

SQL PROJECT

PROPERTY LISTING ANALYSIS

FOR AUSTIN CITY

Analyze different metrics to draw the distinction between the different types of property along with their price listings (bucketize them within 3-4 categories basis your understanding): To achieve this, you can use the following metrics and explore a few yourself as well. Availability within 15,30,45, etc.

```
select a.property_type,sum(a.price) total_price,sum(b.available) as
total_avail,

case when sum(available)<=1000 then 'catag_1'
      when sum(available)>1000 and sum(available)<=10000 then
'catag_2'
      when sum(available)>10000 and sum(available)<=20000 then
'catag_3' else 'catag_4' end as catagorys
from listing_austin_df$ as a
left join df_austin_availability$ as b on a.id =b.listing_id
group by a.property_type
having sum(b.available) is not null
```

order by total_avail;

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
select id, available from df_austin_availability$
where available =1

select a.property_type, sum(a.price) total_price, sum(b.available) as total_avail,
case when sum(available)<=1000 then 'catag_1'
when sum(available)>1000 and sum(available)<=10000 then 'catag_2'
when sum(available)>10000 and sum(available)<=20000 then 'catag_3' else 'catag_4' end as catagorys
from listing_austin_df$ as a
left join df_austin_availability$ as b on a.id =b.listing_id
group by a.property_type
having sum(b.available) is not null
order by total_avail;
```

The Results pane shows the following data:

	property_type	total_price	total_avail	catagorys
1	Private room in bed and breakfast	39170	0	catag_1
2	Private room	37375	0	catag_1
3	Shared room in townhouse	136145	0	catag_1
4	Room in boutique hotel	239263	0	catag_1
5	Private room in condo	586587	0	catag_1
6	Shared room in camper/rv	146000	0	catag_1
7	Tiny home	56955	0	catag_1
8	Shared room in loft	255550	0	catag_1
9	Shared room	19395	0	catag_1
10	Private room in tent	26004	0	catag_1
11	Bus	65485	20	catag_1
12	Bam	56815	31	catag_1
13	Entire condo	7645061	58	catag_1
14	Private room in loft	137655	64	catag_1
15	Tent	45705	85	catag_1

The status bar at the bottom indicates "Query executed successfully." and "DESKTOP-2CS73DE\KARISHMA (1... DESKTOP-2CS73DE\user (71) | project1 | 00:00:00 | 52 rows".

avg_rating of room_types

SELECT distinct room_type,
Avg(review_scores_rating) AS average_of_review_score_rating
FROM [dbo].[listing_austin_df\$]
GROUP BY room_type;

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor at the top contains the following SQL code:

```
group by c.host_is_superhost) as mm  
  
SELECT distinct room_type,  
Avg(review_scores_rating) AS average_of_review_score_rating  
FROM [dbo].[listing_austin_df$]  
GROUP BY room_type
```

Below the query editor, the 'Results' tab is active, displaying the following data:

	room_type	average_of_review_score_rating
1	Hotel room	4.8
2	Shared room	4.599555555555556
3	Private room	4.72432317505315
4	Entire home/aprt	4.77215929953394

The status bar at the bottom indicates 'Query executed successfully.' and '4 rows'.

bedrooms & beds in room types

```
select distinct room_type,  
sum(bedrooms) as sum_of_bedroom,  
sum(beds) as sum_of_beds  
from [dbo].[listing_austin_df$]  
group by room_type  
order by room_type;
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL query:

```

select distinct room_type,
sum(bedrooms) as sum_of_bedroom,
sum(beds) as sum_of_beds
from [dbo].[listing_austin_df$]
group by room_type
order by room_type;

```

The Results pane displays the following data:

	room_type	sum_of_bedroom	sum_of_beds
1	Entire home/apt	17508	24333
2	Hotel room	20	32
3	Private room	1987	2206
4	Shared room	113	142

The status bar at the bottom indicates "Query executed successfully." and "4 rows".

average max & min nights avail for room types

SELECT room_type,

Avg(maximum_nights) **AS** average_of_maximum_nights,

Avg(minimum_nights) **AS** average_of_minimum_nights

FROM [dbo].[listing_austin_df\$]

GROUP BY room_type

order by room_type

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL query:

```
SELECT room_type,
       Avg(maximum_nights) AS average_of_maximum_nights,
       Avg(minimum_nights) AS average_of_minimum_nights
FROM [dbo].[listing_austin_df$]
GROUP BY room_type
order by room_type
```

The Results pane displays the following data:

	room_type	average_of_maximum_nights	average_of_minimum_nights
1	Entire home/apt	591.623060893725	8.61472563093309
2	Hotel room	528.9	2
3	Private room	546.16836461126	8.24343163538874
4	Shared room	749.70796460177	14.0884955752212

The status bar at the bottom indicates "Query executed successfully." and "4 rows".

property types available in year 2022

```
SELECT a.property_type,
sum (available) as avail
from [dbo].[listing_austin_df$] as a
left join [dbo].[df_austin_availability$] as b on a.id= b.listing_id
where year(b.date) = 2022
GROUP by a.property_type
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor at the top contains the following SQL query:

```
SELECT a.property_type,  
sum (available) as avail  
from [dbo].[listing_austin_df$] as a  
left join [dbo].[df_austin_availability$] as b on a.id= b.listing_id  
where year(b.date) = 2022  
GROUP by a.property_type
```

Below the query editor, the 'Results' tab is active, displaying a table with two columns: 'property_type' and 'avail'. The table contains 18 rows of data, representing different property types and their corresponding availability counts for the year 2022.

property_type	avail
Entire loft	6576
Entire villa	846
Boat	544
Tiny home	0
Bus	20
Shared room in rental unit	738
Private room in loft	64
Shared room	0
Entire rental unit	19116
Private room in bungalow	933
Campsite	580
Private room in floor	179
Shared room in camper/rv	0
Entire guesthouse	19999
Private room in condo	0
Entire vacation home	179
Barn	31
Entire residential home	91445

The status bar at the bottom indicates that the query was executed successfully. The Windows taskbar at the very bottom shows the system clock as 13:39 on 27-05-2022.

Property types available in year 2023

```
SELECT a.property_type,  
sum (available) as avail  
from [dbo].[listing_austin_df$] as a  
left join [dbo].[df_austin_availability$] as b on a.id= b.listing_id  
where year(b.date) = 2023  
GROUP by a.property_type
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor at the top contains the following SQL query:

```
SELECT a.property_type,
SUM(available) as avail
from [dbo].[listing_austin_df$] as a
left join [dbo].[df_austin_availability$] as b on a.id= b.listing_id
where year(b.date) = 2023
GROUP by a.property_type
```

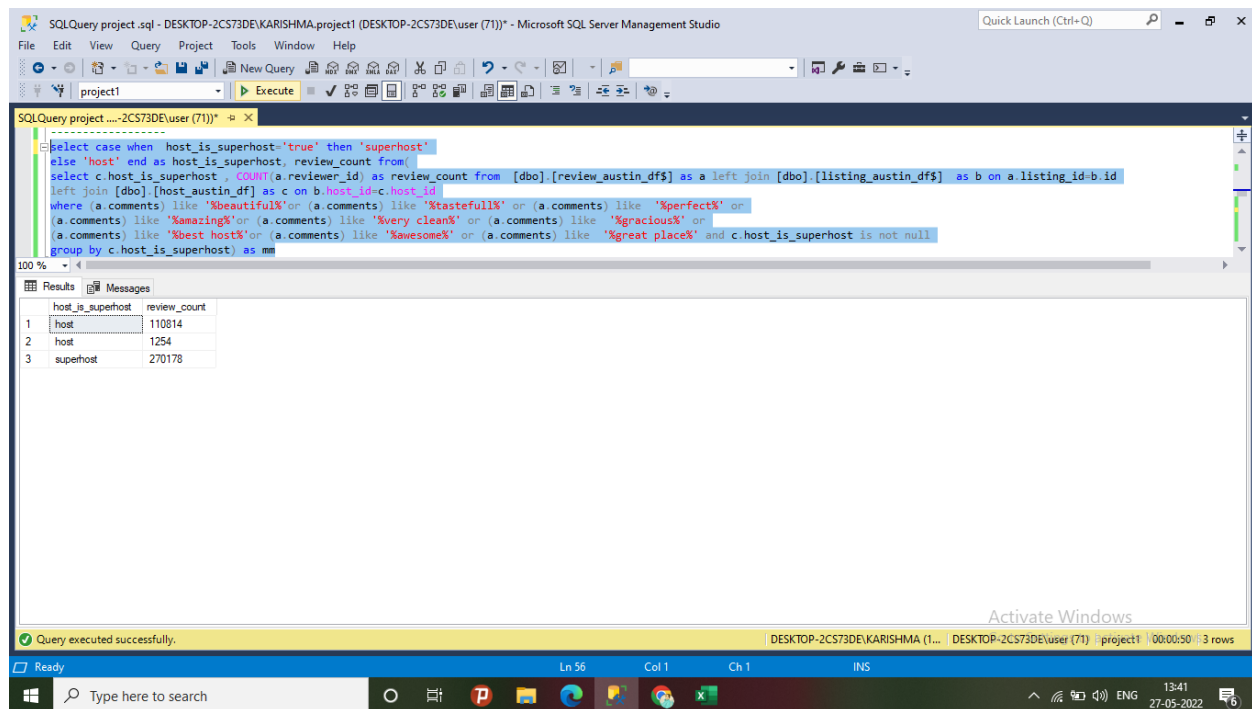
The Results pane below the query editor displays the output of the query as a table with two columns: property_type and avail. The data is as follows:

property_type	avail
Entire residential home	20799
Entire guest suite	1382
Private room in condominium (condo)	352
Private room in guesthouse	70
Shared room in townhouse	0
Shared room in rental unit	280
Private room in loft	0
Shared room	0
Entire loft	1129
Entire villa	282
Entire rental unit	4850
Private room in bungalow	214
Private room in floor	0
Shared room in camper/rv	0
Campsite	140
Entire guesthouse	3415
Private room in condo	0
Rem	0

The status bar at the bottom indicates that the query was executed successfully and returned 52 rows.

positive review for host and super host

```
select case when host_is_superhost='true' then 'superhost'
else 'host' end as host_is_superhost, review_count from(
select c.host_is_superhost , COUNT(a.reviewer_id) as review_count
from [dbo].[review_austin_df$] as a left join [dbo].[listing_austin_df$]
as b on a.listing_id=b.id
left join [dbo].[host_austin_df] as c on b.host_id=c.host_id
where (a.comments) like '%beautiful%' or (a.comments) like
'%tasteful%' or (a.comments) like '%perfect%' or
(a.comments) like '%amazing%' or (a.comments) like '%very clean%' or
(a.comments) like '%gracious%' or
(a.comments) like '%best host%' or (a.comments) like '%awesome%' or
(a.comments) like '%great place%' and c.host_is_superhost is not null
group by c.host_is_superhost) as mm
```



FOR DALLAS CITY

Analyze different metrics to draw the distinction between the different types of property along with their price listings (bucketize them within 3-4 categories basis your understanding): To achieve this, you can use the following metrics and explore a few yourself as well. Availability within 15,30,45, etc.

```

select a.property_type,sum(a.price) total_price,sum(b.available) as
total_avail,
case when sum(available)<=1000 then 'catag_1'
      when sum(available)>1000 and sum(available)<=10000 then
'catag_2'
      when sum(available)>10000 and sum(available)<=20000 then
'catag_3' else 'catag_4' end as catagorys
from [dbo].[listing_dallas_df$] as a
left join df_dallas_availability$ as b on a.id =b.listing_id
group by a.property_type
having sum(b.available) is not null

```


order by total_avail;

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
GROUP BY a.property_type
select a.property_type, sum(a.price) total_price, sum(b.available) as total_avail,
case when sum(available)<=1000 then 'catag_1'
when sum(available)>1000 and sum(available)<=10000 then 'catag_2'
when sum(available)>10000 and sum(available)<=20000 then 'catag_3' else 'catag_4' end as categorys
from [dbo].[listing_dallas_df$] as a
left join df_dallas_availability$ as b on a.id=b.listing_id
group by a.property_type
having sum(b.available) is not null
order by total_avail;
```

The Results pane shows the following data:

property_type	total_price	total_avail	categorys
Entire place	73868	0	catag_1
Private room in serviced apartment	17250	0	catag_1
Entire cabin	95265	7	catag_1
Private room in cottage	18346	65	catag_1
Yurt	35770	80	catag_1
Private room in bungalow	18587	102	catag_1
Private room in loft	84315	109	catag_1
Private room in guesthouse	42044	352	catag_1
Shared room in loft	18615	365	catag_1
Entire villa	233002	663	catag_1
Private room in bed and breakfast	221920	714	catag_1
Room in hotel	281706	717	catag_1
Camper/RV	201489	1056	catag_2
Shared room in rental unit	1047805	1124	catag_2
Entire cottage	441018	1174	catag_2

The status bar at the bottom indicates "Query executed successfully." and "DESKTOP-2CS73DE\KARISHMA (1... | DESKTOP-2CS73DE\user (53) | project | 00:00:02 | 32 rows".

avg_rating of room_types

SELECT distinct room_type,
Avg(review_scores_rating) AS average_of_review_score_rating
FROM [dbo].[listing_dallas_df\$]

GROUP BY room_type;

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
SELECT distinct room_type,  
Avg(review_scores_rating) AS average_of_review_score_rating  
FROM [dbo].[listing_dallas_df$]  
GROUP BY room_type;
```

The Results pane displays the output of the query, showing four rows of data:

	room_type	average_of_review_score_rating
1	Hotel room	4.40333333333333
2	Shared room	4.54710144927536
3	Private room	4.76901185770751
4	Entire home/apt	4.69449563145352

The status bar at the bottom indicates "Query executed successfully." and "DESKTOP-2CS73DE\KARISHMA (1... | DESKTOP-2CS73DE\user (52) | project | 00:00:00 | 4 rows".

bedrooms & beds in room types

```
select distinct room_type,  
sum(bedrooms) as sum_of_bedroom,  
sum(beds) as sum_of_beds  
from [dbo].[listing_dallas_df$]  
group by room_type  
order by room_type;
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The SQL query editor contains the following query:

```
select distinct room_type,
sum(bedrooms) as sum_of_bedroom,
sum(beds) as sum_of_beds
from [dbo].[listing_dallas_df$]
group by room_type
order by room_type;
```

The Results pane displays the following data:

room_type	sum_of_bedroom	sum_of_beds
Entire home/apl	8383	11245
Hotel room	3	5
Private room	742	854
Shared room	118	258

The status bar at the bottom indicates "Query executed successfully." and "4 rows".

average max & min nights avail for room types

SELECT room type,
Avg(maximum_nights) **AS** average_of_maximum_nights,
Avg(minimum_nights) **AS** average_of_minimum_nights
FROM [dbo].[listing_dallas_df\$]
GROUP BY room_type
order by room_type

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'DESKTOP-2CS73DE\KARISHMA (SQL Server)'. The central query window contains the following SQL query:

```
SELECT room_type,
       Avg(maximum_nights) AS average_of_maximum_nights,
       Avg(minimum_nights) AS average_of_minimum_nights
FROM [dbo].[listing_dallas_df$]
GROUP BY room_type
ORDER BY room_type
```

The Results pane at the bottom shows the output of the query, which is a table with three columns: 'room_type', 'average_of_maximum_nights', and 'average_of_minimum_nights'. The data is as follows:

room_type	average_of_maximum_nights	average_of_minimum_nights
Entire home/apt	550.404530744337	9.47335490830636
Hotel room	365	1
Private room	514.380952380952	5.34821428571429
Shared room	438.779661016949	2.63559322033898

The status bar at the bottom indicates 'Query executed successfully.' and '4 rows'.

property types available in year 2022

```
SELECT a.property_type,
sum (available) as avail
from [dbo].[listing_dallas_df$] as a
left join df_dallas_availability$ as b on a.id= b.listing_id
where year(b.date) = 2022
GROUP by a.property_type
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays a database structure with tables like 'dbo.dallas_availability\$'. The central query editor contains the following SQL code:

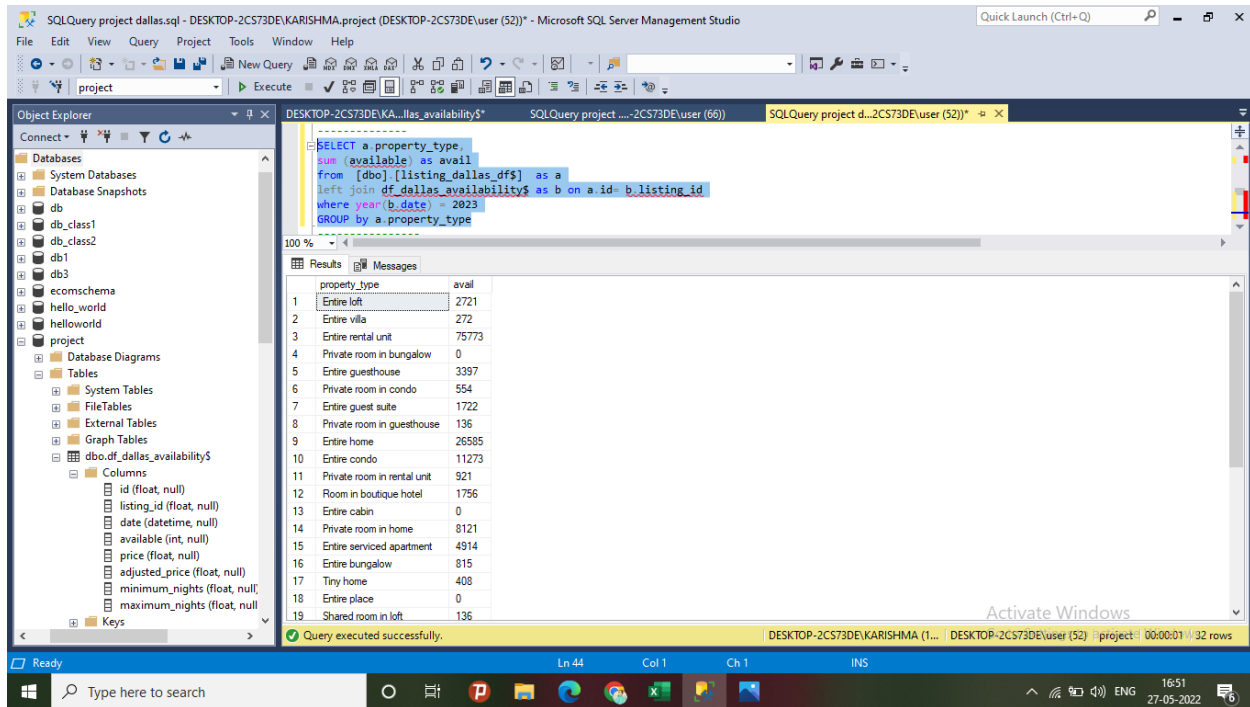
```
SELECT a.property_type,
sum (available) as avail
from [dbo].[listing_dallas_df$] as a
left join df_dallas_availability$ as b on a.id= b.listing_id
where year(b.date) = 2022
GROUP by a.property_type
```

The Results pane on the right shows the output of the query, which is a table with two columns: 'property_type' and 'avail'. The data is as follows:

property_type	avail
Entire guest suite	3457
Private room in guesthouse	216
Private room in cottage	65
Yurt	80
Camper/RV	650
Private room in bed and breakfast	442
Entire cottage	861
Entire bungalow	2235
Private room in home	15367
Entire serviced apartment	7695
Private room in townhouse	2515
Entire loft	4734
Entire villa	391
Entire rental unit	112...
Private room in bungalow	102
Private room in condo	1021
Entire guesthouse	6429
Entire home	55763
Entire condo	18366
Private room in rental unit	1616

property types available in year 2023

```
SELECT a.property_type,
sum (available) as avail
from [dbo].[listing_dallas_df$] as a
left join df_dallas_availability$ as b on a.id= b.listing_id
where year(b.date) = 2023
GROUP by a.property_type
```



positive review for host and super host

```
select case when host_is_superhost='true' then 'superhost'
else 'host' end as host_is_superhost, review_count from (
select c.host_is_superhost , COUNT(a.reviewer_id) as review_count
from [dbo].[review_dallas_df] as a left join [dbo].[listing_dallas_df$]
as b on a.reviewer_id=b.id
left join [dbo].[host_dallas_df$] as c on b.host_id=c.host_id
where (a.comments) like '%beautiful%' or (a.comments) like
'%tasteful%' or (a.comments) like '%perfect%' or
(a.comments) like '%amazing%' or (a.comments) like '%very clean%' or
(a.comments) like '%gracious%' or
(a.comments) like '%best host%' or (a.comments) like '%awesome%' or
(a.comments) like '%great place%' and c.host_is_superhost is not null
group by c.host_is_superhost) as mm
```

SQLQuery project dallas.sql - DESKTOP-2CS73DE\KARISHMA.project (DESKTOP-2CS73DE\user (53)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Object Explorer

- DESKTOP-2CS73DE\KARISHMA (SQL Server 13)
- Databases
- Security
- Server Objects
- Replication
- PolyBase
- Always On High Availability
- Management
- Integration Services Catalogs
- SQL Server Agent (Agent XPs disabled)
- XEvent Profiler

SQLQuery project d...2CS73DE\user (53)) SQLQuery project2CS73DE\user (51))

```
select case when host_is_superhost='true' then 'superhost'
else 'host' end as host_is_superhost, review_count from
select c.host_is_superhost, COUNT(a.reviewer_id) as review_count from [dbo].[review_dallas_df] as a left join [dbo].[listing_dallas_df] as
left join [dbo].[host_dallas_df] as c on b.host_id=c.host_id
where (a.comments) like '%beautiful%' or (a.comments) like '%tasteful%' or (a.comments) like '%perfect%' or
(a.comments) like '%amazing%' or (a.comments) like '%very clean%' or (a.comments) like '%gracious%' or
(a.comments) like '%best host%' or (a.comments) like '%awesome%' or (a.comments) like '%great place%' and c.host_is_superhost is not null
group by c.host_is_superhost as mm
```

Results

host_is_superhost	review_count
1 host	3
2 host	63514

100 %

Query executed successfully.

DESKTOP-2CS73DE\KARISHMA (1... | DESKTOP-2CS73DE\user (53) | project | 00:00:20 | 2 rows

Ready Ln 65 Col 1 Ch 1 INS

Type here to search

ENG 09:01 28-05-2022