**Project ETL**

1. **Sources of data for extraction:**

The identified data sources are:

* **Air B&B Mexico City**: Contains the information of all Air B&B locations in Mexico City registered until October 2019.
  + Variables:
    - *id*
    - *name*
    - *host\_id*
    - *host\_name*
    - *neighbourhood\_group*
    - *neighbourhood*
    - *latitude*
    - *longitude*
    - *room\_type*
    - *price*
    - *minimum\_nights*
    - *number\_of\_reviews*
    - *last\_review*
    - *reviews\_per\_month*
    - *calculated\_host\_listings\_count*
    - *availability\_365*
* **Starbucks Locations Worldwide**: This dataset includes a record for every Starbucks or subsidiary store location currently in operation until February 2017.
  + Variables:
    - *Brand*
    - *Store Number*
    - *Store Name*
    - *Ownership Type*
    - *Street Address*
    - *City*
    - *State/Province*
    - *Country*
    - *Postcode*
    - *Phone Number*
    - *Timezone*
    - *Longitude*
    - Latitude
* **Postal Codes**: Provides the existing postal code for each settlement and municipality in Mexico (Country).
  + Variables
    - *Código Postal*
    - *Estado*
    - *Municipio*
    - *Ciudad*
    - *Tipo de Aseentamiento*
    - *Asentamiento*
    - *Clave de oficina*

1. **Type of transformation needed for this data (cleaning, joining, filtering, aggregating, etc).**

* **Cleaning:**
  + Postal Codes:
    - Change name columns to English
* **Filtering:**
  + Postal Codes:
    - Drop Settlement Type and Office Code
    - Filter only Mexico City Postal Codes
  + AirB&B:
    - Keep name, neighbourhood, latitude, longitude, room\_type and price.
  + Starbucks:
    - Keep store name, street address, state/province, country, postcode, longitude, latitude.
    - Filter only country = MX for Mexico
    - Filter only state/province = DIF for Mexico City Starbucks locations
* **Transform:**
  + Starbucks:
    - Rename the variables without spaces so SQL alchemy run smoothly
  + All tables:
    - Create an index variable as id for each table

1. **Load the information into an integrated (joined) database from Starbucks, Air B&B and Postal Codes:**

* **Generate tables into database:**
  + Create Starbucks, AirB&B and Postal Codes table in a database in pgAdmin application.
* **Join the three tables**
  + Use AirB&B table (with name, neighbourhood, latitude, longitude as variables) as the pivot to join the information from Starbucks shops (with store name, street address and post code). The joining variable is based on the location (longitude and latitude).
  + Take output from previous step and join the settlement and municipality name from postal codes data from postal codes database.
  + Now a new table with the information of Starbuck close to the Air B&B available locations is done
  + The table is exported to CSV file
  + Some locations have more than one Starbucks shop available, in which case the Air B&B is repeated in the table with different Starbucks locations. Some locations do not have any Starbucks close, in which case the information of Starbucks locations is empty.
  + All records of Air B&B are maintained.
* **Bring the tables to Pandas**
  + Use pandas SQL connection to populate each table (Starbucks, AirB&B and Post Codes) into SQL database.
  + Confirm that all tables are contained in database.
  + Fill tables (Starbucks, AirB&B and Post Codes) with the values from dataframes generated in pandas.
  + Run a query to confirm that data has been added to each of the three tables.