

# **Toyota Dealership Database Website**

## **Final Report**

**Jiashan Li**

**Stephanie Ross**

**Shanel Sayson**

**Muzhou Zhai**

## **Abstract**

Our project has created a Toyota dealership database website. Firstly, a car dealership is a place that provides vehicles' information for customers who are wanting to buy cars. There are some roles which are called "entities" that work to support the car dealership such as customers, employees, and vehicles. What's more, there are a number of attributes and relationships based on these entities, such as employees sell vehicles and customers buy vehicles. All these entities, attributes and the relationships form the database system, and our project is aiming at creating a database system that can manage all the data clearly. Secondly, for the database structure we have 8 entities which are dealerships, employee, branch office, contracts, vehicles, truck, sedan, and SUV. Each of these entities has some attributes attached to themselves. The entities truck, sedan, and SUV are the specializations of the entity vehicles, and they inherit all attributes from vehicles along with their unique attributes. There are many relationships and cardinalities between different entities, so that the stability and the complexity of system structure can be guaranteed. Lastly, we used Structured Query Language (SQL), Hypertext Preprocessor (PHP), and Hyper Text Markup Language (HTML) to develop our online database system. The system is basically combined with the front interface and the background database system. The front interface is an online user interface that allows employees of the car dealership to log in. Customers' information can be added to the system by employees, their data will be stored and managed in the database system. The background database system can do some basic managements and operations to the data that stored in the system. For example, the system will show the list of vehicles that link to a certain employee by information searching.

## **Introduction**

The project is an online system that stores and manages the data from a Toyota dealership. We designed a system that contains a front system and a background database system. Firstly, the front system is an online login interface for employees. We use Hypertext Preprocessor (PHP), and Hyper Text Markup Language (HTML) to develop the front system. Employees need to log into the system by user IDs and passwords, so that the system can get the authentication of the employees, managers, and customers. Only the authorized users can use the database. Secondly, the background database system is a database system that stores and manages all the data from the dealership. The data contains data from dealerships, employee, branch office, contracts, vehicles, truck, sedan, and SUV. All the data is being recorded and updated to the system by employees.

For the development of the background database, we use the Structured Query Language tool adding tables, fields, and foreign keys to generate Structured Query Language (SQL) files. We also created some data that stored in the database, such as the customer's information and the vehicle's information, so that the background database system allows users to search and find data by information searching. For example, the manager can see the list of vehicles that link to a certain employee by searching that employee's first name and last name. An employee can find the vehicles that a customer purchased by searching his or her cell phone number.

## **Project Design Section**

### Administration

The administration will be able to login using their administration login information. Once they have logged in they will be able to add new employees and search various transactions. Right at login, they will have a complete view of their total sales and amount of vehicles sold in the month. The administration may insert a new employee using employee ID, employee SIN, name of the employee, the supervisor ID for this particular employee, the branch number of the dealership they got hired at, and the dealership they were hired for. The search transaction link allows for them to see which employee sold what vehicle based on the transaction ID. This is helpful for determining which employee is doing well in sales at which dealership and to determine any fraudulent acts, if any were to arise.

### Employee

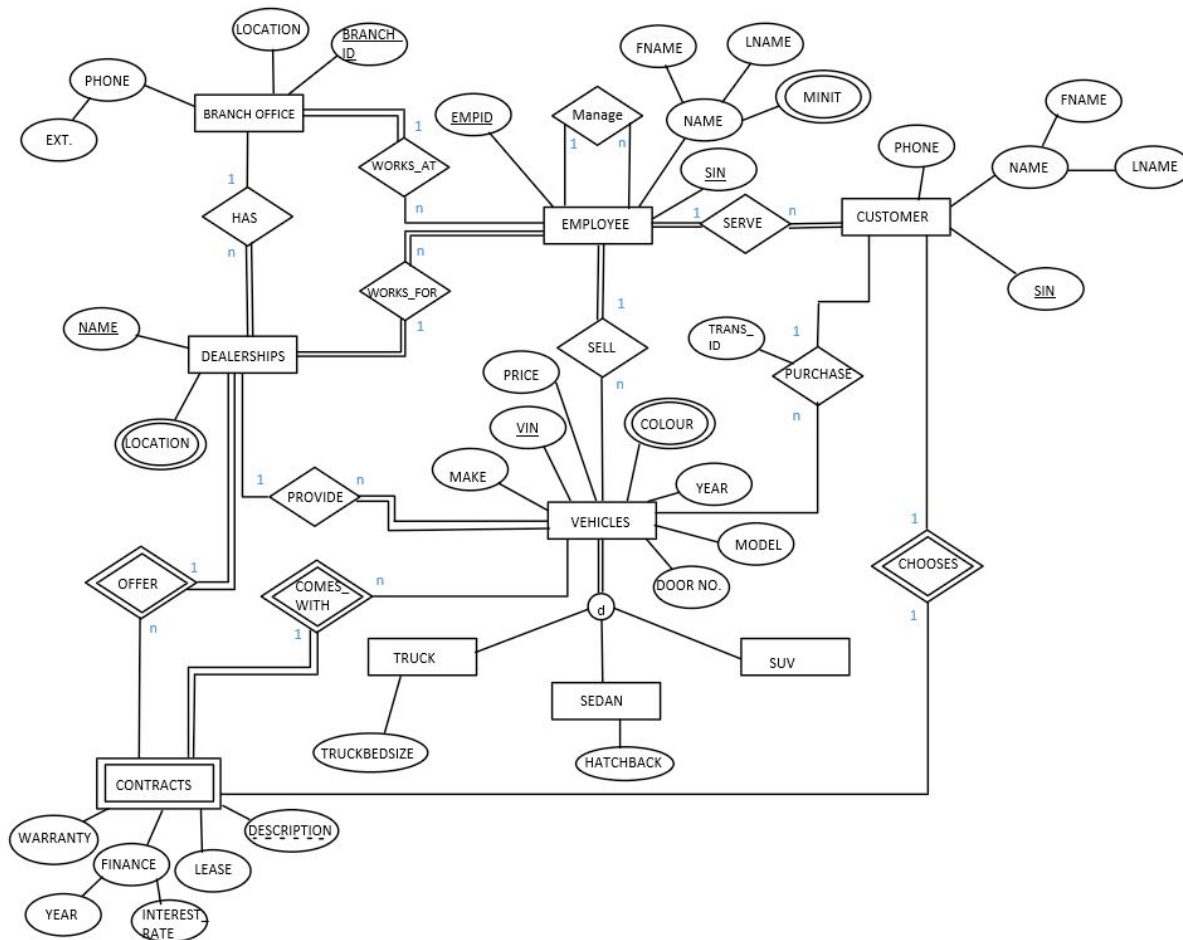
The employee will have access to the system in which they will login using their employee login information. After login, they will be able to view a history of what they had sold. This being the vehicle model, vehicle colour, and the amount sold. They will be able to alter their history where they may insert a new customer that they sold a particular vehicle to. A new window will occur where the submission consists of the employee ID, vehicle identification number (VIN), customer social insurance number (SIN), customer name, contract description, and whether they are on lease, finance or warranty. After they have indicated their desired information, they will

click the submit button, it will insert the values into the following entities: customer, sell, and contracts, and confirm to the user that it has been submitted.

### Customer

The customer is restricted after login, they cannot alter any tables other than their own. Only one attribute may be updated by the customer and that is their phone number. If a customer was to change their number or provided the company with an invalid number, they may login using their customer login information and change their phone number accordingly. A customer may also view their purchase history which is composed of their transaction ID, the vehicle identification number (VIN) of their purchased car and the amount they paid. They may also view their lease information (if any) or financial information (amount left to pay, if any).

## EER-Diagram



Changes made since presentation: Created a CONTRACTS entity along with the following relationships: one customer may *choose* one contract in which vehicles *come with* and one dealership *offers* multiple contracts. The contracts entity is a weak entity type with a partial key attribute, DESCRIPTION. The DESCRIPTION consists of a brief summary of each contract and is of datatype char(n) for any n matching description type. For example, one description for a contract would determine the date it is issued along with the date it will end (if it is to end or be dropped). Firstly, WARRANTY of data type integer, can be determined in years. A customer

may choose either a 1, 2, or 3-year warranty for their vehicle. Secondly, FINANCE (composing of YEAR and INTEREST\_RATE) is also of data type integer in which a customer may finance a vehicle based on a particular interest rate or based on the year it must be fully paid. Lastly, a customer may decide to LEASE a vehicle, note they do not own this vehicle as it is still under the dealership. Leasing is determined in years and it is of datatype integer. In addition, created a PRICE attribute to the VEHICLES entity in order to distinguish the total amount sold by each employee, a SIN attribute to the EMPLOYEE entity so that the admin may distinguish between different employees, and a PHONE attribute in which a customer may update their phone number.

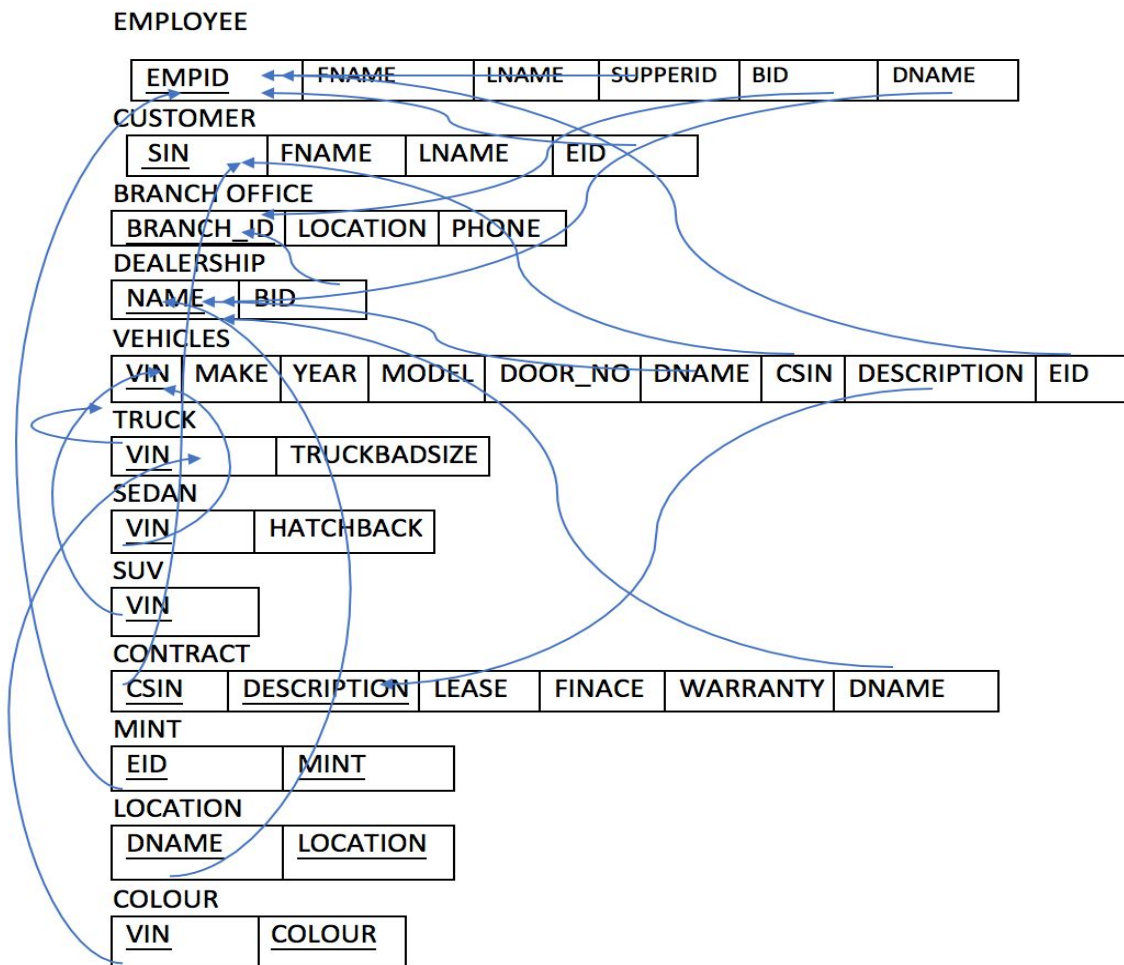
#### Assumptions:

A customer may purchase many vehicles: If a purchase relationship is made, they will receive a TRANS\_ID for that particular purchase. Not every customer will purchase a vehicle and not every vehicle would be sold to a customer.

Each vehicle is mapped to the specializations, TRUCK, SEDAN, and SUV and they are disjoint since a truck cannot also be a sedan, a sedan cannot be an suv, and so on.

The finance attribute represented in the EER diagram is composed of YEAR and INTEREST\_RATE, for the purpose of simplification our implementation uses the finance attribute as an outstanding balance to be paid off by a customer.

## Implementation Section



The image above shows the relational schema diagram of the database. In this process, there are three multi-valued attributes, the middle name of the employee, the location of the dealership and the colour of the vehicle. We created three new relations, one for each of them. Each relation includes the primary key attribute (as a foreign key) of the relation that represents the entity type of relationship and the multi-valued attribute name. In these relations, we can clearly see the value of multi-valued attributes. There are also three specialized entities for the vehicle entity. Truck, SUV and sedan. For each of them, we took them as multi-valued attributes, we created



three new entities and used vehicle number as primary key for all three entities and add their own unique attributes.

We implemented our database using MySQL.

### Customer

This end user has the most limited access to the database of all the end users. They can only use the database to view their purchase info. The following queries provide results that will be visible to this user:

```
$sql = "SELECT p.TRANSID, p.VIN, v.PRICE, v.MAKE, v.MODEL, v.YEAR  
        FROM purchase AS p, vehicles AS v, customer AS c  
        WHERE v.VIN = p.VIN AND p.CSIN = c.SIN"
```

```
$sql = "SELECT con.LEASE, con.FINANCE  
        FROM contracts AS con, customer AS cust  
        WHERE con.CSIN = cust.Sin"
```

Update phone number:

The customer may update their phone number by entering a new phone number into the submit form. Will update old phone number to the new phone number:

```
$phone = $_POST["phone"];  
  
$sql = "UPDATE customer  
        SET phone = $phone  
        WHERE Sin = $sin";
```

## Employee

When the employee logs in, they can automatically see how many cars they've sold, and how much they've made in sales, This is the query (employee id was hardcoded in as employee #8 for easier demonstration):

```
SELECT COUNT(*) as 'total', SUM(PRICE) AS 'sum' FROM sell, vehicles WHERE  
sell.eid = 13 AND sell.VIN = vehicles.VIN
```

The employee can also see the history of their sales in a table. This is the query:

```
SELECT * FROM sell, vehicles, purchase WHERE sell.eid = 8 AND sell.vin =  
purchase.VIN AND sell.vin = vehicles.VIN
```

This end user is able to add new sales to the database. This is done by creating a new customer record, a new sell record, a new purchase record, and a new contracts record. This is achieved by the following queries (the \$\_POST variables are from user input on an HTML form):

```
$sql1 = "INSERT INTO sell VALUES (" .  
$_POST["eid"] . "," .  
$_POST["vin"] . ")";
```

```
$sql2 = "INSERT INTO purchase VALUES (" .  
$_POST["sin"] . "," .  
$_POST["vin"] . "," .  
$_POST["transid"] . ")";
```

```
$sql3 = "INSERT INTO customer(Sin,CName,eid) VALUES (" .  
$_POST["sin"] . "," .  
$_POST["CName"] . "," .  
$_POST["eid"] . ")";
```

```
$sql4 = "INSERT INTO contracts VALUES (" .  
    $_POST["DESCRIPTION"] . ",'" .  
    $_POST["WARRANTY"] . "''," .  
    $_POST["LEASE"] . ",'" .  
    $_POST["FINANCE"] . "');" ;
```

## Administrator

This end user has the least amount of restriction of all the end users. The administrator (who holds a high position in the company) will use the database to add new employees, generate statistics on the company's sales, and look up specific transactions.

Upon logging in, the administrator is greeted with the total number of cars sold to date in the company, as well as the current company sales to date. This is done by the following query:

```
SELECT COUNT(*) as 'total', SUM(PRICE) AS 'sum'
FROM sell, vehicles
WHERE sell.VIN = vehicles.VIN
```

The manager uses form submission to add a new employee to the database upon hire. This is achieved by the following query (the \$\_POST variables are from user input on an HTML form):

```
$sql = "INSERT INTO employee VALUES (" .
$_POST["id"] . "," .
$_POST["sin"] . "," .
$_POST["name"] . "," .
$_POST["super_id"] . "," .
$_POST["branch_no"] . "," .
$_POST["dealership"] . ")";
```

Transaction lookup:

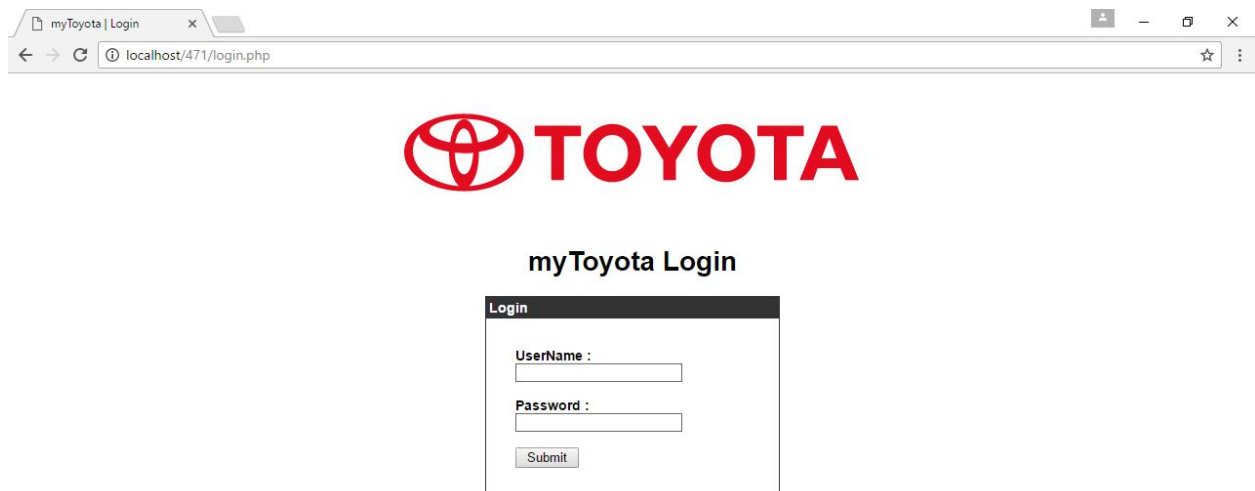
The admin can locate a transaction by looking up the Transaction ID. This is done through the following transaction:

```
SELECT * FROM sell, vehicles, employee, customer, purchase
WHERE purchase.TRANSID = " . $_POST["trans_id"] . " AND
purchase.VIN = sell.vin AND
customer.Sin = purchase.CSIN AND
sell.vin = vehicles.VIN AND
sell.eid = employee.EMPID
```

## User Section

The website has a very minimalistic UI. This is due to it mostly being used for simple data entry and queries. Also, given our lack of expertise in the languages used (PHP and bits of JavaScript), having a simple design allowed us to set realistic goals for what we could implement in the given time frame. We kept black text on white backgrounds, and form fields, buttons and links are self-descriptive to prevent any confusion from users. The following are screenshots that demonstrate the interface that we have designed and how users interact with it.

The login page:



The screenshot shows a web browser window with the title "myToyota | Login" and the address bar displaying "localhost/471/login.php". The page features the Toyota logo in red, followed by the text "myToyota Login" in bold. Below this is a login form with a black header bar labeled "Login". The form contains two input fields: "UserName :" and "Password :", each followed by a text box. A "Submit" button is located at the bottom of the form.

myToyota | Login

localhost/471/login.php

**TOYOTA**

**myToyota Login**

Login

UserName :

Password :

Submit

## Customer

This is what the customer sees after logging in.



### myToyota Customer

Welcome, Andrew Lata! | [Sign Out](#)

#### Customer Purchase History

TRANSID	VIN	Price	Make	Model	Year
12343	10004	27680	TOYOTA	Camry CE	2016

You are on a 7 year lease  
Your finance amount dropped to \$17000

Want to update your phone number?

Fill out your new number below:


Updated phone number to 4031119999

## Employee

This is the screen that an employee sees after logging in:

myToyota | Sales Representative

localhost/471/employee.php



**myToyota Sales Representative**

Welcome, Lisa Scott! | [Sign Out](#)

New Sale

Vehicles sold so far: 1  
Total sales: \$26,780

Trans No.	Make	Model	Year	Price
12347	TOYOTA	Camry LE	2017	26780



This is the form the employee fills out when completing a new car sale to a customer:



## myToyota | New Sale

Enter new purchase information:

Employee ID	VIN	Customer SIN	Customer Name	Contract Description	Warranty	Lease
Finance	Submit					

## Administrator

This is the screen that the admin sees after logging in:



Adding a new employee:



## myToyota Management

### New employee file

Enter new hire information:

<input type="text" value="ID"/>	<input type="text" value="SIN"/>	<input type="text" value="Name"/>	<input type="text" value="Supervisor ID"/>	<input type="text" value="Branch No."/>	<input type="text" value="Dealership"/>	<input type="button" value="Submit"/>
---------------------------------	----------------------------------	-----------------------------------	--	---	---	---------------------------------------

Looking up a transaction:



## myToyota Management

### Transaction Search

Enter Trans #

Dealership	Employee	Customer	Make	Year	Model
COCKRAM	Daniel	Allen Zhang	TOYOTA	2017	Camry LE