WQD7005 - Data Mining

FINAL EXAM

Matrix Number : 17043640

Name : Gunasegarran Magadevan

1. You are required to make a user-agent that will crawl the WWW (your familiar domain) to produce dataset of a particular website.
   - the web site can be as simple as a list of webpages and what other pages they link to
   - the output does not need to be in XHTML (or HTML) form
     a multi-stage approach (e.g. produce the xhtml or html in csv format)

(10 marks)

In [1]:
```
# Import packages
from bs4 import BeautifulSoup
import urllib.request
import pandas as pd
import numpy as np
import csv
from pathlib import Path

url = 'https://files.osf.io/v1/resources/bvn42/providers/osfstorage
req = urllib.request.Request(url1, data=None, headers={'User-Agent'

soup = BeautifulSoup(urllib.request.urlopen(req).read(),"lxml")

#extract data
rows = soup.find('table',{'class': 'genTbl closedTbl historicalTbl'
data = []
for row in rows:
    cols = row.find_all('td')
    cols = [ele.text.strip(' ') for ele in cols]
    data.append([ele for ele in cols if ele])
colnames = soup.find('table',{'class': 'genTbl closedTbl historical
col_names = []
for col in colnames:
    cols = col.find_all('th')
    cols = [ele.text.strip() for ele in cols]
    col_names.append(cols)
col_names = col_names[0]

#Write data to files
df1 = pd.DataFrame(data,columns = col_names)

# Writing the DataFrame: df to CSV file
df.to_csv('HouseData.csv')
```

In [2]:
```
# Displaying top 5 DataFrame: df
df.head()
```

Out[2]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floo
0	7129300520	20141013T000000	221900.0	3	1.00	1180	5650	1
1	6414100192	20141209T000000	538000.0	3	2.25	2570	7242	2
2	5631500400	20150225T000000	180000.0	2	1.00	770	10000	1
3	2487200875	20141209T000000	604000.0	4	3.00	1960	5000	1
4	1954400510	20150218T000000	510000.0	3	2.00	1680	8080	1

5 rows × 21 columns

2. Draw snowflake schema diagram for the above dataset. Justify your attributes to be selected in the respective dimensions.

(10 marks)

1. **Snowflake Schema** is a logical arrangement of tables in a multidimensional database such that the **Entity Relationship Table** resembles a snowflake shape.
2. **Snowflake Schema** is an extension of a **Star Schema**, and it adds additional dimensions.
3. The dimension tables are **normalized** which splits data into additional tables.

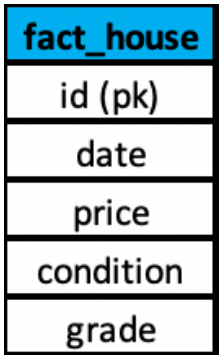
In [3]:
```
# Displaying column name from DataFrame: df
print(df.columns.tolist())
```

```
['id', 'date', 'price', 'bedrooms', 'bathrooms', 'sqft_living', 's
qft_lot', 'floors', 'waterfront', 'view', 'condition', 'grade', 's
qft_above', 'sqft_basement', 'yr_built', 'yr_renovated', 'zipcode'
, 'lat', 'long', 'sqft_living15', 'sqft_lot15']
```

In [4]:
```
# Table normalize to fact_house
fact_house = df[['id', 'date','price','condition','grade']]
fact_house.head(2)
```

Out[4]:

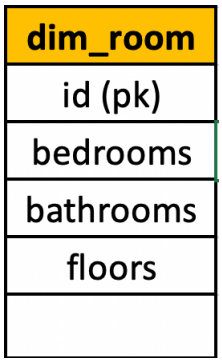
	id	date	price	condition	grade
0	7129300520	20141013T000000	221900.0	3	7
1	6414100192	20141209T000000	538000.0	3	7



In [5]:
```
# Table normalize to dim_room
dim_room = df[['id','bedrooms','bathrooms','floors']]
dim_room.head(2)
```

Out[5]:

	id	bedrooms	bathrooms	floors
0	7129300520	3	1.00	1.0
1	6414100192	3	2.25	2.0



In [6]:
```
# Table normalize to dim_sqft
dim_sqft = df[['id', 'sqft_living','sqft_lot','sqft_above','sqft_ba
dim_sqft.head(2)
```

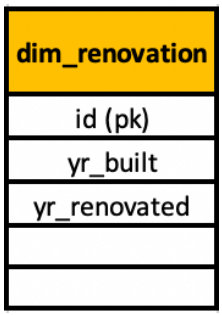
Out[6]:

	id	sqft_living	sqft_lot	sqft_above	sqft_basement	sqft_living15	sqft_lot15
0	7129300520	1180	5650	1180	0	1340	5650
1	6414100192	2570	7242	2170	400	1690	7639

In [7]:
```
# Table normalize to dim_renovation
dim_renovation = df[['id','yr_built','yr_renovated']]
dim_renovation.head(2)
```

Out[7]:

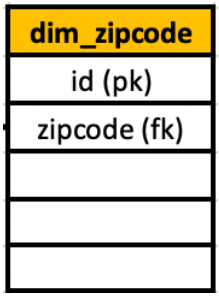
	id	yr_built	yr_renovated
0	7129300520	1955	0
1	6414100192	1951	1991



In [8]:
```
# Table normalize to dim_zipcode
dim_zipcode = df[['id','zipcode','lat','long']]
dim_zipcode.head(2)
```

Out[8]:

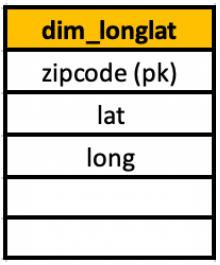
	id	zipcode	lat	long
0	7129300520	98178	47.5112	-122.257
1	6414100192	98125	47.7210	-122.319



In [9]:
```
# Table normalize to dim_longlat
dim_longlat = df[['zipcode','lat','long']]
dim_longlat.head(2)
```

Out[9]:

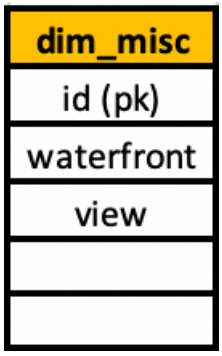
	zipcode	lat	long
0	98178	47.5112	-122.257
1	98125	47.7210	-122.319



In [10]:
```
# Table normalize to dim_misc
dim_misc = df[['id','waterfront','view']]
dim_misc.head(2)
```

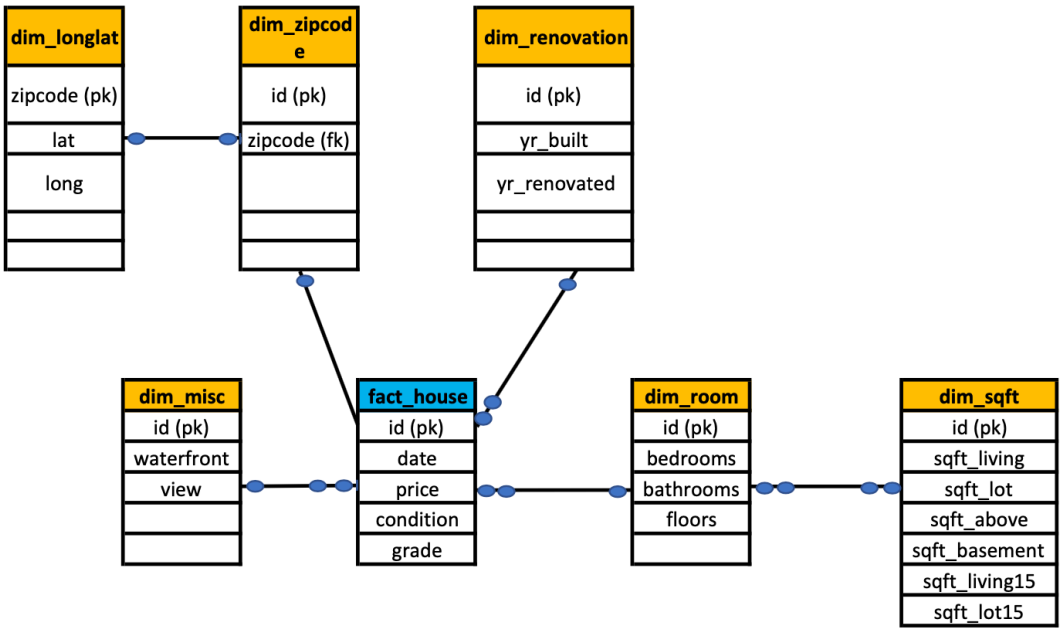
Out[10]:

	id	waterfront	view
0	7129300520	0	0
1	6414100192	0	0



Snowlflakes Schema House Data

Note: The `pk` represent `Primary Key` , while `fk` represent `Foreign Key`



In [ ]: