UNIVERSITY OF TECHNOLOGY, SYDNEY

6 NOVEMBER 2022

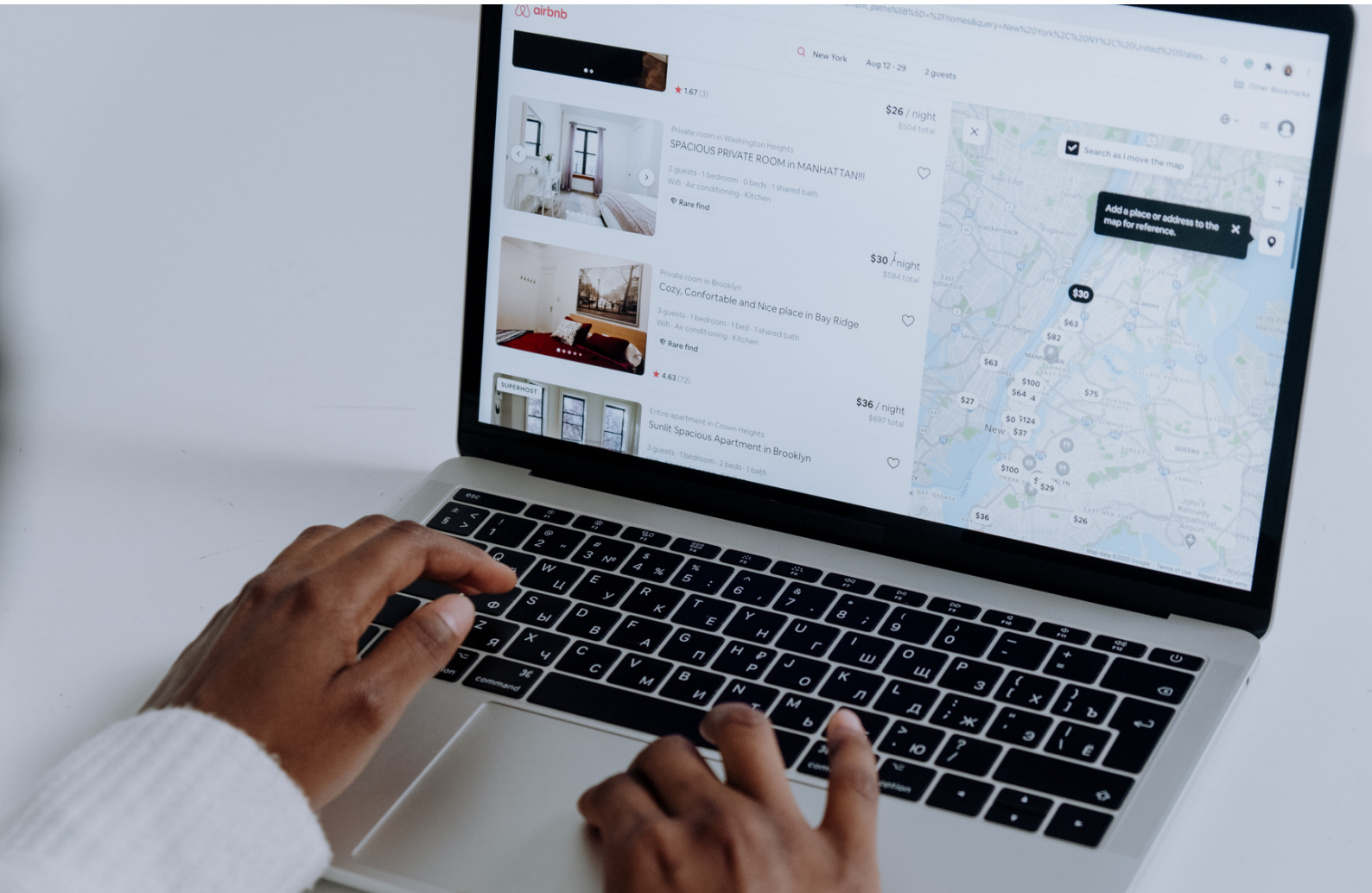
Building ELT Data Pipelines with Airflow

Big Data Engineering - III

Analyzing Airbnb Listings & Census Data

PRESENTED BY

Akshaya Parthasarathy
14081133



Handover Report
Big Data Engineering – Assignment Task 3

1. Overview

Founded by Brian Chesky, Nathan Blecharczyk, and Joe Gebbia in 2008, *Airbnb, Inc*., operates as an online marketplace focused on short-term homesteads and experiences. Airbnb, a shortened version of its original name, *AirBedandBreakfast*, revolutionised traveling by providing easy, accessible ways to customise itineraries without compromising on quality. As one of the world's most liveable cities, Sydney is a popular attraction for various Airbnb's today.

Our dataset deals with 1 year of Airbnb listings in the New South Wales region, from **May 2020 to January 2021**. Containing details such as the type of listing, its neighbourhood, and also about the host, this is an extensive dataset with over **400K** rows. In addition to this, we also have **Australia Census Data** from 2016 that provides details of the public demographics and their households such as various age groups, education levels, incomes etc. These are categorised based on "**LGA**" or local government area.

The aim of the assignment is to build data pipelines using the two different input datasets. These datasets will go through all the necessary steps before being ingested into a data warehouse and used in data marts for analytical purposes. Snowflake will be used for processing and analysing the data while Airflow will be used to build the pipelines and GCP to store the datasets.

2. Dataset Exploration

The **Airbnb** Listing dataset consists of 12 different files each divided by month and year. In total they have **22 columns and 412,122 rows**. There are **two census** tables available each identified as G01 and G02. The **G01** table consists of **109 columns and 132 rows** (each row dedicated to a single LGA). This table entails all information about the different age groups, their education levels and other split-ups of demographics. The **G02** table has a total of **9 columns and 132 rows** and various household details such as median income, household size and mortgage rates etc.

For the purpose of mapping the two datasets for analysis and exploration, **two additional datasets** have been provided. One contains the LGA Code along with the LGA Name, and the other consists of the LGA NAME and Suburb Name.

3. System Architecture

Airflow is a platform that lets you build and run workflows. A workflow is represented as a DAG (a Directed Acyclic Graph), and contains individual pieces of work called Tasks, arranged with dependencies and data flows taken into account. A DAG specifies the dependencies between Tasks, and the order in which to execute them and run retries; the Tasks themselves describe what to do, be it fetching data, running analysis, triggering other systems, or more.

Building a Data Pipeline can automate the ELT process, minimising errors and improving efficiency while doing so. Leveraging Airflow to build these data pipelines allows for easier data ingestion and data processing. The Airflow UI also lets you see what DAGs and their tasks are doing, view logs, and allow for debugging.

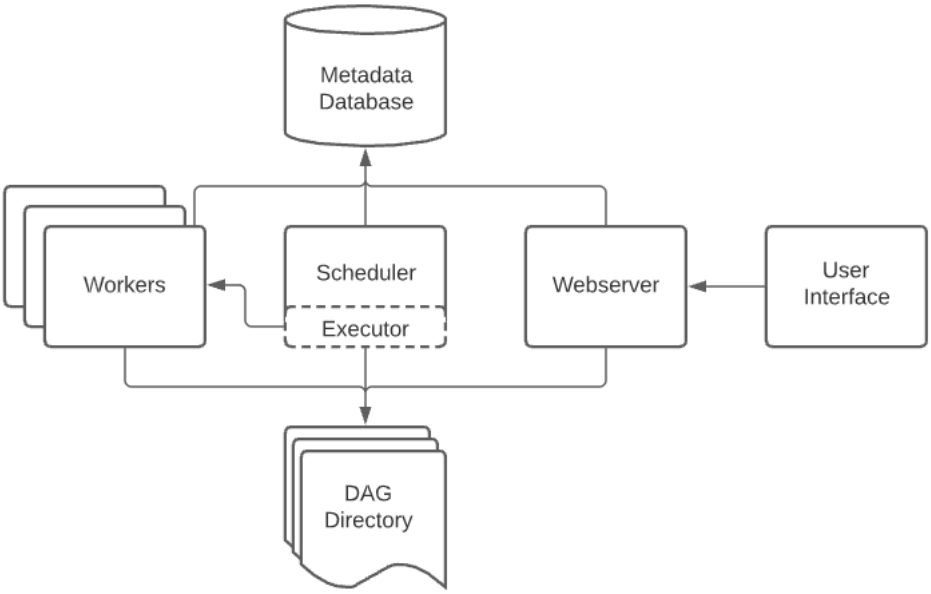


Fig 1: Airflow Architecture

4. Data Ingestion (Part 0)

Note: proceed with this only after completing the following checklist -

- Creation and configuration of Snowflake account & GCP account
- Setup of appropriate libraries and packages in Airflow environment (Appendix 2)
- Creation of custom role for storage GCP service account in Snowflake on GCP (Appendix 3)
- Creation of Warehouse with valid credentials on snowflake to link to Airflow (Appendix 4)

Step 1: Upload all the individual data files from local storage to GCP cloud storage and make note of the URL.

Step 2: In Snowflake environment, create a DATABASE for this project. Use that database and create a SCHEMA named RAW.

Step 3: Create a Storage Integration named GCP with the storage location set to the data URL retrieved earlier from GCP.

Step 4: Create a stage based of the storage integration created and the URL obtained.

Step 5: Verify that a connection has been established by trying the following command to see the list of files: list @stage_gcp.

5. Datawarehouse Design (Part 1)

a) Schema Raw

The raw schema will contain all external tables of the available datasets, ingested in the appropriate file format.

Create a file format called *file_format_csv* with the following arguments so that the CSV files are read properly.

Type as CSV
Field Delimiter as ‘, ‘
Skip Header set to 1
NULL_IF = ('\\N', 'NULL', 'NUL', ‘')
FIELD_OPTIONALLY_ENCLOSED_BY = ‘“‘

Using the above file mart created, begin creating 5 separate external tables for each data base with the following arguments.

1. raw_airbnb with the pattern = ‘.*[0-9]_[0-9]*.csv’
2. raw_g01census with the pattern = ‘.*2016Census_G01_NSW_LGA.csv'
3. raw_g02census with the pattern = '.*2016Census_G02_NSW_LGA.csv'
4. raw_suburb with the pattern ‘.*NSW_LGA_SUBURB.csv’
5. raw_code with the pattern = ‘.*NSW_LGA_CODE.csv’

For verifying whether the data has been processed correctly, display only the first row of each table to see the nested column values.

b) Schema Staging

This step plays an important role as it involves creating a data table by flattening and parsing the existing fields from the raw tables using appropriate column types.

Each raw table must be staged so to be used as dimension tables in the next phase of building a data warehouse.

Example: Create table staging_listing by parsing the first row from raw_airbnb as follows:

Existing Field	Data Type/Parsing Method	New Column Name
C1	VARCHAR	listing_id
C4	VARCHAR	host_id
C5	VARCHAR	host_name
C6	VARCHAR	host_since
C7	CHAR	host_superhost
C8	VARCHAR	host_neighbourhood
C9	VARCHAR	listing_neighbourhood
C10	VARCHAR	propety_type
C11	VARCHAR	room_type
C12	INT	accommodates
C13	INT	price
C14	CHAR	has_availability
C15	INT	availability_30
C16	INT	number_of_reviews
C17	INT	review_scores_rating
C18	INT	review_scores_accuracy
C19	INT	review_scores_cleanliness
C20	INT	review_scores_checkin
C21	INT	review_scores_communication
C22	INT	review_scores_value
metadata$filename	SUBSTR(metadata$filename, 6, 7)	month_year

Repeat the process for 4 other tables using the names:

- staging_code from raw_code
- staging_suburb from raw_suburb
- staging_g01 from raw_g01census
- staging_g02 from raw_g02census

Note: While the general rule for building data warehouse is to parse every column in a raw table, for the purpose of this assignment I have taken columns C1 to C37 for the G01 Census table as those are of interest.

c) Dimension and Fact Table Creation

For the purpose of creating this warehouse, it will be organised using a star schema. A star schema uses a single large fact table to store transactional data or measured data, and around 4 smaller dimensional tables that store attributes of the data, forming the points of a star.

For this project, there are five dimension tables:

1) **Dimension Listing:** With primary key as listing_id, containing all details about a particular listing.
2) **Dimension Hosting**: With primary key as host_id, containing all details about the hosts.
3) **Dimension LGA:** With primary key as lga_code, containing a joined table with lga_suburb and lga_code.
4) **Dimension CensusG01:** With primary key as lga_code and all other columns from the censusG01 table.
5) **Dimension CensusG02:** With primary key as lga_code and all other columns from the censusG02 table.

The one fact table contains foreign-key references to all the primary keys from the dimension tables. A rough diagrammatic understanding of the schematic relationship between these tables is described as below. The schema for the dimension tables can be found in (Appendix).

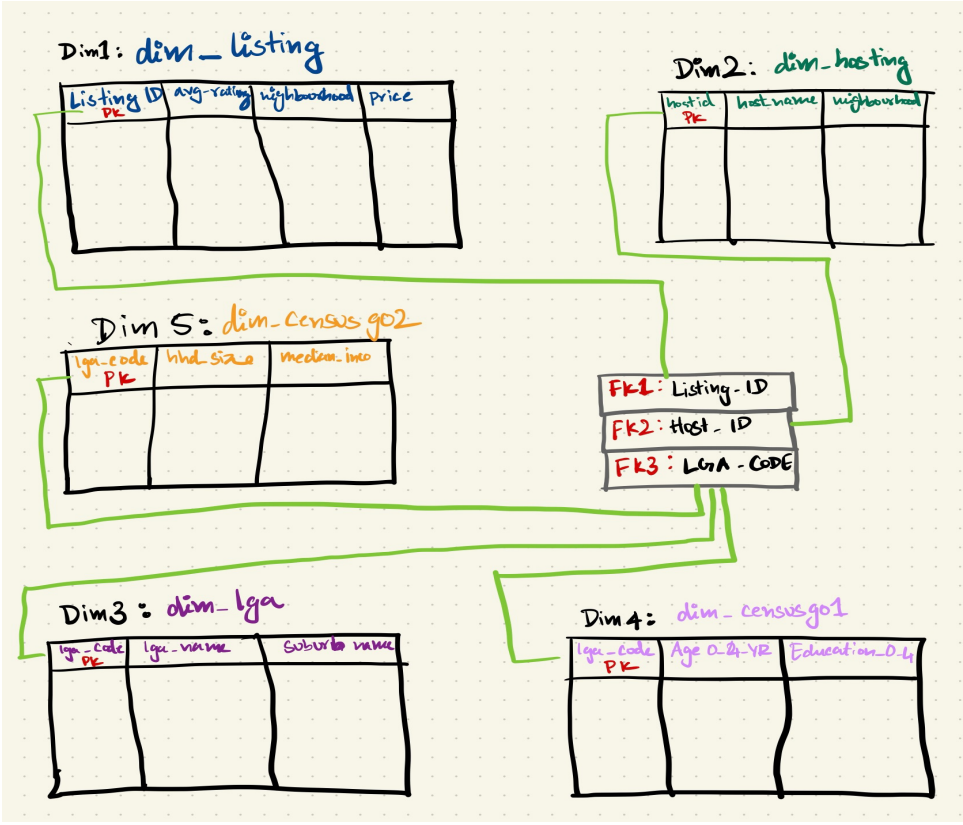


Fig 2: Star Schema Structure for Data Warehouse

d) Data Mart Creation

The three data marts created, dm_listing_neighbourhood, dm_host_neighbourhood, and dm_property_type provide a summarized breakdown of the Airbnb data which can be used to attempt the adhoc questions.

These datamarts were created using a combination of the existing dimension tables and a derivation of the dimensions as smaller datamarts to calculate percentage change in active and inactive listings, total number of stays and estimated revenue, as well as superhost rates.

Breaking down of the entire query creation into smaller tables and joining them to form the datamart allowed for accurate results.

These datamarts then need to be ordered as per instructions given an exported into individual *'.csv'* files.

6. ELT Pattern with Airflow (Part 2)

Once the data warehouse is setup and designed on Snowflake, Airflow can be used to automate the build of a data pipeline. A general DAG script consists of the queries used, classified as tasks, their dependencies and the flows between each task. It is crucial to order the tasks correctly as it can hinder the creation of the warehouse by running the SQL queries incorrectly.

Data Flows in a DAG must only flow in a single direction and avoid any loops, i.e. a later task is dependent on a previous task, creating a deadlock situation.

After importing the necessary modules into the DAG script and configuring the DAG settings as per system requirements, all the SQL queries of designing a data warehouse must be copied in the right order. Each query has its own task_id which will then be used to dictate the order of flow. In the final part of the script, the order of operations are declared to inform Airflow on the execution process.

In case of errors within the DAG script, the logs are available in the Airflow UI.

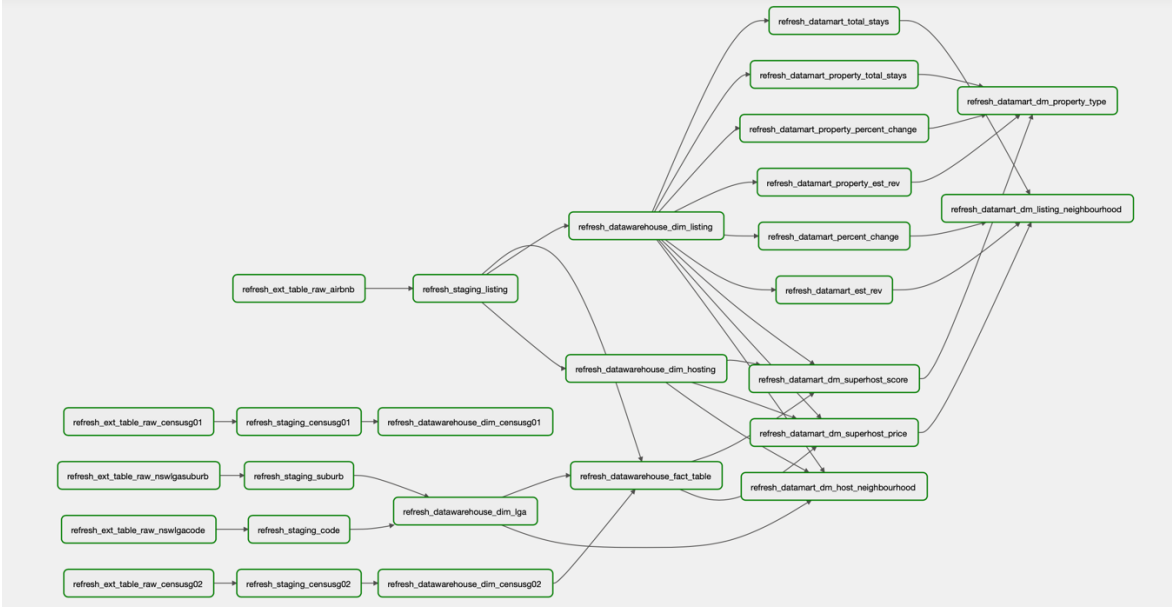


Fig 3: Successful running of the DAG

7. Ad-hoc Analysis (Part 3)

A) We first retrieve the best and worst performing neighbourhood in terms of estimated revenue using the previously created data mart. The estimated revenue over the last 12 months can be calculated using a rolling summation where only the final value from April 2021 will be considered.

This yields the result that Mosman does best with an estimated annual revenue of *AUD$111,922* and Fairfield is worst performing with approximately *AUD$15713.*

27	canterbury-bankstown	2021-04-01	18,006.128198
28	blacktown	2021-04-01	15,779.340585
29	fairfield	2021-04-01	15,713.924124

Fig 4: Worst Performing Listing Neighbourhood

	LISTING_NEIGHBOURHOOD	···	MONTH_YEAR	EST_REV_OVER_12_MONTHS
1	mosman		2021-04-01	111,922.592512
2	northern beaches		2021-04-01	91,062.36946
3	woollahra		2021-04-01	82,785.664014
4	hunters hill		2021-04-01	70,606.577592

Fig 5: Best Performing Listing Neighbourhood

Assuming that the general population that rent out or prevail the services of Airbnb fall within the age categories of 20 years to 64 years it is interesting to note that Mosman has a significantly lower population than Fairfield.

Closely inspecting the "Under 30s" age, we can notice that Fairfield has a population nearly 10 times more than Mosman. The difference in revenue could be attributed to various external factors such as facilities and accessibility within a neighbourhood. Fairfield is almost an hour away from the city while Mosman is just a quick 15-minute taxi away. These factors play a huge role on the popularity of Airbnbs.

	LGA_NAME	···	AIRBNB_POPULATION	UNDER_30S
1	mosman		15,158	4,687
2	fairfield		103,937	42,425

Fig 6: Population Based View

B) We first retrieve the top 5 listing neighbourhoods based on their total estimated revenue. Saving that as a temporary view, we can then rank these neighbourhoods for the type of property, type of room, and accommodation size ordered by their total number of stays.

	LISTING_NEIGHBOURHOOD ···	PROPERTY_TYPE	ROOM_TYPE	ACCOMMODATES	TOTAL_REV	STAYS
1	northern beaches	Entire apartment	Entire home/apt	2	174,572,869.74162	5,686,044,476
2	mosman	Entire apartment	Entire home/apt	2	27,335,624.45019	890,353,562
3	hunters hill	Entire apartment	Entire home/apt	4	2,616,535.96425	47,638,284
4	woollahra	Entire apartment	Entire home/apt	2	81,069,056.50158	2,640,514,884
5	waverley	Entire apartment	Entire home/apt	2	288,291,846.171	9,390,005,800

Fig 7: Top 5 listing neighbourhoods

It is evident that the best kind of property type or room type for all the top neighbourhoods is an entire apartment or home that can accommodate between 2 to 4 people.

C) Upon initial analysis to get an understanding of the number of listings for each host, we can see that there are around 5,095 rows with hosts as having as less as 2 listings and as many as 496.

	NO_OF_LISTINGS	HOST_ID
5088	124	15739069
5089	144	16357713
5090	146	15469257
5091	188	235137306
5092	206	7409213
5093	210	279001183
5094	356	175128252
5095	496	194230296

Fig 8: Hosts with Multiple Listings

For those hosts with multiple listings, there is a general pattern noticed where most of the hosts do have their listing in the same LGA as their own neighbourhood.

		··· NO_OF_LISTINGS	LISTING_LGA	HOSTING_LGA
12		2	sydney	sydney
13		2	northern beaches	northern beaches
14		2	inner west	inner west
15		2	north sydney	north sydney
16		2	sydney	sydney
17		2	waverley	waverley
18		4	inner west	inner west
19		2	waverley	waverley

Fig 9: Hosts living in same areas as listing

With further inspection on certain hosts with multiple listings they have at-least 1 listing in their neighbourhood LGA. This could be attributed to the fact of convenience and ease of

maintenance. For example, the following host has only 2 listings and live in Sydney but has 1 listing based in Sydney and the other in North Sydney.

	HOST_ID	HOST_NEIGHBOURHOOD
6	324966918	null
7	324966918	null
8	324966918	null
9	324966918	null
10	324966918	null
11	324966918	sydney
12	324966918	sydney

Fig 10: Host's neighbourhood

	LISTING_ID	LISTING_NEIGHBOURHOOD
1	41829063	north sydney
2	41515837	sydney

Fig 11: Host's listing's neighbourhood

Similarly, this other host with around 20 listing lives in the Inner West LGA and has a few of their listings in the same area, while a majority is spread around Sydney.

	LISTING_ID	LISTING_NEIGHBOURHOOD
1	29016777	inner west
2	33274326	inner west
3	7464305	northern beaches
4	26308228	inner west
5	21771768	sydney
6	26305702	sydney
7	33859891	sydney
8	30773052	sydney

Fig 12: Host's listing's neighbourhood

The need for expansion as the number of listings grow is understandable from a profit point of perspective. Growing competition in certain areas can also be a reason for moving out of your own LGA and into un-ventured zones.

D) To approach this analysis, two views can be created to analyse the annual revenue and mortgage repayment rate for hosts and their listings in a particular neighbourhood. Hosts with unique listings alone yield around 20K rows.

	HOST_ID	ANNUAL_REVENUE	···	ANNUAL_MORTGAGE_REPAY	LISTING_NEIGHBOURHOOD
1	2281364	1,727,640		374,400	randwick
2	3796216	87,840		432,000	waverley
3	1440732	593,030		360,000	waverley
4	2315641	4,693,320		403,200	northern beaches
5	4059587	255,420		343,200	inner west
6	1440754	32,400		374,400	randwick
7	1468090	787,920		432,000	waverley
8	2650727	451,176		432,000	mosman
9	1340280	124,164		374,400	inner west
10	2874517	5,760		432,000	waverley
11	6489984	432,204		403,200	northern beaches
12	9898323	65,904		403,200	northern beaches
13	10459365	753,900		359,856	sydney
14	4674300	190,608		432,000	waverley

Fig 13: Host's annual revenue and mortgage

The annual revenue is calculated by taking a rolling summation over the 12-month period, and we only look at the final value from April 2021.

Our of the 20,998 hosts around 10,701 or 51% of them can cover the annualised median mortgage repayment rate.

	···	COUNT(*)	CAN_REPAY
1		20,998	10,701

Fig 14: Hosts that can cover

However, we must also consider that the census data reflects on 2016 median mortgage repayment values, and factors such as inflation could've contributed towards raising it. The listing data is also from the time of pandemic where travel was at an all-time low, and companies like Airbnb, faced a severe hit.

8. Possible Issues & Bugs

- While creating the dimension for hosting, the original dataset has a STRING datatype for the column “host_since” which must be converted and parsed into the appropriate date format so that it can be used for calculations.

➢ This can be achieved by using the *TO_DATE* function.

- The census tables contain the LGA code with a prefix LGA-, which prohibits joins on other tables if necessary. Thus, a new column can be created in the dimension which extracts only the numeric code.

➢ This can be achieved by using the *SUBSTR* function.

- Prior to running the DAG script, it is generally recommended to truncate all existing tables to verify the proper working of the code.

9. Conclusion

Designing a data pipeline with Airflow has numerous advantages. The potential of implementing these pipelines with Airflow using Python scripts enables one to build arbitrarily complex pipelines that can carry one’s desired tasks seamlessly. However, these pipelines can only perform as good as the underlying infrastructure supporting it, which is why it is crucial to get the order of dependencies correct. Analysing and attempting to answer the business questions for the given datasets yielded interesting results. An updated census dataset along with latest Airbnb listing details can provide more up-to-date information on the current status of the company’s performance in the Bed & Breakfast industry.

10. Appendix

Appendix 1: Schema of Dimension Tables

Dimension LGA

	name	type	kind
1	LGA_CODE	VARCHAR(16777216)	COLUMN
2	LGA_NAME	VARCHAR(16777216)	COLUMN
3	SUBURB_NAME	VARCHAR(16777216)	COLUMN

Dimension CensusG02

	name	type	kind
1	CLEAN_LGA_CODE	VARCHAR(16777216)	COLUMN
2	LGA_CODE	VARCHAR(16777216)	COLUMN
3	MEDIAN_AGE	NUMBER(38,0)	COLUMN
4	MEDIAN_MORTGAGE_REPAY_MONTHLY	NUMBER(38,0)	COLUMN
5	MEDIAN_TOT_PRSNL_INC_WEEKLY	NUMBER(38,0)	COLUMN
6	MEDIAN_RENT_WEEKLY	NUMBER(38,0)	COLUMN
7	MEDIAN_TOT_FAM_INC_WEEKLY	NUMBER(38,0)	COLUMN
8	AVG_NUM_PSNS_PER_BEDROOM	NUMBER(38,0)	COLUMN
9	MEDIAN_TOT_HHD_INC_WEEKLY	NUMBER(38,0)	COLUMN
10	AVG_HOUSEHOLD_SIZE	NUMBER(38,0)	COLUMN

Dimension CensusG01

	name	type	kind
1	CLEAN_LGA_CODE	VARCHAR(16777216)	COLUMN
2	LGA_CODE	VARCHAR(16777216)	COLUMN
3	TOTAL_POP_M	NUMBER(38,0)	COLUMN
4	TOTAL_POP_F	NUMBER(38,0)	COLUMN
5	TOTAL_POP_P	NUMBER(38,0)	COLUMN
6	AGE_0_4_YR_M	NUMBER(38,0)	COLUMN
7	AGE_0_4_YR_F	NUMBER(38,0)	COLUMN
8	AGE_0_4_YR_P	NUMBER(38,0)	COLUMN
9	AGE_5_14_YR_M	NUMBER(38,0)	COLUMN
10	AGE_5_14_YR_F	NUMBER(38,0)	COLUMN
11	AGE_5_14_YR_P	NUMBER(38,0)	COLUMN

Dimension Hosting

	name	type	kind
1	HOST_ID	VARCHAR(16777216)	COLUMN
2	HOST_NAME	VARCHAR(16777216)	COLUMN
3	HOST_SINCE	DATE	COLUMN
4	HOST_SUPERHOST	VARCHAR(1)	COLUMN
5	HOST_NEIGHBOURHOOD	VARCHAR(16777216)	COLUMN

Dimension Listing

	name	type
1	LISTING_ID	VARCHAR(16777216)
2	HOST_ID	VARCHAR(16777216)
3	LISTING_NEIGHBOURHOOD	VARCHAR(16777216)
4	PROPERTY_TYPE	VARCHAR(16777216)
5	ROOM_TYPE	VARCHAR(16777216)
6	ACCOMMODATES	NUMBER(38,0)
7	PRICE	NUMBER(38,0)
8	HAS_AVAILABILITY	VARCHAR(1)
9	AVAILABILITY_30	NUMBER(38,0)
10	NUMBER_OF_REVIEWS	NUMBER(38,0)
11	REVIEW_SCORES_RATING	NUMBER(38,0)
12	MONTH_YEAR	VARCHAR(16777216)

Appendix 2: Pypi Packages and Airflow Configurations

Required libraries from the Python Package Index (PyPI)

Name	Version
pandas	-
snowflake-connector-python	==2.4.5
snowflake-sqlalchemy	==1.2.4
apache-airflow-providers-snowflake	==1.3.0

The configuration properties are stored in a file called airflow.cfg the first time you run Apache Airflow. You can choose to change these properties and override the default values.

core

enable_xcom_pickling True

Appendix 3: Custom GCP role and Permissions

Appendix 4: Data Warehouse on Snowflake