Assignment 3

Building ELT Data Pipelines with Airflow

Roy Hegde
Student ID: 24667610

94693 - Big Data Engineering - Spring 2024
Master of Data Science and Innovation
University of Technology of Sydney

Table of Contents

1. Project Overview

The goal of this project is to create a reliable, automated data pipeline that can ingest, convert, and analyse census and Airbnb statistics to provide insightful information about neighbourhood demographics and Airbnb listing performance. This pipeline uses Dbeaver-PostgreSQL, Apache Airflow, dbt Cloud, and Google Cloud Storage (GCS) in conjunction with the Medallion Architecture (Bronze, Silver, Gold) to organise data for the best possible analysis.

The Medallion Architecture layers are designed as follows:

- **Bronze Layer**: Retains unprocessed raw data from a variety of sources, including local government entities, Airbnb, and census data.
- **Silver Layer**: Prepares organised tables for analysis by cleaning and standardising the data.
- **Gold Layer**: Uses certain datamart views to address business problems and arranges data into a star schema for effective querying.

Every stage of the project is described in this report, including data ingestion, transformation, Airflow orchestration, and analytical insights from dbt-generated views.

2. Dataset Overview

Both static and dynamic data were utilised in this project; they were all kept in Google Cloud Storage and accessible via Airflow for PostgreSQL ingestion.

- **Airbnb Listings**: A set of monthly CSV files with listing data, including listing_id, host_name, property_type, price, and availability metrics (e.g., 05_2020.csv to 04_2021.csv). To ensure chronological consistency, this data is fed into the pipeline month by month.
- **Census Data:** 2016Census_G01_NSW_LGA.csv and 2016Census_G02_NSW_LGA.csv primary census files include socioeconomic and demographic data on the Local Government Areas (LGAs) of New South Wales. While G02 covers measures like median age, income, and rent, G01 contains population statistics.
- **LGA Codes and Suburb Mapping**: Suburbs are mapped to LGAs using two more static datasets, NSW_LGA_CODE.csv and NSW_LGA_SUBURB.csv. Analysis of listings at the neighbourhood level is supported by this spatial reference.

By storing these datasets in GCS, scalability is guaranteed in the event that additional files are added or dataset sizes grow, and flexible access for ingestion is made possible.

3. Data Ingestion and Staging (Bronze Layer)

In order to ensure data integrity and facilitate traceability, the Bronze layer acts as the fundamental staging area for raw data intake, maintaining each dataset's original structure. With just few modifications, all datasets are fed into PostgreSQL tables inside the bronze schema.

a. Airflow DAG Setup for Data Ingestion

The data intake procedure is coordinated by a load_to_bronze_schema_sequential Apache Airflow Directed Acyclic Graph (DAG). In order to guarantee that data from all sources is present in the Bronze layer prior to downstream operations starting, the DAG comprises jobs for loading static files first, then monthly Airbnb datasets.

The Airflow DAG consists of:

- **Static File Ingestion**: Static files (census data and LGA mappings) are downloaded and loaded from GCS to PostgreSQL in separate operations. Each job loads data into PostgreSQL using a PostgresHook and downloads files using the GCSToLocalFilesystemOperator and a custom load_to_postgres function.
- **Sequential Monthly Airbnb Ingestion**: The DAG loads each Airbnb file one after the other in order to preserve chronological order. A loop that creates a task for every month and uses dependencies between jobs to guarantee the right sequence enforces this order.

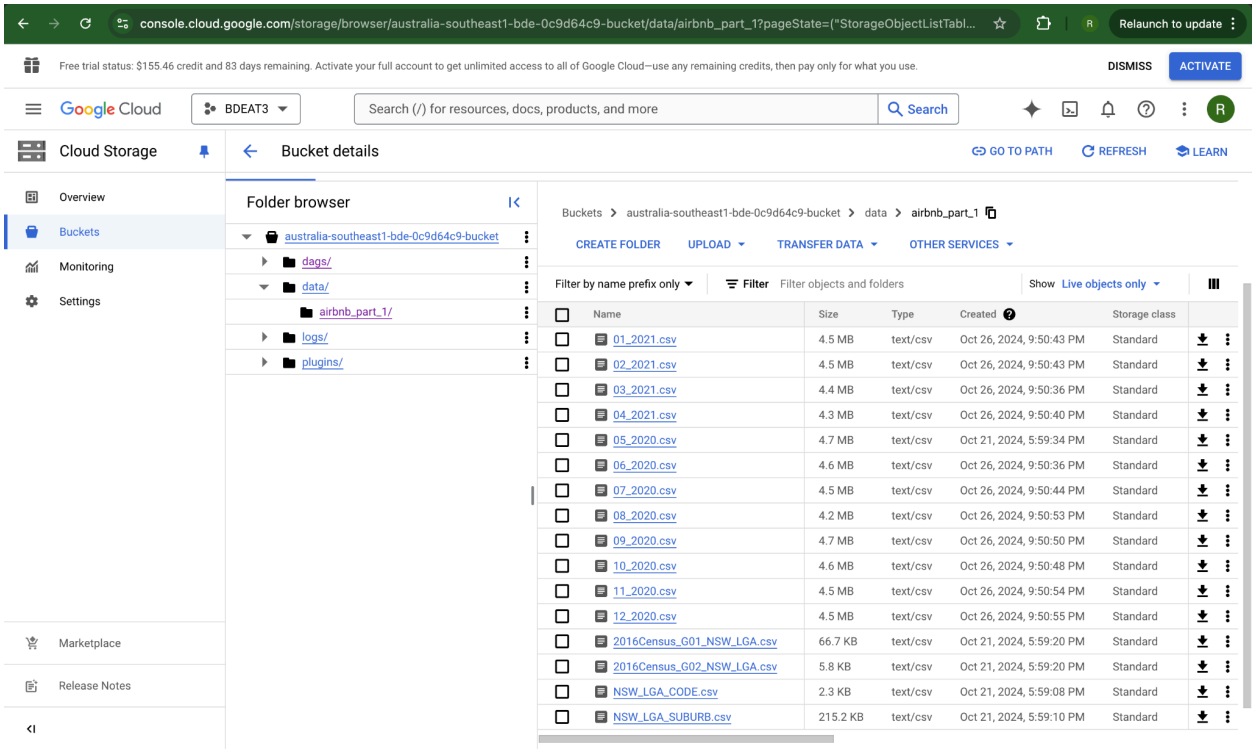


Figure 3.1 Google Cloud Services (GCS) bucket

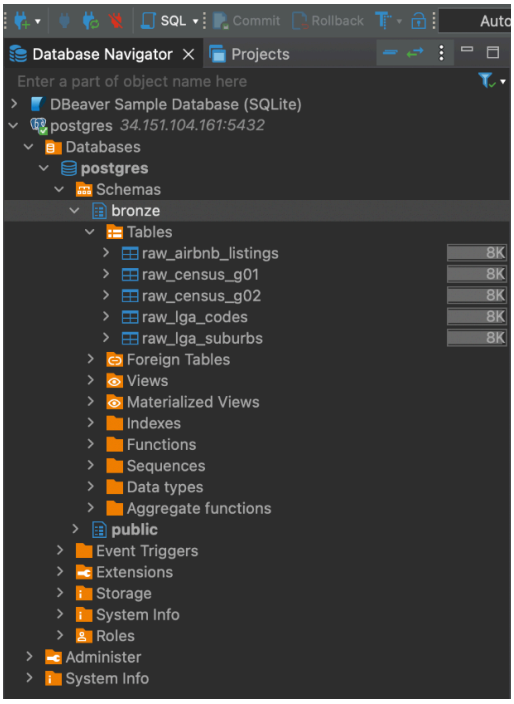


Figure 3.2 DBeaver Postgres Bronze Schema

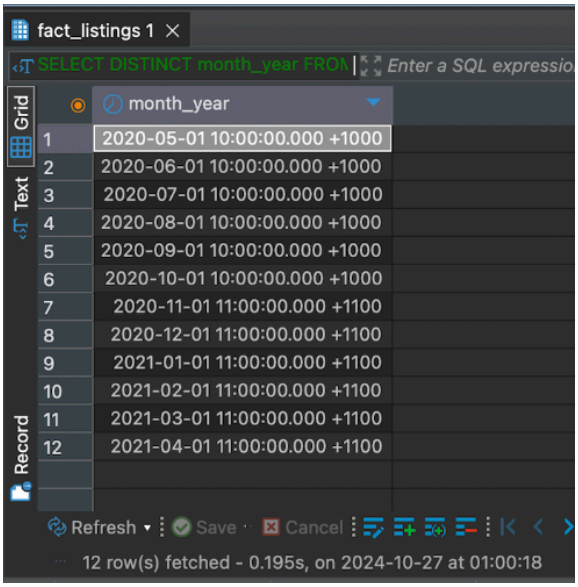


Figure 3.3 All months imported from Airflow

b. Schema and Table Design in the Bronze Layer

Tables in the Bronze schema reflect each CSV file's structure. For instance, Airbnb data with attributes like listing_id, host_name, and price is stored in raw_airbnb_listings. This table acts as a staging region, keeping any raw fields that could require cleaning or restructuring in the following levels while storing data precisely as it appears in the source.

Other tables in the Bronze schema include:

- **raw_census_g01** and **raw_census_g02**: These tables, which include variables like total_population_male, median_age_persons, and median_rent_weekly, hold demographic and socioeconomic data.
- **raw_lga_codes**: Includes LGA names and codes, making it possible to identify LGAs.
- **raw_lga_suburbs**: helps with neighborhood-level aggregations later on by mapping suburbs to their corresponding LGAs.

c. Loading Process and Data Quality Control

The load_to_postgres function manages row-wise insertion into PostgreSQL and eliminates unnecessary columns (such as unnamed columns from CSV outputs) to guarantee data quality. By allowing for precise control over the integrity of each row, this procedure guarantees that only structured data makes it to the Bronze tables.

```python
from airflow import DAG
from airflow.providers.google.cloud.transfers.gcs_to_local import GCSToLocalFilesystemOperator
from airflow.providers.postgres.hooks.postgres import PostgresHook
from airflow.operators.python_operator import PythonOperator
from airflow.models import Variable
from datetime import datetime
import pandas as pd
import requests
import logging

# Default arguments for the DAG
default_args = {
    'start_date': datetime(2024, 1, 1),
    'catchup': False,
    'email_on_failure': True,
    'email_on_retry': False,
    'retries': 2,
}

# List of Airbnb files to be processed sequentially
airbnb_files = [
    '05_2020.csv', '06_2020.csv', '07_2020.csv', '08_2020.csv',
    '09_2020.csv', '10_2020.csv', '11_2020.csv', '12_2020.csv',
    '01_2021.csv', '02_2021.csv', '03_2021.csv', '04_2021.csv'
]

# Define the DAG
with DAG(
    dag_id='load_to_bronze_schema_sequential',
    schedule_interval=None,  # No schedule interval, manually triggered
    default_args=default_args,
    description='Load raw data from GCS to Bronze schema in Postgres sequentially and trigger dbt job',
    catchup=False,
) as dag:

    # Task: Download and load static files (Census and LGA data)
    def load_to_postgres(file_path, table_name):
        postgres_hook = PostgresHook(postgres_conn_id='postgres')
        conn = postgres_hook.get_conn()
        cursor = conn.cursor()

        # Load CSV data into a Pandas DataFrame
        df = pd.read_csv(file_path)
        df = df.loc[:, ~df.columns.str.contains('^Unnamed')]  # Drop unnamed columns if any

        # Insert DataFrame rows into Postgres table
        for _, row in df.iterrows():
            placeholders = ', '.join(['%s'] * len(row))
            columns = ', '.join(df.columns)
            sql = f"INSERT INTO bronze.{table_name} ({columns}) VALUES ({placeholders})"
            cursor.execute(sql, tuple(row))

        conn.commit()
        cursor.close()
        conn.close()
```

Figure 3.4 Airflow DAG (load_to_postgres)

Airflow's logging features are used to keep an eye on job execution and debug any data ingestion problems, including missing files or corrupted rows, during the loading process.

4. Data Cleaning and Transformation (Silver Layer)

The Bronze layer's raw data is transformed into organised, analysis-ready tables by the Silver layer using data transformation and cleaning procedures. This layer makes ensuring that fields are correct, consistent, and structured for effective aggregation and joining in the Gold layer across many datasets.

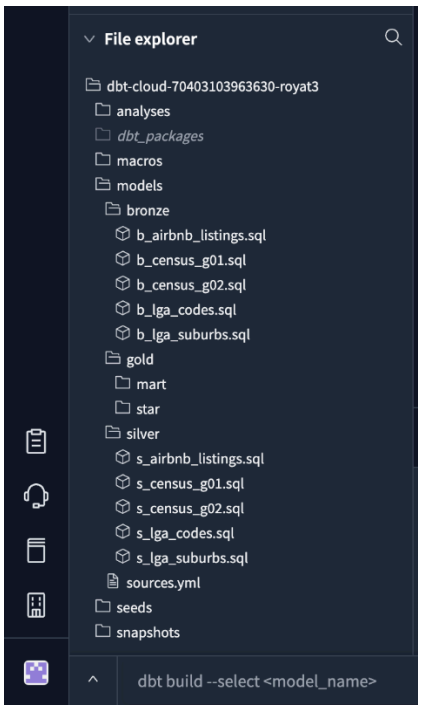


Figure 4.1 dbtcloud folder structure

a. Silver Layer Transformation Design

Every Silver layer model has a corresponding Bronze table and undergoes modifications that deal with:

- **Data Type Casting**: For computations and aggregations, fields such as IDs and prices are converted to the proper kinds (such as integer or numeric).
- **Handling Missing Values**: In fields like host_name and host_neighbourhood, null entries are swapped out for default values like "Unknown."

- **Standardizing Formats**: Ensuring that dates and category categories are formatted consistently across tables, including standardising text fields' case and converting dates to YYYY-MM-DD.
- **Validating Values**: Removing excessive or erroneous numbers, such as out-of-range review scores or negative pricing.

The dbt Cloud models created for this layer include:

- **s_airbnb_listings**: This table filters and converts data types for Airbnb listings. For example, pricing is verified to be within a suitable range, and listing_id and host_id are cast to integers. Review ratings are verified to make sure they fall within a certain range, and columns such as host_is_superhost are transformed to Boolean for precise analysis.
- **s_census_g01 and s_census_g02**: The process of cleaning census data involves converting population and socioeconomic indicators (such as total_population_male and median_rent_weekly) to integer or float types and eliminating non-numeric characters from lga_code_2016. To preserve data integrity, fields that are essential for joining, such as lga_code, must not be null.
- **s_lga_codes and s_lga_suburbs**: By removing whitespace from names and making sure that all lga_code and suburb_name values are non-null, these tables clean and format data pertaining to LGAs. When combining with other tables for geographic analysis, this ensures consistency.

b. Model Execution and Data Quality Checks

In DBT, every Silver layer model is set up to execute with data quality checks that confirm the existence and kind of important fields. Since transformations standardise the data and establish the groundwork for precise analysis in the Gold layer, data quality is essential in this tier. Validation checks on crucial fields (such as primary keys and foreign keys) guarantee that clean, structured data is accessible for additional processing while the models are run within dbt Cloud.

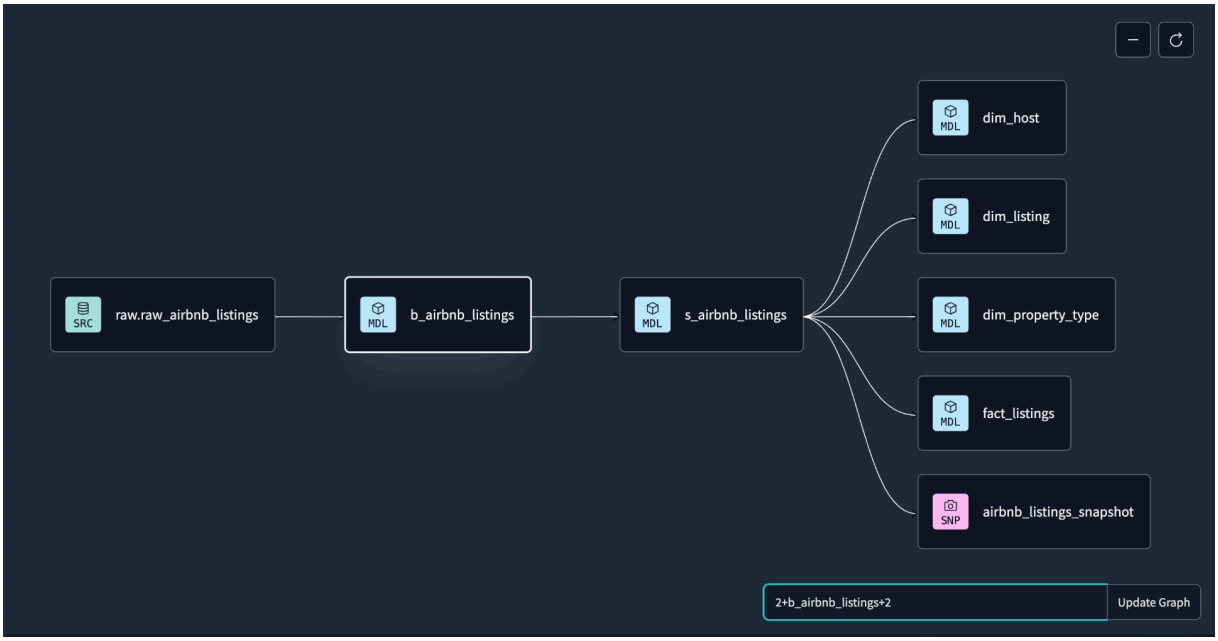


Figure 4.2 Lineage of airbnb_listings

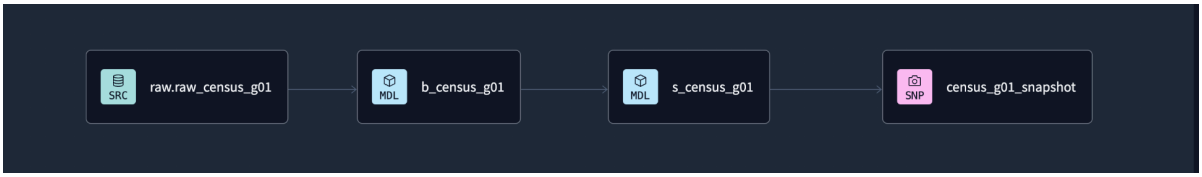


Figure 4.3 Lineage of census_g01

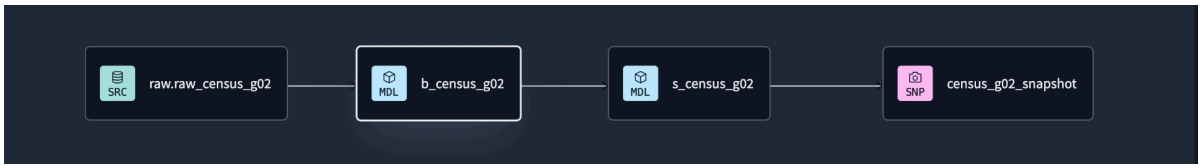


Figure 4.4 Lineage of census_g02

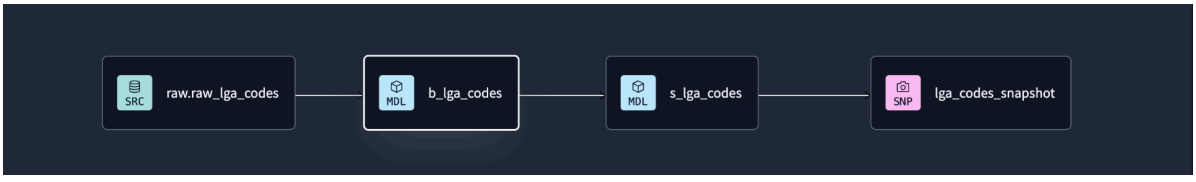


Figure 4.5 Lineage of lga_codes

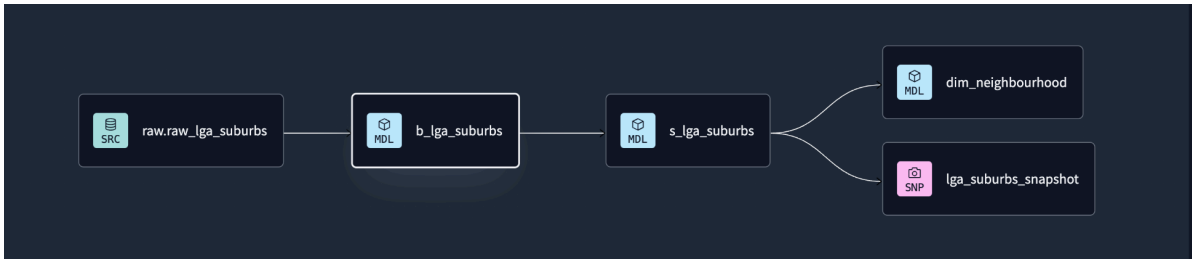


Figure 4.6 Lineage of lga_suburbs

5. Snapshot Implementation

In dbt, snapshots are crucial for monitoring changes over time, especially for variables that could vary over time, such census demographics or listing pricing. Historical analysis and comparisons are made possible by this project's implementation of snapshots on Silver layer core tables.

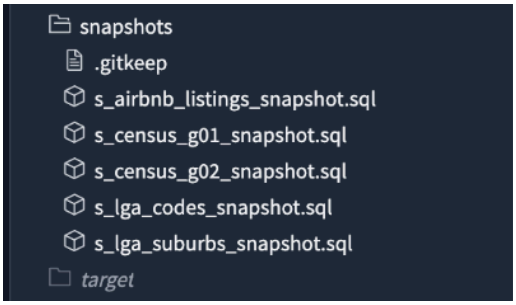


Figure 5.1 dbtcloud snapshots folder

a. Airflow DAG Setup for Data Ingestion

Strategy = 'check' is set up for each snapshot, comparing important fields (check columns) in every table to identify and document changes over time. Important snapshots consist of:

- **s_airbnb_listings_snapshot**: Monthly changes in listing parameters such as price, availability_30, number_of_reviews, and review_scores_rating are captured in this snapshot. Each record is guaranteed to belong to a specific listing by the unique key listing_id, and any modifications made to the check fields cause a new version of that listing to be recorded.

- **s_census_g01_snapshot and s_census_g02_snapshot**: Using distinct keys depending on lga_code, these snapshots document changes in the socioeconomic and demographic makeup of LGAs. To record any changes or new values over time, metrics like median_tot_hhd_inc_weekly and total_population_persons are monitored.

- **s_lga_codes_snapshot and s_lga_suburbs_snapshot**: In order to preserve any revisions to LGA names or suburb mappings for future use, these snapshots save geographic data for LGAs and suburbs. When administrative borders or naming standards change, this is crucial for longitudinal data analysis.

b. Benefits of Snapshotting

By preserving past data, snapshots enable in-depth trend analysis and comparisons across time. The project can address difficult business challenges by putting these pictures into practice, including:

- **Revenue and Price Trends**: Tracking market dynamics is made easier by monthly variations in price and expected revenue across several listings.
- **Demographic Shifts**: As census data changes, socioeconomic assessments of neighbourhoods are made possible, which helps inform judgements about Airbnb activities in certain demographic regions.

6. Star Schema Design (Gold Layer)

In order to facilitate analytics and optimise querying, the Gold layer arranges data into a star schema. This is especially useful for addressing business enquiries about demographic trends, revenue, and occupancy. With the help of dimension tables that offer descriptive qualities and fact tables that store metrics and foreign keys, this design facilitates easy aggregations and efficient joins.

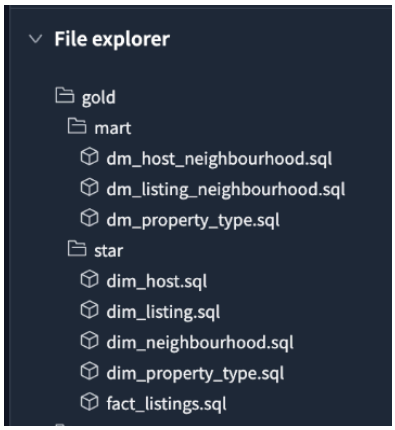


Figure 6.1 dbtcloud gold folder structure

a. Fact and Dimension Tables

The star schema in the Gold layer contains the following tables:

Fact Table: `fact_listings`:

- Key metrics and connections to other dimensions are included in this table via foreign keys like listing_id and host_id. Price, number of reviews, availability_30, number of stays, and expected income are among the metrics kept in this table.
- To enable effective analysis of monthly patterns, each row denotes a distinct listing for each month (month_year). For every listing, calculated data such as number_of_stays and estimated_revenue offer information on occupancy and revenue.

Dimension Tables:

- **dim_host**: host_id, host_name, host_neighbourhood, host_since, and host_is_superhost are among the host properties that are stored. Analysis based on host quality and involvement (e.g., superhost status) is supported by this table.
- **dim_listing**: Includes information on each listing, such as the listing's ID, neighbourhood, kind of property, type of room, and accommodations. This makes it possible to filter by listing attributes like neighbourhood and property type.
- **dim_neighbourhood**: Allows for mappings between suburb_name and lga_name, facilitating multi-level geographic analysis.
- **dim_property_type**: Includes room_type, accommodates, and property_type, which aid in classifying and evaluating postings according to the type of property.

b. Schema Optimization and Data Integrity

This layout reduces data duplication and creates a normalised structure by using foreign keys to connect the fact table to its dimensions. Consistent joins and trustworthy analytics are made possible by data types and non-null constraints, which guarantee that important values are appropriately referenced across tables.

The pipeline can handle large-scale queries with little processing time when using this star schema. This schema is perfect for the datamart views in the following section since it streamlines aggregations, filtering, and joins.

7. Datamart Views

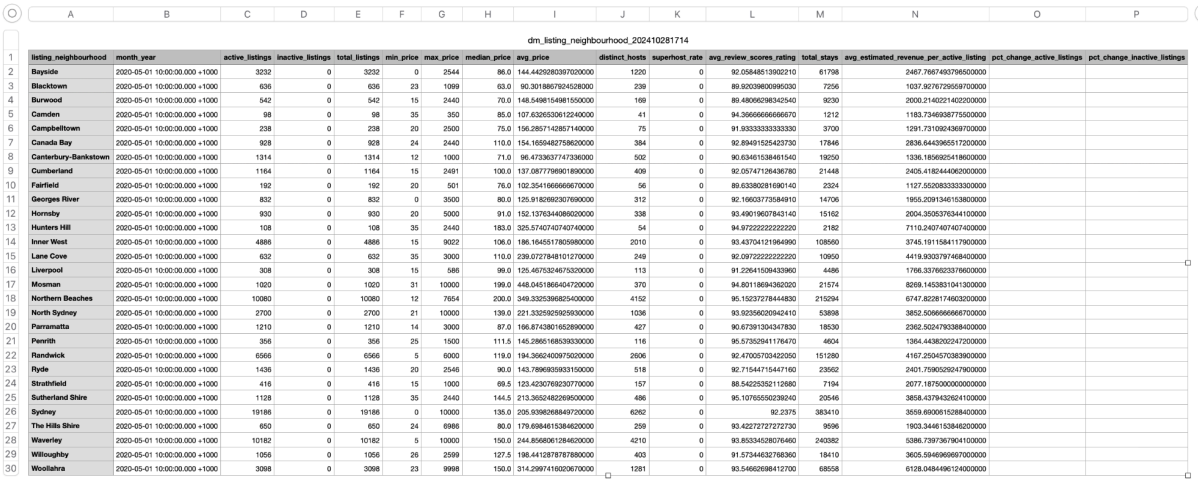
The Gold layer's datamart views offer organised insights designed to address certain business queries. Each view provides high-level metrics for decision-making by combining information from fact and dimension tables. To facilitate temporal analysis and represent Slowly Changing Dimensions Type 2, these perspectives are represented in dbt as monthly snapshots (SCD2).

a. Dm_listing_neighbourhood

Data is compiled by listing neighbourhood and month in the dm_listing_neighbourhood view. Avg_estimated_revenue_per_active_listing, estimated_revenue, distinct_hosts, and percentage changes in active and inactive listings are among the metrics it offers.

Key metrics:

- **Total Listings and Active Listings**: Monthly totals for all residential neighbourhoods' active and total listings.
- **Revenue and Host Engagement**: Total and average revenue figures, superhost rates, and unique host counts are calculated.
- **Trend Analysis**: The percentage change in active and inactive listings from month to month that enables the detection of growth and trends in the neighbourhood



dm_listing_neighbourhood_202410281714

listing_neighbourhood	month_year	active_listings	inactive_listings	total_listings	min_price	max_price	median_price	avg_price	distinct_hosts	superhost_rate	avg_review_scores_rating	total_stays	avg_estimated_revenue_per_active_listing	pct_change_active_listings	pct_change_inactive_listings
Bayside	2020-05-01 10:00:00.000 +1000	3232	0	3232	0	2544	86.0	144.44292803970200000	1220	0	92.05848513902210	61798	2467.766749379650000		
Blacktown	2020-05-01 10:00:00.000 +1000	636	0	636	23	1099	63.0	90.30188678924528000	239	0	89.92039800995030	7256	1037.927672959700000		
Burwood	2020-05-01 10:00:00.000 +1000	542	0	542	15	2440	70.0	148.54981549815500000	169	0	89.48066298342540	9230	2000.214022140220000		
Camden	2020-05-01 10:00:00.000 +1000	98	0	98	35	350	85.0	107.63265306122400000	41	0	94.36666666666670	1212	1183.734693877550000		
Campbelltown	2020-05-01 10:00:00.000 +1000	238	0	238	20	2500	75.0	156.28571428571400000	75	0	91.93333333333330	3700	1291.731092436970000		
Canada Bay	2020-05-01 10:00:00.000 +1000	928	0	928	24	2440	110.0	154.16594827586200000	384	0	92.89491525423730	17846	2836.644396551720000		
Canterbury-Bankstown	2020-05-01 10:00:00.000 +1000	1314	0	1314	12	1000	71.0	96.47363774736100000	502	0	90.63461538461540	19250	1336.185692541860000		
Cumberland	2020-05-01 10:00:00.000 +1000	1164	0	1164	15	2491	100.0	137.08777989018900000	409	0	92.05747126436780	21448	2405.418244406200000		
Fairfield	2020-05-01 10:00:00.000 +1000	192	0	192	20	501	76.0	102.35416666666700000	56	0	89.63380281690140	2324	1127.552083333330000		
Georges River	2020-05-01 10:00:00.000 +1000	832	0	832	0	3500	80.0	125.91826923076900000	312	0	92.16603773584910	14706	1955.209134615380000		
Hornsby	2020-05-01 10:00:00.000 +1000	930	0	930	20	5000	91.0	152.13763440860200000	338	0	93.49019607843140	15162	2004.350537634410000		
Hunters Hill	2020-05-01 10:00:00.000 +1000	108	0	108	35	2440	183.0	325.57407407407400000	54	0	94.97222222222220	2182	7110.240740740740000		
Inner West	2020-05-01 10:00:00.000 +1000	4886	0	4886	15	9022	106.0	186.16455178859800000	2010	0	93.43704121964990	108560	3745.191158411790000		
Lane Cove	2020-05-01 10:00:00.000 +1000	632	0	632	35	3000	110.0	239.07278481012700000	249	0	92.09722222222220	10950	4419.930379746840000		
Liverpool	2020-05-01 10:00:00.000 +1000	308	0	308	15	586	99.0	125.46753246753200000	113	0	91.22641509433960	4486	1766.337662337660000		
Mosman	2020-05-01 10:00:00.000 +1000	1020	0	1020	31	10000	199.0	448.04518664047200000	370	0	94.80118694362020	21574	8269.145383104130000		
Northern Beaches	2020-05-01 10:00:00.000 +1000	10080	0	10080	12	7654	200.0	349.33253968254000000	4152	0	95.15237278444830	215294	6747.822817460320000		
North Sydney	2020-05-01 10:00:00.000 +1000	2700	0	2700	21	10000	139.0	221.33259259259300000	1036	0	93.92356020942410	53898	3852.506666666670000		
Parramatta	2020-05-01 10:00:00.000 +1000	1210	0	1210	14	3000	87.0	166.87438016528900000	427	0	90.67391304347830	18530	2362.502479338840000		
Penrith	2020-05-01 10:00:00.000 +1000	356	0	356	25	1500	111.5	145.28651685393300000	116	0	95.57352941176470	4604	1364.443820224720000		
Randwick	2020-05-01 10:00:00.000 +1000	6566	0	6566	5	6000	119.0	194.36624009750200000	2606	0	92.47005703422050	151280	4167.250457033839000		
Ryde	2020-05-01 10:00:00.000 +1000	1436	0	1436	20	2546	90.0	143.78969359331500000	518	0	92.71544715447160	23562	2401.759052924790000		
Strathfield	2020-05-01 10:00:00.000 +1000	416	0	416	15	1000	69.5	123.42307692307700000	157	0	88.54225352112680	7194	2077.187500000000000		
Sutherland Shire	2020-05-01 10:00:00.000 +1000	1128	0	1128	35	2440	144.5	213.36524822695000000	486	0	95.10765550239240	20546	3858.437943262410000		
Sydney	2020-05-01 10:00:00.000 +1000	19186	0	19186	0	10000	135.0	205.93982688497200000	6262	0	92.2375	383410	3559.690061528840000		
The Hills Shire	2020-05-01 10:00:00.000 +1000	650	0	650	24	6986	80.0	179.69846153846200000	259	0	93.42272727272730	9596	1903.344615384620000		
Waverley	2020-05-01 10:00:00.000 +1000	10182	0	10182	5	10000	150.0	244.85680612846200000	4210	0	93.85334528076460	240382	5386.739736790410000		
Willoughby	2020-05-01 10:00:00.000 +1000	1056	0	1056	26	2599	127.5	198.44128787878800000	403	0	91.57344632768360	18410	3605.594696969700000		
Woollahra	2020-05-01 10:00:00.000 +1000	3098	0	3098	23	9998	150.0	314.29974160206700000	1281	0	93.54662698412700	68558	6128.048449612400000		

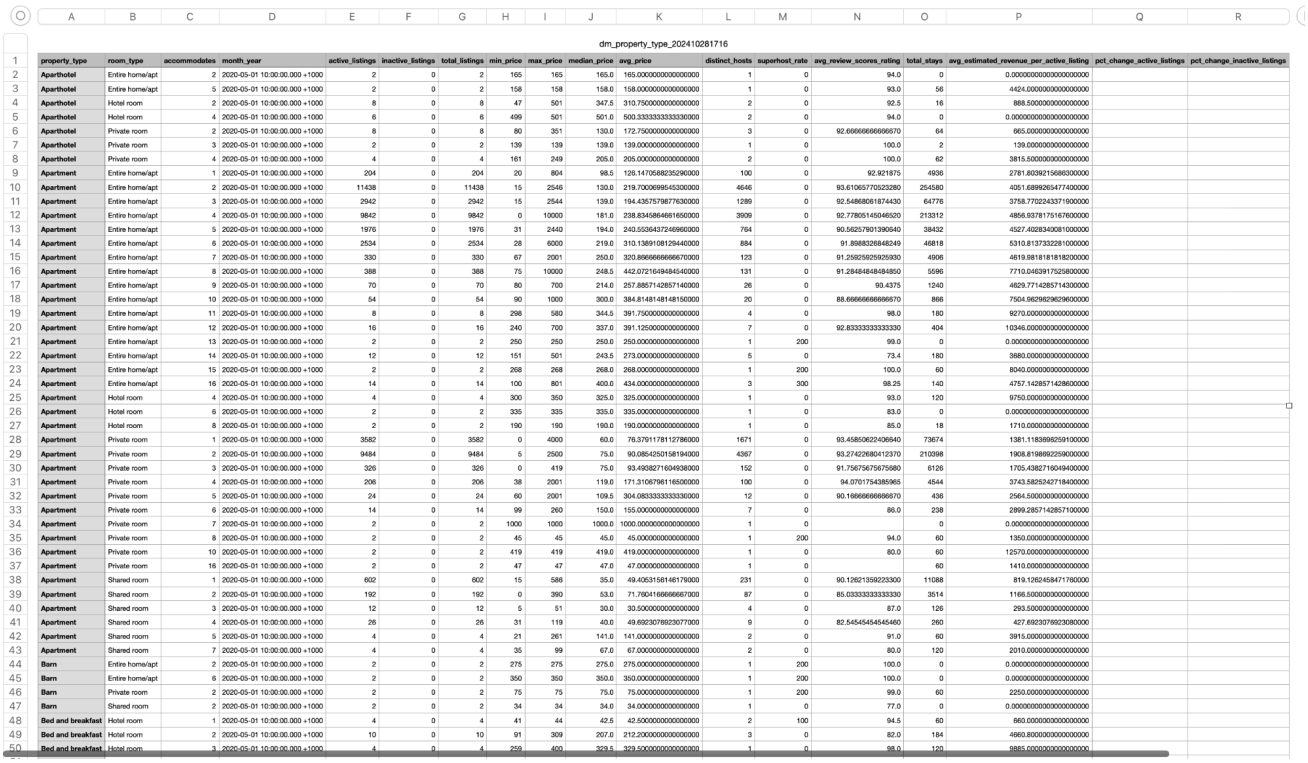
Figure 7.1 dm_listing_neighbourhood table csv

b. Dm_property_type

The dm_property_type view provides information on listing performance based on property attributes by classifying data by property_type, room_type, accommodates, and month_year.

Key metrics:

- **Price and Occupancy Trends**: Contains the projected income for current listings, the total number of stays, and the minimum, maximum, median, and average prices.
- **Host and Review Metrics**: Provides a quality indicator for every kind of property by displaying the average review ratings, distinct host count, and superhost rate.
- **Trend Analysis**: Shows variations in market demand through monthly percentage changes in active and inactive listings by type of property.



dm_property_type_202410281716

property_type	room_type	accommodates	month_year	active_listings	inactive_listings	total_listings	min_price	max_price	median_price	avg_price	distinct_hosts	superhost_rate	avg_review_scores_rating	total_stays	avg_estimated_revenue_per_active_listing	pct_change_active_listings	pct_change_inactive_listings
Aparthotel	Entire home/apt	2	2020-05-01 10:00:00.000 +1000	2	0	2	165	165	165.0	165.000000000000000	1	0	94.0	0	0.000000000000000000		
Aparthotel	Entire home/apt	5	2020-05-01 10:00:00.000 +1000	2	0	2	158	158	158.0	158.000000000000000	1	0	93.0	56	4424.000000000000000		
Aparthotel	Hotel room	2	2020-05-01 10:00:00.000 +1000	8	0	8	47	501	347.5	310.750000000000000	2	0	92.5	16	888.500000000000000		
Aparthotel	Hotel room	4	2020-05-01 10:00:00.000 +1000	6	0	6	499	501	501.0	500.333333333333300	2	0	94.0	0	0.000000000000000000		
Aparthotel	Private room	2	2020-05-01 10:00:00.000 +1000	8	0	8	80	351	130.0	172.750000000000000	3	0	92.6666666666670	64	665.000000000000000		
Aparthotel	Private room	3	2020-05-01 10:00:00.000 +1000	2	0	2	139	139	139.0	139.000000000000000	1	0	100.0	2	139.000000000000000		
Aparthotel	Private room	4	2020-05-01 10:00:00.000 +1000	4	0	4	161	249	205.0	205.000000000000000	2	0	100.0	62	3815.500000000000000		
Apartment	Entire home/apt	1	2020-05-01 10:00:00.000 +1000	204	0	204	20	804	98.5	126.147058823529400	100	0	92.921875	4936	2781.803921568630000		
Apartment	Entire home/apt	2	2020-05-01 10:00:00.000 +1000	11438	0	11438	15	2546	130.0	219.700069954530000	4646	0	93.6106577052328 0	254580	4051.689926547740000		
Apartment	Entire home/apt	3	2020-05-01 10:00:00.000 +1000	2942	0	2942	15	2544	139.0	194.435757977630000	1289	0	92.5486861874430	64776	3758.770224337190000		
Apartment	Entire home/apt	4	2020-05-01 10:00:00.000 +1000	9842	0	9842	0	10000	181.0	238.834586466165000	3909	0	92.7780514504652 0	213312	4856.937817516760000		
Apartment	Entire home/apt	5	2020-05-01 10:00:00.000 +1000	1976	0	1976	31	2440	194.0	240.553643724696000	764	0	90.5625790139064 0	38432	4527.402834008100000		
Apartment	Entire home/apt	6	2020-05-01 10:00:00.000 +1000	2534	0	2534	28	6000	219.0	310.138910812944000	884	0	91.8988326848249	46818	5310.813733228100000		
Apartment	Entire home/apt	7	2020-05-01 10:00:00.000 +1000	330	0	330	67	2001	250.0	320.866666666667000	123	0	91.2592582592593 0	4906	4619.981818181820000		
Apartment	Entire home/apt	8	2020-05-01 10:00:00.000 +1000	388	0	388	75	10000	248.5	442.072164948454400	131	0	91.2848484848485 0	5596	7710.046391752580000		
Apartment	Entire home/apt	9	2020-05-01 10:00:00.000 +1000	70	0	70	80	700	214.0	257.885714285714000	26	0	90.4375	1240	4629.771428571430000		
Apartment	Entire home/apt	10	2020-05-01 10:00:00.000 +1000	54	0	54	90	1000	300.0	384.814814814815000	20	0	88.6666666666670	866	7504.962962962960000		
Apartment	Entire home/apt	11	2020-05-01 10:00:00.000 +1000	8	0	8	298	580	344.5	391.750000000000000	4	0	98.0	180	9270.000000000000000		
Apartment	Entire home/apt	12	2020-05-01 10:00:00.000 +1000	16	0	16	240	700	337.0	391.125000000000000	7	0	92.8333333333330	404	10346.000000000000000		
Apartment	Entire home/apt	13	2020-05-01 10:00:00.000 +1000	2	0	2	250	250	250.0	250.000000000000000	1	200	99.0	0	0.000000000000000000		
Apartment	Entire home/apt	14	2020-05-01 10:00:00.000 +1000	12	0	12	151	501	243.5	273.000000000000000	5	0	73.4	180	3680.000000000000000		
Apartment	Entire home/apt	15	2020-05-01 10:00:00.000 +1000	2	0	2	268	268	268.0	268.000000000000000	1	200	100.0	60	8040.000000000000000		
Apartment	Entire home/apt	16	2020-05-01 10:00:00.000 +1000	14	0	14	100	801	400.0	434.000000000000000	3	300	98.25	140	4757.142857142860000		
Apartment	Hotel room	4	2020-05-01 10:00:00.000 +1000	4	0	4	300	350	325.0	325.000000000000000	1	0	93.0	120	9750.000000000000000		
Apartment	Hotel room	6	2020-05-01 10:00:00.000 +1000	2	0	2	335	335	335.0	335.000000000000000	1	0	83.0	0	0.000000000000000000		
Apartment	Hotel room	8	2020-05-01 10:00:00.000 +1000	2	0	2	190	190	190.0	190.000000000000000	1	0	85.0	18	1710.000000000000000		
Apartment	Private room	1	2020-05-01 10:00:00.000 +1000	3582	0	3582	0	4000	60.0	76.3791178112786000	1671	0	93.4585062240664 0	73674	1381.118399625910000		
Apartment	Private room	2	2020-05-01 10:00:00.000 +1000	9484	0	9484	5	2500	75.0	90.0854250158194000	4367	0	93.2742268041237 0	210398	1908.819889225900000		
Apartment	Private room	3	2020-05-01 10:00:00.000 +1000	326	0	326	0	419	75.0	93.4938271604938000	152	0	91.7567567567568 0	6126	1705.438271604940000		
Apartment	Private room	4	2020-05-01 10:00:00.000 +1000	206	0	206	38	2001	119.0	171.310679611650000	100	0	94.0701754385965	4544	3743.582524271840000		
Apartment	Private room	5	2020-05-01 10:00:00.000 +1000	24	0	24	60	2001	109.5	304.083333333333000	12	0	90.1666666666670	436	2564.500000000000000		
Apartment	Private room	6	2020-05-01 10:00:00.000 +1000	14	0	14	99	260	150.0	155.000000000000000	7	0	86.0	238	2899.285714285710000		
Apartment	Private room	7	2020-05-01 10:00:00.000 +1000	2	0	2	1000	1000	1000.0	1000.000000000000000	1	0		0	0.000000000000000000		
Apartment	Private room	8	2020-05-01 10:00:00.000 +1000	2	0	2	45	45	45.0	45.000000000000000	1	200	94.0	60	1350.000000000000000		
Apartment	Private room	10	2020-05-01 10:00:00.000 +1000	2	0	2	419	419	419.0	419.000000000000000	1	0	80.0	60	12570.000000000000000		
Apartment	Private room	16	2020-05-01 10:00:00.000 +1000	2	0	2	47	47	47.0	47.000000000000000	1	0		60	1410.000000000000000		
Apartment	Shared room	1	2020-05-01 10:00:00.000 +1000	602	0	602	15	586	35.0	49.4053156146179000	231	0	90.1262135922330	11088	819.126245847176000		
Apartment	Shared room	2	2020-05-01 10:00:00.000 +1000	192	0	192	0	390	53.0	71.7604166666667000	87	0	85.0333333333330	3514	1166.500000000000000		
Apartment	Shared room	3	2020-05-01 10:00:00.000 +1000	12	0	12	5	51	30.0	30.5000000000000000	4	0	87.0	126	293.500000000000000		
Apartment	Shared room	4	2020-05-01 10:00:00.000 +1000	26	0	26	31	119	40.0	49.6923076923077000	9	0	82.5454545454546 0	260	427.692307692308000		
Apartment	Shared room	5	2020-05-01 10:00:00.000 +1000	4	0	4	21	261	141.0	141.000000000000000	2	0	91.0	60	3915.000000000000000		
Apartment	Shared room	7	2020-05-01 10:00:00.000 +1000	4	0	4	35	99	67.0	67.0000000000000000	2	0	80.0	120	2010.000000000000000		
Barn	Entire home/apt	2	2020-05-01 10:00:00.000 +1000	2	0	2	275	275	275.0	275.000000000000000	1	200	100.0	0	0.000000000000000000		
Barn	Entire home/apt	6	2020-05-01 10:00:00.000 +1000	2	0	2	350	350	350.0	350.000000000000000	1	200	100.0	0	0.000000000000000000		
Barn	Private room	2	2020-05-01 10:00:00.000 +1000	2	0	2	75	75	75.0	75.0000000000000000	1	200	99.0	60	2250.000000000000000		
Barn	Shared room	2	2020-05-01 10:00:00.000 +1000	2	0	2	34	34	34.0	34.0000000000000000	1	0	77.0	0	0.000000000000000000		
Bed and breakfast	Hotel room	1	2020-05-01 10:00:00.000 +1000	4	0	4	41	44	42.5	42.5000000000000000	2	100	94.5	60	660.000000000000000		
Bed and breakfast	Hotel room	2	2020-05-01 10:00:00.000 +1000	10	0	10	91	309	207.0	212.200000000000000	3	0	82.0	184	4960.800000000000000		
Bed and breakfast	Hotel room	3	2020-05-01 10:00:00.000 +1000	4	0	4	259	400	329.5	329.500000000000000	1	0	98.0	120	9885.000000000000000		

Figure 7.2 dm_property_type table csv

c. Dm_host_neighbourhood

The datamart views answer several key business questions:

- **Revenue and Price Trends**: Stakeholders can monitor the most lucrative neighbourhoods or property types by looking at average price and average predicted revenue per active listing.
- **Host Engagement**: Particularly in areas with high visitor demand, the distinct_hosts counts and superhost_rate statistic provide evaluation of host quality and activity.
- **Occupancy and Demand**: Monitoring shifts in active_listings and inactive_listings reveals possible growth regions and offers insights into occupancy trends.

With these perspectives, the Gold layer provides useful information about Airbnb listings, allowing for data-driven choices to increase income and enhance visitor pleasure.

dm_host_neighbourhood_202410281712

host_neighbourhood_lga	month_year	distinct_hosts	estimated_revenue	estimated_revenue_per_host
ARMIDALE REGIONAL	2020-05-01 10:00:00.000 +1000	1	3600	3600.0000000000000000
BATHURST REGIONAL	2020-05-01 10:00:00.000 +1000	10	376354	37635.400000000000
BAYSIDE	2020-05-01 10:00:00.000 +1000	43	237370	5520.2325581395300000
BAYSIDE	2020-05-01 10:00:00.000 +1000	24	150062	6252.5833333333300000
BAYSIDE	2020-05-01 10:00:00.000 +1000	9	40800	4533.3333333333300000
BAYSIDE	2020-05-01 10:00:00.000 +1000	108	1225390	11346.203703703700
BAYSIDE	2020-05-01 10:00:00.000 +1000	27	328704	12174.222222222200
BAYSIDE	2020-05-01 10:00:00.000 +1000	37	155584	4204.9729729729700000
BAYSIDE	2020-05-01 10:00:00.000 +1000	121	1031510	8524.8760330578500000
BAYSIDE	2020-05-01 10:00:00.000 +1000	55	343314	6242.0727272727300000
BAYSIDE	2020-05-01 10:00:00.000 +1000	44	315812	7177.5454545454500000
BAYSIDE	2020-05-01 10:00:00.000 +1000	4	14940	3735.0000000000000000
BAYSIDE	2020-05-01 10:00:00.000 +1000	7	48450	6921.4285714285700000
BAYSIDE	2020-05-01 10:00:00.000 +1000	23	89588	3895.1304347826100000
BAYSIDE	2020-05-01 10:00:00.000 +1000	2	9232	4616.0000000000000000
BAYSIDE	2020-05-01 10:00:00.000 +1000	4	18600	4650.0000000000000000
BAYSIDE	2020-05-01 10:00:00.000 +1000	11	74910	6810.0000000000000000
BAYSIDE	2020-05-01 10:00:00.000 +1000	41	309162	7540.5365853658500000
BAYSIDE	2020-05-01 10:00:00.000 +1000	92	516034	5609.0652173913000000
BAYSIDE	2020-05-01 10:00:00.000 +1000	15	916970	61131.333333333300
BAYSIDE	2020-05-01 10:00:00.000 +1000	12	29566	2463.8333333333300000
BAYSIDE	2020-05-01 10:00:00.000 +1000	25	328680	13147.200000000000
BAYSIDE	2020-05-01 10:00:00.000 +1000	173	1360854	7866.2080924855500000
BAYSIDE	2020-05-01 10:00:00.000 +1000	3	10912	3637.3333333333300000
BAYSIDE	2020-05-01 10:00:00.000 +1000	2	39300	19650.000000000000
BURWOOD	2020-05-01 10:00:00.000 +1000	5	8580	1716.0000000000000000
BURWOOD	2020-05-01 10:00:00.000 +1000	13	71982	5537.0769230769200000
BURWOOD	2020-05-01 10:00:00.000 +1000	29	149574	5157.7241379310300000
BURWOOD	2020-05-01 10:00:00.000 +1000	57	418434	7340.9473684210500000

Figure 7.3 dm_host_neighbourhood table csv

8. End-to-End Orchestration (Airflow DAG)

The automated coordination of data intake, transformation, and loading using Apache Airflow is a crucial component of this project's design. From raw intake to analytics-ready views, data flows effectively thanks to our end-to-end orchestration, which guarantees a dependable and seamless data pipeline.

a. DAG Design and Sequential Data Processing

The following objectives guide the coordination of data input and transformation via the Airflow DAG, load_to_bronze_schema_sequential:

- **Chronological Data Loading**: To guarantee that each month's data is handled in the correct sequence, the DAG is set up to load Airbnb monthly data files sequentially. Maintaining chronological integrity is crucial, particularly for time-based metrics like number_of_reviews and monthly pricing.
- **Static Data Handling**: First, static datasets are loaded, such as census and LGA mappings. This information serves as the basis for joining and aggregations, which enable listings to be precisely linked to administrative areas and demographic data.

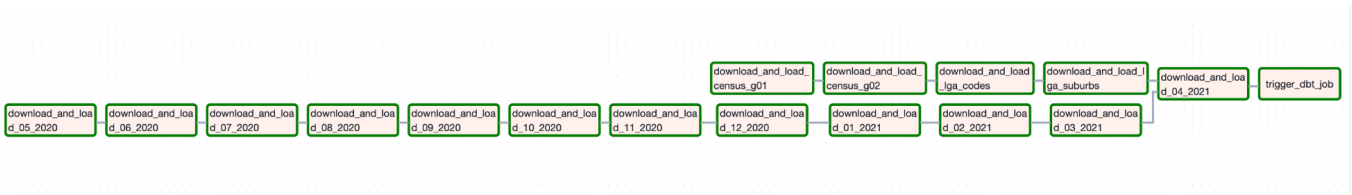


Figure 8.1 Airflow DAG execution of all monthly listings data

b. Triggering dbt Transformations

The DAG initiates a dbt Cloud task to carry out transformations when data loading is finished. By automating the data transfer from the Bronze layer (raw ingestion) to the Silver and Gold levels, this connection with dbt Cloud guarantees fast and reliable updates for all models.

- **API-Based Task**: The DAG pulls the required credentials from Airflow Variables for security and uses a PythonOperator to initiate the dbt Cloud task through an API request. Strong error handling is provided by this method, which logs any failures and raises exceptions to alert the team to problems.
- **Dependencies and Data Integrity**: The DAG guarantees that data is modified only when ingestion is finished, preserving data integrity throughout the pipeline, by imposing dependencies between jobs.

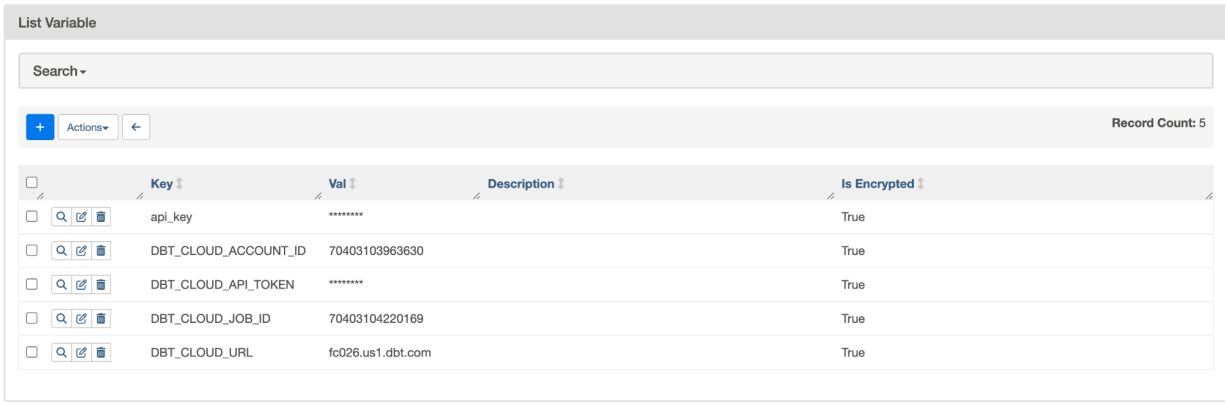


Figure 8.2 Airflow Variables for API trigger (dbtcloud)

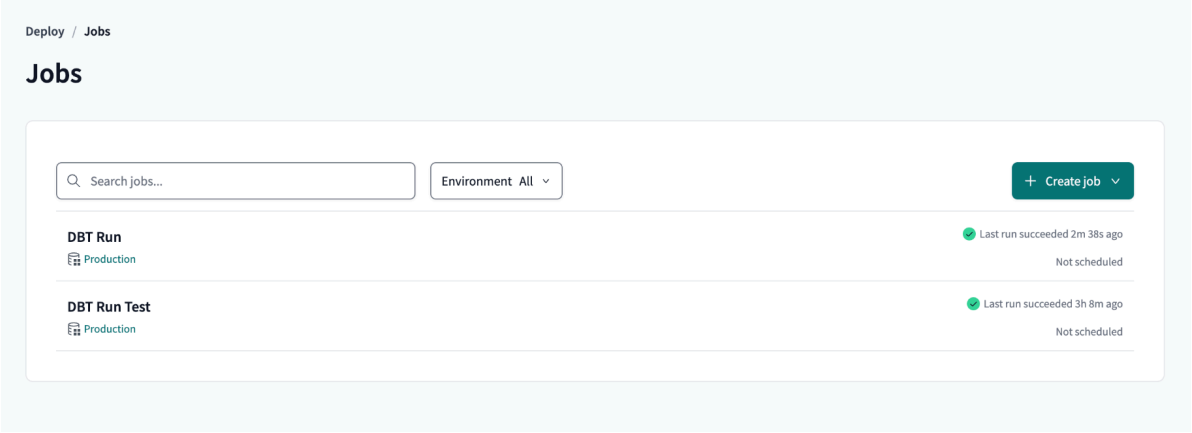


Figure 8.3 dbtcloud jobs

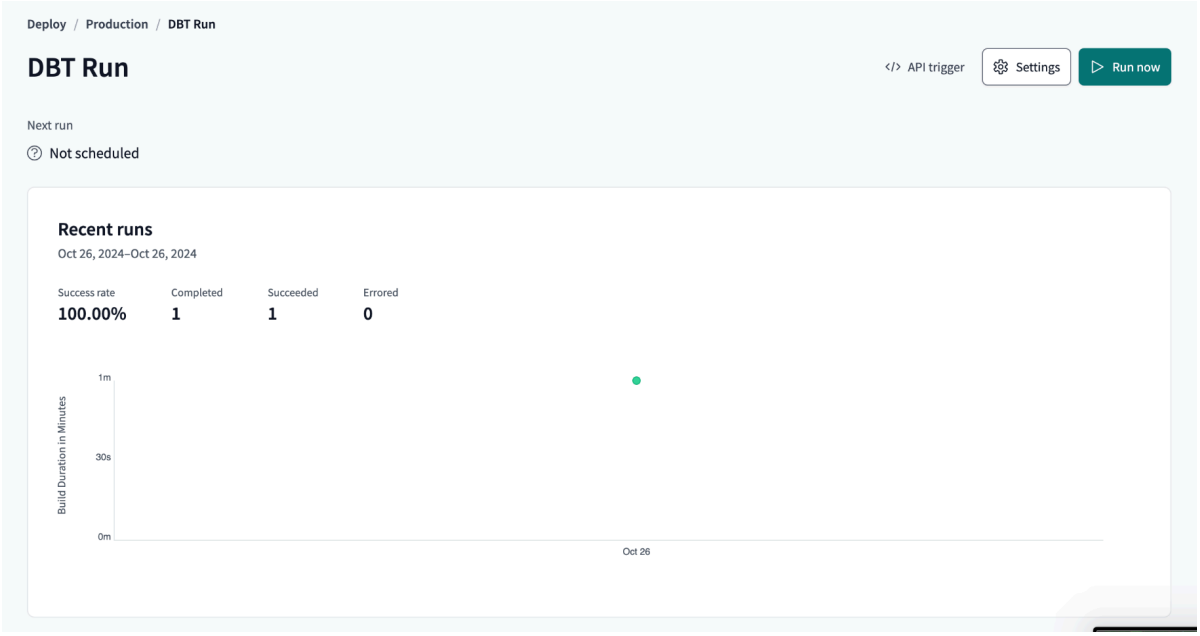


Figure 8.4 Successful dbtcloud job run

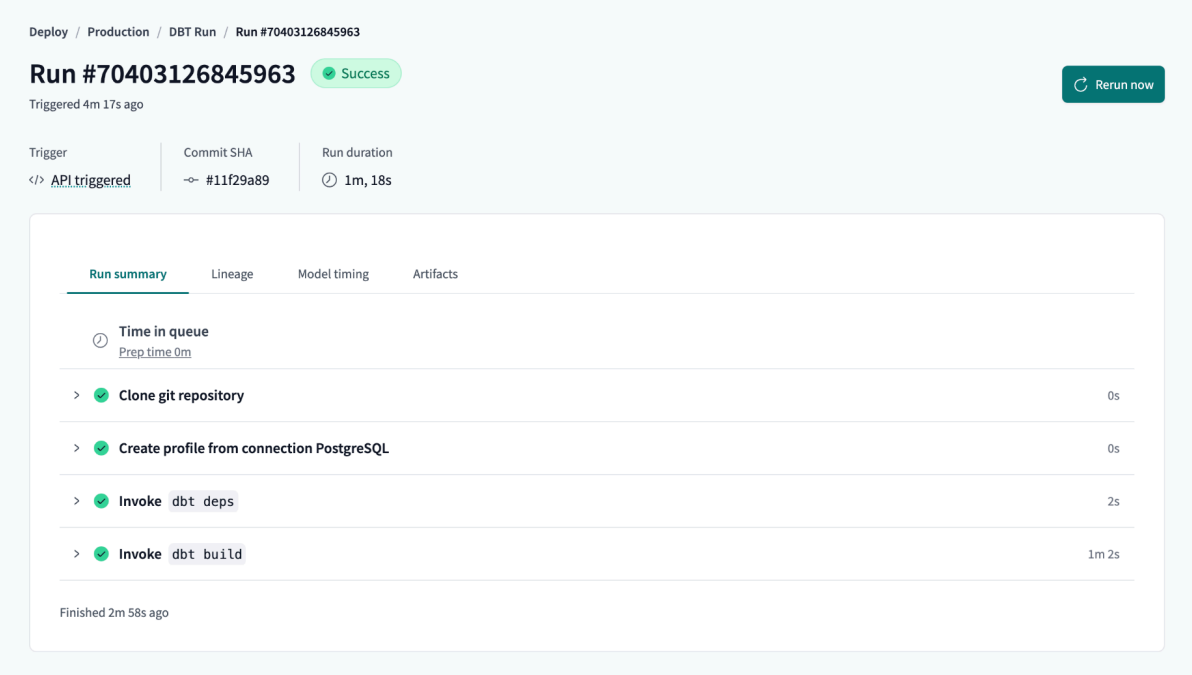


Figure 8.5 dbtcloud job run results

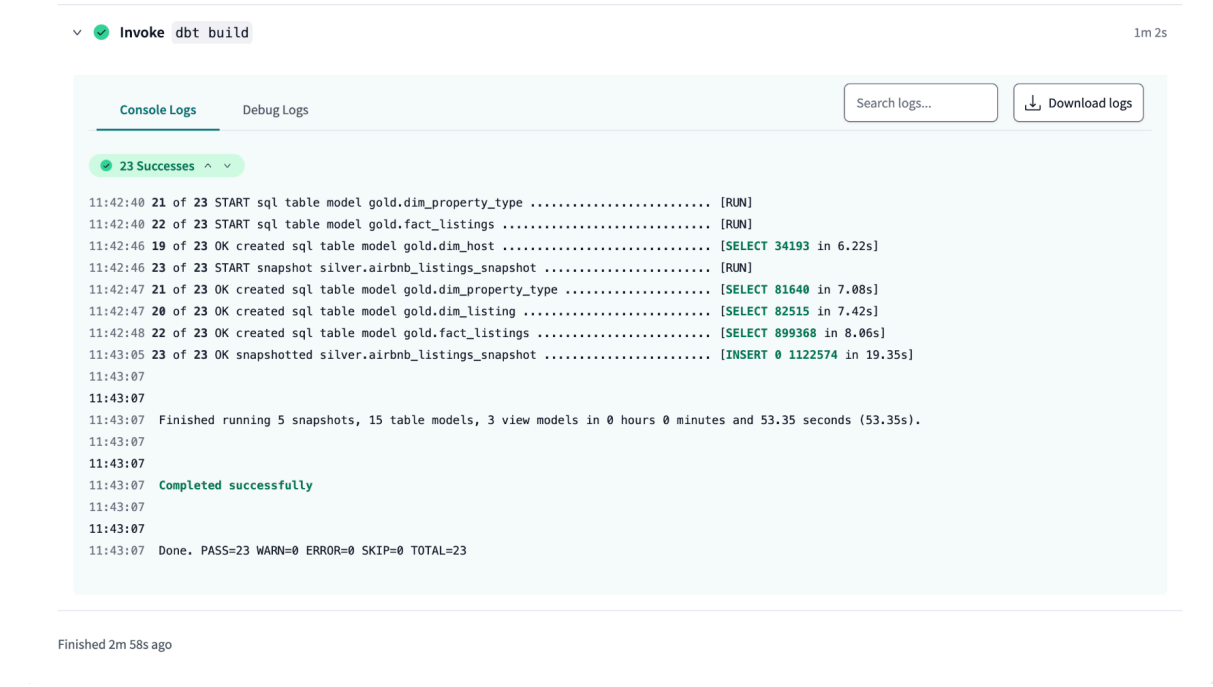


Figure 8.6 Logs of dbtcloud trigger

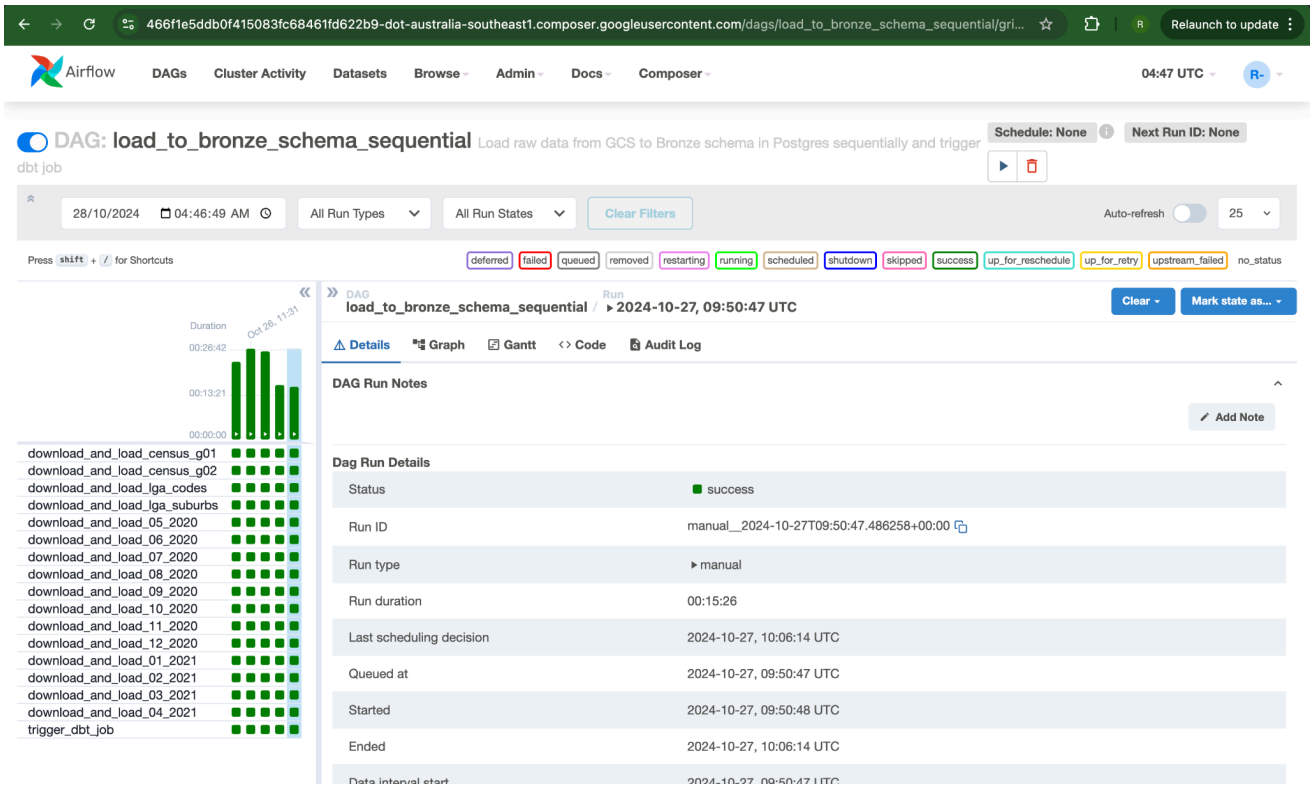


Figure 8.7 Airflow DAG status

c. Monitoring and Error Handling

The way the Airflow DAG is set up to deal with faults is by:

- **Retries and Alerts**: To make sure that team members are informed in the event of problems, the DAG's default_args setting incorporates email notifications upon failure as well as retries for temporary failures.
- **Logging**: Every job keeps track of its progress, giving insight into the transformation and loading procedures. Especially during data import, this recording is useful for pipeline monitoring and problem diagnosis.

By minimising manual involvement and guaranteeing that all transformations and views are promptly updated for precise analysis, this automated orchestration streamlines the data flow.

9. Challenges and Solutions

This project encountered several technical challenges, each addressed with strategic solutions to ensure data integrity, automation, and performance.

a. Sequential Data Loading and Order Integrity

- **Challenge:** Maintaining chronological correctness in the dataset, particularly for time-dependent studies, required that Airbnb data be fed sequentially per month.
- **Solution:** To enforce a rigorous sequence, the Airflow DAG was constructed with dependencies between monthly data loading operations. Processing of the data file for each month didn't start until the preceding file was completely loaded. The integrity of time-based measurements and trends was preserved by this sequential processing.

b. Data Consistency and Type Validation

- **Challenge:** A number of data types and certain inconsistent or missing values (such as wrong dates or negative prices) were included in the raw Airbnb dataset, which may have affected analysis and transformations.
- **Solution:** For fields like price, review_scores_rating, and availability_30, the Silver layer changes used range checks, null handling, and casting. In order to maintain a clean and consistent dataset for analysis, dbt tests were also included to check data types and make sure needed fields weren't null.

c. Integration of Airflow and dbt Cloud

- **Challenge:** To integrate Airflow with dbt Cloud in a secure and dependable manner so that dbt transformations would start instantly after data loading.
- **Solution:** via a PythonOperator and dbt credentials safely kept in Airflow Variables, Airflow initiated dbt Cloud tasks via an API-based methodology. This method made sure that the dbt transformations only executed once the data intake process was successfully finished.

d. Data Integrity in Slowly Changing Dimensions (SCD2)

- **Challenge:** Some tables, such Airbnb listings and census demographics, required historical monitoring in order to be analysed over time.
- **Solution:** SCD Type 2 was set up to record historical changes in particular fields in dbt snapshots. As a result, precise longitudinal analysis was made possible, supporting time-based patterns and demographic shifts in the data with historical context.

10. Conclusion

Using Google Cloud Storage, PostgreSQL, Apache Airflow, and dbt Cloud, this project effectively developed a fully coordinated data pipeline that was organised in accordance with the Medallion Architecture (Bronze, Silver, Gold). The pipeline ensures a stable flow of data from raw input to analytical insights by utilising an effective, tiered design that facilitates automated data ingestion, transformation, and aggregation.

Datamart views and the star schema in the Gold layer offer useful information about neighbourhood demographics and Airbnb listings. Targeted measurements and optimised data formats efficiently address business issues about occupancy trends, revenue trends, and property performance.

The project is positioned as a scalable solution for continuous Airbnb and demographic data analysis because of its automation, data validation, and historical recording, which allow for accurate, repeatable analysis. Future improvements, like adding more years to the dataset or including other pertinent data sources, may be built upon the strong foundation this pipeline offers.