to answer specific business questions. Views were created here using fact and dimension tables, with support for SCD Type 2 to capture historical changes in data attributes.

To build a comprehensive schema, at least four dimension tables were created, covering entities such as listings, hosts, suburbs, and LGAs. Dimension tables were designed to provide descriptive information, whereas fact tables were tailored to store metrics linked to specific dimension records, thus enabling aggregation and analysis.

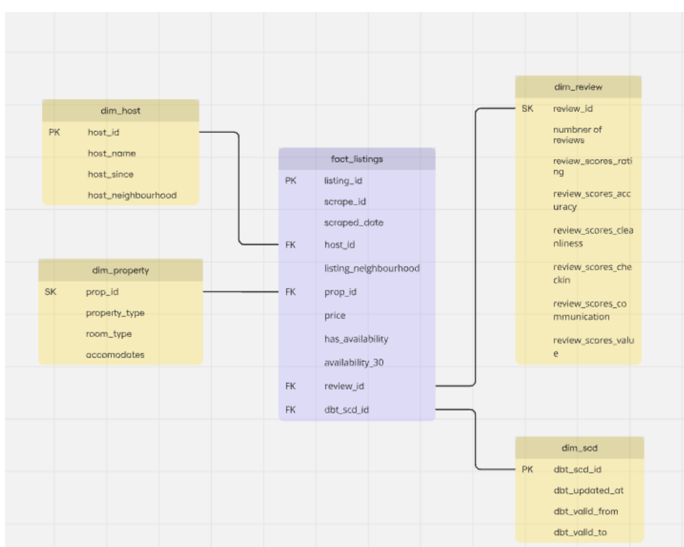


Figure 11: Dimension and Facts Table setup

- **Dimension Tables**: Dimensions like host, review, property and SCD were established to organize Airbnb and Census data in a meaningful way. Data transformations were applied to ensure attributes were clean and reliable for reporting. Snapshots were employed to maintain historical records for each dimension, addressing SCD requirements.
- **Fact Tables**: The fact tables were constructed to store metrics that could be aggregated, such as prices and review scores. The fact tables included foreign keys referencing dimension tables, creating an interconnected schema to support the star schema model.

The datamart in the Gold layer was created through three specific views, each designed to provide answers to predefined business questions. These views aggregate data in various dimensions and present the results in an accessible format.

- **dm_listing_neighbourhood**: This view aggregates data per listing_neighbourhood and month/year, covering metrics such as active listing rates, price statistics, superhost rates, review score averages, and estimated revenue per active listing. The data is ordered by listing_neighbourhood and month/year to facilitate chronological and geographical analysis.

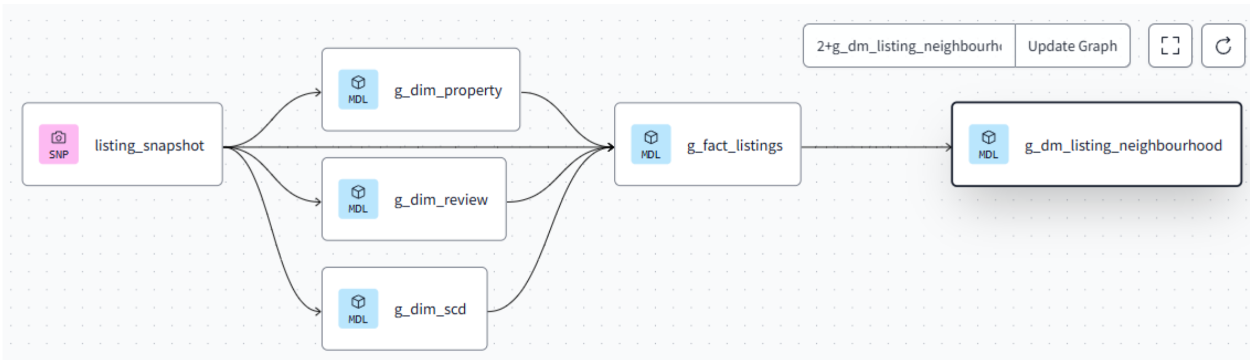


Figure 12: Displaying the lineage of the datamart

	listing_neighbourhood	month_year	active_listing_rate	min_price	max_price	median_price	avg_price	distinct_hosts	superhost_rate	avg_review_scores_rating	total_stays	avg_estimated_revenue_per_
1	BAYSIDE	2020-05-01 00:00:00.000	100	0	15,367	86	167.8596303241	7,466	1.8885614787	58.9358424859	144,478	2,464.9228502545
2	BLACKTOWN	2020-05-01 00:00:00.000	100	23	1,099	61	91.2363636364	1,430	2.1678321678	56.5062937063	16,653	1,071.6342657343
3	BURWOOD	2020-05-01 00:00:00.000	100	15	2,440	70	154.3274853801	1,197	1.5873015873	59.8011695906	21,053	2,120.8629908104
4	CAMDEN	2020-05-01 00:00:00.000	100	35	350	80	106.8056872038	211	4.7393364929	56.1090047393	2,746	1,184.6682464455
5	CAMPBELLTOWN	2020-05-01 00:00:00.000	100	20	2,500	74	136.1097804391	501	3.9920159681	70.2794411178	7,773	1,249.6367265469
6	CANADA BAY	2020-05-01 00:00:00.000	100	24	2,440	110	152.5781968721	2,174	1.7479300828	59.0404783809	41,610	2,795.346826127
7	CANTERBURY-BANKSTOWN	2020-05-01 00:00:00.000	100	12	1,000	70	94.5864711763	3,001	1.6994335222	57.6554481839	44,614	1,345.4901699434
8	CUMBERLAND	2020-05-01 00:00:00.000	100	15	14,999	100	154.8222305389	2,672	2.4700598802	68.6781437126	49,241	3,033.8506736527
9	FAIRFIELD	2020-05-01 00:00:00.000	100	20	501	69	98.2093596059	406	1.2315270936	66.2561576355	5,125	1,123.8645320197
10	GEORGES RIVER	2020-05-01 00:00:00.000	100	0	3,500	80	123.5240334378	1,914	1.776384535	58.6107628004	33,934	1,906.316091954
11	HORNSBY	2020-05-01 00:00:00.000	100	20	5,000	89	148.6868052199	2,069	3.7699371677	71.1300144998	34,117	2,025.4108264862
12	HUNTERS HILL	2020-05-01 00:00:00.000	100	35	2,440	183	327.1074380165	242	1.652892562	61.7190082645	4,990	7,222.7727272727
13	INNER WEST	2020-05-01 00:00:00.000	100	15	9,022	106	185.3365955959	11,444	2.4117441454	67.4292205523	256,181	3,785.2292030758
14	LANE COVE	2020-05-01 00:00:00.000	100	35	3,000	119	239.9214437367	1,413	1.5569709837	63.5966029724	25,065	4,404.8973814579
15	LIVERPOOL	2020-05-01 00:00:00.000	100	15	586	91	118.8215339233	678	2.6548672566	62.3879056047	9,968	1,689.5737463127
16	MOSMAN	2020-05-01 00:00:00.000	100	31	15,309	197	449.4460580913	2,410	1.9502074689	62.8684647303	51,465	8,777.3929460581
17	NORTHERN BEACHES	2020-05-01 00:00:00.000	100	12	7,654	199	342.3471547163	23,249	2.5764549013	65.7328057121	501,512	6,719.1793195406
18	NORTH SYDNEY	2020-05-01 00:00:00.000	100	21	10,000	139	219.8115396676	6,378	2.3988711195	66.5721229225	128,737	3,875.4476324867
19	PARRAMATTA	2020-05-01 00:00:00.000	100	14	3,000	85	167.360959651	2,751	2.0356234097	55.2420937841	42,912	2,418.6779352963
20	PENRITH	2020-05-01 00:00:00.000	100	25	1,500	111	142.1612057667	763	4.1939711664	72.2201834862	10,215	1,402.6788990826

Figure 13: Displaying the first 20 records of the view

- **dm_property_type**: This view presents data by property_type, room_type, accommodates, and month/year. It includes metrics similar to those in dm_listing_neighbourhood but focuses on property characteristics and types. Metrics like active listing rates, price statistics, superhost rates, and revenue estimates allow for detailed insights into property types across time and geography.

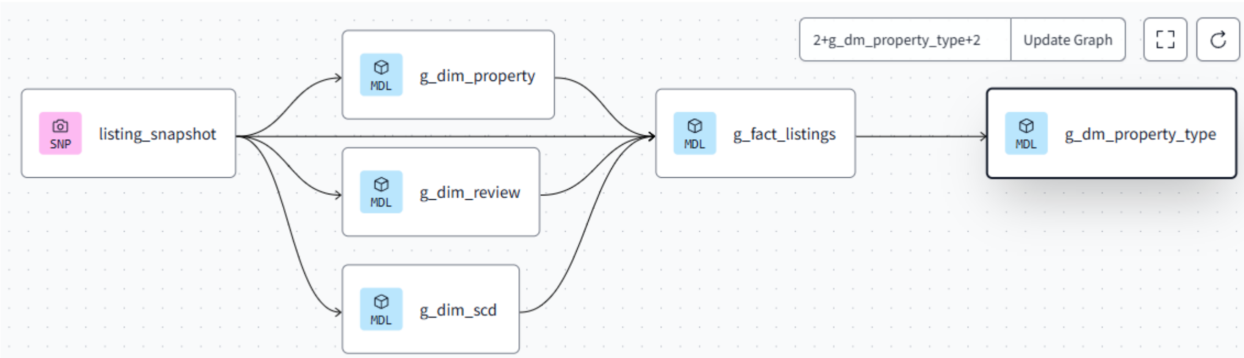


Figure 14: Displaying the lineage of the datamart

	property_type	room_type	accommodates	month_year	active_listing_rate	min_price	max_price	median_price	avg_price	distinct_hosts	s
1	Aparthotel	Entire home/apt	2	2020-05-01 00:00:00.000	100	165	165	165	165	1	
2	Aparthotel	Entire home/apt	5	2020-05-01 00:00:00.000	100	158	158	158	158	1	
3	Aparthotel	Hotel room	2	2020-05-01 00:00:00.000	100	47	501	347.5	310.75	4	
4	Aparthotel	Hotel room	4	2020-05-01 00:00:00.000	100	499	501	501	500.3333333333	3	
5	Aparthotel	Private room	2	2020-05-01 00:00:00.000	100	80	351	130	172.75	12	
6	Aparthotel	Private room	3	2020-05-01 00:00:00.000	100	139	139	139	139	1	
7	Aparthotel	Private room	4	2020-05-01 00:00:00.000	100	161	249	205	205	4	
8	Apartment	Entire home/apt	1	2020-05-01 00:00:00.000	100	20	804	98.5	126.1470588235	408	
9	Apartment	Entire home/apt	2	2020-05-01 00:00:00.000	100	15	12,001	130	221.7600979192	28,595	
10	Apartment	Entire home/apt	3	2020-05-01 00:00:00.000	100	15	2,544	139	194.4357579878	5,884	
11	Apartment	Entire home/apt	4	2020-05-01 00:00:00.000	100	0	10,000	181	238.8345864662	24,605	
12	Apartment	Entire home/apt	5	2020-05-01 00:00:00.000	100	31	2,440	194	240.5536437247	4,940	
13	Apartment	Entire home/apt	6	2020-05-01 00:00:00.000	100	28	6,000	219	310.1389108129	6,335	
14	Apartment	Entire home/apt	7	2020-05-01 00:00:00.000	100	67	2,001	250	320.8666666667	660	
15	Apartment	Entire home/apt	8	2020-05-01 00:00:00.000	100	75	10,000	248.5	442.0721649485	776	
16	Apartment	Entire home/apt	9	2020-05-01 00:00:00.000	100	80	700	214	257.8857142857	105	
17	Apartment	Entire home/apt	10	2020-05-01 00:00:00.000	100	90	1,000	300	384.8148148148	108	
18	Apartment	Entire home/apt	11	2020-05-01 00:00:00.000	100	298	580	344.5	391.75	8	
19	Apartment	Entire home/apt	12	2020-05-01 00:00:00.000	100	240	700	337	391.125	16	
20	Apartment	Entire home/apt	13	2020-05-01 00:00:00.000	100	250	250	250	250	1	
21	Apartment	Entire home/apt	14	2020-05-01 00:00:00.000	100	151	501	243.5	273	18	
22	Apartment	Entire home/apt	15	2020-05-01 00:00:00.000	100	268	268	268	268	1	
23	Apartment	Entire home/apt	16	2020-05-01 00:00:00.000	100	100	801	400	434	21	
24	Apartment	Hotel room	4	2020-05-01 00:00:00.000	100	300	350	325	325	4	
25	Apartment	Hotel room	6	2020-05-01 00:00:00.000	100	335	335	335	335	1	

Figure 15: Displaying the first 25 records of the view

- **dm_host_neighbourhood**: This view provides data aggregated per host_neighbourhood_lga and month/year, transforming host_neighbourhood into the corresponding LGA. Key metrics include the number of distinct hosts, estimated revenue, and estimated revenue per host. This view is structured to support analysis of host behaviors and revenue generation patterns within LGAs.

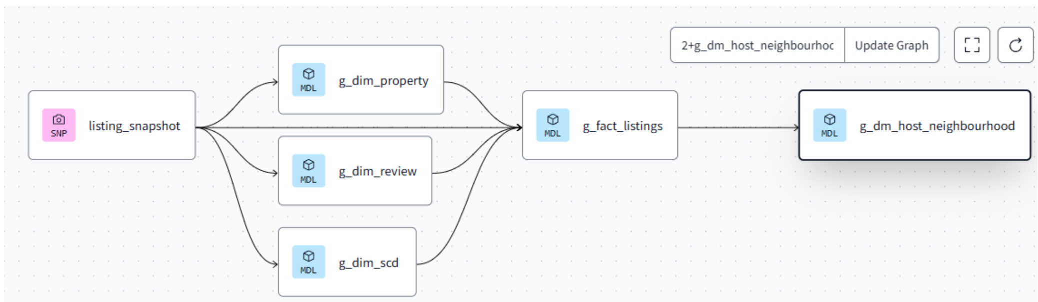


Figure 16: Displaying the lineage of the datamart

	host_neighbourhood_lga	month_year	distinct_hosts	estimated_revenue	estimated_revenue_per_host
1	ABBOTSFORD	2020-05-01 00:00:00.000	16	470,421	29,401.3125
2	ALEXANDRIA	2020-05-01 00:00:00.000	123	2,482,511	20,183.0162601626
3	ALLAWAH	2020-05-01 00:00:00.000	13	126,819	9,755.3076923077
4	ANNANDALE	2020-05-01 00:00:00.000	244	5,311,298	21,767.6147540984
5	ARNCLIFFE	2020-05-01 00:00:00.000	92	1,176,814	12,791.4565217391
6	ARTARMON	2020-05-01 00:00:00.000	26	410,722	15,797
7	ASHBURY	2020-05-01 00:00:00.000	2	33,500	16,750
8	ASHFIELD	2020-05-01 00:00:00.000	63	901,341	14,307
9	AUBURN	2020-05-01 00:00:00.000	22	144,890	6,585.9090909091
10	AVALON BEACH	2020-05-01 00:00:00.000	217	10,513,333	48,448.5391705069
11	BALGOWLAH	2020-05-01 00:00:00.000	192	9,196,396	47,897.8958333333
12	BALMAIN	2020-05-01 00:00:00.000	189	25,020,245	132,382.2486772487
13	BALMORAL	2020-05-01 00:00:00.000	1	16,305	16,305
14	BANKSIA	2020-05-01 00:00:00.000	7	107,115	15,302.1428571429
15	BANKSTOWN	2020-05-01 00:00:00.000	40	665,953	16,648.825
16	BARDWELL PARK	2020-05-01 00:00:00.000	2	22,702	11,351
17	BARDWELL VALLEY	2020-05-01 00:00:00.000	4	37,200	9,300
18	BEACONSFIELD	2020-05-01 00:00:00.000	22	638,685	29,031.1363636364
19	BELFIELD	2020-05-01 00:00:00.000	5	92,400	18,480
20	BELLEVUE HILL	2020-05-01 00:00:00.000	225	9,363,933	41,617.48
21	BELMORE	2020-05-01 00:00:00.000	14	118,495	8,463.9285714286
22	BERALA	2020-05-01 00:00:00.000	4	253,583	63,395.75
23	BEVERLY HILLS	2020-05-01 00:00:00.000	11	213,263	19,387.5454545455
24	BEXLEY	2020-05-01 00:00:00.000	25	713,588	28,543.52
25	BEXLEY NORTH	2020-05-01 00:00:00.000	3	27,194	9,064.6666666667

Figure 17: Displaying the first 25 records of the view

Building up the Schema

After creating all the files and dealing with all the necessary steps to clean and structure the data, the next step is to run the dbt and build up the schema which would reflect on the Dbeaver. And from there on we start the querying.

To do so, the command is “dbt build” and running it if successful would green tick all the created files. And finally we would see the reflection on the Dbeaver.

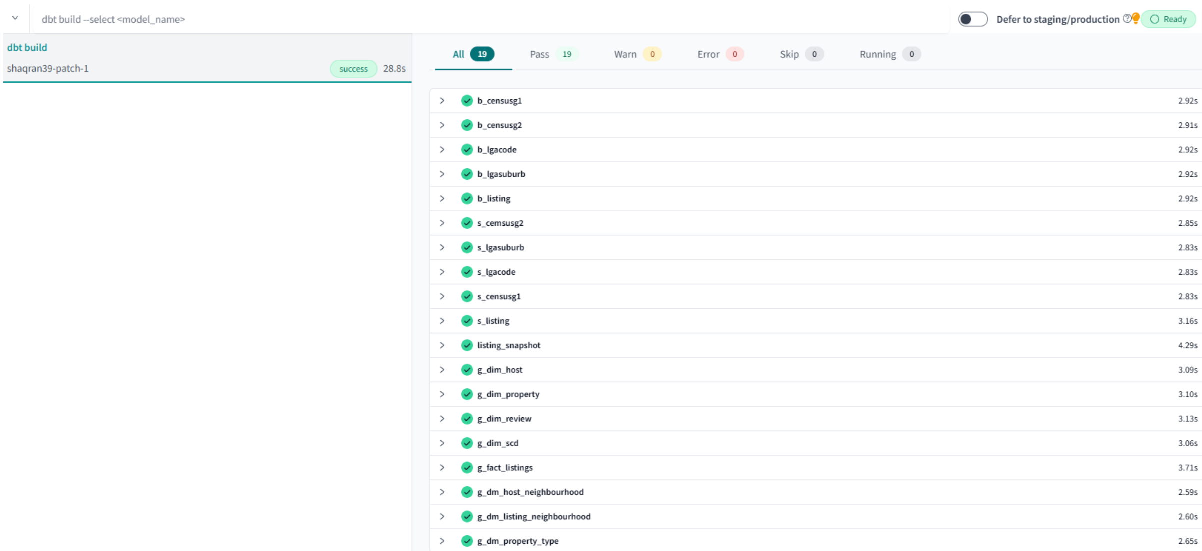


Figure 18: Successful run of “dbt build”

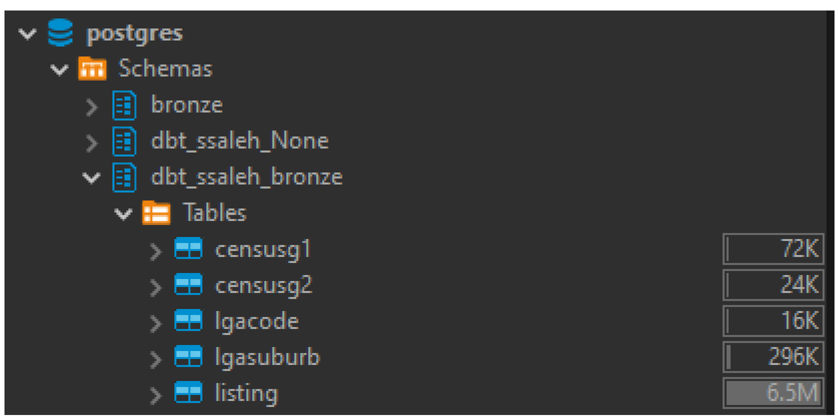


Figure 19: Successful development of Bronze schema on Dbeaver

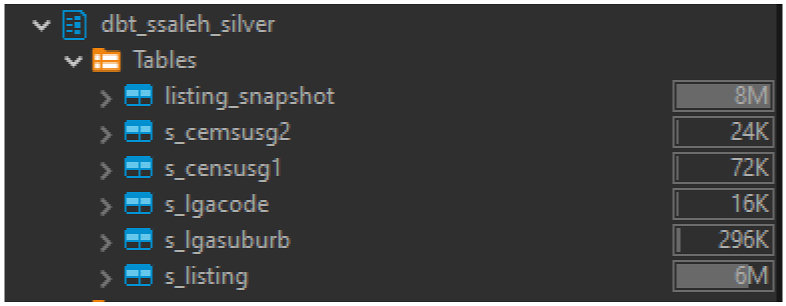


Figure 20: Successful development of Silver schema on Dbeaver

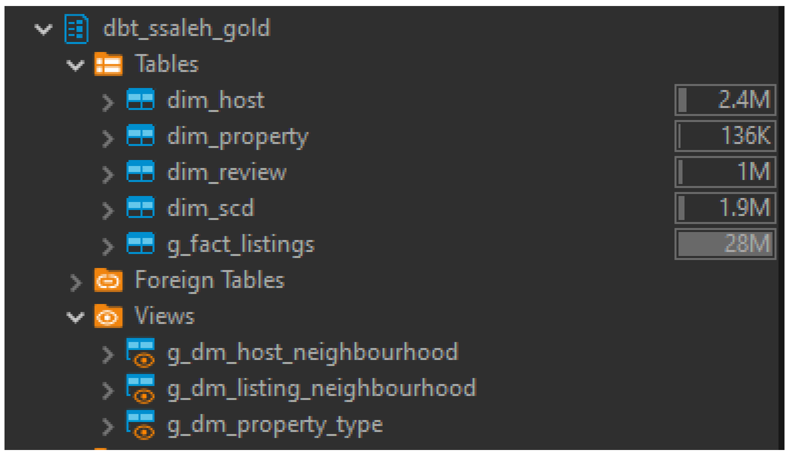
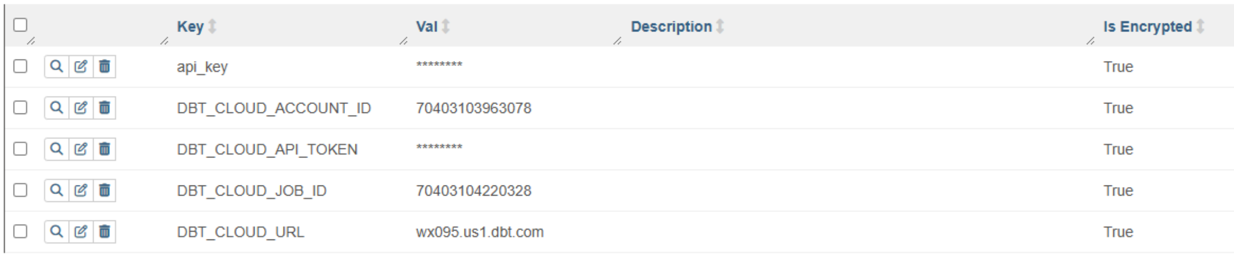


Figure 21: Successful development of Gold schema on Dbeaver

DAG Modification and Loading Remaining Data

Since in the first run of our assignment we only worked with one property listing file, it is imperative to take all the listing files into account. For this we needed to make a slight change to our DAG file such that it loads all the listing files, adds all the data to the same table and trigger the dbt cloud job.

For that we needed 4 new variables and these were added into the airflow variables



Key	Val	Description	Is Encrypted
api_key	********		True
DBT_CLOUD_ACCOUNT_ID	70403103963078		True
DBT_CLOUD_API_TOKEN	********		True
DBT_CLOUD_JOB_ID	70403104220328		True
DBT_CLOUD_URL	wx095.us1.dbt.com		True

Figure 21: Variable addition in Airflow

The values came from dbt. The DBT_CLOUD_ACCOUNT_ID, DBT_CLOUD_JOB_ID and DBT_CLOUD_URL comes from the API Trigger option in the JOB of the dbt production deployment.

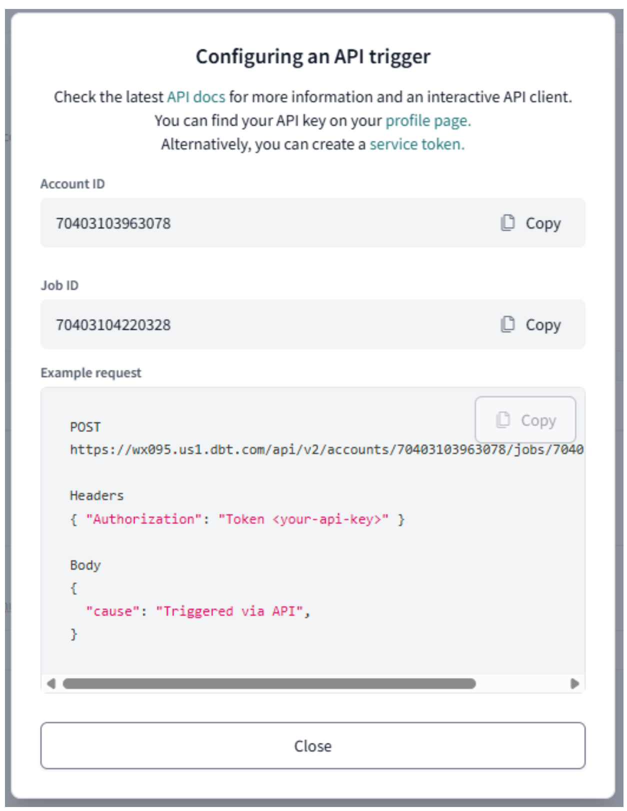


Figure 22: API Trigger information

The value for DBT_CLOUD_API_TOKEN is found as we create a new personal access token from the account settings.

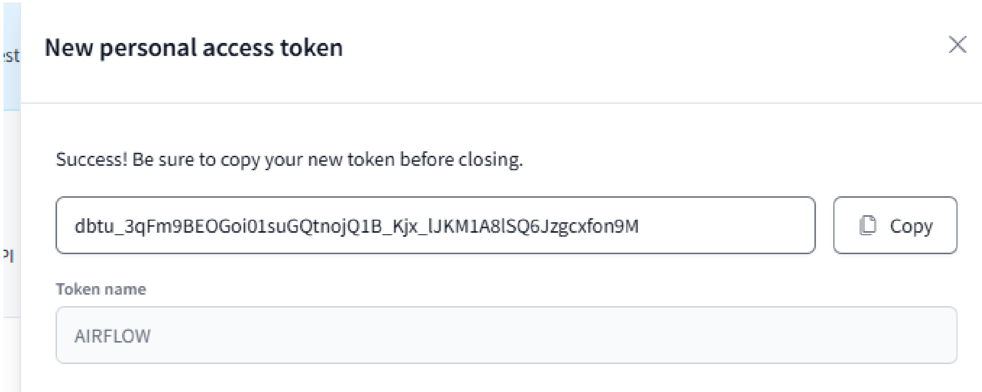


Figure 23: API Token information

Using these informations we update the DAG file and add a new section to it.

```python
#########################################################
#
#   Function to trigger dbt Cloud Job
#
#########################################################

def trigger_dbt_cloud_job(**kwargs):
    # Get the dbt Cloud URL, account ID, and job ID from Airflow Variables
    dbt_cloud_url = Variable.get("DBT_CLOUD_URL")
    dbt_cloud_account_id = Variable.get("DBT_CLOUD_ACCOUNT_ID")
    dbt_cloud_job_id = Variable.get("DBT_CLOUD_JOB_ID")

    # Define the URL for the dbt Cloud job API dynamically using URL, account ID, and job ID
    url = f"https://{dbt_cloud_url}/api/v2/accounts/{dbt_cloud_account_id}/jobs/{dbt_cloud_job_id}/run/"

    # Get the dbt Cloud API token from Airflow Variables
    dbt_cloud_token = Variable.get("DBT_CLOUD_API_TOKEN")

    # Define the headers and body for the request
    headers = {
        'Authorization': f'Token {dbt_cloud_token}',
        'Content-Type': 'application/json'
    }
    data = {
        "cause": "Triggered via API"
    }

    # Make the POST request to trigger the dbt Cloud job
    response = requests.post(url, headers=headers, json=data)

    # Check if the response is successful
    if response.status_code == 200:
        logging.info("Successfully triggered dbt Cloud job.")
        return response.json()
    else:
        logging.error(f"Failed to trigger dbt Cloud job: {response.status_code}, {response.text}")
        raise AirflowException("Failed to trigger dbt Cloud job.")
```

Figure 24: New section on the updated DAG file

Issues Faced

- Issue with Initial Data Loading:

  There were issues with loading the initial datasets into Postgres due to column mismatches between the source CSV files and the database schema.

  Resolution: Manually verified the column names and adjusted the ingestion script to correctly map columns.

- Error in Sequential Data Loading:

  When extending the Airflow DAG to load datasets month-by-month in chronological order, ensuring data integrity proved challenging.

  Resolution: Modified the DAG to enforce sequential loading by setting dependencies between tasks, so each month’s data was processed sequentially without overlap.
- Issue with Triggering DBT Cloud Job:

  Setting up an Airflow task to trigger the DBT Cloud job presented issues due to misconfiguration in API variables.

  Resolution: Corrected the DBT Cloud API setup by ensuring the necessary variables (DBT_CLOUD_URL, DBT_CLOUD_ACCOUNT_ID, DBT_CLOUD_JOB_ID, and DBT_CLOUD_API_TOKEN) were properly set in Airflow. Updated the function to dynamically create the API URL and ensure proper error handling for failed requests.

- Issue with GROUP BY and COUNT Distinct Aggregation:

  Errors occurred when attempting to group data by specific columns (host_neighbourhood or listing_neighbourhood) and using count(distinct ...) for host counts, leading to incorrect aggregations.

  Resolution: Revised aggregation logic in SQL queries by adding COALESCE functions and null handling to avoid errors. Ensured columns used in grouping matched the schema accurately.

- Issue with Task Dependencies in Airflow DAG:

  Ensuring the task to load data sequentially and trigger the DBT Cloud job in the correct order presented a challenge.

  Resolution: Configured task dependencies within the DAG by setting task ordering and using set_downstream to ensure each task executed in the correct sequence, maintaining data integrity across loading and transformation stages.

Business Question Analysis

Question C

The answer to question (c) is presented in the form of a table that shows the best types of listings for the “Northern Beaches” neighborhood based on stay counts, which can be interpreted as the number of nights or instances these properties were booked.

	listing_neighbourhood	property_type	room_type	accommodates	stay_count
1	Northern Beaches	Apartment	Entire home/apt	4	670
2	Northern Beaches	Apartment	Entire home/apt	2	586
3	Northern Beaches	House	Entire home/apt	8	442
4	Northern Beaches	House	Entire home/apt	6	387
5	Northern Beaches	Apartment	Private room	2	387

Figure 24: Desired table for business question C

Analysis:

- Apartments with "Entire home/apt" room type consistently rank at the top, with various capacities (`accommodates` ranging from 2 to 4) having the highest stay counts. This suggests that apartments providing the entire home are the most popular listing type for generating high stay counts in this neighborhood.
- Properties with higher `accommodates` values tend to have slightly lower stay counts. The highest stay count (`670`) is associated with an apartment accommodating 4 people. Lower accommodation capacities (2-4) generally perform better in terms of total stays than properties with higher capacities, like houses accommodating 8 people.
- Both apartments and houses with "Entire home/apt" room type appear frequently, indicating a preference among guests for listings that provide private, entire accommodations.
- Even though "Private room" also appears in the top listings, it has the same stay count as the least popular "Entire home/apt" type.
- For hosts looking to maximize stays, offering smaller apartments with the “Entire home/apt” room type seems beneficial. Although houses are also in demand, they rank lower than apartments, indicating that guests may prefer compact, self-contained accommodations over larger spaces in this neighborhood.
- The analysis shows that for the "Northern Beaches" area, apartments accommodating fewer people are optimal for maximizing bookings. Hosts in this area could consider targeting such configurations to attract more guests. Additionally, it may be beneficial to focus on "Entire home/apt" listings rather than individual private rooms, as they are more popular.

This table provides valuable insights into what type of listings are most attractive to guests in the "Northern Beaches" neighborhood, which could inform listing strategies for Airbnb hosts.

Conclusion

In this report, we developed an end-to-end data pipeline for Airbnb and Census data, utilizing Apache Airflow, dbt Cloud, and PostgreSQL. Through the Medallion Architecture, we successfully transformed raw data into structured layers (Bronze, Silver, and Gold) to support business analysis. The integration of Airbnb listings with Census data provided insightful views and datamarts that answer specific business questions related to demographics, property performance, and host behavior across Local Government Areas (LGAs).

The analysis of question (c) highlighted the optimal property configurations within the "Northern Beaches" neighborhood, revealing that apartments offering "Entire home/apt" listings with lower accommodation capacities have the highest stay counts. This insight underscores the preference for compact, private accommodations among Airbnb guests in this area, offering strategic guidance for hosts to maximize occupancy.

Throughout the pipeline development, we encountered several challenges, such as data loading issues, task dependencies, and query configuration, all of which were addressed to ensure data integrity and seamless execution. This project demonstrates the efficacy of using modern data engineering tools to create robust pipelines capable of generating actionable insights, laying a foundation for further analytical applications and data-driven decision-making in similar domains.

References

Some references used:

- dbt Labs. (2023). dbt Documentation. Retrieved from https://docs.getdbt.com/
- Ellis, J. (2021). The dbt Way: A Framework for Modern Data Transformation. dbt Labs. Retrieved from https://www.getdbt.com/blog/the-dbt-way-a-framework-for-modern-data-transformation/
- Apache Software Foundation. (2023). Apache Airflow Documentation. Retrieved from https://airflow.apache.org/docs/
- Leip, M. (2019). Data Pipelines with Apache Airflow: A hands-on guide for building, scheduling, and monitoring scalable data processing workflows. O'Reilly Media.
- Google Cloud. (2023). Google Cloud Documentation. Retrieved from https://cloud.google.com/docs/
- Sato, K. (2017). Building a Scalable Data Pipeline with Google Cloud Platform. O'Reilly Media.