25 OCTOBER 2024

DATA PIPELINE WITH AIRFLOW AND DBT

NICHOLAS (THANH NHAN) LE

Table of Contents

1 Project Summary

This project aims to build a production-ready pipeline for ingesting, transforming and storing 12 months of Airbnb data using the Extract-Load-Transform (ELT) integration framework. It also combines NSW census data to answer a range of business questions related to hosts' listing performance, demographic information at the local government area (LGA) level, and financing health.

To this end, raw listings and census data are extracted from source systems, and the extraction process assumes a monthly schedule in order to implement Type-2 Slowly Changing Dimensions (SCD-2) as new data arrives.

2 Data Summary

The Airbnb dataset for analysis [1] contains a year of listings information from May 2020 to April 2021 for NSW, including:

- Listings: neighbourhood, property type/size, room type, active listing status
- Hosts: name, tenure, neighbourhood, and metrics (is superhost, 30-day availability, number of reviews, review scores and ratings)

The Census dataset [2] from the Australian Bureau of Statistics contains the 2016 snapshot of NSW's demographic information at the LGA level, including gender counts on:

- Population counts
- Population at different age groups
- Indigenous status: Aboriginal, Torres Strait Islander, or both
- Citizenship status: whether respondents are Australian citizens, and whether they were born in Australia or overseas
- Language: English or non-English speakers at home
- Education status by age group
- Private or other dwellings

There is also a referential dataset on LGA and suburb definitions, whose purpose is for joining information in the above datasets.

3 Google Cloud, Apache Airflow, and dbt Setup

3.1 Set up Apache Airflow in Google Cloud Composer and create a Google Cloud SQL instance

The data and the Airflow Directed Acyclic Graph (DAG) are hosted in the cloud storage bucket within a Google Cloud Composer instance (***Figure 1***), and the whole ELT pipeline is orchestrated via Airflow to move the data into the Cloud SQL instance (***Figure 2***).

3.2 Set up dbt

dbt is a user-friendly tool for building data pipelines, and integrates well with other cloud database applications. For this project, it is used to create an ELT workflow for extracting, loading and transforming data into an industry-standard data warehouse for consumption by business end-users, and pipeline orchestration is controlled in the Airflow DAG workflow via interacting with the dbt API. It also has a cloud IDE with integrated Git to enable full cloud workflows and better version control (***Figure 3***).

Figure 1: *Storage bucket within Google Cloud Composer, which will store Airflow DAGs and all data.*

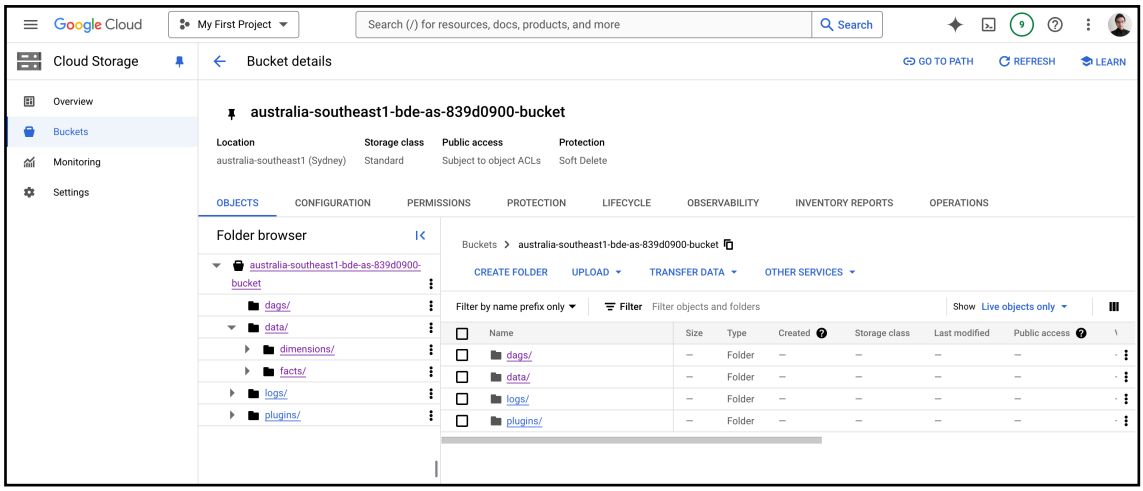


Figure 2: *User interface of the Cloud SQL instance created on Google Cloud.*

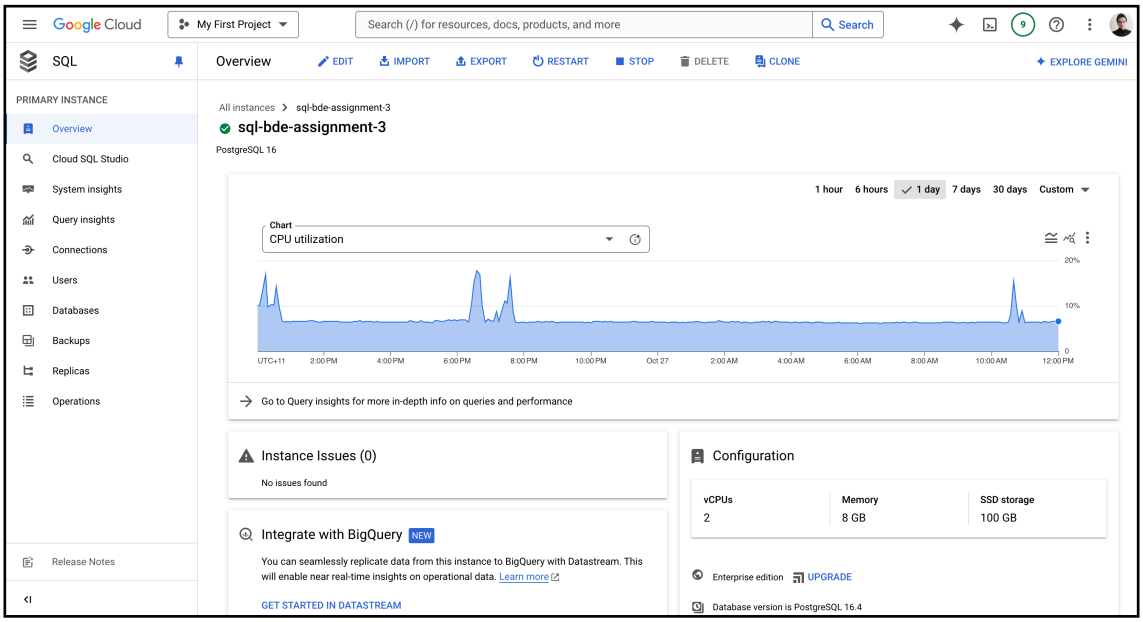
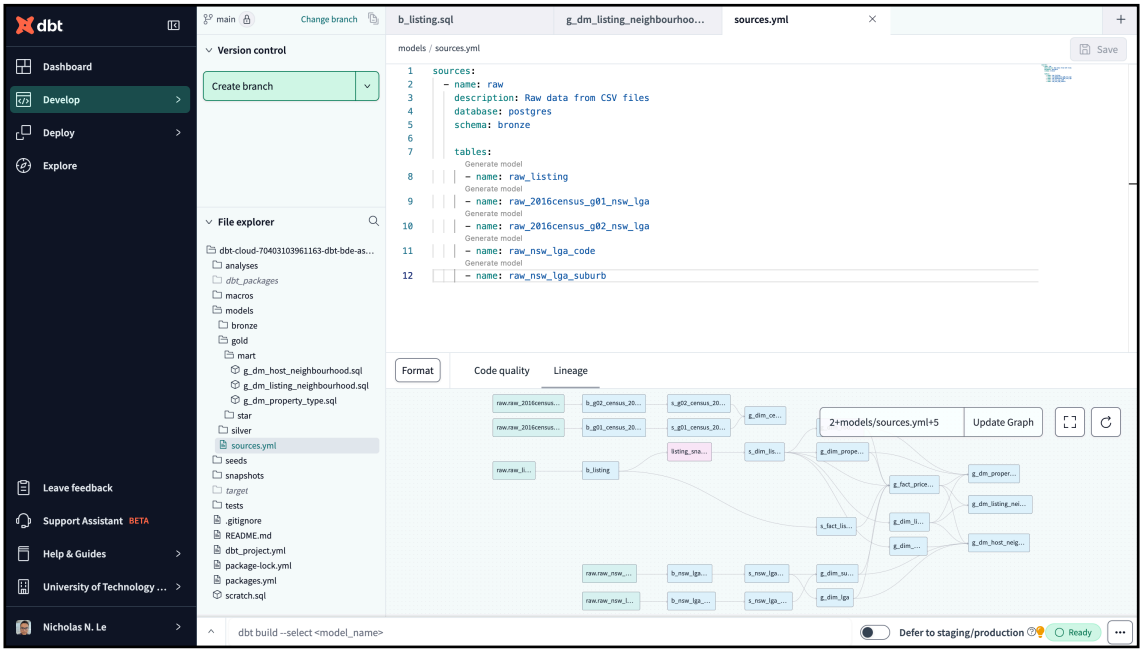


Figure 3: *dbt's cloud IDE, with Git version control.*



4 ELT Pipeline

4.1 Extract: Google Cloud storage

All data are extracted from source systems into CSV files within the Google Cloud bucket, organised into *dimensions* and *facts* subfolders. The *dimensions* subfolder contains raw census data, and the *facts* subfolder contains raw listings data.

Inside each subfolder, an *archive* directory is placed to land already-loaded data in order to avoid duplication when the pipeline is re-run.

4.2 Load: Cloud SQL

In a database management application (I use *DBeaver* for this project), I connect to the Cloud SQL instance created in the previous setup step. Then, I create the landing schemas for all data using the medallion architecture [3], which consists of three schemas: *bronze*, *silver*, and *gold*. These store data in different states, where:

- **Bronze**: stores data in its raw states as much as possible. This means storing most columns as *varchar* of different lengths, with the exception of a few *numeric* columns if those are very well-defined (e.g. ratings and scores).
- **Silver**: stores data that are cleaned, renamed for consistency, and of the correct data types. The details are in **Section 4.3**.
- **Gold**: stores data in *dimension tables* and a *fact table*, following a *star schema*. The details are in **Section 4.3**.

Inside the bronze schema, I create database tables into which all raw data will be loaded (***Figure 4***). I leave the other schemas blank, since dbt will handle the creation of all tables in these schemas upon full pipeline orchestration.

In Airflow, I create a DAG using Python that will load all census data, as well as the Airbnb listings data for the first month (May 2020) (***Figure 5***). After building the ELT pipeline, I will proceed to load the remaining listings data on a month-by-month basis, to ensure that SCD-2 columns generated by dbt are correctly timestamped.

Figure 4: *SQL codes to create the three schemas, and landing tables within the bronze schema.*

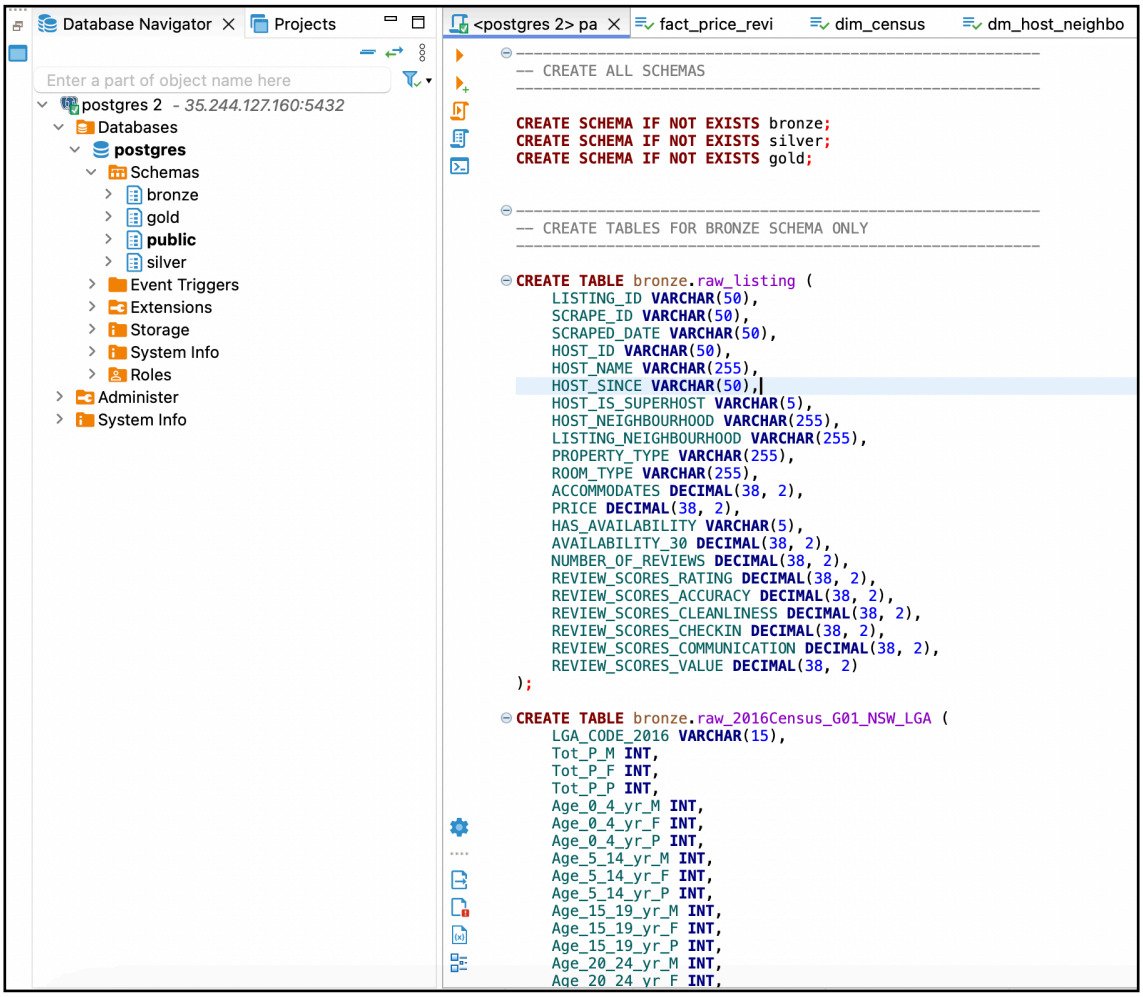


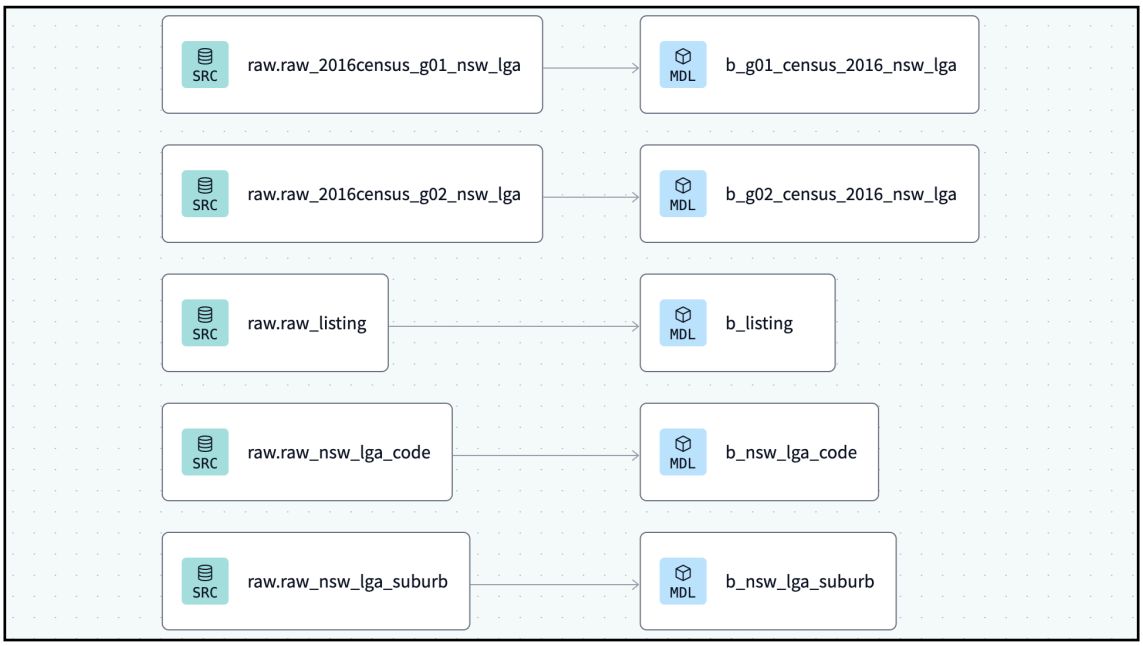
Figure 5: Python DAG that will import all census data, LGA/suburb referential data, as well as the first month's Airbnb listing, May 2020.

```
713    # Task Dependencies
714    [
715        import_load_dim_2016Census_G01_task,
716        import_load_dim_2016Census_G02_task,
717        import_load_dim_NSW_LGA_CODE_task,
718        import_load_dim_NSW_LGA_SUBURB_task,
719        import_load_fact_05_2020_task
720    ] \
```

4.3 Transform: dbt

In the bronze layer, raw datasets are ingested as-is, and go straight into the bronze layer representation in dbt. Unique columns are configured for files that have one, and an md5 hashed surrogate key is created from a range of identifying columns for files that do not have one. The result is 5 models in the bronze layer, which contain the "**b_**" prefix inside dbt.

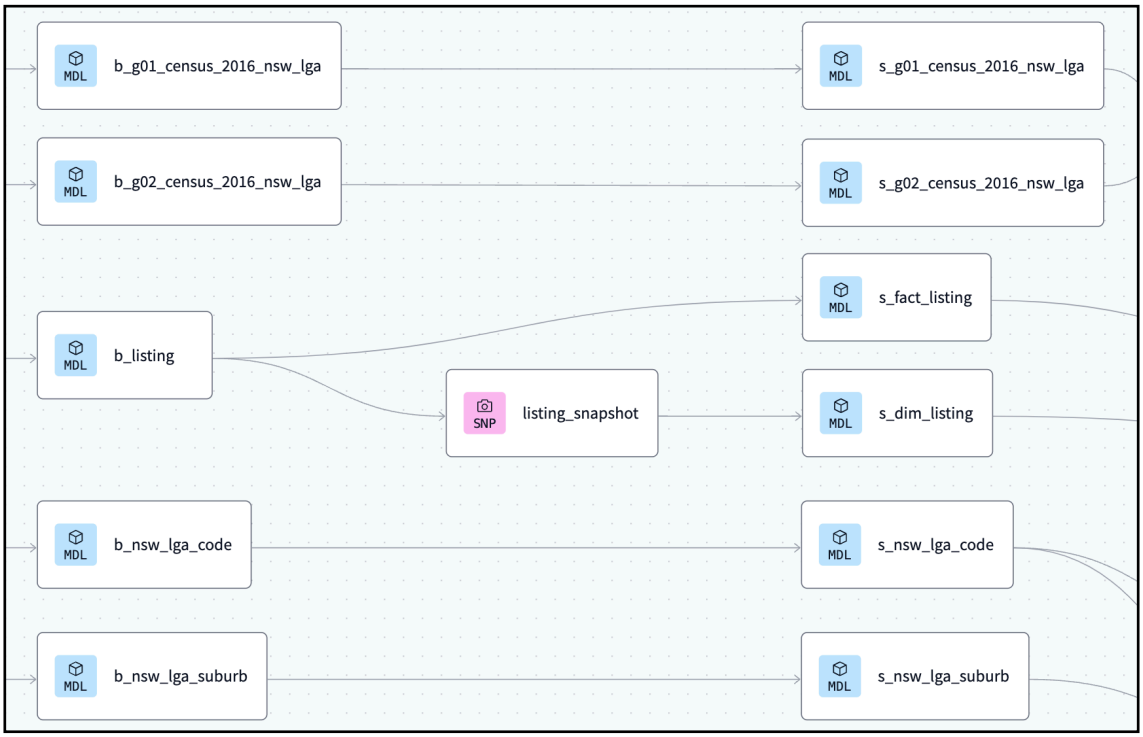
Figure 6: *Raw data (left) are loaded as-is into the bronze layer (right).*



From the bronze layer, data moves into the silver layer (with an "**s_**" prefix in dbt) where file/column names are renamed for consistency, and column values are casted into appropriate data types. The general principles are:

- ID columns to be in *bigint* format for efficient storage and reads
- Date columns to be in correct *date* format
- Numeric and string columns to be of appropriate size for their contents
- NULL values, which can be either the *SQL NULL* or the string *'NaN'*, are uniformly expressed as NULL

Figure 7: *Bronze-layer models are moved into silver-layer models with necessary transformations, while the listings are split into fact and dimension.*



The bronze census data models (which are all models except **b_listing**) retain their shape in the silver layer, while the **b_listing** model is split into 2 separate models, per ***Figure 7***:

- **s_fact_listing**: contains all metrics (facts) from the listings data, with the exception of a few identifier columns to reduce the number of joins in the gold layer. The columns to include are in Table 1.
- **listing_snapshot**: which is a snapshot of the rest of the dimension columns in the bronze listing table, using a timestamp strategy with *scraped_date* being the identifying column for updates. The snapshot produces the following columns:
    - *scd_id*: a unique hash value that identifies a row
    - *dbt_updated_at*: when the row was last updated
    - *dbt_valid_from*: the starting point of validity for a row
    - *dbt_valid_to*: the finishing point of validity for a row, which marks the starting point for a new version of that row. If NULL, the row is current

The **listing_snapshot** model then goes to the **s_dim_listing** model, which undergoes the usual transformation of the silver schema.

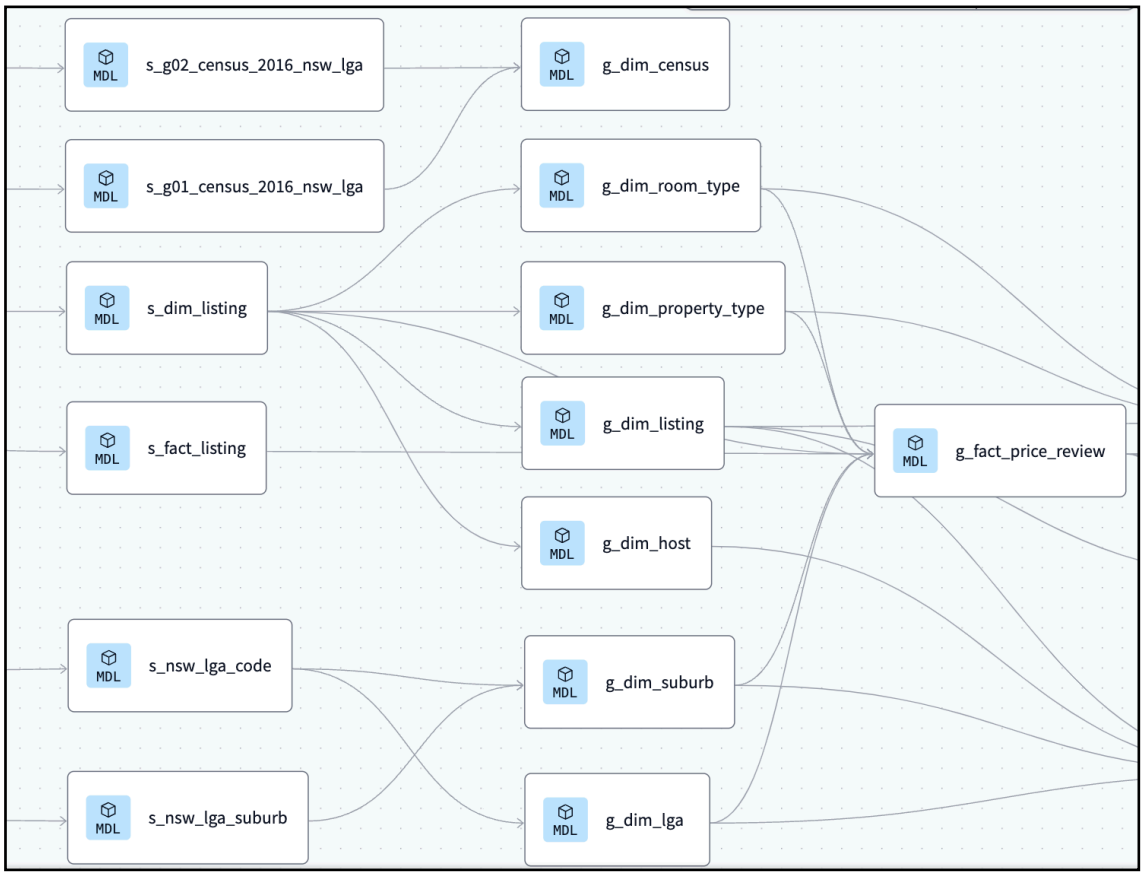
Table 1: Columns in the s_fact_listing model.

Column name	Is metric	Why it is a metric	Why include if not a metric
id	No	—	Surrogate key generated from listing_id and scraped_date
listing_id	No	—	To reduce joins in the gold layer
scraped_date	No	—	Date identifier
host_id	No	—	To reduce joins in the gold layer
host_is_superhost	Yes	Hosts can improve to gain superhost status	—
price	Yes	Can improve pricing to stay competitive	—
availability_30	Yes	Lots of monthly availability may indicate a weak listing	—
number_of_reviews	Yes	Can improve services based on reviews	—
review_scores_rating	Yes	Can improve services based on reviews	—
review_scores_accuracy	Yes	Can improve services based on reviews	—
review_scores_cleanliness	Yes	Can improve services based on reviews	—
review_scores_checkin	Yes	Can improve services based on reviews	—
review_scores_communication	Yes	Can improve services based on reviews	—
review_scores_value	Yes	Can improve services based on reviews	—

The gold layer (star schema) (with the "**g_**" prefix) includes seven dimensions and a fact table (***Figure 8***), including:

- **g_dim_census**: contain the unique *lga_id*, and the combined census information. No tracked SCD-2 columns
- **g_dim_room_type**: extracted from the listings table. Includes: *room_type_id, room_type, valid_from, valid_to*
- **g_dim_property_type**: extracted from the listings table. Includes: *property_type_id, property_type, valid_from, valid_to*
- **g_dim_listing**: extracted from the listings table. Includes: *listing_id, scrape_id, scraped_date, listing_neighbourhood, accommodates, has_availability, valid_from, valid_to*

Figure 8: *Silver-layer models are moved into gold-layer models where they are properly split into fact and dimension tables.*



- **g_dim_host**: extracted from the listings table. Includes: *host_id, host_name, host_since, host_neighbourhood, valid_from, valid_to*
- **g_dim_suburb**: includes *suburb_id, lga_id, suburb_name.* No tracked SCD-2 columns
- **g_dim_lga**: includes *lga_id, lga_name.* No tracked SCD-2 columns
- **g_fact_price_review**: central fact table containing the unique *id* columns from all dimension tables, as well as columns from ***Table 1*** marked as metrics.

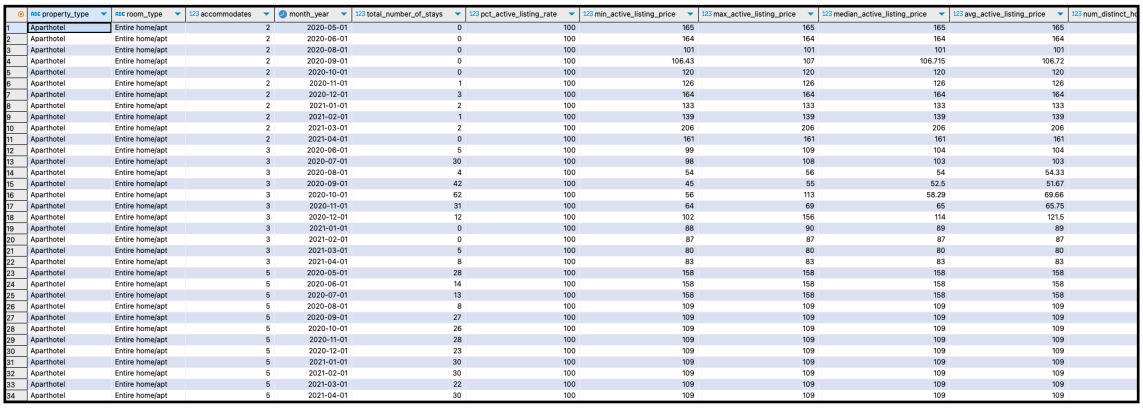
The gold layer (data mart schema) includes business-level aggregations that are derived from dimension and fact tables from the star schema, and are materialised as views. This schema includes the following views:

- **g_dm_listing_neighbourhood**: (***Figure 9***) provides aggregated metrics per listing's LGA neighbourhood and time, including active listings rate and their price statistics, number of hosts, superhost rate, total stays, ratings and review metrics, percentage changes for active/inactive listings, and estimated revenue per active listing.
- **g_dm_property_type**: (***Figure 10***) provides the same metrics as g_dm_host_neighbourhood, but per property characteristics such as property type, room type, and how many people it accommodates.
- **g_dm_host_neighbourhood**: (***Figure 11***) provides aggregated metrics per host's LGA neighbourhood and time, including the number of hosts, and estimated revenue (total and per host).

Figure 9: *First 24 rows of the dm_listing_neighbourhood view from the gold layer - data mart.*

	listing_neighbourhood	month_year	total_number_of_stays	pct_active_listing_rate	min_active_listing_price	max_active_listing_price	median_active_listing_price	avg_active_listing_price	num_distinct_hosts	pct_superhost_rate
1	Bayside	2020-05-01	63,418	99.37	0	15,367	86	208.08	1,248	35.26
2	Bayside	2020-06-01	63,645	99.31	0	14,478	84	202	1,231	35.34
3	Bayside	2020-07-01	59,928	99.14	0	14,376	85	213.47	1,066	41.26
4	Bayside	2020-08-01	53,398	99.36	0	2,904	80	110.89	1,079	42.08
5	Bayside	2020-09-01	70,942	99.32	0	2,903.71	79	107.64	1,188	42.34
6	Bayside	2020-10-01	64,142	99.22	0	2,904	79	108.77	1,173	35.46
7	Bayside	2020-11-01	59,809	99.2	0	2,904	79	112.01	1,151	34.4
8	Bayside	2020-12-01	66,494	99.2	0	2,904	80	114.83	1,164	34.02
9	Bayside	2021-01-01	64,790	99.19	0	2,904	80	120.55	1,163	31.04
10	Bayside	2021-02-01	59,918	99.18	0	2,904	80	117.42	1,146	30.89
11	Bayside	2021-03-01	61,271	99.17	0	5,000	80	121.35	1,141	30.15
12	Bayside	2021-04-01	60,708	99.15	0	5,000	80	117.58	1,121	29.17
13	Blacktown	2020-05-01	6,919	96.46	0	1,099	60	87.32	241	30.29
14	Blacktown	2020-06-01	7,473	96.48	0	1,099	60	87.97	236	33.47
15	Blacktown	2020-07-01	7,735	96.77	0	2,164	60	104.99	213	35.68
16	Blacktown	2020-08-01	6,764	96.26	0	2,000	60	95.43	218	34.86
17	Blacktown	2020-09-01	8,200	96.1	0	2,000	60	89.82	222	42.34
18	Blacktown	2020-10-01	8,207	96.23	0	2,000	64	96.48	218	30.28
19	Blacktown	2020-11-01	6,857	96.24	0	2,000	60	90.69	223	30.04
20	Blacktown	2020-12-01	8,925	96.23	0	2,000	60.5	90.63	219	31.51
21	Blacktown	2021-01-01	8,061	96.19	0	2,228	61	97.17	215	34.42
22	Blacktown	2021-02-01	6,657	96.05	0	2,000	62	91.22	218	33.49
23	Blacktown	2021-03-01	7,155	95.99	0	2,000	64	93.4	218	33.94
24	Blacktown	2021-04-01	7,553	95.96	0	2,000	64	94.7	220	25.91

Figure 10: *First 34 rows of the dm_property_type view from the gold layer - data mart.*



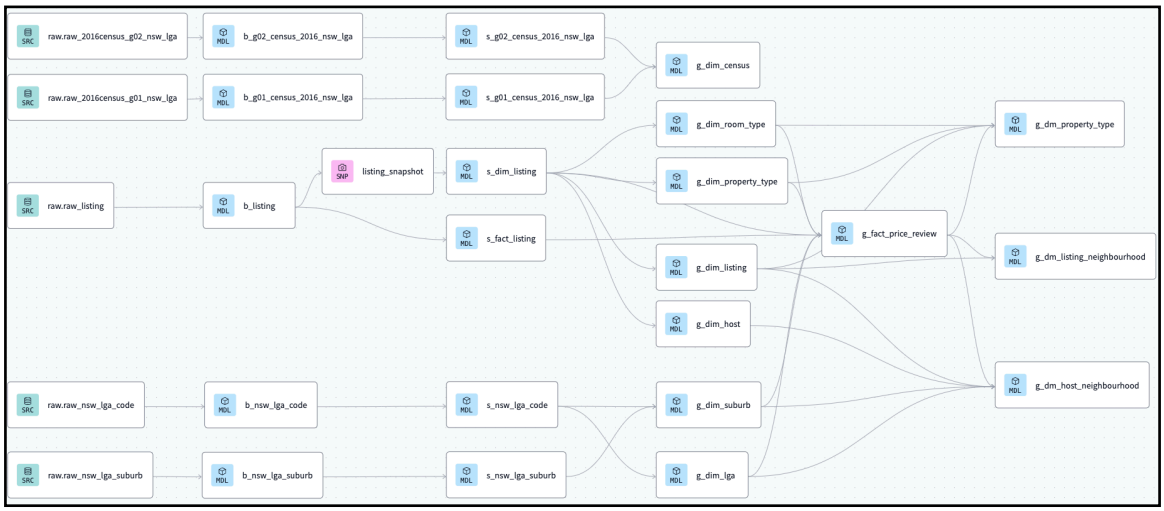
	property_type	room_type	accommodates	month_year	total_number_of_stays	pct_active_listing_rate	min_active_listing_price	max_active_listing_price	median_active_listing_price	avg_active_listing_price	num_distinct_ho
1	Aparthotel	Entire home/apt	2	2020-05-01	0	100	165	165	165	165	
2	Aparthotel	Entire home/apt	2	2020-06-01	0	100	164	164	164	164	
3	Aparthotel	Entire home/apt	2	2020-08-01	0	100	101	101	101	101	
4	Aparthotel	Entire home/apt	2	2020-09-01	0	100	106.43	107	106.715	106.72	
5	Aparthotel	Entire home/apt	2	2020-10-01	0	100	120	120	120	120	
6	Aparthotel	Entire home/apt	2	2020-11-01	1	100	126	126	126	126	
7	Aparthotel	Entire home/apt	2	2020-12-01	3	100	164	164	164	164	
8	Aparthotel	Entire home/apt	2	2021-01-01	2	100	133	133	133	133	
9	Aparthotel	Entire home/apt	2	2021-02-01	1	100	139	139	139	139	
10	Aparthotel	Entire home/apt	2	2021-03-01	2	100	206	206	206	206	
11	Aparthotel	Entire home/apt	2	2021-04-01	0	100	161	161	161	161	
12	Aparthotel	Entire home/apt	3	2020-06-01	5	100	99	109	104	104	
13	Aparthotel	Entire home/apt	3	2020-07-01	30	100	98	108	103	103	
14	Aparthotel	Entire home/apt	3	2020-08-01	4	100	54	56	54	54.33	
15	Aparthotel	Entire home/apt	3	2020-09-01	42	100	45	55	52.5	51.67	
16	Aparthotel	Entire home/apt	3	2020-10-01	62	100	56	113	58.29	69.66	
17	Aparthotel	Entire home/apt	3	2020-11-01	31	100	64	69	65	65.75	
18	Aparthotel	Entire home/apt	3	2020-12-01	12	100	102	156	114	121.5	
19	Aparthotel	Entire home/apt	3	2021-01-01	0	100	88	90	89	89	
20	Aparthotel	Entire home/apt	3	2021-02-01	0	100	87	87	87	87	
21	Aparthotel	Entire home/apt	3	2021-03-01	5	100	80	80	80	80	
22	Aparthotel	Entire home/apt	3	2021-04-01	8	100	83	83	83	83	
23	Aparthotel	Entire home/apt	5	2020-05-01	28	100	158	158	158	158	
24	Aparthotel	Entire home/apt	5	2020-06-01	14	100	158	158	158	158	
25	Aparthotel	Entire home/apt	5	2020-07-01	13	100	158	158	158	158	
26	Aparthotel	Entire home/apt	5	2020-08-01	8	100	109	109	109	109	
27	Aparthotel	Entire home/apt	5	2020-09-01	27	100	109	109	109	109	
28	Aparthotel	Entire home/apt	5	2020-10-01	26	100	109	109	109	109	
29	Aparthotel	Entire home/apt	5	2020-11-01	28	100	109	109	109	109	
30	Aparthotel	Entire home/apt	5	2020-12-01	23	100	109	109	109	109	
31	Aparthotel	Entire home/apt	5	2021-01-01	30	100	109	109	109	109	
32	Aparthotel	Entire home/apt	5	2021-02-01	30	100	109	109	109	109	
33	Aparthotel	Entire home/apt	5	2021-03-01	22	100	109	109	109	109	
34	Aparthotel	Entire home/apt	5	2021-04-01	30	100	109	109	109	109	

Figure 11: *First 24 rows of the dm_host_neighbourhood view from the gold layer - data mart.*

	host_neighbourhood_lga	month_year	num_distinct_hosts	estimated_revenue	estimated_revenue_per_host
1	Armidale Regional	2020-05-01	1	3,600	3,600
2	Armidale Regional	2020-06-01	1	3,600	3,600
3	Armidale Regional	2020-07-01	1	3,600	3,600
4	Armidale Regional	2020-08-01	1	3,600	3,600
5	Armidale Regional	2020-09-01	1	3,600	3,600
6	Armidale Regional	2020-10-01	2	5,280	2,640
7	Armidale Regional	2020-11-01	2	4,720	2,360
8	Armidale Regional	2020-12-01	2	9,396	4,698
9	Armidale Regional	2021-01-01	2	5,900	2,950
10	Armidale Regional	2021-02-01	2	3,740	1,870
11	Armidale Regional	2021-03-01	2	6,600	3,300
12	Armidale Regional	2021-04-01	2	6,600	3,300
13	Bathurst Regional	2020-05-01	11	619,941	56,358.27
14	Bathurst Regional	2020-06-01	10	581,737	58,173.7
15	Bathurst Regional	2020-07-01	9	636,690	70,743.33
16	Bathurst Regional	2020-08-01	9	770,189	85,576.56
17	Bathurst Regional	2020-09-01	9	839,590	93,287.78
18	Bathurst Regional	2020-10-01	10	879,075.24	87,907.52
19	Bathurst Regional	2020-11-01	9	983,035	109,226.11
20	Bathurst Regional	2020-12-01	9	860,769	95,641
21	Bathurst Regional	2021-01-01	9	870,799	96,755.44
22	Bathurst Regional	2021-02-01	9	868,677	96,519.67
23	Bathurst Regional	2021-03-01	9	1,013,153	112,572.56
24	Bathurst Regional	2021-04-01	10	876,092	87,609.2

5 Complete Airflow DAG Orchestration

Figure 12: *Complete ELT lineage in dbt.*



The complete ELT pipeline built with dbt is represented in ***Figure 12***. Next, the Python DAG script is updated, which includes tasks to import data month-by-month in a sequential order, then trigger the dbt pipeline, followed by a 130-second sleep task in between (***Figure 13***) so that Airflow can wait for dbt to finish a run before executing the next data import and the next dbt run. The manual sleep task is important in order to maintain the integrity of SCD-2 columns, since Airflow does not know whether and when dbt finishes a pipeline run.

Every time the pipeline is triggered, the snapshot model checks if any records are changing, and updates the SCD-2 columns appropriately (***Figure 14***). It is observed that dbt still creates new rows even though no changes are detected; however, this is unlikely to be an engineering issue, but rather an issue from dbt, as observed in a related support article from the community. [4]

Figure 13: *Complete Python DAG pipeline in Apache Airflow.*

Figure 14: *Example rows from the listing dimension table, with appropriate valid_from and valid_to timestamps. NULL indicates that the row contains the latest available data.*

	listing_id	scrape_id	scraped_date	listing_neighbourhood	accommodates	has_availability	valid_from	valid_to
1	11,156	20,200,000,000,000	2020-05-11	Sydney	1	t	2020-05-11 00:00:00.000	2020-06-11 00:00:00.000
2	11,156	20,200,000,000,000	2020-06-11	Sydney	1	t	2020-06-11 00:00:00.000	2020-07-16 00:00:00.000
3	11,156	20,200,000,000,000	2020-07-16	Sydney	1	t	2020-07-16 00:00:00.000	2020-08-21 00:00:00.000
4	11,156	20,200,000,000,000	2020-08-21	Sydney	1	t	2020-08-21 00:00:00.000	2020-09-11 00:00:00.000
5	11,156	20,200,000,000,000	2020-09-11	Sydney	1	t	2020-09-11 00:00:00.000	2020-10-15 00:00:00.000
6	11,156	20,200,000,000,000	2020-10-15	Sydney	1	t	2020-10-15 00:00:00.000	2020-11-05 00:00:00.000
7	11,156	20,200,000,000,000	2020-11-05	Sydney	1	t	2020-11-05 00:00:00.000	2020-12-16 00:00:00.000
8	11,156	20,201,200,000,000	2020-12-16	Sydney	1	t	2020-12-16 00:00:00.000	2021-01-13 00:00:00.000
9	11,156	20,210,100,000,000	2021-01-13	Sydney	1	t	2021-01-13 00:00:00.000	2021-02-10 00:00:00.000
10	11,156	20,200,000,000,000	2021-02-10	Sydney	1	t	2021-02-10 00:00:00.000	2021-03-06 00:00:00.000
11	11,156	20,200,000,000,000	2021-03-06	Sydney	1	t	2021-03-06 00:00:00.000	2021-04-12 00:00:00.000
12	11,156	20,200,000,000,000	2021-04-12	Sydney	1	t	2021-04-12 00:00:00.000	[NULL]
13	12,351	20,200,000,000,000	2020-05-10	Sydney	2	t	2020-05-10 00:00:00.000	2020-06-11 00:00:00.000
14	12,351	20,200,000,000,000	2020-06-11	Sydney	2	t	2020-06-11 00:00:00.000	2020-07-15 00:00:00.000
15	12,351	20,200,000,000,000	2020-07-15	Sydney	2	t	2020-07-15 00:00:00.000	2020-08-22 00:00:00.000
16	12,351	20,200,000,000,000	2020-08-22	Sydney	2	t	2020-08-22 00:00:00.000	2020-09-11 00:00:00.000
17	12,351	20,200,000,000,000	2020-09-11	Sydney	2	t	2020-09-11 00:00:00.000	2020-10-16 00:00:00.000
18	12,351	20,200,000,000,000	2020-10-16	Sydney	2	t	2020-10-16 00:00:00.000	2020-11-05 00:00:00.000
19	12,351	20,200,000,000,000	2020-11-05	Sydney	2	t	2020-11-05 00:00:00.000	2020-12-16 00:00:00.000
20	12,351	20,201,200,000,000	2020-12-16	Sydney	2	t	2020-12-16 00:00:00.000	2021-01-11 00:00:00.000
21	12,351	20,210,100,000,000	2021-01-11	Sydney	2	t	2021-01-11 00:00:00.000	2021-02-10 00:00:00.000
22	12,351	20,200,000,000,000	2021-02-10	Sydney	2	t	2021-02-10 00:00:00.000	2021-03-06 00:00:00.000
23	12,351	20,200,000,000,000	2021-03-06	Sydney	2	t	2021-03-06 00:00:00.000	2021-04-15 00:00:00.000
24	12,351	20,200,000,000,000	2021-04-15	Sydney	2	t	2021-04-15 00:00:00.000	[NULL]
25	14,250	20,200,000,000,000	2020-05-11	Northern Beaches	6	t	2020-05-11 00:00:00.000	2020-06-12 00:00:00.000
26	14,250	20,200,000,000,000	2020-06-12	Northern Beaches	6	t	2020-06-12 00:00:00.000	2020-07-15 00:00:00.000
27	14,250	20,200,000,000,000	2020-07-15	Northern Beaches	6	t	2020-07-15 00:00:00.000	2020-08-22 00:00:00.000
28	14,250	20,200,000,000,000	2020-08-22	Northern Beaches	6	t	2020-08-22 00:00:00.000	2020-09-11 00:00:00.000
29	14,250	20,200,000,000,000	2020-09-11	Northern Beaches	6	t	2020-09-11 00:00:00.000	2020-10-16 00:00:00.000
30	14,250	20,200,000,000,000	2020-10-16	Northern Beaches	6	t	2020-10-16 00:00:00.000	2020-11-05 00:00:00.000
31	14,250	20,200,000,000,000	2020-11-05	Northern Beaches	6	t	2020-11-05 00:00:00.000	2020-12-16 00:00:00.000
32	14,250	20,201,200,000,000	2020-12-16	Northern Beaches	6	t	2020-12-16 00:00:00.000	2021-01-13 00:00:00.000
33	14,250	20,210,100,000,000	2021-01-13	Northern Beaches	6	t	2021-01-13 00:00:00.000	2021-02-10 00:00:00.000
34	14,250	20,200,000,000,000	2021-02-10	Northern Beaches	6	t	2021-02-10 00:00:00.000	2021-03-06 00:00:00.000
35	14,250	20,200,000,000,000	2021-03-06	Northern Beaches	6	t	2021-03-06 00:00:00.000	2021-04-14 00:00:00.000
36	14,250	20,200,000,000,000	2021-04-14	Northern Beaches	6	t	2021-04-14 00:00:00.000	[NULL]

6 Addressing Business Questions

6.1 Demographic differences between top- and bottom-performing LGAs based on estimated revenue per active listing

Per ***Figure 15***, the top-performing LGA (*TPLGAs*) group in terms of estimated revenue per active listing consists of Mosman, Northern Beaches and Woollahra; while the bottom-performing LGA group (*BPLGAs*) is Canterbury-Bankstown, Fairfield and Blacktown. The most prominent demographic differences between these groups are:

- **Population**: (***Figure 16***) *BPLGAs* generally have greater population counts than *TPLGAs*, both male and female.
- **Education attendance by age group**: (***Figure 17***) due to population differences, *BPLGAs* generally have more people attending education institutions than *TPLGAs*.
- **Education status**: (***Figure 18***) *BPLGAs* generally have more Year-12 graduates; however, they have significantly more people who did not attend school compared to *TPLGAs*.
- **Indigenous status**: (***Figure 19***) The total indigenous population in *BPLGAs* outnumbers that in *TPLGAs* by a wide margin. This holds also for subcategories such as Aboriginals, Torres Strait Islanders, or both.
- **Language spoken at home**: (***Figure 20***) *BPLGAs* generally have more people speaking languages other than English at home than *TPLGAs*.

Figure 15: *Top- and bottom-performing LGAs based on estimated revenue per active listing.*

	ABC listing_neighbourhood	123 sum_revenue_per_active_listing
1	Mosman	108,937.34
2	Northern Beaches	90,249.97
3	Woollahra	79,610.12
4	Canterbury-Bankstown	18,081.34
5	Fairfield	16,179.92
6	Blacktown	15,305.01

***Figure 16**: Total population. Left: male; right: female.*

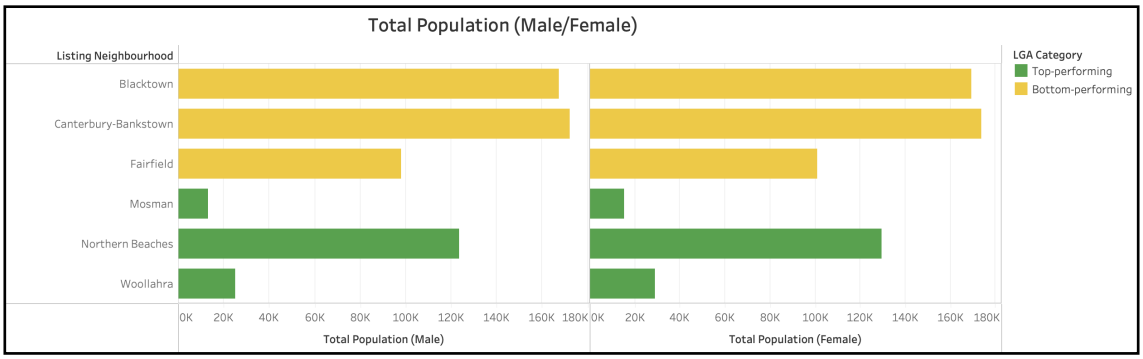


***Figure 17:** Education attendance by age group. Left to right: 0-4, 5-14, 15-19, 20-24, over 25.*

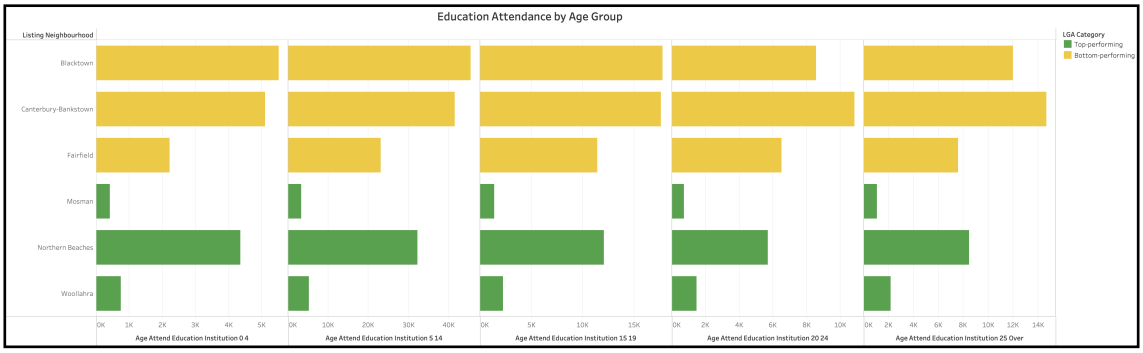


***Figure 18:** Education status. Left: year 12 completed; right: did not attend school.*

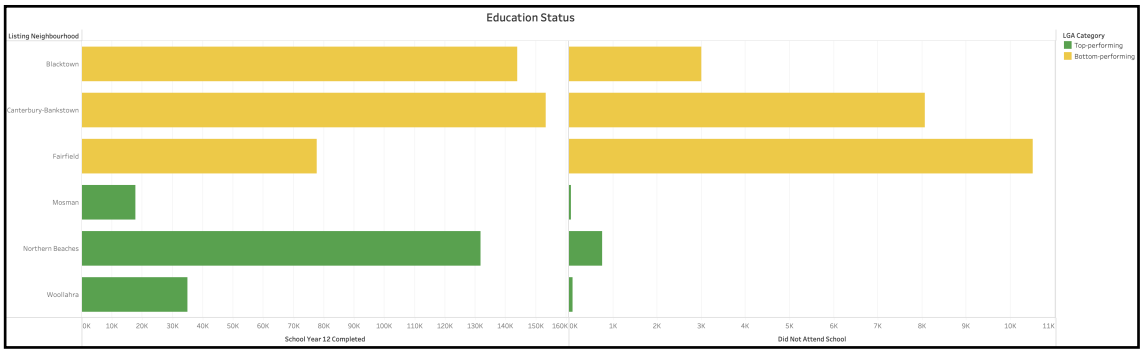


***Figure 19:** Indigenous status. Left to right: total population, Aboriginal, Torres Strait Islanders, and both.*

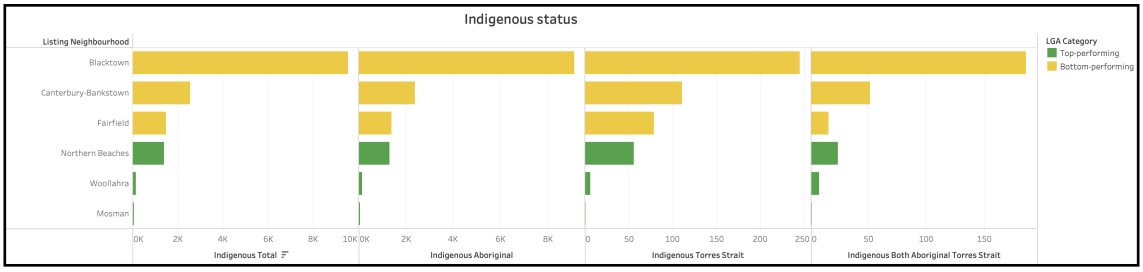
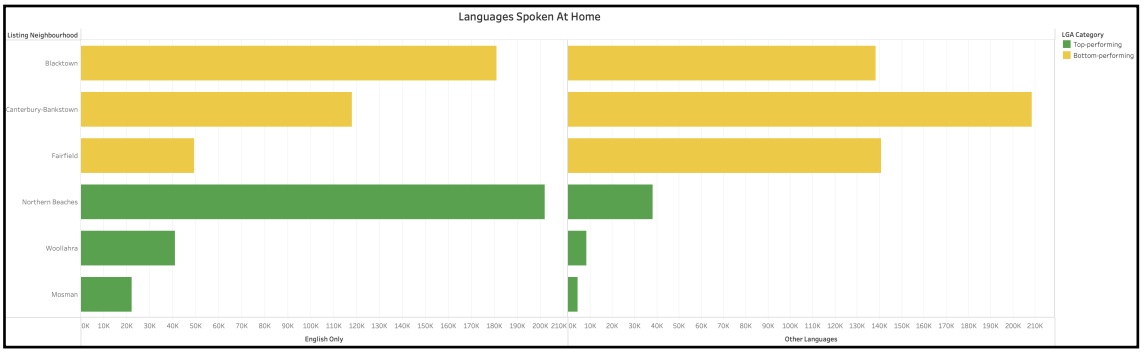


***Figure 20:** Languages spoken at home. Left: English only; right: other languages.*



6.2 *Correlation between median age and estimated revenue per active listing*

Figure 21: *Median age and estimated revenue per active listing, per listing neighbourhood.*

	neighbourhood	median_age	estimated_revenue_per_active_listing
1	Bayside	35	2,344.43
2	Blacktown	33	1,275.42
3	Burwood	33	1,929.44
4	Camden	33	1,767.08
5	Campbelltown	34	2,043.82
6	Canada Bay	36	3,142.01
7	Canterbury-Bankstown	35	1,506.78
8	Cumberland	32	3,690.75
9	Fairfield	36	1,348.33
10	Georges River	37	1,970.61
11	Hornsby	40	2,622.81
12	Hunters Hill	43	5,655.88
13	Inner West	36	3,589.95
14	Lane Cove	36	3,967.51
15	Liverpool	33	2,131.09
16	Mosman	42	9,078.11
17	Northern Beaches	40	7,520.83
18	North Sydney	37	4,250.96
19	Parramatta	34	1,864.32
20	Penrith	34	2,209.9
21	Randwick	34	4,383.4
22	Ryde	36	2,315.81
23	Strathfield	32	1,517.79
24	Sutherland Shire	40	4,449.41
25	Sydney	32	3,793.53
26	The Hills Shire	38	2,228.86
27	Waverley	35	5,678.59
28	Willoughby	37	4,215.86
29	Woollahra	39	6,634.18

Figure 21 shows the median age of a neighbourhood and the average estimated revenue per active listing in that neighbourhood. Per data here, there is a moderately strong correlation (**0.6431**) between these variables.

6.3 *Best property characteristics for a listing*

Figure 22: *Best property characteristics for the top 5 neighbourhoods in terms of estimated revenue per active listing.*

	neighbourhood	property_type	room_type	accommodates	avg_estimated_revenue_per_active_listing
1	Bayside	Boat	Entire home/apt	10	484,928.4
2	Blacktown	Boat	Entire home/apt	10	484,928.4
3	Burwood	Boat	Entire home/apt	10	484,928.4
4	Camden	Boat	Entire home/apt	10	484,928.4
5	Campbelltown	Boat	Entire home/apt	10	484,928.4

Per ***Figure 22***, the best property type that is likely to have the most number of stays is a boathouse that accommodates 10, and is for rent in its entirety.

6.4 *LGA localities for hosts with multiple listings*

For hosts with 2 or more listings, 3909 hosts have properties in a single LGA, and 1186 hosts have properties across different LGAs. This means that the majority of hosts have their properties in dense concentration possibly for better management, especially companies with lots of listings (Figure 23).

Figure 23: *Result table for hosts with more than 2 listings, ordered by number of listings.*

	host_name	host_id	number_of_listings	number_of_distinct_lgas
1	Stay.Live.Grow	194,230,296	496	24
2	Stay.Live.Grow	194,230,296	496	24
3	Stay.Live.Grow	194,230,296	496	24
4	Stay.Live.Grow	194,230,296	496	24
5	Stay.Live.Grow	194,230,296	496	24
6	Stay.Live.Grow	194,230,296	496	24
7	Stay.Live.Grow	194,230,296	496	24
8	Stay.Live.Grow	194,230,296	496	24
9	Stay.Live.Grow	194,230,296	496	24
10	Stay.Live.Grow	194,230,296	496	24
11	Stay.Live.Grow	194,230,296	496	24
12	Stay.Live.Grow	194,230,296	496	24
13	Stay.Live.Grow	194,230,296	496	24
14	Stay.Live.Grow	194,230,296	496	24
15	Stay.Live.Grow	194,230,296	496	24
16	Stay.Live.Grow	194,230,296	496	24
17	Stay.Live.Grow	194,230,296	496	24
18	Stay.Live.Grow	194,230,296	496	24
19	Team Gospodin	175,128,252	356	23
20	Team Gospodin	175,128,252	356	23
21	Team Gospodin	175,128,252	356	23
22	Team Gospodin	175,128,252	356	23
23	Team Gospodin	175,128,252	356	23
24	Team Gospodin	175,128,252	356	23
25	Team Gospodin	175,128,252	356	23
26	Team Gospodin	175,128,252	356	23
27	Team Gospodin	175,128,252	356	23
28	Team Gospodin	175,128,252	356	23
29	Team Gospodin	175,128,252	356	23
30	Team Gospodin	175,128,252	356	23
31	Team Gospodin	175,128,252	356	23
32	Team Gospodin	175,128,252	356	23
33	Team Gospodin	175,128,252	356	23
34	Team Gospodin	175,128,252	356	23

6.5 *Hosts with a single listing: mortgage repayments*

Figure 24: *Counts of hosts with a single listing that can and cannot cover annualised mortgage repayments, per LGA, ordered by percentage of hosts that can cover repayments.*

	lga_name	count_hosts_that_can_cover_mortgage	count_hosts_that_cannot_cover_mortgage	percentage_hosts_that_can_cover_mortgage_in_lga
1	Fairfield	1	0	100
2	Northern Beaches	996	307	76.44
3	Campbelltown	3	1	75
4	Sutherland Shire	87	40	68.5
5	Mosman	165	88	65.22
6	Waverley	1,763	1,207	59.36
7	Sydney	2,351	1,700	58.04
8	North Sydney	423	309	57.79
9	Liverpool	4	3	57.14
10	Hunters Hill	28	21	57.14
11	Randwick	973	746	56.6
12	Woollahra	530	474	52.79
13	Hornsby	11	10	52.38
14	Inner West	767	721	51.55
15	Parramatta	14	14	50
16	Ryde	39	39	50
17	Canada Bay	116	118	49.57
18	Lane Cove	88	92	48.89
19	Willoughby	129	137	48.5
20	Burwood	38	57	40
21	Cumberland	34	52	39.53
22	Canterbury-Bankstown	102	157	39.38
23	Strathfield	35	55	38.89
24	Blacktown	3	5	37.5
25	Bayside	254	432	37.03
26	The Hills Shire	3	6	33.33
27	Georges River	54	122	30.68
28	Penrith	0	4	0

For hosts with a single listing, there is only one host in the Fairfield LGA whose property revenue is enough to cover the annualised mortgage repayments in that LGA, per census data. For all other LGAs, hosts are likely to have a difficult time paying back the annual mortgage if this is their only source of income.

6 Project Challenges

This is a challenging project, where I encountered lots of difficulties including:

- **Tracking SCD-2**: I initially set up the Airflow DAG to process monthly data without a waiting task in between the months, since I was unaware that a successful dbt trigger in Airflow didn't immediately mean an actual successful run in dbt. As a result, there were runs where 7 months of data were already ingested in Airflow but dbt only finished 2 runs, leading to the second dbt run ingesting 6 months of data at once, leading to the wrong SCD-2 columns. Implementing the extra waiting task helped eliminate this issue.
- **Duplicate data from incorrect joins**: The joining of dimension tables, which contains duplicated rows with only different timestamps, with the fact table resulted

in the joined table polynomially blowing up in size due to the many-to-many joins. This caused data quality issues, and also query execution issues when it comes to views. I remedied this issue by selecting only distinct records from dimension tables while leaving out the SCD-2 columns.

- **Lots of joins in the gold schema**: the fact table contains joins from other dimension tables in order to obtain their ID columns, especially for newly created dimensions (e.g. *property_type*, *room_type*). This created many joins, which, while logically valid, slowed query executions. To remedy this issue, for dimensions with existing ID columns that can be utilised (e.g. *listing_id*, *host_id*), I grabbed them straight from **s_fact_listing** in the silver schema (which already contains these ID columns) rather than joining them with new dimensions in the gold schema (i.e. **g_dim_listing**, **g_dim_host**). The dependency in ***Figure 12*** shows this solution in action, where it is observed that not all gold dimension tables are connected to the gold fact table. This helped reduce the number of joins, which led to faster queries.

7 Conclusion

This project has set up an ELT pipeline to load Airbnb and census data, as well as built a data mart for business end-users. To ensure data logicality, future work needs to include an extra step to set up the primary/foreign key constraints in PostgreSQL for all dimension/fact tables so they can be logically linked, hence avoiding any integrity issues in the future.

References

[1] M. Cox, J. Morris, and T. Higgins, “How is Airbnb really being used in and affecting the neighbourhoods of your city?,” *insideairbnb.com*. https://insideairbnb.com/ (accessed Oct. 27, 2024).

[2] Australian Bureau of Statistics, “Census,” *abs.gov.au*, 2021. https://www.abs.gov.au/census (accessed Oct. 27, 2024).

[3] Databricks, “What is a Medallion Architecture?,” *Databricks*. https://www.databricks.com/glossary/medallion-architecture (accessed Oct. 27, 2024).

[4] dbt Labs, “DBT Snapshot creating duplicate rows even though there’s no change in source data,” *dbt Community Forum*, May 10, 2023. https://discourse.getdbt.com/t/dbt-snapshot-creating-duplicate-rows-even-though-theres-no-change-in-source-data/8223/7 (accessed Oct. 28, 2024).