```sql
create database project
use project

---toronto
 select *from toronto_host;
 select*from toronto_listing;
 select*from toronto_availability;
  select*from toronto_review;

  ---vancouver
 select *from vancouver_host;
 select*from vancouver_listing;
 select*from vancouver_availability;
 select*from vancouver_review;







 ---1st insight

 ---a. Analyze different metrics to draw the distinction between Super Host and Other
Hosts:
--To achieve this, you can use the following metrics and explore a few yourself as well.
--Acceptance rate, response rate, instant booking, profile picture, identity verified,
review scores, average no
--of bookings per month, etc
a)
--<acceptance rate>---toronto

select host_is_superhost, avg(host_acceptance_rate) as avg_acceptance from toronto_host
group by host_is_superhost
having host_is_superhost = 'true' or host_is_superhost = 'false';

--<acceptance rate>---vancouver

select host_is_superhost, avg(host_acceptance_rate) as avg_acceptance from vancouver_host
group by host_is_superhost
having host_is_superhost = 'true' or host_is_superhost = 'false';

b)

--<response rate>---toronto

select host_is_superhost, avg(host_response_rate) as avg_response from toronto_host group
by host_is_superhost
having host_is_superhost = 'true' or host_is_superhost = 'false'

--<response rate>---vancouver

select host_is_superhost, avg(host_response_rate) as avg_response from vancouver_host
group by host_is_superhost
having host_is_superhost = 'true' or host_is_superhost = 'false'

c)
```

```
--<instant booking>---toronto

select a.host_is_superhost, count(b.instant_bookable) as count_bookable
from toronto_host as a inner  join toronto_listing as b on a.host_id=b.host_id group by
host_is_superhost
having host_is_superhost = 'true' or host_is_superhost = 'false'

--<instant booking>---vancouver

select a.host_is_superhost, count(b.instant_bookable) as count_bookable
from vancouver_host as a inner  join vancouver_listing as b on a.host_id=b.host_id group
by host_is_superhost
having host_is_superhost = 'true' or host_is_superhost = 'false'

d)

--<profile picture>---toronto

--- total superhost

select host_is_superhost , count(host_is_superhost) as total_superhost from toronto_host
where host_is_superhost = 'true'
group by host_is_superhost

--2353

--superhost who has profile pitcure.

select host_is_superhost ,count(host_has_profile_pic) as superhost_with_PP from
toronto_host where  host_is_superhost
 ='true' and host_has_profile_pic = 'true' group by host_is_superhost;

 --2351


--<profile picture>---vancouver

--- total superhost

select host_is_superhost , count(host_is_superhost) as total_superhost from
vancouver_host where host_is_superhost = 'true'
group by host_is_superhost

--1228

--superhost who has profile pitcure.

select host_is_superhost ,count(host_has_profile_pic) as superhost_with_PP from
vancouver_host where  host_is_superhost
 ='true' and host_has_profile_pic = 'true' group by host_is_superhost;

 --1225

 e)

 --<review scores>---toronto

 select a.host_is_superhost , avg(b.review_scores_value) as avg_review
```

```
from toronto_host as a inner  join toronto_listing as b on a.host_id=b.host_id group by
host_is_superhost
having host_is_superhost = 'true' or host_is_superhost = 'false'


--<review scores>---vancouver

select a.host_is_superhost , avg(b.review_scores_value) as avg_review
from vancouver_host as a inner  join vancouver_listing as b on a.host_id=b.host_id group
by host_is_superhost
having host_is_superhost = 'true' or host_is_superhost = 'false'

f)


--<average no of bookings per month>

--toronto

select a.host_is_superhost, count(a.host_is_superhost) as count_true_false,
avg(a.host_acceptance_rate)as avg_acceptance_rate,
avg(a.host_response_rate) as avg_response , avg(b.review_scores_value) as avg_review
,datepart(month,c.date) as month,
datepart(year,c.date) as year, count(c.id) as total_bookings
from toronto_host as a inner join toronto_listing as b on a.host_id=b.host_id inner join
toronto_availability as c
on c.id=b.id group by host_is_superhost,datepart(month,c.date),datepart(year,c.date)
having host_is_superhost = 'true' or host_is_superhost = 'false'
order by year desc;

-- vancouver

select a.host_is_superhost, count(a.host_is_superhost) as count_true_false,
avg(a.host_acceptance_rate)as avg_acceptance_rate,
avg(a.host_response_rate) as avg_response , avg(b.review_scores_value) as avg_review
,datepart(month,c.date) as month,
datepart(year,c.date) as year,count(c.id) as total_bookings
from vancouver_host as a inner join vancouver_listing as b on a.host_id=b.host_id inner
join vancouver_availability as c
on c.id=b.id group by host_is_superhost,datepart(month,c.date),datepart(year,c.date)
having host_is_superhost = 'true' or host_is_superhost = 'false'
order by year desc;

---b. Using the above analysis, identify top 3 crucial metrics one needs to maintain to
become a Super Host and also, find
--their average values.

--toronto

update toronto_listing set review_scores_value = (0) where review_scores_value  is null;

select a.host_is_superhost, count(a.host_is_superhost) as count_true_false,
avg(a.host_acceptance_rate)as avg_acceptance_rate,
avg(a.host_response_rate) as avg_response , avg(b.review_scores_value) as avg_review
from toronto_host as a inner  join toronto_listing as b on a.host_id=b.host_id group by
host_is_superhost
having host_is_superhost = 'true' or host_is_superhost = 'false'
```

```sql
---vancouver

select a.host_is_superhost, count(a.host_is_superhost) as count_true_false,
avg(a.host_acceptance_rate)as avg_acceptance_rate,
avg(a.host_response_rate) as avg_response , avg(b.review_scores_value) as avg_review
from vancouver_host as a inner join vancouver_listing as b on a.host_id=b.host_id group
by host_is_superhost
having host_is_superhost = 'true' or host_is_superhost = 'false'

---c. Analyze how does the comments of reviewers vary for listings of Super Hosts vs
Other Hosts(Extract words from the
--comments provided by the reviewers)

 select*from toronto_review;
  select*from toronto_listing;
  select*from toronto_host;

  select c.host_is_superhost,a.comments from toronto_host  as c left join toronto_listing
as b
  on c.host_id=b.host_id left join toronto_review as a on b.id = a.id;
```