



Prairie View A&M University

ROY G. PERRY COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE

CAR DEALERSHIP DATABASE

COMP 3395-P01 - Database Management

Instructor: Dr. Mary Kim

Authors:

Marcus Rockwell

Francisco Ekang Mofuman

Joaquin Owono Afugu Ntoo

Francisco Ndong Nengono

November 15, 2022

Table of content

Introduction	3
1. Overview of the System	3
2. Architecture & ER diagram	4
3. Implementation Methods	5
3.1.Database Website.....	6
4. Example Queries/Executions	8
5. Example of Triggers	10
6. Discussion & Conclusion	11
7. Group Statement	11
References	12

Introduction

Having a company, such as a car dealership, that deals with too much unique data can sometimes be stressful and confusing. Especially without a database to store and manage all this data. Moreover, security is another crucial point because we must ensure access limitations to specific information. Company employees should only have access to their information and certain data according to their roles in the company.

With this problem in mind, we decided to create the Car Dealership Database. Our database will help any car dealership company to store, manage, secure and retrieve their data efficiently, as is stated in the following purpose statement.

This database was created with the purpose of helping car dealership companies keep track of the employees that are a part of the organization and the movements that are made within the organization itself. Additionally, to keep track of the activities, purchases, and sales of vehicles by monitoring the flow of all the transactions.

In a car dealership company, we have to deal with three sources of data. The first one is the company employee information, in which we have the personal data of each employee contributing to the purchase and sale of the cars. Secondly, there are providers from whom we are buying the vehicles. Finally, we have the customers, which are the people who are buying the product. We need to organize all this data in order to analyze and make a variety of conclusions. For example, figuring out which providers present the best offers, how we can give better service to our customers, and lastly, keeping track of how the company employees are performing. That will let us know which employees are producing more income for the company, which should be promoted or have a salary increase, or who should be fired, in the case of making the company spend more but without enough income.

The database will be a very helpful tool for any car dealership looking to optimize their profit, offer higher quality service to their customers, and promote workplace productivity.

1. Overview of the System

The database will help car dealership companies organize and control their data better. The database system will work in such a way that the data can be stored, managed, and retrieved efficiently. The data in the database include IDs, passwords, names, salaries, emails, card information, addresses, phone numbers, car models, VIN, prices, years, etc.

The database has five tables: transactions, provider, customers, employee, and car inventory. Each of those tables has a primary key, such as the ProvID in the provider table, the EmployeeID in the Employee table, the CustomerID in the customer table, the VIN in the carinventory table, and the TransactionID in the transaction table.

The system of the database is made so that the company can save all the transactions done, either when buying or when selling the vehicles. All of those transactions are saved in the transaction table. This table contains four foreign keys: CustomerID, ProvID, Employee, and VIN. Those foreign keys get the information from other tables to ensure that data is not lost.

2. Architecture & ER diagram

Figure 2 shows the architecture of the Car Dealership Database, and Figure 3 shows the ER diagram. Both illustrate how the database is structured, the variables that make each of the tables, and the datatypes of each of them. The figures also show how the primary and foreign keys of the tables are connected.

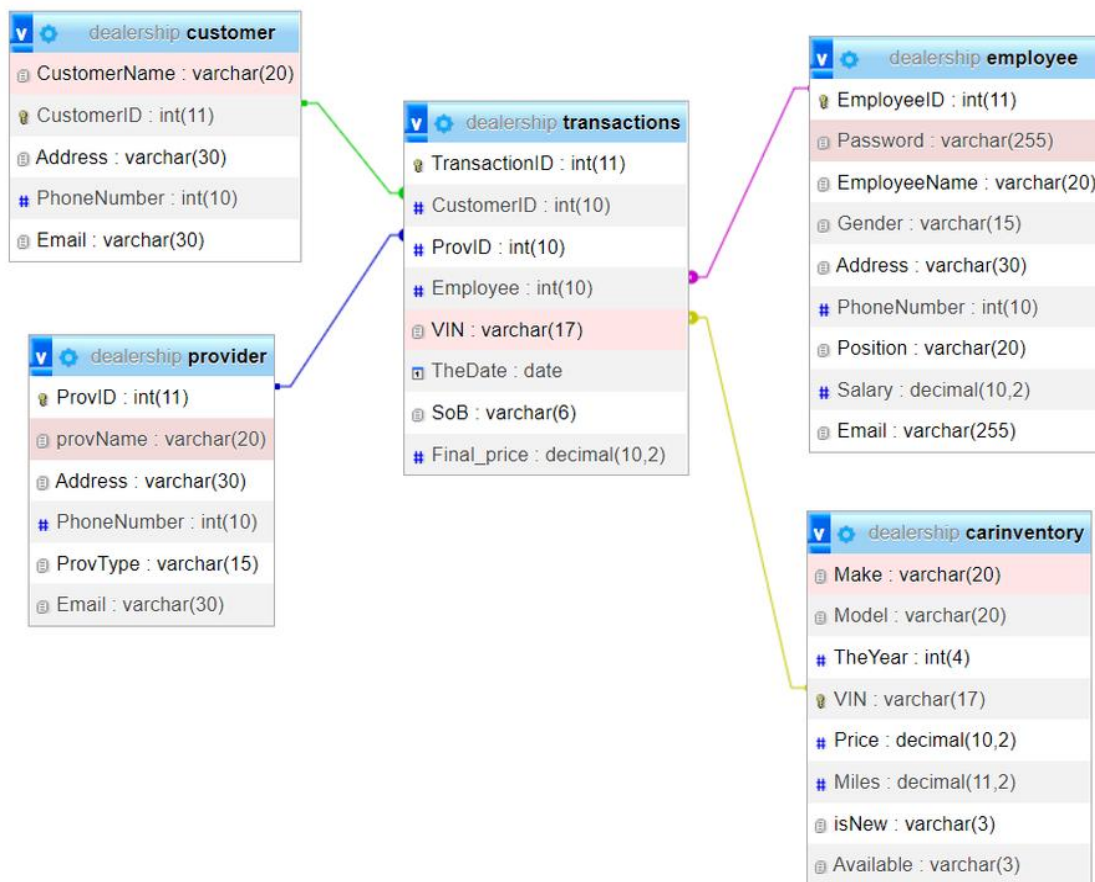


Figure 1. Layout of the Database

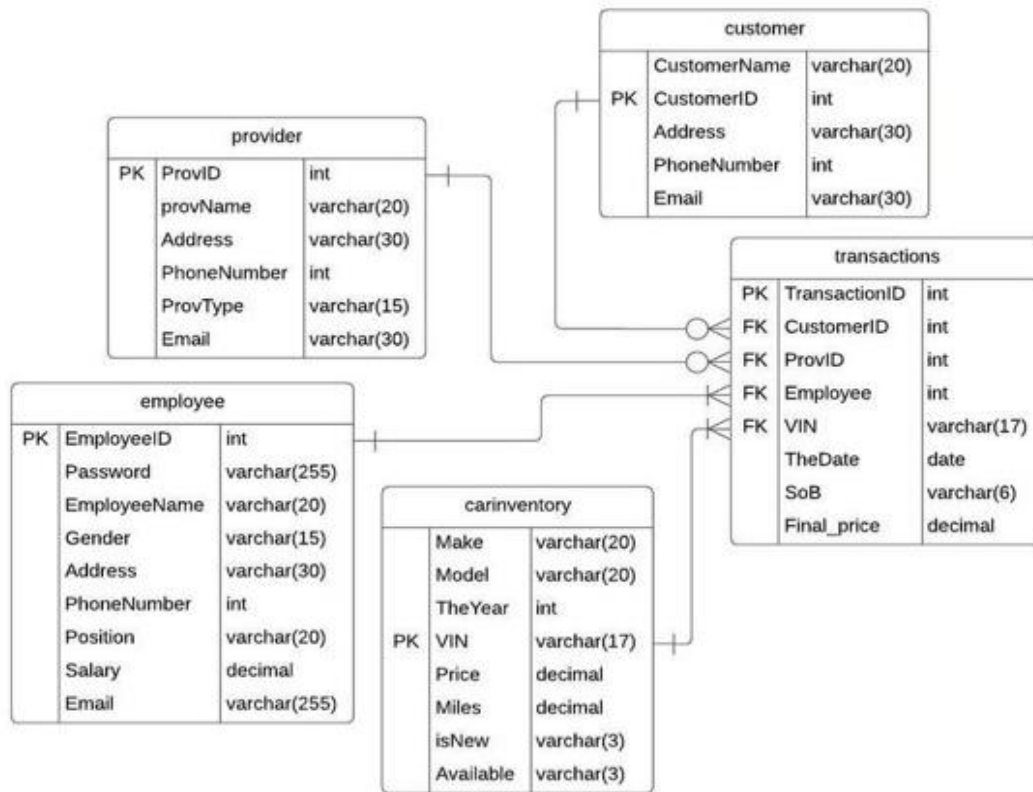


Figure 2. ER Diagram of the Database

3. Implementation Methods

The Car Dealership Database was created using multiple tools provided by XAMPP, namely MySQL and phpMyAdmin. In our prototype phase, we implemented the basic tables using the MySQL command console. Once we had the general structure decided, we used phpMyAdmin to clean up some of the data types and constraints used for the columns as well as adding in records for testing purposes, which you can see below (Figure 3). As a tool, phpMyAdmin proved to be very useful in helping us fine tune our database in a quick and responsive manner, allowing our team to do some minor restructuring as we went. We also used the interface to create a bundle of procedures, functions, and triggers that were implemented in our web interface. Finally, we used CSS and HTML to create a user interface to help employees browse and handle their data in the database.

Server: 127.0.0.1 > Database: dealership > Table: carinventory

Showing rows 0 - 7 (8 total, Query took 0.0002 seconds.)

```
SELECT * FROM `carinventory`
```

Number of rows: 25 Filter rows: Search this table Sort by key: None

	Make	Model	TheYear	VIN	Price	Miles	isNew	Available
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Toyota	Corolla	2014	12werdftyhjuio98	23457.00	11122.00	Yes	No
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Toyota	Hylux	2012	AKDLEU56SR46DT3R1	25000.00	8000.00	Yes	No
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Kia	abcd	2016	FERSHY45AT45DT4JC	8500.00	120100.00	Yes	No
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Honda	paso	2012	GHEY4YS7RTANFK8UR	12500.00	120012.00	No	No
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Toyota	Carina	2012	GIBKMSHR56ST3DTV5	12500.00	412235.00	No	No
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Audi	Q7	2014	PLSGETS53TDNGUA43	35000.00	7500.00	Yes	No
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Kia	Nolose	2015	SYTHSTR5WF3TS6THX	90000.00	120012.00	No	No
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Honda	turismo	2017	WURTCSDRAHTI87ST3	4500.00	47556.00	No	No

Number of rows: 25 Filter rows: Search this table Sort by key: None

Figure 3: A snapshot of our CarInventory table and its test records.

3.1. Database Website

For the creation of the car dealership website, we have used HTML, CSS Styles, and PHP files that contain some functions to perform in the database, which are connected to the PHPMyAdmin server, where the functions to be executed are stored.

The web page has a log in page, where users will have to enter their credentials to access the page which are the Employee ID and password. Next, it will take the user to the main page, where depending on his position in the organization, the user will have a set of actions they could perform since the level of access will vary depending on whether the user is the CEO, a Manager, Salesman, or Assistant. The web page is quite responsive and easy to navigate and the functions to be executed are distributed in icons that are easy to identify and access.

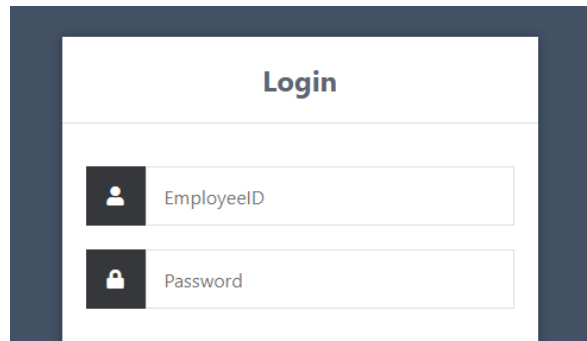
Regarding the hierarchy when executing an action by the user, the conditions are following:

CEO: Has access to all the functions of the dealership and can access them through the website.

Manager: Has access to all the functions of the dealership and can access them through the website

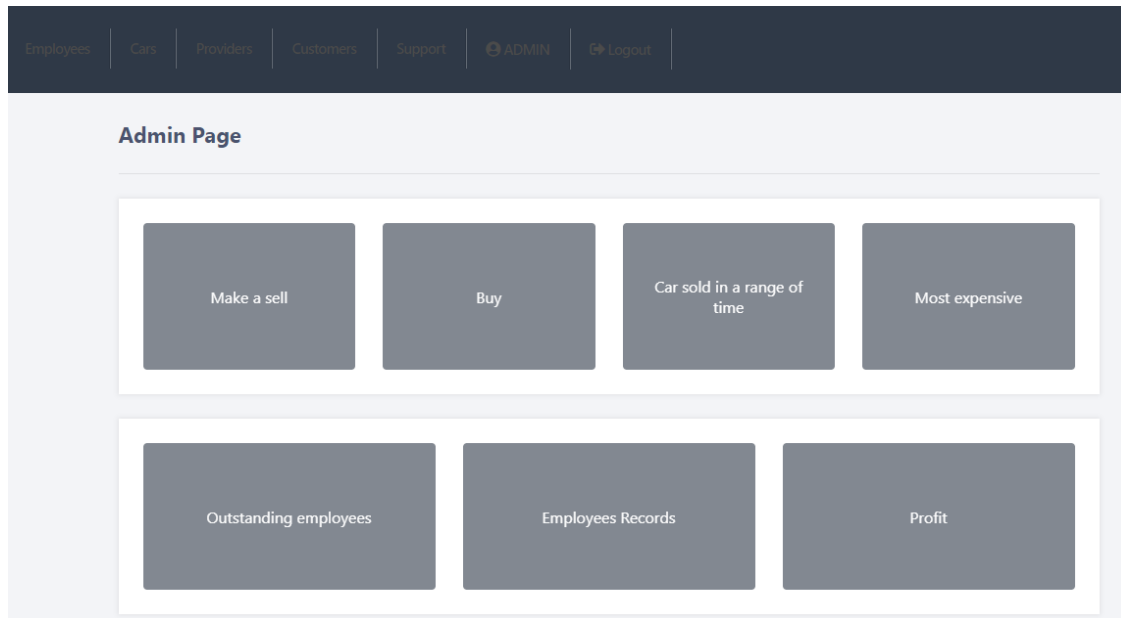
Salesman: Has access to View the list of Employees, make sales and purchases, and view the list of customers. But does not have access to add a new employee to the organization.

Assistant: Has the same level of access as the salesman on the website, except to make purchases and sales.



The login portal features a dark blue header with the word "Login" in white. Below the header, there are two input fields: the first is labeled "EmployeeID" with a person icon, and the second is labeled "Password" with a lock icon. Both fields are white with a light gray border and are set against a dark blue background.

Figure 4. Database login portal



The Admin Page has a dark blue header with navigation links: Employees, Cars, Providers, Customers, Support, ADMIN (active), and Logout. Below the header, the page is titled "Admin Page". It contains two rows of gray buttons. The first row includes "Make a sell", "Buy", "Car sold in a range of time", and "Most expensive". The second row includes "Outstanding employees", "Employees Records", and "Profit".

Figure 5. User interface

Select Month:



The form consists of a "Select Month:" label, a dropdown menu showing "January" with a downward arrow, and a gray "Submit" button.

TransactionID	CustomerID	ProvID	Employee	VIN	TheDate	SoB	Final_price
4440007	3330001		2220000	NFTERAFDRRT45TDRE	2022-11-13	Sold	48000.00
4440008	3330001		2220000	MG85YDHRT45AD3EW4	2022-11-13	Sold	20500.00
4440009	3330002		2220000	MFHTY6STE5TDF7DH	2022-11-25	Sold	11000.00
4440010	3330003		2220000	MKVGTHYD65TSE34RA	2022-11-17	Sold	80000.00

Figure 6. January Transactions

4. Example Queries/Executions

The following figures shows examples of some of the common and queries (answer to business questions) in the Car DealershipDB.

```
MariaDB [dealership]> select * from Largest_Provider;
+-----+-----+-----+
| ProvName | ProvType | COUNT(*) |
+-----+-----+-----+
| Toyota   | Manufacture | 3 |
+-----+-----+-----+
1 row in set (0.001 sec)
```

Figure 7. Display the created view of the best provider

```
MariaDB [dealership]> select * from Largest_Source;
+-----+-----+
| ProvType | COUNT(*) |
+-----+-----+
| Manufacture | 4 |
| Individual | 2 |
| Auction | 1 |
+-----+-----+
3 rows in set (0.001 sec)
```

Figure 8: Display a created view of the sources where we got the most cards

```
MariaDB [dealership]> select * from Best_Customer;
+-----+-----+
| CustomerName | COUNT(*) |
+-----+-----+
| Taylor Johnson | 2 |
| Maria Brown | 2 |
+-----+-----+
2 rows in set (0.001 sec)
```

Figure 9: Displaying the a created view of the best Customers

```
MariaDB [dealership]> select EmployeeName, EmployeeID, CustomerName, customer.CustomerID from employee, customer, transactions where transactions.employee = EmployeeID AND transactions.CustomerID = customer.CustomerID;
+-----+-----+-----+-----+
| EmployeeName | EmployeeID | CustomerName | CustomerID |
+-----+-----+-----+-----+
| ADMIN | 2220000 | Taylor Johnson | 3330001 |
| ADMIN | 2220000 | Taylor Johnson | 3330001 |
| ADMIN | 2220000 | Caleb Solem | 3330002 |
| ADMIN | 2220000 | Samantha Jeferson | 3330003 |
| ADMIN | 2220000 | Maria Brown | 3330004 |
| ADMIN | 2220000 | Maria Brown | 3330004 |
+-----+-----+-----+-----+
6 rows in set (0.001 sec)
```

Figure 10: Displaying the customer information with the employee that served him


```
MariaDB [dealership]> select * from transactions where MONTH(TheDate) = 11;
```

TransactionID	CustomerID	ProvID	Employee	VIN	TheDate	SoB	Final_price
4440001	NULL	1110001	2220000	NFTERAFDRRT45TDRE	2022-11-01	Bought	45000.00
4440002	NULL	1110001	2220000	MBYHT6ETS43FRT56	2022-11-02	Bought	21000.00
4440003	NULL	1110001	2220000	MG85YDHRT45AD3EW4	2022-11-02	Bought	18000.00
4440004	NULL	1110002	2220000	MKVGTHYD65TSE34RA	2022-11-03	Bought	12000.00
4440005	NULL	1110003	2220000	MRTYHBFT46STER34A	2022-11-11	Bought	6000.00
4440006	NULL	1110004	2220000	MFHTY6STE5TDFT7DH	2022-11-12	Bought	8500.00
4440007	3330001	NULL	2220000	NFTERAFDRRT45TDRE	2022-11-13	Sold	48000.00
4440008	3330001	NULL	2220000	MG85YDHRT45AD3EW4	2022-11-13	Sold	20500.00
4440009	3330002	NULL	2220000	MFHTY6STE5TDFT7DH	2022-11-25	Sold	11000.00
4440010	3330003	NULL	2220000	MKVGTHYD65TSE34RA	2022-11-17	Sold	80000.00
4440011	3330004	NULL	2220000	MBYHT6ETS43FRT56	2022-11-15	Sold	62000.00
4440012	NULL	1110005	2220000	GUNSTHEYTHINCR3TS	2022-11-11	Bought	62000.00
4440013	3330004	NULL	2220000	MRTYHBFT46STER34A	2022-11-16	Sold	8500.00

```
13 rows in set (0.001 sec)
```

Figure 11: Displaying all the transaction made in a the 11th month (November)

```
MariaDB [dealership]> select * from transactions where MONTH(TheDate) = 11 and SoB LIKE '%Sold%';
```

TransactionID	CustomerID	ProvID	Employee	VIN	TheDate	SoB	Final_price
4440007	3330001	NULL	2220000	NFTERAFDRRT45TDRE	2022-11-13	Sold	48000.00
4440008	3330001	NULL	2220000	MG85YDHRT45AD3EW4	2022-11-13	Sold	20500.00
4440009	3330002	NULL	2220000	MFHTY6STE5TDFT7DH	2022-11-25	Sold	11000.00
4440010	3330003	NULL	2220000	MKVGTHYD65TSE34RA	2022-11-17	Sold	80000.00
4440011	3330004	NULL	2220000	MBYHT6ETS43FRT56	2022-11-15	Sold	62000.00
4440013	3330004	NULL	2220000	MRTYHBFT46STER34A	2022-11-16	Sold	8500.00

```
6 rows in set (0.001 sec)
```

Figure 12. Display all the sales made in November

```
MariaDB [dealership]> select * from transactions where Thedate BETWEEN '2022-10-29' AND '2022-11-13' AND SoB = 'Bought';
```

TransactionID	CustomerID	ProvID	Employee	VIN	TheDate	SoB	Final_price
4440001	NULL	1110001	2220000	NFTERAFDRRT45TDRE	2022-11-01	Bought	45000.00
4440002	NULL	1110001	2220000	MBYHT6ETS43FRT56	2022-11-02	Bought	21000.00
4440003	NULL	1110001	2220000	MG85YDHRT45AD3EW4	2022-11-02	Bought	18000.00
4440004	NULL	1110002	2220000	MKVGTHYD65TSE34RA	2022-11-03	Bought	12000.00
4440005	NULL	1110003	2220000	MRTYHBFT46STER34A	2022-11-11	Bought	6000.00
4440006	NULL	1110004	2220000	MFHTY6STE5TDFT7DH	2022-11-12	Bought	8500.00
4440012	NULL	1110005	2220000	GUNSTHEYTHINCR3TS	2022-11-11	Bought	62000.00

```
7 rows in set (0.001 sec)
```

Figure 13. Display the bought transactions made between Oct and Nov 13, 2022

5. Example of Triggers

The following are some trigger examples implemented in the database

Make	Model	TheYear	VIN	Price	Miles	isNew	Available
Audi	XQ7	2018	GUNSTHEYTHINCR3TS	62000.00	130000.00	No	Yes
Toyota	Camry	2014	MBYHT6ETS43FRT56	21000.00	10000.00	Yes	No
Kia	Optima	2016	MFHTY6STE5TDF7DH	8500.00	201052.00	No	No
Toyota	Corolla	2016	MG85YDHRT45AD3EW4	18000.00	13000.00	Yes	No
Nissan	Sentra	2014	MKVGTHYD65TSE34RA	12000.00	15000.00	Yes	No
Nissan	Juke	2017	MRTYHBFT46STER34A	6000.00	142222.00	No	Yes
Toyota	Tundra	2020	NTERAFDRRT45TDRE	45000.00	12000.00	Yes	No

Figure 14. Displaying Car inventory table to show that a car is available before selling it.

PhoneNumber	Position	Salary	Email
0	CEO	1000.00	admin@
2147483647	Salesman	75000.00	Lmba@gmail.com
1236547896	Manager	95000.00	Asmith@outlook.com
2147483647	Assistant	60000.00	OcamposS@gmail.com
1254786325	CEO	120000.00	scote@gmail.com

Figure 16. Displaying the table employee to show the initial salary of an employee before the bonus.

Make	Model	TheYear	VIN	Price	Miles	isNew	Available
Audi	XQ7	2018	GUNSTHEYTHINCR3TS	62000.00	130000.00	No	Yes
Toyota	Camry	2014	MBYHT6ETS43FRT56	21000.00	10000.00	Yes	No
Kia	Optima	2016	MFHTY6STE5TDF7DH	8500.00	201052.00	No	No
Toyota	Corolla	2016	MG85YDHRT45AD3EW4	18000.00	13000.00	Yes	No
Nissan	Sentra	2014	MKVGTHYD65TSE34RA	12000.00	15000.00	Yes	No
Nissan	Juke	2017	MRTYHBFT46STER34A	6000.00	142222.00	No	No
Toyota	Tundra	2020	NTERAFDRRT45TDRE	45000.00	12000.00	Yes	No

Figure 15. Displaying Car inventory table to show that the car is not available anymore after selling it.

PhoneNumber	Position	Salary	Email
0	CEO	2000.00	admin@
2147483647	Salesman	75000.00	Lmba@gmail.com
1236547896	Manager	95000.00	Asmith@outlook.com
2147483647	Assistant	60000.00	OcamposS@gmail.com
1254786325	CEO	120000.00	scote@gmail.com

Figure 17. Displaying the table employee to show the final salary of the employee after applying the bonus.

Trigger name:

Table:

Time:

Event:

```

1 BEGIN
2   if new.SOB = 'Sold' THEN
3     UPDATE carinventory
4       SET Available = 'No'
5       WHERE VIN = new.VIN;
6   END IF;
7 END
  
```

Figure 18. Code for the previous trigger that shows if a car still available for sale or not

bonus

transactions

AFTER

INSERT

```

1 BEGIN
2   IF new.SOB = 'Sold'
3   THEN
4     IF NEW.Final_price >= 50000 and NEW.Final_price < 90000
5     THEN
6       UPDATE employee
7       SET salary = salary + 1000
8       WHERE EmployeeID = new.Employee;
9     ELSEIF NEW.Final_price > 90000
10    THEN
11      UPDATE employee
12      SET salary = salary + 2000
13      WHERE EmployeeID = new.Employee;
14    END IF;
15  END IF;
16 END IF;
  
```

Figure 19. Code for the previous trigger that shows the bonus applied to an employee

6. Discussion & Conclusion

In conclusion, the Car DealershipDB is a very helpful tool for car dealership companies. It will not just help them store their information but also secure it. Security is very important when dealing with sensitive data, such as card information, addresses, phone numbers, etc. The overall idea of this database is to keep track of all the transactions that the company performs. This would include either buying cars from individuals, from other companies and auctions, or selling them to any individual that would like to buy them.

During the project, we faced several challenges, including time conflicts, given that sometimes it took an effort to find the right time that worked for everyone. However, we did the project within the established timeline and finished it promptly. Another main challenge we faced was figuring out the code. For example, the makeSale function was not returning any value because the syntax was incorrect. To solve this problem and overcome other challenges, we looked for information from different sources, including the course book and some websites that helped us greatly.

Finally, we needed to create an interface where the users would interact with the database. We decided to create a website to help them do it quickly and efficiently.

Overall, creating the database was a great, informative experience because we learned a lot, and it helped us improve our coding and teamwork skills.

7. Group Statement

Francisco Ekang Mofuman created the web interface that the database can be accessed and modified from. He also played a big role in structuring the database and making sure that everything was up to par.

Francisco Ndong Nengono was Francisco Ekang's right-hand man in creating the web interface. He not only assisted with the HTML, but also created some of the php files used. Francisco Ndong also took the lead on creating triggers that facilitate the process of rewarding exceptional employees.

Joaquin Owono Afugu Ntoo took charge of compiling all the work our group has done into a comprehensive report that includes screenshots and extensive explanations of our database. He also transformed our list of business queries into usable views and view tables that are utilized in our web interface.

Marcus Rockwell focused more on the backend, creating various functions and procedures that were then integrated into the web interface. He also assisted the other group members with debugging and syntax problems as well as proofreading for the report.

References

Zybooks, <https://learn.zybooks.com/zybook/PVAMUCOMP3395KimFall2022>.

“SQL Views.” *W3Schools*, https://www.w3schools.com/sql/sql_view.asp.

Zegarek, “Trigger for Boolean Update (Postgresql).” *Stack Overflow*, 31 Dec. 1969,
<https://stackoverflow.com/questions/74422532/trigger-for-boolean-update-postgresql>.