****

****

OPEN DATA DETAIL

AND

DATA MANAGEMENT PLAN

**TP 31**

Tao Xiang 30161002

Zicong Zheng 32409028

Ounam Gupta 31176097

Mohammad Yaseen Malik 23313404

Mustafa Malik 26366878

**CONTENT**

| S.No. | Title | Page No. |
| --- | --- | --- |
| 1 | Introduction | 1 |
| 2 | Data Description | 1 |
| 3 | Database | 3 |
| 4 | Database Schema | 4 |
| 5 | Database Queries | 4 |
| 5 | Data Collection and Storage  6.1 Data Processing and Presentation  6.2 Data Modelling | 6  7  12 |
| 6 | Data Risk Management | 14 |
| 7 | Data Preservation and Sharing  7.1 Data Preservation  7.2 Data Sharing | 15  15  15 |
| 8 | Data Backup and Archiving  10.1 Data Backup  10.2 Data Archiving | 15  15  16 |
| 10 | Data Ownership | 16 |

**1. Introduction**

This report details:

* Data Management - Data storage, backup & archiving, risks and other responsibilities.
* Data Collection - Open Data and user data
* Data Usage - Data Processing and modelling

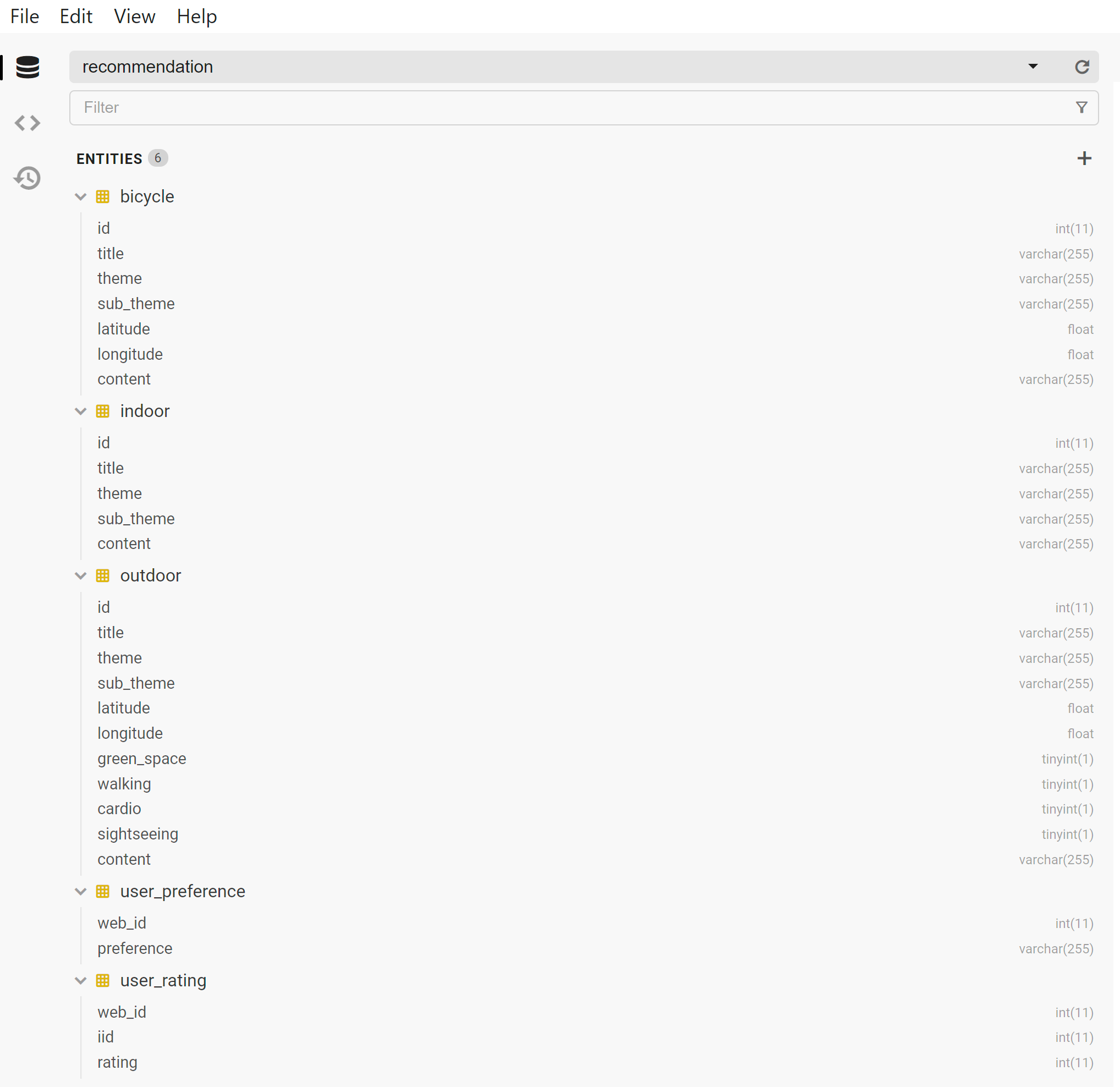
**2. Data Description**

In this project, we are using Open Datasets to visualise the statistics.

| OPEN DATASET | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Names** | **Physical Access** | **Frequency of Source Updates** | **Frequency of Iteration system updates** | **Granularity** | **Copyright/Licensing Details** |
| Outdoor artworks  https://data.melbourne.vic.gov.au/People/Outdoor-artworks/ue3p-kdsu | CSV | Not Provided (Latest by 2020) | Yearly | Precise location of activity | CC-BY |
| Parks and green spaces in Melbourne  <https://data.melbourne.vic.gov.au/People/Landmarks-and-places-of-interest-including-schools/j5vt-ppat/data> | CSV | Monthly | Yearly | Precise location of activity | CC-BY |
| Bicycle tracks in Melbourne  <https://data.melbourne.vic.gov.au/Transport/Bicycle-Network/3bxw-t8fk> | CSV | Not Provided (Latest 2021) | Yearly | Coordinate data for track | CC-BY |
| Bicycle tracks in Melbourne  <https://data.melbourne.vic.gov.au/Transport/Bicycle-routes-including-informal-on-road-and-off-/24aw-nd3i> | CSV | Quarterly | Yealy | Coordinate data for track | CC-BY |
| Public Memorials & Monuments & Fountains  <https://data.gov.au/dataset/ds-melbourne-https%3A%2F%2Fdata.melbourne.vic.gov.au%2Fapi%2Fviews%2Fuqhf-q5h7/details?q=Melbourne%20city> | CSV | Static | Yearly | Location and data about activity | CC-BY |
| Social Indicators for City of Melbourne Residents 2020  <https://data.melbourne.vic.gov.au/People/Social-Indicators-for-City-of-Melbourne-Residents-/x4k3-uj4j/data> | CSV | Static | Weekly | Statistics by category of Physical activity | CC-BY |
| Current Weather  <https://openweathermap.org/> | API | Live | Live | Melbourne current weather | CC-BY |

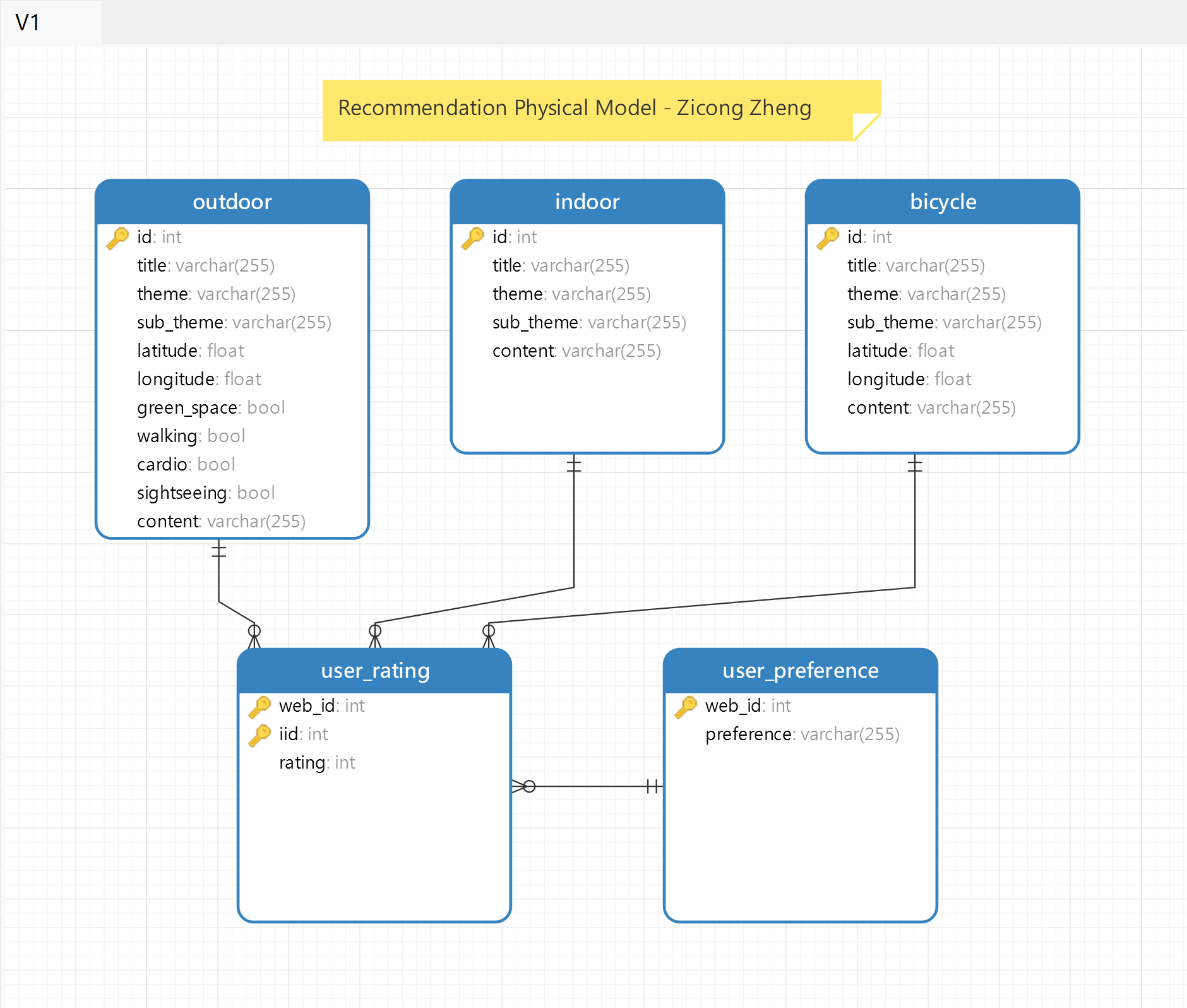
**3. Database**

| Service Provider | Amazon Web Services (AWS) |
| --- | --- |
| Database | MariaDB |



**4. Database Schema**

Database schema in use for Recommendation System (Physical ERD)



**5. Database Queries**

Database queries can be fired from a MySql interface or command line to create the database on a server. The steps to create a database server are mentioned in the Maintenance.docx.

**Queries to create Database for Sittofit website**

Connecting to MySql instance

* Creating database

CREATE DATABASE *recommendation*;

CREATE DATABASE *sittofit*;

* Creating tables

Select recommendation

CREATE TABLE user\_pref(

web\_id int,

pref varchar(50) ,

CONSTRAINT web PRIMARY KEY (web\_id))

CREATE TABLE user\_rating(

web\_id int,

iid int,

rating int,

CONSTRAINT pks PRIMARY KEY (web\_id, iid))

CREATE TABLE bicycle(

id int

title varchar(50),

theme varchar(50),

sub\_theme varchar(50),

latitude float,

longitude float,

content varchar(50),

CONSTRAINT pks PRIMARY KEY id)

CREATE TABLE indoor(

id int

title varchar(50),

theme varchar(50),

sub\_theme varchar(50),

content varchar(50),

CONSTRAINT pks PRIMARY KEY id)

CREATE TABLE outdoor(

id int

title varchar(50),

theme varchar(50),

sub\_theme varchar(50),

latitude float,

longitude float,

green\_space tinyint(1),

walking tinyint(1),

cardio tinyint(1),

sightseeing tinyint(1),

content varchar(50),

CONSTRAINT pks PRIMARY KEY web\_id)

**Inserting Open Dataset to database (Dashboard view)**

It is recommended to use database administration tools to import datasets to the database visually. MySQL Workbench and phpMyAdmin are two widely used tools that can perform the import and export tasks with csv file format.

[Import and export — phpMyAdmin 5.1.4 documentation](https://docs.phpmyadmin.net/en/latest/import_export.html)

[MySQL Workbench Manual :: 6.5.1 Table Data Export and Import Wizard](https://dev.mysql.com/doc/workbench/en/wb-admin-export-import-table.html)

**Inserting values (Used in Python)**

The queries will be fired from Python using PyMySql package, hence the query style is different.

* User Preference:

Inserting rating for a new item:

'''INSERT INTO user\_preference (web\_id, preference) values (%s, %s)'''

Inserting rating when rating for item already exists:

'''UPDATE user\_preference set web\_id = %s, preference = %s where (web\_id = %s)'''

Values that will replace %s (Refer to Python documentation for [string formatting](https://docs.python.org/3/library/string.html#format-examples))

For Insert - (df["web\_id"][0], df["preference"][0])

For Update - (df['web\_id'][0], df["preference"][0], df['web\_id'][0])

* User Rating:

Inserting rating for a new item:

'''INSERT INTO user\_rating (web\_id, iid, rating) values (%s, %s, %s)'''

Inserting rating when rating for item already exists:

'''UPDATE user\_rating set web\_id = %s, iid = %s, rating = %s where (web\_id = %s and iid = %s)'''

Values that will replace %s (Refer to Python documentation for [string formatting](https://docs.python.org/3/library/string.html#format-examples))

For Insert - (df["web\_id"][0], df["iid"][0], df['rating'][0])

For Update - (df['web\_id'][0], df["iid"][0], df["rating"][0], df['web\_id'][0], df["iid"][0])

**6. Data collection and Data Storage**

***6.1.1 Data Collection***

CSV - The data is directly accessible via exporting as CSV for our data requirements.

API - Live data such as weather is accessed in real time to get the Melbourne weather which will be processed directly on the server side (Python API).

1. **OPEN DATA**

| **Names** | **Physical Access** | **Frequency of Source Updates** | **Frequency of Iteration system updates** | **Granularity** | **Copyright/Licensing Details** |
| --- | --- | --- | --- | --- | --- |
| Outdoor artworks  <https://data.melbourne.vic.gov.au/People/Outdoor-artworks/ue3p-kdsu> | CSV | Not Provided (Latest by 2020) | Yearly | Precise location of activity | CC-BY |
| Parks and green spaces in Melbourne  <https://data.melbourne.vic.gov.au/People/Landmarks-and-places-of-interest-including-schools/j5vt-ppat/data> | CSV | Monthly | Yearly | Precise location of activity | CC-BY |
| Bicycle tracks in Melbourne  <https://data.melbourne.vic.gov.au/Transport/Bicycle-routes-including-informal-on-road-and-off-/24aw-nd3i> | CSV | Quarterly | Yearly | Coordinate data for track | CC-BY |
| Public Memorials & Monuments & Fountains  <https://data.gov.au/dataset/ds-melbourne-https%3A%2F%2Fdata.melbourne.vic.gov.au%2Fapi%2Fviews%2Fuqhf-q5h7/details?q=Melbourne%20city> | CSV | Static | Yearly | Location and data about activity | CC-BY |
| Current Weather  <https://openweathermap.org/> | API | Live | Live | Melbourne current weather | CC-BY |

1. **User Data from Interactive Recommendation feature**

The User will push preferences and ratings for the recommendation system which is collected as the user engages with the recommendation feature. The user data will be mapped to their UUID (Unique ID) created on their browser which will be active until the browser cache is removed.

| **Collection Stage** | **Collection Terms** |
| --- | --- |
| User Preferences | User preferences will be collected once per instance (browser data) |
| User Rating | User rating will be collected every time a user interacts with a recommendation card. |

Note: We do not collect personal user information or have implemented a “login” feature for this iteration.

***6.1.2 Data Storage***

* The Open data is stored on MariaDB server in the recommendation database as our project currently requires data in one of the features. The Open Data is exported to CSV and the undergone data wrangling is explained in the steps below.
* The weather API fetches the data and is processed instantaneously, therefore it is not required to be stored on a database server.
* The User data is piped to MariaDB database directly from the recommendation engine API.

**6.2 Data Processing**

The collected data is processed differently using Python and Jupyter notebook IDE. The two major steps are Data Exploration followed by Data Wrangling to prepare the data for our use cases.

The script for Data Wrangling is DataWrangling\_Script.ipynb.

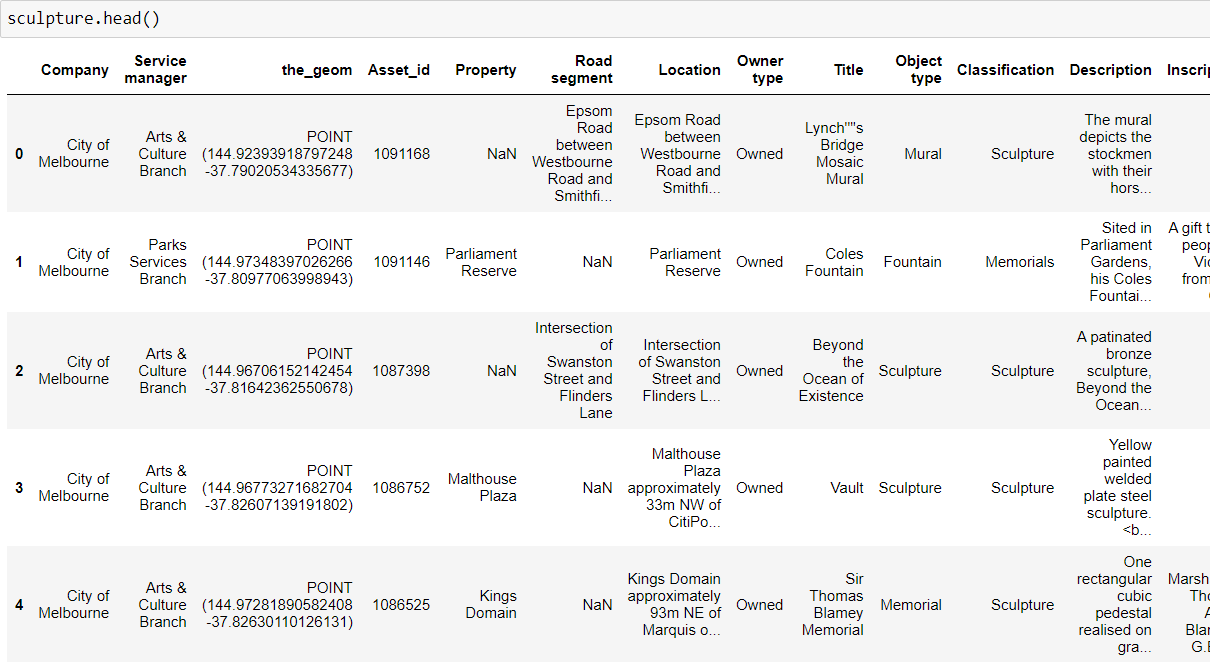
5.2.1 Data Exploration

The data exploration was done for the Open Data CSV files as these are susceptible to improper data formats which depends severely on the collection followed by the responsible organization. This will allow us to visualize the use cases possible with the data.

The exploration is follows a similar approach for all of the datasets:

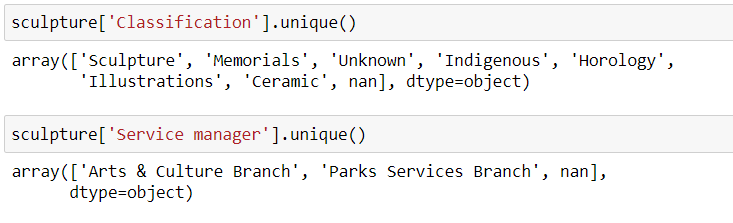
* Data Display

Firstly, we display the dataset to view the type of content in each column and if the dataset is fit for our use case. Using the .head() argument we can display the top 10 rows of the dataset to visualize the content in each column.



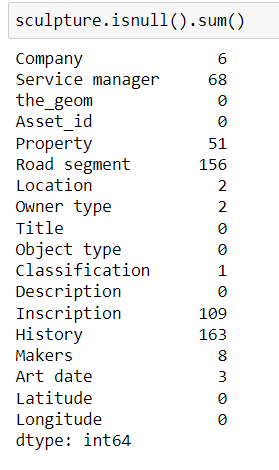
* Unique Values in columns

The categories in the columns are important to determine how they can be treated for our data modeling process. Using the .unique() argument in Python, we obtained the following result for one of the dataset.



* Null values in each column

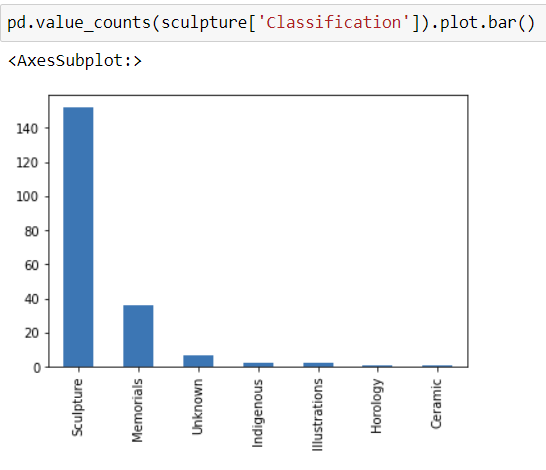
The nulls can significantly reduce the number of rows that we can use if the null/nan value exists in the required column. Using the .null.sum() gives us the number of nulls in each column and we have to determine if the number will be resulting if there is insufficient data.



The columns such as Road Segment, Service manager, Inscription and History are not useful for our use case and can be disregarded. Whereas, the useful columns such as location (lat and lon), Title, etc do not have many null values.

* Category Distribution

The distribution is an important part for proper modeling as it will determine if there are enough values in each category for our model. Using barplot in python, we can visualize the value counts of each unique value in the column.



In the above example, the value count for Ceramic and Horology are very less to be considered as a feature for our model.

***6.2.2 Data Wrangling***

The data wrangling follows two approaches, data cleaning and feature engineering. This is the crucial stage which will set the dataset for our use cases.

*6.2.2.1 Data Cleaning*

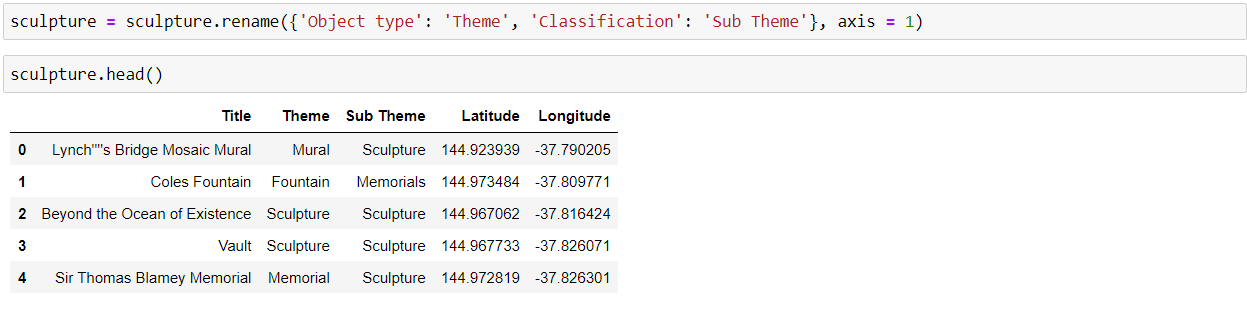
* Drop columns

We will drop the unnecessary data that we do not require fit for our use. As we observed, there are columns with excessive nulls and columns which are not required to us in the Section 5.2.1 Data Display. Using .drop in python we can drop the array of columns we want to remove from out dataset.



* Rename columns for standardization

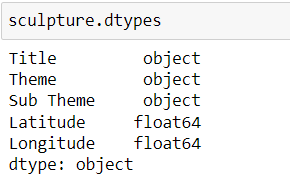
The datasets have distinct column names that we explored in the exploration phase and we have to make them common for the model to understand. Using .rename on the DataFrame (python table structure), we can rename the columns to maintain similar names across all the datasets.



The new column names will remain the same throughout all the datasets.

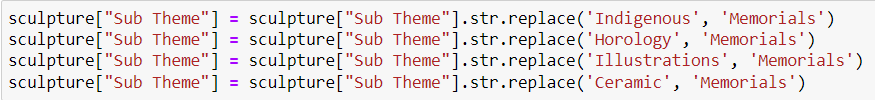
* Data type change

The datatype specifies how the value in a column will be stored. As we will be pushing the data on the database, it has to follow our desired requirements. Using .dtype on dataframe, we can check the format of each column and change it accordingly.



* Combine insufficient categories

As we observed in Section 5.2.1 Category Distribution, there is insufficient data in few categories. We can combine them to a unique category or merge with another category to improve model understanding. However, it is crucial to merge categories into fitting group, else results can be false in the modeling stage. Using .replace, we can replace the category with our desired category.



* Remove unnecessary category in a column

There are categories in the datasets which are irrelevant for a user which can be removed. We will be removing categories if they match our list of unwanted categories. For example, ‘Unknown’ category in the “Sub Theme” column does not represent a quality category and we will remove such ones.

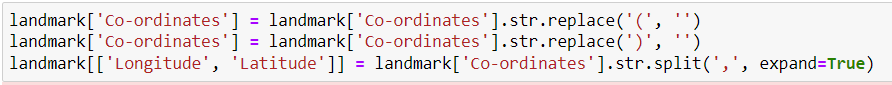


*6.2.2.2 Feature Engineering*

* Regex to split column

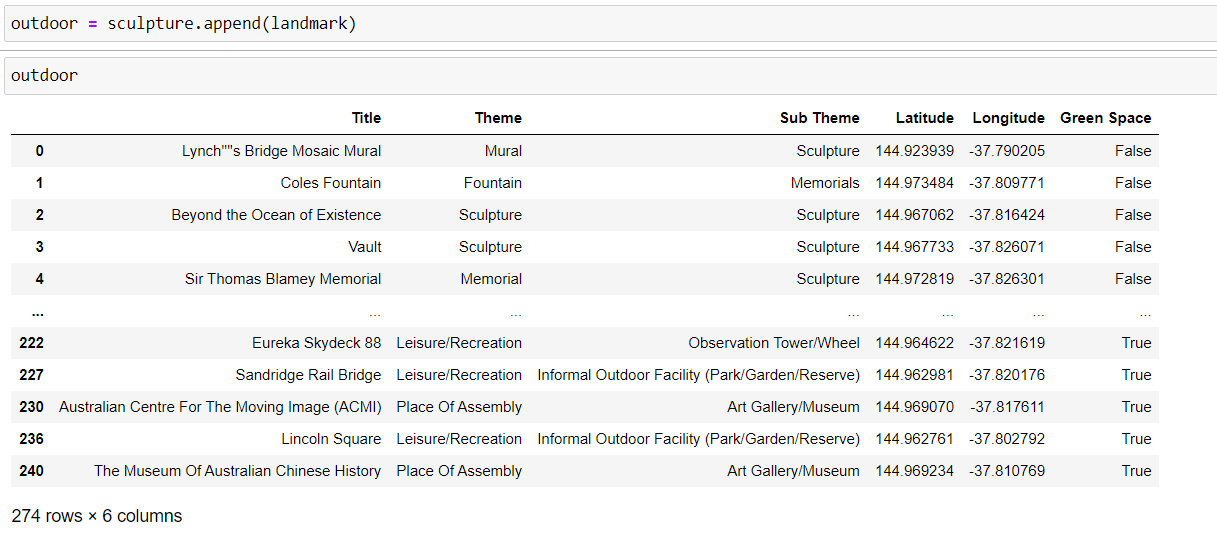
There are instances where columns are combined in the dataset whereas in another dataset the columns are treated separately. Therefore, we can either split or merge the columns wherein we have used split approach to keep it handy while pushing the data on the database and using it on the front-end.

Using .replace() on a column, we can split the column into different columns on a defined principle.



* Merging two datasets

Modeling requires datasets to be combined as it can not intake two separate tables. Therefore, we will merge the similar datasets which have undergone the standardization process as explained in the above steps. For example, we have to merge sculpture dataset with landmark data which will have a common name as outdoor.



* Feature addition

There is a necessity to add more features to create a better understanding for the model to distinguish the data. Therefore, we will add more features based on some keywords which will help model and will be used for filtering rows based on user preferences.

For instance, we create new columns in outdoor dataset based on several categories.



* Exporting wrangled dataset

The changes that we do in wrangling are required to be converted into new files to store the effects.

Using .export to csv, we can create a new file.



**6.3 Data Modelling**

The data modelling will allow us to produce recommendations based on user input by applying a combination of Machine Learning Algorithm and Advanced Filtering.

***6.3.1 Machine Learning Algorithm***

* Naive Bayes

Naive Bayes is a Machine Learning algorithm that models the categorical input data and produces categorical output. The user input is the rating for different cards produced on the display that is binary. Using these ratings, the model will extract more items similar to the highly rated items.



***6.3.2 Advanced Filtering***

* Reducing rows based on User Input

We will take the user preferences and eliminate data irrelevant to the user.

* Distance based filtering

The data uses location based outputs to the user for which we have employed Haversine distance calculation method to find nearby activities to the location liked by the user.

**7**. **Data Risk Management**

We are not collecting sensitive user data from the user at this stage. The open data will be updated frequently (weekly) to maintain integrity of the data in case of breach.

Possible Data Risks and strategies to mitigate:

* The data collected in Feature - Risk Meter will be stored on the AWS MariaDB Server which has enabled security for the data.
* The data collected in Feature - Recommendation System will be stored on AWS MariaDB server which has enabled security for the data. However, the data does not hold any sensitive information about the user.

**8. Data Preservation and Sharing**

**8.1 Data Preservation**

After the project development is completed, the open data stored on our server will be locked to maintain the integrity. The user data that will be collected and stored directly to the hosted AWS MariaDB server.

**8.2 Data Sharing**

There is no sensitive data that will be shared with any other service.

**9. Data Backup and Archiving**

**9.1 Data Backup**

At this stage, AWS Server sets up a backup model for our data as a standard policy.

**9.2 Data Archiving**

At this stage, we have not regulated data archiving strategy as we are not determined to store user data. The user data will be stored from Iteration 3 and from then will be setting the archiving policy.

* Datasets - As we update our open datasets, we will archive our current datasets to a container on AWS under the Australian Data Retention Policy.
* User Data - As we collect user data,we will archive the dataset to a container on AWS under the Australian Data Retention Policy. The user will be provided consent for the collected data and data retention requirements of our product.

**10. Data Ownership**

The data is owned by Sittofit and The It Guyz team, who will be responsible for maintaining the integrity of the data.