

Rent Our Lake House DBMS Project

ISA 245 Section A Group A3

Emma Kate Dygert, Murphy Heffernan, John Mungenast, Yixin Zhang

Jay Palmisano

Oracle Account: zhang 539 Password: m7ami

5/10/2022

1. Project Description

The business we will be researching is Rent Our Lake House, a small business owning multiple vacation properties on Norris Lake, Tennessee, with hopes to expand into other areas. The properties are homes of varying sizes with varying amounts of bedrooms and bathrooms. Additionally, each home has its own appliances and its own amenities. The rentals are short term, usually ranging from 2 nights to 7 nights. Guests can book the homes on websites like VRBO or AirBnb, or contact the host directly. Guests pay per night in the home, plus cleaning fees, and taxes. The properties require maintenance, with the changing seasons and many people coming in and out. Maintenance can include routine events like mowing the grass or fixing things like a broken fridge.

The existing system is an excel sheet keeping track of bookings, payments, guest information, dates, cleaning, maintenance, booking source, etc. Contact information for maintenance and cleaning is not well organized and different people have different systems they use to keep in contact and keep track of payments and services. A lot of information could be harnessed to help the business grow and make strong business decisions.

An example of a relevant business process would be the process of making the booking. There are many different “things” communicating to go through this process. For example, a potential customer must go onto Airbnb or VRBO, then request a booking. The platform used then must notify the owner of the request, and the owner must approve or deny it. Another example of a business process is receiving payment. Based on the time of year, the number of people, length of stay, the price varies, and the owner must keep track of that. In addition, there are different payments due at other times and different payment methods that the owner and guest must agree on. The owner then has to share this information with the bank.

A database system would help this business tremendously. It would help the business market more successfully, stay on top of cleaning and maintenance, and simplify keeping track of guest data. The business could input the data from the past year to start and refine what was needed and, most importantly, to keep track of and find relationships. Then it would be easy to build on that existing data. An example of a relationship between different things would be dates connecting the guest's schedule to the cleaning schedule. A database would make information much easier to track when the business uses multiple platforms to accept bookings. It would also remove data redundancy, as guests will book the same home multiple times, and their data would be easy to find if stored in a database.

The business rules includes, renters have to stay in at least one house, and each property or house can only be booked by one renter at a time. For the house at Southern charm the maximum number of guests is 16, and there cannot be more than 28 guests at Southern Comfort II. We also have flexible payment terms that allow the renter to pay partially or pay after the check out date. And all amenities must stay at the original property so they cannot be moved around. For cleaning and

maintenance crews each of them will need a unique task ID for each task, and all the appliances must be fixed within 2 weeks of being broken.

2. Data Modeling

a. User Views

External Views:

1. Rent Our Lake House website
 2. Airbnb/VRBO
 3. Maintenance Scheduling
 4. Review Section from Airbnb/VRBO
 5. Quote breakdown/invoice
 6. Norris Lake Information
 7. Payment/booking page on Airbnb/VRBO
 8. Availability Screen on Airbnb/VRBO
-
1. reviews – customers experience that already were guests
 - a. number rating (And calculated average rating) (number, 0.0)
 - A number rating of the customer's stay
 - b. customer name (text)
 - The full name of the customer who rented the house
 - c. description (text)
 - The content written by the renter about their stay
 - d. Date posted (date)
 - The date that the review was posted
 - e. Review title phrase (text)
 - A brief title/summary of what to expect from the review
 - f. Date of stay (month/year)
 - When the renter stayed at the house(s)
 - g. Where customer is from (Only on some reviews) (text)
 - Optional field, where the review writer is from

2. house – the rental property that the guest is booking

a. Description (text)

-a description of the house

b. Location (text)

-Where the house is located

d. Price (number \$000,000)

-How much the house costs to rent

e. Number of bed (number 00)

-How many bedrooms are in the house

f. Amenities (text)

-Which amenities are contained within the house

g. Policies (text)

-Rules of the house

h. Availability (date)

-When the house is available to be rented

i. House Name (text)

-Name of the house

J. number of bath (number 00)

-How many bathrooms are in the house

3. Renter- person renting the property

a. Name (text)

-Name of the person renting the house

b. Email (text)

-Email address of the person renting the house

C. phone number (number, 000-000-0000)

-Phone number of the person renting the house

C.date visited (date)

-The date which the renter visited

d. Cost of cleaning (number, \$000.000)

-How much the house costs to clean

e. Number of guests staying (number)

-The number of guests that will be lodging including the renter

f. Housename (text)

-The name of the house

G. age (number, 000) (limited to 30+)

-The age of the person renting the house

4. Maintenance - people who clean the property

a. Names (text)

-The names of the maintenance crews

b. Contact information (text)

-String containing phone number and email address of each crew

c. Dates available for cleaning (??) (date)

-When the crews are available to clean

d. Price to clean (number, \$000.000)

-How much it costs to clean the house

e. Supplies needed in house (??) (text)

-Which supplies the house needs to carry

5. Appliances - fridges, washing machines, etc.

a. Type of appliance (text)

-The name of appliance

b. Repair cost (number, \$000.000)

-How much each appliance costs to fix

- c. Date purchased (date)
 - When the appliance was purchased
 - d. Times repaired (number)
 - How many times the appliance has been repaired
 - e. Cost of purchase (number, \$000.000)
 - How much the appliance costed to purchase
 - f. Currently Broken (yes/no)
 - If the appliance is currently broken
 - g. Room located in (???) (text)
 - Which room in the house the appliance is located in
6. Host - people who own the property
- a. Host name (text)
 - The name of the owner of the house
 - b. Phone number (number, 000-000-0000)
 - The phone number of the person who owns the house
 - c. Email address (text)
 - The email address of the person who owns the house
7. Booking Order
- a. OrderID (number)
 - A unique ID for the order of the renting of the house
 - b. Date booked (date)
 - The date the house was booked
 - c. Check In date (date)
 - The date the guests can arrive
 - d. Check out date (date)
 - The date the guests have to leave by
 - e. Invoice Num (number)
 - Unique ID FK for an individual invoice to the renters
 - f. Renter's name (text)

- The name of the renter

g. Cancellation policy (text)

- Policy behind cancellation

8. Invoices

a. Invoice Num (number)

- A Unique ID for individual invoices to renters

b. Total (number, \$000.000)

- How much the house cost to rent before taxes

c. Tax (number, \$000.000)

- How much taxes were for renting the house

d. Grand Total (number, \$000.000)

- A grand sum of the total and the taxes put together

e. Billing information (text, address)

- Where the invoice will be sent to

f. Date (date)

- Date the invoice was written

g. Description (text)

- Brief description of the invoice

9. Task

a. TaskID (number 00-0000)

- Unique ID for seeing which task needs to be done

b. TaskDescription (text)

- A description of which task needs to be done

c. ApplianceID (Number 000-000)

- Unique ID for each appliance, helps crews know what needs to be repaired

d. HouseID (number 000-0000)

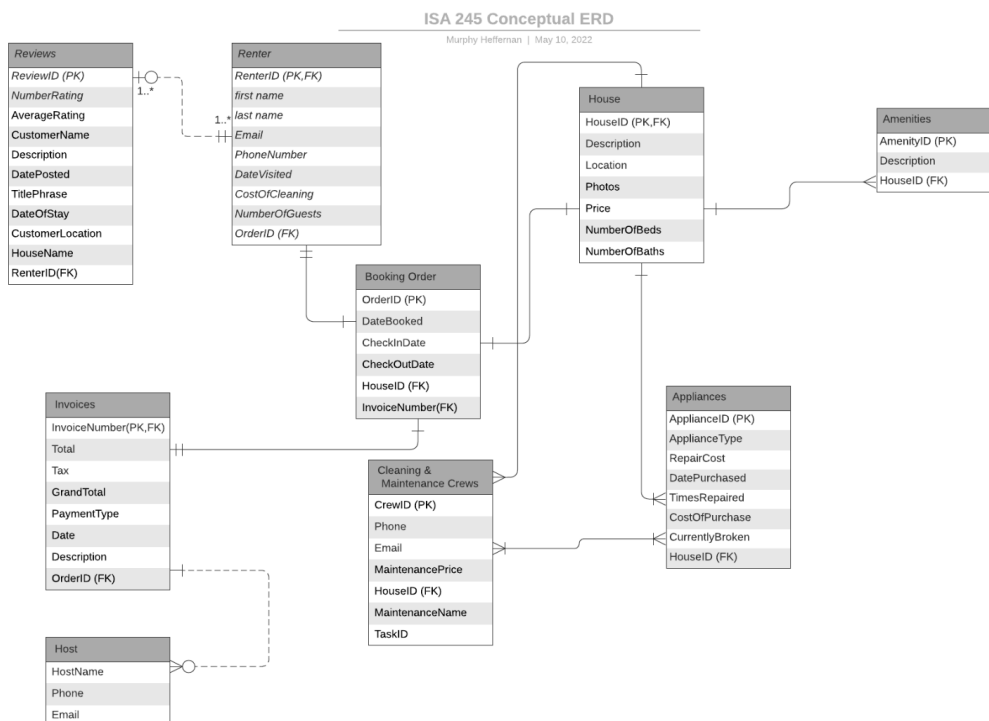
- Unique ID for the house which requires service

10. Amenities

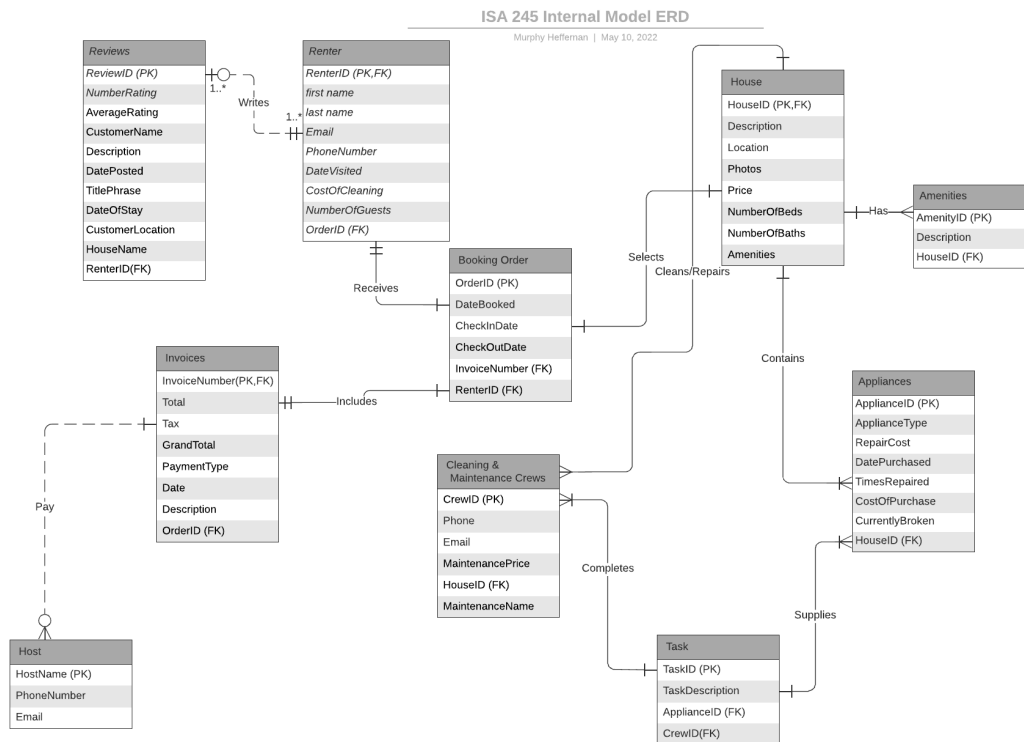
- a. AmenityID NUMBER(5)
 - Unique ID for each amenity
- b. Description VARCHAR2(50)
 - Description of what each amenity is and where it's located
- c. HouseID NUMBER(3)
 - The house ID which the amenity is located in

b. ER Diagrams

Conceptual ERD Model:



Internal ERD Model:



We checked for the third normal form within our internal model by ensuring that all of the data included in the database was necessary and not redundant. In the process of making our model 3NF, we removed attributes from our original user view. An example of this was from the house table in our user view. It had an attribute “policies”, however we deemed it necessary and removed it from the database in order to remove redundant and unnecessary data and ensure our model is in third normal form. We also considered adding a table called “Stay” that would record the date of check ins and check outs from the lake houses, however this would overlap with the dates already in the booking order table and create redundancy in the database, so we decided not to include that table in our model.

c. Descriptions To Make Diagrams Clear

The conceptual model diagram depicts the tables present within our database of “Rent Our Lake House” data. Within each table, an identifier/primary key is present and any tables that have a relationship with each other have a foreign key as well if required. The conceptual model also displays the relationships between tables that we created for our varying user views, whether that be a one-to-one, one-to-many, or many-to-many relationship. The conceptual model diagram additionally demonstrates where there are weak relationships between tables as compared to stronger relationships. The weaker relationships (renter to reviews; invoices to host) are depicted with a dashed relationship line and the stronger relationships are depicted with a solid relationship line. The diagram also includes a many-to-many relationship within the table for cleaning & maintenance crews and appliances, however it does not yet include a

bridge table, as that was an addition made later on in our internal model diagram. The conceptual model is where we first implemented our business rules within the model to help create the relationships and those rules were that renters had to stay in at least one house, which resulted in an optional one to many relationship, each property can only be booked by one renter at a time, which resulted in a one to one relationship between the house and renter, the business rules that limit the number of guests per house are not depicted in the model, however they impact if a renter can rent a house and the values for attributes. Other rules include the renter's to be able to pay partially or all at once, amenities must stay at the house they are assigned with a one-to-many relationship with the house table, and appliances have to be fixed within two weeks, which is managed on the attribute level and not relationship level.

The internal model diagram is a more complete and comprehensive depiction of our database of "Rent Our Lake House" data. The internal model goes a step further than the conceptual model by checking that all the tables are in third normal form. It also includes a bridge table called "Task" that was created for the many-to-many relationship between cleaning & maintenance crews and appliances. The task table has its own primary key, however it also contains the primary keys of each of the tables it connects (cleaning & maintenance crews and appliances) as foreign keys to continue to connect the tables as a many-to-many relationship. The internal model diagram also further explains how the tables are related to each other through the descriptive words/phrases that accompany each table relationship line to allow the reader to understand the connections between our tables and data within the database. The internal model is the most descriptive model of our database and as a result is the model diagram we used to base our SQL database in Oracle off of.

d. Data Dictionary

House: displays information about each house that is part of the business

Name	Null	Type	Business meaning
houseid	Not Null	Number(3)	number to uniquely identify the house. PK of the table
Description		VARCHAR2(50)	description of the house provided on the listing
Location		VARCHAR2(25)	address of the house
Photos		BLOB	hotographs of the house used in the listing (Can be Null)
Price		Number(6,2)	price per night of the house
NumberofBeds		Number(2)	number of bedrooms in the house
Numberofbaths		Number(2)	number of bathrooms in the house

Amenities: displays the amenities in the house

Name	Null	Type	Business meaning
amentiyid	Not Null	Number(5)	number to uniquely identify the amenity. PK of the table
Description		VARCHAR2(50)	description of the amenity and its purpose
Houseid		Number(3)	number to uniquely identify the house, FK in the table

Appliances: displays the appliances in the house

Name	Null	Type	Business meaning
applianceid	Not Null	Number(5)	number to uniquely identify the appliance. PK of the table
appliancetype		VARCHAR2(15)	type of appliance the specific appliance is
Repaircost		Number(5,2)	cost to repair the appliance
Datepurchase		DATE	date the appliance was purchased
Timesrepaired		Number(2)	number of times the appliance has been repaired
CostofPurchase		Number(6,2)	cost of the appliance when purchased
CurrentlyBroken		Char(1)	if the appliance is currently broken. Yes or No
Houseid		Number(3)	number to uniquely identify the house, FK in the table

Booking_Order: displays information about the booking order of the renter for the house

Name	Null	Type	Business meaning
OrderID	Not Null	Number(5)	number to uniquely identify the order. PK of the table
DateBooked		DATE	date the order was submitted
CheckInDate		DATE	start date of the stay
CheckOutDate		DATE	end date of the stay

InvoiceNumber		Number(5)	number to uniquely identify the invoice, FK in this table
Renterid		Number(5)	number to uniquely identify the customer, FK in this table

Renter: displays information about the renter of the house

Name	Null	Type	Business Meaning
renterID	Not Null	Number(5)	number to uniquely identify the customer, PK of the table
FirstName		Char(30)	first name of the customer
LastName		Char(30)	last name of the customer
Email		VARCHAR2(30)	email address of the customer
PhoneNumber		Number(10)	phone number of the customer
DateVisited		DATE	date the customer stayed
CostOfCleaning		Number(8)	cost of the cleaning fee for the stay
NumberOfGuest		Number(2)	number of people staying on the property
OrderID		Number(5)	number to uniquely identify the order, FK in this table

Invoices: displays information about the invoice for the stay

Name	Null	Type	Business Meaning
------	------	------	------------------

InvoiceNumber	Not Null	Number(5)	number to uniquely identify the invoice. PK of the table
Total		Number(8,2)	total price of the stay
Tax		Number(8,2)	amount of taxes for the stay
GrandTotal		Number(8,2)	total price of the stay including tax
DateofBilling		DATE	The date the invoice was sent
Description		VARCHAR2(500)	Description of what the invoice is for
Orderid		Number(5)	Number to uniquely identify the order, FK in the table

Clean_main_crews: displays information about the cleaning or maintenance crew member

Name	Null	Type	Business Meaning
crewID	Not Null	Number(5)	number to uniquely identify the crew member. PK of the table
Phone		Number(10)	phone number of the crew member
Email		VARCHAR2(30)	email address of the crew member
MaintenancePrice		Number(6)	price of the cleaning or maintenance task
MaintenanceName		Char(30)	name of the crew member
HouseID		Number(3)	number to uniquely identify the house, FK in the table

Task: displays the tasks completed by cleaning and maintenance

Name	Null	Type	Business Meaning
TaskID	Not Null	Number(5)	number to uniquely identify the task. PK of the table
applianceID		Number(5)	number to uniquely identify the appliance, FK in the table
CrewID		Number(5)	number to uniquely identify the crew member, FK in the table
Taskdescription		VARCHAR2(500)	description of the task done

Reviews: displays information regarding reviews

Name	Null	Type	Business Meaning
reviewID	Not Null	Number(5)	number that uniquely identifies each review. PK of the table
number_rating		Number(1)	the number rating given in each review
Average_rating		Number(1)	average number rating of all reviews
customer_name		VARCHAR2(30)	name of the individual who posted the review
description		VARCHAR2(500)	word content of the review (Can be NULL)
date_posted		DATE	date the review was posted
Title_phrase		VARCHAR2(30)	title used for the review
DateofStay		DATE	date the customer

			stayed on the property being reviewed
Customer_location		VARCHAR2(30)	where the customer is from. (Can be NULL)
renterid		Number(5)	number to uniquely identify the customer. FK in this table

Host: Displays information about the host of the property

Name	Null	Type	Business Meaning
hostname	Not Null	CHAR(30)	Name of the host
Phonenumber		Number(10)	Phone number of the host
Email		VARCHAR2(30)	Email address of the host

3. Implementation

a. Description of implementation activities

My group decided to go with Oracle instead of Access, because Oracle is user friendly and helps to write queries easily, we have more knowledge about oracles compared with Access. For implementation, we insert the data from excel into Oracle using create table and insert command, we then create queries with condition and calculations to find the information we need to help the owner develop more business insights and generate information to support the decision making.

b. Demonstration queries

1. Uncollected amount percentage (Subquery)

Because the business accept partial payment so the owner want to know

What is the proportion of partial payment and not paid amount in the total income. Therefore this query displays the percentage of each uncollected order amount from total income.

Select firstname,lastname,email,orderid,description,to_char(total,'\$999,999.99') as TotalAmt,

to_char(100*sum(total)/(select sum(total) from invoices),'99.99') as percentofTotalAmt

from invoices

Full Join booking_order using (orderId)

Join renter using (orderid)

Having description IN ('not paid', 'partially paid')

group by firstname,lastname,email,orderid, to_char(total,'\$999,999.99'),description

order by to_char(total,'\$999,999.99') desc

FIRSTNAME	LASTNAME	EMAIL	ORDERID	DESCRIPTION	TOTALAMT	PERCENTOFTOTALAMT
julian	smith	julian01@yahoo.com	12349	partially paid	\$11,000.00	12.93
hannah	bailey	hb@yahoo.com	12346	partially paid	\$10,099.00	11.87
ava	miller	avamiller1@gmail.com	12348	not paid	\$8,000.00	9.41
david	will	davidw@gmail.com	12352	partially paid	\$8,000.00	9.41
thomas	train	choochoo@yahoo.com	12351	not paid	\$7,500.00	8.82

2.Tracking of partial payment (Outer Join)

This query will help the owner or the manager of the business to track the partial payment from each customer mentioned in the business process. The query returns the detailed information about order and customer so the manager can contact the customer and track the payments.

Select firstname,lastname,email,invoices.invoicenumber,orderid,houseid,to_char(total,'\$999,999.99') as TotalAmt,checkindate,checkoutdate,description

from invoices

Full Join booking_order using (orderid)

Join renter using (orderid)

where description='partially paid'

order by to_char(total,'\$999,999.99') desc

FIRSTNAME	LASTNAME	EMAIL	INVOICENUMBER	ORDERID	HOUSEID	TOTALAMT	CHECKINDATE	CHECKOUTDATE	DESCRIPTION
julian	smith	julian01@yahoo.com	54325	12349	112	\$11,000.00	25-Jun-2021	27-Jun-2021	partially paid
hannah	bailey	hb@yahoo.com	54322	12346	111	\$10,099.00	04-Jun-2021	06-Jun-2021	partially paid
david	will	davidw@gmail.com	54328	12352	114	\$8,000.00	16-Jul-2021	18-Jul-2021	partially paid

3. Tracking of not paid orders

This query is different from the partial payment tracking, this is the query for non-paid transactions which is a more serious situation for the business. The query allows the manager to see the amount owed by each customer and also the accumulated uncollected amount, which would help the business to manage the cash flow better.

Select firstname,lastname,email,invoices.invoicenumber,orderid,houseid,to_char(total,'\$999,999.99') as TotalAmt,checkindate,checkoutdate,(select to_char(sum(total),'\$999,999.99') from invoices where description='not paid') as TotalunpaidAmt,description

from invoices

Full Join booking_order using (orderid)

Join renter using (orderid)

Having description='not paid'

group by firstname,lastname,email,invoices.invoicenumber,
orderid,houseid,to_char(total,'\$999,999.99'),checkindate,checkoutdate,description

order by to_char(total,'\$999,999.99') desc

FIRSTNAME	LASTNAME	EMAIL	INVOICENUMBER	ORDERID	HOUSEID	TOTALAMT	CHECKINDATE	CHECKOUTDATE	TOTALUNPAIDAMT	DESCRIPTION
ava	miller	avamiller1@gmail.com	54324	12348	114	\$8,000.00	18-Jun-2021	20-Jun-2021	\$15,500.00	not paid
thomas	train	choochoo@yahoo.com	54327	12351	111	\$7,500.00	09-Jul-2021	11-Jul-2021	\$15,500.00	not paid

4. Estimate revenue for house 111

Since each order will not only generate the income for the business but also income related costs that are very likely to occur. In our business after the guest leaves the house there might be some maintenance and repair needed so the business wants to know the income left from maintenance and repair.

```
select TO_CHAR(sum(total-maintenanceprice-repaircost),'$999,999.99') AS ESTREVENUE ,houseid
```

```
FROM clean_main_crews
```

```
Join appliances using (houseID)
```

```
FULL JOIN booking_order USING (houseID)
```

```
FULL JOIN INVOICES USING (invoiceNUMBER)
```

```
GROUP BY HOUSEID
```

```
HAVING houseid=111
```

ESTREVENUE	HOUSEID
\$190,357.00	111

5. Repair cost by order

This query shows the repair cost by orderid. After each order the house may need to repair several appliances and this can help the owner understand how much they need to spend on repair for each order.

```
Select booking_order.orderid,to_char(sum(repaircost),'$999,999.99')repaircost,to_char(total,'999,999.99')
totalrevenue
```

```
FROM clean_main_crews
```

```
Join appliances using (houseID)
```

FULL JOIN booking_order USING (houseID)

FULL JOIN INVOICES USING (invoiceNUMBER)

Where repaircost is NOT NULL

Group by booking_order.orderid,to_char(total,'999,999.99')

ORDERID	REPAIRCOST	TOTALREVENUE
12345	\$1,945.00	7,150.00
12349	\$310.00	11,000.00
12346	\$1,945.00	10,099.00
12351	\$1,945.00	7,500.00

6.Subquery - Renter stay more than 2 days

This query is focused on the renter because the business wants to focus on the longer orders compared to those shorter ones, because longer stays will bring more income for the business and the business wants to keep in touch with those customers.

SELECT orderid, firstname,lastname,email,checkindate,checkoutdate,(select
sum(checkoutdate-checkindate) from booking_order WHERE checkoutdate-checkindate>2) TotalDays,
sum(checkoutdate-checkindate) DaysStay

From renter

JOIN booking_order USING (orderid)

JOIN invoices USING (orderid)

where checkoutdate-checkindate>2

GROUP BY ORDERID,firstname,lastname,email,checkindate,checkoutdate

ORDERID	FIRSTNAME	LASTNAME	EMAIL	CHECKINDATE	CHECKOUTDATE	TOTALDAYS	DAYSSTAY
12345	emma	kate	emmadyget@gmail.com	20-May-2021	27-May-2021	10	7
12354	jill	mack	jillmack@yahoo.com	30-Jul-2021	02-Aug-2021	10	3

7.unsatisfied customer

From a business operation perspective we want to reach out to customers who are unhappy with the experience and hear feedback from them, so this query would help to find reviews which are lower than the average rating.

Select orderid,firstname,lastname,number_rating, houseid,description,email

From reviews

Join renter USING (renterid)

JOIN Booking_order USING (orderid)

WHERE number_rating< average_rating

ORDERID	FIRSTNAME	LASTNAME	NUMBER_RATING	HOUSEID	DESCRIPTION	EMAIL
12348	ava	miller	3	114	could be better	avamiller1@gmail.com

8. Appliances repaired

Since some of the appliances are repaired more than once and it will increase expenses for business each time when something needs to be repaired. So this table will help the manager to see the cost to repair each appliance and make a decision on whether they want to keep the appliance or replace it.

Select houseid,applianceid,appliancetype, repaircost,timesrepaired,sum(repaircost*timesrepaired) as totalrepaircost

FROM appliances

HAVING timesrepaired>0

group by houseid,applianceid,appliancetype,repaircost,timesrepaired

HOUSEID	APPLIANCEID	APPLIANCETYPE	REPAIRCOST	TIMESREPAIRED	TOTALREPAIRCOST
111	56794	microwave	120	1	120
111	56792	oven	400	1	400
111	56795	Freezer	300	1	300
111	56789	fridge	300	1	300
111	56793	toaster	100	2	200

9. House with amenities

This query gives the overview of what appliances that each house has. And also shows the house that has no appliances.

SELECT houseid,house.description, location, numeroftbeds,numeroftbaths,amenities.description

From house

Full Join amenities using (houseid)

order by houseid

HOUSEID	DESCRIPTION	LOCATION	NUMBEROFBEDS	NUMBEROFBATHS	DESCRIPTION
111	Southern Charm	229 Cameron Loop	6	6	Private dock with 2 floors for boats and lounging
111	Southern Charm	229 Cameron Loop	6	6	Hot Tub
111	Southern Charm	229 Cameron Loop	6	6	Garage
111	Southern Charm	229 Cameron Loop	6	6	Back Patio
111	Southern Charm	229 Cameron Loop	6	6	Outdoor Fireplace
111	Southern Charm	229 Cameron Loop	6	6	Beverage Fridge
112	Big Bertha	481 Esther Lane	2	8	(null)
113	Southern Comfort	10 Ski Lane	9	9	Ping-Pong Table
113	Southern Comfort	10 Ski Lane	9	9	Speaker System
113	Southern Comfort	10 Ski Lane	9	9	Ice Machine
113	Southern Comfort	10 Ski Lane	9	9	Large Grill
114	Comfy Cabin	100 Drive Lane	4	4	(null)
115	Happy Hut	614 Lake View Lane	3	1	(null)
116	Hazel's House	914 Nut Drive	6	5	(null)
117	Pippa's Palace	10 Popper Lane	10	12	(null)
118	Millie's Manor	150 Golden Drive	6	6	(null)
119	Diamond Dock	200 Ginger Street	5	6	(null)
120	Mickey Mouse Clubhouse	001 Daisy Drive	8	8	(null)

10. Appliance cost by order

This query shows the total cost that include the cost of purchase appliances and repair by order id.

Select orderid, houseid, to_char(sum(repaircost+costofpurchase), '\$999,999.99') as totalcost

from appliances

JOIN booking_order USING (houseid)

group by orderid, houseid

order by to_char(sum(repaircost+costofpurchase), '\$999,999.99')

ORDERID	HOUSEID	TOTALCOST
12349	112	\$1,035.00
12345	111	\$7,945.00
12351	111	\$7,945.00
12346	111	\$7,945.00

4. Summary

In summary, the benefits of the database are huge to the business. The biggest advantage of the database is to enable users to create dimensions or check information across different tables easily with the use of the relationship between tables. Compared to the old excel files where users can only view data from one table at a time. In the new system, the owner or the manager can view the total sales of the business from different perspectives. In other words, they can add dimensions to the fact using the 'by condition' which allows the owner of the business to explore more business insights and enhance the decision making. For example, with the implementation of the database, the user can not only see the sales amount from each order or the total sales, but the user can also break the total sales amount down and group the amount by houseid, region, or any other dimension or perspective they want.

The database can also help the owner of the business to filter data and keep track of the information they need. As mentioned above in the project description, the owner wants to track the payment from the renter (customer). So we create two queries for the owner (query 2 and 3) . One is used to track the partial payment and the other is used to track the not paid transactions which are orders that haven't been paid at all.

In addition, the database also adds more accuracy to the data because there are data constraints when inserting the data record into the database and if the data entered violates the constraints the data cannot be entered into the database this approach makes sure that all the data entries are accurate and improves the data quality. For instance, we emphasize that all the order id can only have 5 numbers, so if the user accidentally entered 6 digits to the order id the insert request will be failed.

5. Appendix

a. Oracle DDL statement

```
DROP TABLE reviews CASCADE CONSTRAINTS;
DROP TABLE renter CASCADE CONSTRAINTS;
DROP TABLE booking_order CASCADE CONSTRAINTS;
DROP TABLE house CASCADE CONSTRAINTS;
DROP TABLE host CASCADE CONSTRAINTS;
DROP TABLE invoices CASCADE CONSTRAINTS;
DROP TABLE appliances CASCADE CONSTRAINTS;
DROP TABLE task CASCADE CONSTRAINTS;
DROP TABLE clean_main_crews CASCADE CONSTRAINTS;
DROP TABLE amenities CASCADE CONSTRAINTS;
```

```
CREATE TABLE house
(houseid NUMBER(3) PRIMARY KEY,
Description VARCHAR2(50),
Location VARCHAR2(25),
Photos BLOB NULL,
Price NUMBER(6,2),
Numberofbeds NUMBER(2),
Numberofbaths NUMBER(2));
```

```
CREATE TABLE amenities
(amenityid NUMBER(5) PRIMARY KEY,
Description VARCHAR2(50),
HouseID NUMBER(3),
CONSTRAINT amenities_houseid FOREIGN KEY (houseid) REFERENCES house(houseid));
```

```
CREATE TABLE appliances
(applianceid NUMBER(5) PRIMARY KEY,
Appliancetype VARCHAR2(15) NOT NULL,
Repaircost NUMBER(5,2) ,
Datepurchased DATE,
Timesrepaired NUMBER(2) NULL,
CostOfPurchase NUMBER(6,2),
CurrentlyBroken CHAR(1),
HouseID NUMBER(3),
CONSTRAINT appliances_houseid FOREIGN KEY (houseid) REFERENCES house(houseid));
```

```
CREATE TABLE booking_order
(OrderID NUMBER(5) PRIMARY KEY,
DateBooked DATE,
CheckInDate DATE,
```

```

CheckOutDate DATE,
InvoiceNumber NUMBER(5),
HouseID NUMBER(3),
CONSTRAINT booking_order_houseid FOREIGN KEY (houseid) REFERENCES
house(houseid));

```

```

CREATE TABLE renter
(RenterID NUMBER(5) PRIMARY KEY,
FirstName CHAR(30) NOT NULL,
LastName CHAR(30) NOT NULL,
Email VARCHAR2(30),
PhoneNumber NUMBER(10),
DateVisited DATE,
CostOfCleaning NUMBER(8),
NumberOfGuests NUMBER(2),
OrderID NUMBER(5),
CONSTRAINT renter_OrderID FOREIGN KEY (orderid) REFERENCES
booking_order(OrderID));

```

```

CREATE TABLE invoices
(InvoiceNumber NUMBER(5) PRIMARY KEY,
Total NUMBER(8,2),
Tax NUMBER(8,2),
GrandTotal NUMBER(8,2),
DateOfBilling DATE,
Description VARCHAR2(500),
OrderID NUMBER(5),
CONSTRAINT invoices_OrderID FOREIGN KEY (orderid) REFERENCES
booking_order(OrderID));

```

```

CREATE TABLE clean_main_crews (
CrewID NUMBER(5) PRIMARY KEY,
Phone NUMBER(10),
Email VARCHAR2(30),
MaintenancePrice NUMBER(6),
MaintenanceName CHAR(30),
HouseID NUMBER(3),
CONSTRAINT clean_main_crews_HouseID FOREIGN KEY (houseid) REFERENCES
house(HouseID));

```

```

CREATE TABLE task
(TaskID NUMBER(5) PRIMARY KEY,
TaskDescription VARCHAR2(500),

```

```

ApplianceID NUMBER(5) NULL,
CrewID NUMBER(5),
CONSTRAINT task_ApplianceID FOREIGN KEY (applianceID) REFERENCES
appliances(ApplianceID),
CONSTRAINT task_crewid FOREIGN KEY (crewid) REFERENCES
clean_main_crews(CrewID));

```

```

CREATE TABLE reviews
(reviewid NUMBER(5) PRIMARY KEY,
Number_rating NUMBER(1),
Average_rating NUMBER(1),
Customer_name VARCHAR2(30),
Description VARCHAR2(500),
Date_posted DATE,
Title_phrase VARCHAR2(30),
Dateofstay DATE,
Customer_location VARCHAR2(30) NULL,
RenterID NUMBER(5),
CONSTRAINT reviews_renterid FOREIGN KEY (renterid) REFERENCES renter(renterid));

```

```

CREATE TABLE host
(HostName CHAR(30) PRIMARY KEY,
PhoneNumber NUMBER(10),
Email VARCHAR2(30));

```

-- Inserting House Data Values --

```

INSERT INTO house (houseid, description, location, photos, price, numberofbeds,
numberofbaths)
VALUES (111,'Southern Charm', '229 Cameron Loop', NULL, 1000.00, 06, 6);

```

```

INSERT INTO house (houseid, description, location, photos, price, numberofbeds,
numberofbaths)
VALUES (112, 'Big Bertha', '481 Esther Lane', NULL, 1249.99, 2, 8);

```

```

INSERT INTO house (houseid, description, location, photos, price, numberofbeds,
numberofbaths)
VALUES (113, 'Southern Comfort', '10 Ski Lane', NULL, 900, 9, 9);

```

```

INSERT INTO house (houseid, description, location, photos, price, numberofbeds,
numberofbaths)
VALUES (114, 'Comfy Cabin', '100 Drive Lane', NULL, 500, 4, 4);

```

```
INSERT INTO house (houseid, description, location, photos, price, numberofbeds,
numberofbaths)
```

```
VALUES (115, 'Happy Hut', '614 Lake View Lane', NULL, 400, 3, 1);
```

```
INSERT INTO house (houseid, description, location, photos, price, numberofbeds,
numberofbaths)
```

```
VALUES (116, 'Hazel"s House', '914 Nut Drive', NULL, 750.11, 6, 5);
```

```
INSERT INTO house (houseid, description, location, photos, price, numberofbeds,
numberofbaths)
```

```
VALUES (117, 'Pippa"s Palace', '10 Popper Lane', NULL, 1500, 10, 12);
```

```
INSERT INTO house (houseid, description, location, photos, price, numberofbeds,
numberofbaths)
```

```
VALUES (118, 'Millie"s Manor', '150 Golden Drive', NULL, 1300, 6, 6);
```

```
INSERT INTO house (houseid, description, location, photos, price, numberofbeds,
numberofbaths)
```

```
VALUES (119, 'Diamond Dock', '200 Ginger Street', NULL, 820, 5, 6);
```

```
INSERT INTO house (houseid, description, location, photos, price, numberofbeds,
numberofbaths)
```

```
VALUES (120, 'Mickey Mouse Clubhouse', '001 Daisy Drive', NULL, 1000, 8, 8);
```

```
-- Inserting Booking Order Data Values --
```

```
INSERT INTO booking_order (orderid, datebooked, checkindate, checkoutdate, invoicenum,
houseid)
```

```
VALUES (12345, '01-Dec-2020', '20-May-2021', '27-May-2021', 54321, 111);
```

```
INSERT INTO booking_order (orderid, datebooked, checkindate, checkoutdate, invoicenum,
houseid)
```

```
VALUES (12346, '17-Nov-2020', '4-Jun-2021', '6-Jun-2021', 54322, 111);
```

```
INSERT INTO booking_order (orderid, datebooked, checkindate, checkoutdate, invoicenum,
houseid)
```

```
VALUES (12347, '18-Jan-2021', '11-Jun-2021', '13-Jun-2021', 54323, 116);
```

```
INSERT INTO booking_order (orderid, datebooked, checkindate, checkoutdate, invoicenum,
houseid)
```

```
VALUES (12348, '3-Mar-2021', '18-Jun-2021', '20-Jun-2021', 54324, 114);
```

```
INSERT INTO booking_order (orderid, datebooked, checkindate, checkoutdate, invoicenumbr,
houseid)
VALUES (12349, '22-Oct-2020', '25-Jun-2021', '27-Jun-2021', 54325, 112);
```

```
INSERT INTO booking_order (orderid, datebooked, checkindate, checkoutdate, invoicenumbr,
houseid)
VALUES (12350, '14-Sep-2020', '2-Jul-2021', '4-Jul-2021', 54326, 118);
```

```
INSERT INTO booking_order (orderid, datebooked, checkindate, checkoutdate, invoicenumbr,
houseid)
VALUES (12351, '18-Feb-2021', '9-Jul-2021', '11-Jul-2021', 54327, 111);
```

```
INSERT INTO booking_order (orderid, datebooked, checkindate, checkoutdate, invoicenumbr,
houseid)
VALUES (12352, '4-Dec-2020', '16-Jul-2021', '18-Jul-2021', 54328, 114);
```

```
INSERT INTO booking_order (orderid, datebooked, checkindate, checkoutdate, invoicenumbr,
houseid)
VALUES (12353, '19-Nov-2020', '23-Jul-2021', '25-Jul-2021', 54329, 113);
```

```
INSERT INTO booking_order (orderid, datebooked, checkindate, checkoutdate, invoicenumbr,
houseid)
VALUES (12354, '30-Jan-2021', '30-Jul-2021', '2-Aug-2021', 54330, 118);
```

-- Inserting Appliances Data Values --

```
INSERT INTO appliances (applianceid, appliancetype, repaircost, datepurchased,
timesrepaired, costofpurchase, currentlybroken, houseid)
VALUES (56789, 'fridge', 300.00, '31-Oct-2020', 1, 1000.00, 'N', 111);
```

```
INSERT INTO appliances (applianceid, appliancetype, repaircost, datepurchased,
timesrepaired, costofpurchase, currentlybroken, houseid)
VALUES (56790, 'dryer', 200.00, '12-Mar-2021', 0, 900.00, 'N', 111);
```

```
INSERT INTO appliances (applianceid, appliancetype, repaircost, datepurchased,
timesrepaired, costofpurchase, currentlybroken, houseid)
VALUES (56791, 'washer', 250.00, '12-Mar-2021', 0, 800.00, 'N', 111);
```

```
INSERT INTO appliances (applianceid, appliancetype, repaircost, datepurchased,
timesrepaired, costofpurchase, currentlybroken, houseid)
VALUES (56792, 'oven', 400.00, '29-Jun-2019', 1, 1000.00, 'N', 111);
```

```
INSERT INTO appliances (applianceid, appliancetype, repaircost, datepurchased,
timesrepaired, costofpurchase, currentlybroken, houseid)
```

VALUES (56793, 'toaster', 100.00, '20-Aug-2017', 2, 250.00, 'N', 111);

INSERT INTO appliances (applianceid, appliancetype, repaircost, datepurchased, timesrepaired, costofpurchase, currentlybroken, houseid)

VALUES (56794, 'microwave', 120.00, '01-Sep-2018', 1, 250.00, 'N', 111);

INSERT INTO appliances (applianceid, appliancetype, repaircost, datepurchased, timesrepaired, costofpurchase, currentlybroken, houseid)

VALUES (56795, 'Freezer', 300.00, '31-Oct-2020', 1, 1000.00, 'N', 111);

INSERT INTO appliances (applianceid, appliancetype, repaircost, datepurchased, timesrepaired, costofpurchase, currentlybroken, houseid)

VALUES (56796, 'dishwasher', 275.00, '05-Oct-2020', 0, 800.00, 'N', 111);

INSERT INTO appliances (applianceid, appliancetype, repaircost, datepurchased, timesrepaired, costofpurchase, currentlybroken, houseid)

VALUES (56797, 'coffee maker', 100.00, '17-Feb-2022', 0, 350.00, 'N', 112);

INSERT INTO appliances (applianceid, appliancetype, repaircost, datepurchased, timesrepaired, costofpurchase, currentlybroken, houseid)

VALUES (56798, 'blender', 100.00, '17-Feb-2022', 0, 150.00, 'N', 112);

INSERT INTO appliances (applianceid, appliancetype, repaircost, datepurchased, timesrepaired, costofpurchase, currentlybroken, houseid)

VALUES (56799, 'Mixer', 110.00, '10-Feb-2022', 0, 225.00, 'N', 112);

-- Inserting Amenities Data Values --

INSERT INTO amenities (amenityid, description, houseid)

VALUES (31278, 'Private dock with 2 floors for boats and lounging', 111);

INSERT INTO amenities (amenityid, description, houseid)

VALUES (61429, 'Hot Tub', 111);

INSERT INTO amenities (amenityid, description, houseid)

VALUES (76094, 'Speaker System', 113);

INSERT INTO amenities (amenityid, description, houseid)

VALUES (65432, 'Beverage Fridge', 111);

INSERT INTO amenities (amenityid, description, houseid)

VALUES (78901, 'Ice Machine', 113);

```
INSERT INTO amenities (amenityid, description, houseid)
VALUES (65123, 'Outdoor Fireplace', 111);
```

```
INSERT INTO amenities (amenityid, description, houseid)
VALUES (54637, 'Large Grill', 113);
```

```
INSERT INTO amenities (amenityid, description, houseid)
VALUES (76290, 'Back Patio', 111);
```

```
INSERT INTO amenities (amenityid, description, houseid)
VALUES (82067, 'Garage', 111);
```

```
INSERT INTO amenities (amenityid, description, houseid)
VALUES (98075, 'Ping-Pong Table', 113);
```

-- Inserting Renter Data Values --

```
INSERT INTO renter
(RenterID,FirstName,LastName,Email,PhoneNumber,DateVisited,Costofcleaning,NumberOfGuests,OrderID)
VALUES (11111,'emma', 'kate','emmadygert@gmail.com',6142902117,
'20-May-2020',150,15,12345);
```

```
INSERT INTO renter
(RenterID,FirstName,LastName,Email,PhoneNumber,DateVisited,Costofcleaning,NumberOfGuests,OrderID)
VALUES (11112, 'hannah', 'bailey','hb@yahoo.com',6145675670, '04-JUN-2021',150,10,12346);
```

```
INSERT INTO renter
(RenterID,FirstName,LastName,Email,PhoneNumber,DateVisited,Costofcleaning,NumberOfGuests,OrderID)
VALUES (11113,'lily','smith','smithle@gmail.com',6788003000,'11-JUN-2021',150,13,12347);
```

```
INSERT INTO renter
(RenterID,FirstName,LastName,Email,PhoneNumber,DateVisited,Costofcleaning,NumberOfGuests,OrderID)
VALUES (11114,'ava','miller','avamiller1@gmail.com',9087654444,'18-JUN-2021',150,15,12348);
```

```
INSERT INTO renter
(RenterID,FirstName,LastName,Email,PhoneNumber,DateVisited,Costofcleaning,NumberOfGuests,OrderID)
```

```
VALUES (11115,'julian','smith','julian01@yahoo.com',8790657854,'25-JUN-2021',150,12,12349);
```

```
INSERT INTO renter
```

```
(RenterID,FirstName,LastName,Email,PhoneNumber,DateVisited,Costofcleaning,NumberOfGuests,OrderID)
```

```
VALUES (11116,'sally','walter','waltersally@gmail.com',6541239000,'02-JUL-2021',150,8,12350);
```

```
INSERT INTO renter
```

```
(RenterID,FirstName,LastName,Email,PhoneNumber,DateVisited,Costofcleaning,NumberOfGuests,OrderID)
```

```
VALUES (11117,'thomas','train','choochoo@yahoo.com',8008003131,'09-JUL-2021',150,14,12351);
```

```
INSERT INTO renter
```

```
(RenterID,FirstName,LastName,Email,PhoneNumber,DateVisited,Costofcleaning,NumberOfGuests,OrderID)
```

```
VALUES (11118,'david','will','davidw@gmail.com',8764520980,'06-JUL-2021',150,17,12352);
```

```
INSERT INTO renter
```

```
(RenterID,FirstName,LastName,Email,PhoneNumber,DateVisited,Costofcleaning,NumberOfGuests,OrderID)
```

```
VALUES
```

```
(11119,'andrew','diamond','diamond@gmail.com',6780000456,'23-JUL-2021',150,13,12353);
```

```
INSERT INTO renter
```

```
(RenterID,FirstName,LastName,Email,PhoneNumber,DateVisited,Costofcleaning,NumberOfGuests,OrderID)
```

```
VALUES (11120,'jill','mack','jillmack@yahoo.com',8007811654,'30-JUL-2021',150,7,12354);
```

```
-- Inserting Invoices Data Values --
```

```
INSERT INTO invoices (InvoiceNumber, Total, Tax, GrandTotal, DateOfBilling, Description, orderid)
```

```
VALUES (54321, 7150.00, 100.00, 7250.00, '01-June-2021', 'fully paid', 12345);
```

```
INSERT INTO invoices (InvoiceNumber, Total, Tax, GrandTotal, DateOfBilling, Description, orderid)
```

```
VALUES(54322, 10099.00, 150.00, 10249.00, '01-June-2021', 'partially paid', 12346);
```

```
INSERT INTO invoices (InvoiceNumber, Total, Tax, GrandTotal, DateOfBilling, Description, orderid)
```

```
VALUES(54323, 7150.00, 100.00, 7250.00, '01-June-2021', 'fully paid', 12347);
```



```
INSERT INTO invoices (InvoiceNumber, Total, Tax, GrandTotal, DateOfBilling, Description,
orderid)
```

```
VALUES(54324, 8000.00, 125.00, 8125.00, '01-July-2021', 'not paid', 12348);
```

```
INSERT INTO invoices (InvoiceNumber, Total, Tax, GrandTotal, DateOfBilling, Description,
orderid)
```

```
VALUES(54325, 11000.00, 150.00, 11150.00, '01-July-2021', 'partially paid', 12349);
```

```
INSERT INTO invoices (InvoiceNumber, Total, Tax, GrandTotal, DateOfBilling, Description,
orderid)
```

```
VALUES(54326, 10000.00, 140.00, 10140.00, '01-July-2021', 'fully paid', 12350);
```

```
INSERT INTO invoices (InvoiceNumber, Total, Tax, GrandTotal, DateOfBilling, Description,
orderid)
```

```
VALUES(54327, 7500.00, 100.00, 7600.00, '10-July-2021', 'not paid', 12351);
```

```
INSERT INTO invoices (InvoiceNumber, Total, Tax, GrandTotal, DateOfBilling, Description,
orderid)
```

```
VALUES(54328, 8000.00, 125.00, 8125.00, '20-July-2021', 'partially paid', 12352);
```

```
INSERT INTO invoices (InvoiceNumber, Total, Tax, GrandTotal, DateOfBilling, Description,
orderid)
```

```
VALUES(54329, 7150.00, 100.00, 7250.00, '15-July-2021', 'fully paid', 12353);
```

```
INSERT INTO invoices (InvoiceNumber, Total, Tax, GrandTotal, DateOfBilling, Description,
orderid)
```

```
VALUES(54330, 9000.00, 130.00, 9130.00, '15-July-2021', 'fully paid', 12354);
```

```
-- Inserting Cleaning and Maintenance Crews Data Values --
```

```
INSERT INTO clean_main_crews (crewid, phone, email, maintenanceprice, houseid,
maintenancename)
```

```
VALUES (11111, 9001239876, 'janice@cleaning.com', 75.00, 111, 'Janice Conrad');
```

```
INSERT INTO clean_main_crews (crewid, phone, email, maintenanceprice, houseid,
maintenancename)
```

```
VALUES (22222, 9001239876, 'janice@cleaning.com', 75.00, 112, 'Janice Conrad');
```

```
INSERT INTO clean_main_crews (crewid, phone, email, maintenanceprice, houseid,
maintenancename)
```

```
VALUES (33333, 9009876543, 'builder@building.com', 125.00, 113, 'Bob Builder');
```

```
INSERT INTO clean_main_crews (crewid, phone, email, maintenanceprice, houseid,
maintenancename)
VALUES (44444, 9009876543, 'builder@building.com', 125.00, 114, 'Bob Builder');
```

```
INSERT INTO clean_main_crews (crewid, phone, email, maintenanceprice, houseid,
maintenancename)
VALUES (55555, 9001239876, 'janice@cleaning.com', 75.00, 115, 'Janice Conrad');
```

```
INSERT INTO clean_main_crews (crewid, phone, email, maintenanceprice, houseid,
maintenancename)
VALUES (66666, 9001239876, 'janice@cleaning.com', 75.00, 116, 'Janice Conrad');
```

```
INSERT INTO clean_main_crews (crewid, phone, email, maintenanceprice, houseid,
maintenancename)
VALUES (77777, 9001239876, 'janice@cleaning.com', 75.00, 117, 'Janice Conrad');
```

-- Inserting Task Data Values --

```
INSERT INTO task (taskid, taskdescription, applianceid, crewid)
VALUES (33333, 'fridge deep clean', 56789, 22222);
```

```
INSERT INTO task (taskid, taskdescription, applianceid, crewid)
VALUES (44444, 'change lightbulbs', 56790, 22222);
```

```
INSERT INTO task (taskid, taskdescription, applianceid, crewid)
VALUES (55555, 'Dishwasher Cleaning', 56796, 11111);
```

```
INSERT INTO task (taskid, taskdescription, applianceid, crewid)
VALUES (66666, 'Dryer Repair', 56790, 11111);
```

```
INSERT INTO task (taskid, taskdescription, applianceid, crewid)
VALUES (77777, 'washer repair', 56791, 22222);
```

```
INSERT INTO task (taskid, taskdescription, applianceid, crewid)
VALUES (88888, 'oven cleaning', 56792, 22222);
```

```
INSERT INTO task (taskid, taskdescription, applianceid, crewid)
VALUES (99999, 'freezer repair', 56795, 22222);
```

-- Inserting Reviews Data Values --

```
INSERT INTO reviews (reviewid, number_rating, average_rating, customer_name, description,
date_posted, title_phrase, dateofstay, customer_location, renterid)
VALUES(54321, 5, 4.4, 'emma kate dygert', 'beautiful view!', '30-May-2021', 'great', '20-May-2021', 'Ohio',
11111);
```

```
INSERT INTO reviews (reviewid, number_rating, average_rating, customer_name, description,
date_posted, title_phrase, dateofstay, customer_location, renterid)
VALUES(54322, 4, 4.4, 'hannah bailey', 'a lot of fun!', '02-June-2021', 'fun', '27-May-2021', 'Kentucky',
11112);
```

```
INSERT INTO reviews (reviewid, number_rating, average_rating, customer_name, description,
date_posted, title_phrase, dateofstay, customer_location, renterid)
VALUES(54323, 5, 4.4, 'lily smith', 'excellent experience', '02-June-2021', 'excellent', '27-May-2021',
'Indiana', 11113);
```

```
INSERT INTO reviews (reviewid, number_rating, average_rating, customer_name, description,
date_posted, title_phrase, dateofstay, customer_location, renterid)
VALUES(54324, 3, 4.4, 'ava miller', 'could be better', '10-June-2021', 'unsatisfied', '02-June-2021', 'Ohio',
11114);
```

```
INSERT INTO reviews (reviewid, number_rating, average_rating, customer_name, description,
date_posted, title_phrase, dateofstay, customer_location, renterid)
VALUES(54325, 5, 4.4, 'julian smith', 'was a great stay', '28-June-2021', 'great', '15-June-2021',
'Tennessee', 11115);
```

-- Inserting Host Data Values --

```
INSERT INTO host (hostname, phonenumber, email)
VALUES('john dygert', 6143238831, 'john@johndygert.com');
```

```
INSERT INTO host (hostname, phonenumber, email)
VALUES('alison dygert', 6143234701, 'alisondygert@gmail.com');
```

```
INSERT INTO host (hostname, phonenumber, email)
VALUES('emma kate dygert', 6142902117, 'emmadygert@gmail.com');
```

```
INSERT INTO host (hostname, phonenumber, email)
VALUES('jackson dygert', 6147609422, 'jacksondygert@gmail.com');
```

```
INSERT INTO host (hostname, phonenumber, email)
VALUES('hayden dygert', 6144301717, 'haydendygert@gmail.com');
```