CS 6400 Database Systems Concepts and Design

Team 22 - Phase 2 Report

Jinjun Liu (jliu788)

Zijian Xie (zxie86)

Hui Xia (hxia40)

Chen Zhang (czhang613)

Abstract Code w/SQL

<u>Public Search</u>

Abstract Code:

- Show the total number of vehicles that Repair.*repair_status* != “pending” or “In progress” and are not existed in Sell table.

```
SELECT COUNT(vin)
FROM Vehicle LEFT JOIN Repair ON Vehicle.vin=Repair.vin
WHERE vin NOT IN (SELECT vin FROM Sell)
AND repair_status != 'pending' AND repair_status != 'In progress';
```

- Show drop-down menus for vehicle type, manufacturer, model year and color selections.
- Show blank field for Keyword input.
- Show ***Search*** button and ***Login*** button.
- User enters vehicle type ($entered_type_name), manufacturer ($entered_manufacturer_name), model year ($entered_model_year), color ($entered_vehicle_color) or keyword ($entered_keyword).
- If data validation is successful for all the input fields, then:
    - ❖ When ***Search*** button is clicked:

```
SELECT vin, type_name, model_year, manufacturer_name, vehicle_color, vehicle_mileage, sale_price
FROM Vehicle LEFT JOIN Repair ON Vehicle.vin=Repair.vin
LEFT JOIN VehicleColor ON VehicleColor.vin=Vehicle.vin
WHERE vin NOT IN (SELECT vin FROM Sell)
AND repair_status != 'pending' AND repair_status != 'In progress'
AND
(
type_name='$entered_type_name'
AND manufacturer_name='$entered_manufacturer_name'
AND model_year='$entered_model_year'
AND vehicle_color='$entered_vehicle_color'
AND
(
manufacturer_name LIKE '%$keyword%'
OR model_year LIKE '%$keyword%'
OR model_name LIKE '%$keyword%'
OR vehicle_description LIKE '%$keyword%'
)
)
ORDER BY vin ASC;
```

    - ❖ If no record is found, show error message “Sorry, it looks like we don’t have that in stock!”
- Upon:
    - ❖ User clicks an individual result - Go to **<u>View Vehicle Detail</u>** task;
    - ❖ User clicks ***Employee Login*** button - Go to **<u>Login</u>** task.

<u>Login</u>

Abstract Code:

- User clicked on ***Employee Login*** from **<u>Vehicle Search Form</u>**.

- Show ***Login*** and ***Cancel*** button.
- User enters *username* ($username), *password* ($password) input fields.
- If data validation is successful for both *username* and *password* input fields, then:
    - ❖ When ***Login*** button is clicked:

```
SELECT password FROM Users WHERE Users.username = '$username';
```

- ❖ If User record is not found, or User record is found but Users.*password* != '$password':
    - ➢ Clear the input fields, with error message.
- ❖ Else:
    - ➢ Store login information as session variable '$UserID'.
    - ➢ Go to **Employee Search** task.
- ❖ When ***Cancel*** button is clicked - Go to **Public Search** task.

Employee Search

Abstract Code:

- User logged in successfully.
- Show ***Search*** and ***Logout*** button.
- Determine the permission of logged in user.
- If '$UserID' is in InventoryClerk table, then:
    - ❖ Show ***Add Vehicle*** button.
    - ❖ Show number of vehicles with repairs pending, in progress and vehicles available for purchase

```
// show number of vehicles with repairs pending
SELECT Count(vin) FROM Repair WHERE repair_status ='pending';
// show number of vehicles with repairs in progress
SELECT Count(vin) FROM Repair WHERE repair_status ='In progress';
// show number of vehicles available for purchase
SELECT COUNT(vin)
FROM Vehicle LEFT JOIN Repair ON Vehicle.vin=Repair.vin
WHERE vin NOT IN (SELECT vin FROM Sell)
AND repair_status != 'pending' AND repair_status != 'In progress';
```

- ❖ User enters vehicle type ($entered_type_name), manufacturer ($entered_manufacturer_name), model year ($entered_model_year), color ($entered_vehicle_color), keyword ($keyword) or VIN ($entered_VIN).
- ❖ If data validation is successful for all the input fields, then:
    - ➢ when the ***search*** button is clicked:

```
SELECT vin, type_name, model_year, manufacturer_name, vehicle_color, vehicle_mileage, sale_price
FROM Vehicle LEFT JOIN Repair ON Vehicle.vin=Repair.vin
LEFT JOIN VehicleColor ON VehicleColor.vin=Vehicle.vin
WHERE vin NOT IN (SELECT vin FROM Sell)
AND
(
type_name='$entered_type_name'
AND manufacturer_name='$entered_manufacturer_name'
AND model_year='$entered_model_year'
AND vehicle_color='$entered_vehicle_color'
AND vin='$entered_VIN'
AND
(
manufacturer_name LIKE '%$keyword%'
OR model_year LIKE '%$keyword%'
OR model_name LIKE '%$keyword%'
OR vehicle_description LIKE '%$keyword%'
)
)
ORDER BY vin ASC;
```

        ➢ If no record is found, show error message "Sorry, it looks like we don't have that in stock!"
    ❖ Upon:
        ➢ User clicks an individual result - Go to **View Vehicle Detail** task;
        ➢ User clicks ***Add Vehicle*** button - Go **Add Vehicle** task;

- If '$UserID' is in Salesperson table, then:
    ❖ Show ***Look up customer*** button.
    ❖ User enters vehicle type ($entered_type_name), manufacturer ($entered_manufacturer_name), model year ($entered_model_year), color ($entered_vehicle_color), keyword ($keyword) or VIN ($entered_VIN).
    ❖ If data validation is successful for all the input fields, then:
        ➢ when the ***search*** button is clicked:

```
SELECT vin, type_name, model_year, manufacturer_name, vehicle_color, vehicle_mileage, sale_price
FROM Vehicle LEFT JOIN Repair ON Vehicle.vin=Repair.vin
LEFT JOIN VehicleColor ON VehicleColor.vin=Vehicle.vin
WHERE vin NOT IN (SELECT vin FROM Sell)
AND repair_status != 'pending' AND repair_status != 'In progress'
AND
(
type_name='$entered_type_name'
AND manufacturer_name='$entered_manufacturer_name'
AND model_year='$entered_model_year'
AND vehicle_color='$entered_vehicle_color'
AND vin='$entered_VIN'
AND
(
manufacturer_name LIKE '%$keyword%'
OR model_year LIKE '%$keyword%'
OR model_name LIKE '%$keyword%'
OR vehicle_description LIKE '%$keyword%'
)
)
ORDER BY vin ASC;
```

        ➢ If no record is found, show error message "Sorry, it looks like we don't have that in stock!"
    ❖ Upon:
        ➢ User clicks an individual result - Go to **View Vehicle Detail** task;
        ➢ User clicks ***Look up customer*** button - Go **Add Customer** task;
- If '$UserID' is in Manager table, then:
    ❖ Show ***View Monthly Sales Report, View Repair Statistics Report, View Price Per Condition Report, View Average Time in Inventory Report, View Inventory Age Report, View Seller History Report*** button.
    ❖ Show number of vehicles with repairs pending, in progress and vehicles available for purchase

```
// show number of vehicles with repairs pending
SELECT Count(vin) FROM Repair WHERE repair_status ='pending';
// show number of vehicles with repairs in progress
```

```
SELECT Count(vin) FROM Repair WHERE repair_status ='In progress';
// show number of vehicles available for purchase
SELECT COUNT(vin)
FROM Vehicle LEFT JOIN Repair ON Vehicle.vin=Repair.vin
WHERE vin NOT IN (SELECT vin FROM Sell)
AND repair_status != 'pending' AND repair_status != 'In progress';
```

- ❖ User enters vehicle type ($entered_type_name), manufacturer ($entered_manufacturer_name), model year ($entered_model_year), color ($entered_vehicle_color), keyword ($keyword) or VIN ($entered_VIN).
- ❖ If data validation is successful for all the input fields, then:
    - ➢ when the ***search*** button is clicked and “all vehicles” filter is chosen:

```
SELECT vin, type_name, model_year, manufacturer_name, vehicle_color, vehicle_mileage, sale_price
FROM Vehicle LEFT JOIN Repair ON Vehicle.vin=Repair.vin
LEFT JOIN VehicleColor ON VehicleColor.vin=Vehicle.vin
AND
(
type_name='$entered_type_name'
AND manufacturer_name='$entered_manufacturer_name'
AND model_year='$entered_model_year'
AND vehicle_color='$entered_vehicle_color'
AND vin='$entered_VIN'
AND
(
manufacturer_name LIKE '%$keyword%'
OR model_year LIKE '%$keyword%'
OR model_name LIKE '%$keyword%'
OR vehicle_description LIKE '%$keyword%'
)
)
ORDER BY vin ASC;
```

- ➢ when the ***search*** button is clicked and “sold vehicles” filter is chosen:

```
SELECT vin, type_name, model_year, manufacturer_name, vehicle_color, vehicle_mileage, sale_price
FROM Vehicle LEFT JOIN Repair ON Vehicle.vin=Repair.vin
LEFT JOIN VehicleColor ON VehicleColor.vin=Vehicle.vin
WHERE vin IN (SELECT vin FROM Sell)
AND
(
type_name='$entered_type_name'
AND manufacturer_name='$entered_manufacturer_name'
AND model_year='$entered_model_year'
AND vehicle_color='$entered_vehicle_color'
AND vin='$entered_VIN'
AND
(
manufacturer_name LIKE '%$keyword%'
OR model_year LIKE '%$keyword%'
OR model_name LIKE '%$keyword%'
OR vehicle_description LIKE '%$keyword%'
)
)
ORDER BY vin ASC;
```

- ➢ when the ***search*** button is clicked and “unsold vehicles” filter is chosen:

```
SELECT vin, type_name, model_year, manufacturer_name, vehicle_color, vehicle_mileage, sale_price
FROM Vehicle LEFT JOIN Repair ON Vehicle.vin=Repair.vin
LEFT JOIN VehicleColor ON VehicleColor.vin=Vehicle.vin
WHERE vin NOT IN (SELECT vin FROM Sell)
AND
(
type_name='$entered_type_name'
AND manufacturer_name='$entered_manufacturer_name'
AND model_year='$entered_model_year'
AND vehicle_color='$entered_vehicle_color'
AND vin='$entered_VIN'
AND
(
manufacturer_name LIKE '%$keyword%'
OR model_year LIKE '%$keyword%'
OR model_name LIKE '%$keyword%'
OR vehicle_description LIKE '%$keyword%'
)
)
ORDER BY vin ASC;
```

- ➢ If no record is found, show error message “Sorry, it looks like we don’t have that in stock!”

❖ Upon:
- ➢ User clicks an individual result - Go to **View Vehicle Detail** task;
- ➢ User clicks ***View Monthly Sales Report*** button - Go to **Monthly Sales Report** task;
- ➢ User clicks ***View Repair Statistics Report*** button - Go to **Repair Statistics Report** task;
- ➢ User clicks ***View Average Time in Inventory Report*** button - Go to **Average Time in Inventory Report** task;
- ➢ User clicks ***View Price Per Condition Report*** button - Go to **Price Per Condition Report** task;
- ➢ User clicks ***View Inventory Age Report*** button - Go to **Inventory Age Report** task;
- ➢ User clicks ***View Seller History Report*** button - Go to **Seller History Report** task;

View Vehicle Detail Form

Abstract Code

- Show ***View Vehicle Detail*** on public search or employee search result list
- Upon:
  - ❖ Click ***View Vehicle Detail*** button:
    - ➢ View physical properties, repair information, and/or transaction information as stated in Enabling Conditions;
    - ➢ For public view (do not login, no permission assigned):

If '$UserID' **not in** InventoryClerk.username AND '$UserID' **not in** Manager.username AND '$UserID' **not in** Salesperson.username

```
// assume $entered_VIN of current vehicle is managed by application
// assume $UserID of current user is managed by application
SELECT Vehicle.vin, vehicle_mileage, vehicle_description, model_name, model_year, manufacturer_name,
vehicle_color, sale_price
FROM `Vehicle` JOIN VehicleColor ON Vehicle.vin = VehicleColor.vin
JOIN Repair ON Vehicle.vin = Repair.vin
WHERE repair_status = 'complete' AND Vehicle.vin = '$entered_VIN'
```

```
ORDER BY Vehicle.vin ASC;
```

➢ For clerk permission view:

If '$UserID' **in** InventoryClerk.username AND '$UserID' **not in** Manager.username AND '$UserID' **not in** Salesperson.username:

```
// assume $entered_VIN in of current vehicle is managed by application
// assume $UserID of current user is managed by application
SELECT Vehicle.vin, vehicle_mileage, vehicle_description, model_name, model_year,
manufacturer_name, vehicle_color, sale_price, start_date, end_date, repair_status, repair_description,
repair_cost, vendor_name, Repair.nhtsa_recall_compaign_number, Buy.inventory_clerk_permission, purchase_price
FROM Vehicle JOIN VehicleColor ON Vehicle.vin = VehicleColor.vin
JOIN Buy on Vehicle.vin = Buy.vin
JOIN Repair on Vehicle.vin = Repair.vin
WHERE Vehicle.vin = '$entered_VIN'
ORDER BY Vehicle.vin ASC;
```

- Show ***Repair Description*** pop-out button;
- Show ***Add Repair, Edit Repair, Delete Repair*** button;

➢ For salesperson permission view:

If '$UserID' in Salesperson.username AND '$UserID' not in InventoryClerk.username AND '$UserID' not in Manager.username:

```
// assume $entered_VIN of current vehicle is managed by application
// assume'$UserID of current user is managed by application
SELECT Vehicle.vin, vehicle_mileage, vehicle_description, model_name, model_year, manufacturer_name,
vehicle_color, sale_price
FROM `Vehicle` JOIN VehicleColor ON Vehicle.vin = VehicleColor.vin
JOIN Repair ON Vehicle.vin = Repair.vin
WHERE repair_status = 'complete' AND Vehicle.vin = '$entered_VIN'
ORDER BY Vehicle.vin ASC;
```

- Show ***Repair Description*** pop-out button;
- Show ***Sell Vehicle*** button;

➢ For manager view:

If '$UserID' **in** Manager.username AND '$UserID' **not in** InventoryClerk.username AND '$UserID' **not in** Salesperson.username:
If '$entered_VIN' in Sell.vin:

```
// assume $entered_VIN of current vehicle is managed by application
// assume $UserID of current user is managed by application
SELECT Vehicle.vin, vehicle_mileage, vehicle_description, model_name, model_year, manufacturer_name,
vehicle_color, sale_price,
start_date, end_date, repair_status, repair_description, repair_cost, vendor_name, nhtsa_recall_compaign_number,
Buy.inventory_clerk_permission, Buy.customer_id, purchase_price, purchase_condition
FROM Vehicle JOIN VehicleColor ON Vehicle.vin = VehicleColor.vin
JOIN BUY on Vehicle.vin = Buy.vin
JOIN Repair on Vehicle.vin = Repair.vin
WHERE Vehicle.vin = '$entered_VIN'
ORDER BY Vehicle.vin ASC;
```

If '$entered_VIN' not in Sell.vin:

```
// assume $entered_VIN of current vehicle is managed by application
// assume $UserID of current user is managed by application
SELECT Vehicle.vin, vehicle_mileage, vehicle_description, model_name, model_year, manufacturer_name,
vehicle_color, sale_price,
start_date, end_date, repair_status, repair_description, repair_cost, vendor_name, nhtsa_recall_compaign_number,
Buy.inventory_clerk_permission, Buy.customer_id, purchase_price, purchase_condition,
Sell.salesperson_permission, Sell.customer_id, sale_date, sale_price
FROM Vehicle JOIN VehicleColor ON Vehicle.vin = VehicleColor.vin
JOIN BUY on Vehicle.vin = Buy.vin
JOIN Repair on Vehicle.vin = Repair.vin
WHERE Vehicle.vin = '$entered_VIN'
ORDER BY Vehicle.vin ASC;
```

- Show total amount of cars in different repair status;

Amount fo cars under repair:

```
// assume $entered_VIN of current vehicle is managed by application
// assume $UserID of current user is managed by application
SELECT Count (vin) FROM Repair WHERE repair_status ='In progress';
```

Amount of cars pending for repair:

```
// assume $vin of current vehicle is managed by application
// assume $username of current user is managed by application
SELECT Count (vin) FROM Repair WHERE repair_status ='pending';
```

Amount of cars ready for sell:

```
// assume $vin of current vehicle is managed by application
// assume $username of current user is managed by application
SELECT Count (vin) FROM Repair WHERE repair_status ='complete';
```

- ❖ Click ***Done*** button - Go back to search result list.

<u>**Add Customer Form**</u>

Abstract Code:

- Show ***Add Customer*** button upon clicking ***Sell Vehicle*** button by salesperson or ***Add Vehicle*** button by inventory clerk.
- Upon:
    - ❖ Click ***Add Customer*** button
        - ➢ Run Search Customer subtask, user select person or business in dropdown menu input customer's driver_lisence_number (as a person) or Tax_identification_number (as a business);
        - ➢ If person is selected, user input customer's driver_lisence_number
        - ➢ If input in Person.driver_lisence_number:
            - ■ View Customer;

```
SELECT Customer.customer_id, driver_license_number, customer_first_name, customer_last_name, phone_number,
email, customer_street, customer_city, customer_state, customer_zip
FROM Person JOIN Customer ON Person.customer_id = Customer.customer_id WHERE
Person.drivces_license_bunber = '$entereddriver_license_number';
```

- ■ Show ***Edit Customer*** button;

        - Upon Click ***Edit Customer***:
            - User enters customer information , run edit customer subtask
            - Click ***Save*** button: Update Customer, Person Table; Return to View Customer;

```
// assume current customer_id is managed by application
//Use Customer, Person;
UPDATE Person
SET driver_licesnce_number = '$entereddriver_license_number',
            customer_first_name = '$enteredcustomer_first_name',
            customer_last_name = '$enteredcustomer_last_name'
WHERE customer_id = '$customer_id';

UPDATE Customer
SET Customer.phone_number = '$enteredphone_number',
phone_number = '$enteredphone_number',
email = '$enteredemail',
customer_street = '$enteredcustomer_street',
customer_city = '$enteredcustomer_city',
customer_state= '$enteredcustomer_state',
customer_state= $'enteredcustomer_zip'
WHERE customer_id = '$customer_id';;
```

            - Click ***Cancel*** button: Return to View Customer;
    - If input not in Person.driver_lisence_number:
        - Show message ‘new customer!’
        - Show ***Add New Customer*** button;
        - Upon clicking ***Add New Customer*** button:
            - Run add new customer subtask
            - Click ***Save*** button: Write to Customer Table; Return to Add Customer; View customer;
            - Click ***Cancel*** button: Return to **Add New Customer**;

```
//Use Customer, Person;
INSERT INTO Customer
VALUES(
'$enteredcustomer_id', '$enteredphone_number', '$enteredemail',
'$enteredcustomer_street', '$enteredcustomer_city',  '$enteredcustomer_state', '$enteredcustomer_zip'
);

INSERT INTO Person
VALUES(
'$enteredcustomer_id', '$entereddriver_license_number', '$enterecustomer_first_name', '$enteredcustomer_last_name'
);
```

- Click ***Done*** button – Go back to **Add New Vehicle** Form (inventory clerk) or **Sell Vehicle** Form (salesperson)

    - If business is selected, user input customer’s tax_identification_number
    - If input in Business.tax_identification_number:
        - View Customer;

```
SELECT Customer.customer_id, tax_identification_number, business_name, primary_contact_name,
primary_contact_title, phone_number, email, customer_street, customer_city, customer_state, customer_zip
```

```
FROM Business JOIN Customer ON Business.customer_id = Customer.customer_id WHERE
Business.tax_identification_bunber = '$enteredtax_identification_number'
```

- Show ***Edit Customer*** button;
- Upon Click ***Edit Customer***:
    - User enters customer information , run edit customer subtask
    - Click ***Save*** button: Update Customer, Business Table; Return to View Customer;

```
//assume current customer_id is managed by application
//Use Customer, Business;
UPDATE Business
SET Business.tax_identification_number = '$enteredtax_identification_number',
Business.business_name = '$enteredbusiness_name',
Business.primary_contact_name = '$enteredprimary_contact_name',
Business.primary_contact_title = '$enteredprimary_contact_title'
WHERE customer_id = '$customer_id';;

UPDATE Customer
SET Customer.phone_number = '$enteredphone_number',
Customer.phone_number = '$enteredphone_number',
Customer.email = '$enteredemail',
Customer.customer_street = '$enteredcustomer_street',
Customer.customer_city = '$enteredcustomer_city',
Customer.customer_state= '$enteredcustomer_state',
Customer.customer_state= $'enteredcustomer_zip'
WHERE customer_id = '$customer_id';
```

    - Click ***Cancel*** button: Return to View Customer;

➢ If input not in Business.tax_identification_number:

- Show message ‘new customer!’
- Show ***Add New Customer*** button;
- Upon clicking ***Add New Customer*** button:
    - Run add new customer subtask
    - Click ***Save*** button: Write to Customer, Business Table; Return to View customer;
    - Click ***Cancel*** button: Return to **Add New Customer**;

```
//Use Customer, Business;
INSERT INTO Customer
VALUES(
'$enteredcustomer_id', '$enteredphone_number', '$enteredemail',
'$enteredcustomer_street', '$enteredcustomer_city',  '$enteredcustomer_state', '$enteredcustomer_zip'
);

INSERT INTO  Business
VALUES(
'$enteredcustomer_id', '$enteredtax_identification_number',
'$enteredbusiness_name','$entereprimary_contact_name', '$enteredprimary_contact_title'
);
```

❖ Click ***Done*** button – Go back to **Add New Vehicle** Form (inventory clerk) or **Sell Vehicle** Form (salesperson)

Sale Order Form

Abstract Code

- Show Sell Vehicle on vehicle detail screen.
- Upon:
    - User click ***Sell Vehicle*** button;
        - Show ***Add Customer*** button;
        - Upon Add Customer finished, show Sale Order Form;
            - User enters *sale_date*;
            - Click Done button: Write into Sell Table; Display Vehicle detail form;
            - Click ***Cancel*** Button: Return to Vehicle Detail Form;

```
//assume current $vin, #customer_id, $salesperson_permission is managed by application
//Use Sell;
INSERT INTO Sell
VALUES(
'$vin', '$customer_id', '$salesperson_permission', '$sales_date'
);
```

Add Repair Form

Abstract Code:

- Show ***Add Repair Form***, ***Edit Repair Form, View Repair From, Delete Repair Form*** buttons in **Repair/Recall Repository**
- User clicks ***Add Repair Form*** button and the system displays a new repair form table with ***Submit*** and ***Cancel*** button
- User enters vin, vendor_name, vendor_Address, vendor_phone_number, repair_description, repair_start_date, repair_cost, nhtsa_recall_campagin_number, Inventory_clerk_permission
- User selects one of "*Pending*", "*In Progress*" and "*Complete*" in *Repair_Status*
- Upon:
    - Click ***Submit*** button;
        - If nhtsa_recall_campagin_number is not Null and is not in Recall.nhtsa_recall_campagin_number:
            - Show Error Message and Return to **Repair/Recall Repository**
        - Else if all items except nhtsa_recall_campagin_number are not Null:
            - Write to Repair Table and go back to **Repair/Recall Repository**

```
/*Assume that the inputs are correct and the permission has been validated by the application.*/

INSERT INTO `Repair`
VALUES(
'$enteredvin', '$enteredstart_date', '$enteredend_date',
'$enteredrepair_status', '$enteredrepair_description', '$enteredvendor_name',
'$enteredrepair_cost', '$enteredNHTSA_recall_campagin_Number', '$enteredinventory_clerk_permssion'
);
```

        - Else:
            - Show Error Message and Return to **Repair/Recall Repository**

    - Click ***Cancel*** button - Go back to **Repair/Recall Repository**

Edit Repair Form

Abstract Code:

- Show ***Add Repair Form***, ***Edit Repair Form, View Repair From, Delete Repair Form*** in **Repair/Recall Repository**
- User clicks ***Edit Repair Form*** button and show ***Submit*** and ***Cancel*** button
- User edits Repair Form Items
- User selects one of "*Pending*", "*In Progress*" and "*Complete*" in *Repair_Status*

```
/*Assume that the inputs are correct and the permission has been validated by the application.*/
SELECT vin, start_date, end_date, repair_status, repair_description, vendor_name, repair_cost,
nhtsa_recall_compaign_number, inventory_clerk_permission
FROM `Repair` WHERE Repair.VIN = '$enteredVIN';
```

- Upon:
    - ❖ Click ***Submit*** button;
        - ➢ If nhtsa_recall_campagin_number is not Null and is not in Recall.nhtsa_recall_campagin_number:
            - ■ Show Error Message and Return to **Repair/Recall Repository**
        - ➢ Else if all items except nhtsa_recall_campagin_number are not Null:
            - ■ Write to Recall Table and go back to **Repair/Recall Repository**

```
/*Assume that the inputs are correct and the permission has been validated by the application.*/
UPDATE `Repair`
SET vin = '$enteredVin',
        start_date = '$enteredStart_date',
        end_date = '$enteredEnd_date',
        repair_status  = '$enteredRepair_status',
        repair_description  = '$enteredRepair_Description',
        vendor_name  =  '$enteredVendor_name',
        repair_cost  = '$enteredRepair_cost',
        nhtsa_recall_compaign_number  = '$enteredNHTSA_recall_campagin_Number',
        inventory_clerk_permission = '$enteredInventory_clerk_permssion';
```

        - ➢ Else:
            - ■ Show Error Message and Return to **Repair/Recall Repository**
    - ❖ Click ***Cancel*** button - Go back to **Repair/Recall Repository**

View Repair Form

Abstract Code:

- Show ***Add Repair Form***, ***Edit Repair Form, View Repair From, Delete Repair Form*** in **Repair/Recall Repository**
- User clicks ***View Repair Form*** button and show ***Search*** and ***Done*** button
- User inputs the *VIN* number in the search bar
- Upon:
    - ❖ Click ***Search*** button:
        - ➢ If *VIN* is Null or is not in Recall.vin:
            - ■ Show Error Message and Return to **Repair/Recall Repository**
        - ➢ Else:
            - ■ Find and display all repair forms for this *vin* ordered by the descending start_date

```
/*Assume that the inputs are correct and the permission has been validated by the application.*/
SELECT vin, start_date, end_date, repair_status, repair_description, vendor_name, repair_cost,
nhtsa_recall_compaign_number, inventory_clerk_permission
FROM `Repair` WHERE Repair.vin = '$enteredVIN'
ORDER BY repair_start_date DESC;
```

- ❖ Click ***Done*** button - Go back to **<u>Repair/Recall Repository</u>**

<u>Delete Repair Form</u>

Abstract Code:

- Show ***Add Repair Form***, ***Edit Repair Form, View Repair From, Delete Repair Form*** in **<u>Repair/Recall Repository</u>**
- User clicks ***Delete Repair Form*** button and show search bar with ***Search*** and ***Done*** button
- User inputs the VIN number in the search bar
- Upon:
    - ❖ Click ***Search*** button:
        - ➢ If VIN is Null or is not in Repair.vin
            - ■ Show Error Message and Return to **<u>Repair/Recall Repository</u>**
        - ➢ Else:
            - ■ Find and display all repair forms for this *VIN* ordered by the descending *Start_Date*.

```
/*Assume that the inputs are correct and the permission has been validated by the application.*/

DELETE FROM `Repair` WHERE Repair.vin = '$enteredVin' AND Repair.start_date = '$enteredStart_date';
```

- ■ Select and Delete one or more repair form for this *VIN*
- ❖ Click ***Done*** button - Go back to **<u>Repair/Recall Repository</u>**

<u>Add Vehicle Form</u>

Abstract Code:

- Show ***Add Vehicle Form, Edit Vehicle Form, View Vehicle Form, Delete Vehicle Form*** button in **<u>Vehicle Category</u>**
- User clicks ***Add Vehicle Form*** button and the system displays a new vehicle form table with ***Submit*** and ***Cancel*** button
- User enters vin, vehicle_type, vehicle_manufacturer, model_name, model_year, color, mileage, vehicle_description
- Upon:
    - ❖ Click ***Submit*** button;
        - ➢ If vin is in Vehicle.vin:
            - ■ Show Error Message and Return to **<u>Vehicle Category</u>**
        - ➢ Else if all items except vehicle_description are not Null:
            - ■ Write to Vehicle Table and go back to **<u>Vehicle Category</u>**

```
/*Assume that the inputs are correct and the permission has been validated by the application.*/

INSERT INTO `Vehicle`
VALUES(
'$enteredVin', '$enteredvehicle_mileage', '$enteredvehicle_description', '$enteredmodel_name',
'$enteredmodel_year',  '$enteredtype_name', '$enteredmanufacturer_name', '$enteredsale_price',
'$enteredVehicle_Description'
);
```

```
            ➢ Else:
                ■ Show Error Message and Return to Vehicle Category
        ❖ Click Cancel button - Go back to Vehicle Category
```

Delete Vehicle Form

Abstract Code:

```
● Show Add Vehicle Form, Edit Vehicle Form, View Vehicle Form, Delete Vehicle Form button in Vehicle Category
● User clicks Delete Vehicle Form button and the system displays a search bar with Search and Done button
● User enters vin in the search bar
● Upon:
    ❖ Click Submit button;
        ➢ If vin is Null or is not in Vehicle.vin:
            ■ Show Error Message and Return to Vehicle Category
        ➢ Else if there is Repair.vin same as vin:
            ■ Show Error Message and Return to Vehicle Category
        ➢ Else:
            ■ Display the information of this vin from Vehicle Table
            ■ Delete the vin information and Click Done button: Go back to Vehicle Category
```

```
/*Assume that the inputs are correct and the permission has been validated by the application.*/

DELETE FROM `Vehicle` WHERE Vehicle.vin = '$enteredVin';
```

```
    ❖ Click Done button - Go back to Vehicle Category
```

View Vehicle Form

Abstract Code:

```
● Show Add Vehicle Form, Edit Vehicle Form, View Vehicle Form, Delete Vehicle Form button in Vehicle Category
● User clicks View Vehicle Form button and the system displays a new empty vehicle form with Search and Done
  button
● User enters vin in the search bar
● Upon:
    ❖ Click Submit button;
        ➢ If vin is Null or is not in Vehicle.vin:
            ■ Show Error Message and Return to Vehicle Category
        ➢ Else:
            ■ Display the information of this vin from Vehicle Table
```

```
/*
This query will be identical as the query in View Vehicle Detail Form
Run view vehicle detail form task.
*/
```

        - Click ***Done*** button: Go back to **<u>Vehicle Category</u>**
  - Click ***Done*** button - Go back to **<u>Vehicle Category</u>**

<u>Edit Vehicle Form</u>

Abstract Code:

- Show ***Add Vehicle Form, Edit Vehicle Form, View Vehicle Form, Delete Vehicle Form*** button in **<u>Vehicle Category</u>**
- User clicks ***Edit Vehicle Form*** button and the system displays a search bar with ***Search*** and ***Done*** button
- User enters vin in the search bar

```
/*Assume that the inputs are correct and the permission has been validated by the application.*/

SELECT
vin, vehicle_mileage, vehicle_description,
model_name, model_year, type_name, manufacturer_name, sale_price
FROM `Vehicle` WHERE Vehicle.vin = '$enteredvin';
```

- Upon:
  - Click ***Submit*** button;
    - If vin is Null:
      - Show Error Message and Return to **<u>Vehicle Category</u>**
    - Else:
      - Display the information of this vin from Vehicle Table and User edits the information
      - Click **Done** button:
        - If all of the items except vehicle_description are not null and vehicle_description can be Null or Not Null:
          - Write to the Vehicle table and Go back to **<u>Vehicle Category</u>**

```
/*Assume that the inputs are correct and the permission has been validated by the application.*/

UPDATE `Vehicle`
SET vin = '$enteredvin',
        vehicle_mileage = '$enteredvehicle_mileage',
        vehicle_description = '$enteredvehicle_description',
        model_name = '$enteredmodel_name',
        model_year  = '$enteredmodel_year',
        type_name  = '$enteredtype_name',
        manufacturer_name = '$enteredmanufacturer_name',
        sale_price = '$enteredsale_price';
```

        - Else:
          - Show Error Message and Return to **<u>Vehicle Category</u>**
  - Click ***Done*** button - Go back to **<u>Vehicle Category</u>**

<u>Add Recall Form</u>

Abstract Code:

- Show ***Add Recall Form, Edit Recall Form, View Recall Form, Delete Recall Form*** button in **<u>Repair/Recall Repository</u>**

- User clicks ***Add Recall Form*** button and the system displays a new recall form with ***Submit*** and ***Cancel*** button
- User enters nhtsa_recall_campagin_number, recall_description, recall_manufacture information
- Upon:
    - ❖ Click ***Submit*** button;
        - ➢ If nhtsa_recall_campagin_number is in Recall.nhtsa_recall_campagin_number:
            - ■ Show Error Message and Return to **Repair/Recall Repository**
        - ➢ Else if all items are not Null:
            - ■ Write to Recall Table and go back to **Repair/Recall Repository**

```
/*Assume that the inputs are correct and the permission has been validated by the application.*/

INSERT INTO `Recall`
VALUES(
'$enteredrecall_manufacturer',
'$enteredrecall_description',
'$enteredNHTSA_recall_compaign_number'
);
```

- ➢ Else:
    - ■ Show Error Message and Return to **Repair/Recall Repository**
- ❖ Click ***Cancel*** button - Go back to **Repair/Recall Repository**

Delete Recall Form

Abstract Code:

- Show ***Add Recall Form, Edit Recall Form, View Recall Form, Delete Recall Form*** button in **Repair/Recall Repository**
- User clicks ***Delete Vehicle Form*** button and the system displays a search bar with ***Search*** and ***Done*** button
- User enters nhtsa_recall_campagin_number in the search bar
- Upon:
    - ❖ Click ***Submit*** button;
        - ➢ If nhtsa_recall_campagin_number is Null or is not in Recall.nhtsa_recall_campagin_number:
            - ■ Show Error Message and Return to **Repair/Recall Repository**
        - ➢ Else if there is Repair.nhtsa_recall_campagin_number same as nhtsa_recall_campagin_number:
            - ■ Show Error Message and Return to **Repair/Recall Repository**
        - ➢ Else:
            - ■ Display the information of this nhtsa_recall_campagin_number from Recall Table
            - ■ Select the nhtsa_recall_campagin_number information and delete. Click ***Done*** button: Go back to **Repair/Recall Repository**

```
/*Assume that the inputs are correct and the permission has been validated by the application.*/

DELETE FROM `Recall`
WHERE Recall.nhtsa_recall_compaign_number = '$enteredNHTSA_recall_compaign_number'
AND Recall.nhtsa_recall_compaign_number NOT IN (
SELECT DISTINCT(NHTSA_recall_compaign_number)
FROM Repair
WHERE Repair.nhtsa_recall_compaign_number = '$enteredNHTSA_recall_compaign_number'
);
```

        - Click ***Done*** button - Go back to **Repair/Recall Repository**

View Recall Form

Abstract Code:

- Show ***Add Recall Form, Edit Recall Form, View Recall Form, Delete Recall Form*** button in **Repair/Recall Repository**
- User clicks ***View Recall Form*** button and the system displays a search bar with ***Search*** and ***Done*** button
- User enters nhtsa_recall_campagin_number in the search bar
- Upon:
    - Click ***Submit*** button;
        - If nhtsa_recall_campagin_number is Null or is not in Recall.nhtsa_recall_campagin_number:
            - Show Error Message and Return to **Repair/Recall Repository**
        - Else:
            - Display the information of this *nhtsa_recall_Campagin_Number* from Recall Table

```
/*Assume that the inputs are correct and the permission has been validated by the application.*/

SELECT recall_manufacturer, recall_description, nhtsa_recall_compaign_number
FROM `Recall`
WHERE Recall.nhtsa_recall_compaign_number = '$enteredNHTSA_recall_compaign_number'
```

            - Click ***Done*** button: Go back to **Repair/Recall Repository**
    - Click ***Done*** button - Go back to **Repair/Recall Repository**

Edit Recall Form

Abstract Code:

- Show ***Add Vehicle Form, Edit Vehicle Form, View Vehicle Form, Delete Vehicle Form*** button in **Repair/Recall Repository**
- User clicks ***Edit Vehicle Form*** button and the system displays a search bar with ***Search*** and ***Done*** button
- User enters nhtsa_recall_campagin_number in the search bar

```
/*Assume that the inputs are correct and the permission has been validated by the application.*/

SELECT recall_manufacturer, recall_description, nhtsa_recall_compaign_number
FROM `Recall`
WHERE Recall.nhtsa_recall_compaign_number = '$enteredNHTSA_recall_compaign_number'
```

- Upon:
    - Click ***Submit*** button;
        - If nhtsa_recall_campagin_number is Null or nhtsa_recall_campagin_number is not in Recall.nhtsa_recall_campagin_number:
            - Show Error Message and Return to **Repair/Recall Repository**
        - Else:
            - Display the information of this nhtsa_recall_campagin_number from Recall Table and User edits the information
            - Click ***Done*** button:

            - If all of the items are not Null:
                - Write to the Recall table and Go back to **Repair/Recall Repository**

```
/*If there is no conflict for the nhtsa_recall_campagin_number and assume that the inputs are correct and the
permission has been validated by the application*/

UPDATE `Recall`
SET recall_manufacturer = '$enteredrecall_manufacturer',
            recall_description = '$enteredrecall_description',
            nhtsa_recall_compaign_number = 'enteredNHTSA_recall_compaign_number'
```

            - Else:
                - Show Error Message and Return to **Repair/Recall Repository**
    - Click ***Done*** button - Go back to **Repair/Recall Repository**

View Seller History Report

Abstract Code:

- Show ***View Sale History Report*** button in **Employee Search** Page.
- Show ***Done*** button.
- For each driver_ license_ number and tax_ identification_ number in Customer table:
    - Calculate the total number of vehicles sold to Burdell's in the Sell_transcation Table.
    - Calculate the average price for the vehicles each customer has sold to Burdell's in the Sell_transcation Table.
    - Calculate the average number of repairs per vehicle in the Repair Table.
- Find and display the name of the customers (either first or last name or the company name) from the Customer table in descending order of the total number of vehicles the respective customer have sold to Burdell's.
- Display the average price for the vehicles and the average number of repairs per vehicle for each customer.
- Highlight the customers that show an average of five or more repairs per vehicle sold to Burdell's with a red background.
- Upon:
    - Click ***Done*** button - Go back to **Employee Search** Page.

```
SELECT Buy.customer_id, AVG(repair_count) AS average_repair,
COUNT(Buy.vin) AS number_of_sold_vehicles, AVG(Buy.purchase_price) AS avg_price, name
FROM Buy JOIN Repair on Buy.vin = Repair.vin
JOIN (SELECT CONCAT(ISNULL(Person.last_name, ''), ISNULL(Business.business_name, '')) AS name,
Buy.customer_id AS customerid1 FROM Buy
   LEFT JOIN person ON Buy.customer_id= person.customer_id
   LEFT JOIN business ON Buy.customer_id = business.customer_id
 WHERE Buy.id = '$enteredId') ON Buy.customer_id = customerid1
JOIN (SELECT Buy.customer_id AS customerid, Repair.vin AS repairvin, COUNT(Repair.vin) AS repair_count
 FROM Repair
  JOIN Buy ON Buy.vin = Repair.vin
GROUP BY Repair.vin)
ON Buy.vin = repairvin
GROUP BY customer_id
ORDER BY number_of_sold_vehicles DESC, avg_price ASC;
```

View Inventory Age Report

Abstract Code:

- Show ***View Inventory Age Report*** button in **<u>Employee Search</u>** Page.
- Show ***Done*** button.
- For each type of vehicle (identified with vehicle_type in Vehicle table):
    - ❖ Calculate the minimum, average, and maximum age (decided by model_year in Vehicle table) of unsold vehicles in inventory, in days.
- Find and display vehicle_type in Vehicle table in alphabetical order.
- Display the minimum, average, and maximum age of unsold vehicles in inventory, in days, for each vehicle_type. If a vehicle type has no unsold units, the report should display "N/A" for that vehicle_type.
- Upon:
    - ❖ Click ***Done*** button - Go back to **<u>Employee Search</u>** Page.

```
WITH tbl AS(
SELECT
        Vehicle.type_name,
        AVG(DATE_PART('DAY', CURRENT_DATE - Buy.purchase_date)) AS avg_inventory_age,
        MAX(DATE_PART('DAY', CURRENT_DATE - Buy.purchase_date)) AS max_inventory_age,
        MIN(DATE_PART('DAY', CURRENT_DATE - Buy.purchase_date)) AS min_inventory_age
FROM Vehicle
LEFT OUTER JOIN Buy
ON Vehicle.vin = Buy.vin
RIGHT JOIN VehicleType
ON Vehicle.type_name = VehicleType.type_name
WHERE Vehicle.vin NOT IN (SELECT Sell.vin FROM Sell)
GROUP BY Vehicle.type_name
)

SELECT VehicleType.type_name, avg_inventory_age, max_inventory_age, min_inventory_age
FROM VehicleType
LEFT JOIN tbl
ON VehicleType.type_name = tbl.type_name
ORDER BY tbl.type_name;
```

<u>View Average Time in Inventory Report</u>

Abstract Code:

- Show ***View Inventory Age Report*** button in **<u>Employee Search</u>** Page**.**
- Show ***Done*** button.
- For each type of vehicle (identified with Vehicle_type in Vehicle table):
    - ❖ Calculate the average time in the inventory of unsold vehicles, in days. Using purchase_date from the Buy_transcation Table and sale_date from the Sell_transcation Table.
- Find and display all vehicle_type in Vehicle table in alphabetical order.
- Display the average time in the inventory of unsold vehicles, in days, for each vehicle_type. If a vehicle_type has no unsold units, the report should display "N/A" for that vehicle type.
- Upon:
    - ❖ Click ***Done*** button - Go back to in **<u>Employee Search</u>** Page**.**

```
/*Assume that the inputs are correct and the permission has been validated by the application.*/

SELECT Vehicle.type_name AS type_name, AVE(tbl.dateDiff) AS average_time_in_inventory
FROM (
        SELECT Sell.vin AS vin, DATEDIFF(DAY, Buy.purchase_date, Sell.sale_date) AS dateDiff
FROM Sell
LEFT JOIN Buy
ON Sell.vin = Buy.vin
WHERE Buy.purchase_date IS NOT NULL AND Sell.sale_date IS NOT NULL;
)tbl
LEFT JOIN Vehicle
ON Vehicle.vin = tbl.vin
GROUP BY Vehicle.type_name
ORDER BY Vehicle.type_name;
```

View Price per Condition Report

Abstract Code:

- Show ***View Price per Condition Report*** button in **Employee Search** Page.
- Show ***Done*** button.
- For each vehicle_type in the Vehicle table and for each purchase_condition in the Buy_transcation Table:
    - ❖ Calculate the average price of vehicles purchased. If a vehicle type or condition has never been purchased, the report should display "$0" for that result.
- Display the average price of vehicles per vehicle_type and per purchase_condition in a pivot table.
- Upon:
    - ❖ Click ***Done*** button - Go back to **Employee Search** Page.

```
DROP TABLE IF EXISTS Vehicle_Sales_Table;

CREATE TABLE Vehicle_Sales_Table
AS (
   SELECT Vehicle.vin AS VIN, Vehicle.type_name AS Vehicle_Type,
     COALESCE(Buy.Purchase_price, 0) AS Purchase_price, Buy.purchase_condition AS Vehicle_Condition
   FROM Vehicle
   JOIN Buy
   ON Buy.vin = Vehicle.vin
);

SELECT Vehicle_Type,
 COALESCE(ROUND(AVG(
            CASE
            WHEN Vehicle_Condition = 'Excellent'
            THEN ROUND(Purchase_price,0)
            ELSE null
            END
 ),2),0.00) As Excellent,
 COALESCE(ROUND(AVG(
            CASE
   WHEN Vehicle_Condition = 'Very Good'
   THEN Purchase_price
   ELSE null
   END
 ),2),0.00) As very_good,
 COALESCE(ROUND(AVG(
            CASE
  WHEN Vehicle_Condition = 'Good'
  THEN Purchase_price
  ELSE null
 END
 ),2),0.00) As good,
 COALESCE(ROUND(AVG(
            CASE
  WHEN Vehicle_Condition = 'Fair'
  THEN Purchase_price
  ELSE null
            END
 ),2),0.00) As fair
FROM Vehicle_Sales_Table
GROUP BY Vehicle_Type
ORDER BY Vehicle_Type;
```

View Repair Statistics Report

Abstract Code:

- Show ***View Repair Statistics Report*** button in **Employee Search** Page.
- Show ***Done*** button.
- For each vendor_name in Repair table:
  - ❖ Calculate the number of repairs by that vendor.
  - ❖ Calculate the total cost spent on completed repairs from that vendor using repair_cost in the Repair table.
  - ❖ Calculate the average number of repairs per vehicle (identified by vin from the Vehicle Table) from that vendor using repair_cost in the Repair table.
- Calculate the average length of repair time (in days) from that vendor using start_date and end_date in the Repair table
- Find and display all vendor_name in Repair table in alphabetical order.
- Display the number of repairs, the total cost spent on completed repairs, the average number of repairs per vehicle, and the average repair time (in days) for each vendor.
- Upon:
  - ❖ Click ***Done*** button - Go back to **Employee Search** Page.

```
SELECT
                Repair.vendor_name,
                COUNT(Repair.vendor_name) AS num_of_repairs,
                SUM(Repair.repair_cost) AS total_repair_cost,
                COUNT(Repair.vin)/COUNT(Repair.vendor_name) AS avg_repair_per_vehicle,
                AVG(DATE_PART('DAY',Repair.end_date - Repair.start_date)) AS avg_time_per_repair
FROM Repair
WHERE Repair.repair_status = 'complete'
GROUP BY Repair.vendor_name
ORDER BY Repair.vendor_name;
```

View Monthly Sales Report

Abstract Code:

- Show ***View Monthly Sales Report*** button in **Employee Search** Page.
- Show ***Done*** button in both the mother task and subtask.
- Find and display sale_date in the Sell Table in descending order.
- For each calendar year and month in sale_date:
    - ❖ Calculate and display the number of vehicles sold (i.e. the number of items) from the Sell Table. If a year or month have 0 items in Sell, do not show that year or month.
    - ❖ Calculate and display the total sales income using sale_price from the Sell Table.
    - ❖ Calculate and display the net sales income using sale_price from the Sell Table, purchase_price from the Buy Table, and repair_cost from the Repair Table.

```
--Yearly sale summary page
SELECT
            COUNT(Sell.vin) AS Num_of_vehicle_sold,
            SUM(Vehicle.sale_price) AS total_sale_income,
            (SUM(Vehicle.sale_price)- SUM(Buy.purchase_price) - SUM(Repair.repair_cost)) AS net_income,
            LEFT(Sell.sale_date::text, 4) AS Sale_year
FROM Sell
JOIN Buy
ON Sell.vin = Buy.vin
JOIN Repair
ON Sell.vin = Repair.vin
JOIN Vehicle
ON Vehicle.vin = Repair.vin
GROUP BY Sale_year
ORDER BY Sale_year DESC;
```

```
--Monthly sale summary page
SELECT
            COUNT(Sell.vin) AS Num_of_vehicle_sold,
            SUM(Vehicle.sale_price) AS total_sale_income,
            (SUM(Vehicle.sale_price)- SUM(Buy.purchase_price) - SUM(Repair.repair_cost)) AS net_income,
            LEFT(Sell.sale_date::text, 7) AS Sale_month
FROM Sell
JOIN Buy
ON Sell.vin = Buy.vin
JOIN Repair
ON Sell.vin = Repair.vin
JOIN Vehicle
ON Vehicle.vin = Repair.vin
GROUP BY Sale_month
ORDER BY Sale_month DESC;
```

- For each year or month result, create a clickable link to a drilldown report, find all items from the Salesperson Table as a subitem of the User Table.
- For each item from the Salesperson Table:
  - Calculate the number of vehicles sold (i.e. the number of items) from the Sell_transcation Table.
  - Calculate the total sales using sale_price from the Sell_transcation Table.
- Find and display all items from the Salesperson Table, sorted by total vehicles in descending order, followed by total sales in descending order. Display these items' login_first_name and login_last_name from the User Table.
- Upon:
  - Click ***Done*** button - Go back to **Employee Search** Page**.**
  - Click each individual ***year*** *or* ***month*** link - Go to the drill down report for that respective year or month.

```
--Year sale drill down report
--Shown upon: user clicks a given '$Sale_year' from the yearly sale summary page

SELECT
        MAX(Users.login_first_name) AS top_seller_first_name,
        MAX(Users.login_last_name) AS top_seller_last_name,
        COUNT(Sell.vin) AS num_vehicle_sold,
        SUM(Vehicle.sale_price) AS total_sales
FROM Sell
JOIN Vehicle
ON Vehicle.vin = Sell.vin
JOIN Salesperson
ON Sell.salesperson_permission = Salesperson.salesperson_permission
JOIN Users
ON Salesperson.username = Users.username
WHERE LEFT(Sell.sale_date::text, 4) = '$Sale_year'
GROUP BY Salesperson.username
ORDER BY
num_vehicle_sold DESC,
total_sales DESC
LIMIT 1;
```

```
--Month sale drill down report
--Shown upon: user clicks a given '$Sale_month' from the yearly sale summary page

SELECT
        MAX(Users.login_first_name) AS top_seller_first_name,
        MAX(Users.login_last_name) AS top_seller_last_name,
        COUNT(Sell.vin) AS num_vehicle_sold,
        SUM(Vehicle.sale_price) AS total_sales
FROM Sell
JOIN Vehicle
ON Vehicle.vin = Sell.vin
```

```
JOIN Salesperson
ON Sell.salesperson_permission = Salesperson.salesperson_permission
JOIN Users
ON Salesperson.username = Users.username
WHERE LEFT(Sell.sale_date::text, 7) = '$Sale_month'
GROUP BY Salesperson.username
ORDER BY
num_vehicle_sold DESC,
total_sales DESC
LIMIT 1;
```