Database Systems Project

Anna Dubé

Athens State University

adube@my.athens.edu

00091750

July 18, 2021

Table of Contents

I. Assumptions

This section includes all the assumptions made about a company that rents river equipment. These assumptions are used to determine the data needed for the data dictionary and the design of the user views. The scope of the project depends on the assumptions made here.

- I assumed this is a small company and equipment can be tracked visually as well as programmatically. I assume that a larger company would have to track more data.
- I assumed the insurance is going to be a flat percentage across all the equipment.
- Figure 2 does not include the fishing equipment and the life jackets because I assumed these items would be selected for rental after the other equipment has been selected.
- I assumed that the Reservation starts when the first equipment rental starts and stops when the last equipment rental ends.

II. Users and Mock Designs

This section goes over the views associated to the different users. There are two types of users: customers and administrators. A customer should only have access to their reservation information, the available equipment for rentals, and the surveys. An administrator should have access to employee, customer, equipment, ad, survey, and reservation information. An administrator should also be able to edit that information.

A. Customer View

Figure 1 shows the home page for Water Way Zen's website. From here, customers can access information on the company's equipment and rental prices.

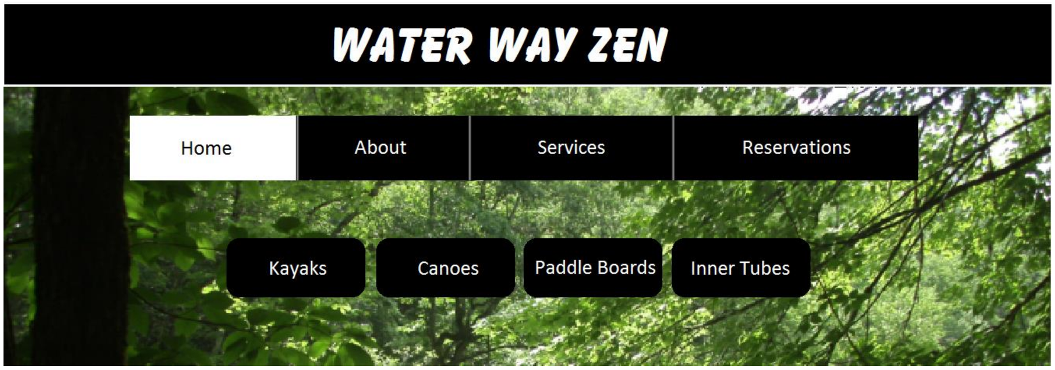
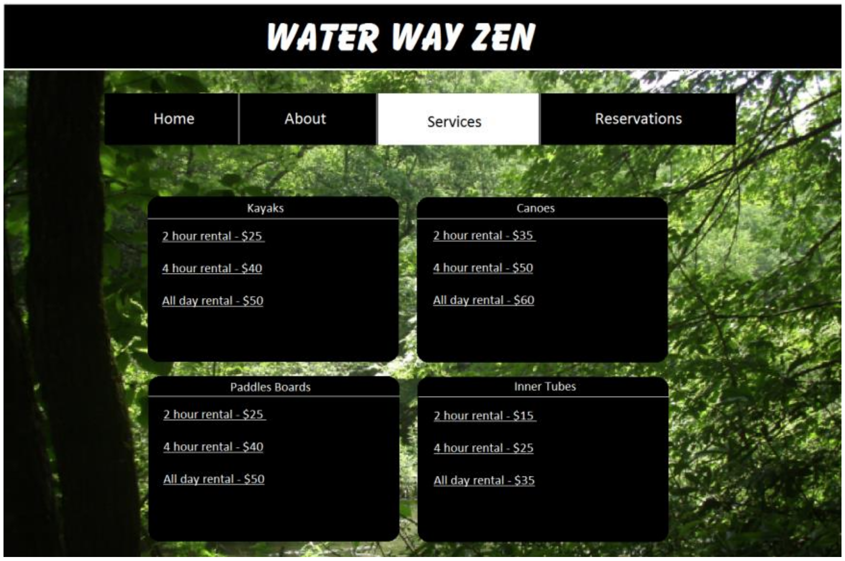


Figure 1. Customer View: Home Page

Figure 2 shows the Services page which allows a customer to view the rental prices on the company's equipment. A customer can select a piece of equipment and view the availability of that type of equipment. Three common rental times are listed to make it easier for a customer to make a reservation. The customer can select the equipment and time limit they want and then make a reservation.



Kayaks

2 hour rental - $25

4 hour rental - $40

All day rental - $50

Canoes

2 hour rental - $35

4 hour rental - $50

All day rental - $60

Paddles Boards

2 hour rental - $25

4 hour rental - $40

All day rental - $50

Inner Tubes

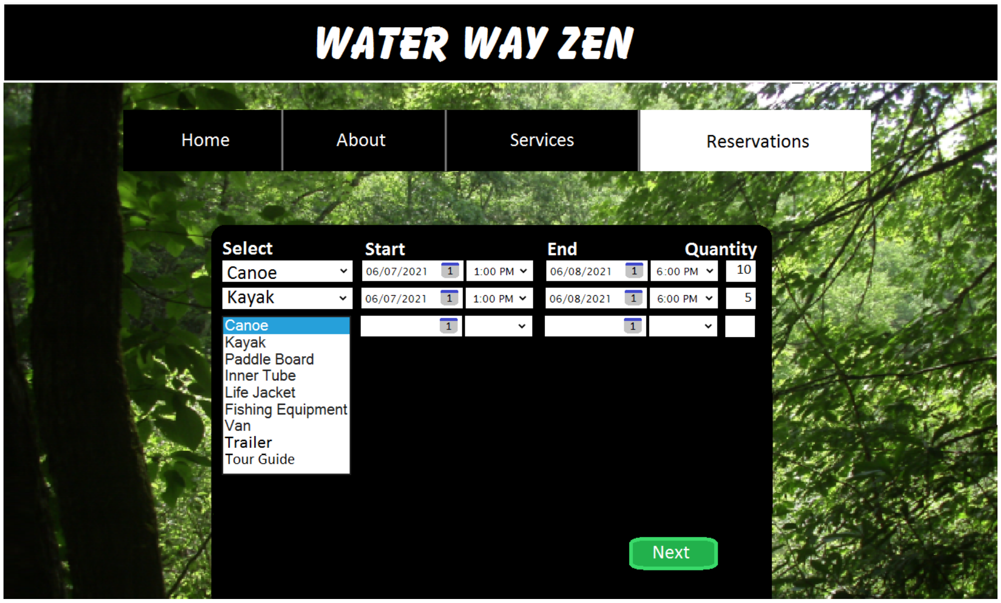
2 hour rental - $15

4 hour rental - $25

All day rental - $35

Figure 2. Customer View: Equipment and Services Information

If a customer selects a time limit and type of equipment from the Services page or if the customer selects the Reservation tab, they are taken to the page shown in figure 3. Here customers can select the equipment they want, enter in their own time limits, and enter the amount of each type of equipment they want to rent. This is also where customers can select life jackets, fishing equipment, vans, and tour guides.



Select	Start		End		Quantity
Canoe	06/07/2021	1:00 PM	06/08/2021	6:00 PM	10
Kayak	06/07/2021	1:00 PM	06/08/2021	6:00 PM	5

Figure 3. Customer View: Equipment Rental

After customers choose their equipment, they are taken to the page shown in figure 4. A customer has to provide their name, address, email address, and phone number. This data will be stored in the database.

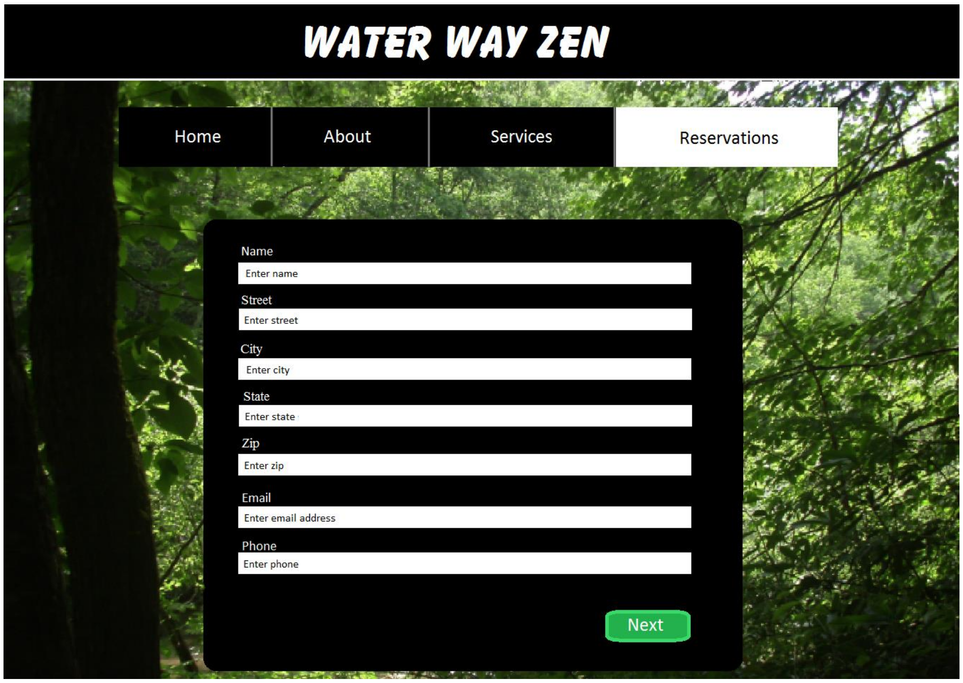
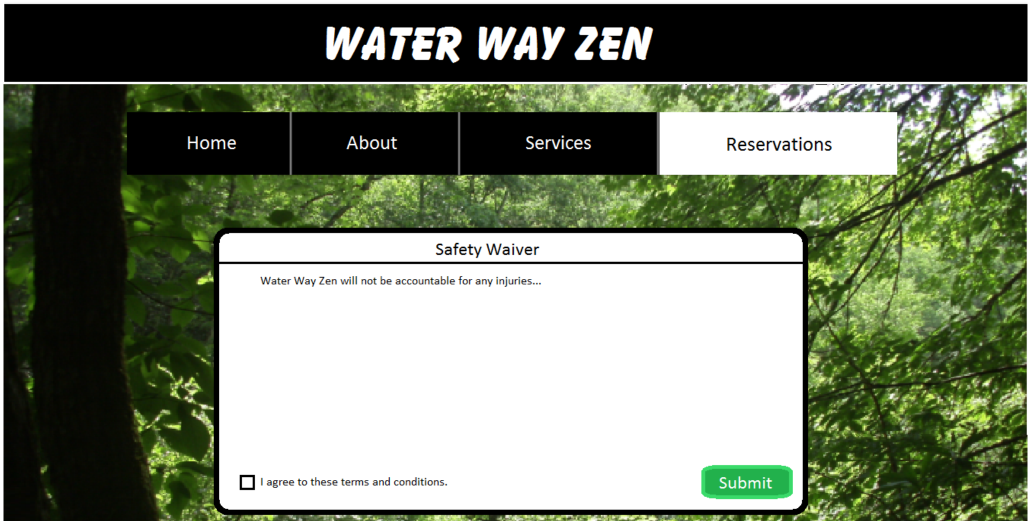


Figure 4. Customer View: Reservations

After customers select their equipment and enter their personal information, they have to fill out the safety waiver shown in figure 5.

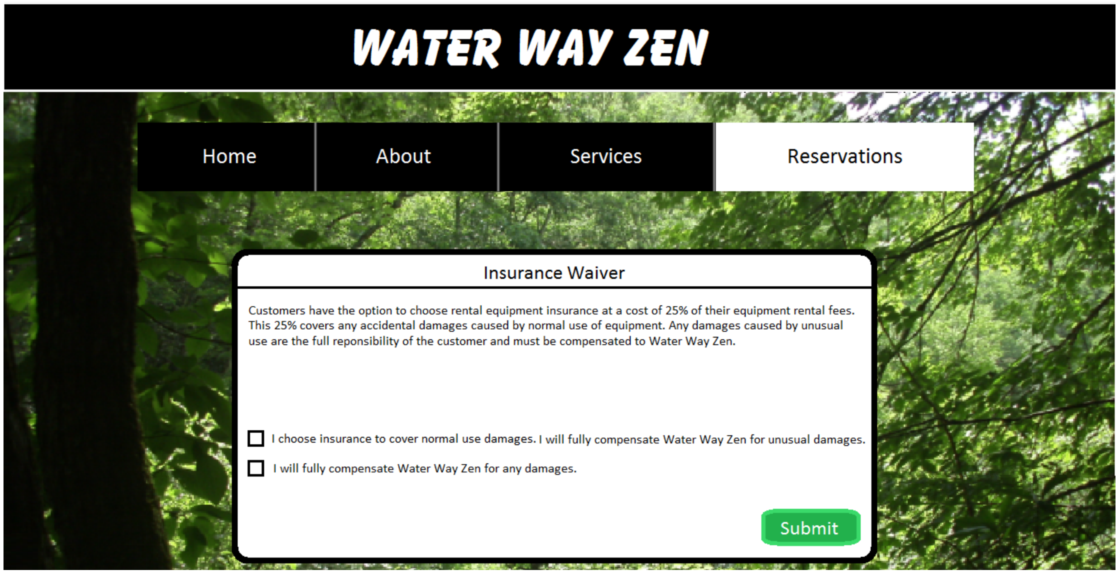


Water Way Zen will not be accountable for any injuries...

□ I agree to these terms and conditions.

Figure 5. Customer View: Safety Waiver

Figure 6 shows the insurance waiver that a customer will sign if they do not choose the option of purchasing insurance.



Insurance Waiver

Customers have the option to choose rental equipment insurance at a cost of 25% of their equipment rental fees. This 25% covers any accidental damages caused by normal use of equipment. Any damages caused by unusual use are the full reponsibility of the customer and must be compensated to Water Way Zen.

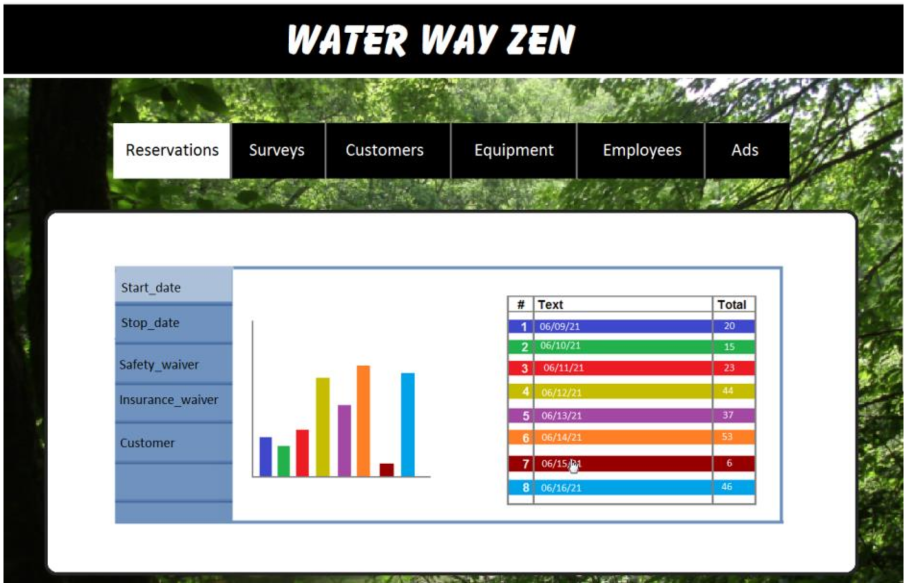
□ I choose insurance to cover normal use damages. I will fully compensate Water Way Zen for unusual damages.

□ I will fully compensate Water Way Zen for any damages.

Figure 6. Customer View: Insurance Waiver

B. *Administrator's View*

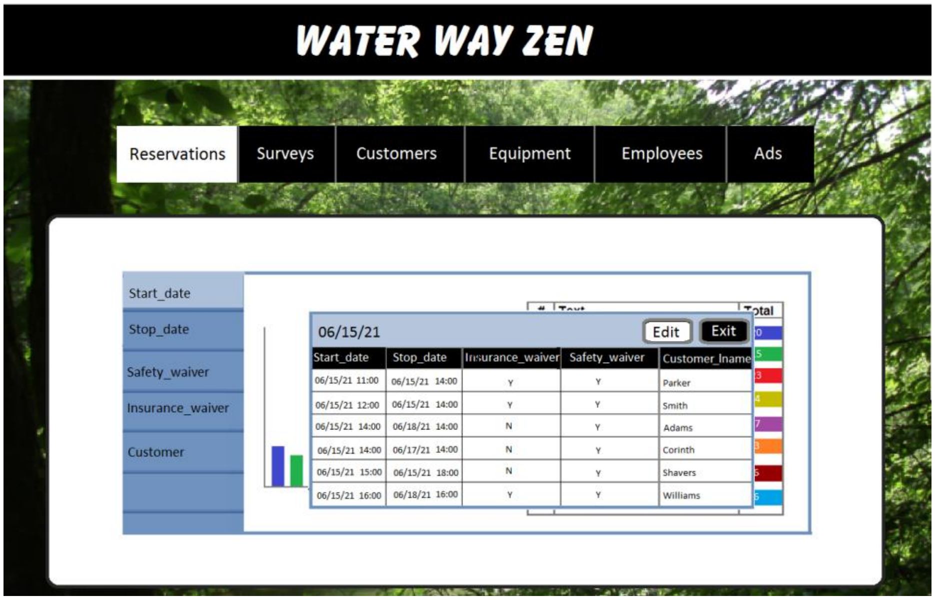
Figure 7 shows the administrator's view. An administrator can select a tab to view information about the company. Figure 7 shows the Reservations tab is selected and the reservations data is being displayed by Start_date. The figure also shows that the dates listed in the table can be selected to view reservations data for that particular day.



#	Text	Total
1	06/09/21	20
2	06/10/21	15
3	06/11/21	23
4	06/12/21	44
5	06/13/21	37
6	06/14/21	53
7	06/15/21	6
8	06/16/21	46

Figure 7. Administrator View: Reservations

Figure 8 shows a date has been selected from the table in figure 7. When a date is selected, a new window appears, displaying the data for that date. The administrator is also given the option to edit the data.



Start_date	Stop_date	Insurance_waiver	Safety_waiver	Customer_lname
06/15/21 11:00	06/15/21 14:00	Y	Y	Parker
06/15/21 12:00	06/15/21 14:00	Y	Y	Smith
06/15/21 14:00	06/18/21 14:00	N	Y	Adams
06/15/21 14:00	06/17/21 14:00	N	Y	Corinth
06/15/21 15:00	06/15/21 18:00	N	Y	Shavers
06/15/21 16:00	06/18/21 16:00	Y	Y	Williams

Figure 8. Administrator View: Reservations

III. Data Dictionary

This section goes over the data dictionary. The data dictionary is used to determine all the data that an organization needs to track. It includes a data label, a data description, a data type, and the owner. The data label is the name given for the data. The data description describes what data is being stored. The data type determines what type of data can be stored. The owner refers to the table each attribute belongs to.

The data dictionary in Table 1 shows all of the data the company, Water Way Zen, will need to track. This data is used to keep track of employees, customers, reservations, equipment, ads, and surveys. Each data label corresponds with a specific owner or table in the database.

Table 1. Data Dictionary

Data Label	Description	Data Type	Owner
Employee_id	Unique number to identify an employee.	Integers(8)	EMPLOYEE
Employee_fname	An employee’s first name.	Characters(15)	EMPLOYEE
Employee_mname	An employee’s middle name.	Characters(15)	EMPLOYEE
Employee_lname	An employee’s last name.	Characters(15)	EMPLOYEE

Employee_gender	This is an employee's gender.	Character(1)	EMPLOYEE
Employee_street	The street name in an employee's address.	Characters(30)	EMPLOYEE
Employee_city	The city name in an employee's address.	Characters(15)	EMPLOYEE
Employee_state	The state in an employee's address.	Characters(2)	EMPLOYEE
Employee_zip	The zip code in an employee's address.	Integers(5)	EMPLOYEE
Employee_cell_phone	An employee's cell phone number.	Varchar(13)	EMPLOYEE
Employee_home_phone	An employee's home phone number.	Varchar(13)	EMPLOYEE
Employee_email	An employee's email.	Varchar(30)	EMPLOYEE
Employee_pay	This is the hourly pay rate of an employee.	Decimal(5)	EMPLOYEE
Employee_dob	This is the employee's birthdate	Date	EMPLOYEE
Date_hired	This is the date an employee was hired.	Date	EMPLOYEE
Customer_id	Unique number to identify a customer.	Integers(8)	CUSTOMER
Customer_fname	This is a customer's first name.	Characters(15)	CUSTOMER
Customer_mname	A customer's middle name.	Characters(15)	CUSTOMER
Customer_lname	This is a customer's last name.	Characters(15)	CUSTOMER
License_no	This is the employee's driver license number.	Integer(8)	EMPLOYEE
Customer_gender	This is a customer's gender.	Character(1)	CUSTOMER
Customer_street	The street name in a customer's address.	Characters(30)	CUSTOMER
Customer_city	The city name in a customer's address.	Characters(15)	CUSTOMER
Customer_state	The state in a customer's address.	Characters(2)	CUSTOMER
Customer_zip	The zip code in a customer's address.	Integers(5)	CUSTOMER
Customer_phone	A customer's phone number.	Varchar(13)	CUSTOMER
Customer_email	A customer's email.	Varchar(30)	CUSTOMER
Reservation_id	Unique number to identify a reservation that has been made.	Integers(8)	RESERVATION
Start_date	The date and time a specific reservation starts.	Date	RESERVATION
Stop_date	The date and time a specific reservation stops.	Date	RESERVATION
Number_in_party	This is the number of people in the reservation.	Integer(3)	RESERVATION

Res_equip_quantity	The quantity of the specific equipment being rented.	Integers(3)	RESERVATION
Estimate	This is the estimate of the cost.	Decimal(6)	RESERVATION
Paid_amount	This is the actual amount paid.	Decimal(6)	RESERVATION
Equipment_id	Unique number to identify equipment.	Integers(8)	EQUIPMENT
Equipment_type	The type of equipment being rented.	Characters(15)	EQUIPMENT
Equipment_in	This stores whether or not equipment is currently rented. This is the availability of a specific piece of equipment.	Characters(3)	EQUIPMENT
License_no	The license number of a company van.	Varchar(20)	VAN
Max_passenger_no	The amount of passengers each van can hold.	Integers(2)	VAN
Registration_no	The registration number for a van.	Integers(15)	VAN
Registration_date	The date the van was registered.	Date	VAN
Vehicle_id	This is the vehicle identification.	Varchar(20)	VAN
Equipment_condition	This stores the condition of a piece of equipment.	Characters(10)	EQUIPMENT
In_service_date	The date equipment last went into service.	Date	EQUIPMENT
Out_service_date	The date equipment last went out of service.	Date	EQUIPMENT
State_registered	This is the state a van was registered in.	Characters(2)	VAN
Manufacturer	This is the name of the manufacturer.	Characters(30)	VAN
Model	This is the name of the model.	Characters(30)	VAN
Purchase_price	The purchase price of each piece of equipment.	Decimal(7)	EQUIPMENT
Depreciation_by_day	The change in the value of equipment.	Decimal(5)	EQUIPMENT
Purchase_date	The date equipment was bought.	Date	EQUIPMENT
Rental_price	This is the price for renting a piece of equipment.	Decimal(6)	EQUIPMENT
Ad_id	Unique number to identify an ad used.	Integers(8)	ADVERTISEMENT
Ad_start_date	The date an ad started.	Date	ADVERTISEMENT
Ad_end_date	The date an ad ended.	Date	ADVERTISEMENT
Ad_content	The content of an ad.	Varchar(max)	ADVERTISEMENT
Media	Type of the ad.	Characters(15)	ADVERTISEMENT

Location_printed	The location where an ad was printed.	Characters(30)	ADVERTISEMENT
Ad_cost	The cost of an ad.	Decimal(6)	ADVERTISEMENT
Frequency	The frequency an ad appears.	Decimal(6)	ADVERTISEMENT
Contact_fname	The first name of the contact person.	Characters(30)	ADVERTISEMENT
Contact_lname	The last name of the contact person.	Characters(30)	ADVERTISEMENT
Contact_phone	The contact person's phone number.	Varchar(13)	ADVERTISEMENT
Promotion_code	This is the promotion code for an ad.	Varchar(15)	ADVERTISEMENT
Survey_id	Unique number to identify a survey.	Integers(8)	SURVEY
Emp_rating	The customer rating of employee performance.	Integer(1)	SURVEY
Equip_rating	The customer rating of the equipment.	Integer(1)	SURVEY
Cust_rating	The customer rating of the experience.	Integer(1)	SURVEY
Saftey_waiver	A waiver that has to be signed by the customer during the reservation process. The only thing stored is whether the customer agreed to the conditions or not (Y/N).	Characters(1)	WAIVER
Waiver_date_signed	This is the date a safety waiver was signed.	Date	WAIVER
Waiver_id	This is a unique number to identify a signed safety waiver.	Integers(8)	WAIVER
Insurance_id	This is a unique number to identify an insurance payment.	Integers(8)	INSURANCE
Days_covered	This is the number of days the insurance covers.	Integer(2)	INSURANCE
Total_paid	This is the amount paid for the insurance.	Decimal(5)	INSURANCE
Date_purchased	This is the date the insurance was purchased.	Date	INSURANCE

Table 2 shows all the attributes that were removed from the data dictionary.

Table 2. Data removed from the data dictionary

Data Label	Data Description	Data Type
Question_id	A number used to reference a question.	Integers(4)

Question_text	A question on a survey.	Characters(256)
Answer_id	A number used to reference a customer’s answer.	Characters(4)
Answer_text	An answer to a question on the survey.	Varchar(max)
Insurance_waiver	A waiver that is signed by the customer during the reservation process. The only thing stored for this attribute is whether the customer agreed bought insurance or not (Y/N).	Characters(1)
Res_equip_start	The start time of the specific equipment rental.	Date
Res_equip_stop	The stop time of the specific equipment rental.	Date

IV. ERD

The Entity Relationship Diagram shows how the different tables in the database are related. The ERD for Water Way Zen has nine entities: CUSTOMER, EMPLOYEE, RESERVATION, EQUIPMENT, SURVEY, INSURANCE, WAIVER, VAN and ADVERTISEMENT.

The ERD for Water Way Zen is shown in figure 9. An EMPLOYEE can work on zero to many RESERVATIONs depending on the type of EMPLOYEE. A RESERVATION includes one EMPLOYEEs. A CUSTOMER makes at least one RESERVATION, and a RESERVATION is made by one CUSTOMER. A RESERVATION can be made from a person following an ADVERTISEMENT, and an ADVERTISEMENT can be associated with zero to many RESERVATIONs. An ADVERTISEMENT is can be viewed by many CUSTOMERs, and a CUSTOMER can view one ADVERTISEMENT. A RESERVATION includes one to many pieces of EQUIPMENT, and a piece of EQUIPMENT belongs to one RESERVATION at a time. A SURVEY is associated with a RESERVATION, and a RESERVATION can contain one SURVEY. A SURVEY is completed by one customer, and a CUSTOMER may complete zero to one SURVEYs per RESERVATION. This is an optional one because a CUSTOMER might not complete the SURVEY. An INSURANCE payment belongs to a RESERVATION, and a RESERVATION can contain at the most one INSURANCE payment. This is also an optional one because INSURANCE might not be purchased. A safety WAIVER belongs to one RESERVATION, and a RESERVATION contains one safety WAIVER. A CUSTOMER signs a safety WAIVER, and WAIVER is signed by one CUSTOMER. A VAN is driven by one EMPLOYEE, and an EMPLOYEE can drive any of the VANs. A VAN is part of a RESERVATION, and RESERVATION may include multiple VANs.

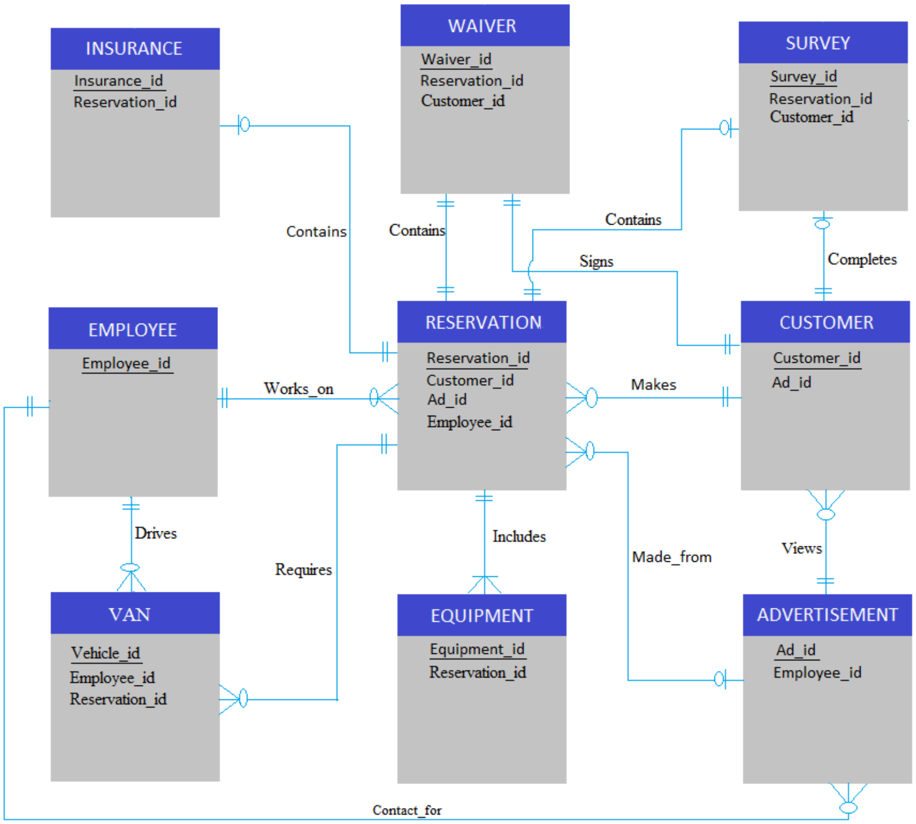


Figure 9. The ERD for Water Way Zen

V. E-ERD

The ERD for Water Way Zen can be enhanced in a few ways. Specializations of some entities can be added to enhance the diagram. Two areas of the diagram that can be enhanced are the EQUIPMENT and EMPLOYEE entities.

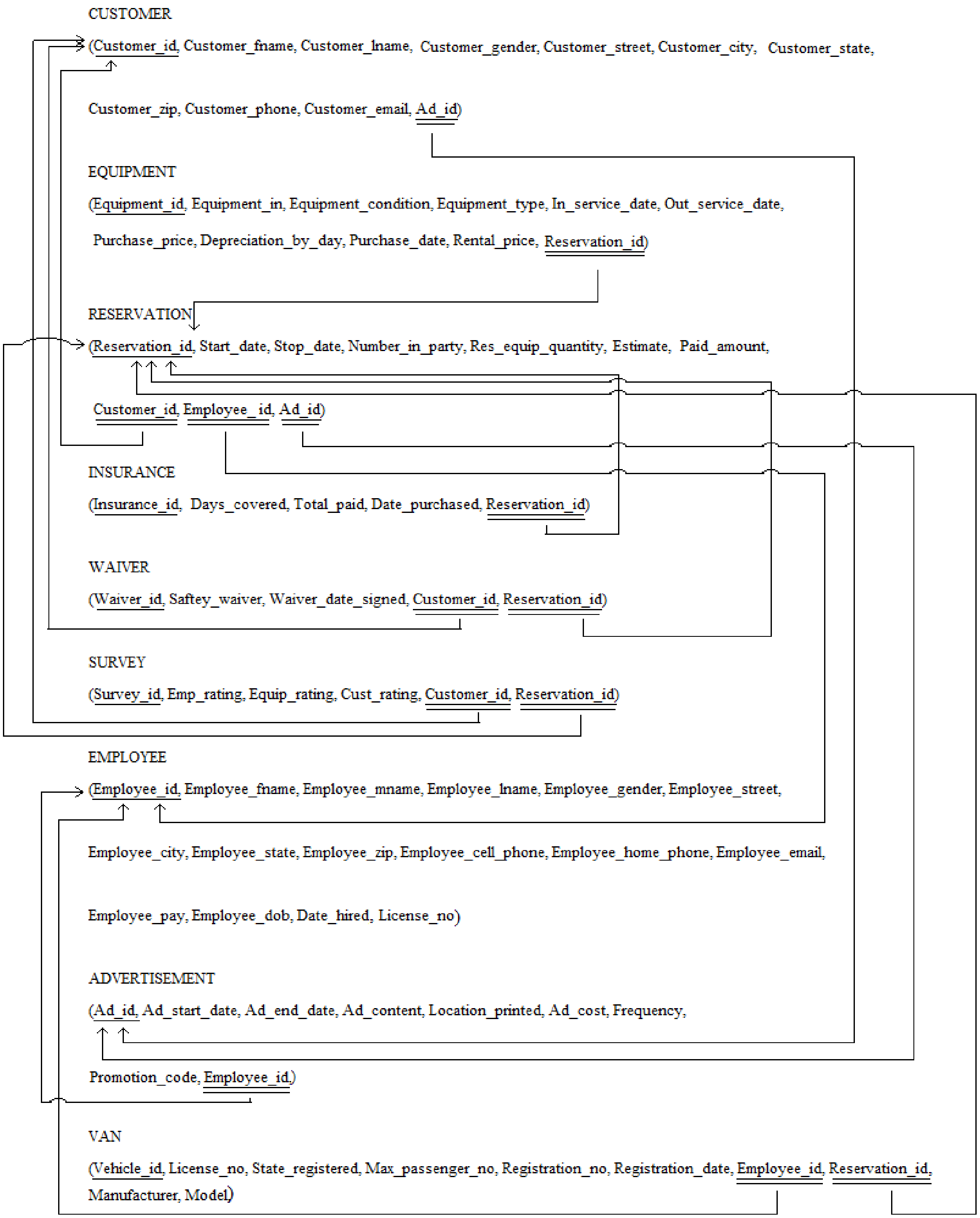
EQUIPMENT is a generalized term, and the diagram could be changed to include subclasses of EQUIPMENT. The subclasses would include the different categories of EQUIPMENT that Water Way Zen has which are: paddle boards, kayaks, canoes, inner tubes, life jackets, and fishing equipment. This enhancement will improve the ERD because not all of the equipment types have the same attributes.

The EMPLOYEE entity can be broken down into the different types of employees, like tour guides and van drivers. Like the EQUIPMENT enhancements this improves the ERD because not all EMPLOYEEs have the same attributes. For example, a van driver may have to have a driver's license.

VI. Relational Database Model

A relational database model shows the relationships between the different entities in a database. This is how the tables will be linked together in the database. Two tables are linked together by placing the primary key of one entity into the other entity as a foreign key. A line is drawn from the primary key to the foreign key to show how the two tables are related.

Figure 10 shows the relational model for Water Way Zen. This model shows how the nine entities for the company's database are related to each other. It shows the primary keys and the foreign keys with referential integrity lines along with the other attributes for each entity.



CUSTOMER

(Customer_id, Customer_fname, Customer_lname, Customer_gender, Customer_street, Customer_city, Customer_state,

Customer_zip, Customer_phone, Customer_email, Ad_id)

EQUIPMENT

(Equipment_id, Equipment_in, Equipment_condition, Equipment_type, In_service_date, Out_service_date,

Purchase_price, Depreciation_by_day, Purchase_date, Rental_price, Reservation_id)

RESERVATION

(Reservation_id, Start_date, Stop_date, Number_in_party, Res_equip_quantity, Estimate, Paid_amount,

Customer_id, Employee_id, Ad_id)

INSURANCE

(Insurance_id, Days_covered, Total_paid, Date_purchased, Reservation_id)

WAIVER

(Waiver_id, Saftey_waiver, Waiver_date_signed, Customer_id, Reservation_id)

SURVEY

(Survey_id, Emp_rating, Equip_rating, Cust_rating, Customer_id, Reservation_id)

EMPLOYEE

(Employee_id, Employee_fname, Employee_mname, Employee_lname, Employee_gender, Employee_street,

Employee_city, Employee_state, Employee_zip, Employee_cell_phone, Employee_home_phone, Employee_email,

Employee_pay, Employee_dob, Date_hired, License_no)

ADVERTISEMENT

(Ad_id, Ad_start_date, Ad_end_date, Ad_content, Location_printed, Ad_cost, Frequency,

Promotion_code, Employee_id)

VAN

(Vehicle_id, License_no, State_registered, Max_passenger_no, Registration_no, Registration_date, Employee_id, Reservation_id,

Manufacturer, Model)

Figure 10. Relational Model

VII. Functional Dependencies

This section shows the functional dependencies that exist between the different attributes listed in the data dictionary. These functional dependencies place constraints on the possible tuples that can be formed with the attributes. The candidate keys are placed on the left side of the arrow, and the dependent attributes are on the right. For example, the first dependency shows that all of the customer information is dependent on the Customer_id.

FD 1: Customer_id-> Customer_fname, Customer_lname, Customer_gender, Customer_street, Customer_city, Customer_state, Customer_zip, Customer_phone, Customer_email, Ad_id

FD 2: Equipment_id-> Equipment_in, Equipment_condition, Equipment_type, In_service_date, Out_service_date, Purchase_price, Depreciation_by_day, Purchase_date, Rental_price, Reservation_id

FD 3: Reservation_id-> Start_date, Stop_date, Number_in_party, Res_equip_quantity, Estimate, Paid_amount, Customer_id, Employee_ id, Ad_id

FD 4: Insurance_id-> Days_covered, Total_paid, Date_purchased, Reservation_id

FD 5: Waiver_id-> Saftey_waiver, Waiver_date_signed, Customer_id, Reservation_id

FD 6: Survey_id-> Emp_rating, Equip_rating, Cust_rating, Customer_id, Reservation_id

FD 7: Employee_id-> Employee_fname, Employee_mname, Employee_lname, Employee_gender, Employee_street, Employee_city, Employee_state, Employee_zip, Employee_cell_phone, Employee_home_phone, Employee_email, Employee_pay, Employee_dob, Date_hired, License_no

FD 8: Ad_id-> Ad_start_date, Ad_end_date, Ad_content, Location_printed, Ad_cost, Frequency, Promotion_code, Employee_id

FD 9: Vehicle_id-> License_no, State_registered, Max_passenger_no, Registration_no, Registration_date, Employee_id, Reservation_id, Manufacturer, Model

VIII. Normalization

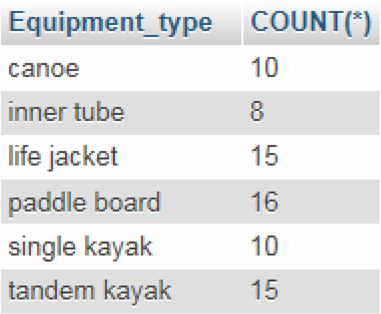
Normalization is the process of breaking up attributes into smaller and simpler relations. The attributes from the data dictionary have been normalized into the nine relations from the ERD. This database is already in second normal form because there are no multivalued attributes, and there are no partial functional dependencies. There are no transitive dependencies, so the database is also in third normal form.

IX. Queries

This section includes eleven queries that were performed against the database. The figures contain the SQL code on the left side and the resulting table on the right side.

Figure 11 shows the result of the first query. The number of times each different type of equipment is reserved is calculated and displayed in the resulting table. To get an accurate count, the reservation table and the equipment table were joined together and the rows were grouped by the type of equipment. The type along with its count was selected and displayed.

```
SELECT DISTINCT(Equipment_type), COUNT(*)
FROM reservation r JOIN equipment e ON
r.Reservation_id = e.Reservation_id
GROUP BY Equipment_type;
```

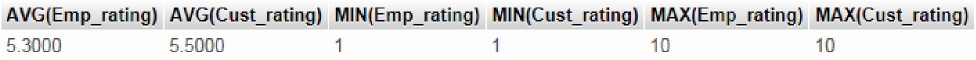


Equipment_type	COUNT(*)
canoe	10
inner tube	8
life jacket	15
paddle board	16
single kayak	10
tandem kayak	15

Figure 11. Query 1

Figure 12 shows a query that returns the average, minimum, and maximum employee and overall ratings that were entered by customers. The average, minimum, and maximum were calculated for each of the columns selected from the survey table.

```
SELECT
AVG(Emp_rating),
AVG(Cust_rating),
MIN(Emp_rating),
MIN(Cust_rating),
MAX(Emp_rating),
MAX(Cust_rating)
FROM survey;
```

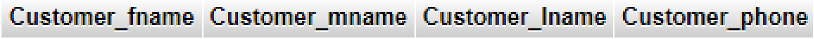


AVG(Emp_rating)	AVG(Cust_rating)	MIN(Emp_rating)	MIN(Cust_rating)	MAX(Emp_rating)	MAX(Cust_rating)
5.3000	5.5000	1	1	10	10

Figure 12. Query 2

Figure 13 shows query number three. The resulting table includes all of the customers who placed the minimum rating on all three categories on the waiver. There are no current customers who have placed the minimum rating on all of the categories, so the resulting table is empty. In the query, the survey table is joined to the customer table, so that the customer names can be selected. In the where clause, each of the customer's ratings are checked to see if they are the minimum rating.

```
SELECT Customer_fname,
Customer_mname,
Customer_lname, Customer_phone
FROM survey JOIN customer ON
survey.Customer_id =
customer.Customer_id
WHERE Equip_rating =
```



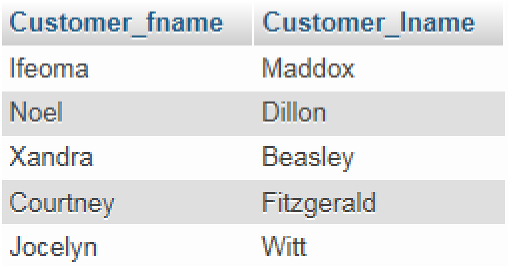
Customer_fname	Customer_mname	Customer_lname	Customer_phone

```
            (SELECT
MIN(Equip_rating)
   FROM survey)
   AND Emp_rating =
   (SELECT MIN(Emp_rating)
   FROM survey)
   AND Cust_rating =
   (SELECT MIN(Cust_rating)
            FROM survey);
```

Figure 13. Query 3

The next query, shown in figure 14, returns the names of customers who have not signed the safety waiver. The customer table is joined to the waiver table, so the customer names can be selected if the condition in the where clause is true.

```
SELECT Customer_fname, Customer_lname
FROM waiver JOIN customer ON waiver.Customer_id
= customer.Customer_id
WHERE Safety_waiver = 'no';
```

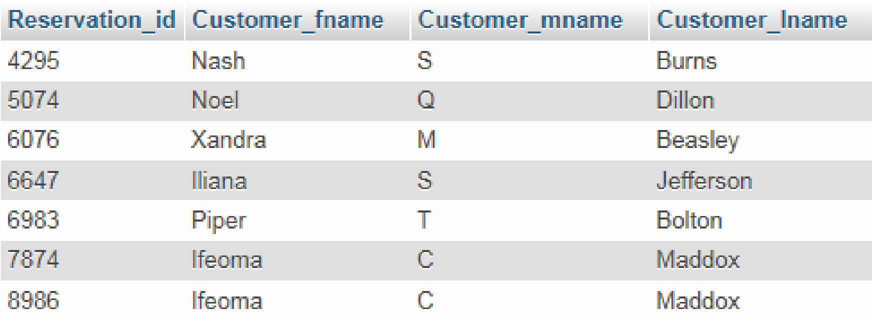


Customer_fname	Customer_lname
Ifeoma	Maddox
Noel	Dillon
Xandra	Beasley
Courtney	Fitzgerald
Jocelyn	Witt

Figure 14. Query 4

Figure 15 shows a query that joins the tables: reservation, customer, and insurance. The result of the join will include information from these three tables for all of the customers who have placed reservations and purchased insurance with those reservations. The query will return only the reservation number and the customer name from the resulting join operation.

```
SELECT r.Reservation_id,
Customer_fname,
Customer_mname,
Customer_lname
FROM customer c JOIN
reservation r ON
c.Customer_id =
r.Customer_id
JOIN insurance i ON
r.Reservation_id =
i.Reservation_id;
```

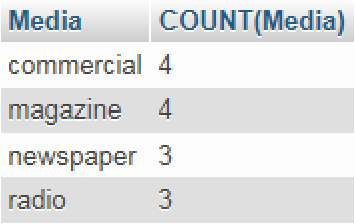


Reservation_id	Customer_fname	Customer_mname	Customer_lname
4295	Nash	S	Burns
5074	Noel	Q	Dillon
6076	Xandra	M	Beasley
6647	Iliana	S	Jefferson
6983	Piper	T	Bolton
7874	Ifeoma	C	Maddox
8986	Ifeoma	C	Maddox

Figure 15. Query 5

The query in figure 16 calculates the total number of customers who have seen a particular type of ad. The customer table is joined to the advertisement table and grouped by the Media type to find the amount of customers who have viewed each type of ad. This join is necessary because the advertisement table does not contain any customer information. The different media categories and the count associated to each category are selected to be displayed.

```
SELECT DISTINCT(Media),
COUNT(Media)
FROM customer c JOIN advertisement a ON
c.Ad_id = a.Ad_id
GROUP BY Media;
```

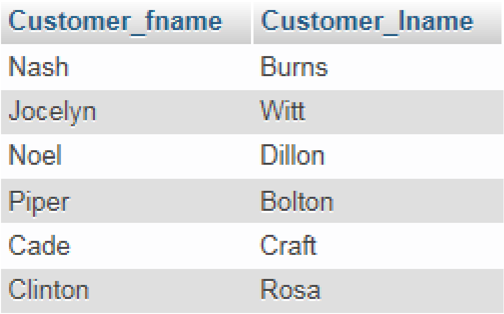


Media	COUNT(Media)
commercial	4
magazine	4
newspaper	3
radio	3

Figure 16. Query 6

The query shown in figure 17 selects the names of all of the customers who have placed a reservation and paid the amount due before July 8, 2021. The customer table is joined with the reservation table to find the names of the customers who have placed reservations. The where clause removes the rows that contain a Start_date greater than July 8, 2021. Only the customer first and last names are selected from the table to be displayed.

```
SELECT Customer_fname, Customer_lname
FROM customer c JOIN reservation r ON
c.Customer_id = r.Customer_id
WHERE Start_date < '2021-07-08';
```

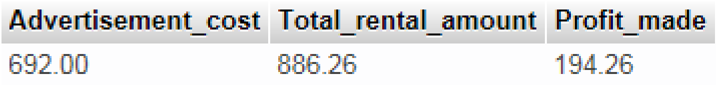


Customer_fname	Customer_lname
Nash	Burns
Jocelyn	Witt
Noel	Dillon
Piper	Bolton
Cade	Craft
Clinton	Rosa

Figure 17. Query 7

In figure 18, a new table has been created to display advertisement costs, profits made from rentals, and the net profit. The values are then calculated and inserted into each of these fields.

```
CREATE Table Profit(Advertisement_cost
decimal(8,2), Total_rental_amount
decimal(8,2), Profit_made decimal(8,2));
INSERT INTO Profit values(0, 0, 0);
UPDATE Profit set Advertisement_cost =
(Select SUM(Ad_cost) FROM
advertisement);
UPDATE Profit set Total_rental_amount =
(Select SUM(Paid_amount) FROM
reservation);
UPDATE Profit set Profit_made =
Total_rental_amount-Advertisement_cost;
SELECT * FROM Profit;
```



Advertisement_cost	Total_rental_amount	Profit_made
692.00	886.26	194.26

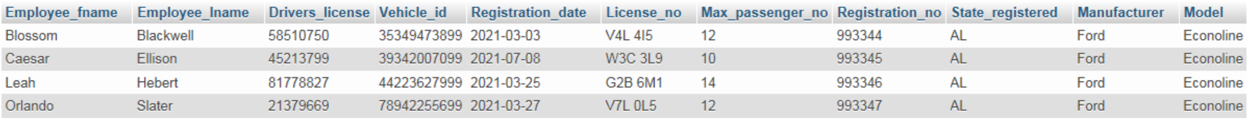
Figure 18. Query 8

Table 3 shows the result of the following query:

```
SELECT  Employee_fname, Employee_lname, e.License_no AS Drivers_license,
        Vehicle_id, Registration_date, v.License_no, Max_passenger_no,
        Registration_no, State_registered, Manufacturer, Model
FROM van v JOIN employee e ON v.Employee_id = e.Employee_id
```

This query selects and displays the names and driver's license numbers of the employees that drive vans. The information for each van is also displayed. The van table and the employee table are joined together to connect the employee name to the van that the employee drives.

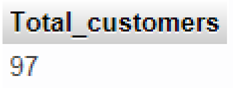
Table 3. Query 9



Employee_fname	Employee_lname	Drivers_license	Vehicle_id	Registration_date	License_no	Max_passenger_no	Registration_no	State_registered	Manufacturer	Model
Blossom	Blackwell	58510750	35349473899	2021-03-03	V4L 4I5	12	993344	AL	Ford	Econoline
Caesar	Ellison	45213799	39342007099	2021-07-08	W3C 3L9	10	993345	AL	Ford	Econoline
Leah	Hebert	81778827	44223627999	2021-03-25	G2B 6M1	14	993346	AL	Ford	Econoline
Orlando	Slater	21379669	78942255699	2021-03-27	V7L 0L5	12	993347	AL	Ford	Econoline

Figure 19 shows a query that returns the total number of customers that have made a reservation. The attribute, Number_in_party from the reservation table, is a numeric field. When the query is ran, the sum of the Number_in_party field is calculated and displayed.

```
SELECT SUM(Number_in_party)
AS Total_customers
FROM reservation;
```

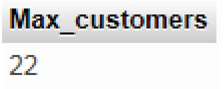


Total_customers
97

Figure 19. Query 10

Figure 20 shows the largest number of customers that made a reservation on the same day. The data in the reservation table is grouped by Start_date. Next the sum of customers for each distinct Start_date is calculated. The maximum number of customers in one day is then selected and displayed.

```
SELECT MAX(Total_party) AS Max_customers
FROM reservation r INNER JOIN
(SELECT DISTINCT(Start_date), SUM(Number_in_party) AS
Total_party
FROM reservation
GROUP BY Start_date) mc ON mc.Start_date = r.Start_date;
```



Max_customers
22

Figure 20. Query 11

X. Database Security

This section addresses the possible database security problems that may arise and the possible solutions to these problems. A database stores sensitive data about an organization and needs to be protected against any attempts by an attacker to gain unauthorized access.

Three major security issues are loss of integrity, loss of availability, and loss of confidentiality. Loss of integrity is when unauthorized modification of the data in the database occurs. When this ensues, the data is no longer accurate and cannot be used without causing further damage to the company. Loss of availability occurs when an authorized user cannot access the database. This could slow company operations greatly if they cannot access the database to modify information, like equipment availability. Loss of confidentiality occurs when database confidentiality is compromised through unauthorized disclosure of information.

Access control, inference control, flow control, and data encryption are some of the ways to provide database security. Access controls manage and restrict access to the database. This is done through the use of user accounts and passwords. User activity is also recorded in the system log. Discretionary access controls are based on granting and revoking privileges. When a user needs to perform a task that they do not have clearance for, they are granted access. Once they have completed the task, the privilege is revoked. Mandatory access controls are based on security classes. Different levels of security are imposed to restrict access to data. Role-based access controls are based on user roles. Privileges are granted to a user based on their role in the company. Inference controls prevent users from accessing all of the data stored in a database. These controls are usually used to protect personal data. Flow control restricts the flow of data to prevent it from reaching unauthorized users. Data contained in one object should not be accessible through another object. Data encryption is used to protect data being transmitted. Data is encoded and only authorized users are given the decoding algorithms.

Another security concern is SQL injection. An attacker can manipulate SQL statements to get access to the database. SQL statements can also be manipulated to make changes to the data stored in the database. One way to protect against this type of attack is to use bind variables, so that user input is not inserted directly into the SQL statement. Database functions can also be restricted to prevent attacks. Attackers also use escape characters to inject manipulation attacks, so to prevent attacks, escape characters can be removed from strings in the SQL statements.

XI. Conclusion

I think this project was a good experience. I was able to take a scenario and turn it into a functioning database. I learned how to design and create a database, and I also learned how to create a web form with a php connection program that allows a user to enter information into the database. I would prefer a career that relies on database knowledge because I enjoy the structuring and organizing of data. I like front-end, client-side web development which requires a good backend database design. I think I like the database development part better than the database design part, so I would probably like the web developer role better than the business analyst role for that reason.

XII. Appendix

This section includes the SQL code to create the database. The SQL code consists of the statements to create the tables, the insert statements for each table, and the primary and foreign key constraints.

```
create database dubedb;
use dubedb;
```

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

CREATE TABLE `advertisement` (
  `Ad_id` int(8) NOT NULL,
  `Media` char(15) DEFAULT NULL,
  `Ad_start_date` date DEFAULT NULL,
  `Ad_end_date` date DEFAULT NULL,
  `Ad_content` varchar(100) DEFAULT NULL,
  `Location_printed` char(30) DEFAULT NULL,
  `Ad_cost` decimal(8,2) DEFAULT NULL,
  `Frequency` char(15) DEFAULT NULL,
  `Promotion_code` varchar(15) DEFAULT NULL,
  `Employee_id` int(8) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO `advertisement` (`Ad_id`, `Media`, `Ad_start_date`, `Ad_end_date`, `Ad_content`,
`Location_printed`, `Ad_cost`, `Frequency`, `Promotion_code`, `Employee_id`) VALUES
(391, 'newspaper', '2021-07-13', '2022-07-13', 'This is ad 5 content.', 'Huntsville, Alabama', '54.00', 'Weekly',
'Ma4sO9', 8818),
(419, 'magazine', '2020-09-15', '2021-09-15', 'This is ad 10 content.', 'Huntsville, Alabama', '31.00', 'Monthly',
'TR56de1', 6145),
(496, 'commercial', '2021-03-10', '2022-02-10', 'This is ad 4 content.', NULL, '103.00', 'Hourly', '409GLs3', 2265),
(541, 'commercial', '2021-02-18', '2022-08-11', 'This is ad 2 content.', NULL, '103.00', 'Hourly', '57jdGH3', 8818),
(551, 'radio', '2020-08-11', '2020-12-11', 'This is ad 1 content.', NULL, '91.00', 'Daily', 'Ab34kd', 2265),
(591, 'commercial', '2021-04-07', '2021-09-01', 'This is ad 7 content.', NULL, '103.00', 'Hourly', 'ti2P56', 2265),
(623, 'magazine', '2020-12-16', '2021-08-01', 'This is ad 6 content.', 'Huntsville, Alabama', '31.00', 'Monthly',
'jdks789', 1325),
(629, 'radio', '2020-04-10', '2021-04-10', 'This is ad 3 content.', 'NULL', '91.00', 'Daily', 'R579Ghls', 1325),
(780, 'magazine', '2020-11-03', '2021-08-15', 'This is ad 9 content.', 'Huntsville, Alabama', '31.00', 'Monthly',
'hfhs41T', 1325),
(796, 'newspaper', '2021-04-30', '2021-11-30', 'This is ad 8 content.', 'Huntsville, Alabama', '54.00', 'Weekly',
'AA3219b', 8818);

CREATE TABLE `customer` (
  `Customer_id` int(8) NOT NULL,
  `Customer_fname` char(15) DEFAULT NULL,
  `Customer_mname` char(1) DEFAULT NULL,
  `Customer_lname` char(15) DEFAULT NULL,
  `Customer_gender` char(1) DEFAULT NULL,
  `Customer_street` varchar(30) DEFAULT NULL,
  `Customer_city` char(15) DEFAULT NULL,
  `Customer_state` char(2) DEFAULT NULL,
  `Customer_zip` int(5) DEFAULT NULL,
  `Customer_phone` varchar(20) DEFAULT NULL,
  `Customer_email` varchar(30) DEFAULT NULL,
  `Ad_id` int(8) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
INSERT INTO `customer` (`Customer_id`, `Customer_fname`, `Customer_mname`, `Customer_lname`,
`Customer_gender`, `Customer_street`, `Customer_city`, `Customer_state`, `Customer_zip`, `Customer_phone`,
`Customer_email`, `Ad_id`) VALUES
(2, 'Xandra', 'M', 'Beasley', 'F', '6301 Quis St.', 'Oklahoma City', 'OK', 73139, '(505) 789-1832',
'XandraBeasely@gmail.com', 541),
(32, 'Courtney', 'T', 'Fitzgerald', 'F', '4604 Sed St.', 'Wichita Falls', 'TX', 76308, '(902) 625-4371',
'CourtneyFitzgerald@gmail.com', 496),
(51, 'Camille', 'A', 'Sosa', 'F', 'Ap #279-9361 Maecenas Rd.', 'Humansville', 'MO', 65674, '(345) 788-1256',
'CamilleSosa@gmail.com', 551),
(114, 'Clinton', 'L', 'Rosa', 'M', '5359 Diam Road', 'Woodworth', 'ND', 58469, '(455) 711-1886',
'ClintonRosa@gmail.com', 780),
(129, 'Cade', 'Z', 'Craft', 'M', '1117 Penatibus Road', 'Biloxi', 'MS', 39540, '(565) 708-1246', 'CadeCraft@gmail.com',
780),
(131, 'Piper', 'T', 'Bolton', 'F', '1329 Nullam St.', 'Mobile', 'AL', 36603, '(365) 799-1756', 'PiperBolton@gmail.com',
796),
(156, 'Iliana', 'S', 'Jefferson', 'F', '3720 Sedgeway Avenue', 'Ruidoso', 'NM', 88345, '(305) 702-8893',
'IlianaJefferson@gmail.com', 391),
(161, 'Jocelyn', 'N', 'Witt', 'F', '7063 Dictum Ave', 'Adams Mill', 'TN', 37641, '(255) 889-1556',
'JocelynWitt@gmail.com', 419),
(210, 'Noel', 'Q', 'Dillon', 'M', '992 Euismod St.', 'Cascadia', 'OR', 97345, '(800) 456-9956', 'NoelDillon@yahoo.com',
391),
(224, 'Nash', 'S', 'Burns', 'M', '1873 Facilisis Avenue', 'Riverside', 'CA', 91718, '(355) 123-8854',
'Nashburns@gmail.com', 591),
(235, 'Ifeoma', 'C', 'Maddox', 'F', '241 Vel Avenue', 'Grand Forks', 'ND', 58208, '(209) 657-3294',
'IfeomaMaddox@yahoo.com', 551),
(238, 'Doris', 'P', 'Mendez', 'F', '9245 Pellentesque St.', 'South Bend', 'IN', 46615, '(900) 568-3344',
'DorisMendez@gmail.com', 623),
(378, 'Valentine', 'O', 'Ashley', 'F', 'P.O. Box 366, 7146 Mauris Rd.', 'Johnson City', 'KS', 67855, '(405) 299-4566',
'AshleyValentine@gmail.com', 541),
(493, 'Melissa', 'R', 'Padilla', 'F', '6453 Mi St.', 'Ruston', 'LA', 71270, '(477) 492-7888', 'MellissaPadilla@gmail.com',
629);

CREATE TABLE `employee` (
  `Employee_id` int(8) NOT NULL,
  `Employee_fname` char(15) DEFAULT NULL,
  `Employee_mname` char(15) DEFAULT NULL,
  `Employee_lname` char(15) DEFAULT NULL,
  `Employee_gender` char(1) DEFAULT NULL,
  `Employee_street` varchar(30) DEFAULT NULL,
  `Employee_city` char(15) DEFAULT NULL,
  `Employee_state` char(2) DEFAULT NULL,
  `Employee_zip` int(5) DEFAULT NULL,
  `Employee_cell_phone` varchar(13) DEFAULT NULL,
  `Employee_home_phone` varchar(13) DEFAULT NULL,
  `Employee_email` varchar(30) DEFAULT NULL,
  `Employee_pay` decimal(7,2) DEFAULT NULL,
  `Employee_dob` date DEFAULT NULL,
  `Date_hired` date DEFAULT NULL,
  `License_no` int(8) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
INSERT INTO `employee` (`Employee_id`, `Employee_fname`, `Employee_mname`, `Employee_lname`,
`Employee_gender`, `Employee_street`, `Employee_city`, `Employee_state`, `Employee_zip`,
`Employee_cell_phone`, `Employee_home_phone`, `Employee_email`, `Employee_pay`, `Employee_dob`,
`Date_hired`, `License_no`) VALUES
(1325, 'Leah', 'R', 'Hebert', 'F', '3134 Diam Road', 'Huntsville', 'AL', 35702, '(144) 154-052', '(747) 245-992', '1997-
10-16', '0.00', '0000-00-00', '2077-03-19', 81778827),
(2265, 'Blossom', 'S', 'Blackwell', 'F', '1290 Nullam Road', 'Huntsville', 'AL', 35702, '(477) 713-919', '(535) 479-207',
'Blackwell@gmail.com', '20000.00', '1996-12-01', '2076-08-29', 58510750),
(6145, 'Orlando', 'P', 'Slater', 'M', '2456 Quis Road', 'Huntsville', 'AL', 35702, '(306) 227-915', '(556) 301-096', '2008-
12-04', '0.00', '0000-00-00', '2091-09-11', 21379669),
(6644, 'Savannah', 'B', 'Cameron', 'F', '1552 Nullam Road', 'Huntsville', 'AL', 35702, '(211) 890-919', '(653) 567-801',
'2000-07-03', '0.00', '0000-00-00', '2080-09-21', 63384287),
(8818, 'Caesar', 'L', 'Ellison', 'M', '5366 St. James Road', 'Huntsville', 'AL', 35702, '(905) 729-794', '(711) 437-895',
'Ellison@gmail.com', '20000.00', '2015-01-15', '2095-02-22', 45213799);
```

```
CREATE TABLE `equipment` (
  `Equipment_id` varchar(20) NOT NULL,
  `Equipment_type` char(15) DEFAULT NULL,
  `Equipment_in` char(3) DEFAULT NULL,
  `Equipment_condition` char(10) DEFAULT NULL,
  `In_service_date` date DEFAULT NULL,
  `Out_service_date` date DEFAULT NULL,
  `Purchase_price` decimal(7,0) DEFAULT NULL,
  `Depreciation_by_day` decimal(8,5) DEFAULT NULL,
  `Purchase_date` date DEFAULT NULL,
  `Rental_price` decimal(6,0) DEFAULT NULL,
  `Reservation_id` int(8) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
INSERT INTO `equipment` (`Equipment_id`, `Equipment_type`, `Equipment_in`, `Equipment_condition`,
`In_service_date`, `Out_service_date`, `Purchase_price`, `Depreciation_by_day`, `Purchase_date`, `Rental_price`,
`Reservation_id`) VALUES
('009492968-00004', 'life jacket', 'in', 'good', '2020-01-05', NULL, '20', NULL, '2018-03-08', '5', 8986),
('020923850-00000', 'paddle board', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2020-03-01', '15', 3645),
('044537736-00009', 'paddle board', 'out', 'good', '2020-01-05', NULL, '100', '0.00050', '2018-03-08', '15', 7874),
('063091896-00009', 'paddle board', 'out', 'good', '2020-01-05', '2019-12-18', '100', '0.00050', '2018-03-08', '15',
4073),
('095577532-00005', 'life jacket', 'out', 'good', '2020-01-05', NULL, '20', NULL, '2018-03-01', '5', 3769),
('118853126-00001', 'paddle board', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2018-03-08', '15', 5074),
('123315392-00001', 'tandem kayak', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2018-03-08', '15', 4427),
('142873926-00000', 'life jacket', 'in', 'good', '2020-02-05', NULL, '20', NULL, '2020-01-08', '5', 4073),
('149055857-00002', 'life jacket', 'in', 'good', '2020-01-05', NULL, '20', NULL, '2018-03-08', '5', 8986),
('154318455-00008', 'tandem kayak', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2018-03-08', '15', 5074),
('161670005-00008', 'inner tube', 'in', 'good', '2020-01-05', NULL, '50', '0.00050', '2018-03-08', '10', 6647),
('163106206-00002', 'inner tube', 'in', 'good', '2020-01-05', NULL, '50', '0.00050', '2018-03-08', '10', 8176),
('166877688-00001', 'single kayak', 'out', 'poor', '2020-01-05', NULL, '200', '0.00050', '2020-03-01', '11', 3645),
('174542035-00004', 'single kayak', 'out', 'good', '2020-01-05', NULL, '170', '0.00050', '2018-03-08', '15', 8176),
('180918047-00002', 'tandem kayak', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2020-03-20', '15', 4295),
('200493070-00003', 'inner tube', 'in', 'good', '2020-01-05', NULL, '55', '0.00050', '2019-07-08', '10', 6206),
('220878771-00009', 'single kayak', 'in', 'poor', '2020-01-05', NULL, '200', '0.00050', '2018-03-08', '15', 4073),
('221805096-00007', 'canoe', 'out', 'poor', '2021-06-09', NULL, '150', '0.00050', '2019-03-08', '15', 4295),
('228239356-00006', 'tandem kayak', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2020-06-08', '15', 9975),
('232304410-00003', 'tandem kayak', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2018-03-08', '15', 6983),
('233797091-00003', 'single kayak', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2018-03-08', '15', 7874),
('247500622-00003', 'single kayak', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2018-03-08', '15', 8986),
('259923670-00006', 'inner tube', 'in', 'good', '2020-01-05', NULL, '50', '0.00050', '2019-07-08', '10', 6647),
('265400234-00000', 'single kayak', 'in', 'good', '2021-01-05', '2020-12-12', '200', '0.00050', '2019-07-08', '15', 6206),
('292238029-00006', 'canoe', 'in', 'good', '2020-07-05', '2020-06-04', '200', '0.00050', '2020-03-01', '15', 4073),
('295358048-00003', 'canoe', 'in', 'good', '2020-02-05', NULL, '100', '0.00050', '2020-01-08', '15', 4073),
('298765207-00006', 'inner tube', 'in', 'good', '2020-01-05', NULL, '50', NULL, '2018-03-08', '15', 8176),
('325663060-00003', 'tandem kayak', 'out', 'poor', '2020-01-05', NULL, '200', '0.00050', '2020-03-01', '12', 3645),
('352350813-00007', 'inner tube', 'in', 'poor', '2020-01-05', NULL, '50', '0.00050', '2018-03-08', '8', 4295),
('353139389-00004', 'canoe', 'in', 'poor', '2020-01-05', NULL, '200', '0.00050', '2018-03-08', '12', 6206),
('358648228-00007', 'paddle board', 'in', 'good', '2020-02-05', NULL, '200', '0.00050', '2020-02-01', '15', 4073),
('365845080-00008', 'inner tube', 'in', 'good', '2020-01-05', NULL, '50', '0.00050', '2020-07-22', '10', 4295),
('393527908-00007', 'canoe', 'in', 'poor', '2020-01-05', NULL, '200', '0.00050', '2018-03-08', '12', 6206),
('412470254-00001', 'life jacket', 'out', 'good', '2020-01-05', NULL, '20', NULL, '2018-03-08', '5', 7874),
('436210959-00009', 'paddle board', 'in', 'good', '2020-01-05', NULL, '100', '0.00050', '2018-03-08', '15', 4427),
('438864639-00002', 'inner tube', 'in', 'poor', '2020-02-05', NULL, '200', '0.00050', '2020-01-01', '10', 3769),
('440353100-00009', 'tandem kayak', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2019-03-08', '15', 6647),
('442113890-00002', 'life jacket', 'in', 'good', '2018-03-05', NULL, '20', NULL, '2020-03-01', '5', 3645),
('458732336-00003', 'single kayak', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2018-03-08', '15', 8986),
('458982741-00001', 'life jacket', 'in', 'good', '2020-01-09', '2019-11-19', '20', NULL, '2019-07-08', '5', 6206),
('460910433-00008', 'life jacket', 'in', 'good', '2020-01-05', NULL, '20', NULL, '2018-03-08', '5', 5074),
('508306008-00006', 'paddle board', 'in', 'good', '2021-01-05', '2020-11-17', '200', '0.00050', '2019-07-08', '15', 6206),
('524867207-00002', 'tandem kayak', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2018-03-08', '15', 5074),
('527482582-00006', 'life jacket', 'in', 'good', '2020-01-05', '2019-12-02', '20', NULL, '2018-03-08', '5', 6076),
('547942573-00007', 'canoe', 'in', 'good', '2020-01-05', NULL, '150', '0.00050', '2018-03-08', '15', 8986),
('548328855-00000', 'tandem kayak', 'out', 'good', '2020-01-05', NULL, '200', '0.00050', '2018-03-08', '15', 8176),
('560600553-00005', 'tandem kayak', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2018-03-08', '15', 4073),
('610052482-00003', 'paddle board', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2019-07-08', '15', 6647),
('619923014-00002', 'paddle board', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2019-04-05', '15', 6076),
('635228653-00005', 'paddle board', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2018-03-08', '15', 8176),
('647452796-00004', 'life jacket', 'in', 'good', '2020-02-05', NULL, '20', NULL, '2020-01-08', '5', 3769),
('652322082-00007', 'life jacket', 'in', 'good', '2020-01-05', NULL, '20', NULL, '2018-03-08', '5', 8176),
('662156207-00000', 'life jacket', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2018-03-08', '5', 6983),
('667890230-00002', 'paddle board', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2018-03-08', '15', 4427),
('679824334-00008', 'tandem kayak', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2018-03-19', '15', 4427),
('712059856-00005', 'life jacket', 'in', 'good', '2020-02-05', NULL, '20', NULL, '2020-01-01', '5', 3769),
('739639565-00004', 'canoe', 'in', 'good', '2020-01-05', NULL, '500', '0.00050', '2018-03-08', '15', 3769),
('746780113-00005', 'tandem kayak', 'in', 'good', '2020-01-05', '2019-11-17', '200', '0.00050', '2019-07-08', '15',
6206),
('753178037-00002', 'canoe', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2019-03-08', '15', 4427),
('754296010-00004', 'paddle board', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2018-03-08', '15', 7874),
('792341521-00005', 'tandem kayak', 'in', 'good', '2020-02-05', NULL, '200', '0.00050', '2020-01-01', '15', 3769),
('794778050-00003', 'life jacket', 'in', 'poor', '2020-01-05', NULL, '20', NULL, '2018-03-08', '3', 4427),
('842467052-00002', 'tandem kayak', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2018-03-08', '15', 6983),
('852623461-00005', 'canoe', 'in', 'poor', '2020-01-05', NULL, '100', '0.00050', '2018-03-08', '15', 4073),
('856767892-00008', 'single kayak', 'out', 'good', '2020-01-05', NULL, '175', '0.00050', '2018-03-08', '15', 8176),
('867600405-00007', 'paddle board', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2020-06-08', '15', 9975),
('877805671-00006', 'paddle board', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2020-06-08', '15', 9975),
('878437599-00003', 'tandem kayak', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2020-06-08', '15', 9975),
('879310522-00005', 'paddle board', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2018-03-08', '15', 5074),
('909077695-00007', 'life jacket', 'out', 'good', '2018-03-05', NULL, '20', NULL, '2020-03-01', '5', 3645),
('909481558-00007', 'single kayak', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2018-03-08', '15', 4295),
('935738591-00004', 'paddle board', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2020-06-08', '15', 9975),
('957771983-00005', 'single kayak', 'in', 'poor', '2020-01-05', NULL, '200', '0.00050', '2018-03-08', '15', 4295),
('976226217-00009', 'canoe', 'in', 'good', '2020-01-05', NULL, '200', '0.00050', '2018-03-08', '15', 7874);

CREATE TABLE `insurance` (
  `Insurance_id` int(8) NOT NULL,
  `Date_purchased` date DEFAULT NULL,
  `Days_covered` int(2) DEFAULT NULL,
  `Total_paid` decimal(7,2) DEFAULT NULL,
  `Reservation_id` int(8) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO `insurance` (`Insurance_id`, `Date_purchased`, `Days_covered`, `Total_paid`, `Reservation_id`)
VALUES
(25172, '2021-02-27', 2, '12.00', 7874),
(96843, '2021-03-17', 1, '6.00', 5074),
(315412, '2021-03-17', 1, '6.00', 4295),
(564887, '2021-07-25', 3, '18.00', 8986),
(654611, '2021-02-27', 1, '6.00', 6983),
(888113, '2021-12-20', 2, '12.00', 6076),
(948778, '2021-05-28', 1, '6.00', 6647);

CREATE TABLE `profit` (
  `Advertisement_cost` decimal(8,2) DEFAULT NULL,
  `Total_rental_amount` decimal(8,2) DEFAULT NULL,
  `Profit_made` decimal(8,2) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;


INSERT INTO `profit` (`Advertisement_cost`, `Total_rental_amount`, `Profit_made`) VALUES
('692.00', '886.26', '194.26');

CREATE TABLE `reservation` (
  `Reservation_id` int(8) NOT NULL,
  `Start_date` date DEFAULT NULL,
  `Stop_date` date DEFAULT NULL,
  `Number_in_party` int(3) DEFAULT NULL,
  `Res_equip_quantity` int(3) DEFAULT NULL,
  `Estimate` decimal(8,2) DEFAULT NULL,
  `Paid_amount` decimal(8,2) DEFAULT NULL,
  `Customer_id` int(8) NOT NULL,
  `Ad_id` int(8) NOT NULL,
  `Employee_id` int(8) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO `reservation` (`Reservation_id`, `Start_date`, `Stop_date`, `Number_in_party`,
`Res_equip_quantity`, `Estimate`, `Paid_amount`, `Customer_id`, `Ad_id`, `Employee_id`) VALUES
(3645, '2022-03-06', '2022-03-06', 4, 6, '60.00', '67.87', 378, 551, 8818),
(3769, '2022-05-08', '2022-05-12', 5, 6, '60.00', '66.83', 32, 391, 6145),
(4073, '2022-05-26', '2022-05-28', 11, 5, '50.00', '61.62', 51, 551, 2265),
(4295, '2021-03-30', '2021-03-31', 12, 3, '30.00', '47.18', 224, 796, 6644),
(4427, '2021-03-30', '2021-04-01', 10, 5, '50.00', '63.88', 161, 629, 6644),
(5074, '2021-03-27', '2021-03-29', 6, 7, '70.00', '79.89', 210, 796, 1325),
(6076, '2021-12-25', '2021-12-27', 1, 2, '20.00', '23.47', 2, 391, 8818),
(6206, '2022-05-27', '2022-05-29', 9, 8, '80.00', '92.31', 493, 551, 1325),
(6647, '2022-05-31', '2022-06-02', 5, 6, '60.00', '71.57', 156, 391, 6644),
(6983, '2021-03-03', '2021-03-04', 4, 5, '50.00', '56.61', 131, 796, 1325),
(7874, '2022-03-03', '2022-03-06', 6, 5, '50.00', '57.88', 235, 391, 2265),
(8176, '2020-07-27', '2020-07-28', 10, 5, '50.00', '66.83', 129, 629, 2265),
(8986, '2021-07-08', '2021-07-08', 11, 4, '40.00', '53.29', 235, 796, 6145),
(9975, '2021-03-16', '2021-03-19', 3, 7, '70.00', '77.03', 114, 629, 6145);

CREATE TABLE `survey` (
  `Survey_id` int(8) NOT NULL,
  `Cust_rating` int(1) DEFAULT NULL,
  `Emp_rating` int(1) DEFAULT NULL,
  `Equip_rating` int(1) DEFAULT NULL,
  `Customer_id` int(8) NOT NULL,
  `Reservation_id` int(8) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO `survey` (`Survey_id`, `Cust_rating`, `Emp_rating`, `Equip_rating`, `Customer_id`,
`Reservation_id`) VALUES
(377, 1, 7, 8, 378, 3645),
(383, 3, 5, 4, 235, 7874),
(433, 4, 2, 1, 493, 6206),
(634, 2, 10, 7, 51, 4073),
(647, 6, 2, 9, 129, 8176),
(656, 9, 1, 6, 224, 4295),
(668, 10, 10, 2, 235, 8986),
(795, 5, 2, 9, 210, 5074),
(882, 7, 6, 4, 2, 6076),
(934, 8, 8, 2, 161, 4427);

CREATE TABLE `van` (
  `Vehicle_id` varchar(20) NOT NULL,
  `Registration_date` date DEFAULT NULL,
  `License_no` varchar(10) DEFAULT NULL,
  `Max_passenger_no` int(2) DEFAULT NULL,
  `Registration_no` int(15) DEFAULT NULL,
  `State_registered` char(2) DEFAULT NULL,
  `Manufacturer` varchar(20) DEFAULT NULL,
  `Model` varchar(20) DEFAULT NULL,
  `Reservation_id` int(8) DEFAULT NULL,
  `Employee_id` int(8) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO `van` (`Vehicle_id`, `Registration_date`, `License_no`, `Max_passenger_no`, `Registration_no`,
`State_registered`, `Manufacturer`, `Model`, `Reservation_id`, `Employee_id`) VALUES
('35349473899', '2021-03-03', 'V4L 4I5', 12, 993344, 'AL', 'Ford', 'Econoline', 4073, 2265),
('39342007099', '2021-07-08', 'W3C 3L9', 10, 993345, 'AL', 'Ford', 'Econoline', 3645, 8818),
('44223627999', '2021-03-25', 'G2B 6M1', 14, 993346, 'AL', 'Ford', 'Econoline', 6206, 1325),
('78942255699', '2021-03-27', 'V7L 0L5', 12, 993347, 'AL', 'Ford', 'Econoline', 3769, 6145);
```

```
CREATE TABLE `waiver` (
  `Waiver_id` varchar(15) NOT NULL,
  `Waiver_date_signed` date DEFAULT NULL,
  `Safety_waiver` char(3) DEFAULT NULL,
  `Customer_id` int(8) NOT NULL,
  `Reservation_id` int(8) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO `waiver` (`Waiver_id`, `Waiver_date_signed`, `Safety_waiver`, `Customer_id`, `Reservation_id`)
VALUES
('10287489-7', NULL, 'no', 235, 8986),
('12444936-7', '2022-05-26', 'yes', 51, 4073),
('14812709-3', NULL, 'no', 210, 5074),
('15180766-6', '2020-07-27', 'yes', 129, 8176),
('15702534-1', NULL, 'no', 2, 6076),
('20791366-9', '2022-03-03', 'yes', 235, 7874),
('32838369-1', '2021-03-16', 'yes', 114, 9975),
('33014751-2', '2022-05-31', 'yes', 156, 6647),
('38667837-5', '2022-03-06', 'yes', 378, 3645),
('42029098-5', '2021-03-30', 'yes', 224, 4295),
('46273411-5', '2022-05-27', 'yes', 493, 6206),
('6263817-6', '2022-02-17', 'yes', 131, 6983),
('8177376-9', NULL, 'no', 32, 3769),
('8749646-5', NULL, 'no', 161, 4427);

ALTER TABLE `advertisement`
  ADD PRIMARY KEY (`Ad_id`),
  ADD KEY `Employee_id` (`Employee_id`);
ALTER TABLE `customer`
  ADD PRIMARY KEY (`Customer_id`),
  ADD KEY `Ad_id` (`Ad_id`);

ALTER TABLE `employee`
  ADD PRIMARY KEY (`Employee_id`);
ALTER TABLE `equipment`
  ADD PRIMARY KEY (`Equipment_id`),
  ADD KEY `Reservation_id` (`Reservation_id`);

ALTER TABLE `insurance`
  ADD PRIMARY KEY (`Insurance_id`),
  ADD KEY `Reservation_id` (`Reservation_id`);
ALTER TABLE `reservation`
  ADD PRIMARY KEY (`Reservation_id`),
  ADD KEY `Ad_id` (`Ad_id`),
  ADD KEY `Customer_id` (`Customer_id`),
  ADD KEY `Employee_id` (`Employee_id`);

ALTER TABLE `survey`
  ADD PRIMARY KEY (`Survey_id`),
  ADD KEY `Customer_id` (`Customer_id`),
  ADD KEY `Reservation_id` (`Reservation_id`);
```

```
ALTER TABLE `van`
  ADD PRIMARY KEY (`Vehicle_id`),
  ADD KEY `Employee_id` (`Employee_id`),
  ADD KEY `Reservation_id` (`Reservation_id`);

ALTER TABLE `waiver`
  ADD PRIMARY KEY (`Waiver_id`),
  ADD KEY `Customer_id` (`Customer_id`),
  ADD KEY `Reservation_id` (`Reservation_id`);

ALTER TABLE `advertisement`
  ADD CONSTRAINT `advertisement_ibfk_1` FOREIGN KEY (`Employee_id`) REFERENCES `employee`
(`Employee_id`);

ALTER TABLE `customer`
  ADD CONSTRAINT `customer_ibfk_1` FOREIGN KEY (`Ad_id`) REFERENCES `advertisement` (`Ad_id`);

ALTER TABLE `equipment`
  ADD CONSTRAINT `equipment_ibfk_1` FOREIGN KEY (`Reservation_id`) REFERENCES `reservation`
(`Reservation_id`);

ALTER TABLE `insurance`
  ADD CONSTRAINT `insurance_ibfk_1` FOREIGN KEY (`Reservation_id`) REFERENCES `reservation`
(`Reservation_id`);

ALTER TABLE `reservation`
  ADD CONSTRAINT `reservation_ibfk_1` FOREIGN KEY (`Ad_id`) REFERENCES `advertisement` (`Ad_id`),
  ADD CONSTRAINT `reservation_ibfk_2` FOREIGN KEY (`Customer_id`) REFERENCES `customer`
(`Customer_id`),
  ADD CONSTRAINT `reservation_ibfk_3` FOREIGN KEY (`Employee_id`) REFERENCES `employee`
(`Employee_id`);

ALTER TABLE `survey`
  ADD CONSTRAINT `survey_ibfk_1` FOREIGN KEY (`Customer_id`) REFERENCES `customer`
(`Customer_id`),
  ADD CONSTRAINT `survey_ibfk_2` FOREIGN KEY (`Reservation_id`) REFERENCES `reservation`
(`Reservation_id`);

ALTER TABLE `van`
  ADD CONSTRAINT `van_ibfk_1` FOREIGN KEY (`Employee_id`) REFERENCES `employee`
(`Employee_id`),
  ADD CONSTRAINT `van_ibfk_2` FOREIGN KEY (`Reservation_id`) REFERENCES `reservation`
(`Reservation_id`);

ALTER TABLE `waiver`
  ADD CONSTRAINT `waiver_ibfk_1` FOREIGN KEY (`Customer_id`) REFERENCES `customer`
(`Customer_id`),
  ADD CONSTRAINT `waiver_ibfk_2` FOREIGN KEY (`Reservation_id`) REFERENCES `reservation`
(`Reservation_id`);
COMMIT;
```

XIII. References

[1] “Ocoee Paddleboarding,” *Ocoee Paddleboarding*. http://www.ocoeepaddleboarding.com/home.html (accessed Jun. 07, 2021).