SQL PROJECT:

Property Listing Analysis

1. Find out sum rental price of each the room type available in the database

```
SELECT Room_Type, SUM (price) AS SUM FROM listing_venice_df GROUP BY
room_type;
```

2. Analyze the difference between property vacancy rate between the months in the year 2022.

```
SELECT MONTH(DATE) AS MONTH, YEAR(DATE) AS YEAR, COUNT(available) AS
COUNT FROM df_venice_availability
WHERE YEAR(DATE) = 2022 and available=1
GROUP BY MONTH(DATE),YEAR(DATE)
ORDER BY COUNT(available) DESC;
```

3. Find The Cummulative Sum Of Price By Property_Type

```
SELECT ID,PRICE, property_type, SUM(PRICE) OVER(PARTITION BY
PROPERTY_TYPE ORDER BY PRICE) AS Cumm_Price
FROM listing_venice_df;
```

4. Find out top 5 less selling/rented property types available.

```
SELECT top 5 a.property_type,COUNT(d.available) AS
number_of_availabilty FROM
df_venice_availability AS d
INNER JOIN
listing_venice_df AS a
ON d.listing_id= a.id
WHERE d.available =0 and YEAR(d.DATE)=2022
GROUP BY a.property_type
ORDER BY number_of_availabilty ASC;
```

5. Finding out month in year in which mostlty property type are available and categories them on the basis of minimum availability

```
SELECT a.property_type, COUNT(d.available) AS
number_of_availabilty, MONTH(d.DATE)
AS MONTH, AVG(a.price) AS AVG_price FROM df_venice_availability AS d INNER
JOIN
listing_venice_df AS a
ON d.listing_id= a.id
WHERE d.available =1 and MONTH(d.DATE)=3
GROUP BY a.property_type,MONTH(d.DATE)
ORDER BY COUNT(d.available) ASC;
```

6. Try to search if there is any relation between hosts response time and property being rented

```
SELECT t1.host_response_time , COUNT(t3.available) AS not_available
FROM host_venice_df AS t1 INNER JOIN listing_venice_df AS t2 ON t1.host_id =
t2.host_id
INNER JOIN df_venice_availability AS t3 ON t2.id = t3.listing_id
WHERE t3.available = 0 and t1.host_response_time IS NOT NULL GROUP BY
t1.host_response_time;
```

7. Categorize property ratings into different types based on the score of ratings into average, good, best. Find total count of properties in each category and find out the average rating of each segment.

```
SELECT
CASE
WHEN review_scores_rating < 4.0 THEN 'average'
WHEN review_scores_rating >= 4.7 THEN 'premium'
WHEN review_scores_rating >= 4.0 THEN 'good'
ELSE 'not_rated'
END AS rating_category , COUNT(*) AS total_property ,
AVG(review_scores_rating)
```

```
AS AVG_rating
FROM listing_venice_df
GROUP BY
CASE
WHEN review_scores_rating < 4.0 THEN 'average'
WHEN review_scores_rating >= 4.7 THEN 'premium'
WHEN review_scores_rating >= 4.0 THEN 'good'
ELSE 'not_rated'
END
ORDER BY total_property DESC
```

8. Find out the average price of properties by their bedroom count along with their neighbourhood area.

```
SELECT neighbourhood_cleansed , bedrooms , AVG(price) as Avg_Price FROM
listing_venice_df
GROUP BY neighbourhood_cleansed , bedrooms
ORDER BY bedrooms;
```

9. Find out no. of property rented as per their area as well as their average price and order them as from higher priced property to lower price properties.

```
SELECT neighbourhood_cleansed , AVG(t1.price) AS AVG_price,
COUNT(available) AS Rented_Property
FROM listing_venice_df t1 INNER JOIN df_venice_availability t2 ON t1.id =
t2.listing_id
WHERE available = 0
GROUP BY neighbourhood_cleansed
ORDER BY rented_property DESC;
```

10. Find out top 5 best selling/rented property types available.

```
SELECT top 5 a.property_type,COUNT(d.available) AS
number_of_availabilty FROM
df_venice_availability AS d
INNER JOIN
listing_venice_df AS a
```

```
ON d.listing_id= a.id
WHERE d.available =0 and YEAR(d.DATE)=2022
GROUP BY a.property_type
ORDER BY number_of_availabilty DESC;
```