**Big Data Engineering**

**Assignment 3: Building ELT data pipelines with Airflow**

**Aim:**

The goal of this assignment is to build production-ready data pipelines with Airflow. You will work with two different input datasets that will need to be processed and cleaned before loading this insightful information separately into a data warehouse (using ELT pipelines) and a data mart for analytical purposes.

**Introduction to the datasets**

1. **Airbnb**

Airbnb is an online-based marketing company that connects people looking for accommodation (Airbnb guests) to people looking to rent their properties (Airbnb hosts) on a short-term or long-term basis. The rental properties include apartments (dominant), homes, boats, and a whole lot more. As of 2019, there are 150 million users of Airbnb services in 191 countries, making it a major disruptor of the traditional hospitality industry (this is akin to how Uber and other emerging transportation services have disrupted the traditional intra-city transportation services). As a rental ecosystem, Airbnb generates tons of data including but not limited to: density of rentals across regions (cities and neighbourhoods), price variations across rentals, host-guest interactions in the form of reviews, and so forth.

We will focus on Sydney for this assignment, you can find the original data and more information on this [link](http://insideairbnb.com/get-the-data.html), however the website is purging the available data regularly.

The modified dataset used in this assignment is from May 2020 to April 2021.

1. **Census**

The Census of Population and Housing (Census) is Australia’s largest statistical collection undertaken by the Australian Bureau of Statistics (ABS). For more than 100 years, the Census has provided a snapshot of Australia, showing how the country has changed over time, allowing it to plan for the future. The aim of the Census is to accurately collect data on the key characteristics of people in Australia on Census night and the dwellings in which they live. In 2016, the Census counted close to 10 million dwellings and approximately 24 million people, the largest number counted to date.

The information provided in the Census helps estimate Australia’s population, which is used to distribute government funds and plan services for the community – housing, transport, education, industry, hospitals and the environment. Census data is also used by individuals and organisations in the public and private sectors to make informed decisions on policy and planning issues that impact the lives of all Australians.

You can find the original dataset and find more information on this [link](https://datapacks.censusdata.abs.gov.au/datapacks/).

**Tasks:**

You will have to set up an Airflow and Postgres environment using GCP (Cloud Composer and SQL instance) and dbt with a **private** github repo.

**Part 0: Download the datasets:**

1. 12 months of Airbnb listing data for Sydney: [link](https://drive.google.com/file/d/1_AvGzOLrCNCnDJyStSj2XH0bTUtsKgb_/view?usp=sharing)
2. The tables G01 (“Selected Person Characteristics by Sex”) and G02 (“Selected Medians and Averages”) of the General Community Profile Pack from the 2016 census at the LGA level: [link](https://drive.google.com/file/d/1AbfLWOCgPfAY8bBRX1blZdL0-dO2joXT/view?usp=sharing).
3. A dataset to help you join both datasets based on LGAs code and a mapping between LGAs and Suburbs: [link](https://drive.google.com/file/d/1y962EkNhG2nBGiMsV8sYN2BeFsIy6zO5/view?usp=sharing/view?usp=sharing).

**Part 1: Use Airflow to load raw data into Postgres**

1. Upload the dataset into the AirFlow storage bucket
2. Create a raw schema on Postgres and the relevant raw tables which will contain the raw data using DBeaver.
3. Create an Airflow Dag (set the schedule\_interval to None) which will read the data from the storage bucket and load the raw data into the raw schema.

**Part 2: Design a data warehouse using dbt**

1. Design the architecture of a data warehouse on Postgres with 4 layers (raw, staging, warehouse, datamart). You will also need to have at least 4 dimensions tables (e.g. listing, host, suburb, lga,etc…) + the two tables from the census as dimension tables in your warehouse layer. The 4 layers will have the following:
   1. Raw: contains the raw tables + snapshots of dimensions with strategy based on timestamp (<https://docs.getdbt.com/reference/resource-configs/strategy>). The snapshot models will have the following naming convention: **name\_snapshot** (i.e. property\_type\_snapshot).
   2. Staging: **Cleaning/transformations and renaming** of raw/snapshot data. You will find some issues around the listing dates and the LGAs. Models in this layer need to be materialised as views. The staging models will have the following naming convention: **name\_stg** (i.e. property\_type\_stg).
   3. Warehouse: Star schema with dimensions and fact tables. In this assignment you can directly drop the dimensions id in the fact table and bring the descriptions/labels. The fact table is expected to have taken into consideration the SCD dimensions. Models need to be materialised as tables. The warehouse models will have the following naming convention: **dim\_name** or **facts\_name** (i.e. dim\_property\_type or facts\_listings).
   4. Datamart : This is where the answers to the following questions will live. Needs to be materialised as views.
2. **For the datamart, create the 3 following views:**
   1. Per “**listing\_neighbourhood**” and “**month/year**”:

* 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

The view needs to be ordered by “listing\_neighbourhood” and “month/year”

**Named it “dm\_listing\_neighbourhood”**

* 1. Per “**property\_type**”, “**room\_type**” ,“**accommodates**” and “**month/year**”:
* 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

The view needs to be ordered by “property\_type”, “room\_type” ,“accommodates” and “month/year”

**Named it “dm\_property\_type”**

* 1. Per “**host\_neighbourhood\_lga**” which is “**host\_neighbourhood” transformed to an LGA** (e.g host\_neighbourhood = 'Bondi' then you need to create host\_neighbourhood\_lga = 'Waverley') and “**month/year**”:
* Number of distinct host
* Estimated Revenue
* Estimated Revenue per host (distinct)

The view needs to be ordered by “host\_neighbourhood\_lga” and “month/year”

**Named it “dm\_host\_neighbourhood”**

Some definitions:

* Active listings = listing with "has\_availability" = "t"
* Active Listing Rate = (total Active listings / total listing) \* 100
* Superhost Rate = (total distinct hosts with "host\_is\_superhost" = 't' / total distinct hosts) \* 100
* Percentage change (month to month) = ((final value - original value) / original value) \* 100
* Number of stays (only for active listings) = 30 - availability\_30
* Estimated revenue per active listings = for each active listing/period: number of stays \* price
* Estimated revenue per host= Total Estimated revenue per active listings/ total distinct hosts

1. **Push your dbt project to your private Github Repo and share it with William (william.so@uts.edu.au) and Alice (**[**alice.finidori@uts.edu.au**](mailto:alice.finidori@uts.edu.au)**).** If none of us has received your github repo link, it will be considered missing.

Advises:

Truncate all tables before running your dag for the first time.

Be careful of the order of operation, especially when loading dimension and fact data.

Be careful with SCD in the facts table (use the snapshots models).

Use “dbt run --full-refresh” if you want to recreate every model.

**Part 3: Ad-hoc analysis**

Answer the following questions with supporting results (write SQL on Postgres):

1. 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?
2. 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?
3. Do hosts with multiple listings are more inclined to have their listings in the same LGA as where they live?
4. 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”?

Add the answers with screenshots in your report.

This is an individual assignment, each student will be marked individually.

**Deliverables:**

Each student will have to submit

* SQL queries (.sql files) used for Part 1 on Postgres in a single file named “**part\_1.sql**”.
* An Airflow DAG script for Part 1 (you can only have 1 dag file)
* SQL queries (.sql files) used for Part 3 on Postgres in a single file named “**part\_3.sql**”.
* Private Github repo containing your dbt project (send invite to teachers, no need to include it the submission files)
* A “handover” written report
* Any other relevant files or documents

**The report should not exceed 3000 words** (figures and tables are not counted).

Compress all deliverables into a single zip file and use the following file naming format for the submission:

**Assignment\_3\_FirstName\_LastName.zip**

A good “handover” report should contained:

1. High-level view of your project.
2. Explanation for the different steps of your project.
3. Any issues/bugs you faced and how you solved them.
4. Answers to the business questions with supporting evidence.
5. Relevant screenshots/images/diagrams/flows if necessary.

You can assume that the reader of your report will have a similar understanding and knowledge of any technical skills (Python/SQL).

**Assessment Criteria:**

* Quality of code (Python/SQL/dbt).
* Justification of data transformation, data formats, data storage, DAGs structure and accuracy of results with evidence supporting claims
* Quality of findings and recommendations for business questions.
* Clarity and quality of written report.

**Criteria Details and weights:**

| Criteria | Further Details | Weight |
| --- | --- | --- |
| Quality of code (Python/SQL/dbt) | 1. Code can be executed without raising an error and is well commented 2. Datawarehouse is correctly set up (layers, fact, dimension, SCD). 3. Data Mart is kept up to date | 40% |
| Justification of data transformation, data formats, data storage, DAGs structure, accuracy of results with evidence supporting claims. | 1. High level explanation of each major step and decision. 2. Follows the good “handover” report guidelines | 30% |
| Quality of findings and recommendations for business questions. | 1. Correct answers to the business questions. 2. Recommendations to the business are relevant. | 20% |
| Clarity and quality of written report. | 1. Complete and professionally formatted report (spelling, grammar, punctuation, layout). 2. Report is not exceeding the maximum length | 10% |

**Due Date:**

All assignments need to be submitted before the **due date (31st October)** on Canvas. Penalties will be applied for late submission

**Early Submission Bonus:**

1. + 15 point if submitted within the first week (before Tuesday 17th of October at 23:59)
2. + 10 point if submitted within the second week (before Tuesday 24th of October at 23:59)
3. + 5 point if submitted within the third week (before Tuesday 31st of October at 23:59)