

Case Study

Airbnb

connecting hosts with travelers

urvi kalia

Table of Contents

Background.....	2
Problem statement.....	2
Stakeholders.....	2
Scope.....	2
References.....	2
Queries from different stake holders.....	3
Management Team.....	3
Marketing Team.....	3
Operational Team.....	3
Design & Implementation Approach.....	4
ER Diagram.....	4
Queries.....	5
Queries by Management Team.....	5
Queries for Marketing Team.....	6
Queries for Operational Team.....	7
Sample R code for data generation.....	8

Background

Airbnb is an online marketplace for vacation rentals that connects hosts and travelers. It has over 1,500,000 listings in 34,000 cities and 190 countries. Unlike traditional hotels, Airbnb scales not by scaling inventory but by increasing hosts and travelers and matching them with each other. Airbnb's primary source of revenue comes from service fee from bookings.

Problem statement

Analysing the given data, provide useful insights about :

1. Improve ground operations.
2. Targeting specific users based on various parameters.
3. Any other value addition to the business directly / indirectly.

Stakeholders

1. Operational team : Responsible for group operations.
2. Marketing team : Responsible for marketing.
3. Management team : Responsible for making business decisions.

Scope

1. This case study is not covering all the aspects of Airbnb.
2. Have limited its scope only for India.
3. Data used is populated using code and not real.

References

1. Airbnb website : <https://www.airbnb.com/>
2. <https://en.wikipedia.org/wiki/Airbnb>
3. <http://yourstory.com/2014/05/airbnb-india/>

Queries from different stake holders

Below are the queries , which have come from different stake holders. Need to provide data satisfying the given condition.

Management Team

1. Management team needs to understand number of guests newly registered on yearly basis.
2. Identify the guests who are dormant in the year 2016.
3. List of Guest's with their cancellation rates.
4. Management want's to know which properties are best rated .List of property's (their cities) whose average rating is above 4.
5. Team want's to know percentage of property's city wise.

Marketing Team

Marketing team has comeup with queries , basically they want to identify guest/host who meet the specific requirement. So as to send them promotional mails, or reminders. Requirements are as below:

1. List of guests along with their mailid's who have searched for the same location more than once.
2. List of properties , which provide ammenities , which are amongst the top 2 ammenities across india.
3. Which is the device most used for surfing airbnb website.

Operational Team

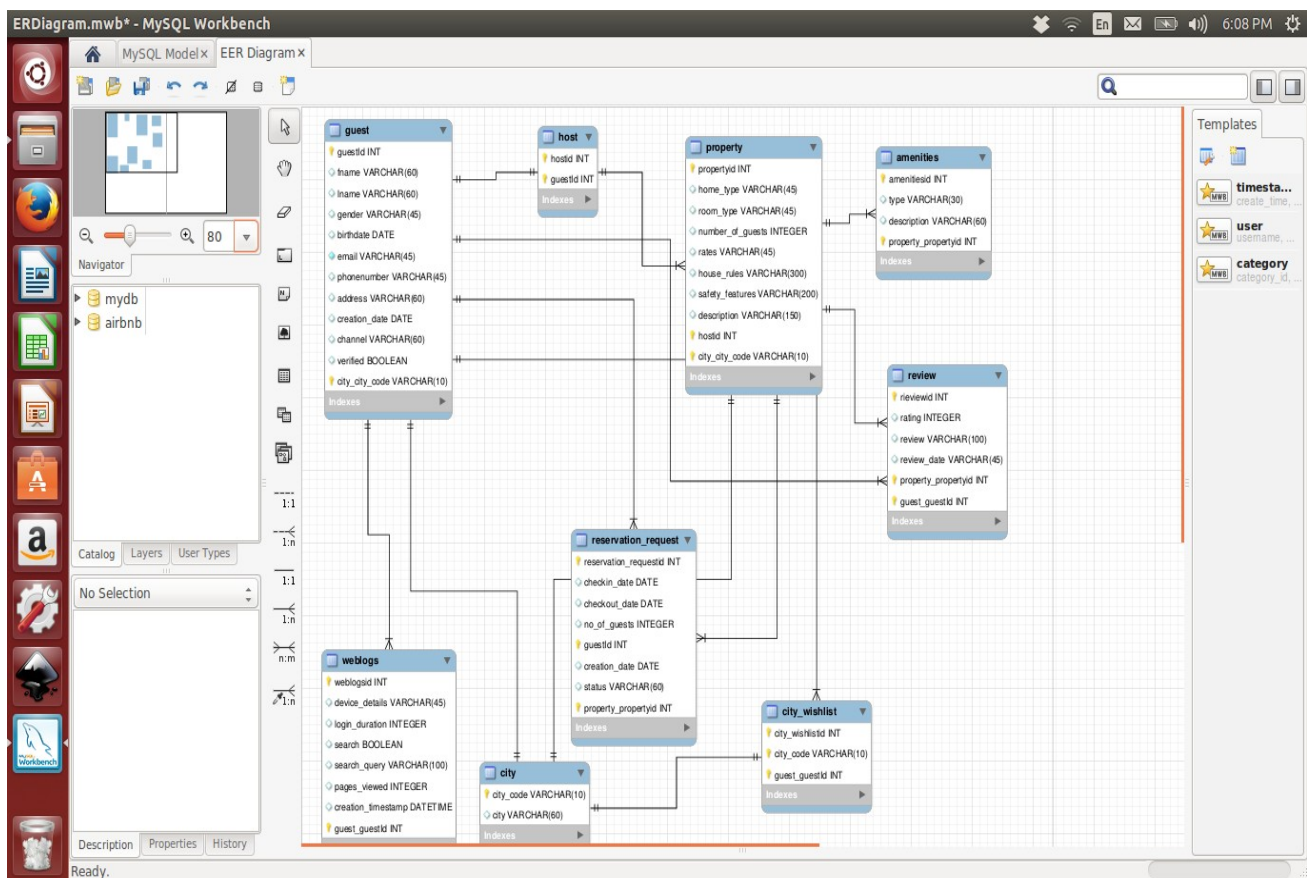
Operational team is responsible for taking some actions/ detailed analysis according to the findings from the data .

1. Identify Property against which a reservation was never done.
2. Which city has maximum and minimum number of travelers.
3. Guest's emailId , phone number who haven't ever made a single reservation.
4. Host names and email who have served guests in the year 2015

Design & Implementation Approach

Myself being an avid traveller , airbnb came to me as a natural choice of subject for study. Browsed through their website , understanding the important entities and information being captured. Using that as a reference point came up with ER diagram. ER diagram was build using mysql workbench. Further using forward engineering option in mysql workbench created the database. Used R programming language for generating dummy data for the tables created. Further wrote the queries for the requirements which were self crafted.

ER Diagram



Queries

Queries by Management Team

1. Management team needs to understand number of guests newly registered on yearly basis.

```
select year(g.creation_date),count(*) from guest g
group by year(g.creation_date) order by year(g.creation_date);
```

2. Identify the guests who are dormant in the year 2016.

```
select fname,email,phonenumner from guest
where guestId not in
(select guestId from reservation_request r
where year(r.creation_date) =2016);
```

3. List of Guest's with their cancellation rates .

```
select y.guestId, (t.cnlcnt/y.allcnt)*100 as cancellation_rate from (select guestId,count(*) allcnt from
reservation_request group by guestId) y
, (select guestId,count(*) cnlcnt from reservation_request where status ="cancelled" group by guestId)
t where y.guestId =t.guestId;
```

4. List of property's (their cities) whose average rating is 4 or above.

```
select p.propertyid,p.city_city_code from property p,review r
where p.propertyid =r.property_propertyid
group by r.property_propertyid having avg(r.rating) >4;
```

5. Team want's to know percentage of property's city wise.

```
select p.city_city_code,round((count(*)/(select count(*) from property))*100) as percentage from
property p
group by p.city_city_code order by percentage desc;
```

Queries for Marketing Team

1. List of guests(email) who have searched for the same location multiple times.

--Unable to do the below query

```
select w1.guest_guestId from weblogs w1
where w1.search_query = (select search_query from weblogs w2 where w1.guest_guestId
=w2.guest_guestId)
```

2. List of properties , which provide ammenities , which are amongst the top 2 ammenities across property's.

```
select propertyId from property p,amenities a where
p.propertyid = a.property_propertyid
and a.type in
(select a.type from property p,amenities a
where p.propertyid = a.property_propertyid
group by a.type order by count(*) desc limit 0,2)
```

Note : -- Note mysql is not allowing to use limit in subquery (in the version i am using).

kindly let me know other option

3. Which is the device most used for surfing airbnb website.

```
select device_details from weblogs
group by device_details
order by count(*)
desc limit 1
```

Queries for Operational Team

1. Identify Property against which a reservation was never done.

```
select propertyid from property where propertyid not in  
(select distinct property_propertyid from reservation_request);
```

2. Which city has maximum and minimum number of travelers.

```
select c.city,count(*) from guest g,city c where c.city_code =g.city_city_code  
group by g.city_city_code order by count(*) desc;
```

3. Guests who haven't ever accepted/made a single reservation.

```
select fname,email,phonenumber from guest where guestId not in  
(select g.guestId from guest g,reservation_request r  
where g.guestId =r.guestId);
```

4. Host names and email who have served guests in the year 2015.

```
select fname,email from guest where guestId in(  
select guestId from host where hostid in(  
select distinct hostid from reservation_request r,property p where year(creation_date) =2015  
and status ="accepted" and year(checkin_date) =2015 and year(checkout_date) =2015  
and r.property_propertyid =p.propertyid));
```

Sample R code for data generation

```

15 #Master data
16 propertyId <- 1:91
17 homeType <- c("Apartment","House","Bread & Breakfast")
18 roomType <- c("Entire House/apt","Private House","Shared Room")
19 noOfGuests <- c(1,2,3,4)
20 rates <- c(600,1200,2400,3500)
21 houseRules <- c("No smoking", "Only rule is no rule","No alcohol","help yourself")
22 safety_features <- c("Heat sensor","LPG Gas detector")
23 description <- c("Home away from home","friendly place","peaceful space")
24 hostId <- read.csv("hostData.csv")
25 hostId <- hostId[,1]
26 cityCodeID <- 1:8
27 cityCodes <- c("MUM","DEL","KOL","CHN","BGL","HYD","AHM","PUN")
28 #####
29
30
31 houseList <- NULL
32 roomtypeList <- NULL
33 guests <- NULL
34 rateList <- NULL
35 ruleList <- NULL
36 safetyList <- NULL
37 descriptionList <- NULL
38 hostIdList <- NULL
39 cityCodeList <- NULL
40
41 for( x in seq(1:91))
42 {
43   #data <- sample(c(1:length(homeType)),1)
44   #houseList <- c(houseList,homeType[data])
45
46   data <- sample(c(1:length(roomType)),1)
47   roomtypeList <- c(roomtypeList,roomType[data])
48
49   data <- sample(c(1:length(noOfGuests)),1)
50   guests <- c(guests,data)
51
52   data <- sample(c(1:length(rates)),1)
53   rateList <- c(rateList,rates[data])
54
55   data <- sample(c(1:length(houseRules)),1)
56   ruleList <- c(ruleList,houseRules[data])
57
58   data <- sample(c(1:length(safety_features)),1)
59   safetyList <- c(safetyList,safety_features[data])
60
61   data <- sample(c(1:length(description)),1)
62   descriptionList <- c(descriptionList,description[data])
63
64   #hostId
65   data <- sample(c(1:length(hostId)),1)
66   hostIdList <- c(hostIdList,data)
67
68   #cityCodeID
69   data <- sample(c(1:length(cityCodes)),1)
70   cityCodeList <- c(cityCodeList,cityCodes[data])
71 }
72
73 property <- cbind(propertyId,houseList,roomtypeList,guests,rateList,ruleList,safetyList,des
74 descriptionList,hostIdList,cityCodeList)
75
76 write.csv(file = "propertyData.csv",x = property)
77

```

The Environment pane shows the following data:

Variable	Values
guestData	chr [1:180, 1:11] "uknbt" "nagkq" "ldrqt" "wehah" "he..."
address	"jebhnanqcof"
addressList	chr [1:180] "gsfxawvzbzhj" "azdtqfxbrcv" "whevbiymroq..."
bdateData	"1989-2-11"
bdateList	chr [1:180] "1988-6-15" "1996-10-15" "1985-9-25" "1999-..."
beta0	7.071
beta1	-0.545
booleanData	chr [1:2] "yes" "no"

```

43 #data <- sample(c(1:length(homeType)),1)
44 #houseList <- c(houseList,homeType[data])
45
46 data <- sample(c(1:length(roomType)),1)
47 roomtypeList <- c(roomtypeList,roomType[data])
48
49 data <- sample(c(1:length(noOfGuests)),1)
50 guests <- c(guests,data)
51
52 data <- sample(c(1:length(rates)),1)
53 rateList <- c(rateList,rates[data])
54
55 data <- sample(c(1:length(houseRules)),1)
56 ruleList <- c(ruleList,houseRules[data])
57
58 data <- sample(c(1:length(safety_features)),1)
59 safetyList <- c(safetyList,safety_features[data])
60
61 data <- sample(c(1:length(description)),1)
62 descriptionList <- c(descriptionList,description[data])
63
64 #hostId
65 data <- sample(c(1:length(hostId)),1)
66 hostIdList <- c(hostIdList,data)
67
68 #cityCodeID
69 data <- sample(c(1:length(cityCodes)),1)
70 cityCodeList <- c(cityCodeList,cityCodes[data])
71 }
72
73 property <- cbind(propertyId,houseList,roomtypeList,guests,rateList,ruleList,safetyList,des
74 descriptionList,hostIdList,cityCodeList)
75
76 write.csv(file = "propertyData.csv",x = property)
77

```

The Environment pane shows the same data as the first screenshot.