```
| select host_name , host_location,
| case | when host_acceptance_rate <= 25 then 'worst' |
| when host_acceptance_rate <= 50 then 'normal' |
| when host_acceptance_rate <= 75 then 'good' |
| when host_acceptance_rate <=100 then 'excellent' |
| else 'not applicable' |
| end as ratings |
| from host_vancouver_df$;
```

% ▼ 4						
Results 📲 Messages						
host_name	host_location	ratings				
Lynn	Vancouver, British Columbia, Canada	good				
Iris	Vancouver, British Columbia, Canada	excellent				
Peter & Alison	Vancouver, British Columbia, Canada	good				
Jason	Vancouver, British Columbia, Canada	excellent				
Q	CA	worst				
Elaine	Vancouver, British Columbia, Canada	normal				
James	Vancouver, British Columbia, Canada	excellent				
Ran	Vancouver, British Columbia, Canada	not applicable				
Sylvain & Alexis	Vancouver, British Columbia, Canada	excellent				
Dean	Vancouver, British Columbia, Canada	excellent				
Kai	Vancouver, British Columbia, Canada	good				
Bevin	Vancouver, British Columbia, Canada	excellent				
Shirin	Vancouver, British Columbia, Canada	worst				
Tony And Rachel	Vancouver, British Columbia, Canada	excellent				
Adam	Vancouver, British Columbia, Canada	excellent				
Julia	Vancouver, British Columbia, Canada	excellent				
Alicia	Vancouver, British Columbia, Canada	excellent				
116	Vancouver Pritish Columbia Canada	oveellent				

Query executed successfully.

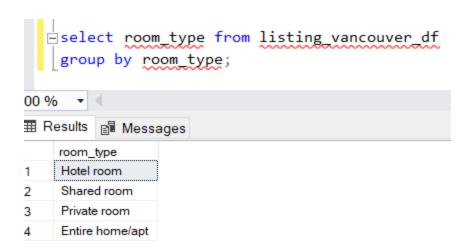
For toronto

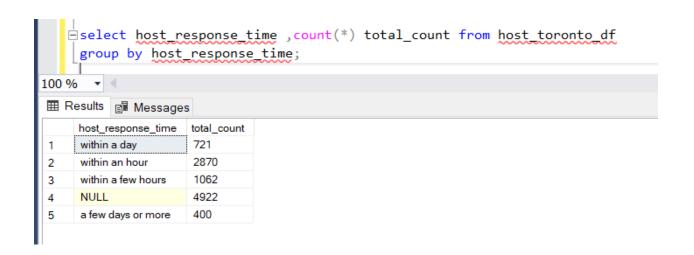
```
∃select host name , host_location,
      case
      when host_acceptance_rate <= 25 then 'worst'
      when host_acceptance_rate <= 50 then 'normal'
      when host acceptance rate <= 75 then 'good'
      when host_acceptance_rate <=100 then 'excellent'
      else 'not applicable'
      end as ratings
      from host_toronto_df;
100 %
       •

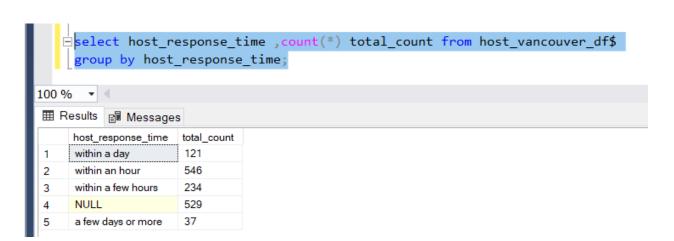
    ⊞ Results

    Messages

      host name
                               host_location
                                                                 ratings
      Faiza And Adam
                               Toronto, Ontario, Canada
 260
                                                                 good
      Darius
                               Toronto, Ontario, Canada
 261
                                                                 normal
      Stefan
                               Toronto, Ontario, Canada
 262
                                                                 good
 263
      Roman
                               Toronto, Ontario, Canada
                                                                 good
      Melanie
                               Montreal, Quebec, Canada
 264
                                                                 good
      Muhammad
                               Toronto, Ontario, Canada
 265
                                                                 not applicable
                               Toronto, Ontario, Canada
 266
      Amir
                                                                 not applicable
      Cindy
                               Toronto, Ontario, Canada
                                                                 normal
 267
 268
      Arghavan
                               Toronto, Ontario, Canada
                                                                 excellent
      Mike
                               Toronto, Ontario, Canada
 269
                                                                 excellent
      Juli
                               Toronto, Ontario, Canada
 270
                                                                 good
 271
      Stephen
                               Toronto, Ontario, Canada
                                                                 not applicable
                               Toronto, Ontario, Canada
      Laurie
                                                                 not applicable
 272
      Isil
                               Toronto, Ontario, Canada
 273
                                                                 good
 274
      Kona Kie
                               Toronto, Ontario, Canada
                                                                 not applicable
 275
      Karen
                               Toronto, Ontario, Canada
                                                                 aood
      Alexandra
                               Toronto, Ontario, Canada
                                                                 excellent
 276
      Laura
                               Toronto, Ontario, Canada
                                                                 worst
 277
      Gloria
                               Toronto, Ontario, Canada
                                                                 not applicable
 278
      Kirill
                               Toronto, Ontario, Canada
                                                                 not applicable
 279
      Matthias
                               Toronto, Ontario, Canada
 280
                                                                 not applicable
      Fariba
                               Toronto, Ontario, Canada
                                                                 not applicable
 281
```







100 % ▼

	property_type	min_prices	max_prices
1	Boat	250	2600
2	Camper/RV	102	175
3	Entire bungalow	89	259
4	Entire cabin	25	275
5	Entire condo	42	2100
6	Entire condominium (condo)	45	6500
7	Entire cottage	106	214
8	Entire guest suite	40	610
9	Entire guesthouse	49	599
10	Entire home	21	1200
11	Entire loft	50	1588
12	Entire place	68	352
13	Entire rental unit	27	3250
14	Entire residential home	25	9999
15	Entire serviced apartment	45	9999
16	Entire timeshare	210	247
17	Entire townhouse	80	827
18	Entire vacation home	179	1500
19	Entire villa	99	1200
20	Floor	146	146
21	Houseboat	208	325
22	Private room in bed and breakfast	50	801
23	Private room in boat	350	1700
24	Private room in bungalow	35	215

select property type, min(price) as min_prices ,max(price) as max_prices from listing toronto df group by property type;

	property_type	min_prices	max_prices
1	Entire loft	50	6848
2	Entire condominium (condo)	29	11500
3	Camper/RV	188	400
4	Entire cottage	100	699
5	Shared room in barn	60	60
6	Shared room in boat	150	150
7	Private room in bungalow	14	500
8	Entire villa	120	5000
9	Private room in bed and breakfast	36	209
10	Private room in home	13	11800
11	Entire serviced apartment	70	1613
12	Private room in guest suite	25	168
13	Private room in condo	14	1248
14	Shared room in townhouse	14	125
15	Shared room in condo	39	80
16	Entire place	97	499
17	Shared room in hotel	38	38
18	Private room in rental unit	14	2000
19	Entire rental unit	22	12400
20	Private room in vacation home	79	85
21	Private room in earth house	60	60
22	Tiny house	55	160
23	Private room in loft	35	1400
24	Entire guesthouse	48	301

```
select id ,review_scores_rating,
      case
      when review_scores_rating<= 2 then 'worst'
      when review_scores_rating <= 3 then 'normal'
      when review_scores_rating <= 4 then 'good'
      when review_scores_rating <= 5 then 'excellent'
      else 'not applicable'
      end as ratings
      from listing_toronto_df
100 %

    ⊞ Results

    Messages

     id
             review_scores_rating
                               ratings
      1419
                                excellent
      8077
             4.84000015258789
                                excellent
      23691
            4.75
                                excellent
3
      27423
             4.92000007629395
                                excellent
      30931
 5
                                excellent
      40456
           4.63000011444092
                               excellent
      41887 4.71999979019165
                               excellent
      42892 NULL
8
                               not applicable
      43964 4.96999979019165
 9
                               excellent
      44452 4.15000009536743
                               excellent
 10
      44469
 11
                                good
      45399 4.13000011444092
 12
                               excellent
      45453 4.13000011444092
                               excellent
 13
      45601 NULL
                               not applicable
 14
      45893 4.26999998092651
                               excellent
 15
      50110 4.61999988555908
 16
                               excellent
      51616 4.88000011444092
 17
                               excellent
      64635
           4.55000019073486
 18
                               excellent
      64641 4 76000022888184
                              evcellent

    Query executed successfully.
```

```
select id ,review_scores_rating,
   case
   when review_scores_rating<= 2 then 'worst'
   when review_scores_rating <= 3 then 'normal'
   when review scores rating <= 4 then 'good'
   when review_scores_rating <= 5 then 'excellent'
   else 'not applicable'
   end as ratings
   from listing vancouver df
) %
Results Messages
   id
          review_scores_rating
                            ratings
   13358
          4.67000007629395
                            excellent
   13490
          4.92000007629395
                            excellent
   14267
          4.76000022888184
                            excellent
   16254
          4.80000019073486
                            excellent
   16611
                            good
   17158 5
                            excellent
   17765
          4.59999990463257
                            excellent
   18270
          4.53999996185303
                            excellent
   18589
          4.98000001907349
                            excellent
   18773 4.21999979019165
                            excellent
   18795
          4.55000019073486
                            excellent
   19527
          4.69000005722046
                            excellent
   19621
          NULL
                            not applicable
   20085
          4.82000017166138
                            excellent
   21912 4.69000005722046
                            excellent
   21913
          4.73999977111816
                            excellent
   24273
                            excellent
   26047
          4.76000022888184
                            excellent
```

evcellent

22150 / 06000070010165

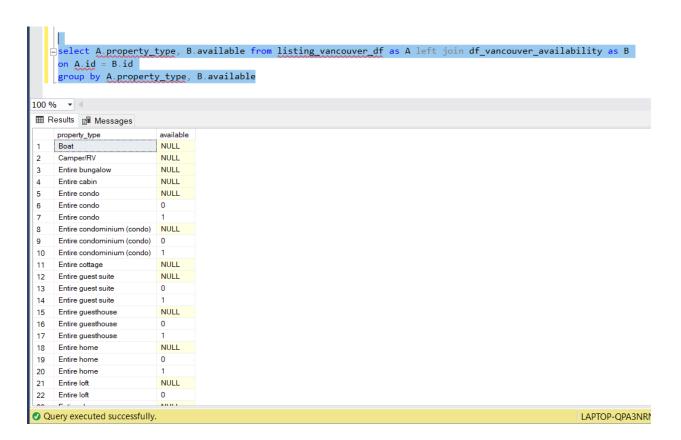
select property_type from listing_vancouver_df group by property_type;

100 % ▼ ◀

	E- moodagoo
	property_type
1	Boat
2	Camper/RV
3	Entire bungalow
4	Entire cabin
5	Entire condo
6	Entire condominium (condo)
7	Entire cottage
8	Entire guest suite
9	Entire guesthouse
10	Entire home
11	Entire loft
12	Entire place
13	Entire rental unit
14	Entire residential home
15	Entire serviced apartment
16	Entire timeshare
17	Entire townhouse
18	Entire vacation home
19	Entire villa
20	Floor
21	Houseboat
22	Private room in bed and breakfast
23	Private room in boat
24	Private room in bungalow
25	Private room in camper/rv
26	Private room in casa particular
27	Private room in condo
28	Private room in condominium (condo)
29	Private room in earth house







Count super host VS normal host over profile pic

```
use projectsql
 select
   sum( case when host_is_superhost='true' and host_has_profile_pic='true'then 1 else 0 end ) as superhost_with_profilepic,
   sum( case when host_is_superhost='true' and host_has_profile_pic='false'then 1 else 0 end ) as superhost_without_profilepic,
   sum( case when host_is_superhost='false' and host_has_profile_pic='true'then 1 else 0 end ) as superhost_with_profilepic,
   sum( case when host_is_superhost='false' and host_has_profile_pic='false'then 1 else 0 end ) as superhost_without_profilepic
  from host_toronto_df;
)% -
Results Messages

        superhost_with_profilepic
        superhost_without_profilepic
        superhost_with_profilepic
        superhost_without_profilepic

                                         7532
   select
     sum( case when host_is_superhost='true' and host_has_profile_pic='true'then 1 else 0 end ) as superhost_with_identity,
     sum( case when host_is_superhost='true' and host_has_profile_pic='false'then 1 else 0 end ) as superhost_without_identity,
     sum( case when host_is_superhost='false' and host_has_profile_pic='true'then 1 else 0 end ) as superhost_with_identity,
     sum( case when host_is_superhost='false' and host_has_profile_pic='false'then 1 else 0 end ) as superhost_without_identity
    from host_toronto_df;
100 % ▼ 4
superhost_with_identity superhost_without_identity superhost_with_identity superhost_without_identity
    2351
                     2
                                         7532
                                                           84
```

For vancouver

```
use projectsql
   select
    sum( case when host_is_superhost='true' and host_has_profile_pic='true'then 1 else 0 end ) as superhost_with_profilepic,
    sum( case when host_is_superhost='true' and host_has_profile_pic='false'then 1 else 0 end ) as superhost_without_profilepic,
    sum( case when host is superhost='false' and host has profile pic='true'then 1 else 0 end ) as superhost with profilepic,
    sum( case when host_is_superhost='false' and host_has_profile_pic='false'then 1 else 0 end ) as superhost_without_profilepic
   from host_vancouver_df$;
.00 % 🔻 🔻
superhost_with_profilepic superhost_without_profilepic superhost_with_profilepic superhost_without_profilepic
                                           816
    ----identity verified
  select
    .
sum( case when host_is_superhost=<mark>'true'</mark> and host_has_profile_pic=<mark>'true'then 1 else 0 end</mark> ) as superhost_with_identity,
    sum( case when host_is_superhost='true' and host_has_profile_pic='false'then 1 else 0 end ) as superhost_without_identity,
    sum( case when host_is_superhost='false' and host_has_profile_pic='true'then 1 else 0 end ) as superhost_with_identity,
    sum( case when host_is_superhost='false' and host_has_profile_pic='false'then 1 else 0 end ) as superhost_without_identity
   from host_vancouver_df$;
.00 % 🔻 🖪

    ■ Results    ■ Messages
    superhost_with_identity superhost_without_identity superhost_with_identity superhost_without_identity
```

```
⊟Select distinct h.host_id , Avg(L.price) over (partition by H. Host_id)
    From listing_vancouver_df as L
    Inner join host_vancouver_df$ H on L.Host_id = H. Host_id
    Where H.host_is_superhost = 'FALSE'
    order by h. Host_id;
   ⊟Select distinct h.host_id , Avg(L.price) over (partition by H. Host_id)
     From listing_vancouver_df as L
     Inner join host_vancouver_df$ H on L.Host_id = H. Host_id
     Where H.host_is_superhost = 'TRUE'
     order by h. Host_id;
   Select h.host_id , Avg(L.price) as AVG_PRICE_PER_HOST, Count(A.available) as AVAILABLE_COUNT
     From listing_vancouver_df as L
     Inner join host_vancouver_df$ H on L.Host_id = H. Host_id
     Inner join df_vancouver_availability as A on A.listing_id = L.id
    Where A.available = 'TRUE' and
     YEAR(A.Date) = '2023' and
    H.host_is_superhost = 'TRUE'
    group by H.host_id
    having AVG(L.price) > 200;
100 % 🔻 🖪
host id
             AVG_PRICE_PER_HOST AVAILABLE_COUNT
    100287076 600
    13505875
                                68
    12260950
3
              300
                               68
    66638660
                               68
              214
                               68
5
    141458230 236
    7151163
              240
                               69
6
    64481384
                               65
              327
8
    8073767
              299
                               68
    8924838
              258
                               69
10
    160895495 220
                               68
    126002388 247
                               69
11
    55967577
                               69
              450
12

    Query executed successfully.
```

```
---no of super host atleast one big listing

| select count(host_id) from(
| select h.host_id, count(h.host_id) as CC from host_vancouver_df$ H
| inner join listing_vancouver_df L on H.host_id = L.host_id|
| where (property_type like '%entire%' or
| property_type like '%Barn%' or
| property_type like '%Barn%' or
| property_type like '%Boat%')
| and host_is_superhost = 'True'
| group by h.host_id) a
```

For toronto

```
Eselect count(host_id) from(
select h.host_id, count(h.host_id) as CC from host_toronto_df H
inner join listing_toronto_df L on H.host_id = L.host_id
where (property_type like '%entire%' or
property_type like '%houeboat%' or
property_type like '%Barn%' or
property_type like '%Boat%')
and host_is_superhost = 'True'
group by h.host_id) a

## Results Messages

(No column name)

1 1915
```

For toronto

Total no of host who are local to their listing property and having more than acceptance rate of 80

Total number of host who are not local to their listing property and having more than acceptance rate of 80

```
----total no of host who are not local to their listing property and having more than acceptance rate of 80

select count (*) from(
select H.host_id, H.host_acceptance_rate from host_vancouver_df$ as H
inner join listing_vancouver_df as L on H.host_neighbourhood = L.neighbourhood_cleansed and H.host_id = L.host_id
group by H.host_id, H.host_acceptance_rate
having H.host_acceptance_rate > 80) AA

100 % 

Results Messages

(No column name)
1 390
```

```
---- for toronto
----- select count (*) from(
select H.host_id, H.host_acceptance_rate from host_toronto_df as H
inner join listing_toronto_df as L on H.host_neighbourhood = L.neighbourhood_cleansed and H.host_id = L.host_id
group by H.host_id, H.host_acceptance_rate
having H.host_acceptance_rate > 80) AA

100 % 
Results 
Res
```