**Project-2: ETL Final Report**

*Berta Devenyi, Deepa Nair, Kevin McConchie, Richard Luong*

**In order to query postgres through jupyter notebook, a config file with variables “pg\_user” and “pg\_pass” is required, with respective fields entered.**

**In postgres, a database with the name “Melb\_data\_db” must be created in order to run the jupyter notebook without errors.**

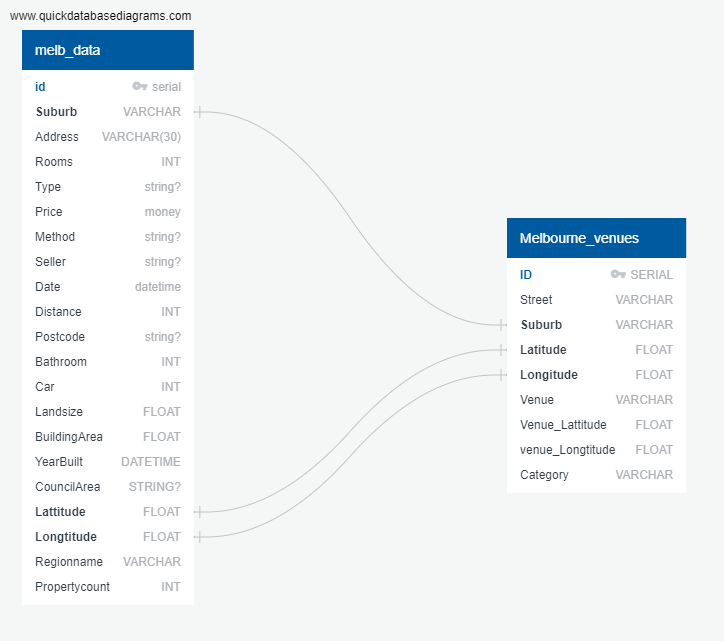
**Data sets used and their sources:**

[Melbourne Venues | Kaggle](https://www.kaggle.com/datasets/patelkeviin/melbourne-venues)

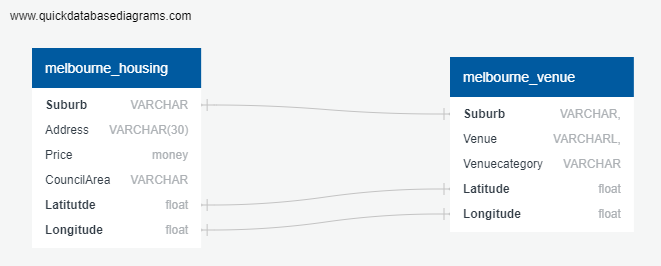
CSV file used as the source of our melbourne “home” data

[Melbourne data | Kaggle](https://www.kaggle.com/datasets/hebatarek/melbourne-data)  
CSV file used as the source of our melbourne “venue” data

**EBD diagram:**

****

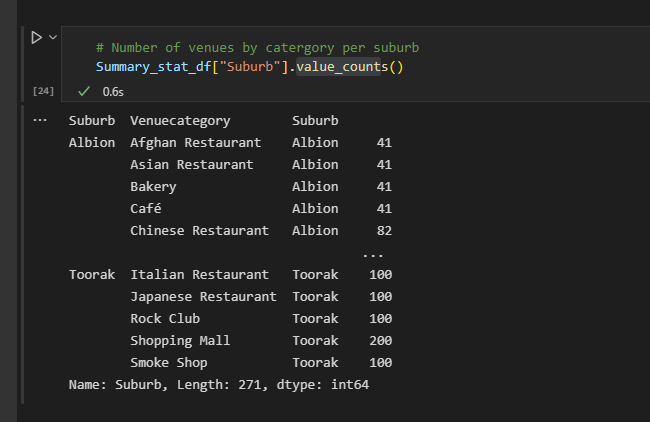
First EBD diagram based on the CSV files imported from kaggle. The CSV files were cleaned, where column names that were deemed unsuitable to draw relations between were dropped.

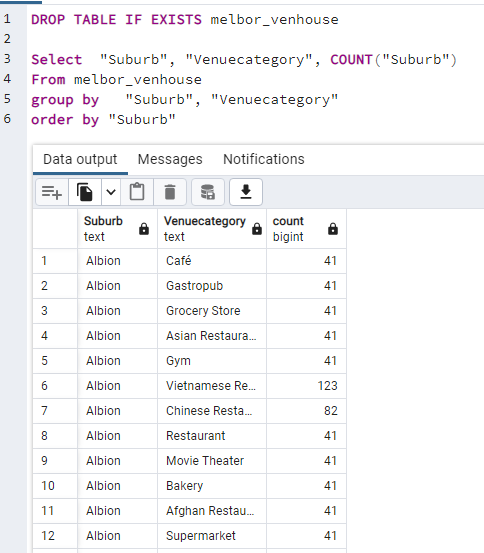


Subsequent diagram, depicting a relational database where the relationships are made up of “Suburb” as well as “Latitude” and “Longitude”.

**Types of data wrangling performed**

1. Imported the both csv files into data frames using Pandas.
2. Created two new tables from the data frames for the data we were interested
3. Made sure both tables included columns for primary keys.
4. Corrected misspelt column names.
5. Loaded the new tables into our sql database by creating an engine.
6. Checked engine for tables to ensure we had a connection
7. Pushed the table information into sql with to\_sql function
8. Ran sql queries using the read\_sql\_query function. These queries include:
   1. Left join and query
   2. Selecting the max price in melbor\_housing, grouped by suburb
   3. Looking for venues where house price is <2M and has >4 rooms





In the screenshots above, a query made in SQL is also made in our jupyter notebook using pandas, where we created an engine to access the database made in sql.

Our full list of queries are listed as below:

DROP TABLE IF EXISTS melbor\_venhouse

Select "Suburb", "Venuecategory", COUNT("Suburb")

From melbor\_venhouse

group by "Suburb", "Venuecategory"

order by "Suburb"

SELECT "Suburb", MAX("Price") AS "MAX price"

FROM melbor\_housing

GROUP BY "Suburb";

SELECT melbor\_venue."Venue", melbor\_venue."Venuecategory", melbor\_venue."Suburb", melbor\_housing."Price", melbor\_housing."Rooms"

FROM melbor\_venue

LEFT JOIN melbor\_housing ON

melbor\_venue."Suburb" = melbor\_housing."Suburb"

and melbor\_venue."Latitude" = melbor\_housing."Lattitude"

and melbor\_venue."Longitude" = melbor\_housing."Longtitude"

where melbor\_housing."Rooms" >3 ;

SELECT "Suburb", count("Venuecategory") AS "MAX category"

FROM melbor\_venue

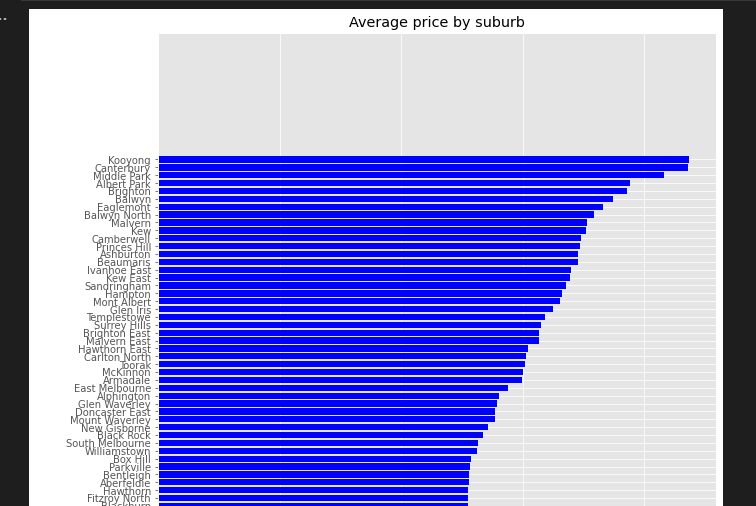
GROUP BY "Suburb";

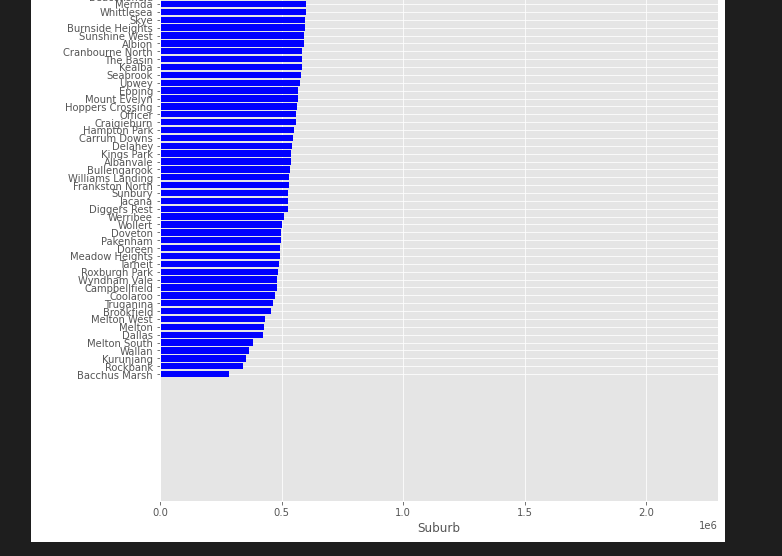
SELECT "Suburb", MAX("Price") AS "MAX price"

FROM melbor\_housing

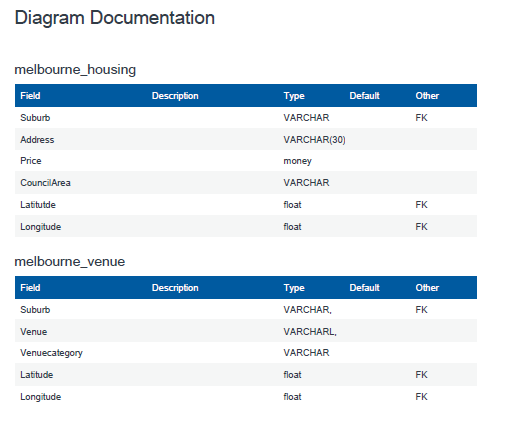
GROUP BY "Suburb";

1. Used matplot lib to plot a graph depicting average price by council, in ascending order, (only the top and bottom of graphs shown)





**Final Schema**

****