



NOVEMBER 6, 2022


ASSIGNMENT 3: BUILDING ELT DATA PIPELINES WITH AIRFLOW

BIG DATA ENGINEERING SPRING 2022

RAJAT SINGH

14330397

Rajat.singh-2@student.uts.edu.au



Contents

1. Introduction	2
2. Datasets	2
3. Technology Platforms Used	2
4. Setup	3
4.1. Airflow Setup using Google Cloud Platform	3
4.2. Data Storage in Google Cloud Platform	3
4.3. Google Cloud Platform and Snowflake Connection.....	3
5. Snowflake Data Warehouse Design	4
5.1. Raw Layer	4
5.2. Staging Layer	5
5.3. Warehouse Layer	5
5.3.1. Fact Table	5
5.3.2. Dimension Tables	6
5.4. Data Mart Layer	6
6. ETL Using Airflow	7
7. Issues, Bugs and Resolutions	8
7.1. No Active Warehouse Selected.....	8
7.2. Date Formatting Issues	9
7.3. Setting Up Data Types for Table with a lot of Columns	9
7.4 Unresolved Issue: Airflow DAG running at the same time	9
8. Business Question	10
9. Conclusion	11
10. References	12
11. Appendix	12

1. Introduction

The aim of the project is to build a production ready data pipeline with Airflow to populate Airbnb and Census dataset into a Snowflake Data Warehouse. The data should be processed and cleaned before loading the information into Snowflake. Moreover, a data mart layer is created to help business for analytical purposes.

2. Datasets

The following two datasets were used in this project: -

- *Airbnb Dataset*

Airbnb is a digital marketing firm that links people looking for lodging (Airbnb guests) with those looking to rent out their properties (Airbnb hosts) for a short-term or long-term period. Apartments (which make up most of the rentals), houses, yachts, and numerous other items are also available. Airbnb is a significant disruptor of the established hotel sector, with 150 million users in 191 countries as of 2019. (This is akin to how Uber and other emerging transportation services have disrupted the traditional intra-city transportation services). Airbnb creates a tonne of data as a rental ecosystem, including but not limited to rental density across regions (cities and neighbourhoods), pricing variances across rentals, host-guest interactions in the form of reviews, and so on.

- *Census Dataset*

The Population and Housing Census (Census) is Australia's largest collection of statistics conducted by the Australian Bureau of Statistics (ABS). For over 100 years, the Census has provided a snapshot of Australia, showing how the country has changed over time and allowing us to plan. The purpose of the Census is to collect accurate data on the main characteristics of Australians and the housing they live in on Census night. The 2016 census counted about 10 million households and about 24 million people, the largest ever counted.

3. Technology Platforms Used

The following technology were used to implement this project

- *Snowflake*

Snowflake's Data Cloud is powered by an advanced data platform provided as Software-as-a-Service (SaaS). Snowflake enables data storage, processing, and analytic solutions that are faster, easier to use, and far more flexible than traditional offerings (Snowflake, n.d).

- *Google Cloud Platform*

Google Cloud is made up of a collection of physical resources, such as computers and hard drives, and virtual resources, such as virtual machines (VMs), which are in Google's data centres around the world. Google Cloud Platform provides infrastructure as a service, platform as a service, and serverless computing environments.

- *Airflow*

Apache Airflow is used for planning and orchestrating data pipelines or workflows. Data pipeline orchestration refers to the arrangement, coordination, planning and management of complex data pipelines from various sources.

4. Setup

4.1. Airflow Setup using Google Cloud Platform

- The Cloud Composer API is enabled, if not enabled already.
- A new cloud environment is created with Composer 1 using the attributes in the following table.




Table 1 GCP Cloud Environment Setup Attributes

Attribute	Value
Name	Bde-at3
Location	australia-southeast1
Machine type	n1-standard-2
Disk Size (GB)	30
Airflow Image Version	composer-1.19.12-airflow-2.3.3
Service Account	<Default Service Account>

- The following PYPI packages are added to the environment.
 - Pandas
 - Snowflake-connector-python: Version 2.4.5
 - Snowflake-sqlalchemy: Version 1.2.4
 - Apache-airflow-providers-snowflake: Version 1.3.0
- Enable_xcom_picking is enabled in Airflow Configuration overrides.

4.2. Data Storage in Google Cloud Platform

In the Google Cloud Storage, the bucket linked with Composer Environment is selected and three new folders are created inside it.

<input type="checkbox"/>	Name	Size	Type
<input type="checkbox"/>	 Census_LGA/	—	Folder
<input type="checkbox"/>	 NSW_LGA/	—	Folder
<input type="checkbox"/>	 listings/	—	Folder

The census, LGA and Airbnb listing data files are added to their corresponding folders.

4.3. Google Cloud Platform and Snowflake Connection

- The path of data folder inside cloud storage bucket is copied.
- A new worksheet is created in Snowflake and a query is executed to create a new external storage integration with the name of GCP.
- DESCRIBE INTEGRATION GCP command is used to see the parameters of the connection and STORAGE_GCP_SERVICE_ACCOUNT is copied.

- A new role is created in 'IAM and Admin' on Google Cloud Platform with the name of GCP Storage. The following permissions were added to the GCP role.
 - + storage.buckets.get
 - + storage.objects.create
 - + storage.objects.delete
 - + storage.objects.get
 - + storage.objects.list
 - The newly created Role is used to give permissions to STORAGE_GCP_SERVICE_ACCOUNT added as a new principal is added to cloud storage bucket corresponding to composer environment for airflow.
 - A new Warehouse is created in Snowflake with the following attributes.
 - + Name: AIRBNB_WAREHOUSE
 - + Size: Small 2 credits/hour
 - The Snowflake URL on the web browser is used to obtain the account and region. The URL could be in one of the two formats.
 - + [https://app.snowflake.com/\[REGION\]/\[ACCOUNT\]](https://app.snowflake.com/[REGION]/[ACCOUNT])
 - + [https://\[ACCOUNT\].\[REGION\].snowflakecomputing.com](https://[ACCOUNT].[REGION].snowflakecomputing.com)
 - In the Composer Interface on GCP, the Airflow UI is opened to add a new record in Admin tab for connecting airflow to Snowflake.
- The following table has the attributes are added to the new record.

Table 2 The new record created in Airflow UI Admin

Attribute	Value
Connection ID	Snowflake_conn_id
Connection Type	Snowflake
Host	[ACCOUNT].[REGION].snowflakecomputing.com
Database	BDE-AT3
Account	[ACCOUNT]
Region	[REGION]
Warehouse	AIRBNB_WAREHOUSE
Password	[SNOWFLAKE PASSWORD]

5. Snowflake Data Warehouse Design

A four layer architecture was designed on Snowflake for our project.

5.1. Raw Layer

- A new stage stage_gcp is created using storage integration GCP and the fold path of GCP storage bucket.
- To read all the csv files in the folder, a new file format was created.
- Different tables were created for different kinds of data files stored in the bucket.

- All of the different types of dataset was kept into different folders, as mentioned section 4.2, so datasets were accessed using these folder names. Wildcards are also used when specifying a pattern.

5.2. Staging Layer

The staging layers transformed the raw data tables into a structured tabular format by providing the data types and column names for each column. Mostly the original names were used to avoid any discrepancies.

The data dictionary could be found in the Appendix at the end of the document.

The important steps performed during the Staging Process:

- The month_year was fetched using the filenames in the listing table.
- Since, the host since was not using Snowflake datetime format, it was parsed from varchar to date column.
- A new column code was added to both census tables by removing LGA prefix from LGA_Code column.
- A new Excel file was created to create sections of the create table sql statements, since some of the tables had more than hundred rows, to save typing effort. This Excel sheet could be found linked in the submission folder.

5.3. Warehouse Layer

The warehouse layer followed a star-schema with a fact and many dimension tables. The staging tables were broken into fact and dimension tables to optimise data storage space as well.

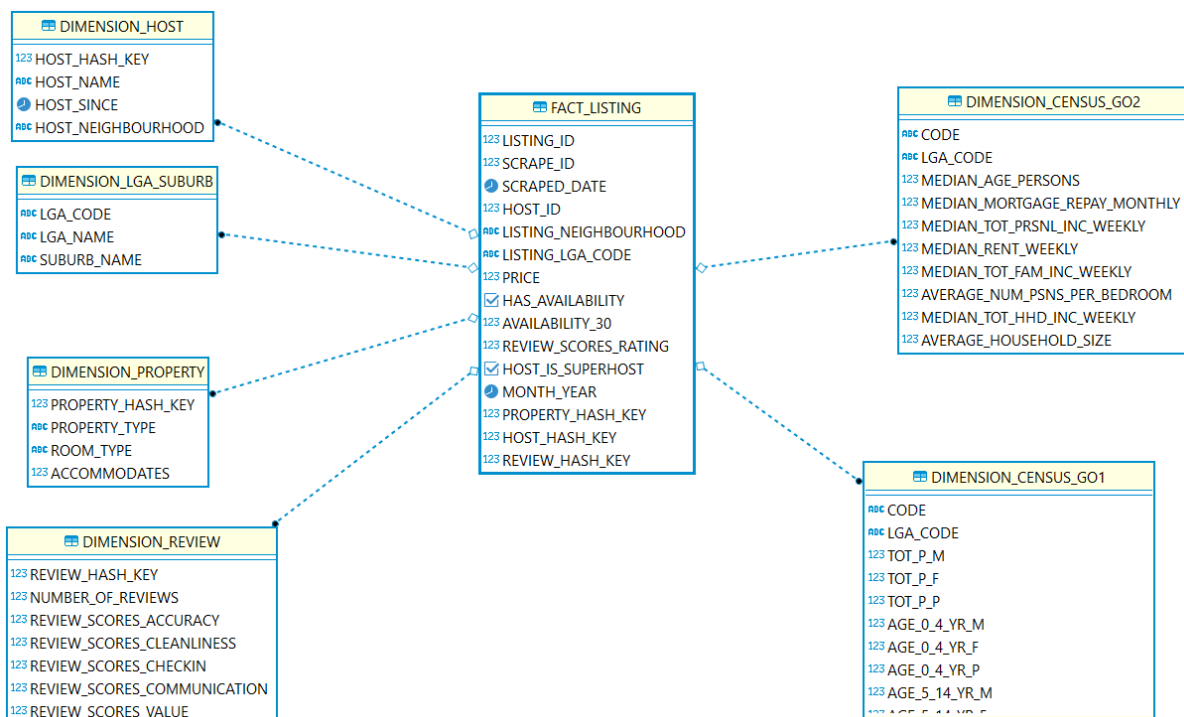


Figure 1 Entity Relation Diagram of Star Schema in Warehouse

5.3.1. Fact Table

The table consisted of all the IDs, metrics and dates from the Airbnb listing tables.

Since the fact table must be able to map to other dimensions, this was done by creating a hash for each dimension. Hashing is the process of transforming any given key or string to another value, in case of Snowflake that is 16-digit unsigned number. The number could be used to uniquely identify a row of data.

For mapping to property dimension, a `property_hash_key` was created which was the hash value of `property_type`, `room_type` and `accommodates`. Similarly, a `host_hash_key` and `review_hash_key` was also created.

The `listing_neighbourhood` column was used to determine LGA codes by mapping it to the corresponding suburbs.

5.3.2. Dimension Tables

The following dimension tables were created from Airbnb listing and census dataset:

- `dimension_host`: This dimension uniquely identifies all the host for Airbnb properties in NSW area. This dimension was created using Listing table. Mapped to Fact table using
- `dimension_property`: The property dimension identifies all the attributes associated with the properties. Property dimension was also created using the Listing table.
- `dimension_review`: The review table contains all the possible reviews of associated with the listing.
- `dimension_lga_suburb`: The `lga` and the `lga_suburb` tables are joined to create a single dimension containing information about `lga_codes`, `lga_name` and `suburb_name`. Mapped to Fact table using `lga_codes`.
- `dimension_census_go1` and `dimension_census_go2`: The dimensions are created by simply copying all the contents from census table from staging layers. Mapped to Fact table by `lga_codes`.

For more information about the structure, and table properties, please check Appendix and ER diagram in Warehouse section.

5.3. Data Mart Layer

In the data mart layer, the following three views were created by joining the fact with dimension table as per the requirements:

- `Dm_listing_neighbourhood`: For each `listing_neighbourhood` and `month_year`, the following properties were calculated:
 - + Active listings rate
 - + Minimum, maximum, median and average price for active listings
 - + Number of distinct hosts
 - + Superhost rate
 - + Average of `review_scores_rating` for active listings
 - + Percentage change for active listings
 - + Percentage change for inactive listings
 - + Total Number of stays
 - + Average Estimated revenue per active listings
- `dm_property_type`: For each `property_type`, `room_type`, `accommodates` and `month_year`, the following properties were calculated:
 - + Active listings rate
 - + Minimum, maximum, median and average price for active listings

- + Number of distinct hosts
 - + Superhost rate
 - + Average of review_scores_rating for active listings
 - + Percentage change for active listings
 - + Percentage change for inactive listings
 - + Total Number of stays
 - + Average Estimated revenue per active listings
- dm_host_neighbourhood: For each host_neighbourhood_lga (host_neighbourhood transformed to an LGA) and month_year, the following properties were calculated:
 - + Number of distinct host
 - + Estimated Revenue
 - + Estimated Revenue per host (distinct)

6. ETL Using Airflow

Airflow is used to populate and update the data warehouse in an Extract, Load and Transfer(ELT) pipeline. Three separate DAG files are created to create and populate the data in Snowflake data warehouse, these three DAG should have different schedule intervals to optimise the use of Airflow.

The following DAG files are created in Airflow: -

- Setup_DAG: The DAG is used to create a new database, schema and storage integration for the project. Since, the database should be initialised once, the schedule interval is set to once. Refer the figure given below for the structure.

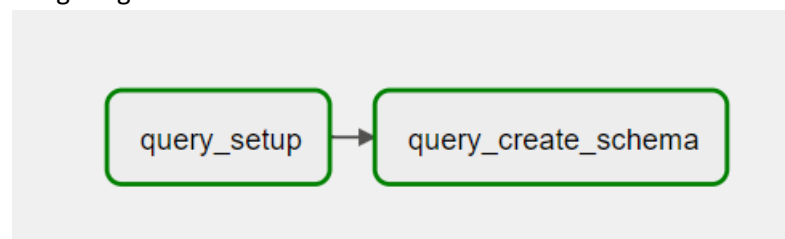


Figure 2 Setup_DAG Structure

- Load_Census_data_DAG: This DAG is used to create or replace table for census and demographic data. Since, the data is mostly static and doesn't change within 5 years. So, the DAG should be scheduled to run after every five years. For demonstration purposes, the schedule interval is set to once. Refer the figure given below for the structure.

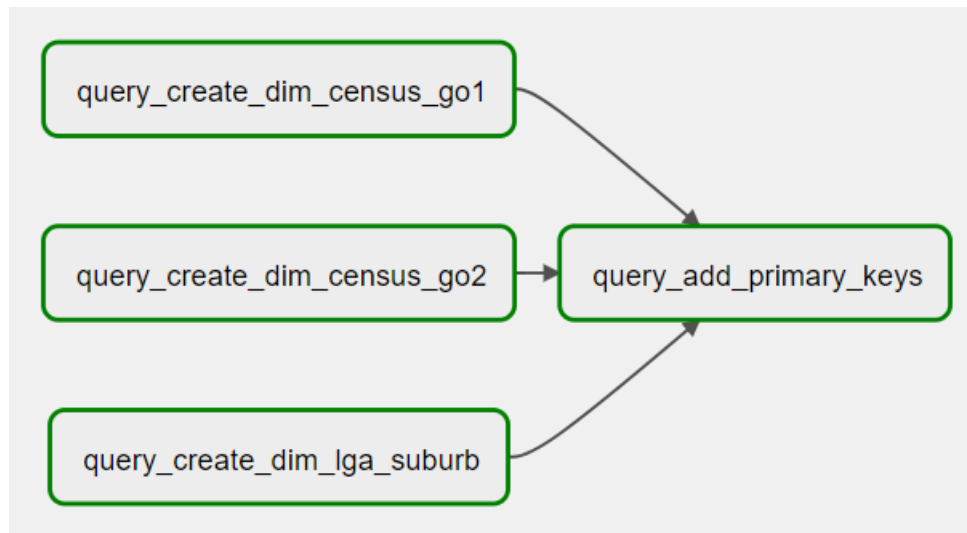


Figure 3 - Load_Census_data_DAG DAG Structure

- Load_Listing_data_DAG: This DAG is used to create or replace the tables associated with listing data. The business requirement doesn't mention how often the data would be refreshed, but it could be assumed the DAG to run every month. For demonstration purposes, the schedule interval is set to once. Refer the figure given below for the structure.

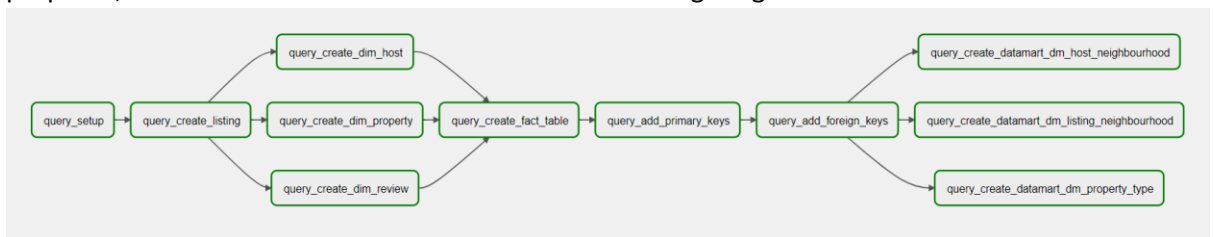


Figure 4 - Load_Listing_data_DAG Structure

It should be ensured these DAG files are run one after the other in the chronological order as per the list mentioned above.

7. Issues, Bugs and Resolutions

The following issues and bugs were resolved while working on this project:

7.1. No Active Warehouse Selected

While working on Snowflake, sometimes a 'No active Warehouse' error occurred.

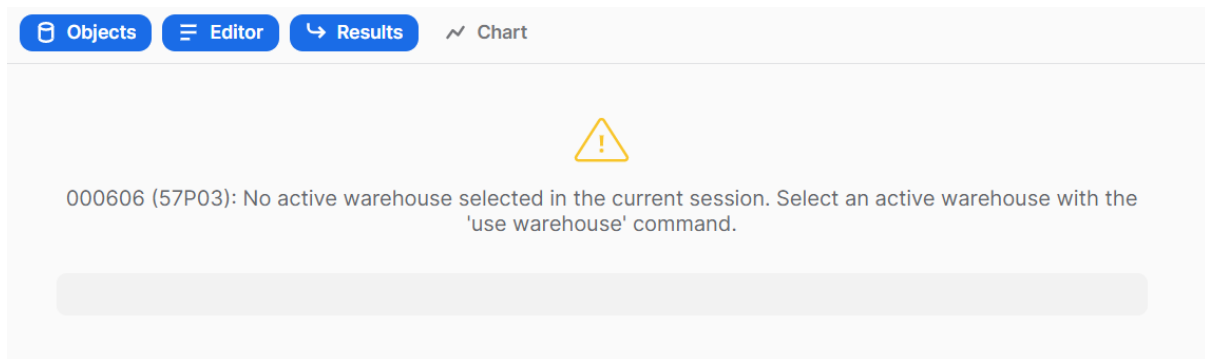


Figure 5 No Active Warehouse Error

Resolution:

Mostly the Warehouse session was suspended due to inactivity. So, the warehouse session should be resumed in Snowflake by going to 'Warehouse' in 'Admin' menu and then resuming the desired warehouse from the list.

The Use warehouse command is also used to select the desired warehouse while executing a command (Stack Overflow).

7.2. Date Formatting Issues

Sometimes, the date in the dataset doesn't follow the snowflake date format.

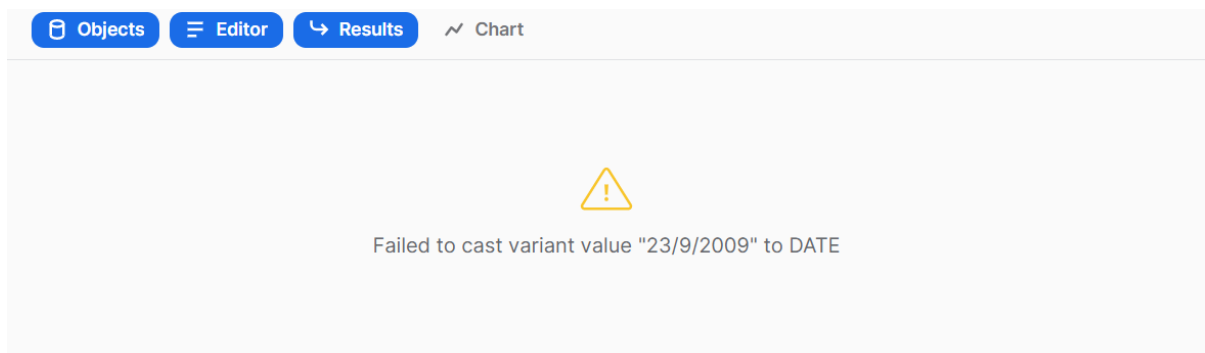


Figure 6 Invalid Date Error

Resolution: The date should be converted into varchar and then to_date() function is used to cast the varchar into date.

7.3. Setting Up Data Types for Table with a lot of Columns

The table census table has more than hundreds of columns, so assigning data types to every column is lengthy and time taking activity.

Resolution: A excel file was created to create section of SQL script by using column name and datatype.

7.4 Unresolved Issue: Airflow DAG running at the same time

As mentioned in section 6, Load_Census_data_DAG would run once in five years, whereas Load_Listing_data_DAG would run monthly. So, it could be possible, but highly unlikely that the DAGs would run at the same time. If these two DAG are updating the data at the same time, it could lead to data discrepancies in our Data Warehouse.

Fix: Instead of Scheduling Load_Census_data_DAG, this DAG could be run manually every five years after updating the census dataset in the storage bucket when Load_Listing_data_DAG is not running.

Cross-DAG dependencies could also be created, but it hasn't done since the impact is low and the census data must be updated in the bucket manually anyways.

8. Business Question

The following business questions were answered after creating and populating data warehouse.

- *What are the main differences from a population point of view (i.g. higher population of under 30s) between the best performing "listing neighbourhood" and the worst (in terms of estimated revenue per active listings) over the last 12 months?*

The best performing listing neighbourhood is Mosman whereas the worst performing neighbourhood is Fairfield.

The graph below indicates that the population of Fairfield is 7.7 times than of Mosman.

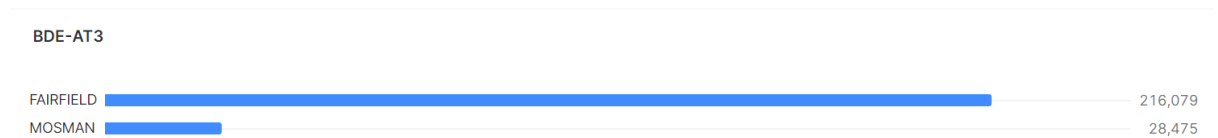


Figure 7 Population Comparison between Fairfield and Mosman

But if we take compare the demographic data between different age group an interesting pattern emerges. People aged between 20 to 24 is more 7 times mores in Fairfield than Mosman, but if I compare the population between the age of 45 to 54 the ratio is 6:1. Indicating that Mosman has more people in later stages of life and hence has more money to spend on the Airbnb listing to get higher return.

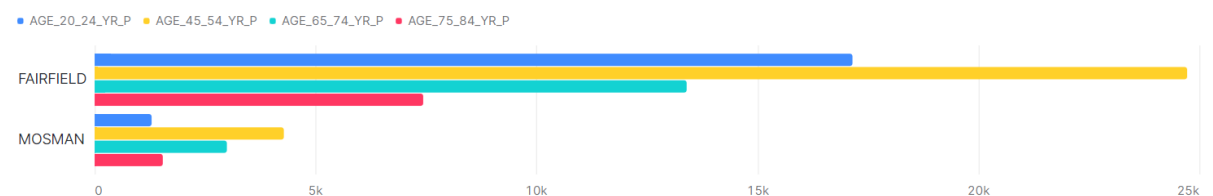


Figure 8 Population Comparison between Fairfield and Mosman depending upon their age.

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

LISTING NEIGHBOURHOOD	PROPERTY_TYPE	ROOM_TYPE	REVENUE	REVENUE_PER_PERSON
Mosman	Villa	Entire home/apt	8	14047.78947
Woollahra	Entire villa	Entire home/apt	2	41769.2

Waverley	Entire home/apt	Entire home/apt	9	12514.62963
Hunters Hill	Townhouse	Entire home/apt	4	3375
Northern Beaches	Room in boutique hotel	Hotel room	2	25552.5

In the top 5 listing neighbourhood, there were about 800 combination of type of listing with a maximum stay of 30 days. So, we needed a new criteria to find the best possible listing, therefore Revenue per person was calculated and the best five listing are given below.

- *Do hosts with multiple listings are more inclined to have their listings in the same LGA as where they live?*

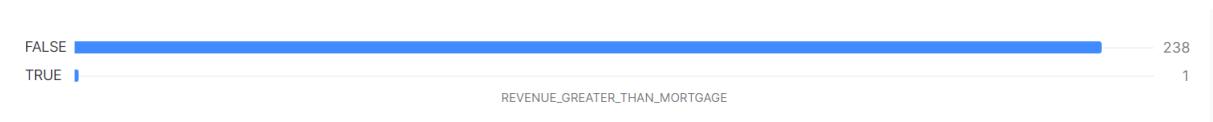
Yes, the host with multiple listings are more inclined to have their listings in the same LGA. This could be proven by the table below.

CONDITION	COUNT
Distinct Host with more than 3 listing	27681
Distinct Host with more than 2 listing	28684
Distinct Host with more than 1 listing	29815
All Distinct Host	31001
Number of Listing Neighbourhood when host has more than 1 listing	31891

We can see that even if we have twenty-nine thousand count host has more than 1 listing, the number of listing neighbourhood when host has more than 1 house is still thirty-one thousand. The two numbers are so similar to each other, therefore it could be implied that most of host with multiple listings are inclined to have their listing in same LGA.

- *For hosts with a unique listing, does their estimated revenue over the last 12 months can cover the annualised median mortgage repayment of their listing's "listing_neighbourhood"?*

In our findings, it could be concluded that estimated revenue is not enough to cover the annualised median mortgage repayment of the listing neighbourhood. Overwhelming, the estimated revenue was less than annualised median mortgage repayment.



9. Conclusion

This document includes the steps involved in creating a data pipeline with Airflow to populate Airbnb and Census dataset. A four layered architecture is created in Snowflake dataset including a data mart layer that could be used by external users to query reports.

Moreover, all of the business question are answered using the dataset.

10. References

Snowflake. n.d. Snowflake Documentation.<https://docs.snowflake.com/en/user-guide/intro-key-concepts.htm>

Szozda, Lukasz. 2021, September 8. Stackoverflow.

<https://stackoverflow.com/questions/69104269/no-active-warehouse-selected-in-the-current-session-select-an-active-warehouse>

11. Appendix

Table Name:		FACT_LISTING		Table Type:		TABLE			
Table Description:				Catalog:		BDE_AT3			
				Schema:		WAREHOUSE			

Table Name:		DIMENSION_REVIEW		Table Type:		TABLE					
Table Description:				Catalog:		BDE_AT3					
				Schema:		WAREHOUSE					
<div>Columns</div>		Column Name	#	Data Type	Length	Scale	Not Null	Auto Generated	Auto Increment	Default	Description
<div>Keys</div>		123 REVIEW_HASH_KEY	1	NUMBER	19		[]	[]	[]		
<div>Foreign Keys</div>		123 NUMBER_OF_REVIEWS	2	DOUBLE			[]	[]	[]		
<div>References</div>		123 REVIEW_SCORES_ACCURA...	3	NUMBER	38		[]	[]	[]		
<div>DDL</div>		123 REVIEW_SCORES_CLEANU...	4	NUMBER	38		[]	[]	[]		
<div>Virtual</div>		123 REVIEW_SCORES_CHECKIN	5	NUMBER	38		[]	[]	[]		
		123 REVIEW_SCORES_COMMU...	6	NUMBER	38		[]	[]	[]		
		123 REVIEW_SCORES_VALUE	7	NUMBER	38		[]	[]	[]		

Table Name:		DIMENSION_PROPERTY		Table Type:		TABLE				
Table Description:				Catalog:		BDE_AT3				
				Schema:		WAREHOUSE				
<div><div>Columns</div><div><div>Keys</div><div>Foreign Keys</div><div>References</div><div>DDL</div><div>Virtual</div></div></div>										
	Column Name	#	Data Type	Length	Scale	Not Null	Auto Generated	Auto Increment	Default	Description
	123 PROPERTY_HASH_KEY	1	NUMBER	19		[]	[]	[]		
	123 PROPERTY_TYPE	2	VARCHAR	16,777,216		[]	[]	[]		
	123 ROOM_TYPE	3	VARCHAR	16,777,216		[]	[]	[]		
	123 ACCOMMODATES	4	NUMBER	38		[]	[]	[]		

Table Name: DIMENSION_LGA_SUBURB

Table Type: TABLE

Table Description:

Catalog: BDE_AT3

Schema: WAREHOUSE

Columns

	Column Name	#	Data Type	Length	Scale	Not Null	Auto Generated	Auto Increment	Default	Description
Keys	LGA_CODE	1	VARCHAR	16,777,216		[]	[]	[]		
Foreign Keys	LGA_NAME	2	VARCHAR	16,777,216		[]	[]	[]		
References	SUBURB_NAME	3	VARCHAR	16,777,216		[]	[]	[]		
DDL										
Virtual										

Table Name: DIMENSION_HOST

Table Type: TABLE

Table Description:

Catalog: BDE_AT3

Schema: WAREHOUSE

Columns

	Column Name	#	Data Type	Length	Scale	Not Null	Auto Generated	Auto Increment	Default	Description
Keys	HOST_HASH_KEY	1	NUMBER	19		[]	[]	[]		
Foreign Keys	HOST_NAME	2	VARCHAR	16,777,216		[]	[]	[]		
References	HOST_SINCE	3	DATE			[]	[]	[]		
DDL	HOST_NEIGHBOURHOOD	4	VARCHAR	16,777,216		[]	[]	[]		
Virtual										

Table Name: DIMENSION_CENSUS_GO2

Table Type: TABLE

Table Description:

Catalog: BDE_AT3

Schema: WAREHOUSE

Columns

	Column Name	#	Data Type	Length	Scale	Not Null	Auto Generated	Auto Increment	Default	Description
Keys	CODE	1	VARCHAR	16,777,216		[]	[]	[]		
Foreign Keys	LGA_CODE	2	VARCHAR	16,777,216		[]	[]	[]		
References	MEDIAN_AGE_PERSONS	3	NUMBER	38		[]	[]	[]		
DDL	MEDIAN_MORTGAGE_REP...	4	NUMBER	38		[]	[]	[]		
Virtual	MEDIAN_TOT_PRSNL_INC_WEEKLY	5	NUMBER	38		[]	[]	[]		
	MEDIAN_RENT_WEEKLY	6	NUMBER	38		[]	[]	[]		
	MEDIAN_TOT_FAM_INC_W...	7	NUMBER	38		[]	[]	[]		
	AVERAGE_NUM_PSNS_PER...	8	NUMBER	38		[]	[]	[]		
	MEDIAN_TOT_HHD_INC...	9	NUMBER	38		[]	[]	[]		
	AVERAGE_HOUSEHOLD_SI...	10	NUMBER	38		[]	[]	[]		