Phase 2 Report

2021Fall CS6400 Team011

Table of Contents

1 Task: Search Vehicle:

- Display number of Unsold Vehicle.

```
SELECT count (VIN)
FROM Vehicle
WHERE VIN NOT IN
        (SELECT VIN
        FROM SalesTransaction);
```

- If Click ***Search Dropdown*** – Display dropdown selections to Search by Vehicle_Type, Manufacturer, Color or Year;
- If Click ***Vehicle Type*** – Display dropdown selections to Vehicle Types.
    - User makes selection
- If Click ***Manufacturer*** – Display dropdown selections to Manufacturer.

```
SELECT name
FROM Manufacturer;
```

-
    - User makes selection
- If Click ***Color*** – Display dropdown selections to Color.

```
SELECT name
FROM Color;
```

-
    - User makes selection
- If Click ***Year*** – Display dropdown selections to Years.
    - User makes selection
- If Click Listed Price – Display a listed price range input.
    - User makes selection
- If User is logged in: Display option to search by VIN.
    - User makes selection
- User enter Search field (Optional).

- Click ***Search Vehicle***
    - If VIN is selected in Search Criteria: ***Search Vehicle*** by VIN using Search field and Display Results.

```
WITH VehicleCTE
```

```
AS(
SELECT v.VIN, description, model_year, model_name,
invoice_price,manufacturer_name , inv_writer_user_name, add_date,  'Car' as
vehicle_type
FROM Vehicle v INNER JOIN Car c ON  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name,
invoice_price,manufacturer_name , inv_writer_user_name, add_date,
'Convertible'
FROM Vehicle v INNER JOIN Convertible c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name,
invoice_price,manufacturer_name , inv_writer_user_name, add_date, 'Truck'
FROM Vehicle v INNER JOIN Truck t on  v.VIN = t.VIN
UNION
SELECT v.VIN, description, model_year, model_name,
invoice_price,manufacturer_name , inv_writer_user_name, add_date, 'Van'
FROM Vehicle v INNER JOIN Van  on  v.VIN = Van.VIN
UNION
SELECT v.VIN, description, model_year, model_name,
invoice_price,manufacturer_name , inv_writer_user_name, add_date, 'SUV'
FROM Vehicle v INNER JOIN SUV s on  v.VIN = s.VIN;
)
SELECT VIN, vehicle_type, model_year, manufacturer_name, model_name,
description, invoice_price*125%, STRING_AGG(color, ',')
FROM VehicleCTE NATURAL JOIN VehicleHasColor
WHERE VehicleCTE.VIN = '$VIN'
```

- Else: Search Vehicle by selected criteria.
    - If Search field is entered, ***Search Result*** by List price or keyword using input and Display Results (The result gives the vehicles that meet all of the searching filter).

```
WITH VehicleCTE
AS(
SELECT v.VIN, description, model_year, model_name,
invoice_price,manufacturer_name , inv_writer_user_name, add_date,  'Car' as
vehicle_type
FROM Vehicle v INNER JOIN Car c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name,
invoice_price,manufacturer_name , inv_writer_user_name, add_date,
'Convertible'
FROM Vehicle v INNER JOIN Convertible c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name,
invoice_price,manufacturer_name , inv_writer_user_name, add_date, 'Truck'
FROM Vehicle v INNER JOIN Truck t on  v.VIN = t.VIN
UNION
SELECT v.VIN, description, model_year, model_name,
invoice_price,manufacturer_name , inv_writer_user_name, add_date, 'Van'
FROM Vehicle v INNER JOIN Van  on  v.VIN = Van.VIN
UNION
SELECT v.VIN, description, model_year, model_name,
invoice_price,manufacturer_name , inv_writer_user_name, add_date, 'SUV'
FROM Vehicle v INNER JOIN SUV s  on  v.VIN = s.VIN
)
SELECT VIN, vehicle_type, model_year, manufacturer_name, model_name,
description, invoice_price*125%, STRING_AGG(color, ',') FROM VehicleCTE
NATURAL JOIN VehicleHasColor
WHERE vehicle_type = '$vehicle_type' AND model_year = '$model_year' AND
manufacturer_name = '$manufacturer_name' AND (model_year LIKE
'%$KEYWORD%' OR model_name LIKE '%$KEYWORD%' OR description
LIKE '%$KEYWORD%') AND invoice_price*1.25 BETWEEN
'$min_listed_price' AND '$max_listed_price';
GROUP BY VehicleCTE.VIN
ORDER BY VehicleCTE.VIN ASC;
```

- If Vehicle not found: display “Sorry, it looks like we don't have that in stock!”
- Else: for each matching Vehicle display VIN, Type, Year, Manufacturer, Model, Color and List price.

- Else:

```
WITH VehicleCTE
AS(
SELECT v.VIN, description, model_year, model_name,
invoice_price,manufacturer_name , inv_writer_user_name, add_date,  'Car' as
vehicle_type
FROM Vehicle v INNER JOIN Car c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name,
invoice_price,manufacturer_name , inv_writer_user_name, add_date,
'Convertible'
FROM Vehicle v INNER JOIN Convertible c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name,
invoice_price,manufacturer_name , inv_writer_user_name, add_date, 'Truck'
FROM Vehicle v INNER JOIN Truck t on  v.VIN = t.VIN
UNION
SELECT v.VIN, description, model_year, model_name,
invoice_price,manufacturer_name , inv_writer_user_name, add_date, 'Van'
FROM Vehicle v INNER JOIN Van  on  v.VIN = Van.VIN
UNION
SELECT v.VIN, description, model_year, model_name,
invoice_price,manufacturer_name , inv_writer_user_name, add_date, 'SUV'
FROM Vehicle v INNER JOIN SUV s  on  v.VIN = s.VIN
)
SELECT VIN, vehicle_type, model_year, manufacturer_name, model_name,
description, invoice_price*125%, STRING_AGG(color, ',') FROM VehicleCTE
NATURAL JOIN VehicleHasColor
WHERE vehicle_type = '$vehicle_type' AND model_year = '$model_year' AND
manufacturer_name = '$manufacturer_name' AND invoice_price*1.25
BETWEEN  '$min_listed_price' AND '$max_listed_price';
GROUP BY VehicleCTE.VIN
ORDER BY VehicleCTE.VIN ASC;
```

- If Vehicle not found: display “Sorry, it looks like we don't have that in stock!”
- Else: for each matching Vehicle display VIN, Type, Year, Manufacturer, Model, Color and List price.

2 Task: log In

- Upon: Click ***Login*** - Display **<u>Login Form</u>**.
- User enters *username*, *password* input fields.
- If data validation is successful for username and password input fields:
- Upon:
    - Click ***Enter***:

```
SELET password
FROM Users
WHERE user_name = '$user_name';
```

- If Users record is not found or Users.password!='$Password': Go back to **<u>Login Form</u>**, Display error message.
- Else: go to ***Search Vehicle*** and display functions according to Users. permission.

```
WITH UserpermissionCTE
AS
(
SELECT user_name, ' ServiceWriter ' as Userrole FROM ServiceWriter
UNION
SELECT user_name, ' SalesPeople ' FROM SalesPeople
UNION
SELECT user_name, ' Owner ' FROM Owner
UNION
SELECT user_name, ' Manager ' FROM Manager
UNION
SELECT user_name, ' InventoryClerk ' FROM InventoryClerk
)
SELECT Userrole
FROM UerpermissionCTE
WHERE user_name = '$user_name';
```

  - if only one record is found, then we get the user's permission type
  - if 5 records are found, then the user is the Owner

- Else: display **Login Form**, with error message.

3 Task: Lookup Customer

- User enters Customer input fields.
  - If data validation is successful for input fields:
  - Upon:
    - Click ***Search***:
      - If Individual is selected:

```
SELET Individual.customerID
FROM  Individual
WHERE driver_license = '$driver_license';
```

        - if Customer is found: return customerID
        - else: ***Add Customer***.
      - If Business is selected:

```
SELET Business.customerID
FROM  Business
WHERE tax_ID = '$tax_ID';
```

        - if Customer is found: return customerID
        - else: ***Add Customer***.

  - Else: Display error message.

4 Task: Add Customer

- Upon:
- Click ***Select***:
    - If ***Individual*** is selected: Display **Individual Info Form**.
    - Else If ***Business*** is selected: Display **Business Info Form**.
    - Else: Display an error message.

- User Enters input fields of displayed Info Form.
    - If data validation is successful for input fields:
    - Upon:
        - Click ***Confirm***: ***Add Customer***.
        - If Individual is selected:

```
INSERT INTO Customer (email, phone_number, add_by_user_name,
street_address, city, state, postal_code)
VALUES ('$email', '$phone_number', '$add_by_user_name',
'$street_address', '$city', '$state', '$postal_code');

INSERT INTO Individual (driver_license, first_name, last_name,
customerID)
SELECT  '$diver_license', '$first_name', '$last_name', customerID
FROM Customer
WHERE phone_number = '$phone_number' AND add_by_user_name =
'$add_by_user_name' AND street_address = '$street_address' AND city =
'$city' AND state = '$state' AND postal_code = '$postal_code';
```

        - If Business is selected:

```
INSERT INTO Customer (email, phone_number, add_by_user_name,
street_address, city, state, postal_code)
VALUES ('$email', '$phone_number', '$add_by_user_name',
'$street_address', '$city', '$state', '$postal_code');

INSERT INTO Business (tax_ID, contact_title, contact_name,
business_name, customerID)
SELECT  '$tax_ID', '$contact_title', '$contact_name', '$business_name',
Customer.customerID
FROM Customer
WHERE phone_number = '$phone_number' AND add_by_user_name =
'$add_by_user_name' AND street_address = '$street_address' AND city =
'$city' AND state = '$state' AND postal_code = '$postal_code';
```

    - Else: Display an error message.

5 Task: View Vehicle Detail

- Find the Vehicle using the $VIN

```
WITH VehicleCTE
AS(
SELECT v.VIN, description, model_year, model_name,
invoice_price,manufacturer_name , inv_writer_user_name, add_date,  'Car' as
vehicle_type
FROM Vehicle v INNER JOIN Car c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name,
invoice_price,manufacturer_name , inv_writer_user_name, add_date, 'Convertible'
FROM Vehicle v INNER JOIN Convertible c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name,
invoice_price,manufacturer_name , inv_writer_user_name, add_date, 'Truck'
FROM Vehicle v INNER JOIN Truck t on  v.VIN = t.VIN
UNION
SELECT v.VIN, description, model_year, model_name,
invoice_price,manufacturer_name , inv_writer_user_name, add_date, 'Van'
FROM Vehicle v INNER JOIN Van  on  v.VIN = Van.VIN
UNION
SELECT v.VIN, description, model_year, model_name,
invoice_price,manufacturer_name , inv_writer_user_name, add_date, 'SUV'
FROM Vehicle v INNER JOIN SUV s  on  v.VIN = s.VIN
)
SELECT VIN, description, model_year, model_name, invoice_price,
manufacturer_name, inv_writer_user_name, add_date, vehicle_type,
STRING_AGG(color, ',') AS vehicle_color
FROM VehicleCTE NATURAL JOIN VehicleHasColor
WHERE VIN = '$VIN' ;
```

- Display VIN, manufacturer_name, model_year, model_name, vehicle_type, list_price(=invoice_price*1.25), description, vehicle_color

- Display Vehicle Type corresponding attributes(like roof type and back seat count for convertible)

If vehicle_type = 'Car'

```
SELECT number_of_doors
FROM Car
WHERE Car.VIN='$VIN';
```

Else if vehicle_type = 'Convertible'

```
SELECT roof_type, back_seat_count
FROM Convertible
WHERE Convertible.VIN= '$VIN';
```

Else if vehicle_type= 'Truck'

```
SELECT cargo_capacity, cargo_cover_type, num_of_rear_axies
FROM Truck
WHERE Truck.VIN = '$VIN' ;
```

Else if vehicle_type='Van'

```
SELECT has_driver_side_back_door
FROM Van
WHERE Van.VIN= '$VIN' ;
```

Else if vehicle_type='SUV'

```
SELECT number_of_cupholder, drivetrain_type
FROM SUV
WHERE SUV.VIN= '$VIN';
```

- If user role is InventoryClerk or Manager or Owner :
    - Display invoice_price
    - If user role is Manager or Owner:

```
SELECT T.customerID, email, phone_number, street_address, city, state, postal_code,
sold_price, sales_writer_user_name, sold_price, transaction_date, S.first_name,
S.last_name AS SalesInfo
FROM SalesTransaction AS T, Customer, Users NATRUAL JOIN SalesPeople AS S
WHERE Transaction.VIN = '$VIN' AND Transaction.CustomerID =
Customer.customerID;
```

- If the SalesInfo is not NULL(means the car has been sold):
    - Display email, phone_number, street_address, city, state, postal_code, sold_price, transaction_date, S.first_name, S.last_name. Store customerID as @customerID

```
SELECT driver_license, first_name, last_name AS CustomerResult
FROM Individual AS I
WHERE I.customerID= @customerID;
```

        - If CustomerResult is not NULL (customer is individual rather than business)
            - Display I.first_name, I.last_name
        - Else

```
SELECT contact_title, contact_name, business_name AS CustomerResult
FROM Business AS B
WHERE B.customerID=@customerID;
```

            - Display contact_title, contact_name, business_name

--if this vehicle has been repaired, show repair history

```
SELECT C.business_name + C.first_name + SPACE(1) + C.last_name AS customer_name,
repair_starter, start_date, complete_date, labor_charge, parts_cost AS Repair_For_This_Vehicle
FROM (Customer LEFT OUTER JOIN Individual LEFT OUTER JOIN Business) AS C,
(RepairDetails NATRUAL JOIN (
        SELECT VIN, start_date, SUM (quantity*price) AS parts_cost
        FROM Part
        GROUP BY (VIN, start_date)
        ))AS R
WHERE R.customerID = C.customerID AND R.VIN = '$VIN' AND
RepairDetails.complete_date IS NOT NULL
ORDER BY RepairDetails.start_date DESC, RepairDetails.complete_date ASC;
```

    - If the Repair_For_This_Vehicle is not NULL:
        - Display list of Repair_Details (customer_name, repair_starter, start_date, complete_date, labor_charge, parts_cost, total cost=labor_charge+parts_cost)

- If user role is SalesPeople:
    - Display ***Sell this Vehicle*** button
- If ***Back to Search*** Button Clicked, go to **<u>Access Form</u>**
- If ***Sell this Vehicle*** Button Clicked, go to **<u>Sales Order Form</u>**

6 Task: Add Vehicle

- Populate Manufacturer dropdowns

```
SELECT Manufacturer.name
FROM Manufacturer;
```

Populate Vehicle_Type related attributes input space

- While no buttons are pushed, do nothing.
- When a button is pushed, then do the following:
    - If ***Back to Search***, go to **<u>Access </u>**Form
    - If ***Add Vehicle***:
        - {If there is blank fields, populate error message,
        - If there is entering that is not valid, populate error message,
        - Else **Add Vehicle** with user input *VIN*($VIN), *manufactuer*($Manufactuer), *model_year*($Model_year), *model_type*($Model_type), *vehicle_type*($Vehicle_type), *vehicle_colors*($Vehicle_colors), *invoice_price*($Invoice_price), *description*($Description), *add_date*($Add_date), Then go to **<u>Vehicle Detail Form</u>.**}

    (current_user_name as '$Current_user_name', get current_date as ($Current_date)

```
INSERT INTO Vehicle
VALUES ('$VIN', '$Description', '$Model_year', '$Model_name', '$Invoice_price',
'$Manufacturer', '$Current_user_name', '$Current_date');
```

If $Vehicle_type='Car'

```
INSERT INTO Car
VALUES ('$VIN', '$number_of_doors');
```

If $Vehicle_type='Convertible'

```
INSERT INTO Convertible
VALUES ('$VIN', '$roof_type', '$back_sear_count');
```

If $Vehicle_type='Truck'

```
INSERT INTO Truck
VALUES ('$VIN', '$cargo_capacity', '$cargo_cover_type', '$num_of_rear_axies');
```

If $Vehicle_type='Van'

```
INSERT INTO Van
VALUES ('$VIN', '$has_driver_side_back_door');
```

If $Vehicle_type='SUV'

```
INSERT INTO SUV
VALUES('$VIN', '$number_of_cupholder', '$drivetrain_type');
```

7 Task: Create/Confirm Sale

- Display VIN for current Vehicle, display current date as Sold date
- While no buttons are pushed, do nothing.
- When a button is pushed, then do the following:
    - if ***Look up Customer***:
        - **Go to <u>Look up Customer</u>** Form

        - Display Customer name found by the customerID retured by **Look up Customer** Form

```
SELECT C.business_name + C.first_name + SPACE(1) + C.last_name + AS
customer_name,
FROM ( Customer LEFT OUTER JOIN Individual LEFT OUTER JOIN Business)AS C
WHERE C.customerID = '$CustomerID_Returned';
```

    - if ***Cancel & Back to Vehicle***, go back to **Vehicle details** for current Vehicle
    - If ***Confirm Order***:
        - If there is blank fields(no customer, no price), populate error message,
        - If there is entering is not valid(sold price less than 95% invoice price & not owner) populate error message,
        - Else insert transaction information and go back to **Privileged Access Common Form**

```
INSERT INTO Transaction
VALUES ('$VIN', '$CustomerID', '$Current_user_name', '$Sold_price', '$Current_date');
```

8 Task: Start New Repair Order

- Search by VIN in SalesTransaction table.

```
WITH VehicleCTE
AS(
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date,  'Car' as vehicle_type
FROM Vehicle v INNER JOIN Car c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Convertible'
FROM Vehicle v INNER JOIN Convertible c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Truck'
FROM Vehicle v INNER JOIN Truck t on  v.VIN = t.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Van'
FROM Vehicle v INNER JOIN Van  on  v.VIN = Van.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'SUV'
FROM Vehicle v INNER JOIN SUV s  on  v.VIN = s.VIN
)
```

```
SELECT VIN, vehicle_type, model_year, manufacturer_name, model_name,
STRING_AGG(color, ',') AS vehicle_color
FROM VehicleCTE NATURAL JOIN VehicleHasColor
WHERE VehicleCTE.VIN = '$VIN';
```

- If VIN is NULL (not found),
  - Display the Error message "Vehicle Not Found or Not Sold".
- If VIN is not NULL, display vehicle details {VIN, vehicle type, Model Year, Model Name, Manufacturer, and color(S)}

```
SELECT R.VIN, R.labor_charge, R.description
FROM RepairDetails AS R
WHERE R.VIN = '$VIN' AND start_date IS NOT NULL AND complete_date IS NULL;
```

  - If no repair are open for vehicle (R.VIN is NULL), display the empyty repair form with place to enter odometer reading.
    - When ***Look Up Customer*** Button is Clicked
      - Display Customer_name found by the customerID retured by **Look up Customer** Form

```
SELECT C.business_name + C.first_name + SPACE(1) + C.last_name AS customer_name,
FROM ( Customer LEFT OUTER JOIN Individual LEFT OUTER JOIN Business ) AS C
WHERE C.customerID='$customerID_returned';
```

    - When the ***Create Repair*** button is clicked,
      - If Odometer_reading is (Null or not a valid number) or Customer_name is Null, Display error message
      - else start new repair and go to **Update&Complete Repair** Form

```
INSERT INTO Repair_Details
VALUES ('$VIN', '$current_date', 'cusotmerID', '$current_user_name,
'$odometer_value', '$description', '$labor_charges', NULL);
```

9 Task: Update& Complete Repair

- Search by VIN in SalesTransaction table.

```
WITH VehicleCTE
AS(
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date,  'Car' as vehicle_type
FROM Vehicle v INNER JOIN Car c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Convertible'
FROM Vehicle v INNER JOIN Convertible c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Truck'
FROM Vehicle v INNER JOIN Truck t on  v.VIN = t.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Van'
FROM Vehicle v INNER JOIN Van  on  v.VIN = Van.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'SUV'
FROM Vehicle v INNER JOIN SUV s  on  v.VIN = s.VIN
)
SELECT VIN, vehicle_type, model_year, manufacturer_name, model_name,
STRING_AGG(color, ',') AS vehicle_color
FROM VehicleCTE NATURAL JOIN VehicleHasColor
WHERE VehicleCTE.VIN = '$VIN'；
```

- If VIN is NULL (not found),
    - Display the Error message "Vehicle Not Found or Not Sold".
- If VIN is not NULL, display vehicle details {VIN, vehicle type, Model Year, Model Name, Manufacturer, and color(S)}

```
SELECT R.VIN, labor_charge, description, start_date
FROM RepairDetails AS R
WHERE R.VIN = '$VIN' AND start_date IS NOT NULL AND complete_date IS NULL;
```

- If there is repair open for vehicle (R.VIN is NOT NULL), @start_date is stored in order to be used in querys below
    - display labor charge and description for update
        - if ***Update Labor Charge*** is clicked
          Validation: if not owner, labor charge cannot be less than previous value(we checked the logged in user type from the **LogIn Form)**

```
UPDATE RepairDetail
SET labor_charges = '$Labor_Charges'
WHERE VIN = '$VIN' AND start_date = @start_date；
```

- if ***Update Description*** is clicked

```
UPDATE RepairDetail
SET description = '$Description'
WHERE VIN = '$VIN' AND start_date = @start_date；
```

- display parts and places to add part.

```
SELECT part_number, quantity, price, vendor
FROM Part
WHERE VIN='$VIN', start_date=@start_date;
```

- if ***Add Part*** Button is clicked, validate if the input format is correct, then check if the part number already exists in the repair

```
SELECT part_number, quantity, price, vendor
FROM Part
WHERE VIN='$VIN', start_date = @start_date, part_number = '$part_number';
```

  - if the part is not already exist in this repair(quantity=NULL), insert part, refresh parts list

```
INSERT INTO Part
VALUES ('$VIN', @start_date, '$part_number', '$quantity', '$price', '$vendor');
```

```
SELECT part_number, quantity, price, vendor
FROM Part
WHERE VIN='$VIN', start_date=@start_date;
```

  - if the part already exist in this repair, check if $vendor, $price are the same,
    - if not, display error
    - else, upate quantity, refresh parts list

```
UPDATE Part
SET quantity=quantity+'$quantity'
WHERE VIN='$VIN', start_date=@start_date, part_number='$part_number';
```

```
SELECT part_number, quantity, price, vendor
FROM Part
WHERE VIN='$VIN', start_date=@start_date;
```

- When the ***Complete Repair*** button is click,

```
UPDATE RepairDetail
SET complete_date = '$current_date'
WHERE VIN = '$VIN' AND start_date = @start_date;
```

10 Task: Report Form

- User ***login*** from the **Login Form**
- If the user's type is Manager or Owner(we checked user's type from **Login Form)**
  Display **View Report** button on the **Privileged Access Common Form**

- If User click view report dropdown list for ***Report Request***,
  - Display Dropdown list of selections that include:
    1. Sales by Color
    2. Sales by Type
    3. Sales by Manufacturer
    4. Gross Customer Income
    5. Repairs by Manufacturers/Type/Model
    6. Below Cost Sales
    7. Average Time in Inventory
    8. Parts Statics
    9. Monthly Sales
- User click on one of the selections
  - Display the selected report

10.1 View Sales by Color Report

- Sales by Color Report
  - No. of vehicles sold in selected date range
  - if a vehicle has multiple colors, it should be classified as “multiple” and not included in the count of vehicles that have only one color.
  - Showing 0 for color without any sales

```
SELECT thirtydays.color, thirtydays.totalsales as thirtyDaysSales,  lastyear.totalsales as lastYearSales, overall.totalsales  overallSales
FROM (
SELECT Color.name as color, ISNULL(temp.sales, 0) AS totalsales
FROM Color
LEFT JOIN (
SELECT DISTINCT vc.color, COUNT(DISTINCT s.VIN) AS sales
FROM (
SELECT VIN, COUNT(1) count
FROM  VehicleHasColor GROUP BY VIN) c
LEFT JOIN VehicleHasColor vc ON vc.VIN = c.VIN
LEFT JOIN SalesTransaction s ON vc.VIN = s.VIN
WHERE s. transaction_date > (
SELECT dateadd (DAY,-30, MAX(transaction_date) )
FROM SalesTransaction)
AND c. count = 1
GROUP BY vc.color
) temp ON temp.color = Color.name
UNION
```

```
SELECT DISTINCT 'multiple', COUNT(DISTINCT s.VIN) AS sales
FROM (SELECT VIN, COUNT(1) count FROM VehicleHasColor GROUP BY VIN) c
LEFT JOIN VehicleHasColor vc ON vc.VIN = c.VIN
LEFT JOIN SalesTransaction s ON vc.VIN = s.VIN
WHERE s. transaction_date > (SELECT dateadd(DAY,-30, MAX(transaction_date) )
FROM SalesTransaction) AND c. count > 1
) thirtydays
INNER JOIN
(
SELECT Color.name as color, ISNULL(temp.sales, 0) AS totalsales
FROM
Color LEFT JOIN (
SELECT DISTINCT vc.color, COUNT(DISTINCT s.VIN) AS sales
FROM (SELECT VIN, COUNT(1) count FROM  VehicleHasColor GROUP BY VIN) c
LEFT JOIN VehicleHasColor vc ON vc.VIN = c.VIN
LEFT JOIN SalesTransaction s ON vc.VIN = s.VIN
WHERE s. transaction_date > (SELECT dateadd(YEAR,-1,
MAX(transaction_date) )FROM SalesTransaction)  AND c. count = 1
GROUP BY vc.color
) temp ON temp.color = Color.name
UNION
SELECT DISTINCT 'multiple', COUNT(DISTINCT s.VIN) AS sales
FROM (SELECT VIN, COUNT(1) count FROM VehicleHasColor GROUP BY VIN) c
LEFT JOIN VehicleHasColor vc ON vc.VIN = c.VIN
LEFT JOIN SalesTransaction s ON vc.VIN = s.VIN
WHERE s. transaction_date > (SELECT dateadd(YEAR,-1, MAX(transaction_date) )
FROM SalesTransaction)  AND c. count > 1
) lastyear ON thirtydays.color = lastyear.color
INNER JOIN
(
SELECT Color.name as color, ISNULL(temp.sales, 0) AS totalsales
FROM
Color LEFT JOIN (
SELECT DISTINCT vc.color, COUNT(DISTINCT s.VIN) as sales
FROM (SELECT VIN, COUNT(1) count FROM  VehicleHasColor GROUP BY VIN) c
LEFT JOIN VehicleHasColor vc ON vc.VIN = c.VIN
LEFT JOIN SalesTransaction s ON vc.VIN = s.VIN
WHERE  c. count = 1
GROUP BY vc.color
) temp ON temp.color = Color.name
UNION
SELECT DISTINCT 'multiple' AS multiColor, COUNT(DISTINCT s.VIN) AS sales
FROM (SELECT VIN, COUNT(1) count FROM  VehicleHasColor GROUP BY VIN) c
```

```
LEFT JOIN VehicleHasColor vc ON vc.VIN = c.VIN
LEFT JOIN SalesTransaction s ON vc.VIN = s.VIN
WHERE c. count > 1
) overall ON thirtydays.color = overall.color;
```

10.2 View Sales by Type Report

- Sales by Type Report
  - No. of vehicle sold in selected range of days for a type

```
WITH VehicleCTE
AS(
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date,  'Car' as vehicle_type
FROM Vehicle v INNER JOIN Car c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Convertible'
FROM Vehicle v INNER JOIN Convertible c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Truck'
FROM Vehicle v INNER JOIN Truck t on  v.VIN = t.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Van'
FROM Vehicle v INNER JOIN Van  on  v.VIN = Van.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'SUV'
FROM Vehicle v INNER JOIN SUV s  on  v.VIN = s.VIN
)

SELECT thirtydays.vehicle_type, thirtydays.totalsales AS thirtyDaysSales,  lastyear.totalsales
AS lastYearSales, overall.totalsales AS overallSales
FROM (
SELECT all_vehicle_types.vehicle_type, ISNULL(temp.sales, 0) AS totalsales
FROM (SELECT DISTINCT VehicleCTE.vehicle_type FROM VehicleCTE) as
all_vehicle_types
LEFT JOIN (
SELECT DISTINCT v.vehicle_type, COUNT(DISTINCT s.VIN) AS sales
FROM  VehicleCTE v
INNER JOIN SalesTransaction s ON v.VIN = s.VIN
WHERE s. transaction_date > (
SELECT DATEADD(DAY,-30, MAX(transaction_date) )
FROM SalesTransaction)
GROUP BY v.vehicle_type
) temp ON temp.vehicle_type = all_vehicle_types.vehicle_type

) thirtydays
INNER JOIN
(
SELECT all_vehicle_types.vehicle_type, ISNULL(temp.sales, 0) AS totalsales
FROM (SELECT DISTINCT VehicleCTE.vehicle_type FROM VehicleCTE) as
all_vehicle_types
LEFT JOIN (
SELECT DISTINCT v.vehicle_type, COUNT(DISTINCT s.VIN) AS sales
FROM  VehicleCTE v
INNER JOIN SalesTransaction s ON v.VIN = s.VIN
WHERE s. transaction_date > (SELECT DATEADD(YEAR,-1,
MAX(transaction_date) )FROM SalesTransaction)
GROUP BY v.vehicle_type
) temp ON temp.vehicle_type = all_vehicle_types.vehicle_type
) lastyear ON thirtydays.vehicle_type = lastyear.vehicle_type
INNER JOIN
(
SELECT all_vehicle_types.vehicle_type, ISNULL(temp.sales, 0) AS totalsales
FROM (SELECT DISTINCT VehicleCTE.vehicle_type FROM VehicleCTE) as
all_vehicle_types
LEFT JOIN (
SELECT DISTINCT v.vehicle_type, COUNT(DISTINCT s.VIN) AS sales
FROM  VehicleCTE v
INNER JOIN SalesTransaction s ON v.VIN = s.VIN
GROUP BY v.vehicle_type
) temp ON temp.vehicle_type = all_vehicle_types.vehicle_type
) overall ON thirtydays.vehicle_type = overall.vehicle_type;
```

10.3 View Sales by Manufacturer Report

- Sales by Manufacturer Report
    - No. of vehicle sold in different length of days for a manufacturer

- If a manufacturer does not have sales, does not display

```
WITH VehicleCTE
AS(
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date,  'Car' as vehicle_type
FROM Vehicle v INNER JOIN Car c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Convertible'
FROM Vehicle v INNER JOIN Convertible c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Truck'
FROM Vehicle v INNER JOIN Truck t on  v.VIN = t.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Van'
FROM Vehicle v INNER JOIN Van  on  v.VIN = Van.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'SUV'
FROM Vehicle v INNER JOIN SUV s  on  v.VIN = s.VIN
)

SELECT thirtydays.manufacturer_name, thirtydays.sales AS thirtydaysSales,  lastyear.sales AS
lastYearSales, overall.sales AS overallSales
FROM (
SELECT DISTINCT v. manufacturer_name, COUNT(DISTINCT s.VIN) AS sales
FROM  VehicleCTE v
INNER JOIN SalesTransaction s ON v.VIN = s.VIN
WHERE s. transaction_date > (
SELECT DATEADD(DAY,-30, MAX(transaction_date) )
FROM SalesTransaction)
GROUP BY v.manufacturer_name
) thirtydays
INNER JOIN
(
SELECT DISTINCT v. manufacturer_name, COUNT(DISTINCT s.VIN) AS sales
FROM  VehicleCTE v
INNER JOIN SalesTransaction s ON v.VIN = s.VIN
WHERE s. transaction_date > (SELECT DATEADD(YEAR,-1,
MAX(transaction_date) )FROM SalesTransaction)
GROUP BY v.manufacturer_name
) lastyear ON thirtydays.manufacturer_name = lastyear.manufacturer_name
```

```
INNER JOIN
(
SELECT DISTINCT v. manufacturer_name, COUNT(DISTINCT s.VIN) AS sales
FROM  VehicleCTE v
INNER JOIN SalesTransaction s ON v.VIN = s.VIN
GROUP BY v.manufacturer_name
) overall ON thirtydays.manufacturer_name = overall.manufacturer_name;
```

10.4 View Gross Customer Income Report

- Gross Customer Income Report
  - Part 1: display top 15 customers include{customer name, first sale date, recent action date(sale or repair), number of sales, number of repair, gross income} order by gross income, recent action date desc

```
WITH tempcte AS(
SELECT customer_ID,  SUM(sold_price - invoice_price*1.25) AS income,
MIN(transaction_date) AS  first_date ,
MAX(transaction_date) AS last_date FROM SalesTransaction
GROUP BY customerID
UNION
SELECT customerID, SUM(total_part_cost + labor_charge) ,
MIN(start_date)
MAX(start_date) FROM RepairDetails r
INNER JOIN
(SELECT SUM(quantity* price)  AS total_part_cost
FROM Part p
WHERE p.VIN = r.VIN AND p.start_date = r. start_date
GROUP BY p.VIN, p.start_date)
GROUP BY customerID
)

SELECT TOP 15
CASE WHEH in.driver_license IS NOT NULL THEN in CONCAT(in.first_name, in.last_name)
ELSE bu.business_name END AS CustomerName,
SUM(income),MIN(first_date), MAX(last_date) from tempcte t
LEFT JOIN  Individual in ON t.customerID = in.customerID
LEFT JOIN Business bu ON t.customerID = bu.customerID
GROUP BY
CASE WHEH in.driver_license IS NOT NULL THEN in CONCAT(in.first_name, in.last_name)
ELSE bu.business_name END
ORDER BY SUM(income) DESC, MAX(last_transaction_date) DESC;
```

- Part 2: if User click on certain customer for more details, display:

- {vehicle: sale date, sold price, VIN, year, manufacturer, model, salesperson name } order by sale date desc, VIN ascending

```
WITH VehicleCTE
AS(
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date,  'Car' as vehicle_type
FROM Vehicle v INNER JOIN Car c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Convertible'
FROM Vehicle v INNER JOIN Convertible c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Truck'
FROM Vehicle v INNER JOIN Truck t on  v.VIN = t.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Van'
FROM Vehicle v INNER JOIN Van  on  v.VIN = Van.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'SUV'
FROM Vehicle v INNER JOIN SUV s  on  v.VIN = s.VIN
)

SELECT S.transaction_date, S.sold_price, S.VIN, V.model_year, V.manufacturer_name,
V.model_name, U.first_name, U.last_name
FROM Customer AS C
INNER JOIN SalesTransaction AS S ON C.customerID=S.customerID
INNER JOIN VehicleCTE AS V ON S.VIN=V.VIN
INNER JOIN Users AS U ON U.user_name=S.sales_writer_user_name
WHERE (C.customerID= '$SelectedCustomerID')
ORDER BY S.transaction_date DESC, S.VIN ASC;
```

- {repair: start date, end date, VIN, odometer value, parts cost, labor cost, total cost, service writer}

```
WITH VehicleCTE
AS(
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date,  'Car' as vehicle_type
FROM Vehicle v INNER JOIN Car c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Convertible'
FROM Vehicle v INNER JOIN Convertible c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Truck'
FROM Vehicle v INNER JOIN Truck t on  v.VIN = t.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Van'
FROM Vehicle v INNER JOIN Van  on  v.VIN = Van.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'SUV'
FROM Vehicle v INNER JOIN SUV s  on  v.VIN = s.VIN
)

SELECT R.start_date, R.complete_date, R.VIN, R.odometer_value, SUM(P.quantity*P.price)
AS parts_cost, R.labor_charge, (parts_cost+ R.labor_charge) AS total_cost, U.first_name,
U.last_name
FROM RepairDetails AS R
INNER JOIN VehicleCTE AS V ON R.VIN=V.VIN
INNER JOIN Parts AS P ON P.VIN=R.VIN
INNER JOIN Users AS U ON U.user_name=R.repair_starter
WHERE (R.customerID= '$SelectedCustomerID')
GROUP BY R.VIN, R.start_date
ORDER BY R.start_date DESC, R.complete_date DESC, R.VIN ASC;
```

10.5 View Repairs by Manufacturers/Type/Model Report

- Part 1. Repairs by Manufacturers
  - Display repair by Manufacturers/Type/Model
    - {Count of repairs, the sum of all parts costs, the sum of all labor costs, and the sum of total repair costs, including any repairs in progress}
  - Exclude manufacturers without repair

```
WITH VehicleCTE
AS(
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date,  'Car' as vehicle_type
FROM Vehicle v INNER JOIN Car c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Convertible'
FROM Vehicle v INNER JOIN Convertible c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Truck'
FROM Vehicle v INNER JOIN Truck t on  v.VIN = t.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Van'
FROM Vehicle v INNER JOIN Van  on  v.VIN = Van.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'SUV'
FROM Vehicle v INNER JOIN SUV s  on  v.VIN = s.VIN
)

SELECT V.manufacturer_name, COUNT(DISTINCT R.VIN, R.start_date),
SUM(P.quantity*P.price) AS total_parts_cost, SUM(R.labor_charge) AS total_labor_cost,
(total_parts_cost+total_labor_cost) AS total_repair_cost
FROM RepairDetails AS R
INNER JOIN VehicleCTE as V ON V.VIN=R.VIN
INNER JOIN Parts AS P ON P.VIN=R.VIN
GROUP BY V.manufacturer_name
ORDER BY V.manufacturer_name ASC;
```

- Part 2. When choosing to view more repairs detail by type
  - Detail by type

```
WITH VehicleCTE
AS(
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date,  'Car' as vehicle_type
FROM Vehicle v INNER JOIN Car c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Convertible'
FROM Vehicle v INNER JOIN Convertible c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Truck'
FROM Vehicle v INNER JOIN Truck t on  v.VIN = t.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Van'
FROM Vehicle v INNER JOIN Van  on  v.VIN = Van.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'SUV'
FROM Vehicle v INNER JOIN SUV s  on  v.VIN = s.VIN
)

SELECT COUNT(DISTINCT R.VIN, R.start_date),  SUM(P.quantity*P.price) AS
total_parts_cost, SUM(R.labor_charge) AS total_labor_cost, (total_parts_cost+total_labor_cost)
AS total_repair_cost
FROM RepairDetails AS R
INNER JOIN VehicleCTE as V ON V.VIN=R.VIN
INNER JOIN Parts AS P ON P.VIN=R.VIN
WHERE (V.manufacturer_name='$SelectedManufacturerName')
GROUP BY V.vehicle_type
ORDER BY V.vehicle_type ASC;
```

- Detail by model under the type

```
WITH VehicleCTE
AS(
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date,  'Car' as vehicle_type
FROM Vehicle v INNER JOIN Car c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Convertible'
FROM Vehicle v INNER JOIN Convertible c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Truck'
FROM Vehicle v INNER JOIN Truck t on  v.VIN = t.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Van'
FROM Vehicle v INNER JOIN Van  on  v.VIN = Van.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'SUV'
FROM Vehicle v INNER JOIN SUV s  on  v.VIN = s.VIN
)
```

```
SELECT COUNT(DISTINCT R.VIN, R.start_date),  SUM(P.quantity*P.price) AS
total_parts_cost, SUM(R.labor_charge) AS total_labor_cost, (total_parts_cost+total_labor_cost)
AS total_repair_cost
FROM RepairDetails AS R
INNER JOIN VehicleCTE as V ON V.VIN=R.VIN
INNER JOIN Parts AS P ON P.VIN=R.VIN
WHERE (V.manufacturer_name='$SelectedManufacturerName') AND
(V.vehicle_type='$SelectedType')
GROUP BY V.model_name
ORDER BY COUNT(DISTINCT R.VIN, R.start_date) DESC;
```

10.6 View Below Cost Sales Report

- Below Cost Sales Report
  - Display all vehicles sold below Invoice Price
    - {Include transaction detail of that sale, and highlight sale<=95%}

```
WITH VehicleCTE
AS(
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date,  'Car' as vehicle_type
FROM Vehicle v INNER JOIN Car c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Convertible'
FROM Vehicle v INNER JOIN Convertible c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Truck'
FROM Vehicle v INNER JOIN Truck t on  v.VIN = t.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Van'
FROM Vehicle v INNER JOIN Van  on  v.VIN = Van.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'SUV'
FROM Vehicle v INNER JOIN SUV s  on  v.VIN = s.VIN
)

SELECT (C.business_name + C.first_name + SPACE(1) + C.last_name) AS customer_name,
S.transaction_date, S.VIN, S.sold_price, (S.sold_price/S.invoice_price*100) AS ratio,
V.model_year, V.manufacturer_name, V.model_name, V.invoice_price, Users.first_name,
Users.last_name
FROM (Customer LEFT OUTER JOIN Individual AS I ON C.customerID=I.customerID
LEFT OUTER JOIN Business AS B ON C.customerID=B.customerID) AS C
INNER JOIN SalesTransaction AS S ON C.customerID=S.customerID
INNER JOIN VehicleCTE AS V ON S.VIN=V.VIN
INNER JOIN Users AS U ON U.user_name=S.sales_writer_user_name
WHERE(S.sold_price<V.invoice_price)
ORDER BY S.transaction_date  DESC,  (S.sold_price/S.invoice_price*100) DESC ;
```

10.7 View Average Time in Inventory Report

- Average Time in Inventory Report
  - No. of days vehicle has stayed in inventory for a Vehicle_Type
  - If there is no sale for a Vehicle_Type, display N/A

```
WITH VehicleCTE
AS(
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date,  'Car' as vehicle_type
FROM Vehicle v INNER JOIN Car c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Convertible'
FROM Vehicle v INNER JOIN Convertible c on  v.VIN = c.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Truck'
FROM Vehicle v INNER JOIN Truck t on  v.VIN = t.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'Van'
FROM Vehicle v INNER JOIN Van  on  v.VIN = Van.VIN
UNION
SELECT v.VIN, description, model_year, model_name, invoice_price,manufacturer_name ,
inv_writer_user_name, add_date, 'SUV'
FROM Vehicle v INNER JOIN SUV s  on  v.VIN = s.VIN
)
 SELECT v.vehicle_type, ISNULL(SUM(DATEDIFF(DAY,transaction_date, add_date) +
1)/COUNT(v.VIN), 'N/A') AS avg_day_in_inventory
FROM VehicleCTE v LEFT JOIN SalesTransaction t ON v.vin = t.vin
GROUP BY v.vehicle_type;
```

10.8 View Parts Statics Report

- Parts Statics Report
    - Display by vendor, including the number of parts supplied by that vendor, and the total dollar amount spent on parts

```
SELECT vendor, SUM(quantity) AS totalpartssupplied, SUM(quantity*price) AS totalspent
FROM Part
GROUP BY vendor;
```

10.9 View Monthly Sales Report

- Monthly Sales Report
    - Display all Sales_Transaction by year and month, total number of vehicles sold, total sales income, the total net income, and the sold price/invoice price ratio as a percentage
        - If a month/year does not have sales, does not display
        - If price/invoice for a month is >= 125%, row highlighted as a green background.
        - If the ratio is <= 110%, highlighted as yellow background.
        - Order by year and month descending

```
SELECT YEAR(transaction_date) AS Year , MONTH(transaction_date) AS Month,
COUNT(VIN) as totalVehicleSold ,SUM(sold_price - invoice_price) as totalNetIncome,
SUM(sold_price ) as SalesIncome, SUM(sold_price)/SUM(invoice_price) as Ratio
from SalesTransaction t inner join Vehicle v on t.VIN = v.VIN
group By Year(transaction_date), Month(transaction_date)
order by Year(transaction_date) desc ,  Month(transaction_date) desc;
```

- Display drilldown report for year and month
    - Display top performing Salespeople
      {include first and last name, the number of vehicles and total sale in that year and month}
    - Order by total vehicles descending followed by total sales descending.

```
SELECT u.first_name, u.last_name, COUNT(VIN) AS TotalVehicleSold
FROM SalesTransaction INNER JOIN Salespeople s
user_name ON sales_writer_user_name = s.user_name
INNER JOIN Users u on s.user_name = u.user_name
WHERE YEAR(transaction_date) = '$Year' and MONTH(transaction_date) = '$Month'
GROUP BY u.first_name, u.last_name,
ORDER BY COUNT(VIN) DESC;
```