

Star Management Company

I. Executive Summary

Star Management Company is a property management business that supervises Airbnb properties. Recently, there was a concerning rise in negative reviews for some of the properties. The reviews highlighted potential issues impacting guest satisfaction threatening reputation and profitability. In response to the problem, Star Management Company created and managed a database to keep track of reviews and important information such as booking information, property features, policies, and more to identify the reason for dissatisfaction.

Through ongoing data analysis, several key factors impacted the negative reviews. One of them was delayed responses to maintenance on the properties. Earlier this year, property guests expressed dissatisfaction with the heater not working. The issue was not addressed in a timely manner. The delayed response between management and staff negatively impacted their experience. The company implemented a system to ensure timely communication between the guests, managers, and staff.

Another factor that negatively impacted guest satisfaction was the outdated property photos. A customer expressed disappointment upon reading the property description and photos presented. The guest expected an updated living room and an accessible pool upon arrival, different from the listing. Once the issue was identified, the company established a policy where the photos and descriptions were regularly updated monthly for up-to-date and accurate listings.

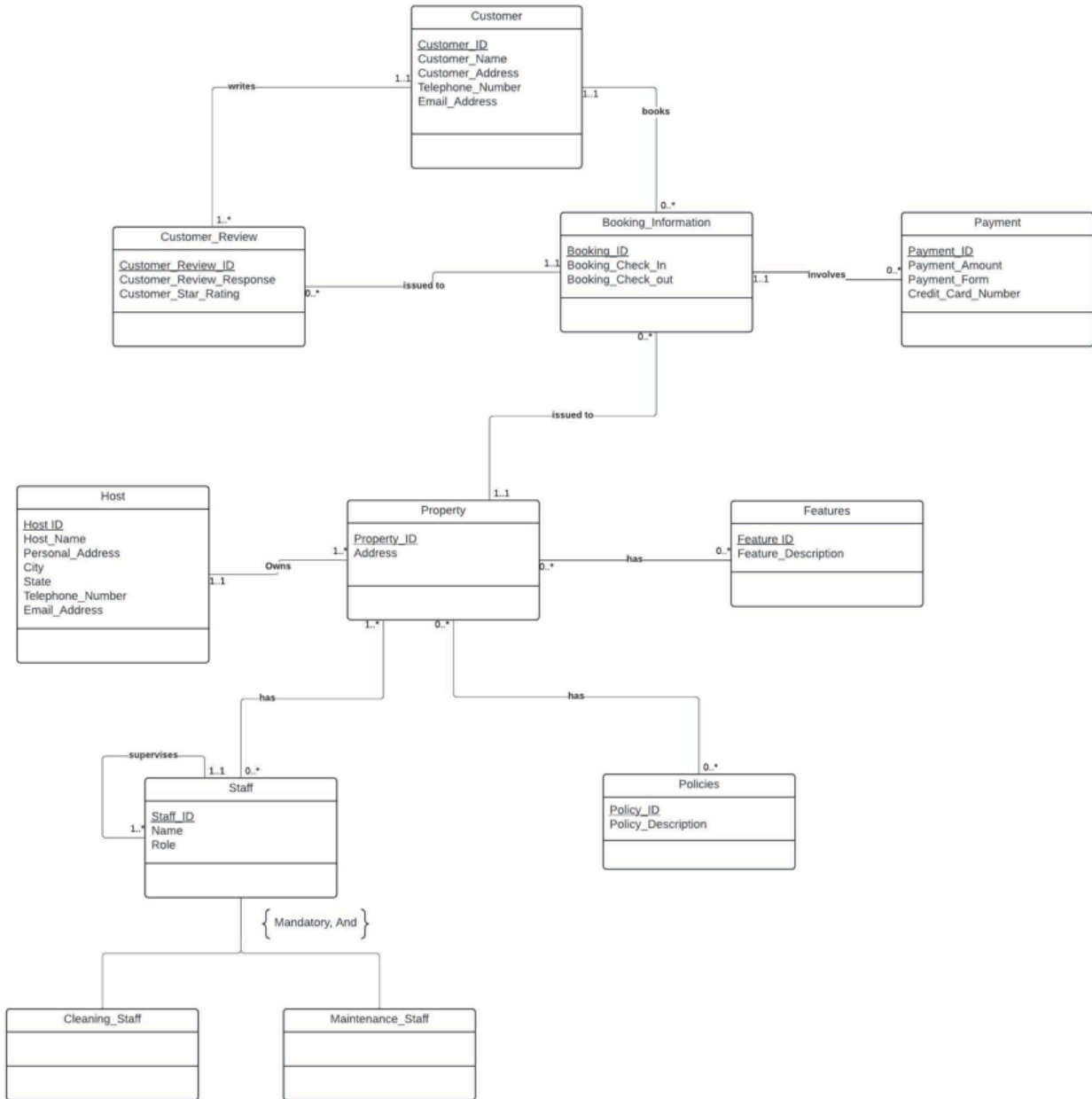
Star Management Company remains committed to identifying the root causes of the problem. The company takes pride in improving company efficiency through data-driven solutions and database management. With these issues in hand, the database system will be regularly maintained to ensure guest satisfaction and property profitability.

II. Introduction

Star Management Company, a property management business, supervises over 150 Airbnb properties ranging from apartments to villas. The company has coordinated with Airbnb hosts and has managed various Airbnb properties along the East Coast of the United States for over five years. Over time, the company has noticed an increase in negative reviews for some of its Airbnb properties. The company is fearful that these negative reviews will discourage potential and returning customers from booking stays with them in the future. To mitigate this issue, Star Management will create an exhaustive database tracking relevant information regarding customers, bookings, hosts, properties, and staff. The company hopes to gain insights by identifying patterns among the lowest-rated customer reviews to tackle this challenge. Ultimately by detecting the major issues in their current properties and customer experience, Star Management will ensure that future customers receive streamlined and positive experiences at all of their Airbnb properties.

III. ERD Model

Airbnb Property Management Company
ERD



Relationship Sentences:

Customer-Booking_Information (1:*)

One customer books 0 or many booking_information.

One booking_information belongs to 1 customer.

Customer-Customer_Review (1:*)

One customer writes 1 or many customer reviews.

One customer_review belongs to 1 customer.

Customer_Review-Booking_Information (1:*)

One customer_review is issued to 1 booking_information.

One booking_information has 0 or many customer_reviews.

Booking_Information-Payment (1:*)

One booking_information involves 0 or many payments.

One payment is associated with 1 booking_information.

Property-Booking_Information (1:*)

One property has 0 or many booking_information.

One booking_information is issued to 1 property.

Property-Host (1:*)

One host owns 1 or many properties.

One property is owned by 1 host.

Property-Features (*:*)

One property has 0 or many features.

One feature belongs to 0 or many properties.

Property-Staff (1:*)

One property has 0 or many staff.

One staff belongs to zero or many properties.

Property-Policies (*:*)

One property has 0 or many policies.

One policy belongs to 0 or many properties.

Supervisor-Staff (1:*)

One supervisor can supervise 1 or many staff.

One staff can be supervised by 1 supervisor.

IV.RDM

Host(Host_ID, Host_Name, Personal_Address, City, State, Telephone_Number, Email_Address)

Property(Property_ID, Address, Host_ID (fk))

Features(Feature_ID, Feature_Description)

Property_Features(Property_ID, Features_ID)

Policies(Policy_ID, Policy_Description)

Property_Policies(Property_ID, Policy_ID)

Staff(Staff_ID, Name, Role, Type_of_staff, parent_staff_id (fk))

Property_Staff(Property_ID, Staff_ID, Date_Serviced, Work_Performed)

Booking_Information(Booking_ID, Booking_Check_In, Booking_Check_Out, Property_ID (fk), Customer_ID (fk))

Payment(Payment_ID, Payment_Amount, Payment_Form, Credit_Card_Number, Booking_ID (fk))

Customer (Customer_ID, Customer_Name, Customer_Address, Telephone_Number, Email_Address)

Customer_Review(Customer_Review_ID, Customer_Review_Response, Customer_Star_Rating, Customer_ID (fk), Booking_ID (fk))

V. Normalization

Host (Host_ID, Host_Name, Personal_Address, City, State, Telephone_Number, Email_Address)

- Step 1: First Normal Form (1NF): Do we have a primary key? YES, 1NF
 - FD1: Host_ID → Host_Name, Personal_Address, City, State Telephone_Number, Email_Address
- Step 2: Second Normal Form (2NF): Do we have any primary-key dependencies?
 - No, 2NF
- Step 3: Third Normal Form (3NF): Do we have any transitive dependencies?
 - No, 3NF
- Step 4: Do we have any non-key attributes as determinants for key attributes?
 - No, BCNF
- Final Relation: Host(Host_ID, Host_Name, Personal_Address, Telephone_Number, Email_Address)

Property (Property_ID, Address, Host_ID (fk))

- Step 1: First Normal Form (1NF): Do we have the primary key? YES, 1NF
 - FD1: Property_ID → Address, Host_ID
- Step 2: Second Normal Form (2NF): Do we have any primary-key dependencies?
 - No, 2NF

- Step 3: Third Normal Form (3NF): Do we have any transitive dependencies?
 - No, 3NF
- Step 4: Do we have any non-key attributes as determinants for key attributes?
 - No, BCNF
- Final Relation: Property(Property_ID, Address, Host_ID (fk))

Features (Feature_ID, Feature_Description)

- Step 1: First Normal Form (1NF): Do we have the primary key? YES, 1NF
 - FD1: Feature_ID → Feature_Description
- Step 2: Second Normal Form (2NF): Do we have any primary-key dependencies?
 - No, 2NF
- Step 3: Third Normal Form (3NF): Do we have any transitive dependencies?
 - No, 3NF
- Step 4: Do we have any non-key attributes as determinants for key attributes?
 - No, BCNF
- Final Relation: Features(Feature_ID, Feature_Description)

Property_Features (Property_ID, Features_ID)

- Step 1: First Normal Form (1NF): Do we have the primary key? YES, 1NF
- Step 2: Second Normal Form (2NF): Do we have any primary-key dependencies?
 - No, 2NF
- Step 3: Third Normal Form (3NF): Do we have any transitive dependencies?
 - No, 3NF
- Step 4: Do we have any non-key attributes as determinants for key attributes?
 - No, BCNF
- Final Relation: Property_Features(Property_ID, Features_ID)

Policies (Policy_ID, Policy_Description)

- Step 1: First Normal Form (1NF): Do we have a primary key? YES, 1NF
 - FD1: Policy_ID → Policy_Description
- Step 2: Second Normal Form (2NF): Do we have any primary-key dependencies?
 - No, 2NF
- Step 3: Third Normal Form (3NF): Do we have any transitive dependencies?
 - No, 3NF
- Step 4: Do we have any non-key attributes as determinants for key attributes?
 - No, BCNF
- Final Relation: Policies(Policy_ID, Policy_Description)

Property_Policies (Property_ID, Policy_ID)

- Step 1: First Normal Form (1NF): Do we have the primary key? YES, 1NF
 - Property_ID, Policy_ID
- Step 2: Second Normal Form (2NF): Do we have any primary-key dependencies?
 - No
- Step 3: Third Normal Form (3NF): Do we have any transitive dependencies?
 - No
- Step 4: Do we have any non-key attributes as determinants for key attributes?
 - No, BCNF
- Final Relation: Property_Policies(Property_ID, Policy_ID)

Staff (Staff_ID, Name, Role, Type_of_staff, parent_staff_id (fk))

- Step 1: First Normal Form (1NF): Do we have the primary key? YES, 1NF
 - Staff_ID → Name, Role, Type_of_Staff, parent_staff_id
- Step 2: Second Normal Form (2NF): Do we have any primary-key dependencies?
 - No
- Step 3: Third Normal Form (3NF): Do we have any transitive dependencies?
 - No
- Step 4: Do we have any non-key attributes as determinants for key attributes?
 - No, BCNF
- Final Relation: Cleaning_Staff(Staff_ID, Name, Role, Type_of_staff, parent_staff_id (fk))

Property_Staff (Property_ID, Staff_ID, Date_serviced, Work_Performed)

- Step 1: First Normal Form (1NF): Do we have a primary key? YES, 1NF
 - FD1: Property_ID, Staff_ID → Date_serviced, Work_Performed
- Step 2: Second Normal Form (2NF): Do we have any primary-key dependencies?
 - No, 2NF
- Step 3: Third Normal Form (3NF): Do we have any transitive dependencies?
 - No, 3NF
- Step 4: Do we have any non-key attributes as determinants for key attributes?
 - No, BCNF
- Final Relation: Property_staff (Property_ID, Staff_ID, Date_serviced, Work_Performed)

Booking_Information (Booking_ID, Booking_Check_In, Booking_Check_Out, Property_ID (fk), Customer_ID (fk))

- Step 1: First Normal Form (1NF): Do we have the primary key? YES, 1NF
 - FD1: Booking_ID → Booking_Check_In, Booking_Check_Out, Property_ID, Customer_ID
- Step 2: Second Normal Form (2NF): Do we have any primary-key dependencies?

- No, 2NF
- Step 3: Third Normal Form (3NF): Do we have any transitive dependencies?
 - No, 3NF
- Step 4: Do we have any non-key attributes as determinants for key attributes?
 - No, BCNF
- Final Relation: Booking_Information(Booking_ID, Booking_Check_In, Booking_Check_Out, Property_ID (fk), Customer_ID (fk))

Payment (Payment_ID, Payment_Amount, Payment_Form, Credit_Card_Number, Booking_ID (fk))

- Step 1: First Normal Form (1NF): Do we have a primary key? YES, 1NF
 - FD1: Payment_ID → Payment_Amount, Payment_Form, Credit_Card_Number, Booking_ID
- Step 2: Second Normal Form (2NF): Do we have any primary-key dependencies?
 - No, 2NF
- Step 3: Third Normal Form (3NF): Do we have any transitive dependencies?
 - No, 3NF
- Step 4: Do we have any non-key attributes as determinants for key attributes?
 - No, BCNF
- Final Relation: Payment(Payment_ID, Payment_Amount, Payment_Form, Credit_Card_Number, Booking_ID (fk))

Customer (Customer_ID, Customer_Name, Customer_Address, Telephone_Number, Email_Address)

- Step 1: First Normal Form (1NF): Do we have a primary key? YES, 1NF
 - FD1: Customer_ID → Customer_Name, Customer_Address, Telephone_Number, Email_Address
- Step 2: Second Normal Form (2NF): Do we have any primary-key dependencies?
 - No, 2NF
- Step 3: Third Normal Form (3NF): Do we have any transitive dependencies?
 - No, 3NF
- Step 4: Do we have any non-key attributes as determinants for key attributes?
 - No, BCNF
- Final Relation: Customer(Customer_ID, Customer_Name, Customer_Address, Telephone_Number, Email_Address)

Customer_Review (Customer_Review_ID, Customer_Review_Response, Customer_Star_Rating, Customer_ID (fk), Booking_ID (fk))

- Step 1: First Normal Form (1NF): Do we have a primary key? YES, 1NF

- FD1: Customer_Review_ID → Customer_Review_Response, Customer_Star_Rating, Customer_ID, Booking_ID
- Step 2: Second Normal Form (2NF): Do we have any primary-key dependencies?
 - No, 2NF
- Step 3: Third Normal Form (3NF): Do we have any transitive dependencies?
 - No, 3NF
- Step 4: Do we have any non-key attributes as determinants for key attributes?
 - No, BCNF
- Final Relation: Customer_Review(Customer_Review_ID, Customer_Review_Response, Customer_Star_Rating, Customer_ID (fk), Booking_ID (fk))

VI. SQL DDL

Host

```
CREATE TABLE Host (  
Host_ID NUMBER NOT NULL,  
Host_Name VARCHAR(50),  
Personal_Address VARCHAR(60),  
City VARCHAR(60),  
State VARCHAR(2),  
Telephone_Number NUMBER,  
Email_Address VARCHAR(50) ,  
CONSTRAINT pk_Host PRIMARY KEY (Host_ID)  
)  
  
INSERT INTO Host VALUES (100, "Anna Tee", "123 Personal St", "Patterson", "NJ", 1234567890,  
"a.tee@gmail.com")  
  
INSERT INTO Host VALUES (200, "Ben All", "234 Personal St", "New York", "NY", 2345678901,  
"b.all@gmail.com")
```

Property

```
CREATE TABLE Property (  
Property_ID NUMBER NOT NULL,  
Address VARCHAR(60),  
Host_ID NUMBER,  
CONSTRAINT pk_Property PRIMARY KEY (Property_ID),  
CONSTRAINT fk_Property FOREIGN KEY (Host_ID) REFERENCES Host (Host_ID)  
)  
  
INSERT INTO Property VALUES (1, "123 Property St", 100)  
  
INSERT INTO Property VALUES (2, "234 Property St", 200)
```

Features

```
CREATE TABLE Features (  

```

```
Feature_ID NUMBER NOT NULL,  
Feature_Description VARCHAR(60),  
CONSTRAINT pk_Features PRIMARY KEY (Feature_ID)  
  
)  
  
INSERT INTO Features VALUES (1, "Outdoor Pool")  
  
INSERT INTO Features VALUES (2, "Hot Tub Jacuzzi")
```

Property_Features

```
CREATE TABLE Property_Features (  
Property_ID NUMBER NOT NULL,  
Feature_ID NUMBER NOT NULL,  
CONSTRAINT pk_Property_Features PRIMARY KEY (Property_ID, Features_ID),  
CONSTRAINT fk_Property_Features FOREIGN KEY (Property_ID) REFERENCES Property (Property_ID),  
CONSTRAINT fk_Property_Features_One FOREIGN KEY (Feature_ID) REFERENCES Features (Feature_ID)  
  
)  
  
INSERT INTO Property_Features VALUES (1, 1)  
  
INSERT INTO Property_Features VALUES (1, 3)
```

Policies

```
CREATE TABLE Policies (  
Policy_ID NUMBER NOT NULL,  
Policy_Description VARCHAR(100),  
CONSTRAINT pk_Policies PRIMARY KEY (Policy_ID)  
  
)  
  
INSERT INTO Policies VALUES (1, "Property doesn't allow smoking indoors")  
  
INSERT INTO Policies VALUES (2, "Pool area doesn't allow diving")
```

Property_Policies

```
CREATE TABLE Property_Policies (  
Property_ID NUMBER NOT NULL,
```

```

Policy_ID NUMBER NOT NULL,

CONSTRAINT pk_Property_Policies PRIMARY KEY (Property_ID, Policy_ID),

CONSTRAINT fk_Property_Policies FOREIGN KEY (Property_ID) REFERENCES Property (Property_ID),

CONSTRAINT fk_Property_Policies_One FOREIGN KEY (Policy_ID) REFERENCES Policies (Policy_ID)

)

INSERT INTO Property_Policies VALUES (1, 2)

INSERT INTO Property_Policies VALUES (2, 3)

```

Staff

```

CREATE TABLE Staff (

Staff_ID NUMBER NOT NULL,

Name VARCHAR(50),

Role VARCHAR(50),

Type_of_staff VARCHAR(50),

Parent_staff_id NUMBER,

CONSTRAINT pk_Staff PRIMARY KEY (Staff_ID)

)

INSERT INTO Staff VALUES (1, "Anna Benjamin", "Housekeeper", "Cleaning Staff", 11)

INSERT INTO Staff VALUES (2, "Blake Carter", "Inspector", "Cleaning Staff", NULL)

```

Property_Staff

```

CREATE TABLE Property_Staff (

Property_ID NUMBER NOT NULL,

Staff_ID NUMBER NOT NULL,

Date_serviced DATE,

Work_Performed VARCHAR(100),

CONSTRAINT pk_Property_Staff PRIMARY KEY (Property_ID, Staff_ID),

```

CONSTRAINT fk_Property_Staff **FOREIGN KEY** (Property_ID) **REFERENCES** Property (Property_ID)

CONSTRAINT fk_Property_Staff_One **FOREIGN KEY** (Staff_ID) **REFERENCES** Staff (Staff_ID)

)

INSERT INTO Property_Staff **VALUES** (1, 1, 3/5/2024, "Changed linens and removed any leftover garbage.")

INSERT INTO Property_Staff **VALUES** (2, 3, 6/14/2024, "Vacuum and mopped down floors.")

Customer

CREATE TABLE Customer (

Customer_ID **NUMBER NOT NULL**,

Customer_Name **VARCHAR(50)**,

Customer_Address **VARCHAR(250)**,

Telephone_Number **NUMBER**,

Email_Address **VARCHAR(50)**,

CONSTRAINT pk_Customer **PRIMARY KEY** (Customer_ID)

)

INSERT INTO Customer **VALUES** (1001, "Aaron Bennett", "123 Whale Street", 9876543210, "anna.b@gmail.com")

INSERT INTO Customer **VALUES** (1002, "Brayden Collins", "234 Sea Avenue", 8765432109, "brayden.c@gmail.com")

Booking_Information

CREATE TABLE Booking_Information (

Booking_ID **NUMBER NOT NULL**,

Booking_Check_In **DATE**,

Booking_Check_Out **DATE**,

Property_ID **NUMBER**,

Customer_ID **NUMBER**,

CONSTRAINT pk_Booking_Information **PRIMARY KEY** (Booking_ID),

CONSTRAINT fk_Booking_Information **FOREIGN KEY** (Property_ID) **REFERENCES** Property (Property_ID),

CONSTRAINT fk_Booking_Information_One **FOREIGN KEY** (Customer_ID) **REFERENCES** Customer (Customer_ID)

)

INSERT INTO **Booking_Information** VALUES (101, 2/10/2024, 2/12/2024, 3, 1001)

INSERT INTO **Booking_Information** VALUES (121, 3/10/2024, 3/17/2024, 10, 1002)

Payment

CREATE TABLE Payment (

Payment_ID **NUMBER NOT NULL**,

Payment_Amount **NUMBER**,

Payment_Form **VARCHAR(20)**,

Credit_Card_Number **NUMBER**,

Booking_ID **NUMBER**,

CONSTRAINT pk_Payment **PRIMARY KEY** (Payment_ID),

CONSTRAINT fk_Payment **FOREIGN KEY** (Booking_ID) **REFERENCES** Booking_Information (Booking_ID)

)

INSERT INTO Payment VALUES (1, 150, "Credit Card", 378282246310005, 123)

INSERT INTO Payment VALUES (2, 200, "Debit Card", 4147872511115623, 456)

Customer_Review

CREATE TABLE Customer_Review (

Customer_Review_ID **NUMBER NOT NULL**,

Customer_Review_Response **VARCHAR(100)**,

Customer_Star_Rating **NUMBER**,

Customer_ID **NUMBER NOT NULL**,

Booking_ID **NUMBER NOT NULL**,

CONSTRAINT pk_Customer_Review **PRIMARY KEY** (Customer_Review_ID),

CONSTRAINT fk_Customer_Review **FOREIGN KEY** (Customer_ID) **REFERENCES** Customer (Customer_ID),

CONSTRAINT fk_Customer_Review_One **FOREIGN KEY** (Booking_ID) **REFERENCES** Booking_Information (Booking_ID)

)

INSERT INTO Customer_Review **VALUES** (1, "Private garden was a nice touch, but the rest of the stay did not meet my standards. Space was very cluttered and not maintained.", 1, 1003, 123)

INSERT INTO Customer_Review **VALUES** (2, "Perfect location, but we had a lot of issues with fireplace.", 2, 1009, 456)

VI. Scenarios

Scenario 1: Retrieve Supervisor Information

Query: Retrieve the names of all the employees that have supervisors and their supervisor's name from each department.

```
SELECT a.Name AS employee, b.name AS supervisor
FROM staff a, staff b
WHERE a.Parent_staff_id = b.Staff_ID
```

Employee	Supervisor	
Anna Benjamin	Emma Brooks	
Olivia Morgan	Noah Bennett	
Ethan Parker	Noah Bennett	
Mia Sullivan	Emma Brooks	
Sophia Cooper	Noah Bennett	
Liam Turner	Noah Bennett	
Ava Harris	Noah Bennett	
Lucas Gray	Emma Brooks	
Jack Foster	Noah Bennett	
Bennett Clark	Leo Hunt	
Lila Fox	Leo Hunt	

Scenario 2: Display Host and Location Information of All Airbnb Properties Along with its Property Features

Query: Show the details of the hosts Airbnb property locations and property features

```
SELECT Host.HOST_ID AS Host_ID, Host.Host_Name AS Host_Name,
Property.Property_ID AS Property_ID, Property.Address AS Property_Address,
Features.Feature_Description AS Property_Feature
FROM ((Host
INNER JOIN Property ON Host.Host_ID = Property.Host_ID)
INNER JOIN Property_Features ON Property.Property_ID =
Property_Features.Property_ID)
INNER JOIN Features ON Property_Features.Feature_ID = Features.Feature_ID;
```

Host_ID ▾	Host_Name ▾	Property_ID ▾	Property_Address ▾	Property_Feature ▾
100	Anna Tee	1	123 Property St	Outdoor Pool
100	Anna Tee	1	123 Property St	Only Washer No Dryer
100	Anna Tee	1	123 Property St	Pet Friendly Features
200	Ben All	2	234 Property St	Hot Tub Jacuzzi
200	Ben All	2	234 Property St	Only Dryer No Washer
300	Chris Top	3	89 Maple Lane	Indoor Pool
400	Daniel Fit	4	36 Cedar Street	Home Theater Room
400	Daniel Fit	4	36 Cedar Street	Hot Tub Jacuzzi
400	Daniel Fit	4	36 Cedar Street	Private Garden
500	Evelyn Smith	5	201 Pine Avenue	Balcony and Terrace
600	George Tan	6	74 Birch Road	Private Garden
700	Frank Red	7	190 Spruce Street	Home Theater Room
800	Janet Reed	8	45 Willow Avenue	Washer and Dryer
900	Alice Won	9	132 Chestnut Lane	Outdoor Pool
900	Alice Won	9	132 Chestnut Lane	Indoor Pool
1000	Devin Fry	10	88 Ash Street	Outdoor Fire Pit
300	Chris Top	11	66 Sycamore Avenue	Pet Friendly Features
600	George Tan	12	152 Popular Street	Indoor Fireplace
600	George Tan	12	152 Popular Street	Pet Friendly Features
400	Daniel Fit	13	145 Whitestone Lane	Outdoor Pool
400	Daniel Fit	14	86 Handle Drive	Hot Tub Jacuzzi

Scenario 3: Retrieve Below Average Customer Review

Query: Show the details of all customer reviews below a 3 star rating.

```

SELECT Customer.Customer_ID, Customer.Customer_Name,
       Customer_Review.Customer_Review_Response,
       Customer_Review.Customer_Star_Rating
FROM Customer
INNER JOIN Customer_Review ON Customer.Customer_ID =
       Customer_Review.Customer_ID
WHERE Customer_Review.Customer_Star_Rating < 3;

```

Customer_ID ▾	Customer_Name ▾	Customer_Review_Response ▾	Customer_Star_Rating ▾
1003	Christie Dan	Private garden was a nice touch, but the rest of the stay did not meet my standards. Space was very	1
1009	Ingrid Joy	Perfect location, but we had a lot of issues with fireplace.	2
1005	Evan Frank	Experience would have been better if property was cleaner. Disappointed with the cleanliness and c	1
1007	Gary Harry	Washer and dryer broken unlike what was advertised. Needed to leave the property to accommoda	1
1010	Janice Kin	Appreciate the pet-friendliness. Fireplace did not work as advertised, so living room got cold very	2

Scenario 4: Display specific features for properties

Query: List all properties along with the features they have.

```
SELECT Property_Features.Property_ID, Property.Address, Features.Feature_Description
FROM Property, Property_Features, Features
WHERE Property.Property_ID = Property_Features.Property_ID
AND Property_Features.Feature_ID = Features.Feature_ID
```

Property_ID	Address	Feature_Description
1	123 Property St	Outdoor Pool
1	123 Property St	Only Washer No Dryer
1	123 Property St	Pet Friendly Features
2	234 Property St	Hot Tub Jacuzzi
2	234 Property St	Only Dryer No Washer
3	89 Maple Lane	Indoor Pool
4	36 Cedar Street	Home Theater Room
4	36 Cedar Street	Hot Tub Jacuzzi
4	36 Cedar Street	Private Garden
5	201 Pine Avenue	Balcony and Terrace
6	74 Birch Road	Private Garden
7	190 Spruce Street	Home Theater Room
8	45 Willow Avenue	Washer and Dryer
9	132 Chestnut Lane	Outdoor Pool
9	132 Chestnut Lane	Indoor Pool
10	88 Ash Street	Outdoor Fire Pit
11	66 Sycamore Avenue	Pet Friendly Features
12	152 Popular Street	Indoor Fireplace
12	152 Popular Street	Pet Friendly Features
13	145 Whitestone Lane	Outdoor Pool
14	86 Handle Drive	Hot Tub Jacuzzi

Scenario 5: Retrieve booking details with payment information

Query: List out all the details of customers that check out on 4/2/2024. (e.g. check-in, check-out dates, along with payment Payment_amount, Payment_Form.

```
SELECT Booking_Information.Booking_ID, Booking_Information.Booking_Check_In,
Booking_Information.Booking_Check_Out, Payment.Payment_Amount,
Payment.Payment_Form
FROM Booking_Information, Payment
WHERE Booking_Information.Booking_ID = Payment.Booking_ID AND
Booking_Information.Booking_Check_Out = #04/02/2024#
```

Booking_ID	Booking_Check_In	Booking_Check_Out	Payment_Amount	Payment_Form
161	3/30/2024	4/2/2024	150	Credit Card
184	4/1/2024	4/2/2024	75	Debit Card

Scenario 6: Retrieve Policies for Specific Properties

Query: Display properties that have “Property doesn’t allow smoking indoors” as Policy_Description, including the property address, property_ID, and the policy descriptions

```
SELECT Property.Property_ID, Property.Address, Policies.Policy_Description
FROM Property, Policies, Property_Policies
WHERE Property_Policies.Policy_ID = Policies.Policy_ID
AND Policies.Policy_Description = "Property doesn't allow smoking indoors"
```

Property_ID ▾	Address ▾	Policy_Description ▾
1	123 Property St	Property doesn't allow smoking indoors
2	234 Property St	Property doesn't allow smoking indoors
3	89 Maple Lane	Property doesn't allow smoking indoors
4	36 Cedar Street	Property doesn't allow smoking indoors
5	201 Pine Avenue	Property doesn't allow smoking indoors
6	74 Birch Road	Property doesn't allow smoking indoors
7	190 Spruce Street	Property doesn't allow smoking indoors
8	45 Willow Avenue	Property doesn't allow smoking indoors
9	132 Chestnut Lane	Property doesn't allow smoking indoors
10	88 Ash Street	Property doesn't allow smoking indoors
11	66 Sycamore Avenue	Property doesn't allow smoking indoors
12	152 Popular Street	Property doesn't allow smoking indoors
13	145 Whitestone Lane	Property doesn't allow smoking indoors
14	86 Handle Drive	Property doesn't allow smoking indoors

Scenario 7: Get Hosts Managing Multiple Properties

Query: List all hosts who own more than one property, including their name and contact information

```
SELECT Host.Host_ID, Host.Host_Name, Host.Email_Address,
Host.Telephone_Number, COUNT (Property.Property_ID) AS Property_Count
FROM Host, Property
WHERE Host.Host_ID = Property.Host_ID
GROUP BY Host.Host_ID, Host.Host_Name, Host.Email_Address,
Host.Telephone_Number
HAVING COUNT(Property.Property_ID) > 1
```

Host_ID ▾	Host_Name ▾	Email_Address ▾	Telephone_Number ▾	Property_Count ▾
300	Chris Top	c.top@gmail.com	3456789012	2
400	Daniel Fit	d.fit@yahoo.com	4567890123	3
600	George Tan	g.tan@yahoo.com	6789012345	2

VII. Conclusion

After building out the database system, collecting Airbnb property management information, and analyzing data, Star Management found the root cause of their negative reviews to be delayed service and lack of maintenance of property features. The new Airbnb management database created includes a total of 12 data tables which summarizes all the relevant management information including (but not limited to): hosts, customers, bookings, properties, and staff.

To ensure a clear structure for the database system, we first created an Entity Relationship Diagram (ERD) and converted it into a Relational Data Model (RDM) to visualize and standardize the relationships between selected entities and corresponding attributes. After normalizing the RDM, we used SQL queries to create tables for each entity and built relationships between them using primary and foreign keys. Finally, we inserted all the data to the tables, ensuring everything was organized and connected properly.

Moving forward, with the addition of the extensive database system, the company will automatically flag lower-rated reviews as main priorities. Following this, the company will send additional maintenance and cleaning staff to these high-priority Airbnb properties for inspections before any additional bookings can be made. Additionally, the company will also allow Airbnb hosts to more efficiently request servicing from the main company. Ultimately, the new system will regularly track the maintenance and status of the Airbnb properties to ensure consistent and the best quality Airbnb properties on the East Coast.

Since then, this database system has helped the company with efficiently and effectively managing the 150 Airbnb properties to positively impact the customer experience. Subsequently, Star Management Company is well-positioned to continue this positive trajectory of consistent customer satisfaction, improved operational efficiency, and continuous support of its hosts in delivering the best customer experience.