

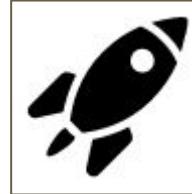


Relational Database Management System for AirBnB Inc.

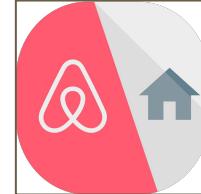


Group Members
Divya Rahunathan
Sai Punith Godasi
Soham Chaudhari
Syed Mohd.Asim
Swati Kohli

Contents



1. Introduction



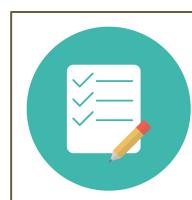
2. Description



3. Objective & Area Focus



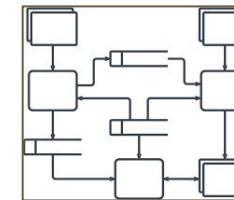
4. Entity & Attributes



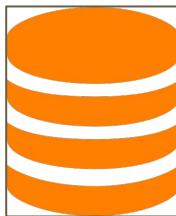
5. Business Rules



6. EERD



7. Normalizations & ERD



8. Database



9. Queries



10. Benefits



11. Challenges



Introduction

This is John and his family. They are ready to go on a vacation to San Francisco





John opens the Airbnb website and creates an account as a “Guest”.

John is allotted a UserID and a GuestID

John finds a beautiful 2 bedroom property in San Francisco with a kitchen and books the place for 3 nights at economical price.

John is given an OrderID corresponding to his guestID and the property's propertyID

John and his family loved the stay and left a great review and rating about the experience!

**A ReviewID is generated corresponding to John's review **



This is Cara. She owns a beautiful home with 2 bedroom and a kitchen in San Francisco that she wants to rent out.





Cara opens the Airbnb website and creates an account as a “Host”.

Cara is allotted a UserID and a HostID

Cara adds her property as a listing. Enters all the property features.

**Cara’s property is given a PropertyID **

Cara mentions the dates she would like to rent her property

**A SlotID is generated corresponding to each slot **



Organization Description

Business Sector:
Hospitality/ Travel Lodging

- Need:**
- Huge data repository
 - Requires DBMS
 - effective running and growth of the company



An Infographic from Fivizly

The Rise of Airbnb



founded in

2008[°]



number of countries

191



number of cities

65,000[°]



Airbnb Has More Listings than the Top 3 Hotel Chains Combined



Airbnb
3 Million Listings

Marriott
1.1 Million Rooms

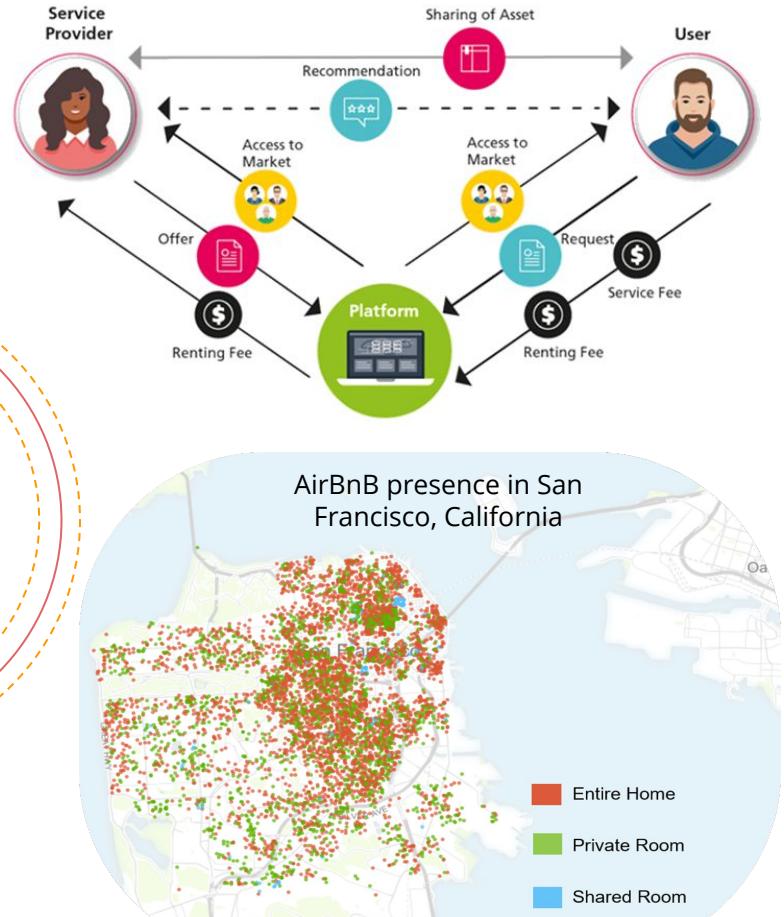
Hilton
773,000 Rooms

IHG
766,000 Rooms



Objective & Area Focus

1. Management of Property listing, Guest & Host
2. Illustrate utilization of RDBMS and Database for the same
3. Through Airbnb's perspective, enabling data-backed decisions:
 - a. **For Guests and Hosts** to choose interactions in both online and offline world.
 - b. **For organization** internally to track their periodic and annual performance.
4. Scalable Schema Design for a city - **San Francisco, California** to allow expansion across States and Countries





Entities and Attributes

	Entity- User Attributes-	A user is someone who creates an account on the Airbnb website. User ID, Name, Address, Date of Birth, Gender The attributes are general information about the User.
	Entity- Host Attributes-	Host is a type of User. A host is someone who can list their properties on the Airbnb website for the guests to book. Response Rate- Rate at which the host responds to the property booking Host Since - The date since the user has been a host. Host Verification - If the host has been verified or not Acceptance Rate - The rate of bookings the host accepts Is Super host - If the host is a super host i.e a status earned by a host through time and good reviews
	Entity- Guest Attributes-	Guest is a type of User. A guest is someone who can make a booking on the Airbnb website. Guest Rating- Rating received for each guest Guest verification- If the guest has been verified or not
	Entity - Review Attributes-	Review entity stores values specific to one guest and one property. A guest leaves a review for a specific property. Review date - Date of review Rating - Rating given by the guest - 1,2,3,4,5



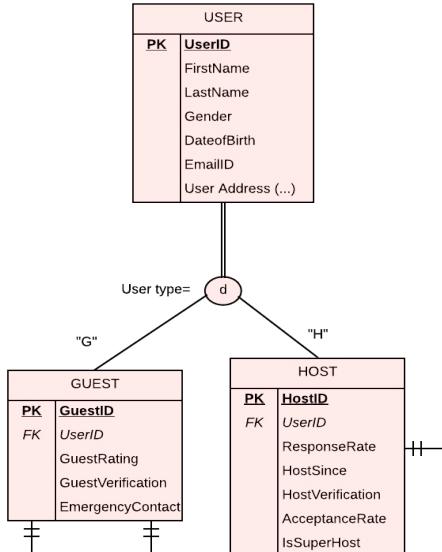
Entities and Attributes

A photograph of a bedroom featuring a large bed with white linens, two nightstands with lamps, and framed pictures on the wall.	Entity- Property Attributes-	<p>Each property is owned/managed by a single host. Properties are listings on the Airbnb website available for a guest to book.</p> <p>Space - Summary of the type of property</p> <p>Guests included- Number of guests in the published price</p> <p>Extra people rate- Extra charge per person</p> <p>Kitchen, pet friendly - If the property has a kitchen or is pet friendly respectively</p>
An illustration of a woman handing a red folder to a man at a desk, symbolizing a booking process.	Entity- Booking order Attributes-	<p>Booking entity stores values specific to a single booking made by a guest. When a guest makes a booking for a property, he receives a booking order ID</p> <p>Extra guests - Number of additional people that would be staying</p>
An illustration of a yellow calendar showing a month's worth of dates.	Entity- Calendar Slot Attributes-	<p>Calendar slot entity stores values specific to a property. Each property may have many slots available in a year. Each property available slot is identified in the calendar slot entity by the Slot ID.</p> <p>Available - "Y" for available, "N" for not available</p> <p>Adj price - Price after discounts</p>

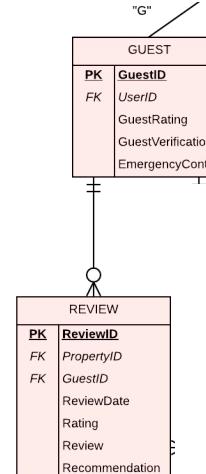


Business Rules

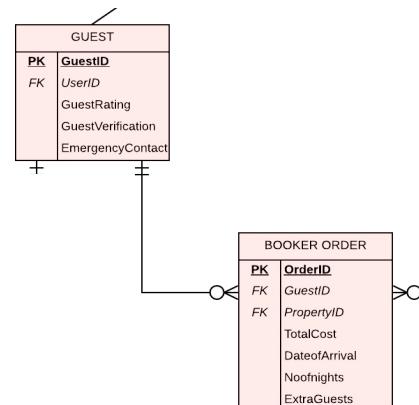
- A User may be a Guest.
- A Guest must be a registered User.
- A User may be a Host.
- A Host must be a User.



- A Guest can give a Review for the stay experience with respect to every booking order he/she makes.
- Each Review is given by a single Guest.



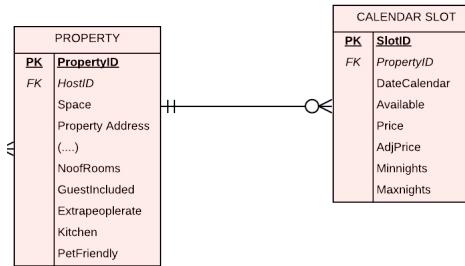
- Each Guest must have at least one Booking (or is only a User until then)
- Each Booking must correspond to only one Guest.



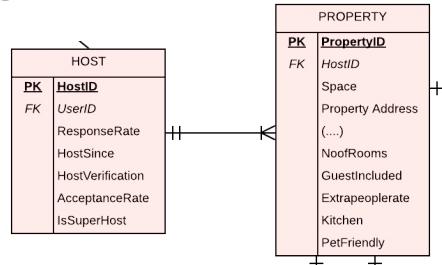


Business Rules

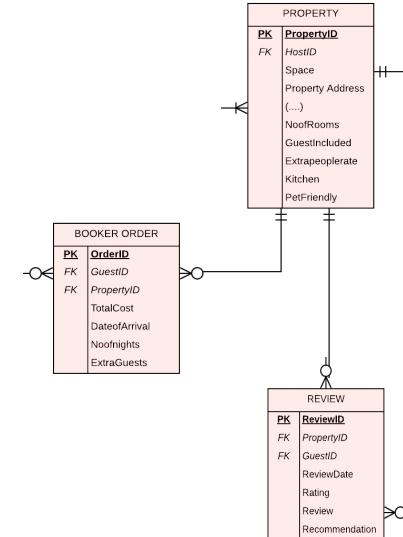
- A Property may have none or multiple Calendar_Slots of availability.
- A Calendar_Slot must correspond to one Property only.

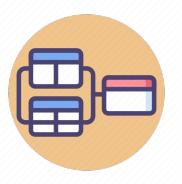


- A Host should have at least one Property listed (or is only a User until then).
- Each Property is associated with only one Host.



- A Property may or may not have a Booking. Also, a property can be booked multiple times.
- Each Booking corresponds to one Property only.
- A Property can have one or more Reviews.
- Each Review associates only with a single Property.





Data Table (Source) from AirBnB

Data Source: <http://insideairbnb.com/get-the-data.html>

Data Table for Host, Property, Bed

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	
1	property_id	host_id	host_name	host_since	host_location	host_response_rate	host_acceptance_rate	host_is_superhost	street	city	state	zipcode	country	property_type	bedrooms	beds	bed_type	price	guests_included	extra_people	minimum_nights	maximum_nights
2	958	1169	Holly	31-07-2008	San Francisco	86%	N/A	t	San Francisco	San Francisco	CA	94117	US	Apartment	1	2	Real Bed	\$170.00	2	\$25.00	1	30
3	3850	4921	Kevin	08-12-2008	San Francisco	100%	N/A	t	San Francisco	San Francisco	CA	94131	US	House	1	1	Real Bed	\$99.00	2	\$20.00	1	5
4	5858	8904	Philip And	02-03-2009	San Francisco	60%	N/A	f	San Francisco	San Francisco	CA	94110	US	Apartment	2	3	Real Bed	\$235.00	2	\$0.00	30	60
5	7918	21994	Aaron	17-06-2009	San Francisco	83%	N/A	t	San Francisco	San Francisco	CA	94117	US	Apartment	1	1	Real Bed	\$65.00	1	\$12.00	32	60
6	8142	21994	Aaron	17-06-2009	San Francisco	83%	N/A	t	San Francisco	San Francisco	CA	94117	US	Apartment	1	1	Real Bed	\$65.00	1	\$12.00	32	90

Data Table for Review

A	B	C	D	E	F	G	H
1	listing_id	id	date	reviewer_id	reviewer_name	comments	
2	958	5977	23-07-2009	15695	Edmund C	Our	
3	958	6660	03-08-2009	26145	Simon	Returning to San Francisco is a r	
4	958	11519	27-09-2009	25839	Denis	We were very pleased with the a	
5	958	16282	05-11-2009	33750	Anna	We highly recommend this acco	
6	958	26008	13-02-2010	15416	Venetia	Holly's place was great. It was e	

Data Table for Calendar Slot

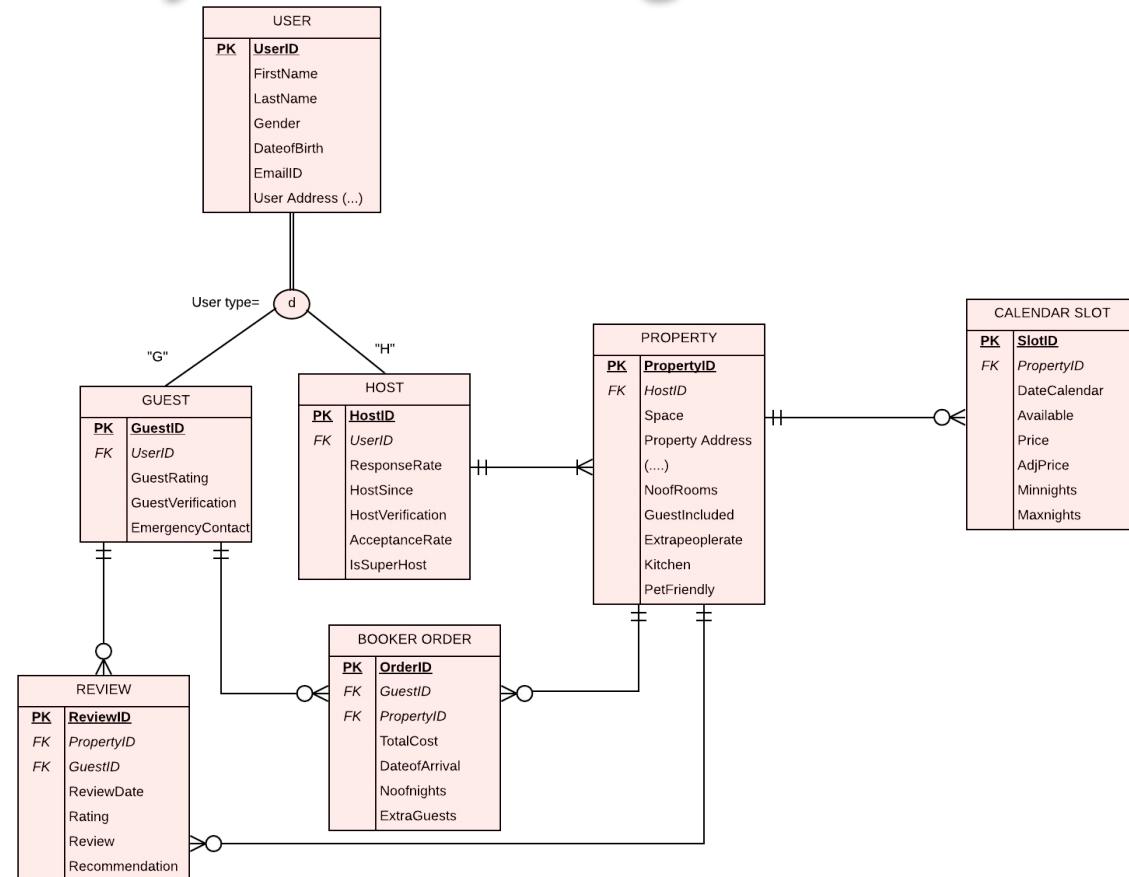
A	B	C	D	E	F	G	
1	listing_id	date	available	price	adjusted_price	minimum_nights	maximum_nights
2	958	04-12-2019	f	\$135.00	\$135.00	1	30
3	958	05-12-2019	f	\$135.00	\$135.00	1	30
4	958	06-12-2019	f	\$135.00	\$135.00	1	30
5	958	07-12-2019	f	\$135.00	\$135.00	1	30
6	958	08-12-2019	f	\$135.00	\$135.00	1	30

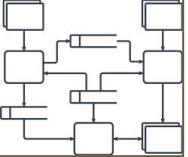


Enhanced Entity Relation Diagram

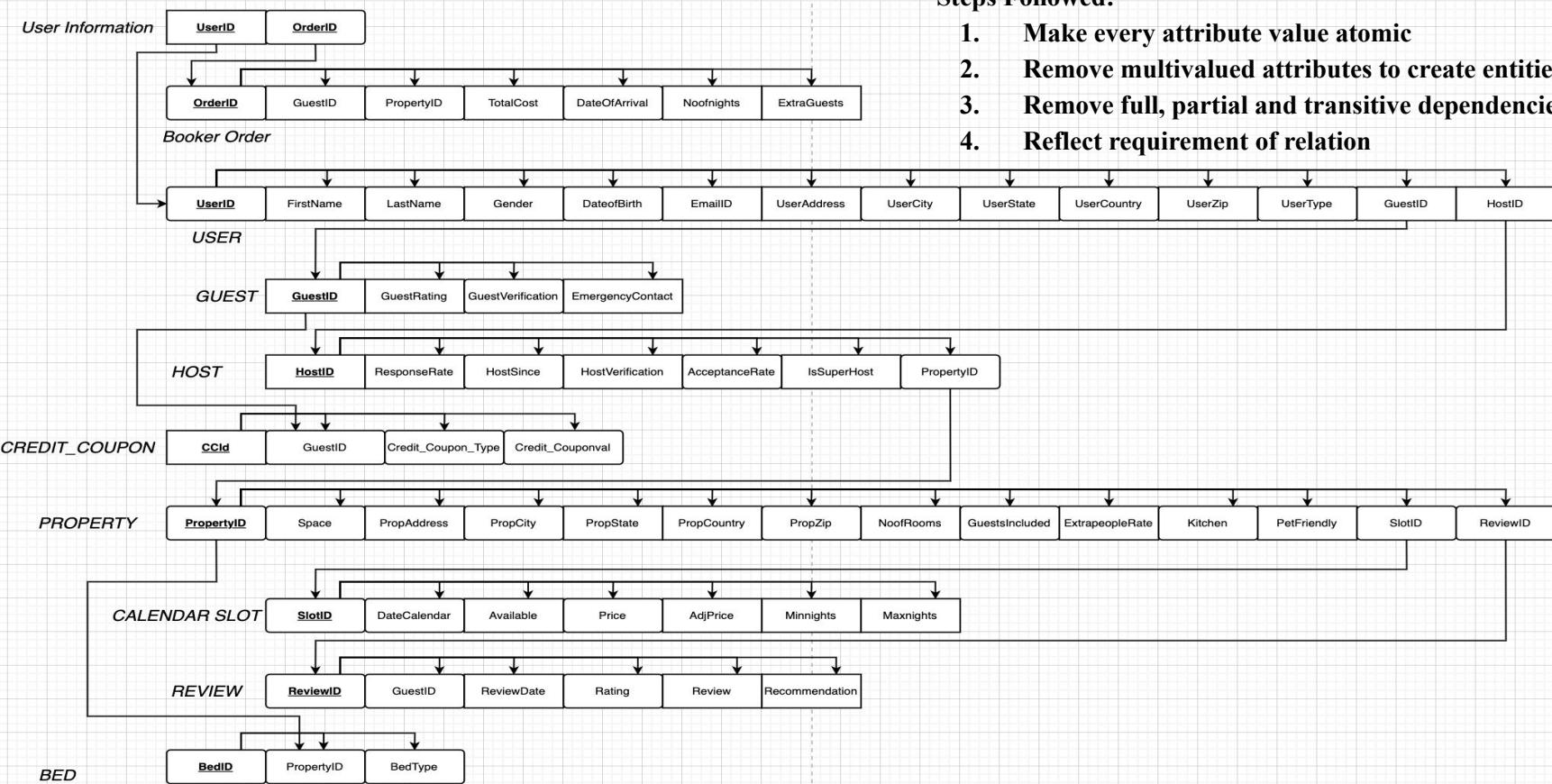
Through 2 steps:

1. Normalization to 1NF
2. Study research to add other Entities and Data required for the project



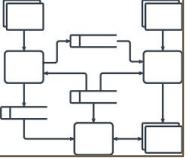


A complete 3NF relational model (Also as 2NF)

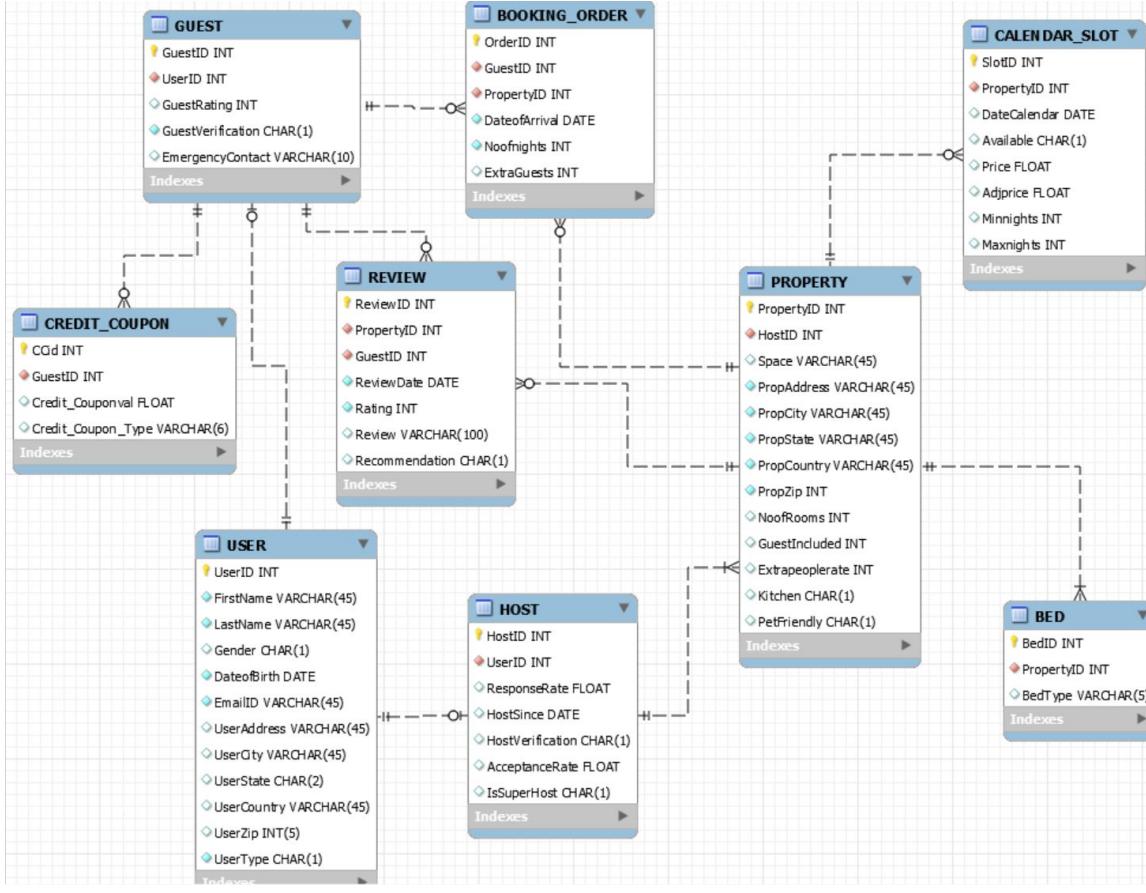


Steps Followed:

1. Make every attribute value atomic
2. Remove multivalued attributes to create entities
3. Remove full, partial and transitive dependencies
4. Reflect requirement of relation



Entity Relation Diagram (from 3NF)





Database Tables

Total of nine tables in the database - 7 main and 2 multivalued.

Screenshot of some records:

1. User

UserID	FirstName	LastName	Gender	DateofBirth	EmailID	UserAddress	UserZip	UserType
101	Mitzi	Clein	F	1987-05-01	mz12@gmail.com	123 Floribunda Ave	94016	G
102	Renu	Chopra	F	1990-10-10	choprar@gmail.com	123 abc	94088	G

2. Host

HostID	UserID	ResponseRate	HostSince	HostVerification	AcceptanceRate	IsSuperHost
301	111	0.86	2008-07-31	N	0.8	Y
302	112	1	2008-12-08	Y	1	N

3. Guest

GuestID	UserID	GuestRating	GuestVerification	EmergencyContact
201	101	4	Y	5105331221
202	102	3	Y	9167796688

4. Review

ReviewID	PropertyID	GuestID	ReviewDate	Rating	Review	Recommendation
601	401	201	2019-01-02	3	stay was okish	N
602	401	202	2019-01-06	4	Fantastic and highly recommend!!	Y



Database Tables

Total of nine tables in the database - 7 main and 2 multivalued.

Screenshot of some records:

5. Property

PropertyID	HostID	SpaceType	PropAddress
401	301	Apartment	121 Alma St
402	302	House	Berkeley Way

NoofRooms	GuestIncluded	ExtrapeopleRate	Kitchen	PetFriendly
1	2	25	NULL	Y
2	2	20	Y	Y

6. Bed (Multivalued)

BedID	PropertyID	BedType
801	401	King
802	402	Twin

7. Calender_Slot

SlotID	PropertyID	DateCalendar	Available	Price	Adjprice	Minnights	Maxnights
701	401	2019-01-01	Y	135	130	1	30
702	401	2019-03-01	Y	148	148	1	30

8. Booking_Order

OrderID	GuestID	PropertyID	BookingDate	DateofArrival	Noofnights	ExtraGuests
501	201	401	2018-12-01	2018-12-15	3	1
502	203	402	2019-05-05	2019-06-28	2	1

9. Credit_Coupon (Multivalued)

CCid	GuestID	Credit_Couponval	Credit_Coupon_Type
901	201	50	Credit
902	202	15	Coupon



Queries

Sequel SELECT queries using various elements.

Specifically, the elements covered are:

1. **Select Clauses:** SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY
2. **Functions** - COUNT, ROUND, AVERAGE, SUM, DATE_FORMAT
3. **Boolean Operators:** AND , NOT, LIKE
4. **Comparison Operators:** =, >, <
5. **Multiple Tables Queries:** INNER JOIN, ON,
6. **SubQuery :** IN, NOT IN, NOT EXISTS, Derived table, UNION,
7. **Conditional:** CASE (WHEN,THEN) with Derived Table



Queries #1

Purpose of Query:

Find the percentage availability of the properties in 2019.

Commands Used:

COUNT, ROUND, DATE_FORMAT, GROUP BY, SUB QUERY

Output:

	Property_ID	Availability_Rate
▶	401	25.00
	402	25.00
	403	37.50
	404	12.50

SQL Code:

```
SELECT b.Property_ID, ROUND((b.counts/a.total)*100,2)
AS Availability_Rate FROM
(SELECT calendar_slot.propertyid AS Property_ID, COUNT(calendar_slot.slotid)
AS counts FROM calendar_slot
WHERE calendar_slot.available='Y' AND
CALENDAR_SLOT.DateCalendar> DATE_FORMAT('2018-12-31','%y-%m-%d')
AND CALENDAR_SLOT.DateCalendar < DATE_FORMAT('2020-01-01','%y-%m-%d')
GROUP BY calendar_slot.PropertyID)b,
(SELECT calendar_slot.propertyid AS p2, COUNT(calendar_slot.slotid)
AS total FROM calendar_slot
WHERE calendar_slot.available='Y'
AND CALENDAR_SLOT.DateCalendar> DATE_FORMAT('2018-12-31','%y-%m-%d')
AND CALENDAR_SLOT.DateCalendar < DATE_FORMAT('2020-01-01','%y-%m-%d'))a;
```



Queries #2

Purpose of Query:

List the available properties from cheapest to most expensive in San Francisco

Commands Used:

- ORDER BY
- LIKE

Output:

PropertyID	Price	PropCity
402	48	San Francisco
402	77	San Francisco
403	129	San Francisco
403	129	San Francisco
401	135	San Francisco
401	148	San Francisco
404	194	San Francisco
403	196	San Francisco

SQL Code:

```
SELECT Calendar_slot.PropertyID, CALENDAR_SLOT.Price, PROPERTY.PropCity
FROM CALENDAR_SLOT, Property
WHERE CALENDAR_SLOT.Available="Y"
AND PROPERTY.PropertyID=CALENDAR_SLOT.PropertyID
AND PROPERTY.PropCity LIKE '%San Francisco%'
ORDER BY CALENDAR_SLOT.PRICE ;
```



Queries #3

Purpose of Query:

Find the top hot properties (property IDs) in San Francisco, their original price and associated Host ID, name and IsSuperhost based on the average review rating of that property to be greater than 3. Display their average rating as well.

Commands Used:

Conditions, GROUP BY, AVERAGE, HAVING

Output:

SQL Code:

```
SELECT p.PropertyID, h.HostID, u.FirstName, u.LastName,  
h.IsSuperhost, AVG(r.Rating), c.Price  
FROM USER u, review r, HOST h, property p, calender_slot c  
WHERE p.HostID = h.HostID  
AND p.PropCity = 'San Francisco'  
AND h.UserID = u.UserID  
AND r.PropertyID = p.PropertyID  
AND p.PropertyID = c.PropertyID  
GROUP BY r.PropertyID  
Having AVG (r.Rating) > 3;
```

Filter Rows:		Export:		Wrap Cell Content:	
HostID	FirstName	LastName	IsSuperhost	AVG(r.Rating)	Price
301	Holly	Lopez	Y	3.5000	135
303	Ivan & Wendy	Baker	Y	3.6667	129



Queries #4

Purpose of Query:

List properties in an area of San Francisco that have at least one King Bed, come with a kitchen and are Pet friendly.

Commands Used:

INNER JOIN, LIKE

Output:

Result Grid					Filter Rows:	Export:
	PropertyID	Kitchen	PetFriendly	BedType	PropAddress	
▶	403	Y	Y	King	22nd st	

SQL Code:

```
SELECT PROPERTY.PropertyID, PROPERTY.Kitchen,  
PROPERTY.PetFriendly, BED.BedType,  
PROPERTY.PropAddress FROM PROPERTY  
INNER JOIN BED  
WHERE PROPERTY.PropertyID = BED.PropertyID  
AND PROPERTY.PropAddress LIKE '%22nd st%'  
AND PROPERTY.Kitchen = 'Y'  
AND PROPERTY.PetFriendly = 'Y'  
AND BED.BedType LIKE '%King%';
```



Queries #5

Purpose of Query:

Display the properties in the area of San Francisco that are 2 bedroom houses with a rating of 3 or above.

Commands Used:

ON, INNER JOIN, LIKE

Output:

SQL Code:

```
Use airbnb;
SELECT PROPERTY.PropertyID, PROPERTY.NoofRooms, PROPERTY.SpaceType,
REVIEW.RATING, PROPERTY.PropAddress, PROPERTY.PropCity, PROPERTY.PropState
FROM PROPERTY INNER JOIN REVIEW
ON PROPERTY.PropertyID = REVIEW.PropertyID
WHERE PROPERTY.SpaceType LIKE 'House'
AND PROPERTY.NoofRooms = 2
AND REVIEW.RATING = 3;
```

	PropertyID	NoofRooms	SpaceType	RATING	PropAddress	PropCity	PropState
▶	402	2	House	3	Berkeley Way	San Francisco	CA
	402	2	House	3	Berkeley Way	San Francisco	CA



Queries #6

Purpose of Query:

Find the total value of credit and coupon owned by each guest.

Commands Used:

NOT EXISTS, UNION, SUB QUERY, SUM

Output:

GuestID	TotalValue
201	50
202	15
203	15
204	10
205	25.34999990463257
206	0
207	30
208	150.5
209	100
210	0

SQL Code:

```
SELECT GuestID, SUM(Credit_couponval)as TotalValue
FROM CREDIT_COUPON
GROUP BY GuestID
UNION
SELECT GuestID, 0
FROM GUEST
WHERE NOT EXISTS
(SELECT * FROM CREDIT_COUPON
WHERE CREDIT_COUPON.GuestID = Guest.GuestID);
```



Queries

#7

Purpose of Query:

Search based query to list properties that have never been booked by a guest

Commands Used:

NOT IN, SUB QUERY

Output:

	PropertyID
▶	405
	406
*	NULL

SQL Code:

```
SELECT PropertyID FROM PROPERTY  
WHERE PROPERTY.PropertyID  
NOT IN (SELECT PropertyID from BOOKING_ORDER);
```



Queries

#8

Purpose of Query:

List out the full name, Guest ID and total sum of credit and coupon values of the guests that have both credit and coupon values in their account.

Commands Used:

IN, SUB QUERY, GROUP BY, SUM, CASE
(When, Then), HAVING

Output:

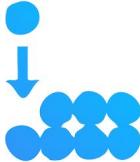
Result Grid				
	firstname	lastname	guestid	sum(Credit_Couponval)
▶	Tyler	Jones	205	25.34999990463257
	Steve	Williams	208	150.5

SQL Code:

```
SELECT u.firstname, u.lastname, g.guestid, sum(Credit_Couponval)
FROM credit_coupon c, guest g, User u
WHERE g.guestid = c.guestid
AND g.userid = u.userid
AND g.guestid IN
(SELECT c.guestid
FROM Credit_Coupon c
GROUP BY c.guestid
HAVING SUM(CASE WHEN c.Credit_Coupon_Type = 'Credit' THEN 1
WHEN c.Credit_Coupon_Type = 'Coupon' THEN 1
END) = 2)
group by c.guestid;
```



Benefits



Self Describing nature of RDBMS because metadata defines and describes the data and the relationships define between tables in the database.

Map a business through this information and database.



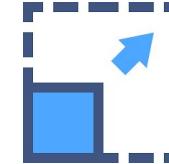
RDBMS provides data **Security, Privacy, Consistency and Backup and recovery.**

Maintain data integrity by adding restriction or rule to dictate information entered or edited in a table such as phone number is 10 digit long.



Prevention of data redundancy or duplication. A change is reflected immediately.

Data independence because metadata are separated from the application programs



Capacity to be a **scalable** database system design for a growing business especially across countries.



Challenges



Project Approach

Humongous AirBnB data
of myriad countries
top down or bottom up
approach?



Data Integrity

Data entry is error prone
and incorporating Data
Integrity to avoid conflict
or overlap was a
challenge.



Finding Meaningful Relationships

Figure effective relations for
accurate data selection through
queries which sets the foundation
for effective Data Analysis

Solution

EERD and a scalable
approach

Solution

Incorporated checks in
metadata descriptions

Solution

Brainstorming for ideas,
Initial EERD and in turn
normalization steps.

THANK YOU!
