UNIVERSITI MALAYSIA PAHANG
AL-SULTAN ABDULLAH

BSD2423 DATA WAREHOUSING

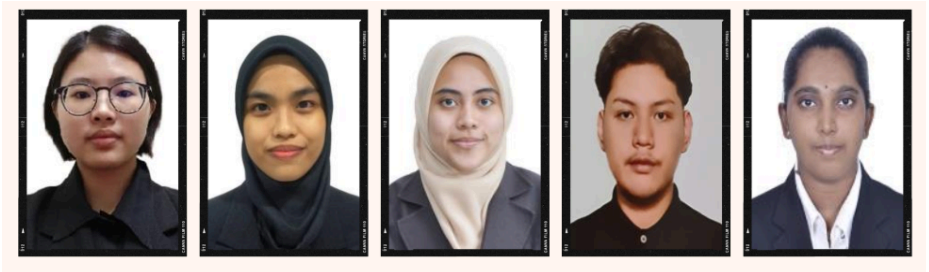
GROUP PROJECT

TITLE:

UNICON ENERGY CONSUMPTION DATASET

LECTURER:

DR. NOR AZUANA BINTI RAMLI



PREPARED BY:

MATRIC ID	NAME	SECTION
SD22059	LOW ANN GIE	01G
SD22012	PUTRI BALQIS BATRISYIA BINTI MOHD RIZAL	02G
SD22017	BATRISYIA BINTI ISMAIL	01G
SD22018	FAIZNAJMI BIN ABD RAUF	01G
SD22041	TANUSHALANI A/P MUNIANDY	02G

TABLE OF CONTENT

1.0 Background

1.1 Project Background

Recently, energy consumption has increased rapidly due to the development of the economic sector. Therefore, energy sustainability has become a global concern. The increase in energy consumption will harm the environment. The negative impacts include solid waste disposal, thermal pollution, climate change, and air pollution. In this project, we shall employ the currently available data to establish different sustainable practices in the use of energy and energy utilisation.

The data source for this study is from the UNICON dataset on energy, and consumption at La Trobe University as sourced from Kaggle. This dataset was posted to the public by La Trobe University to address this objective of Net Zero Carbon Emissions by 2029. The university also has developed the La Trobe Energy AI/Analytics Platform (LEAP) to achieve this goal. The UNICON dataset includes electricity, water and gas consumption at all five of La Trobe's campuses over four years from 2018 to 2021. Besides, the raw data in UNICON includes hourly gas meter data and smart water and electricity meter data recorded every 15 minutes. In addition, the weather data is also provided in the dataset.

It ties in with the "Responsible Consumption and Production" theme of the United Nation's Sustainable Development Goals (SDG 12). This project will investigate the factors contributing to high utility consumption. Besides, the findings of this project will aim at capturing the energy consumption and pattern at La Trobe University. This makes it possible for La Trobe University to determine the campus that is most saturating power and hence affects energy consumption most.

1.2 Description of Data

The UNICON data from La Trobe University is the dataset that we utilised for this study. This data contains various tables, including building consumption, meta, submeter consumption, calendar, campus meta, events, gas consumption, nmi consumption, nmi meta, water consumption and weather data.

1. building_consumption1

Variables	Data Type	Description
campus_id	text	Unique identifier for each campus
building_meter_id	text	Unique identifier for each building meter
timestamp	timestamp	The date and time of the meter reading
building_consumption	numeric	The amount of utility consumption for the building

2. building_meta1

Variables	Data Type	Description
campus_id	text	Unique identifier for each campus
building_meter_id	text	Unique identifier for each building meter
built_year	integer	The year the building was constructed
category	text	The category or type of the building
gross_floor_area	numeric	The total floor area of the building in square meters

room_area	numeric	The total room area of the building in square meters
capacity	numeric	Capacity of the building, typically in terms of occupancy

3. building_submeter_consumption1

Variables	Data Type	Description
building_meter_id	text	Unique identifier for each building meter
building_submeter_id	text	Unique identifier for each submeter within the building
campus_id	text	Unique identifier for each campus
timestamp	timestamp	The date and time of the submeter reading
building_submeter_consumption	numeric	The amount of utility consumption recorded by the submeter
current	numeric	The current measurement in amperes
voltage	numeric	The voltage measurement in volts
power	numeric	The power measurement in watts

4. calender1

Variables	Data Type	Description
date	date	The specific date
is_holiday	text	Indicates if the date is a holiday (yes/no)
is_semester	text	Indicates if the date falls within a semester (yes/no)
is_exam	text	Indicates if the date is during an exam period (yes/no)

5. campus_meta1

Variables	Data Type	Description
campus_id	text	Unique identifier for each campus
name	text	Name of the campus
capacity	integer	Capacity of the campus, usually in terms of occupancy

6. events1

Variables	Data Type	Description
building_meter_id	text	Unique identifier for each building meter
event_type	text	Type of event (e.g., ECM, M&V activity)
date	date	The date of the event
event_description	text	Description of the event

7. gas_consumption1

Variables	Data Type	Description
campus_id	text	Unique identifier for each campus
timestamp	timestamp	The date and time of the gas consumption reading
gas_consumption	numeric	The amount of gas consumption

8. nmi_consumption1

Variables	Data Type	Description
campus_id	text	Unique identifier for each campus
nmi_meter_id	text	Unique identifier for each NMI meter
timestamp	timestamp	The date and time of the NMI consumption reading
consumption	numeric	The amount of utility consumption
demand_kW	numeric	The demand measurement in kilowatts
demand_kVA	numeric	The demand measurement in kilovolt-amperes

9. nmi_meta1

Variables	Data Type	Description
nmi_meter_id	text	Unique identifier for each NMI meter
campus_id	text	Unique identifier for each campus
peak_demand	numeric	Peak demand measurement for the meter

10. water_consumption1

Variables	Data Type	Description
campus_id	text	Unique identifier for each campus
water_meter_id	text	Unique identifier for each water meter
timestamp	timestamp	The date and time of the water consumption reading
water_consumption	numeric	The amount of water consumption

11. weather_data1

Variables	Data Type	Description
campus_id	text	Unique identifier for each campus
timestamp	timestamp	The date and time of the weather data recording
apparent_temperature	numeric	The apparent temperature measurement
air_temperature	numeric	The air temperature measurement
dew_point_temperature	numeric	The dew point temperature measurement
relative_humidity	numeric	The relative humidity measurement

1.3 Problem To Be Solved

La Trobe University has set an ambitious goal to achieve Net Zero Carbon Emissions by 2029, as the need for educational institutions to prioritise sustainability and carbon neutrality grows. Innovative approaches are needed for this project, to minimise the consumption of utilities, such as gas, water, and electricity throughout all of its campus. The UNICON dataset, which provides utility use information from each of La Trobe's five campuses during a four-year period(2018-2021), is a helpful resource for this study. La Trobe University needs to determine the most effective way to use the UNICON data for utility consumption observing, forecasting, and optimisation in order to decrease overall usage and achieve its sustainability goals.

The following highlights are some important inquiries that need to be stated:

1. What are the main factors that cause the usage of utilities?
2. What are the utility consumption trends by year and month?
3. Which building has the highest utility consumption?

1.4 Objective

The objective of this project:

- To identify the main factors that cause the usage of utilities.
- To investigate the utility consumption trends by year and month.
- To determine which building has the highest utility consumption.

1.5 Data Schema

A data schema is a collection of database objects, including tables, views, indexes, and synonyms. There are a variety of ways of arranging schema objects in the schema models designed for data warehousing. The list dataset consists of 11 tables which are building consumption, building meta, building submeter consumption, calender, campus meta, events, gas consumption, nmi consumption, nmi meta, water consumption and weather data.

We used 2 libraries in Jupyter to display data schema which are Pandas and Numpy libraries.

import Pandas as pd
import Numpy as np

1. building consumption

```
[4]: building_consumption = pd.read_csv("building_consumption1.csv")

[5]: building_consumption.dtypes

[5]: campus_id               int64
     building_meter_id       int64
     timestamp              object
     building_consumption  float64
     dtype: object

[6]: building_consumption.info()

     <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 8095524 entries, 0 to 8095523
     Data columns (total 4 columns):
      #   Column               Dtype
     ---  ------               -----
      0   campus_id            int64
      1   building_meter_id    int64
      2   timestamp            object
      3   building_consumption  float64
     dtypes: float64(1), int64(2), object(1)
     memory usage: 247.1+ MB
```

Figure 1.5.1: building consumption tables

Based on Figure 1.5.1 above, the data schema for the building consumption consists of 4 columns. This data frame consists of three data types which are integer, object and float. Only two columns have integer data types while one of the columns is object data type and another one is float data type.

2. building meta

```
[8]: building_meta = pd.read_csv("building_meta1.csv")

[9]: building_meta.dtypes

[9]: campus_id              int64
     building_meter_id      int64
     built_year             int64
     category              object
     gross_floor_area     float64
     room_area            float64
     capacity               int64
     dtype: object

[10]: building_meta.info()

     <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 64 entries, 0 to 63
     Data columns (total 7 columns):
      #   Column             Non-Null Count  Dtype
     ---  ------             --------------  -----
      0   campus_id          64 non-null     int64
      1   building_meter_id  64 non-null     int64
      2   built_year         64 non-null     int64
      3   category           64 non-null     object
      4   gross_floor_area   64 non-null     float64
      5   room_area          64 non-null     float64
      6   capacity           64 non-null     int64
     dtypes: float64(2), int64(4), object(1)
     memory usage: 3.6+ KB
```

Figure 1.5.2: building meta tables

Based on Figure 1.5.2 above, the data schema for the building meta consists of 7 columns. This data frame consists of three data types which are integer, object and float. Four columns have integer data types while two of the columns are in float data type and another one is object data type.

3. building submeter consumption

```
[11]: building_submeter_consumption = pd.read_csv("building_submeter_consumption1.csv")

[12]: building_submeter_consumption.dtypes

[12]: building_meter_id                int64
      building_submeter_id             int64
      campus_id                        int64
      timestamp                       object
      building_submeter_consumption  float64
      current                        float64
      voltage                        float64
      power                          float64
      dtype: object

[13]: building_submeter_consumption.info()

      <class 'pandas.core.frame.DataFrame'>
      RangeIndex: 1312464 entries, 0 to 1312463
      Data columns (total 8 columns):
       #   Column                         Non-Null Count    Dtype
      ---  ------                         --------------    -----
       0   building_meter_id              1312464 non-null  int64
       1   building_submeter_id           1312464 non-null  int64
       2   campus_id                      1312464 non-null  int64
       3   timestamp                      1312464 non-null  object
       4   building_submeter_consumption  1312464 non-null  float64
       5   current                        1312464 non-null  float64
       6   voltage                        1312464 non-null  float64
       7   power                          1312464 non-null  float64
      dtypes: float64(4), int64(3), object(1)
      memory usage: 80.1+ MB
```

Figure 1.5.3: building submeter consumption tables

Based on Figure 1.5.3 above, the data schema for the building submeter consumption consists of 8 columns. This data frame consists of three data types which are integer, object and float. Three columns have integer data types, four columns are in float data type and another one is object data type.

4. Calender

```
[14]: calender = pd.read_csv("calender1.csv")

[15]: calender.dtypes

[15]: date           object
      is_holiday    float64
      is_semester     int64
      is_exam         int64
      dtype: object

[16]: calender.info()

      <class 'pandas.core.frame.DataFrame'>
      RangeIndex: 2312 entries, 0 to 2311
      Data columns (total 4 columns):
       #   Column       Non-Null Count  Dtype
      ---  ------       --------------  -----
       0   date         2312 non-null   object
       1   is_holiday   2312 non-null   float64
       2   is_semester  2312 non-null   int64
       3   is_exam      2312 non-null   int64
      dtypes: float64(1), int64(2), object(1)
      memory usage: 72.4+ KB
```

Figure 1.5.4: calender tables

Based on Figure 1.5.4 above, the data schema for the calender consists of 4 columns. This data frame consists of three data types which are integer, object and float. Two columns have integer data types while one column is float data type and another one is object data type.

5. campus meta

```
[17]: campus_meta = pd.read_csv("campus_meta1.csv")

[18]: campus_meta.dtypes

[18]: campus_id     int64
      name         object
      capacity      int64
      dtype: object

[19]: campus_meta.info()

      <class 'pandas.core.frame.DataFrame'>
      RangeIndex: 5 entries, 0 to 4
      Data columns (total 3 columns):
       #   Column     Non-Null Count  Dtype
      ---  ------     --------------  -----
       0   campus_id  5 non-null      int64
       1   name       5 non-null      object
       2   capacity   5 non-null      int64
      dtypes: int64(2), object(1)
      memory usage: 252.0+ bytes
```

Figure 1.5.5: campus meta tables

Based on Figure 1.5.5 above, the data schema for the campus meta consists of 3 columns. This data frame consists of three data types which are integer and object. Two columns have integer data types and another one is object data type.

6. events

```
[20]: events = pd.read_csv("events1.csv")

[21]: events.dtypes

[21]: building_meter_id     int64
      event_type           object
      date                 object
      event_description    object
      dtype: object

[22]: events.info()

      <class 'pandas.core.frame.DataFrame'>
      RangeIndex: 106 entries, 0 to 105
      Data columns (total 4 columns):
       #   Column             Non-Null Count  Dtype
      ---  ------             --------------  -----
       0   building_meter_id  106 non-null    int64
       1   event_type         106 non-null    object
       2   date               106 non-null    object
       3   event_description  106 non-null    object
      dtypes: int64(1), object(3)
      memory usage: 3.4+ KB
```

Figure 1.5.6: events tables

Based on Figure 1.5.6 above, the data schema for the events consists of 4 columns. This data frame consists of three data types which are integer and object. Four columns have integer data types and another one has integer data types.

7. gas consumption

```
[23]: gas_consumption = pd.read_csv("gas_consumption1.csv")

[24]: gas_consumption.dtypes

[24]: campus_id            int64
      timestamp           object
      gas_consumption    float64
      dtype: object

[25]: gas_consumption.info()

      <class 'pandas.core.frame.DataFrame'>
      RangeIndex: 25747 entries, 0 to 25746
      Data columns (total 3 columns):
       #   Column           Non-Null Count  Dtype
      ---  ------           --------------  -----
       0   campus_id        25747 non-null  int64
       1   timestamp        25747 non-null  object
       2   gas_consumption  25747 non-null  float64
      dtypes: float64(1), int64(1), object(1)
      memory usage: 603.6+ KB
```

Figure 1.5.7: gas consumption tables

Based on Figure 1.5.7 above, the data schema for the gas consumption consists of 3 columns. This data frame consists of three data types which are integer, float and object. One column has integer data types while another column is in object data type and one is integer data type.

8. nmi consumption

```
[26]: nmi_consumption = pd.read_csv("nmi_consumption1.csv")

[27]: nmi_consumption.dtypes

[27]: campus_id            int64
      nmi_meter_id         int64
      timestamp           object
      nmi_consumption    float64
      demand_kw          float64
      demand_kva         float64
      dtype: object

[28]: nmi_consumption.info()

      <class 'pandas.core.frame.DataFrame'>
      RangeIndex: 1048575 entries, 0 to 1048574
      Data columns (total 6 columns):
       #   Column           Non-Null Count    Dtype
      ---  ------           --------------    -----
       0   campus_id        1048575 non-null  int64
       1   nmi_meter_id     1048575 non-null  int64
       2   timestamp        1048575 non-null  object
       3   nmi_consumption  1048575 non-null  float64
       4   demand_kw        1048575 non-null  float64
       5   demand_kva       1048575 non-null  float64
      dtypes: float64(3), int64(2), object(1)
      memory usage: 48.0+ MB
```

Figure 1.5.8: nmi consumption tables

Based on Figure 1.5.8 above, the data schema for the nmi consumption consists of 6 columns. This data frame consists of three data types which are integer, float and object. Two columns have integer data types while another three columns are in float data type and one is object data type.

9. nmi meta

```
[29]: nmi_meta = pd.read_csv("nmi_meta1.csv")

[30]: nmi_meta.dtypes

[30]: nmi_meter_id      int64
      campus_id         int64
      peak_demand     float64
      dtype: object

[31]: nmi_meta.info()

      <class 'pandas.core.frame.DataFrame'>
      RangeIndex: 14 entries, 0 to 13
      Data columns (total 3 columns):
       #   Column        Non-Null Count  Dtype
      ---  ------        --------------  -----
       0   nmi_meter_id  14 non-null     int64
       1   campus_id     14 non-null     int64
       2   peak_demand   14 non-null     float64
      dtypes: float64(1), int64(2)
      memory usage: 468.0 bytes
```

Figure 1.5.9: nmi meta tables

Based on Figure 1.5.9 above, the data schema for the nmi meta consists of 3 columns. This data frame consists of three data types which are integer and float. Two columns have integer data types and one is float data type.

10. water consumption

```
[32]: water_consumption = pd.read_csv("water_consumption1.csv")

[33]: water_consumption.dtypes

[33]: campus_id             int64
      water_meter_id        int64
      timestamp            object
      water_consumption    float64
      dtype: object

[34]: water_consumption.info()

      <class 'pandas.core.frame.DataFrame'>
      RangeIndex: 488058 entries, 0 to 488057
      Data columns (total 4 columns):
       #   Column             Non-Null Count   Dtype
      ---  ------             --------------   -----
       0   campus_id          488058 non-null  int64
       1   water_meter_id     488058 non-null  int64
       2   timestamp          488058 non-null  object
       3   water_consumption  488058 non-null  float64
      dtypes: float64(1), int64(2), object(1)
      memory usage: 14.9+ MB
```

Figure 1.5.10: water consumption tables

Based on Figure 1.5.10 above, the data schema for the water consumption consists of 4 columns. This data frame consists of three data types which are integer, object and float. Two columns have integer data types while another column is in object data type and one is float data type.

11. weather data

```
[35]: weather_data = pd.read_csv("weather_data1.csv")

[36]: weather_data.dtypes

[36]: campus_id                int64
      timestamp               object
      apparent_temperature   float64
      air_temperature        float64
      dew_point_temperature  float64
      relative_humidity      float64
      dtype: object

[37]: weather_data.info()

      <class 'pandas.core.frame.DataFrame'>
      RangeIndex: 7396520 entries, 0 to 7396519
      Data columns (total 6 columns):
       #   Column                Dtype
      ---  ------                -----
       0   campus_id             int64
       1   timestamp             object
       2   apparent_temperature  float64
       3   air_temperature       float64
       4   dew_point_temperature float64
       5   relative_humidity     float64
      dtypes: float64(4), int64(1), object(1)
      memory usage: 338.6+ MB
```

Figure 1.5.11: weather data tables

Based on Figure 1.5.11 above, the data schema for the weather data consists of 6 columns. This data frame consists of three data types which are integer, object and float. Four columns have integer data types while another column is in object data type and one is float data type.

2.0 Architecture and ETL Pipeline

2.1 Data Architecture

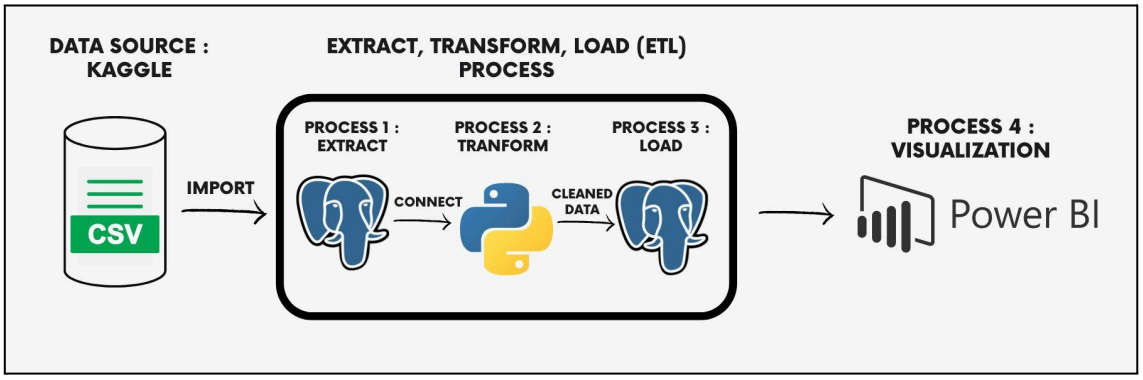


Figure 2.1.1: Data Architecture

Based on Figure 2.1.1, we get the CSV dataset of UNICON energy consumption from Kaggle. The dataset we received contains 11 tables which are calender, events, building meta, building consumption, building submeter consumption, campus meta, gas consumption, water consumption, nmi consumption, weather data, and nmi meta. The dataset was imported into the PostgreSQL database to begin the extraction process. After successfully importing all tables into PostgreSQL, we proceed to further operations where we connect the dataset from PostgreSQL using Python in Jupyter Notebook for the transformation process. At this point, we clean and modify the data where we carefully identify and remove null values for future use. After the transformation, the cleaned data is loaded again into the PostgreSQL database for Online Analytical Processing (OLAP) operations like rollup, slicing, dicing and pivot. OLAP operations have been structured and optimised to support efficient and high-performance analytical queries. Finally, we use Power BI to visualise the cleaned data for analysis, creating dimension tables and fact tables to determine the relationship between the data, which connects to the PostgreSQL database to generate various data visualisations. This whole process ensures that the raw data from the CSV file is effectively transformed into a meaningful and attractive structure for the analysis.

2.1.1 Pipeline Structure

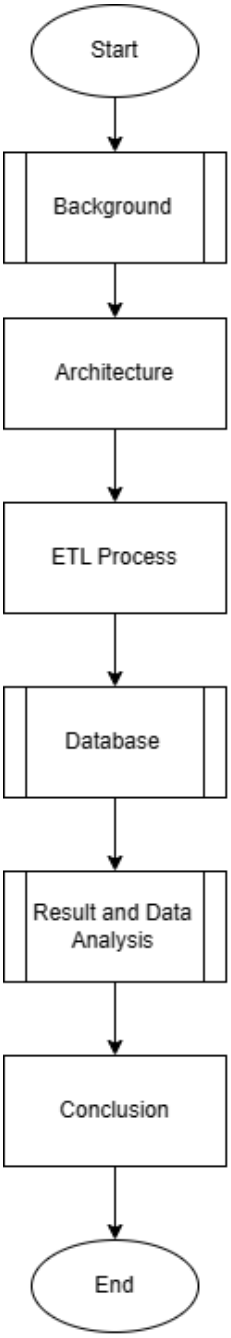


Figure 2.1.1.1: Flow chart of the project

The figure above shows the process of the whole project. We start our project by introducing the background of our project. Besides, we draw the data architecture and ETL pipeline. We will extract the data from Excel to PostgreSQL and transform it into Jupyter. Then, we load the cleaned data into Postgres. Furthermore, we present the database in a relational model. After that, we perform some OLAP operations and interpret the visualisation. Lastly, we conclude our project.

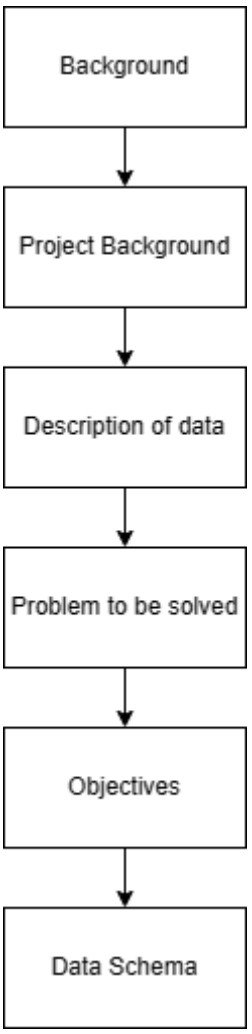


Figure 2.1.1.2: Flow of background

After choosing the dataset, we start by exploring and understanding the dataset. We introduce the project background and describe the data with its data type and data description. After that, we identify the problem that needs to be addressed and the objectives of our project. We will show the data schema of the dataset.

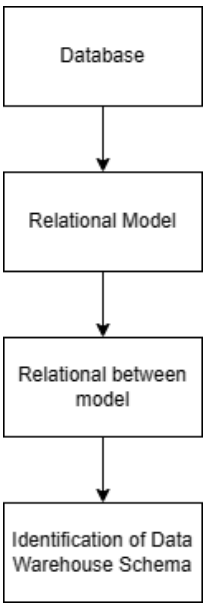


Figure 2.1.1.3: Flow of database

For the database, we identify the relationship between each table and connect their relationship using Power BI. Then, we will identify the data warehouse schema for the relational model.

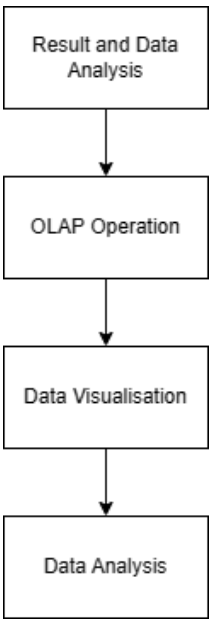


Figure 2.1.1.4: Flow of result and data analysis

After loading the data into Postgres, we perform OLAP operations such as roll up, slicing, dicing and pivot. Besides, we carry out our data visualisation using Power BI. After data visualisation, we start to interpret the result that we obtained and find meaningful insight.

2.2 ETL Pipeline

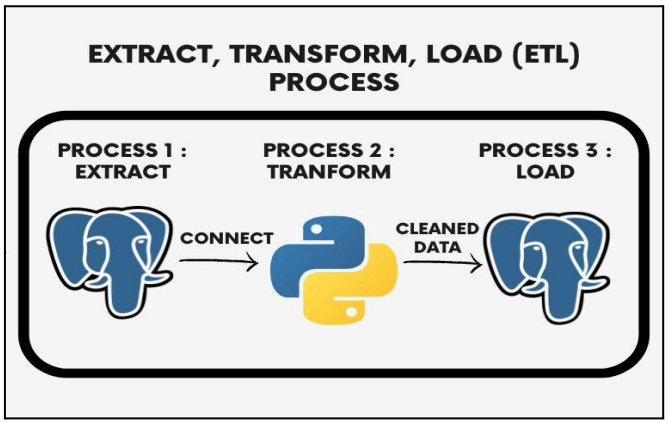


Figure 2.2.1: ETL pipeline

Figure shows three main stages of the pipeline that consists of Extract, Transform, and Load (ETL). It is a data integration process that collects data from multiple sources, transforms it into a consistent format, and loads it into a data warehouse for analysis. This ETL pipeline is important in the Kimball data warehousing approach for preparing our data for business-focused data marts. In this project, we extract the raw data from the CSV file using PostgreSQL. This stage is critical for gathering the required data for the analysis. After the data is extracted, the data is connected from PostgreSQL and transformed the data to Jupyter Notebook with Python. In this stage, we clean the data, remove duplicate data, convert the data types and also ensure the consistency and accuracy of the data. After cleaning the data, we loaded the data into PostgreSQL for doing OLAP operations. After that, we load the updated PostgreSQL data into PowerBI to generate dimension tables and fact tables to discover the relationship between the data in the data marts. These data marts will eventually be connected to create a complete data warehouse. Finally, we choose the Kimball technique in our project since it is easy to use, flexible, and provides the outcomes that we need. Additionally, the Kimball methodology emphasises the creation of these data marts that are tailored to specific business areas and then integrated into a complete data warehouse using a "bottom-up" approach.

2.3 ETL Process

2.3.1 Extract

Data extraction is a process of collecting data from various sources and storing it in a centralised location to make it easily accessible and transformed. We start our data extraction process by creating a database named “UNICON” in PostgreSQL. Then, we use queries to create 11 tables in the “UNICON” database for storing raw data obtained from Kaggle.

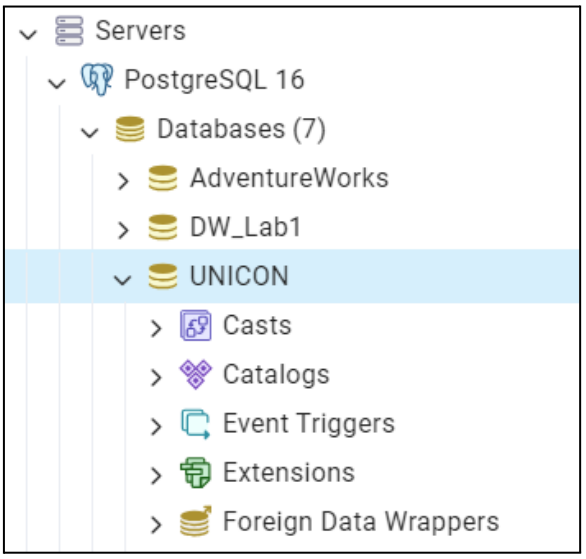


Figure 2.3.1.1: “UNICON” database is created

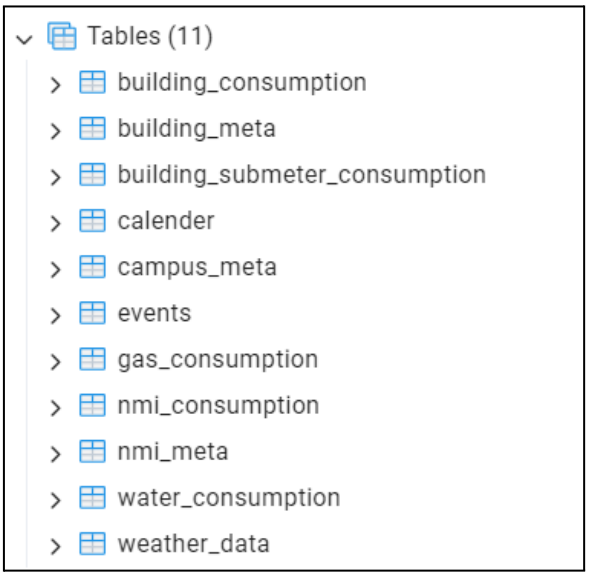
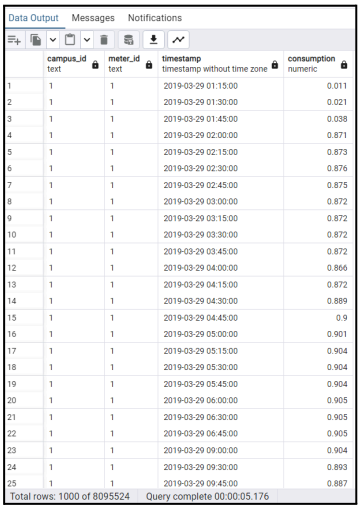


Figure 2.3.1.2: Tables are created to extract raw data

Queries and Output of data in each table

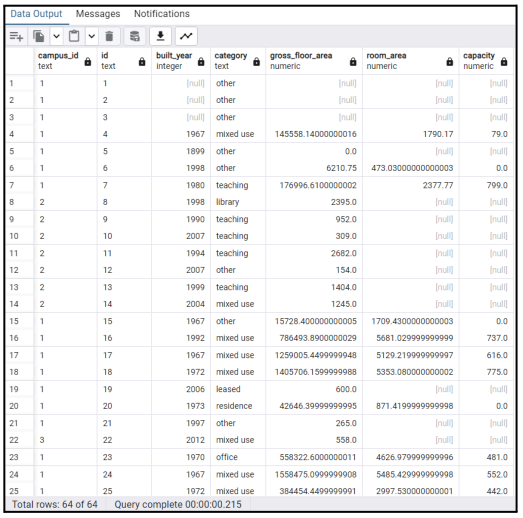
SELECT * FROM building_consumption;



	campus_id text	meter_id text	timestamp timestamp without time zone	consumption numeric
1	1	1	2019-03-29 01:15:00	0.011
2	1	1	2019-03-29 01:30:00	0.021
3	1	1	2019-03-29 01:45:00	0.038
4	1	1	2019-03-29 02:00:00	0.871
5	1	1	2019-03-29 02:15:00	0.873
6	1	1	2019-03-29 02:30:00	0.876
7	1	1	2019-03-29 02:45:00	0.875
8	1	1	2019-03-29 03:00:00	0.872
9	1	1	2019-03-29 03:15:00	0.872
10	1	1	2019-03-29 03:30:00	0.872
11	1	1	2019-03-29 03:45:00	0.872
12	1	1	2019-03-29 04:00:00	0.866
13	1	1	2019-03-29 04:15:00	0.872
14	1	1	2019-03-29 04:30:00	0.889
15	1	1	2019-03-29 04:45:00	0.9
16	1	1	2019-03-29 05:00:00	0.901
17	1	1	2019-03-29 05:15:00	0.904
18	1	1	2019-03-29 05:30:00	0.904
19	1	1	2019-03-29 05:45:00	0.904
20	1	1	2019-03-29 06:00:00	0.905
21	1	1	2019-03-29 06:30:00	0.905
22	1	1	2019-03-29 06:45:00	0.905
23	1	1	2019-03-29 09:00:00	0.904
24	1	1	2019-03-29 09:30:00	0.893
25	1	1	2019-03-29 09:45:00	0.887

Total rows: 1000 of 8095524 Query complete 00:00:05.176

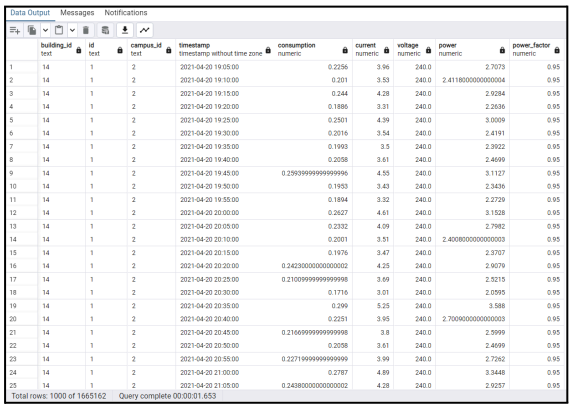
SELECT * FROM building_meta;



	campus_id text	id text	built_year integer	category text	gross_floor_area numeric	room_area numeric	capacity numeric
1	1	1	[null]	other	[null]	[null]	[null]
2	1	2	[null]	other	[null]	[null]	[null]
3	1	3	[null]	other	[null]	[null]	[null]
4	1	4	1967	mixed use	145558.14000000016	1790.17	79.0
5	1	5	1899	other	0.0	[null]	[null]
6	1	6	1998	other	6210.75	473.03000000000003	0.0
7	1	7	1980	teaching	176996.6100000002	2377.77	799.0
8	2	8	1998	library	2395.0	[null]	[null]
9	2	9	1990	teaching	952.0	[null]	[null]
10	2	10	2007	teaching	309.0	[null]	[null]
11	2	11	1994	teaching	2682.0	[null]	[null]
12	2	12	2007	other	154.0	[null]	[null]
13	2	13	1999	teaching	1404.0	[null]	[null]
14	2	14	2004	mixed use	1245.0	[null]	[null]
15	1	15	1967	other	15728.400000000005	1709.4300000000003	0.0
16	1	16	1992	mixed use	786493.8900000029	5681.029999999999	737.0
17	1	17	1967	mixed use	1259005.4499999948	5129.219999999997	616.0
18	1	18	1972	mixed use	1405706.1599999988	5353.080000000002	775.0
19	1	19	2006	leased	600.0	[null]	[null]
20	1	20	1973	residence	42646.39999999995	871.4199999999998	0.0
21	1	21	1997	other	265.0	[null]	[null]
22	3	22	2012	mixed use	558.0	[null]	[null]
23	1	23	1970	office	558322.6000000011	4626.979999999996	481.0
24	1	24	1967	mixed use	1558475.0999999908	5485.429999999998	552.0
25	1	25	1972	mixed use	384454.4499999991	2997.530000000001	442.0

Total rows: 64 of 64 Query complete 00:00:00.215

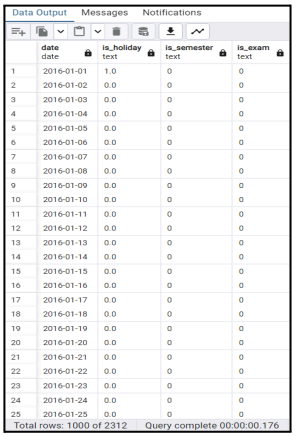
SELECT * FROM building_submeter_consumption;



	building_id text	id text	campus_id text	timestamp timestamp without time zone	consumption numeric	current numeric	voltage numeric	power numeric	power_factor numeric
1	14	1	2	2021-04-20 19:05:00	0.2256	3.96	240.0	2.7073	0.95
2	14	1	2	2021-04-20 19:10:00	0.201	3.53	240.0	2.411800000000004	0.95
3	14	1	2	2021-04-20 19:15:00	0.244	4.28	240.0	2.9284	0.95
4	14	1	2	2021-04-20 19:20:00	0.1886	3.31	240.0	2.2636	0.95
5	14	1	2	2021-04-20 19:25:00	0.2501	4.39	240.0	3.0009	0.95
6	14	1	2	2021-04-20 19:30:00	0.2016	3.54	240.0	2.4191	0.95
7	14	1	2	2021-04-20 19:35:00	0.1993	3.5	240.0	2.3922	0.95
8	14	1	2	2021-04-20 19:40:00	0.2058	3.61	240.0	2.4699	0.95
9	14	1	2	2021-04-20 19:45:00	0.2593999999999996	4.55	240.0	3.1127	0.95
10	14	1	2	2021-04-20 19:50:00	0.1953	3.43	240.0	2.3436	0.95
11	14	1	2	2021-04-20 19:55:00	0.1894	3.32	240.0	2.2729	0.95
12	14	1	2	2021-04-20 20:00:00	0.2627	4.61	240.0	3.1528	0.95
13	14	1	2	2021-04-20 20:05:00	0.2332	4.09	240.0	2.7982	0.95
14	14	1	2	2021-04-20 20:10:00	0.2001	3.51	240.0	2.400800000000003	0.95
15	14	1	2	2021-04-20 20:15:00	0.1976	3.47	240.0	2.3707	0.95
16	14	1	2	2021-04-20 20:20:00	0.24230000000000002	4.25	240.0	2.9079	0.95
17	14	1	2	2021-04-20 20:25:00	0.2100999999999998	3.69	240.0	2.5215	0.95
18	14	1	2	2021-04-20 20:30:00	0.1716	3.01	240.0	2.0595	0.95
19	14	1	2	2021-04-20 20:35:00	0.299	5.25	240.0	3.588	0.95
20	14	1	2	2021-04-20 20:40:00	0.2251	3.95	240.0	2.700900000000003	0.95
21	14	1	2	2021-04-20 20:45:00	0.2166999999999998	3.8	240.0	2.5999	0.95
22	14	1	2	2021-04-20 20:50:00	0.2058	3.61	240.0	2.4699	0.95
23	14	1	2	2021-04-20 20:55:00	0.2271999999999999	3.99	240.0	2.7262	0.95
24	14	1	2	2021-04-20 21:00:00	0.2787	4.89	240.0	3.3448	0.95
25	14	1	2	2021-04-20 21:05:00	0.24380000000000002	4.28	240.0	2.9257	0.95

Total rows: 1000 of 1665162 Query complete 00:00:01.653

SELECT * FROM calender;



	date date	is_holiday text	is_semester text	is_exam text
1	2016-01-01	1.0	0	0
2	2016-01-02	0.0	0	0
3	2016-01-03	0.0	0	0
4	2016-01-04	0.0	0	0
5	2016-01-05	0.0	0	0
6	2016-01-06	0.0	0	0
7	2016-01-07	0.0	0	0
8	2016-01-08	0.0	0	0
9	2016-01-09	0.0	0	0
10	2016-01-10	0.0	0	0
11	2016-01-11	0.0	0	0
12	2016-01-12	0.0	0	0
13	2016-01-13	0.0	0	0
14	2016-01-14	0.0	0	0
15	2016-01-15	0.0	0	0
16	2016-01-16	0.0	0	0
17	2016-01-17	0.0	0	0
18	2016-01-18	0.0	0	0
19	2016-01-19	0.0	0	0
20	2016-01-20	0.0	0	0
21	2016-01-21	0.0	0	0
22	2016-01-22	0.0	0	0
23	2016-01-23	0.0	0	0
24	2016-01-24	0.0	0	0
25	2016-01-25	0.0	0	0

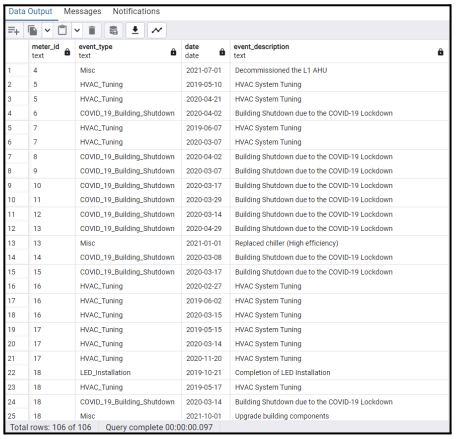
Total rows: 1000 of 2312 Query complete 00:00:00.176

SELECT * FROM campus_meta;

Data Output | Messages | Notifications

	id text	name text	capacity integer
1	1	Bundoora	26000
2	2	Albury-Wodonga	800
3	3	Bendigo	5000
4	4	Mildura	500
5	5	Shepparton	700

SELECT * FROM events;

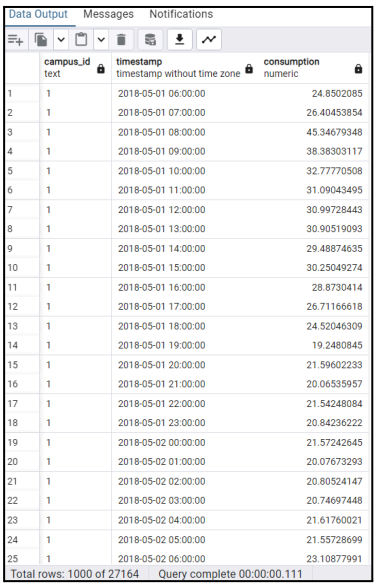


Data Output | Messages | Notifications

	meter_id text	event_type text	date date	event_description text
1	4	Misc	2021-07-01	Decommissioned the L1 AHU
2	5	HVAC_Tuning	2019-05-10	HVAC System Tuning
3	5	HVAC_Tuning	2020-04-21	HVAC System Tuning
4	6	COVID_19_Building_Shutdown	2020-04-02	Building Shutdown due to the COVID-19 Lockdown
5	7	HVAC_Tuning	2019-06-07	HVAC System Tuning
6	7	HVAC_Tuning	2020-03-07	HVAC System Tuning
7	8	COVID_19_Building_Shutdown	2020-04-02	Building Shutdown due to the COVID-19 Lockdown
8	9	COVID_19_Building_Shutdown	2020-03-07	Building Shutdown due to the COVID-19 Lockdown
9	10	COVID_19_Building_Shutdown	2020-03-17	Building Shutdown due to the COVID-19 Lockdown
10	11	COVID_19_Building_Shutdown	2020-03-29	Building Shutdown due to the COVID-19 Lockdown
11	12	COVID_19_Building_Shutdown	2020-03-14	Building Shutdown due to the COVID-19 Lockdown
12	13	COVID_19_Building_Shutdown	2020-04-29	Building Shutdown due to the COVID-19 Lockdown
13	13	Misc	2021-01-01	Replaced chiller (High efficiency)
14	14	COVID_19_Building_Shutdown	2020-03-08	Building Shutdown due to the COVID-19 Lockdown
15	15	COVID_19_Building_Shutdown	2020-03-17	Building Shutdown due to the COVID-19 Lockdown
16	16	HVAC_Tuning	2020-02-27	HVAC System Tuning
17	16	HVAC_Tuning	2019-06-02	HVAC System Tuning
18	16	HVAC_Tuning	2020-03-15	HVAC System Tuning
19	17	HVAC_Tuning	2019-05-15	HVAC System Tuning
20	17	HVAC_Tuning	2020-03-14	HVAC System Tuning
21	17	HVAC_Tuning	2020-11-20	HVAC System Tuning
22	18	LED_Installation	2019-10-21	Completion of LED Installation
23	18	HVAC_Tuning	2019-05-17	HVAC System Tuning
24	18	COVID_19_Building_Shutdown	2020-03-14	Building Shutdown due to the COVID-19 Lockdown
25	18	Misc	2021-10-01	Upgrade building components

Total rows: 106 of 106 Query complete 00:00:00.097

SELECT * FROM gas_consumption;

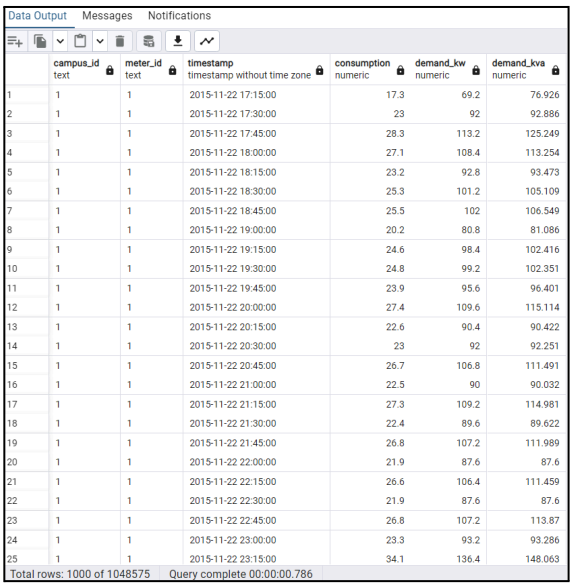


Data Output | Messages | Notifications

	campus_id text	timestamp timestamp without time zone	consumption numeric
1	1	2018-05-01 06:00:00	24.8502085
2	1	2018-05-01 07:00:00	26.40453854
3	1	2018-05-01 08:00:00	45.34679348
4	1	2018-05-01 09:00:00	38.38303117
5	1	2018-05-01 10:00:00	32.77770508
6	1	2018-05-01 11:00:00	31.09043495
7	1	2018-05-01 12:00:00	30.99728443
8	1	2018-05-01 13:00:00	30.90519093
9	1	2018-05-01 14:00:00	29.48874635
10	1	2018-05-01 15:00:00	30.25049274
11	1	2018-05-01 16:00:00	28.8730414
12	1	2018-05-01 17:00:00	26.71166618
13	1	2018-05-01 18:00:00	24.52046309
14	1	2018-05-01 19:00:00	19.2480845
15	1	2018-05-01 20:00:00	21.59602233
16	1	2018-05-01 21:00:00	20.06535957
17	1	2018-05-01 22:00:00	21.54248084
18	1	2018-05-01 23:00:00	20.84236222
19	1	2018-05-02 00:00:00	21.57242645
20	1	2018-05-02 01:00:00	20.07673293
21	1	2018-05-02 02:00:00	20.80524147
22	1	2018-05-02 03:00:00	20.74697448
23	1	2018-05-02 04:00:00	21.61760021
24	1	2018-05-02 05:00:00	21.55728699
25	1	2018-05-02 06:00:00	23.10877991

Total rows: 1000 of 27164 Query complete 00:00:00.111

SELECT * FROM nmi_consumption;



Data Output | Messages | Notifications

	campus_id text	meter_id text	timestamp timestamp without time zone	consumption numeric	demand_kw numeric	demand_kva numeric
1	1	1	2015-11-22 17:15:00	17.3	69.2	76.926
2	1	1	2015-11-22 17:30:00	23	92	92.886
3	1	1	2015-11-22 17:45:00	28.3	113.2	125.249
4	1	1	2015-11-22 18:00:00	27.1	108.4	113.254
5	1	1	2015-11-22 18:15:00	23.2	92.8	93.473
6	1	1	2015-11-22 18:30:00	25.3	101.2	105.109
7	1	1	2015-11-22 18:45:00	25.5	102	106.549
8	1	1	2015-11-22 19:00:00	20.2	80.8	81.086
9	1	1	2015-11-22 19:15:00	24.6	98.4	102.416
10	1	1	2015-11-22 19:30:00	24.8	99.2	102.351
11	1	1	2015-11-22 19:45:00	23.9	95.6	96.401
12	1	1	2015-11-22 20:00:00	27.4	109.6	115.114
13	1	1	2015-11-22 20:15:00	22.6	90.4	90.422
14	1	1	2015-11-22 20:30:00	23	92	92.251
15	1	1	2015-11-22 20:45:00	26.7	106.8	111.491
16	1	1	2015-11-22 21:00:00	22.5	90	90.032
17	1	1	2015-11-22 21:15:00	27.3	109.2	114.981
18	1	1	2015-11-22 21:30:00	22.4	89.6	89.622
19	1	1	2015-11-22 21:45:00	26.8	107.2	111.989
20	1	1	2015-11-22 22:00:00	21.9	87.6	87.6
21	1	1	2015-11-22 22:15:00	26.6	106.4	111.459
22	1	1	2015-11-22 22:30:00	21.9	87.6	87.6
23	1	1	2015-11-22 22:45:00	26.8	107.2	113.87
24	1	1	2015-11-22 23:00:00	23.3	93.2	93.286
25	1	1	2015-11-22 23:15:00	34.1	136.4	148.063

Total rows: 1000 of 1048575 Query complete 00:00:00.786

SELECT * FROM nmi_meta;

Data Output | Messages | Notifications

	id text	campus_id text	peak_demand numeric
1	1	1	1225.89
2	2	4	121.0
3	3	5	145.0
4	4	3	143.0
5	5	1	260.0
6	6	2	100.0
7	7	1	90.0
8	8	2	930.0
9	9	1	22.68
10	10	3	626.0
11	11	3	[null]
12	12	1	7659.0
13	13	1	120.0
14	14	1	192.751

SELECT * FROM water_consumption;

Data Output | Messages | Notifications

	campus_id text	meter_id text	timestamp timestamp without time zone	consumption numeric
1	1	1	2021-01-10 00:00:00	438.32800000000003
2	1	1	2021-01-10 00:15:00	438.32800000000003
3	1	1	2021-01-10 00:30:00	438.32800000000003
4	1	1	2021-01-10 00:45:00	438.32800000000003
5	1	1	2021-01-10 01:00:00	438.32800000000003
6	1	1	2021-01-10 01:15:00	438.32800000000003
7	1	1	2021-01-10 01:30:00	438.32800000000003
8	1	1	2021-01-10 01:45:00	438.32800000000003
9	1	1	2021-01-10 02:00:00	438.32800000000003
10	1	1	2021-01-10 02:15:00	438.32800000000003
11	1	1	2021-01-10 02:30:00	438.32800000000003
12	1	1	2021-01-10 02:45:00	438.32800000000003
13	1	1	2021-01-10 03:00:00	438.32800000000003
14	1	1	2021-01-10 03:15:00	438.32800000000003
15	1	1	2021-01-10 03:30:00	438.32800000000003
16	1	1	2021-01-10 03:45:00	438.32800000000003
17	1	1	2021-01-10 04:00:00	438.32800000000003
18	1	1	2021-01-10 04:15:00	438.32800000000003
19	1	1	2021-01-10 04:30:00	438.32800000000003
20	1	1	2021-01-10 04:45:00	438.32800000000003
21	1	1	2021-01-10 05:00:00	438.32800000000003
22	1	1	2021-01-10 05:15:00	438.32800000000003
23	1	1	2021-01-10 05:30:00	438.32800000000003
24	1	1	2021-01-10 05:45:00	438.32800000000003
25	1	1	2021-01-10 06:00:00	438.32800000000003

Total rows: 1000 of 245040 Query complete 00:00:00.170

SELECT * FROM weather_data;

Data Output | Messages | Notifications

	campus_id text	timestamp timestamp without time zone	apparent_temperature numeric	air_temperature numeric	dew_point_temperature numeric	relative_humidity numeric	wind_speed numeric	wind_direction numeric
1	1	2018-01-01 00:00:00	16.6	16.2	13.5	84.0	3.6	142.0
2	1	2018-01-01 00:01:00	17.2	16.1	13.6	85.0	0.0	134.0
3	1	2018-01-01 00:02:00	16.9	16.1	13.6	85.0	1.8	130.0
4	1	2018-01-01 00:03:00	16.9	16.1	13.6	85.0	1.8	130.0
5	1	2018-01-01 00:04:00	16.0	16.0	13.5	85.0	5.4	129.0
6	1	2018-01-01 00:05:00	16.7	16.0	13.5	85.0	1.8	129.0
7	1	2018-01-01 00:06:00	16.7	16.0	13.5	85.0	1.8	128.0
8	1	2018-01-01 00:07:00	16.7	16.0	13.5	85.0	1.8	139.0
9	1	2018-01-01 00:08:00	16.6	15.9	13.4	85.0	1.8	154.0
10	1	2018-01-01 00:09:00	16.6	15.9	13.4	85.0	1.8	146.0
11	1	2018-01-01 00:10:00	16.6	15.9	13.4	85.0	1.8	134.0
12	1	2018-01-01 00:11:00	17.0	15.9	13.4	85.0	0.0	134.0
13	1	2018-01-01 00:12:00	17.0	15.9	13.4	85.0	0.0	134.0
14	1	2018-01-01 00:13:00	17.0	15.9	13.6	86.0	0.0	134.0
15	1	2018-01-01 00:14:00	16.3	15.9	13.6	86.0	3.6	134.0
16	1	2018-01-01 00:15:00	16.0	15.9	13.6	86.0	5.4	134.0
17	1	2018-01-01 00:16:00	16.7	15.9	13.6	86.0	1.8	134.0
18	1	2018-01-01 00:17:00	17.0	15.9	13.6	86.0	0.0	134.0
19	1	2018-01-01 00:18:00	16.3	15.9	13.6	86.0	3.6	134.0
20	1	2018-01-01 00:19:00	15.8	15.8	13.5	86.0	5.4	158.0
21	1	2018-01-01 00:20:00	16.3	15.9	13.6	86.0	3.6	140.0
22	1	2018-01-01 00:21:00	16.0	15.9	13.6	86.0	5.4	133.0
23	1	2018-01-01 00:22:00	16.3	15.9	13.6	86.0	3.6	132.0
24	1	2018-01-01 00:23:00	16.2	15.8	13.5	86.0	3.6	132.0
25	1	2018-01-01 00:24:00	15.8	15.8	13.5	86.0	5.4	132.0

Total rows: 1000 of 7396520 Query complete 00:00:06.131

After extracting the raw data, we have to connect the data from PostgreSQL to Jupyter for the transformation process. Firstly, the "AdventureWorks" database is created. The database owner is set to the user called "postgres". Then, the character encoding is set as Unicode Transformation Format, 8-bit (UTF-8) and the collation (LC_COLLATE) and character classification (LC_CTYPE) for the dataset is set as "English_Malaysia.1252". The "TABLESPACE = pg_default" query instructs PostgreSQL to store database objects in the default tablespace. The connection limit is set to -1, meaning the number of concurrent connections to the database is unlimited.

```
CREATE DATABASE "AdventureWorks"
    WITH
    OWNER = postgres
    ENCODING = 'UTF8'
    LC_COLLATE = 'English_Malaysia.1252'
    LC_CTYPE = 'English_Malaysia.1252'
    TABLESPACE = pg_default
    CONNECTION LIMIT = -1;
```

Figure 2.3.1.3: The query for creating the "AdventureWorks" database

After creating the "AdventureWorks" database, we connect it to the user's name called etl:

```
CREATE USER etl with PASSWORD 'demopass';
```

Figure 2.3.1.4: The query to create a user called etl with a password

```
GRANT CONNECT ON DATABASE "AdventureWorks" TO etl;
```

Figure 2.3.1.5: The query for connecting the "AdventureWorks" database to etl user

```
GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA public TO etl;
```

Figure 2.3.1.6: The query for selecting, inserting, and deleting on all tables in the public schema to etl user

After creating etl user, we will install several packages like SQLAlchemy and pandas.

```
[1]: !pip install pandas
```

Figure 2.3.1.7: Installation for pandas

```
[2]: !pip install SQLAlchemy
```

Figure 2.3.1.8: Installation for SQLAlchemy

```
[3]: from sqlalchemy import create_engine
     from sqlalchemy.engine import URL
     import pandas as pd
```

Figure 2.3.1.9: Calling the installed packages

```
[4]: uid = 'etl'
     pwd = 'demopass'
     server = "localhost"
     database = "UNICON"
```

Figure 2.3.1.10: Assign the corresponding value to the user ID, password, server, and database

```
[5]: engine = create_engine(f'postgresql://{uid}:{pwd}@{server}:5432/{database}')
```

Figure 2.3.1.11: The “create_engine” function is used to connect Jupyter with the PostgreSQL database

2.3.2 Transform

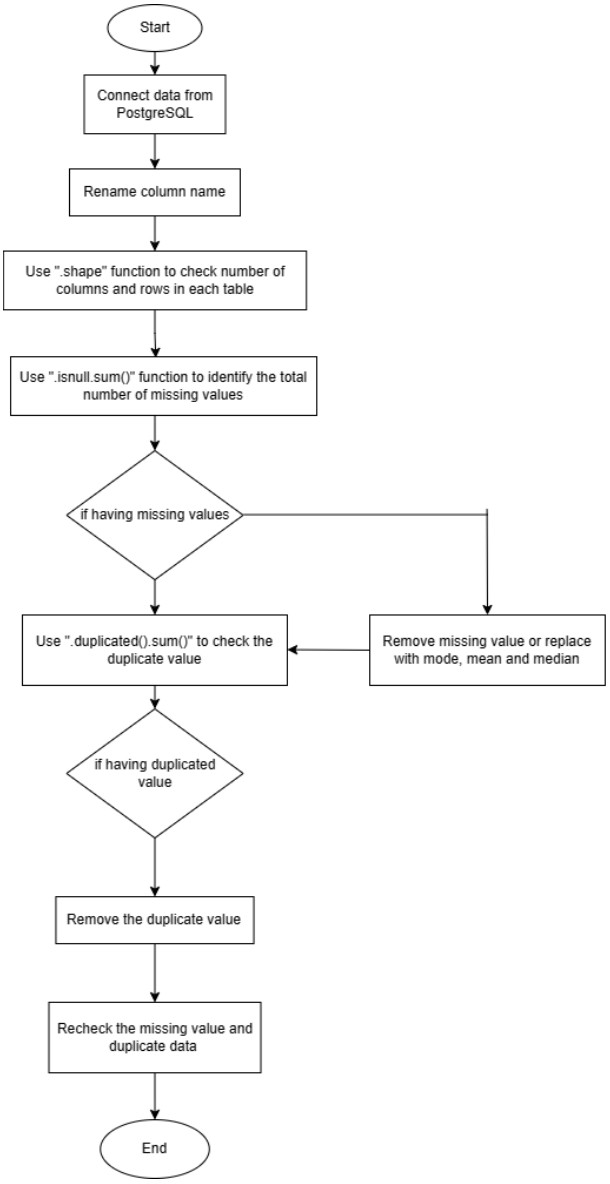


Figure 2.3.2.1: Process of cleaning raw data

Data transformation converts raw data into a cleansed, validated, and ready-to-use format. Firstly, we have to obtain each table's data from PostgreSQL. Then, we rename the column names that are commonly used in other tables. This step can help avoid confusion and easily identify each column. We use the "shape" function to check whether the number of columns and rows of tables imported from PostgreSQL is correct. To check for the total missing values in each table using the "isnull.sum()" function. Besides, we remove the missing values or fill the missing value with mean, mode, and median. After checking the missing values, we check whether the data contains duplicate values. If there is an existing duplicate value, we will remove it. Lastly, we recheck to make sure the dataset does not contain any missing values or duplicate values.

The figure below shows the command for the transformation process in one of the table:

```
[7]: building_meta = "SELECT*FROM building_meta;"
```

Figure 2.3.2.2: Request all data from building metatable

```
[8]: df2 = pd.read_sql_query(building_meta, engine)
```

Figure 2.3.2.3: Store data in a data frame using pandas

```
[9]: df2
```

[9]:

	campus_id	id	built_year	category	gross_floor_area	room_area	capacity
0	1	1	NaN	other	NaN	NaN	NaN
1	1	2	NaN	other	NaN	NaN	NaN
2	1	3	NaN	other	NaN	NaN	NaN
3	1	4	1967.0	mixed use	145558.14	1790.17	79.0
4	1	5	1899.0	other	0.00	NaN	NaN
...	...	...	...	...	...	...	...
59	1	60	1966.0	mixed use	1208347.84	7756.22	691.0
60	1	61	1966.0	mixed use	1208347.84	7756.22	691.0
61	1	62	1968.0	teaching	1274716.17	7615.16	1331.0
62	1	63	1972.0	mixed use	1640647.38	12387.62	1362.0
63	1	64	1899.0	other	5233.40	253.33	0.0

64 rows × 7 columns

Figure 2.3.2.4: Display the data in a data frame

```
[10]: df2.rename(columns={'id':'building_meter_id'}, inplace=True)
```

Figure 2.3.2.5: Rename the “id” column to “building_meter_id”

```
[11]: df2.shape

[11]: (64, 7)
```

Figure 2.3.2.6: use the “shape” function

```
[12]: df2.isnull().sum()

[12]: campus_id            0
      building_meter_id    0
      built_year           6
      category             0
      gross_floor_area     4
      room_area           24
      capacity            24
      dtype: int64
```

Figure 2.3.2.7: Check the total missing value in each column

```
[13]: df2["built_year"] = df2["built_year"].fillna(df2["built_year"].mode()[0])

[14]: df2["gross_floor_area"] = df2["gross_floor_area"].fillna(df2["gross_floor_area"].median())

[15]: df2["room_area"] = df2["room_area"].fillna(df2["room_area"].median())

[16]: df2["capacity"] = df2["capacity"].fillna(df2["capacity"].median())
```

Figure 2.3.2.8: Replace the missing value

```
[17]: df2.duplicated().sum()

[17]: 0
```

Figure 2.3.2.9: "duplicated().sum()" command is executed to check the total number of duplicate data in the building_meta table

2.3.3 Load

[20]:
```
from IPython.display import HTML
import base64
import pandas as pd

def create_download_link( df2, title = "Download CSV file", filename = "building_meta.csv"):
    csv = df2.to_csv(index =False)
    b64 = base64.b64encode(csv.encode())
    payload = b64.decode()
    html = '<a download="{filename}" href="data:text/csv;base64,{payload}" target="_blank">{title}</a>'
    html = html.format(payload=payload,title=title,filename=filename)
    return HTML(html)

create_download_link(df2)
```

[20]: Download CSV file

Figure 2.3.3.1: Create a link to download cleaned data in CSV file

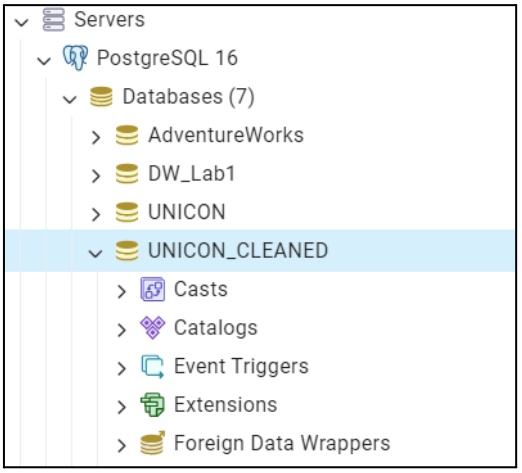


Figure 2.3.3.2: Create new database named as "UNICON_CLEANED"

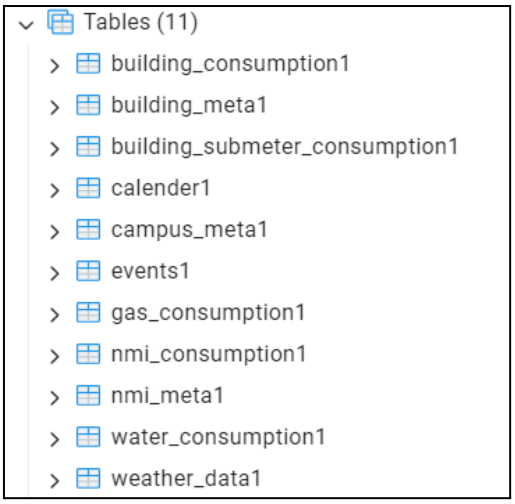


Figure 2.3.3.3: New tables for cleaned data to load into PostgreSQL are created

Queries and Output of data in each table

SELECT * FROM building_consumption1;

	campus_id text	building_meter_id text	timestamp timestamp without time zone	building_consumption numeric
1	1	1	2019-03-29 01:15:00	0.011
2	1	1	2019-03-29 01:30:00	0.021
3	1	1	2019-03-29 01:45:00	0.038
4	1	1	2019-03-29 02:00:00	0.871
5	1	1	2019-03-29 02:15:00	0.873
6	1	1	2019-03-29 02:30:00	0.876
7	1	1	2019-03-29 02:45:00	0.875
8	1	1	2019-03-29 03:00:00	0.872
9	1	1	2019-03-29 03:15:00	0.872
10	1	1	2019-03-29 03:30:00	0.872
11	1	1	2019-03-29 03:45:00	0.872
12	1	1	2019-03-29 04:00:00	0.866
13	1	1	2019-03-29 04:15:00	0.872
14	1	1	2019-03-29 04:30:00	0.889
15	1	1	2019-03-29 04:45:00	0.9
16	1	1	2019-03-29 05:00:00	0.901
17	1	1	2019-03-29 05:15:00	0.904
18	1	1	2019-03-29 05:30:00	0.904
19	1	1	2019-03-29 05:45:00	0.904
20	1	1	2019-03-29 06:00:00	0.905
21	1	1	2019-03-29 06:30:00	0.905
22	1	1	2019-03-29 06:45:00	0.905

Total rows: 1000 of 8095524 Query complete 00:00:06.889

SELECT * FROM building_meta1;

	campus_id text	building_meter_id text	built_year integer	category text	gross_floor_area numeric	room_area numeric	capacity numeric
1	1	1	1899	other	145392.17	5174.47	644
2	1	2	1899	other	145392.17	5174.47	644
3	1	3	1899	other	145392.17	5174.47	644
4	1	4	1967	mixed use	145558.14	1790.17	79
5	1	5	1899	other	0	5174.47	644
6	1	6	1998	other	6210.75	473.03	0
7	1	7	1980	teaching	176996.61	2377.77	799
8	2	8	1998	library	2395	5174.47	644
9	2	9	1990	teaching	952	5174.47	644
10	2	10	2007	teaching	309	5174.47	644
11	2	11	1994	teaching	2682	5174.47	644
12	2	12	2007	other	154	5174.47	644
13	2	13	1999	teaching	1404	5174.47	644
14	2	14	2004	mixed use	1245	5174.47	644
15	1	15	1967	other	15728.4	1709.43	0
16	1	16	1992	mixed use	786493.89	5681.03	737
17	1	17	1967	mixed use	1259005.45	5129.22	616
18	1	18	1972	mixed use	1405706.16	5353.08	775
19	1	19	2006	leased	600	5174.47	644
20	1	20	1973	residence	42646.4	871.42	0
21	1	21	1997	other	265	5174.47	644
22	2	22	2012	mixed use	558	5174.47	644

Total rows: 64 of 64 Query complete 00:00:00.101

SELECT * FROM building_submeter_consumption1;

	building_meter_id text	building_submeter_id text	campus_id text	timestamp timestamp without time zone	building_submeter_consumption numeric	current numeric	voltage numeric	power numeric
1	14	1	2	2021-04-20 19:05:00	0.2256	3.96	240.0	2.7073
2	14	1	2	2021-04-20 19:10:00	0.201	3.53	240.0	2.411800000000004
3	14	1	2	2021-04-20 19:15:00	0.244	4.28	240.0	2.9284
4	14	1	2	2021-04-20 19:20:00	0.1886	3.31	240.0	2.2636
5	14	1	2	2021-04-20 19:25:00	0.2501	4.39	240.0	3.0009
6	14	1	2	2021-04-20 19:30:00	0.2016	3.54	240.0	2.4191
7	14	1	2	2021-04-20 19:35:00	0.1993	3.5	240.0	2.3922
8	14	1	2	2021-04-20 19:40:00	0.2058	3.61	240.0	2.4699
9	14	1	2	2021-04-20 19:45:00	0.2593999999999996	4.55	240.0	3.1127
10	14	1	2	2021-04-20 19:50:00	0.1953	3.43	240.0	2.3436
11	14	1	2	2021-04-20 19:55:00	0.1894	3.32	240.0	2.2729
12	14	1	2	2021-04-20 20:00:00	0.2627	4.61	240.0	3.1528
13	14	1	2	2021-04-20 20:05:00	0.2332	4.09	240.0	2.7982
14	14	1	2	2021-04-20 20:10:00	0.2001	3.51	240.0	2.4008000000000003
15	14	1	2	2021-04-20 20:15:00	0.1976	3.47	240.0	2.3707
16	14	1	2	2021-04-20 20:20:00	0.2423000000000002	4.25	240.0	2.9079
17	14	1	2	2021-04-20 20:25:00	0.2100999999999998	3.69	240.0	2.5215
18	14	1	2	2021-04-20 20:30:00	0.1716	3.01	240.0	2.0595
19	14	1	2	2021-04-20 20:35:00	0.299	5.25	240.0	3.588
20	14	1	2	2021-04-20 20:40:00	0.2251	3.95	240.0	2.7009000000000003
21	14	1	2	2021-04-20 20:45:00	0.2166999999999998	3.8	240.0	2.5999
22	14	1	2	2021-04-20 20:50:00	0.2058	3.61	240.0	2.4690

Total rows: 1000 of 1312464 Query complete 00:00:01.732

SELECT * FROM calender1;

	date date	is_holiday text	is_semester text	is_exam text
1	2016-01-01	1.0	0	0
2	2016-01-02	0.0	0	0
3	2016-01-03	0.0	0	0
4	2016-01-04	0.0	0	0
5	2016-01-05	0.0	0	0
6	2016-01-06	0.0	0	0
7	2016-01-07	0.0	0	0
8	2016-01-08	0.0	0	0
9	2016-01-09	0.0	0	0
10	2016-01-10	0.0	0	0
11	2016-01-11	0.0	0	0
12	2016-01-12	0.0	0	0
13	2016-01-13	0.0	0	0
14	2016-01-14	0.0	0	0
15	2016-01-15	0.0	0	0
16	2016-01-16	0.0	0	0
17	2016-01-17	0.0	0	0
18	2016-01-18	0.0	0	0
19	2016-01-19	0.0	0	0
20	2016-01-20	0.0	0	0
21	2016-01-21	0.0	0	0
22	2016-01-22	0.0	0	0

Total rows: 1000 of 2312 Query complete 00:00:00.176

SELECT * FROM campus_meta1;

	campus_id text	name text	capacity integer
1	1	Bundoora	26000
2	2	Albury-Wodonga	800
3	3	Bendigo	5000
4	4	Mildura	500
5	5	Shepparton	700

Total rows: 5 of 5 Query complete 00:00:00.075

SELECT * FROM events1;

	building_meter_id text	event_type text	date date	event_description text
1	4	Misc	2021-07-01	Decommissioned the L1 AHU
2	5	HVAC_Tuning	2019-05-10	HVAC System Tuning
3	5	HVAC_Tuning	2020-04-21	HVAC System Tuning
4	6	COVID_19_Building_Shutdown	2020-04-02	Building Shutdown due to the COVID-19 Lockdown
5	7	HVAC_Tuning	2019-06-07	HVAC System Tuning
6	7	HVAC_Tuning	2020-03-07	HVAC System Tuning
7	8	COVID_19_Building_Shutdown	2020-04-02	Building Shutdown due to the COVID-19 Lockdown
8	9	COVID_19_Building_Shutdown	2020-03-07	Building Shutdown due to the COVID-19 Lockdown
9	10	COVID_19_Building_Shutdown	2020-03-17	Building Shutdown due to the COVID-19 Lockdown
10	11	COVID_19_Building_Shutdown	2020-03-29	Building Shutdown due to the COVID-19 Lockdown
11	12	COVID_19_Building_Shutdown	2020-03-14	Building Shutdown due to the COVID-19 Lockdown
12	13	COVID_19_Building_Shutdown	2020-04-29	Building Shutdown due to the COVID-19 Lockdown
13	13	Misc	2021-01-01	Replaced chiller (High efficiency)
14	14	COVID_19_Building_Shutdown	2020-03-08	Building Shutdown due to the COVID-19 Lockdown
15	15	COVID_19_Building_Shutdown	2020-03-17	Building Shutdown due to the COVID-19 Lockdown
16	16	HVAC_Tuning	2020-02-27	HVAC System Tuning
17	16	HVAC_Tuning	2019-06-02	HVAC System Tuning
18	16	HVAC_Tuning	2020-03-15	HVAC System Tuning
19	17	HVAC_Tuning	2019-05-15	HVAC System Tuning
20	17	HVAC_Tuning	2020-03-14	HVAC System Tuning
21	17	HVAC_Tuning	2020-11-20	HVAC System Tuning
22	18	LED_Installation	2019-10-21	Completion of LED Installation

Total rows: 106 of 106 Query complete 00:00:00.095

SELECT * FROM gas_consumption1;

	campus_id text	timestamp timestamp without time zone	gas_consumption numeric
1	1	2018-05-01 06:00:00	24.8502085
2	1	2018-05-01 07:00:00	26.40453854
3	1	2018-05-01 08:00:00	45.34679348
4	1	2018-05-01 09:00:00	38.38303117
5	1	2018-05-01 10:00:00	32.77770508
6	1	2018-05-01 11:00:00	31.09043495
7	1	2018-05-01 12:00:00	30.99728443
8	1	2018-05-01 13:00:00	30.90519093
9	1	2018-05-01 14:00:00	29.48874635
10	1	2018-05-01 15:00:00	30.25049274
11	1	2018-05-01 16:00:00	28.8730414
12	1	2018-05-01 17:00:00	26.71166618
13	1	2018-05-01 18:00:00	24.52046309
14	1	2018-05-01 19:00:00	19.2480845
15	1	2018-05-01 20:00:00	21.59602233
16	1	2018-05-01 21:00:00	20.06535957
17	1	2018-05-01 22:00:00	21.54248084
18	1	2018-05-01 23:00:00	20.84236222
19	1	2018-05-02 00:00:00	21.57242645
20	1	2018-05-02 01:00:00	20.07673293
21	1	2018-05-02 02:00:00	20.80524147
22	1	2018-05-02 03:00:00	20.74697448

Total rows: 1000 of 25747 Query complete 00:00:00.124

SELECT * FROM nmi_consumption1;

	campus_id text	nmi_meter_id text	timestamp timestamp without time zone	nmi_consumption numeric	demand_kw numeric	demand_kva numeric
1	1	1	2015-11-22 17:15:00	17.3	69.2	76.926
2	1	1	2015-11-22 17:30:00	23.0	92.0	92.886
3	1	1	2015-11-22 17:45:00	28.3	113.2	125.249
4	1	1	2015-11-22 18:00:00	27.1	108.4	113.254
5	1	1	2015-11-22 18:15:00	23.2	92.8	93.473
6	1	1	2015-11-22 18:30:00	25.3	101.2	105.109
7	1	1	2015-11-22 18:45:00	25.5	102.0	106.549
8	1	1	2015-11-22 19:00:00	20.2	80.8	81.086
9	1	1	2015-11-22 19:15:00	24.6	98.4	102.416
10	1	1	2015-11-22 19:30:00	24.8	99.2	102.351
11	1	1	2015-11-22 19:45:00	23.9	95.6	96.401
12	1	1	2015-11-22 20:00:00	27.4	109.6	115.114
13	1	1	2015-11-22 20:15:00	22.6	90.4	90.422
14	1	1	2015-11-22 20:30:00	23.0	92.0	92.251
15	1	1	2015-11-22 20:45:00	26.7	106.8	111.491
16	1	1	2015-11-22 21:00:00	22.5	90.0	90.032
17	1	1	2015-11-22 21:15:00	27.3	109.2	114.981
18	1	1	2015-11-22 21:30:00	22.4	89.6	89.622
19	1	1	2015-11-22 21:45:00	26.8	107.2	111.989
20	1	1	2015-11-22 22:00:00	21.9	87.6	87.6
21	1	1	2015-11-22 22:15:00	26.6	106.4	111.459
22	1	1	2015-11-22 22:30:00	21.9	87.6	87.6

Total rows: 1000 of 1048575 Query complete 00:00:01.001

SELECT * FROM nmi_meta1;

	nmi_meter_id text	campus_id text	peak_demand numeric
1	1	1	1225.89
2	2	4	121.0
3	3	5	145.0
4	4	3	143.0
5	5	1	260.0
6	6	2	100.0
7	7	1	90.0
8	8	2	930.0
9	9	1	22.68
10	10	3	626.0
11	11	3	145.0
12	12	1	7659.0
13	13	1	120.0
14	14	1	192.751

Total rows: 14 of 14 Query complete 00:00:00.107

SELECT * FROM water_consumption1;

	campus_id text	water_meter_id text	timestamp timestamp without time zone	water_consumption numeric
1	1	1	2021-01-10 00:00:00	438.32800000000003
2	1	1	2021-01-10 00:15:00	438.32800000000003
3	1	1	2021-01-10 00:30:00	438.32800000000003
4	1	1	2021-01-10 00:45:00	438.32800000000003
5	1	1	2021-01-10 01:00:00	438.32800000000003
6	1	1	2021-01-10 01:15:00	438.32800000000003
7	1	1	2021-01-10 01:30:00	438.32800000000003
8	1	1	2021-01-10 01:45:00	438.32800000000003
9	1	1	2021-01-10 02:00:00	438.32800000000003
10	1	1	2021-01-10 02:15:00	438.32800000000003
11	1	1	2021-01-10 02:30:00	438.32800000000003
12	1	1	2021-01-10 02:45:00	438.32800000000003
13	1	1	2021-01-10 03:00:00	438.32800000000003
14	1	1	2021-01-10 03:15:00	438.32800000000003
15	1	1	2021-01-10 03:30:00	438.32800000000003
16	1	1	2021-01-10 03:45:00	438.32800000000003
17	1	1	2021-01-10 04:00:00	438.32800000000003
18	1	1	2021-01-10 04:15:00	438.32800000000003
19	1	1	2021-01-10 04:30:00	438.32800000000003
20	1	1	2021-01-10 04:45:00	438.32800000000003
21	1	1	2021-01-10 05:00:00	438.32800000000003
22	1	1	2021-01-10 05:15:00	438.32800000000003

Total rows: 1000 of 488058 Query complete 00:00:00.441

SELECT * FROM weather_data1;

	campus_id text	timestamp timestamp without time zone	apparent_temperature numeric	air_temperature numeric	dew_point_temperature numeric	relative_humidity numeric
1	1	2018-01-01 00:00:00	16.6	16.2	13.5	84.0
2	1	2018-01-01 00:01:00	17.2	16.1	13.6	85.0
3	1	2018-01-01 00:02:00	16.9	16.1	13.6	85.0
4	1	2018-01-01 00:03:00	16.9	16.1	13.6	85.0
5	1	2018-01-01 00:04:00	16.0	16.0	13.5	85.0
6	1	2018-01-01 00:05:00	16.7	16.0	13.5	85.0
7	1	2018-01-01 00:06:00	16.7	16.0	13.5	85.0
8	1	2018-01-01 00:07:00	16.7	16.0	13.5	85.0
9	1	2018-01-01 00:08:00	16.6	15.9	13.4	85.0
10	1	2018-01-01 00:09:00	16.6	15.9	13.4	85.0
11	1	2018-01-01 00:10:00	16.6	15.9	13.4	85.0
12	1	2018-01-01 00:11:00	17.0	15.9	13.4	85.0
13	1	2018-01-01 00:12:00	17.0	15.9	13.4	85.0
14	1	2018-01-01 00:13:00	17.0	15.9	13.6	86.0
15	1	2018-01-01 00:14:00	16.3	15.9	13.6	86.0
16	1	2018-01-01 00:15:00	16.0	15.9	13.6	86.0
17	1	2018-01-01 00:16:00	16.7	15.9	13.6	86.0
18	1	2018-01-01 00:17:00	17.0	15.9	13.6	86.0
19	1	2018-01-01 00:18:00	16.3	15.9	13.6	86.0
20	1	2018-01-01 00:19:00	15.8	15.8	13.5	86.0
21	1	2018-01-01 00:20:00	16.3	15.9	13.6	86.0
22	1	2018-01-01 00:21:00	16.0	15.9	13.6	86.0

Total rows: 1000 of 7396520 Query complete 00:00:09.119

3.0 Database

3.1 Relational Model

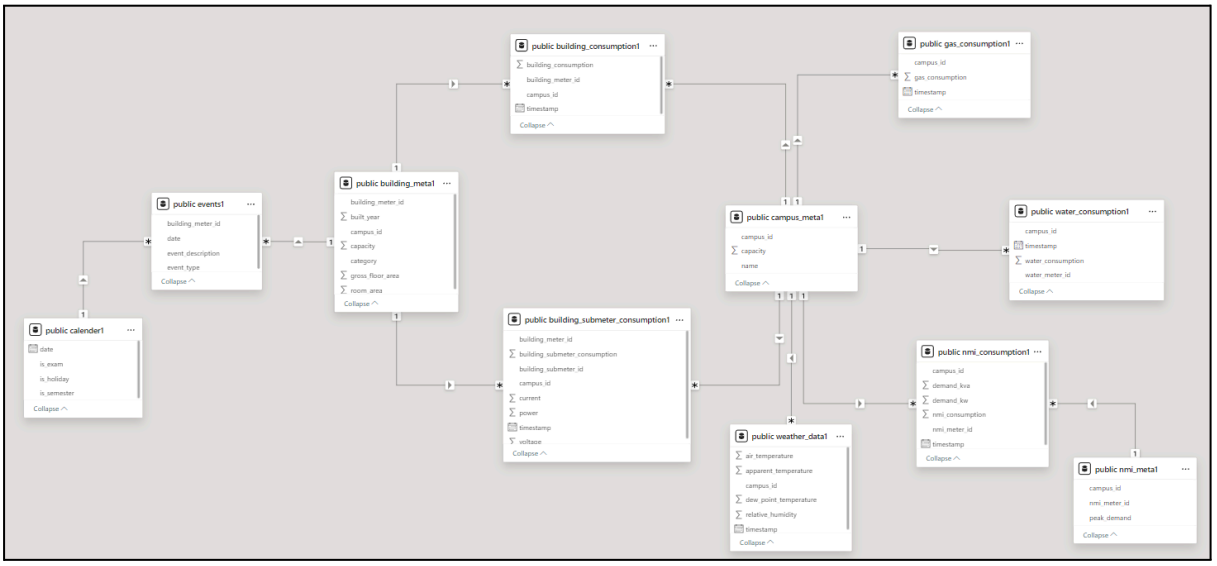


Figure 3.1.1: Relational model using PowerBI

The figure shows the relational model using Power BI. It involves understanding the structure of the data and also the relationship between the different tables and entities. Each table represents a specific category of data, like public gas consumption, public water consumption, public buildings, and others. The tables are linked together based on their common fields. It will allow for easy navigation and analysis of related data points. By analysing the relationships of the data, we may find the key insights and patterns of the consumption data. In addition, measures and calculated columns can provide additional context by performing calculations or collecting the data. For example, the sum of capacity, the sum of voltage consumption, and others. Besides that, we can gain deeper insights into the data model by exploring it interactively, filtering information, and also drilling down into specific details.

3.2 Relational Between Model

Data	Relationship
campus meta → gas consumption	one to many
campus meta → water consumption	one to many
campus meta → nmi consumption	one to many
campus meta → nmi meta	many to many
campus meta → weather	one to many
campus meta → building consumption	one to many
campus meta → building submeter consumption	one to many
campus meta → building meta	many to many
building meta → building consumption	one to many
building meta → building submeter consumption	one to many
building meta → events	one to many
building meta → calender	many to many

3.3 Identification Of Data Warehouse Schema

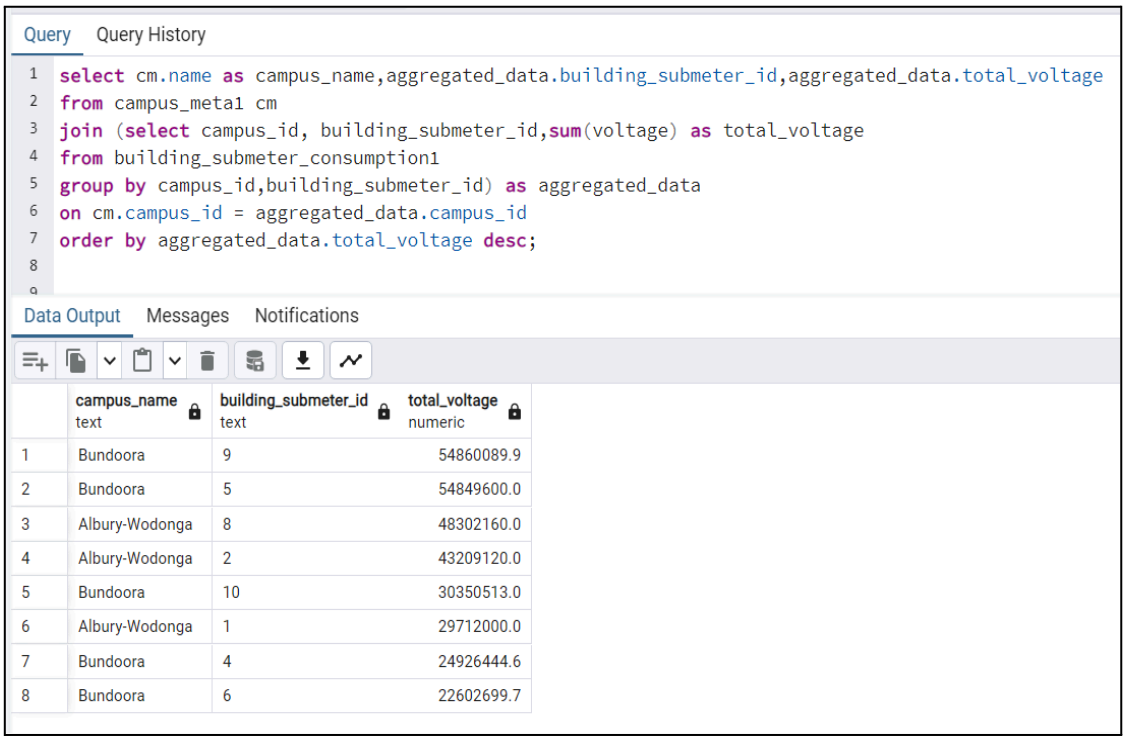
Based on Figure 3.1.1, Galaxy schema, also known as Fact Constellation schema is the data warehouse schema used for these datasets. It is because this schema represents the multidimensional model with multiple fact tables that share their dimension table. This schema represents complex and interconnected business processes, with multiple-star schemas linked by the common dimension tables. The fact tables in this case are public building meta1 and public campus meta1. Then, the dimension table that connects both of the fact tables is public building consumption1 and public building submeter consumption1. Next, the fact table of public building meta1 connects with public events1 and public calender1 while the fact table of public campus meta1 connects with public weather data1, public nmi consumption1, public nmi meta1, public water consumption1, and public gas consumption1. The galaxy schema allows for more flexible and complete querying as well as visualisation because it can accommodate a wider range of data analysis scenarios compared to simpler schemas like the star or snowflake schemas.

4.0 Result and Data Analysis

After the data integration stage, we used Power BI to create dynamic visualisations that bring insights to life. Furthermore, we used PostgreSQL to perform OLAP techniques like rollup, dicing, pivoting, and slicing to conduct a deeper analysis that revealed new insights within our dataset.

4.1 OLAP Operations

4.1.1 Rollup Operation



Query | Query History

```
1  select cm.name as campus_name,aggregated_data.building_submeter_id,aggregated_data.total_voltage
2  from campus_meta1 cm
3  join (select campus_id, building_submeter_id,sum(voltage) as total_voltage
4  from building_submeter_consumption1
5  group by campus_id,building_submeter_id) as aggregated_data
6  on cm.campus_id = aggregated_data.campus_id
7  order by aggregated_data.total_voltage desc;
8
```

Data Output | Messages | Notifications

	campus_name text	building_submeter_id text	total_voltage numeric
1	Bundoora	9	54860089.9
2	Bundoora	5	54849600.0
3	Albury-Wodonga	8	48302160.0
4	Albury-Wodonga	2	43209120.0
5	Bundoora	10	30350513.0
6	Albury-Wodonga	1	29712000.0
7	Bundoora	4	24926444.6
8	Bundoora	6	22602699.7

Figure 4.1.1.1: Rollup Operation

The picture shows a table of data from the UNICON dataset, a key resource for La Trobe University's efforts to achieve Net Zero Carbon emissions by 2029. The table consists of the campus name, building submeter ID, and total voltage for various entries. There are two campus names listed Bundoora and Albury-Wodonga with multiple building submeter IDs associated with each campus. The total voltage indicates differences in energy consumption. the highest total voltage is 54860089.3 which is being accumulated by campus Bundoora that submeter ID 9. Meanwhile, the lowest voltage consumption is 22602699.7 by campus Bundoora with building submeter ID 6. By analysing the data from the UNICON dataset, La Trobe University can identify high-consumption areas and develop targeted strategies to reduce energy usage and carbon emissions, ultimately helping the university achieve its sustainability goals.

4.1.2 Dicing Operation

Query | Query History

```
1  select event_type, category, sum (building_consumption) as sum_of_building_consumption
2  from building_meta1 as m inner join events1 as e
3  on m. building_meter_id = e. building_meter_id
4  inner join building_consumption1 as b
5  on e. building_meter_id = b.building_meter_id
6  where event_type = 'LED_Installation' and category = 'teaching'
7  group by (category, event_type)
```

Data Output | Messages | Notifications

	event_type text	category text	sum_of_building_consumption numeric
1	LED_Installation	teaching	1619543.78125

Figure 4.1.2.1: Dicing Operation

The information above appears to be a table displaying the results of a query on a database containing data related to energy consumption at La Trobe University. We calculate the total energy consumption for a specific type of event that includes LED installation by categorising teaching. This data originates from three tables such as building_meta1, events1, and building_consumption1 that joined based on their building meter ID. The results show that there were LED installation events in the teaching category, which had a total energy consumption of approximately 1.62 million units. This data is relevant to the project background as it demonstrates the use of data analytics and artificial intelligence to optimise energy consumption at La Trobe University, specifically in this case by analysing the energy savings from LED installation events in the teaching category.

4.1.3 Slicing Operation

Query | Query History

```
SELECT
    e.building_meter_id,
    e.event_type,
    e.date,
    e.event_description
FROM
    events1 e
WHERE
    e.building_meter_id = '18' AND e.date BETWEEN '2019-01-01' AND '2020-12-31';
```

Data Output | Messages | Notifications

	building_meter_id text	event_type text	date date	event_description text
1	18	LED_Installation	2019-10-21	Completion of LED Installation
2	18	HVAC_Tuning	2019-05-17	HVAC System Tuning
3	18	COVID_19_Building_Shutdown	2020-03-14	Building Shutdown due to the COVID-19 Lockdown

Figure 4.1.3.1: Slicing Operation

Figure 4.1.3.1 shows the Event data from the events1 table using the given OLAP slicing code, which focuses on events related to building meter ID 18 from January 1, 2019, to December 31, 2020. The output table shows a variety of event types, such as 'LED Installation', 'HVAC Tuning', and 'COVID-19 Building Shutdown' provides insights into maintenance and operational changes during the time period. These data are important to monitor building performance and support sustainability efforts. This enables stakeholders to track energy-saving strategies like LED installations and estimate the impact of external factors like pandemic-related shutdowns. Stakeholders can make decisions to optimise building operations by utilising these event tracking which reduce energy consumption and work towards achieving sustainability goals such as La Trobe University's Net Zero Carbon Emission target by 2029.

4.1.4 Pivot Operation

Query | Query History

```
CREATE EXTENSION IF NOT EXISTS tablefunc;
SELECT *
FROM crosstab(
 'SELECT "name", EXTRACT(MONTH FROM "timestamp"), SUM("building_consumption")
 FROM campus_meta1 c, building_consumption1 b
 WHERE c.campus_id = b.campus_id
 GROUP BY (1,2)
 ORDER BY (1,2)',
 'VALUES(''1''), (''2''), (''3''), (''4''), (''5''), (''6''), (''7''), (''8''), (''9''), (''10''), (''11''), (''12'')'
) AS ("name" text, "1" REAL, "2" REAL, "3" REAL, "4" REAL,
      "5" REAL, "6" REAL, "7" REAL, "8" REAL,
      "9" REAL, "10" REAL, "11" REAL, "12" REAL);
```

Data Output | Messages | Notifications

	name text	1 real	2 real	3 real	4 real	5 real	6 real	7 real	8 real	9 real	10 real	11 real	12 real
1	Albury-Wodonga	596183.1	565362.56	545438.06	413154.8	307687.34	284862.94	253161.48	201644.69	249996.36	263499.6	272214.72	
2	Bendigo	36343.117	62510.93	87947.44	79631.99	95270.57	101559.78	102522.766	111307.1	93481.21	85115.57	54363.375	
3	Bundoora	9.694084e+06	9.43955e+06	1.0200289e+07	9.257603e+06	8.1091445e+06	7.561483e+06	7.7920105e+06	7.9832035e+06	7.552211e+06	7.9315295e+06	7.6175865e+06	7.5

Figure 4.1.4.1: Pivot Operation

The building consumption data from the ‘building_consumption1’ table is transformed efficiently using the provided SQL pivot code for monthly aggregate and comparison across campuses. To obtain the month from the timestamp, combine ‘campus_meta1’ and ‘building_consumption1’ on ‘campus_id’, and then for grouping by campus name and month, the query aggregates the building consumption for each group. After the data is pivoted, the ‘crosstab’ method provides a table consisting of the names of campuses and their monthly consumption from January to December. Based on the output, Bundoora has consumed more energy compared to other campuses. With this query, stakeholders can track and contrast monthly patterns in energy consumption, identify trends and abnormalities, and create energy-saving strategies. At last, this project supports more general sustainability goals by helping to optimise building operations and reduce the overall carbon footprint.

4.2 Power BI Visualisation

4.2.1 Sum of Building Consumption by Year

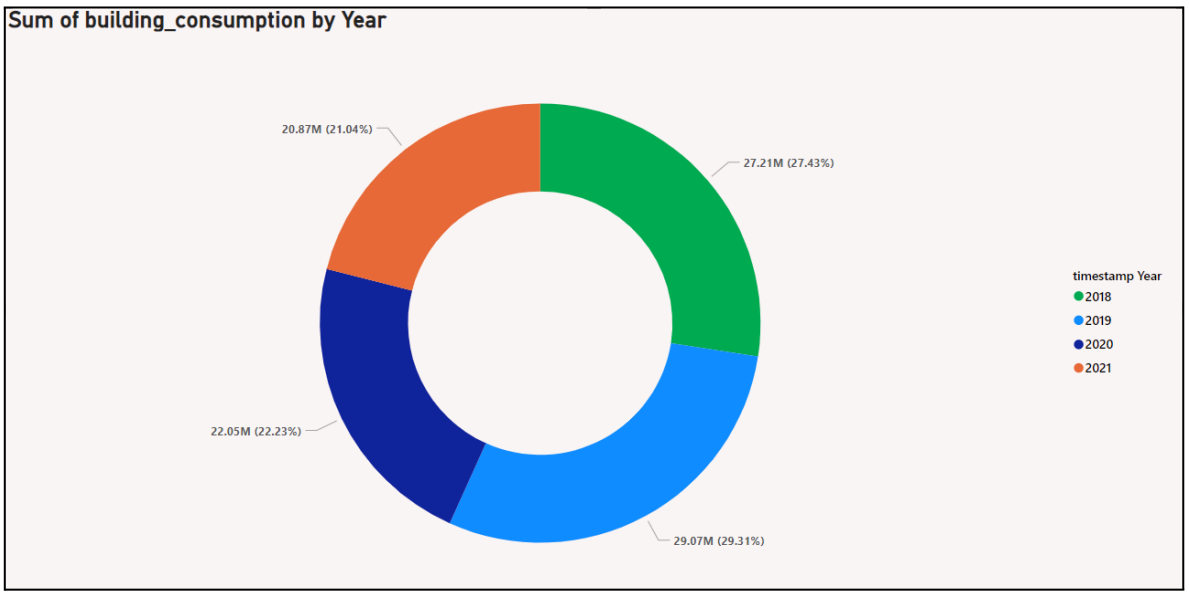


Figure 4.2.1.1: Sum of building consumption by year

Based on Figure 4.2.1.1, the donut chart reveals the sum of building consumption trends by year which is from 2018 until 2021. Each slice represents the sum and the percentage of the building consumption for a specific year with different colours compared to energy use over 4 years. The largest slice represents the highest value of energy consumption in the building for a specific year while the smallest slice represents the lowest value of energy consumption in the building for a specific year. In this case, the largest slice represents 2019 with the highest energy consumption in the building which is 29.97 million usage which represents 29.31% of energy consumption compared to 2018, 2020, and 2021. Besides that, in 2021, the sum of energy use in the building shows the lowest consumption with 20.87 million usage which is 21.04% of energy consumption in the building compared to other years. It could happen due to the reduced building usage or improvement of the energy efficiency measured. At the same time, in 2021, COVID-19 has hit the world, causing energy consumption in universities to decrease as the number of students there is decreasing.

4.2.2 Sum of Water Consumption by Month and Water Meter ID

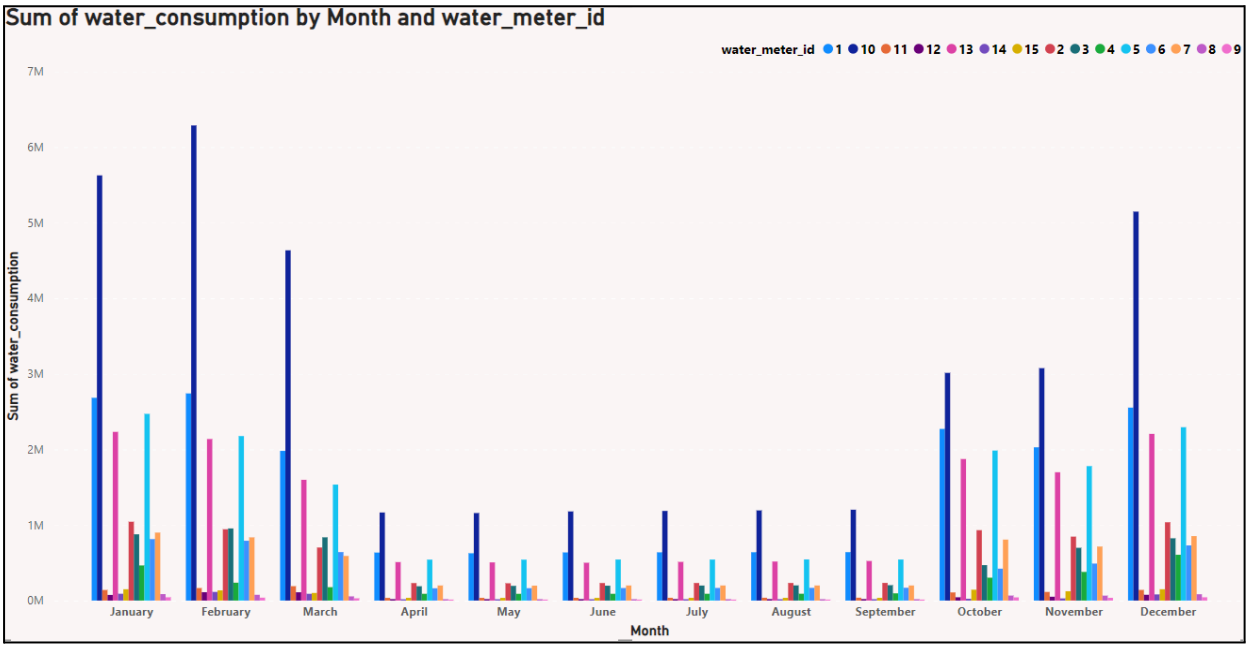


Figure 4.2.1.1: Sum of water consumption by month and water meter ID

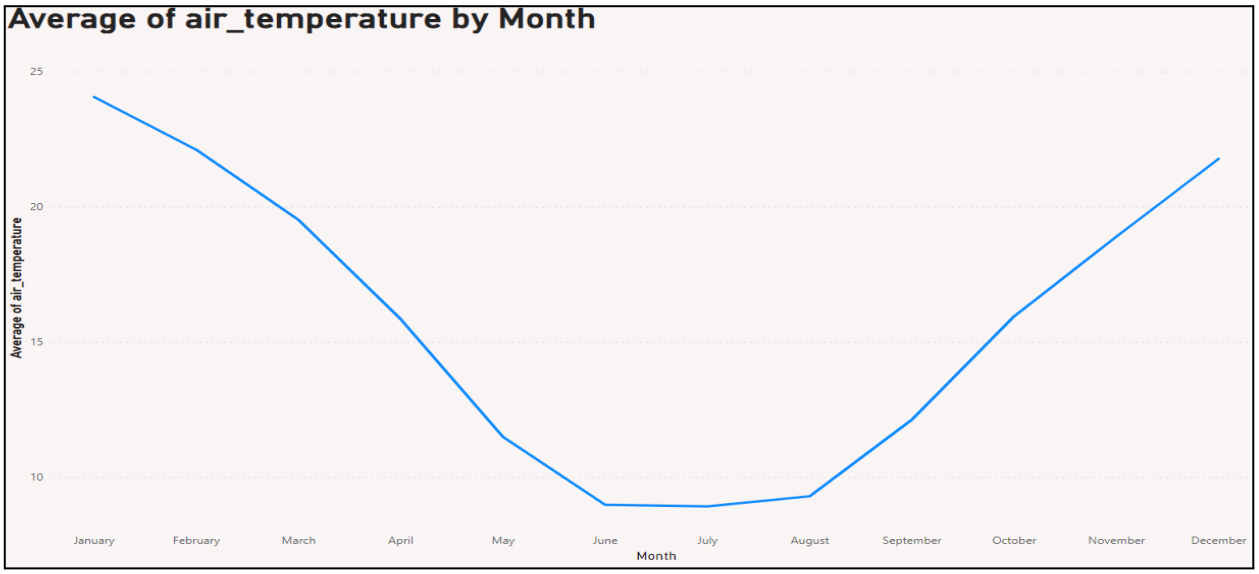


Figure 4.2.1.2: Average of air temperature by month

Figure shows the side-by-side bar graph that represents the sum of water consumption in La Trobe University by month and meter IDs. It provides a comprehensive visual representation of water consumption patterns over the month and across the multiple meter IDs. Each water meter ID is represented by a different colour each month to enable easier visual comparison between the various meter IDs. The water meter ID “10” in February recorded the highest water consumption in La Trobe University, with the amount of

6,285,445.80 cubic metres. It is caused by the increase in the number of new students intakes in February. In addition, in February, the lowest amount of water consumption was recorded by water meter ID 9, which is only 38,134 cubic metres of total water consumption in that month. It is due to the lowest consumption of water in that area. Overall analysis, the lowest water consumption was recorded in May compared to other months. In May, the highest consumption was recorded by water meter ID 10 with 1,157,449.48 cubic meters in that month. The lowest consumption was also recorded by water meter ID 9 with 10,215.41 cubic meters. In addition, the shape of the histogram graph from January until December of water consumption is shown as U-shaped. From April until September, the trend of water consumption is consistent at the lowest consumption. It is due to the air temperature changes throughout that month. We can see from the Figure 4.2.1.2 that shows the line chart that the average air temperature from April until September is dropping until the lowest temperature. So, based on these two figures, we can conclude that water consumption is related to the air temperature. When the air temperature is high, the water consumption is increasing while when the air temperature drops, the water consumption is decreasing.

4.2.3 Average Energy Consumption by Campus and Building Category

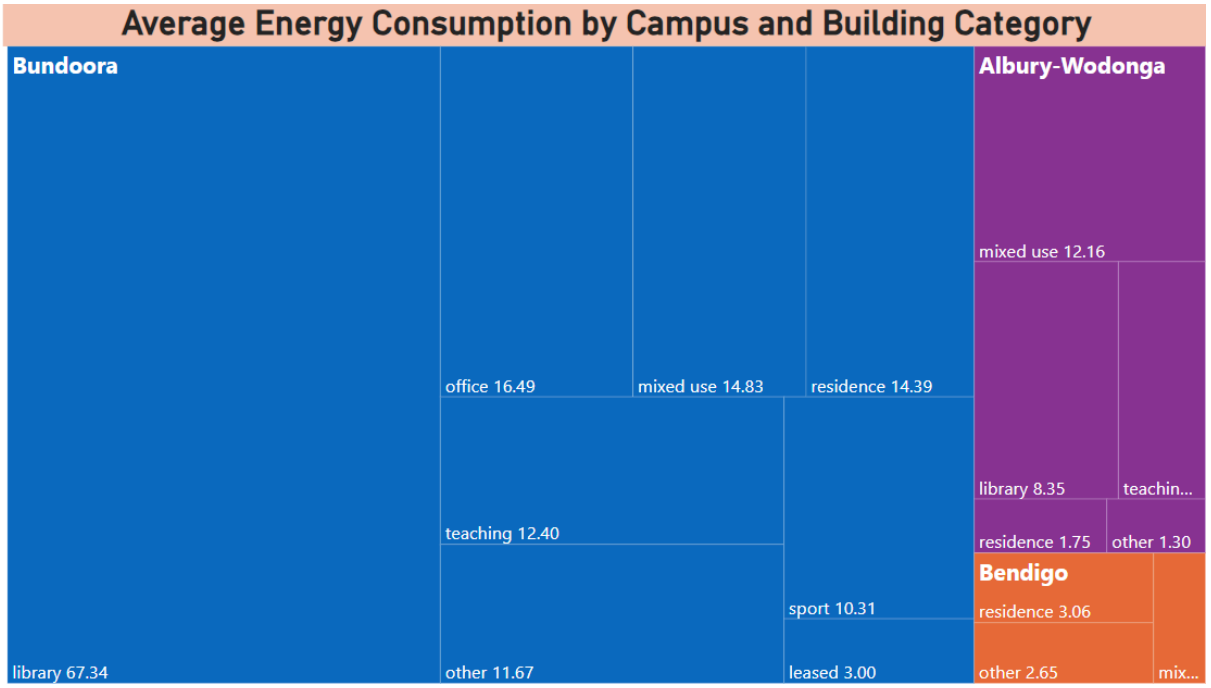


Figure 4.2.3.1: Average Energy Consumption by Campus and Building Category

The figure above shows the average energy consumption by campus and building. The campus includes Bundoora, Albury-Wodonga and Bendigo. Among these campuses, Bundoora campus has consumed the highest average amount of energy whereas Bendigo campus has consumed the lowest amount of energy. Bundoora campus consists of several buildings including a library, office, mixed-use, residence, teaching, other, sport and leased buildings. In the Bundoora campus, the library building has consumed the highest amount of energy with 67.34kWh whereas the leased building has consumed the lowest amount of energy with 3.00kWh. The library has to operate continuously for long hours compared to other buildings so it requires more energy for air-conditioning and lighting. Besides, Albury-Wodonga consists of several buildings including mixed-use, library, teaching, residence and other buildings. In the Albury-Wodonga campus, mixed-use buildings have consumed the highest amount of energy with 12.16kWh. Apart from that, Bendigo consists of several buildings including residence, mixed-use and other buildings. In the Bendigo campus, residence building consumes the highest energy of 3.06kWh whereas the mixed-use building consumes the lowest amount of energy reaching 1.65kWh. The residence building requires more energy for electric appliances which include lighting, water heaters, washing machines and refrigerators.

4.2.4 Sum of Electricity Consumption by Month

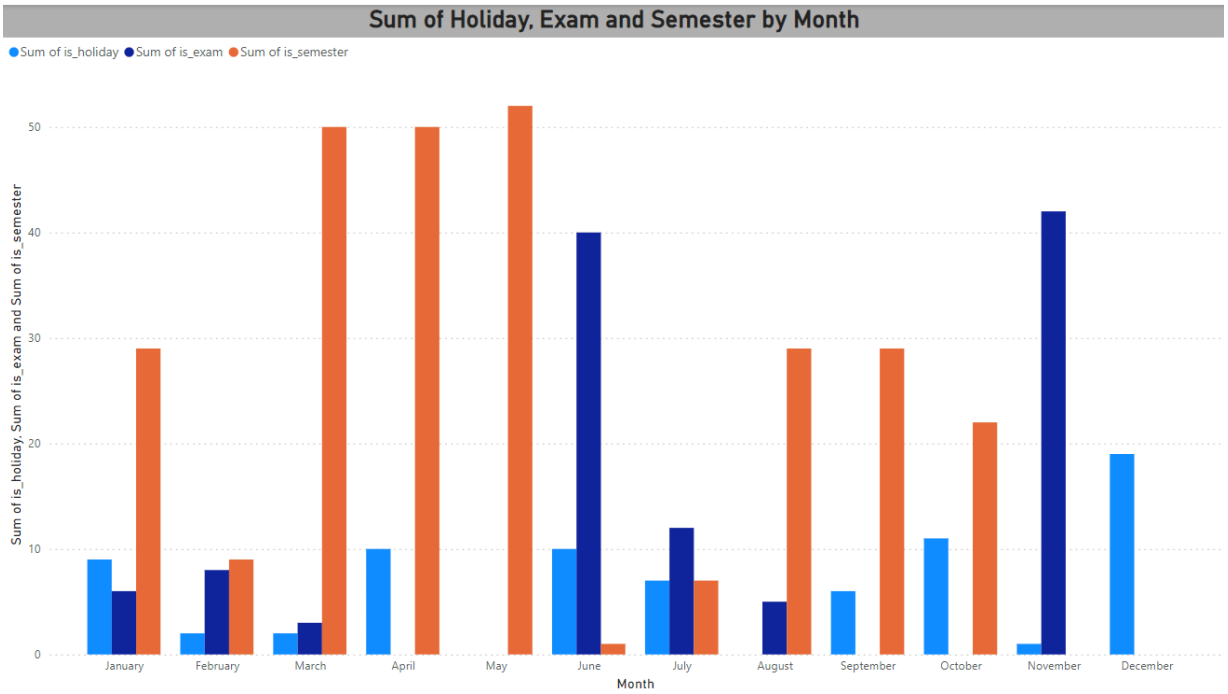


Figure 4.2.4.1: Sum of Holiday, Exam, and Semester by Month

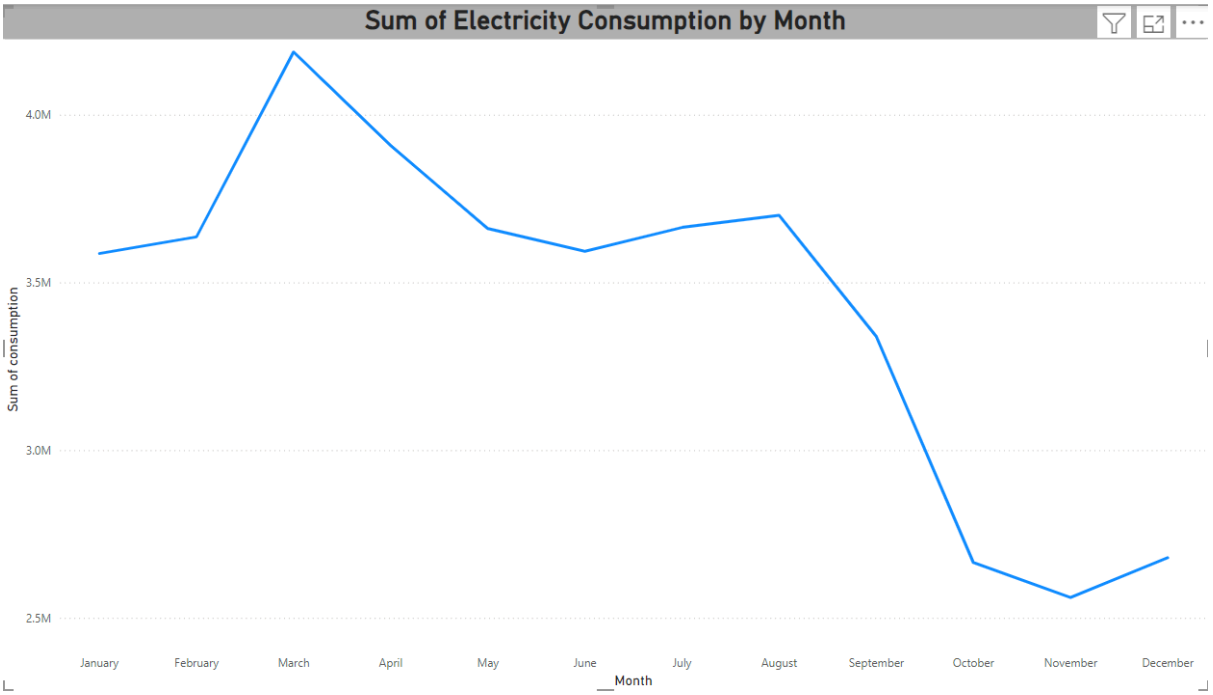


Figure 4.2.4.2: Sum of Electricity Consumption by Month

Based on the figures 4.2.4.1 and 4.2.4.2, the bar chart illustrates the sum of holidays, exams, and semester by month while the line chart illustrates the total electricity consumption by month. These two charts represent the relationship between academic activities and electricity consumption patterns over a year. The bar chart represents high peaks for the semester in May, for exams in June and November, and for holidays in December. The line chart shows a major increase in electricity consumption in April, just before the peak academic activities in June. This indicates that educational institutions use more electricity during preparation for their exams. By the end of year, electricity consumption starts to drop along with fewer academic activities and more holidays, showing a strong connection between the school's operating schedule and its resource utilisation.

4.2.5 Sum of Building, Building Submeter, Gas, Electricity and Water Consumption by Campus

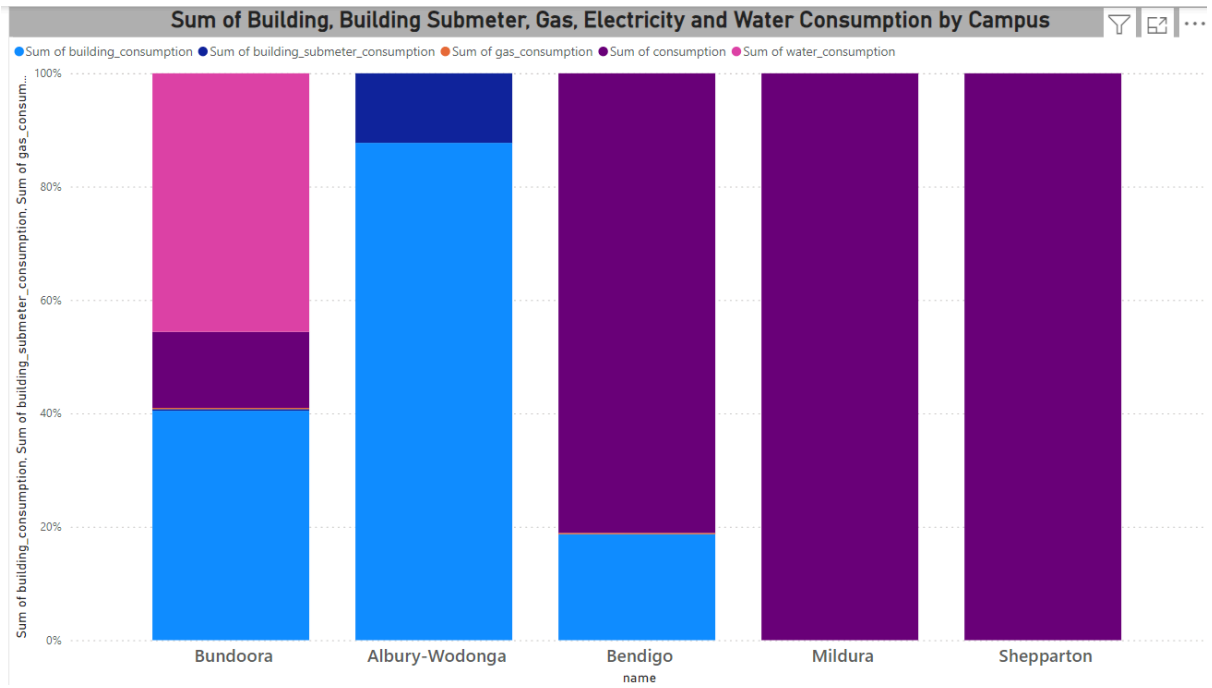


Figure 4.2.5.1: Sum of Building, Building Submeter, Gas, Electricity and Water Consumption by Campus

As indicated by the charts, it show the sum of building, building submeter, gas, electricity and water consumption by campus. The Albury-Wodonga campus has the highest building consumption at 87.73%, as well as the highest building submeter consumption at 12.27%. The Bendigo campus has the highest gas consumption at 0.23%. Both the Mildura and Shepparton campuses have the highest electricity consumption, each at 100%. The Bundoora campus leads in water consumption with 45.6%. This analysis highlights how different campuses have varying levels of utility usage across different categories. The Bendigo campus uses a lot of gas, probably for heating or other things. The Mildura and Shepparton campuses have the highest electricity consumption, which means that they need a lot of electrical power. The Bundoora campus uses a lot of water meaning that may be because it has big gardens or a big pool.

5.0 Conclusion

In conclusion, we found that the Bundoora campus contributes to the highest building consumption whereas the Albury-Wadonga campus contributes to the highest building submeter consumption. Among the Bundoora campus, Albury-Wodonga campus, Bendigo campus, Mildura campus and Shepparton campus, the Bundoora campus has contributed the highest energy consumption. Besides, the electricity consumption will be affected by the examination. If the total number of examinations in that month is high, the electricity consumption also will increase. Furthermore, we notice that the electricity is less being consumed during the university holiday. In addition, we found that the water consumption is high during the hot weather.

The recommendations of the research let the top managers make better decisions about the resources' usage and strengthen the sustainable development. This project can allow La Trobe University finding a way to manage the energy use and embrace sustainability goals. For instance, the university is advisable to use intelligent heating and LED lamps to reduce the energy consumption. In other cases, students and staff may suggest new behaviour such as turn off the lights and our laptops when we do not need them. This study can contribute to the achievement of sustainable development goals and provide useful information about energy consumption and usage behaviours of La Trobe University.

6.0 Reference

Achuo, E. D., Miamo, C. W., & Nchofoung, T. N. (2022). Energy consumption and environmental sustainability: What lessons for posterity? *Energy Reports*, *8*, 12491–12502. https://doi.org/10.1016/j.egyr.2022.09.033

Daud, S., Wan Hanafi, W. N., Ayodele, B. V., Rajadurai, J., Mustapa, S. I., Ahmad, N. N., Wan Abdullah, W. M., Toolib, S. N., Asha'ari, M. J., & Afsarizal, H. A. (2023). Residential consumers' lifestyle energy usage and energy efficiency in selected states in Malaysia. *Energies*, *16*(8), 3514. https://doi.org/10.3390/en16083514

Eakpoom Boonthum. (2018, October). (PDF) study of energy consumption in library building. https://www.researchgate.net/publication/328603233_STUDY_OF_ENERGY_CONSUMPTION_IN_LIBRARY_BUILDING

Environmental impact of energy. European Environment Agency. (2017, February 14). https://www.eea.europa.eu/help/glossary/eea-glossary/environmental-impact-of-energy

Teba, C. (2022, August 31). *What does energy consumption mean?*. Spacewell Energy (Dexma). https://www.dexma.com/blog-en/energy-consumption-definition/

7.0 APPENDIX

Queries to create tables for extracting raw data into PostgreSQL

```
CREATE TABLE building_consumption(
        campus_id text,
        meter_id text,
        "timestamp" timestamp,
        consumption numeric
);

CREATE TABLE building_meta(
        campus_id text,
        "id" text,
        built_year integer,
        category text,
        gross_floor_area numeric,
        room_area numeric,
        capacity numeric
);

CREATE TABLE building_submeter_consumption(
        building_id text,
        "id" text,
        campus_id text,
        "timestamp" timestamp,
        consumption numeric,
        "current" numeric,
        voltage numeric,
        "power" numeric,
        power_factor numeric
);
```

```
CREATE TABLE calender(
        "date" date,
        is_holiday text,
        is_semester text,
        is_exam text
);

CREATE TABLE campus_meta(
        "id" text,
        name" text,
        capacity integer
);

CREATE TABLE events(
        meter_id text,
        event_type text,
        "date" date,
        event_description text
);

CREATE TABLE gas_consumption(
        campus_id text,
        "timestamp" timestamp,
        consumption numeric
);

CREATE TABLE nmi_consumption(
        campus_id text,
        meter_id text,
        "timestamp" timestamp,
        consumption numeric,
        demand_kW numeric,
        demand_kVA numeric
);
```

```
CREATE TABLE nmi_meta(
        "id" text,
        campus_id text,
        peak_demand numeric
);

CREATE TABLE water_consumption(
        campus_id text,
        meter_id text,
        "timestamp" timestamp,
        consumption numeric
);

CREATE TABLE weather_data(
        campus_id text,
        "timestamp" timestamp,
        apparent_temperature numeric,
        air_temperature numeric,
        dew_point_temperature numeric,
        relative_humidity numeric,
        wind_speed numeric,
        wind_direction numeric
);
```

Queries for creating tables to load cleaned data into PostgreSQL

```
CREATE TABLE building_consumption1(
        campus_id text,
        building_meter_id text,
        "timestamp" timestamp,
        building_consumption numeric
);

CREATE TABLE building_meta1(
        campus_id text,
        building_meter_id text,
        built_year integer,
        category text,
        gross_floor_area numeric,
        room_area numeric,
        capacity numeric
);

CREATE TABLE building_submeter_consumption1(
        building_meter_id text,
        building_submeter_id text,
        campus_id text,
        "timestamp" timestamp,
        building_submeter_consumption numeric,
        "current" numeric,
        voltage numeric,
        "power" numeric
);

CREATE TABLE calender1(
        "date" date,
        is_holiday text,
        is_semester text,
```

```
    is_exam text
);

CREATE TABLE campus_meta1(
    campus_id text,
    "name" text,
    capacity integer
);

CREATE TABLE events1(
    building_meter_id text,
    event_type text,
    "date" date,
    event_description text
);

CREATE TABLE gas_consumption1(
    campus_id text,
    "timestamp" timestamp,
    gas_consumption numeric
);

CREATE TABLE nmi_consumption1(
    campus_id text,
    nmi_meter_id text,
    "timestamp" timestamp,
    consumption numeric,
    demand_kW numeric,
    demand_kVA numeric
);

CREATE TABLE nmi_meta1(
    nmi_meter_id text,
    campus_id text,
```

```
        peak_demand numeric
);

CREATE TABLE water_consumption1(
        campus_id text,
        water_meter_id text,
        "timestamp" timestamp,
        water_consumption numeric
);

CREATE TABLE weather_data1(
        campus_id text,
        "timestamp" timestamp,
        apparent_temperature numeric,
        air_temperature numeric,
        dew_point_temperature numeric,
        relative_humidity numeric
);
```