***SQL QUERIES***

**Queries for Table1:**

Retrieve properties with balconies, sorted by the number of bedrooms in descending order:

SELECT \*

FROM Table1

WHERE Balcony > 0

ORDER BY Beds DESC;

 Find the top 5 cities with the highest average number of bedrooms per property:

SELECT City, AVG(Beds) AS Avg\_Beds

FROM Table1

GROUP BY City

ORDER BY Avg\_Beds DESC

LIMIT 5;

 Count the number of properties in each city:

SELECT City, COUNT(\*) AS Num\_Properties

FROM Table1

GROUP BY City;

 Retrieve all properties with at least 3 bedrooms and 2 bathrooms:

SELECT \*

FROM Table1

WHERE Beds >= 3 AND Bathrooms >= 2;

 Find properties in a specific state with a certain landmark (e.g., state = 'California', landmark = 'Golden Gate Bridge'):

SELECT \*

FROM Table1

WHERE State\_code = 'CA' AND Landmark = 'Golden Gate Bridge';

**Queries for Table2:**

 Calculate the average price per square foot for properties built before 2010:

SELECT AVG(Price\_per\_Area) AS Avg\_Price\_per\_Area

FROM Table2

WHERE Year\_built < 2010;

 Find the total number of properties on each floor:

SELECT Floor, COUNT(\*) AS Num\_Properties

FROM Table2

GROUP BY Floor;

 Retrieve properties with a carpet area greater than 1000 square feet and a status of 'Under Construction':

SELECT \* FROM Table2

WHERE Carpet\_area > 1000 AND Construction\_stats = 'Under Construction';

 Calculate the average price per square foot for each transaction type:

SELECT Transaction\_type, AVG(Price\_per\_Area) AS Avg\_Price\_per\_Area

FROM Table2

GROUP BY Transaction\_type;

 Find the properties with the highest price per square foot, sorted in descending order

SELECT \*

FROM Table2

ORDER BY Price\_per\_Area DESC;

#### Queries for Table3:

Retrieve all properties with a furnished status of 'Fully Furnished' and a facing direction of 'East':

SELECT \*

FROM Table3

WHERE Furnished\_status = 'Fully Furnished' AND Facing = 'East';

 Calculate the average booking amount for properties with and without car parking:

SELECT Car\_park, AVG(Booking\_amount) AS Avg\_Booking\_Amount

FROM Table3

GROUP BY Car\_park;

Find the total price of properties with different types of ownership:

SELECT Ownership\_type, SUM(Price) AS Total\_Price

FROM Table3

GROUP BY Ownership\_type;

 Retrieve properties with a booking amount greater than 50000 and a furnished status of 'Semi Furnished':

SELECT \*

FROM Table3

WHERE Booking\_amount > 50000 AND Furnished\_status = 'Semi Furnished';

 Find the property with the highest booking amount:

SELECT \*

FROM Table3

ORDER BY Booking\_amount DESC

LIMIT 1;

#### Join Queries using all 3 tables:

1. Retrieve properties from Table1 that have a higher price per square foot than the average price per square foot in Table2:

SELECT t1.\*

FROM Table1 t1

INNER JOIN (

SELECT AVG(Price\_per\_Area) AS Avg\_Price\_per\_Area

FROM Table2

) t2\_avg ON t1.Price\_per\_Area > t2\_avg.Avg\_Price\_per\_Area;

 Find the properties in Table1 that are located in cities where the average price per square foot in Table2 is higher than the overall average price per square foot:

SELECT t1.\*

FROM Table1 t1

INNER JOIN (

SELECT City, AVG(Price\_per\_Area) AS Avg\_Price\_per\_Area

FROM Table2

GROUP BY City

) t2\_avg ON t1.City = t2\_avg.City

WHERE t2\_avg.Avg\_Price\_per\_Area > (

SELECT AVG(Price\_per\_Area) AS Overall\_Avg\_Price\_per\_Area

FROM Table2

);

 Retrieve properties from Table1 with a certain landmark that have a lower price per square foot than the average price per square foot for properties with the same landmark in Table2 (choose landmark on your own):

SELECT t1.\*

FROM Table1 t1

INNER JOIN (

SELECT Landmark, AVG(Price\_per\_Area) AS Avg\_Price\_per\_Area

FROM Table2

GROUP BY Landmark

) t2\_avg ON t1.Landmark = t2\_avg.Landmark

WHERE t1.Price\_per\_Area < t2\_avg.Avg\_Price\_per\_Area

AND t1.Landmark = 'Your Chosen Landmark';

 Retrieve properties from Table2 with a price per square foot higher than the average booking amount in Table3:

SELECT t2.\*

FROM Table2 t2

INNER JOIN (

SELECT AVG(Booking\_amount) AS Avg\_Booking\_Amount

FROM Table3

) t3\_avg ON t2.Price\_per\_Area > t3\_avg.Avg\_Booking\_Amount;

 Count the number of properties in Table2 with more bedrooms than the maximum number of bedrooms in Table3:

SELECT COUNT(\*)

FROM Table2 t2

WHERE t2.Beds > (

SELECT MAX(Beds)

FROM Table3

);

 Find the cities where the average booking amount in Table3 is higher than the overall average booking amount, and retrieve properties from Table1 located in those cities:

SELECT t1.\*

FROM Table1 t1

INNER JOIN (

SELECT City, AVG(Booking\_amount) AS Avg\_Booking\_Amount

FROM Table3

GROUP BY City

) t3\_avg ON t1.City = t3\_avg.City

WHERE t3\_avg.Avg\_Booking\_Amount > (

SELECT AVG(Booking\_amount) AS Overall\_Avg\_Booking\_Amount

FROM Table3

);

 Retrieve properties from Table1 with a furnished status of 'Unfurnished' and a facing direction that does not exist in Table3:

sql

Copy code

SELECT t1.\*

FROM Table1 t1

WHERE t1.Furnished\_status = 'Unfurnished'

AND t1.Facing NOT IN (

SELECT DISTINCT Facing

FROM Table3

);