In [1]:

```
import pandas as pd
```

In [49]:

```
import warnings
warnings.simplefilter("ignore")
```

In [51]:

```
hotel_data = pd.read_excel("Hotel_data.xlsx")
scrape_data = pd.read_excel("scrape_data.xlsx")
```

Noticing that the hotel_data file is quite large and will be tedious to work with for the remainder of this analysis, I have gone into SQL to perform the necessary joins and filters that will allow me to continue analysing within this notebook.

In [52]:

```
hotels_unmatched = pd.read_csv("hotels_unmatched.csv")
webrez_hotels = pd.read_csv("scrape_data_v2.csv")
```

Hotel matching statistics

In [53]:

```
hotels_matched = len(webrez_hotels['Building Name'])
hotels_matched_unique = webrez_hotels['Building Name'].nunique()
hotels_database = hotel_data['Building Name'].nunique()
```

In [54]:

```
pct_matched = 100*hotels_matched_unique/hotels_database
print(f"The total number of hotels matched to the database is: {hotels_matched}")
print(f"The percentage of scraped hotels matched to the database is: {pct_matched:.2f}%, however, due to limitations on Google's side this is not a comprehensive list of WebRezPro customers.")
```

```
The total number of hotels matched to the database is: 102
The percentage of scraped hotels matched to the database is: 0.06%, however, due to limitations on Google's side this is not a comprehensive list of WebRezPro customers.
```

In [55]:

```
results_scraped = 300
pct_scraped = 100*hotels_matched/results_scraped
pct_scraped_unique = 100*hotels_matched_unique/results_scraped

print(f"The percentage of total hotels matched from the initial web scrape is: {pct_scraped:.2f}% and unique hotels is {pct_scraped_unique:.2f}%")
```

```
The percentage of total hotels matched from the initial web scrape is: 34.00% and unique hotels is 24.67%
```

In [56]:

```
hotels_unmatched_size = len(hotels_unmatched)
hotels_unmatched['title'].head(10)
```

Out[56]:

```
0    Hallmark Resort & Spa Cannon Beach - WebRezPro...
1           MTN House by Basecamp - WebRezPro: Sign In
2                                Booking Request Form
3                            Northridge Inn & Resort
4                                Diamond Mills Hotel
```

```
4                                  Diamond Mills Hotel
5                                The The Jack London Lodge
6                               Palms Hotel Fire Island
7                                     The Caboose Motel
8                                      Creekside Resort
9                                      Stanley's Resort
Name: title, dtype: object
```

In [57]:

```
print(f"The above is an example of some of the hotels that couldn't be matched to the dat
abase. In total there were {hotels_unmatched_size} unmatched rows.")
```

```
The above is an example of some of the hotels that couldn't be matched to the database. Ir
total there were 194 unmatched rows.
```

Understanding the data available from the web scrape

The first 3 variables come straight from the Google search and are straightforward:

- **Link e.g. https://secure.webrez.com/hotel/3574/**
- **Title e.g. MTN House by Basecamp - WebRezPro: Sign In**
- **Description i.e. The short, 2-3 sentence description visible on the Google search page**

The next 3 variables are taken from the individual page of each search result

- **Details_1 & Details_2**
  - **If we were to click on the link, this is the information below the hotel name and is most usually an address and contact details**
- **Features & Amenities**
  - **If the hotel on WebRezPro lists of their features and amenities available in their rooms, they will appear**

What are we working with?

In [58]:

```
## Understanding the different columns we have and how many missing values they contain
na_columns = webrez_hotels.isna().sum()
print(na_columns)
```

```
link                      0
title                     0
description               0
details_1                 0
details_2                 0
features_amenities        0
Row Count                 0
Building Name             0
Property ID               0
Salesforce Account Id   102
Parent Company          102
Brand                   102
Territory                 0
Country                   0
City                      0
ZIP Code                 10
Address                   0
Scale                     0
Class                     0
Hotel Location Type       0
Nr. of Rooms              0
Secondary Type            1
Continent                10
Subcontinent             10
Market                    1
Submarket                 0
Submarket Cluster         0
Region                   99
```

```
State                    13
County                   14
Stories                  14
Tenancy                  91
Mtg Rooms                90
Total Mtg Space          76
Parking Spaces           64
Parking Ratio            66
Operation Status          0
Operation Type            0
Parking Spaces/Room      64
Primary Corridors        14
Recorded Owner           56
True Owner               41
Hotel Operator           72
Hotel Grade              93
Green Rating            102
Star Rating               0
Status                    0
FIRM Id                  56
Year Built               15
Year Renov               91
Zoning                   31
Type                      0
dtype: int64
```

In [59]:

```
## Remove unnecessary columns and columns with many missing values
columns_to_remove = ['Row Count', 'Property ID', 'Salesforce Account Id', 'ZIP Code', 'Address', 'County',
                     'Tenancy', 'Operation Status', 'Primary Corridors', 'Status', 'FIRM Id', 'Year Renov', 'Zoning',
                     'Type', 'Parent Company', 'Brand', 'Region', 'Stories', 'Mtg Rooms', 'Total Mtg Space',
                     'Parking Spaces/Room', 'Recorded Owner', 'True Owner', 'Hotel Operator', 'Green Rating',
                     'Hotel Grade', 'Parking Spaces', 'Parking Ratio']
webrez_hotels = webrez_hotels.drop(columns = columns_to_remove)
```

Geography Analysis

In [60]:

```
## Look into the location of these hotels

webrez_continent = webrez_hotels['Continent'].value_counts()
webrez_continent = pd.DataFrame(webrez_continent)
print(webrez_continent)
```

```
          Continent
Americas         85
Europe            7
```

In [61]:

```
webrez_hotels[['Building Name', 'Country', 'City']].loc[webrez_hotels['Continent'] != 'Americas']

# Not confident on Astoria Hotel and Hotel Metro are all under the same ownership,
# in which case they are not necessarily customers
```

Out[61]:

	Building Name	Country	City
0	Astoria Hotel	Indonesia	Bandar Lampung
1	Siesta Suites Hotel	Mexico	Cabo San Lucas
2	Astoria Hotel	Germany	Göttingen

	Building Name	Country	City
3	Astoria Hotel	Germany	Ratingen
4	Hotel Metro	Hungary	Budapest
5	Hotel Metro	Poland	Klodzko
6	Hotel Metro	Poland	Warszawa
7	Astoria Hotel	Ukraine	Lviv
8	Astoria Hotel	Greece	Thessaloníki
9	Astoria Hotel	Italy	Rapallo
10	Astoria Hotel	Portugal	Coimbra
11	Astoria Hotel	United Arab Emirates	Dubai
12	Astoria Hotel	Norway	Kristiansand
13	Alpine Village	New Zealand	Queenstown
14	Astoria Hotel	United Kingdom	Blackpool
15	Astoria Hotel	United Kingdom	Blackpool
16	Dryburgh Abbey Hotel	United Kingdom	Melrose

In [62]:

```
# Remove Astoria Hotel and Hotel Metro
webrez_hotels = webrez_hotels[(webrez_hotels['title'] != 'Astoria Hotel') &
                              (webrez_hotels['title'] != 'Hotel Metro')]

webrez_hotels[['Building Name', 'Country', 'City']].loc[(webrez_hotels['Continent'] != '
Americas')]
```

Out[62]:

	Building Name	Country	City
1	Siesta Suites Hotel	Mexico	Cabo San Lucas
13	Alpine Village	New Zealand	Queenstown
16	Dryburgh Abbey Hotel	United Kingdom	Melrose

In [63]:

```
webrez_location = webrez_hotels[['Building Name', 'Country','State', 'City']].loc[webrez
_hotels['Continent'] == 'Americas']
webrez_states = webrez_location[['Country', 'State']].value_counts().head(10)
webrez_states = pd.DataFrame(webrez_states)
print(webrez_states)
```

```
                     0
Country       State
United States CA     9
Canada        BC     9
              AB     7
              ON     5
United States WA     5
              CO     5
              NY     4
              FL     3
              ID     3
              OR     3
```

West coast North America - California and British Columbia round out the top states.
Strong presence in Canada, and a large spread across the rest of America.
Only have 3 hotels found from outside North America, found in Mexico, New Zealand and the UK.

Note: From here on out we have removed 'Astoria Hotel' and 'Hotel Metro' as we can not be sure they are WebRezPro customers

Location Analysis

Location Analysis

In [64]:

```
webrez_location = webrez_hotels['Hotel Location Type'].value_counts()
webrez_location = pd.DataFrame(webrez_location)
print(webrez_location)
```

```
                  Hotel Location Type
Small Metro/Town                   47
Resort                             15
Suburban                           12
Urban                               8
Interstate                          4
```

Here we see a heavy distribution towards hotels in Small Metro/Town areas.

Scale Analysis

In [65]:

```
webrez_scale = webrez_hotels['Scale'].value_counts()
webrez_scale = pd.DataFrame(webrez_scale)
print(webrez_scale)
```

```
             Scale
Independent     86
```

100% of these hotels are Independently owned.

Class Analysis

In [66]:

```
class_counts = webrez_hotels['Class'].value_counts()

class_table = pd.crosstab(index=webrez_hotels['Class'], columns='Count')
class_table['Percentage'] = round(class_table['Count'] / class_table['Count'].sum() * 10
0,2)
class_table = class_table.sort_values(by='Percentage', ascending=False)
# Display the table
print(class_table)
```

```
col_0           Count  Percentage
Class
Economy            24       27.91
Midscale           14       16.28
Upper Upscale      14       16.28
Upscale            14       16.28
Luxury             13       15.12
Upper Midscale      7        8.14
```

A pretty even spread of hotels. This tells us WebRezPro isn't targeting one area of the market in terms of class.

In [67]:

```
# Look at Star Rating

# Assuming 'webrez_hotels' is your DataFrame
star_rating_counts = webrez_hotels['Star Rating'].value_counts()

# Create a DataFrame from the value counts
star_rating_table = pd.DataFrame({'Star Rating': star_rating_counts.index, 'Count': star
_rating_counts.values})

# Sort the DataFrame by 'Star Rating' in descending order
star_rating_table = star_rating_table.sort_values(by='Star Rating', ascending=False)
```

```
# Display the sorted table
print(star_rating_table)
```

```
   Star Rating  Count
3            5      5
2            4     22
0            3     31
1            2     28
```

Mostly 2-3 star hotels which is in line with their large Economy and Midscale customerbase.

Size Analysis

In [68]:

```
# Start off by looking at the number of rooms in the hotels

# create bins
bin_edges = [0, 20, 40, 60, 80, 100, 150, float('inf')]
bin_labels = ['0-20', '21-40', '41-60', '61-80', '81-100', '101-150', '150+']

# Create a new column 'Room Size Bucket' in the DataFrame based on bin edges and labels
webrez_hotels['Room Size Bucket'] = pd.cut(webrez_hotels['Nr. of Rooms'], bins=bin_edges,
labels=bin_labels, right=False)

# Count the number of hotels in each bucket
room_size_counts = webrez_hotels['Room Size Bucket'].value_counts()

# Create a DataFrame from the value counts
room_size_table = pd.DataFrame({'Number of Rooms': room_size_counts.index, 'Count': room
_size_counts.values})

# Display the count of hotels in each bucket
print(room_size_table)
```

```
  Number of Rooms  Count
0           21-40     30
1            0-20     27
2           41-60     18
3           61-80      7
4            150+      3
5          81-100      1
6         101-150      0
```

In [69]:

```
webrez_hotels[['Building Name', 'City', 'Country']].loc[webrez_hotels['Nr. of Rooms'] >=
150]
```

Out[69]:

	Building Name	City	Country
21	**Canmore Inn & Suites**	**Canmore**	**Canada**
48	**The Grove Hotel**	**Boise**	**United States**
79	**Hallmark Resort Newport**	**Newport**	**United States**

The majority of WebRezPro's customers have between 0 and 60 rooms in their hotels.

The names of the 3 hotels with 150+ room:

- **Canmore Inn & Suits in Canmore, Canada**
- **The Grove Hotel in Boise, US**
- **Hallmark Resort Newport in Newport, US**

Year Built Analysis

In [70]:

```
# create bins
bin_edges = [0, 1900, 1950, 1975, 2000, 2010, 2020, float('inf')]
bin_labels = ['0-1900', '1901-1950', '1951-1975', '1976-2000', '2001-2010', '2011-2020',
'2020+']

# Create a new column 'Room Size Bucket' in the DataFrame based on bin edges and labels
webrez_hotels.loc[:, 'Year Built Bucket'] = pd.cut(webrez_hotels['Year Built'], bins=bin_
edges, labels=bin_labels, right=False)

# Count the number of hotels in each bucket
year_built_counts = webrez_hotels['Year Built Bucket'].value_counts()

# Create a DataFrame from the value counts
year_built_table = pd.DataFrame({'Year': year_built_counts.index, 'Count': year_built_co
unts.values})

# Display the count of hotels in each bucket
print(year_built_table)
```

```
        Year  Count
0  1976-2000     27
1  1901-1950     17
2  1951-1975     14
3     0-1900      9
4  2001-2010      5
5  2011-2020      5
6      2020+      5
```

A mix of old and modern buildings.

Features & Amenities Analysis

In [71]:

```
# This column from the web scrape comes with every feature/amenity in one column so we ne
ed to separate each of them

df = webrez_hotels['features_amenities'].loc[webrez_hotels['features_amenities']!='[]']
df = pd.DataFrame(df)

# Define a regular expression pattern to match items between quotes
pattern = r"'(.*?)'"

# Use str.extractall to extract all matches and reset_index to make it a DataFrame
extracted_items = df['features_amenities'].str.extractall(pattern).reset_index(level=1,
drop=True)

# Rename the columns for clarity
extracted_items.columns = ['extracted_items']

# Join the extracted items back to the original DataFrame
df = df.join(extracted_items)

# Display the result
df['extracted_items'].value_counts()
```

Out[71]:

```
WiFi Internet                                 7
Mini-fridge                                   7
Balcony                                       6
Microwave                                     6
Pets not allowed                              6
Fireplace                                     5
Barbeque                                      5
Kitchenette                                   4
Fridge                                        4
Water view                                    4
```

```
Accessible room                                  3
Garden view                                      3
Full bath (toilet, sink, shower & tub)           3
Pet friendly                                     3
Blue Bay Cottages                                2
Private pool                                     2
Ocean view                                       2
Stove top                                        2
Blue Bay Motel                                   2
Hot tub in unit                                  2
Full kitchen                                     2
Mountain view                                    1
Ocean front                                      1
Access to hot tub (private or shared)            1
Oven                                             1
Access to a pool (either private or shared)      1
Water front                                      1
3/4 bath (toilet, sink & shower)                 1
Lagoon view                                      1
Park view                                        1
Name: extracted_items, dtype: int64
```

Unfortunately this is not a large enough sample size of all the features and amenities that WebRezPro customers are offering to their guests so there is not much valuable information to gain at this point.

Conclusion

After scraping Google to get an understanding of the hotels WebRezPro is working with, this analysis attempted to look at different variables such as class, location, size in an effort to understand as much as possible ablout WebRezPro's customer base.
Key discoveries include:

- **Almost entirely made up of hotels in North America with the highest concentration being on the West coast in California and British Columbia.**
- **A mixed group when looking at Class with an almost even spread ranging from economy to luxury.**
- **Majority of hotels have less than 60 rooms in total.**

Limitations:

- **It was not possible to get a comprehensive list of all WebRezPro's customers, for 2 reasons:**
  - **Google was not showing more than 300 results on the search despite there being more results out there.**
  - **While we were able to get 300 rows of data, this only translated into matching 102 hotels.**
    - **Due to incosistencies in how the HTML code was written for each hotel resulting in cases where the hotel name is instead appearing as "Availability today" or the address of the hotel.**
    - **Also potentially due to a mismatch between the naming from the hotel database.**