ETL Project Report: Australian renewable energy installations history and their respective output database in relation with the solar post code's ratings.

Contributors: Arron Nguyen, Kiran Bano, Zohaib Hashmi

Project Proposal

Aim of the project is to make a relational database for Australian renewable energy / small scale installations that could be referred back to post code data, zones and their ratings using ETL process.

For the purpose, we analysed the data of previous years from http://www.cleanenergyregulator.gov.au/RET/Forms-and-resources/Postcode-data-for-small-scale-installations#SGU--Solar-Deemed. Data showed interesting trends as shown in Figure 1, number of installations was booming between 2009 and 2011 and after dipping down until 2016 number of installations started increasing again from 2017-2019. In 2020, major decrease was observed (~85000 installations) across Australia that can be attributed to the global pandemic.

Based on these facts we decided to select post code data of 2017, 2018 and 2019 for our project. Analysis of data also showed that hydro and wind power installations were negligible as compared to solar powered installations. Therefore, the post code zones and rating data of solar installations were also extracted to be included into our database.

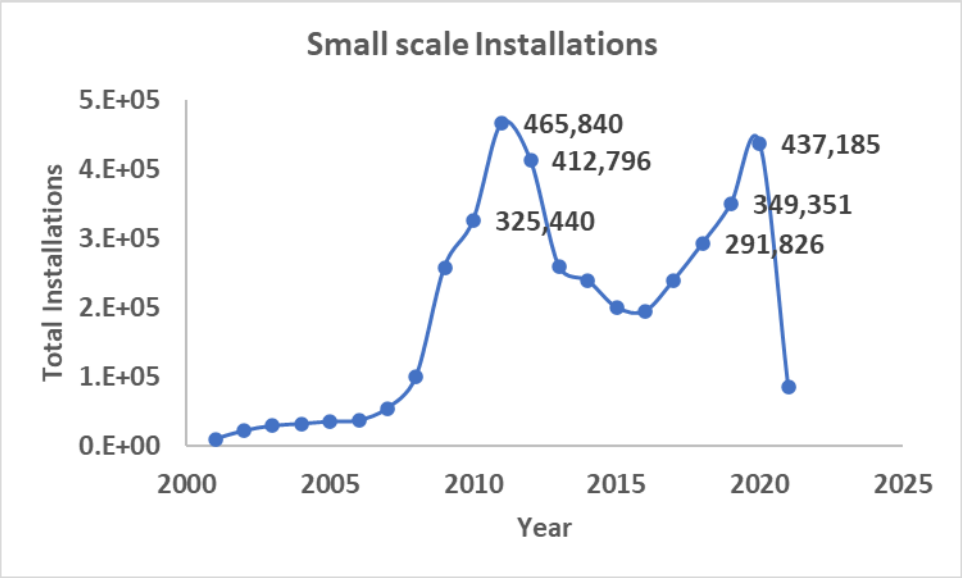


Figure 1: Graphical representation of number of small-scale installations in last decades across Australia.

Analysis Tools:

- Python Pandas, PostgreSQL, Excel

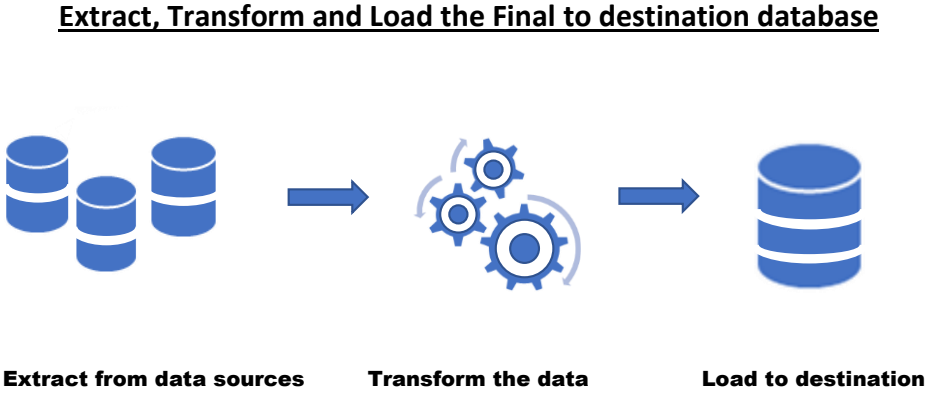


Figure 2: Schematic representation of ETL process used in this project

Extract:

All CSV files were read and loaded into a data frame.

- Source for renewable energy installations: http://cleanenergyregulator.gov.au/RET/Forms-and-resources/Postcode-data-for-small-scale-[…]/historical-postcode-data-for-small-scale-installations
- Source for Australian post code data: https://gist.github.com/randomecho/5020859
- Source for post code zones rating: http://www.cleanenergyregulator.gov.au/DocumentAssets/Pages/Postcode-zone-ratings-and-postcode-zones-for-solar-panel-systems.aspx

Transform:

- CSV files for renewable energy installations were renamed for convenience.
- Post code rating data was in word format, data was copied into excel and saved as CSV file.
- Australian post code data was added to a table created in PostgreSQL and was formatted by removing ('\') from the names to avoid syntax error. Serial id for post code was set as primary key.

➢ For Ipynb

- All CSV files were loaded and read using Jupyter python note book. New data frames were created for wind, solar and hydro installations with the information selected from the data of 2017, 2018 and 2019.

In [227]:
```
# create df with only required info
#hydro 17 - 19
hydro_2019_cl = hydro_2019[["Small Unit Installation Postcode", "Installations Quantity Total", "SGU Rated Output In kW Total
hydro_2019_df = hydro_2019_cl

hydro_2018_cl = hydro_2018[["Small Unit Installation Postcode", "Installations Quantity Total", "SGU Rated Output In kW Total
hydro_2018_df = hydro_2018_cl

hydro_2017_cl = hydro_2017[["Small Unit Installation Postcode", "Installations Quantity Total", "SGU Rated Output In kW Total
hydro_2017_df = hydro_2017_cl
```

In [228]:
```
# df with only required info
# Wind 17 - 19
wind_2019_cl = wind_2019[["Small Unit Installation Postcode", "Installations Quantity Total", "SGU Rated Output In kW Total"]
wind_2019_df = wind_2019_cl

wind_2018_cl = wind_2018[["Small Unit Installation Postcode", "Installations Quantity Total", "SGU Rated Output In kW Total"]
wind_2018_df = wind_2018_cl

wind_2017_cl = wind_2017[["Small Unit Installation Postcode", "Installations Quantity Total", "SGU Rated Output In kW Total"]
wind_2017_df = wind_2017_cl
```

In [229]:
```
# df with only required info
# Solar 17 - 19
solar_2019_cl = solar_2019[["Small Unit Installation Postcode", "Installations Quantity Total", "SGU Rated Output In kW Total
solar_2019_df = solar_2019_cl

solar_2018_cl = solar_2018[["Small Unit Installation Postcode", "Installations Quantity Total", "SGU Rated Output In kW Total
solar_2018_df = solar_2018_cl

solar_2017_cl = solar_2017[["Small Unit Installation Postcode", "Installations Quantity Total", "SGU Rated Output In kW Total
solar_2017_df = solar_2017_cl
```

- Columns were renamed to make it simple and beautiful for example:

In [230]:
```
# raname hydro
hydro_2019_df = hydro_2019_df.rename(columns={          "Small Unit Installation Postcode": "Postcode",
                                                        "Installations Quantity Total": "Installations Hydro 2019",
                                                        "SGU Rated Output In kW Total": "Output In kW Hydro 2019"})

hydro_2018_df = hydro_2018_df.rename(columns={          "Small Unit Installation Postcode": "Postcode",
                                                        "Installations Quantity Total": "Installations Hydro 2018",
                                                        "SGU Rated Output In kW Total": "Output In kW Hydro 2018"})

hydro_2017_df = hydro_2017_df.rename(columns={          "Small Unit Installation Postcode": "Postcode",
                                                        "Installations Quantity Total": "Installations Hydro 2017",
                                                        "SGU Rated Output In kW Total": "Output In kW Hydro 2017"})
```

- Data of 2017, 2018 and 2019 were merged by year for wind, air and solar power data on “Post code”.

Merging DataFrame By Year

In [54]:
```
#Hydro Installation
Hydro_1 = pd.merge(hydro_2017_df,hydro_2018_df, on = "Postcode", how = "left")
Hydro = pd.merge(Hydro_1,hydro_2019_df, on = "Postcode", how = "left")
```

In [58]:
```
#  Wind Installation
Wind_1 = pd.merge(wind_2017_df,wind_2018_df, on = "Postcode", how = "left")
Wind = pd.merge(Wind_1,wind_2019_df, on = "Postcode", how = "left")
```

In [59]:
```
# solar Installation
Solar_1 = pd.merge(solar_2017_df,solar_2018_df, on = "Postcode", how = "left")
Solar = pd.merge(Solar_1,solar_2019_df, on = "Postcode", how = "left")
```

- In the merged columns, “NaN” were replaced with zero using numpy.

- Merged data frame was then split into installations and output for hydro, wind and solar power

#Splitting Installation and Output DataFrame

In [89]:
```
Hydro_installations= Hydro[["Postcode","Installations Hydro 2017", "Installations Hydro 2018", "Installations Hydro 2019"]]
```

In [66]:
```
Wind_installations= Wind[["Postcode","Installations Wind 2017", "Installations Wind 2018", "Installations Wind 2019"]]
```

In [91]:
```
Solar_installations= Solar[["Postcode","Installations Solar 2017", "Installations Solar 2018", "Installations Solar 2019"]]
```

In [93]:
```
Hydro_output= Hydro[["Postcode", "Output In kW Hydro 2017", "Output In kW Hydro 2018","Output In kW Hydro 2019"]]
```

In [94]:
```
Wind_output= Wind[["Postcode", "Output In kW Wind 2017", "Output In kW Wind 2018","Output In kW Wind 2019"]]
```

In [96]:
```
Solar_output= Solar[["Postcode", "Output In kW Solar 2017", "Output In kW Solar 2018","Output In kW Solar 2019"]]
Solar_output
```

Out[96]:

	Postcode	Output In kW Solar 2017	Output In kW Solar 2018	Output In kW Solar 2019
0	0	4.460	4.460	4.460
1	200	0.080	0.080	0.080
2	800	810.830	1,338.570	2021.050
3	801	103.008	134.868	134.868
4	803	11.685	11.685	11.685
...	...	...	...	...
2795	7469	119.395	119.395	125.445
2796	7470	97.572	103.872	146.477
2797	7802	2.450	2.450	2.450
2798	8576	6.840	6.840	6.840
2799	9729	0.875	0.875	0.875

- Solar rating data was also transformed by replacing “NaN “values with zero and resetting the index

Load:

- To load the transformed data into our database, an engine was created to connect with PostgreSQL

Create database connection

```
In [76]: #set up connection with SQL db
         connection_string = "postgres:POSTGRES*@localhost:5432/ETLPROJECT_DB"
         engine = create_engine(f'postgresql://{connection_string}')

In [77]: # Confirm tables currently in db
         engine.table_names()

Out[77]: ['postcodes_geo']
```

- All the data tables were then loaded to SQL database.
- Finally Queried to check if the DB has all the tables.

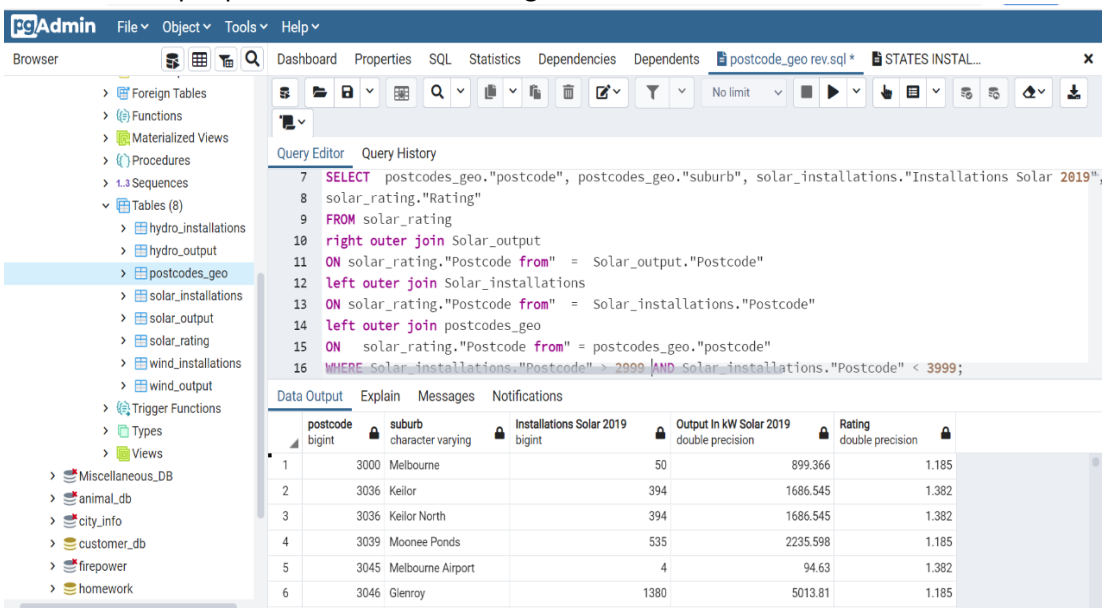
Load DataFrames into database

```
In [80]: #Load df and table into db
         Hydro_output.to_sql(name='hydro_output', con=engine, if_exists='append', index=True)

In [81]: #Load df and table into db
         Wind_output.to_sql(name='wind_output', con=engine, if_exists='append', index=True)

In [82]: #Load df and table into db
         Solar_output.to_sql(name='solar_output', con=engine, if_exists='append', index=True)

In [83]: #Load df and table into db
         Hydro_installations.to_sql(name='hydro_installations', con=engine, if_exists='append', index=True)

In [84]: Wind_installations.to_sql(name='wind_installations', con=engine, if_exists='append', index=True)

In [85]: Solar_installations.to_sql(name='solar_installations', con=engine, if_exists='append', index=True)

In [86]: #Load df and table into db
         solar_rating.to_sql(name='solar_rating', con=engine, if_exists='append', index=True)

In [87]: # Confirm tables has been loaded
         engine.table_names()

Out[87]: ['hydro_output',
          'wind_output',
          'solar_output',
          'hydro_installations',
          'wind_installations',
          'solar_installations',
          'solar_rating',
          'postcodes_geo']

In [ ]:
```

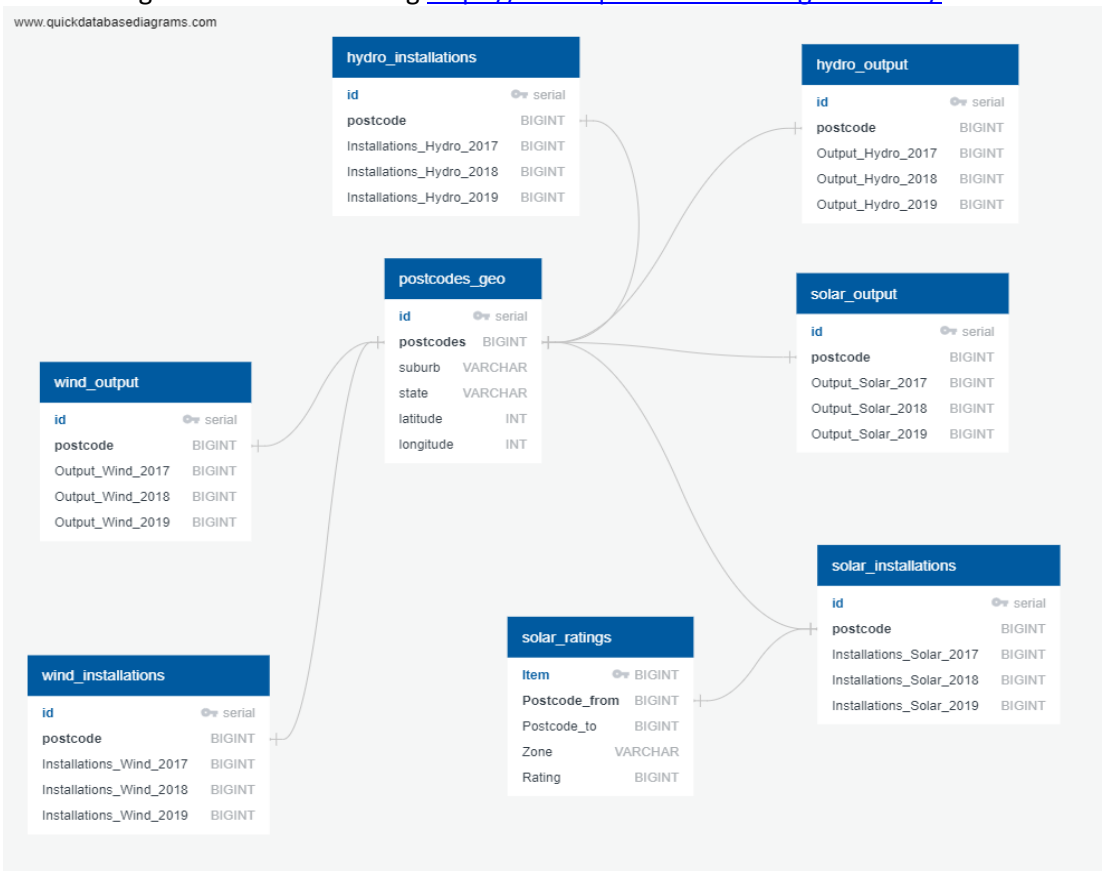
- Final **E**xtract**T**ransform**L**oad work is shown in: ETL_project final.ipynb

ERD:

- Following query was used to demonstrate the use of our database for solar installations in Victoria as per postcodes and their ratings.



- ERD diagram was created using https://www.quickdatabasediagrams.com/



ERD NOTES:

- ERD diagram is saved as a PNG file “ETL_project_ERD.png”
- DB testing query is stored in “Queries_on_database”.