CS 6400

Phase 2 Report

GTSI Team 04

GTSI Team 04
2021-10-31

Table of Contents

Login

Abstract Code

- If user clicks ***Search Vehicle*** button, go to **Search Vehicle** task
- User enters *username*('$Username') and *password*('$Password') in input field on Login Form
- When ***Login*** button is clicked:
    - If data validation is successful for both username and password, then:

```
SELECT password
FROM `User`
WHERE email= '$Email';
```

        - If User record is not found or User.password != '$Password'
            - Go back to **Login Form**, with Error Message('Login failed, please try again.')
        - Else:
            - Store USERNAME='$Username' in cache(can be accessed throughout this session)
            - Open **Navigation Panel**

```
SELECT username
FROM  `Owner`
WHERE username = USERNAME;
```

                - If owner record exists:
                    - USER_TYPE='Owner'
                    - ***Search Vehicle, Log Out, Add Vehicle, Search Repair*** and ***View Report*** buttons are shown
                - Else:

```
SELECT username
FROM `InventoryClerk`
WHERE username = USERNAME;
```

                    - If inventory clerk record exists
                        - USER_TYPE='Inventory Clerk'
                        - ***Search Vehicle, Log Out*** and ***Add Vehicle*** buttons are shown

```
SELECT username
FROM `Salespeople`
WHERE username = USERNAME;
```

                    - If salespeople record exists
                        - USER_TYPE='Salespeople'
                        - ***Search Vehicle*** and ***Log Out*** buttons are shown(because need to search before selling)

```
SELECT username
FROM `ServiceWriter`
WHERE username = USERNAME;
```

                    - If service writer record exists
                        - USER_TYPE='Service Writer'
                        - ***Search Vehicle, Log Out*** and ***Search Repair*** buttons are shown

```
SELECT username
FROM `Manager`
WHERE username = USERNAME;
```

                    - If manager record exists
                        - USER_TYPE='Manager'
                        - ***Search Vehicle, Log Out*** and 9 View Report (***View Sales by Color, View Sales by Type, View Sales by Manufacturer, View Gross Customer Income, View Repairs by Manufacturer/Type/Model, View Below Cost Sales, View Average Time in Inventory, View Parts Statistics, View Monthly Sales***) buttons are shown
            - Store USER_TYPE in cache
            - Upon:

- Click ***Search Vehicle*** button - Jump to the **Search Vehicle** task
- Click ***Add Vehicle*** button - Jump to the **Add Vehicle** task
- Click ***Search Repair*** button - Jump to the **Search Repair** task
- Click one of the 9 View Report buttons - Jump to corresponding task
- Click ***Log Out*** button - Invalidate login session and go back to **Login Form**

Search Vehicle

Abstract Code

- If USERNAME is NULL:
  - Anonymous user optionally selects *vehicle type*('$vehicle_Type'), *color*('$color'), *manufacturer name*('$manufacturer_name') and enters *model year*('$model_year'), *list price*('$list_price') with selecting '<' or '>' and *keyword*('$keyword') in input field on **Vehicle List** form.
  - When ***Search*** button is clicked:
    - If list price selects '>':

```
SELECT
    v.vin as vin,
    v.vehicle_type as vehicle_type,
    v.model_year as model_year,
    v.manufacturer as manufacturer,
    v.model_name as model_name,
    v.description as description,
    GROUP_CONCAT(vc.color) as color,
    v.invoice_price*1.25 as list_price
FROM `VehicleColor` as vc
GROUP BY vc.vin
INNER JOIN `Vehicle` as v
ON v.vin = vc.vin
WHERE
    v.vehicle_type = '$vehicle_type'
    AND vc.color = '$color'
    AND v.manufacturer = '$manufacturer_name'
    AND v.model_year = '$model_year'
    AND v.invoice_price*1.25 > '$list_price'
    AND v.description LIKE "%'key_word'%"
    AND NOT IN
    (SELECT vin
    FROM `Sale`)
ORDER BY vin ASC;
```

    - Else:

```
SELECT
    v.vin as vin,
    v.vehicle_type as vehicle_type,
    v.model_year as model_year,
    v.manufacturer as manufacturer,
    v.model_name as model_name,
    v.description as description,
    GROUP_CONCAT(vc.color) as color
    v.invoice_price*1.25 as list_price
FROM `VehicleColor` as vc
GROUP BY vc.vin
INNER JOIN `Vehicle` as v
ON v.vin = vc.vin
WHERE
    v.vehicle_type = '$vehicle_type'
    AND vc.color = '$color'
```

```
        AND v.manufacturer = '$manufacturer_name'
        AND v.model_year = '$model_year'
        AND v.invoice_price*1.25 < '$list_price'
        AND v.description LIKE "%'key_word'%"
        AND vin NOT IN
        ( SELECT vin
        FROM `Sale` s )
ORDER BY vin ASC;
```

- If not exists(no unsold vehicles meets the search criteria), then:
  - return an error message: “Sorry, it looks like we don’t have that in stock!”, allowing the user to search again
- Else:
  - Display the result
  - User click ***Details*** button:
    - Jump to **View Vehicle Detail** task and pass in '$vin' and '$vehicle_type'

- Else (logged in user):
  - User optionally selects *vehicle type*('$vehicle_type'), *color*('$color'), *manufacturer name*('$manufacturer_name') and enters *model year*('$model_year'), *list price*('$list_price') with selecting ‘<’ or ‘>’ , *keyword*('$keyword') and *vin*('$vin')

```
SELECT
        v.vin as vin,
        v.vehicle_type as vehicle_type,
        v.model_year as model_year,
        v.manufacturer as manufacturer,
        v.model_name as model_name,
        v.description as description,
        GROUP_CONCAT(vc.color) as color
        v.invoice_price*1.25 as list_price
FROM `VehicleColor` as vc
GROUP BY vc.vin
INNER JOIN `Vehicle` as v
INNER ON v.vin = vc.vin
WHERE
        v.vin = '$vin'
        v.vehicle_type = '$vehicle_type'
        AND vc.color = '$color'
        AND v.manufacturer = '$manufacturer_name'
        AND v.model_year = '$model_year'
        AND v.invoice_price*1.25 <(or >) '$list_price'
        AND v.description LIKE "%'key_word'%"
        AND vin
        NOT IN
        ( SELECT vin
        FROM `Sale` s )
ORDER BY vin ASC;
```

    - If not exists(no unsold vehicles meets the search criteria), then:
      - return an error message: “Sorry, it looks like we don’t have that in stock!”, allowing the user to search again
    - Else:
      - Display the result
      - User click ***Details*** button:
        - Jump to the **View Vehicle Detail** task and pass in '$vin' and '$vehicle_type'
- If user clicks ***Cancel*** button:
  - If USER_TYPE is Null:
    - Go back to **login form**
  - Else:

- Go back to **Navigation Panel**

View Vehicle Detail

Abstract Code

- Get '$vin' and '$vehicle_type' from Search Vehicle task
- If USERNAME = NULL:
  - If '$vehicle_type' == car:

```
SELECT
        v.vin as vin,
        v.vehicle_type as vehicle_type,
        c.number_of_doors as number_of_doors,
        v.model_year as model_year,
        v.model_name as model_name,
        v.manufacturer as manufacturer,
        GROUP_CONCAT(vc.color) as color,
        v.invoice_price*1.25 as list_price,
        v.description as description
FROM
        `VehicleColor` as vc
GROUP BY vc.vin
INNER JOIN `Vehicle` as v
ON v.vin = vc.vin
INNER JOIN `Car` as c
ON v.vin = c.vin
WHERE
v.vin = '$vin';
```

    - Display the result
  - If '$vehicle_type' == convertible:

```
SELECT
        v.vin as vin,
        v.vehicle_type as vehicle_type,
        c.roof_type as roof_type,
        c.back_seat_count as back_seat_count,
        v.model_year as model_year,
        v.model_name as model_name,
        v.manufacturer as manufacturer,
        GROUP_CONCAT(vc.color) as color,
        v.invoice_price*1.25 as list_price,
        v.description as description
FROM
        `VehicleColor` as vc
GROUP BY vc.vin
INNER JOIN `Vehicle` as v
ON v.vin = vc.vin
INNER JOIN `Convertible` as c
ON v.vin = c.vin
WHERE
v.vin = '$vin';
```

    - Display the result
  - If '$vehicle_type' == truck:

```
SELECT
        v.vin as vin,
        v.vehicle_type as vehicle_type,
        c.cargo_capacity as cargo_capacity,
```

```
        c.cargo_cover_type as cargo_cover_type,
        c.number_of_rear_axles as number_of_rear_axles,
        v.model_year as model_year,
        v.model_name as model_name,
        v.manufacturer as manufacturer,
        GROUP_CONCAT(vc.color) as color,
        v.invoice_price*1.25  as list_price,
        v.description as description
FROM
        `VehicleColor` as vc
GROUP BY vc.vin
INNER JOIN `Vehicle` as v
ON v.vin = vc.vin
INNER JOIN `Truck` as c
ON v.vin = c.vin
WHERE
v.vin = '$vin';
```

- Display the result

- If '$vehicle_type' == vanMinivan:

```
SELECT
        v.vin as vin,
        v.vehicle_type as vehicle_type,
        c.has_drivers_side_back_door as has_drivers_side_back_door,
        v.model_year as model_year,
        v.model_name as model_name,
        v.manufacturer as manufacturer,
        vc.color as color,
        v.invoice_price*1.25  as list_price,
        v.description as description
FROM
        Vehicle as v
        INNER JOIN VehicleColor as vc ON v.vin = vc.vin
        INNER JOIN VanMinivan as c ON v.vin = c.vin
WHERE
v.vin = '$vin';
```

- Display the result

- If '$vehicle_type' == suv:

```
SELECT
        v.vin as vin,
        v.vehicle_type as vehicle_type,
        c.drivetrain_type as drivetrain_type,
        c.number_of _cupholders as number_of _cupholders,
        v.model_year as model_year,
        v.model_name as model_name,
        v.manufacturer as manufacturer,
        GROUP_CONCAT(vc.color) as color,
        v.invoice_price*1.25  as list_price,
        v.description as description
FROM
        VehicleColor as vc
INNER JOIN Vehicle as v
ON v.vin = vc.vin
INNER JOIN Car as c ON v.vin = c.vin
GROUP BY vc.vin
WHERE
v.vin = '$vin';
```

- Display the result

- If USER_TYPE = 'Inventory Clerk':
  - Details page will additionally display invoice price
  - If '$vehicle_type' == car:

```
SELECT
        v.vin as vin,
        v.vehicle_type as vehicle_type,
        c.number_of_doors as number_of_doors,
        v.model_year as model_year,
        v.model_name as model_name,
        v.manufacturer as manufacturer,
        GROUP_CONCAT(vc.color) as color,
        v.invoice_price*1.25 as list_price,
        v.description as description,
        v.invoice_price as invoice_price
FROM
        VehicleColor as vc
INNER JOIN Vehicle as v
ON v.vin = vc.vin
INNER JOIN Car as c
ON v.vin = c.vin
GROUP BY vc.vin
WHERE
v.vin = '$vin';
```

    - Display the result
  - If '$vehicle_type' == convertible:

```
SELECT
        v.vin as vin,
        v.vehicle_type as vehicle_type,
        c.roof_type as roof_type,
        c.back_seat_count as back_seat_count,
        v.model_year as model_year,
        v.model_name as model_name,
        v.manufacturer as manufacturer,
        GROUP_CONCAT(vc.color) as color,
        v.invoice_price*1.25 as list_price,
        v.description as description,
        v.invoice_price as invoice_price
FROM
        VehicleColor as vc
INNER JOIN Vehicle as v
ON v.vin = vc.vin
INNER JOIN Convertible as c
ON v.vin = c.vin
GROUP BY vc.vin
WHERE
v.vin = '$vin';
```

    - Display the result
  - If '$vehicle_type' == truck:

```
SELECT
        v.vin as vin,
        v.vehicle_type as vehicle_type,
        c.cargo_capacity as cargo_capacity,
        c.cargo_cover_type as cargo_cover_type,
        c.number_of_rear_axles as number_of_rear_axles,
        v.model_year as model_year,
        v.model_name as model_name,
        v.manufacturer as manufacturer,
```

```
        GROUP_CONCAT(vc.color) as color,
        v.invoice_price*1.25 as list_price,
        v.description as description,
        v.invoice_price as invoice_price
FROM
        VehicleColor as vc
INNER JOIN Vehicle as v
ON v.vin = vc.vin
INNER JOIN Truck as c
ON v.vin = c.vin
GROUP BY vc.vin
WHERE
v.vin = '$vin';
```

- Display the result

- If '$vehicle_type' == vanMinivan:

```
SELECT
        v.vin as vin,
        v.vehicle_type as vehicle_type,
        c.has_drivers_side_back_door as has_drivers_side_back_door,
        v.model_year as model_year,
        v.model_name as model_name,
        v.manufacturer as manufacturer,
        GROUP_CONCAT(vc.color) as color,
        v.invoice_price*1.25 as list_price,
        v.description as description,
        v.invoice_price as invoice_price
FROM    VehicleColor as vc
INNER JOIN Vehicle as v
ON v.vin = vc.vin
INNER JOIN VanMinivan as c
ON v.vin = c.vin
GROUP BY vc.vin
WHERE   v.vin = '$vin';
```

- Display the result

- If '$vehicle_type' == suv:

```
SELECT
        v.vin as vin,
        v.vehicle_type as vehicle_type,
        c.drivetrain_type as drivetrain_type,
        c.number_of _cupholders as number_of _cupholders,
        v.model_year as model_year,
        v.model_name as model_name,
        v.manufacturer as manufacturer,
        GROUP_CONCAT(vc.color) as color,
        v.invoice_price*1.25 as list_price,
        v.description as description,
        v.invoice_price as invoice_price
FROM    VehicleColor as vc
INNER JOIN Vehicle as v
ON v.vin = vc.vin
INNER JOIN Car as c
ON v.vin = c.vin
GROUP BY vc.vin
WHERE   v.vin = '$vin';
```

- Display the result

- If USER_TYPE = 'Salesperson':
  - Display the same result as anonymous user

- If ***Sell*** button on the detail page is clicked, go to **Sell Vehicle** task with '$vin' and '$vehicle_type'

- If USER_TYPE = 'Manager':
  - Details page will additionally display the inventory who added the vehicle, invoice price, added_date
  - If '$vehicle_type' == car:

```
SELECT
        v.vin as vin,
        v.vehicle_type as vehicle_type,
        c.number_of_doors as number_of_doors,
        v.model_year as model_year,
        v.model_name as model_name,
        v.manufacturer as manufacturer,
        GROUP_CONCAT(vc.color) as color,
        v.invoice_price*1.25 as list_price,
        v.description as description,
        v.inventory_clerk_username as inventory_clerk_username,
        v.invoice_price as invoice_price,
        v.added_date as added_date
FROM
        VehicleColor as vc
        INNER JOIN Vehicle as v
        ON v.vin = vc.vin
        INNER JOIN Car as c
        ON v.vin = c.vin
        GROUP BY vc.vin
WHERE v.vin = '$vin';
```

  - If '$vehicle_type' == convertible:

```
SELECT
        v.vin as vin,
        v.vehicle_type as vehicle_type,
        c.roof_type as roof_type,
        c.back_seat_count as back_seat_count,
        v.model_year as model_year,
        v.model_name as model_name,
        v.manufacturer as manufacturer,
        GROUP_CONCAT(vc.color) as color,
        v.invoice_price*1.25 as list_price,
        v.description as description,
        v.inventory_clerk_username as inventory_clerk_username,
        v.invoice_price as invoice_price,
        v.added_date as added_date
FROM
        VehicleColor as vc
        INNER JOIN Vehicle as v
        ON v.vin = vc.vin
        INNER JOIN Convertible as c
        ON v.vin = c.vin
        GROUP BY vc.vin
WHERE v.vin = '$vin';
```

  - If '$vehicle_type' == truck:

```
SELECT
        v.vin as vin,
        v.vehicle_type as vehicle_type,
        c.cargo_capacity as cargo_capacity,
        c.cargo_cover_type as cargo_cover_type,
        c.number_of_rear_axles as number_of_rear_axles,
        v.model_year as model_year,
        v.model_name as model_name,
```

```
        v.manufacturer as manufacturer,
        GROUP_CONCAT(vc.color) as color,
        v.invoice_price*1.25 as list_price,
        v.description as description,
        v.inventory_clerk_username as inventory_clerk_username,
        v.invoice_price as invoice_price,
        v.added_date as added_date
FROM
        VehicleColor as vc
        INNER JOIN Vehicle as v
        ON v.vin = vc.vin
        INNER JOIN Truck as c
        ON v.vin = c.vin
        GROUP BY vc.vin
WHERE   v.vin = '$vin';
```

- If '$vehicle_type' == vanMinivan:

```
SELECT
        v.vin as vin,
        v.vehicle_type as vehicle_type,
        c.has_drivers_side_back_door as has_drivers_side_back_door,
        v.model_year as model_year,
        v.model_name as model_name,
        v.manufacturer as manufacturer,
        GROUP_CONCAT(vc.color) as color,
        v.invoice_price*1.25 as list_price,
        v.description as description,
        v.inventory_clerk_username as inventory_clerk_username,
        v.invoice_price as invoice_price,
        v.added_date as added_date
FROM
        VehicleColor as vc
        GROUP BY vc.vin
        INNER JOIN Vehicle as v
        ON v.vin = vc.vin
        INNER JOIN VanMinivan as c
        ON v.vin = c.vin
WHERE v.vin = '$vin';
```

- If '$vehicle_type' == suv:

```
SELECT
        v.vin as vin,
        v.vehicle_type as vehicle_type,
        c.drivetrain_type as drivetrain_type,
        c.number_of _cupholders as number_of _cupholders,
        v.model_year as model_year,
        v.model_name as model_name,
        v.manufacturer as manufacturer,
        GROUP_CONCAT(vc.color) as color,
        v.invoice_price*1.25 as list_price,
        v.description as description,
        v.inventory_clerk_username as inventory_clerk_username,
        v.invoice_price as invoice_price,
        v.added_date as added_date
FROM
        VehicleColor as vc
        GROUP BY vc.vin
        INNER JOIN Vehicle as v
        ON v.vin = vc.vin
        INNER JOIN Car as c ON v.vin = c.vin
```

```
WHERE    v.vin = '$vin';
```

- If the vehicle has been sold, the buyer's information(except driver's license number of tax ID number) , List price, sold price, sales date and salesperson's first and last name are shown on Details page

```
SELECT
vin
FROM Sale
WHERE vin =  '$vin';
```

  - If exists, the vehicle has been sold:

```
SELECT
        c.city as city,
        c.postal_code as postal_code,
        c.state as state,
        c.street_address as street_address,
        c.phone_number as phone_number,
        c.email as email,
        CONCAT(i.first_name, "", i.last_name) as customer_name,
        v.invoice_price as invoice_price,
        s.sold_price as sold_price,
        s.purchase_date as purchase_date,
        CONCAT(u.first_name, " ", u.last_name) as salesperson_name
FROM
        Sale s
INNER JOIN Vehicle v
ON s.vin = v.vin
INNER JOIN Customer c
ON s.customer_id = c.id
INNER JOIN Individual i
ON s.customer_id = i.customer_id
INNER JOIN User u
ON u.username = s.salespeople_username
WHERE s.vin = '$vin';
```

    - If not exists, customer is in Business:

```
SELECT
        c.city as city,
        c.postal_code as postal_code,
        c.state as state,
        c.street_address as street_address,
        c.phone_number as phone_number,
        c.email as email,
        b.business_name as business_name,
        b.primary_contact_title as primary_contact_title,
        b.primary_contact_name as primary_contact_name,
        v.invoice_price as invoice_price,
        s.sold_price as sold_price,
        s.purchase_date as purchase_date,
        CONCAT(u.first_name, " ", u.last_name) as
salesperson_name
FROM
        Sale s
INNER JOIN Vehicle v
ON s.vin = v.vin
INNER JOIN Customer c
ON s.customer_id = c.id
INNER JOIN Business b
ON s.customer_id = b.customer_id
INNER JOIN User u
ON u.username = s.salespeople_username
```

```
WHERE s.vin = '$vin';
```

- If the vehicle has repair records, the customer name, service writer's first and last name, start date, complete date, labor charges, parts cost and total cost of each record are shown in the repair section

```
SELECT
vin
FROM Repair
WHERE vin =  '$vin';
```

  - If exists:

```
SELECT
        CONCAT(i.first_name, " ", i.last_name) as customer_name,
        CONCAT(u.first_name, " ", u.last_name) as service_writer_name,
        r.start_date as start_date,
        r.complete_date as end_date,
        r.labor_charge as labor charges,
        r.parts_cost as parts_cost,
        (r.parts_cost + r.parts_cost) as total cost
FROM
        Repair r
INNER JOIN Customer c
ON c.id = r.customer_id
INNER JOIN Individual i
ON c.id = i.customer_id
INNER JOIN User u
ON u.username = r.service_writer_username
WHERE r.vin = '$vin';
```

  - If not exists:

```
SELECT
        b.business_name as customer_name,
        CONCAT(u.first_name, " ", u.last_name) as service_writer_name,
        r.start_date as start_date,
        r.complete_date as end_date,
        r.labor_charge as labor charges,
        r.parts_cost as parts_cost,
        (r.parts_cost + r.parts_cost) as total cost
FROM
Repair r
INNER JOIN Customer c
ON c.id = r.customer_id
INNER JOIN Business b
ON c.id = b.customer_id
INNER JOIN User u
ON u.username = r.service_writer_username
WHERE r.vin = '$vin';
```

- If USER_TYPE = 'Owner':
  - Display all information
  - If '$vehicle_type' == car:

```
SELECT
        v.vin as vin,
        v.vehicle_type as vehicle_type,
        c.number_of_doors as number_of_doors,
        v.model_year as model_year,
        v.model_name as model_name,
        v.manufacturer as manufacturer,
        GROUP_CONCAT(vc.color) as color,
        v.invoice_price*1.25 as list_price,
        v.description as description,
        v.invoice_price as invoice_price
```

```
FROM
        VehicleColor as vc
INNER JOIN Vehicle as v
ON v.vin = vc.vin
INNER JOIN Car as c
ON v.vin = c.vin
GROUP BY vc.vin
WHERE   v.vin = '$vin';
```

- If '$vehicle_type' == convertible:

```
SELECT
        v.vin as vin,
        v.vehicle_type as vehicle_type,
        c.roof_type as roof_type,
        c.back_seat_count as back_seat_count,
        v.model_year as model_year,
        v.model_name as model_name,
        v.manufacturer as manufacturer,
        GROUP_CONCAT(vc.color) as color,
        v.invoice_price*1.25 as list_price,
        v.description as description,
        v.invoice_price as invoice_price
FROM
        VehicleColor as vc
INNER JOIN Vehicle as v
ON v.vin = vc.vin
INNER JOIN Convertible as c
ON v.vin = c.vin
GROUP BY vc.vin
WHERE
v.vin = '$vin';
```

- If '$vehicle_type' == truck:

```
SELECT
        v.vin as vin,
        v.vehicle_type as vehicle_type,
        c.cargo_capacity as cargo_capacity,
        c.cargo_cover_type as cargo_cover_type,
        c.number_of_rear_axles as number_of_rear_axles,
        v.model_year as model_year,
        v.model_name as model_name,
        v.manufacturer as manufacturer,
        GROUP_CONCAT(vc.color) as color,
        v.invoice_price*1.25 as list_price,
        v.description as description,
        v.invoice_price as invoice_price
FROM
        VehicleColor as vc
INNER JOIN Vehicle as v
ON v.vin = vc.vin
INNER JOIN Truck as c
ON v.vin = c.vin
GROUP BY vc.vin
WHERE   v.vin = '$vin';
```

- If '$vehicle_type' == vanMinivan:

```
SELECT
        v.vin as vin,
        v.vehicle_type as vehicle_type,
        c.has_drivers_side_back_door as has_drivers_side_back_door,
```

```
        v.model_year as model_year,
        v.model_name as model_name,
        v.manufacturer as manufacturer,
        GROUP_CONCAT(vc.color) as color,
        v.invoice_price*1.25 as list_price,
        v.description as description,
        v.invoice_price as invoice_price
FROM
        VehicleColor as vc
INNER JOIN Vehicle as v
ON v.vin = vc.vin
INNER JOIN VanMinivan as c
ON v.vin = c.vin
GROUP BY vc.vin
WHERE
v.vin = '$vin';
```

- If '$vehicle_type' == suv:

```
SELECT
        v.vin as vin,
        v.vehicle_type as vehicle_type,
        c.drivetrain_type as drivetrain_type,
        c.number_of _cupholders as number_of _cupholders,
        v.model_year as model_year,
        v.model_name as model_name,
        v.manufacturer as manufacturer,
        GROUP_CONCAT(vc.color) as color,
        v.invoice_price*1.25 as list_price,
        v.description as description,
        v.invoice_price as invoice_price
FROM
        VehicleColor as vc
INNER JOIN Vehicle as v
ON v.vin = vc.vin
INNER JOIN Car as c
ON v.vin = c.vin
GROUP BY vc.vin
WHERE
v.vin = '$vin';
```

- If ***Sell*** button on the detail page is clicked, go to **Sell Vehicle** task
- If the vehicle has been sold, the buyer's information(except driver's license number of tax ID number) , List price, sold price, sales date and salesperson's first and last name are shown on Details page

```
SELECT vin
FROM Sale
WHERE vin =  '$vin';
```

  - If exists, the vehicle has been sold:

```
SELECT
        c.city as city,
        c.postal_code as postal_code,
        c.state as state,
        c.street_address as street_address,
        c.phone_number as phone_number,
        c.email as email,
        CONCAT(i.first_name, "", i.last_name) as customer_name,
        v.invoice_price as invoice_price,
        s.sold_price as sold_price,
        s.purchase_date as purchase_date,
        CONCAT(u.first_name, " ", u.last_name) as salesperson_name
```

```
FROM
        Sale s
INNER JOIN Vehicle v
ON s.vin = v.vin
INNER JOIN Customer c
ON s.customer_id = c.id
INNER JOIN Individual i
ON s.customer_id = i.customer_id
INNER JOIN User u
ON u.username = s.salespeople_username
WHERE s.vin = '$vin';
```

- If not exists, customer is in Business:

```
SELECT
        c.city as city,
        c.postal_code as postal_code,
        c.state as state,
        c.street_address as street_address,
        c.phone_number as phone_number,
        c.email as email,
        b.business_name as business_name,
        b.primary_contact_title as primary_contact_title,
        b.primary_contact_name as primary_contact_name,
        v.invoice_price as invoice_price,
        s.sold_price as sold_price,
        s.purchase_date as purchase_date,
        CONCAT(u.first_name, " ", u.last_name) as
salesperson_name
FROM
        Sale s
INNER JOIN Vehicle v
ON s.vin = v.vin
INNER JOIN Customer c
ON s.customer_id = c.id
INNER JOIN Business b
ON s.customer_id = b.customer_id
JOIN User u
ON u.username = s.salespeople_username
WHERE s.vin = '$vin';
```

- If the vehicle has repair records, the customer name, service writer's first and last name, start date, complete date, labor charges, parts cost and total cost of each record are shown in the repair section

```
SELECT vin
FROM Repair
WHERE vin =  '$vin';
```

  - If exists:

```
SELECT
        CONCAT(i.first_name, " ", i.last_name) as customer_name
        CONCAT(u.first_name, " ", u.last_name) as service_writer_name,
        r.start_date as start_date,
        r.complete_date as end_date,
        r.labor_charge as labor charges,
        r.parts_cost as parts_cost,
        (r.parts_cost + r.parts_cost) as total cost
FROM
        Repair r
INNER JOIN Customer c
ON c.id = r.customer_id
INNER JOIN Individual i
```

```
ON c.id = i.customer_id
INNER JOIN User u
ON u.username = r.service_writer_username
WHERE r.vin = '$vin';
```

- If not exists:

```
SELECT
        b.business_name as customer_name,
        CONCAT(u.first_name, " ", u.last_name) as service_writer_name,
        r.start_date as start_date,
        r.complete_date as end_date,
        r.labor_charge as labor charges,
        r.parts_cost as parts_cost,
        (r.parts_cost + r.parts_cost) as total cost
FROM
Repair r
INNER JOIN Customer c
ON c.id = r.customer_id
INNER JOIN Business b
ON c.id = b.customer_id
INNER JOIN User u
ON u.username = r.service_writer_username
WHERE r.vin = '$vin';
```

- If user clicks ***Cancel*** button:
  - Go back to **Search Vehicle** task

Add Vehicle

Abstract Code

- User enters *VIN*('$vin') to search the target vehicle.

```
SELECT vin
FROM
Vehicle
WHERE vin = '$vin';
```

  - If result exists: display error message "Already exists"
  - Else:
    - User enters *description*('$description'), *added_date*('$current_date'), *model year*('$model_year'), *invoice price*('$invoice_price') in input field on **Vehicle Form**.
    - User selects or enters *manufacturer name*('$manufacturer_name'), multiply selects *color*('$color') and selects *vehicle type*('$vehicle_type') in the input field on **Vehicle Form**.
    - When User clicks ***ADD*** button and data validation is complete:

```
INSERT INTO Vehicle
( vin, description, added_Date, model_year, invoice_price, manufacturer, vehicle_type )
VALUES
( '$VIN', '$description', '$current_date', '$model_year', '$invoice_price', 'manufacturer_name',
'vehicle_type' );
```

    - Iteratively run the following SQL query:

```
INSERT INTO VehicleColor
VALUES ( '$VIN', '$color' );
```

    - If '$vehicle type' = 'Car':
      - User enters *number of doors*('$number_of_doors')

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

    - If '$vehicle type' = 'Convertible':
      - User enters *roof type*('$roof_type'), *back seat count*('$back_seat_count')

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

- If '$vehicle type' = 'Truck':
  - User enters *cargo capacity*('$cargo_capacity'), *number of rear axises*('$number_of_rear_axles')

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

- If '$vehicle type' = 'Van/Minivan':
  - User enters *has drivers side back door*('$has_drivers_side_back_door')

```
INSERT INTO VanMinivan
VALUES ('$VIN', 'vin');
```

- If '$vehicle type' = 'SUV':
  - User enters *number of cupholders*('$number_of_cupholders'), *drivetrain type*('$drivetrain_type')

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

- If successfully add the vehicle:
  - Jump to the **View Vehicle Detail** task with '$vin' and '$vehicle_type'

- If user clicks ***Cancel*** button:
  - Go back to **Navigation Panel**

Sell Vehicle

Abstract Code

- With '$vin' and '$vehicle_type' from View Vehicle Detail task:

```
SELECT
        vin
FROM
        Sale
WHERE
        vin = '$vin';
```

  - If no record exists:
    - Display an error message: "Sorry, it has already been sold!", jump back to **Search Vehicle** task
  - Else:
    - User enters *purchase date*('$current_date'), *sold price*('$sold_price') in input field on **Sales Order Form**
    - User clicks ***Lookup/Add Customer*** button, go to **Lookup/Add Customer** task and get '$customer_id'
    - When ***Sell*** button is clicked:
      - If '$customer_id' is null:
        - Display an error message: "No customer selected."
      - Else if USER_TYPE = 'Salespeople' and '$Sold Price' < Vehicle.invoice_price * 0.95:
        - Display error message ("Sold price is too low.")
      - Else:

```
INSERT INTO
Sale
VALUES
('$VIN', USER_TYPE, '$customer_id', '$current date', '$sold price');
```

- If user clicks ***Cancel*** button:
  - Go back to **View Vehicle Detail** Task

Look Up / Add Customer

Abstract Code

- User enters *driver license number*('$driver_license_number') or *tax identification number*('$tax_id_number') and clicks ***Search*** button
  - if user enters driver license number:

```
SELECT
customer_id
FROM
Individual
WHERE
driver_license_number = '$driver_license_number';
```

    - If record exists:

```
SELECT
id, city, postal_code, state, street_address, phone_number, email
FROM
Customer
WHERE
id = '$record.customer_id';
```

      - Display the result
    - else:
      - Display an error message: "No Customer Found", go back to previous page
  - if user entered tax identification number:

```
SELECT
customer_id
FROM
Business
WHERE
tax_id_number = '$tax_id_number';
```

    - If record exists:

```
SELECT
id, city, postal_code, state, street_address, phone_number, email
FROM
Customer
WHERE `id` = '$record.customer_id';
```

      - Display the result
    - else:
      - Display an error message, "No Customer Found", go back to previous page
- User selects customer type in a drop-down list:
  - If 'Individual' is selected:
    - User enters *first name*('$first_name'), *last name*('$last_name'), *driver license number*('$driver_license_number'), along with *address*('$city', '$postal_code', '$state', '$street_address'), *phone number*('$phone_number') and *email*('$email')
  - If 'Business' is selected:
    - User enters *primary contact name*('$primary_contact_name'), *title*('$primary_contact_title'), *tax identification number*('$tax_id_number'), *business name*('$business_name'), along with *address*('$city', '$postal_code', '$state', '$street_address'), *phone number*('$phone_number') and *email*('$email')
- User clicks ***Add*** button:
  - If data validation succeeds, then:
    - Store all the information in Customer table first:

```
INSERT INTO
Customer
OUTPUT Inserted.id
VALUES ('$city', '$postal_code', '$state', '$street_address', '$phone_number', '$email');
```

    - Get returned result and store in a variable '$customer_id'
    - If "Individual" is selected:

- Store all the information in Individual table:

```
INSERT INTO
Individual
VALUES ('$driver_license_number', '$first_name', '$last_name', '$customer_id');
```

- If "Business" is selected:
  - Store all the information in Business table:

```
INSERT INTO
Business
VALUES ('$tax_id_number', '$business_name', '$primary_contact_title',
'$primary_contact_name', '$customer_id');
```

- Else:
  - An error message is displayed: "data validation failed", allowing user to edit again

- If user clicks ***Cancel*** button:
  - Go back to **Add Repair** or **Sell Vehicle** task

View Repair

Abstract Code

- User enters *VIN*('$vin') in the input field
- If user clicks ***Search*** button:

```
SELECT vin
FROM Vehicle
WHERE  vin = '$vin';
```

  - If no vehicle record is found:
    - display an error message: "No corresponding vehicle"
  - Else:

```
SELECT vin
FROM Sale
WHERE vin = '$vin';
```

    - if no record exists:
      - Display an error message: "The vehicle has not been sold!"
    - else:

```
SELECT r.vin, r.start_date, r.complete_date, r.odometer, r.labor_charge,  r.description,
r.service_writer_username, r.customer_id, v.model_year,  v.vehicle_type,
v.model_year, v.manufacturer, vc.color
FROM Repair r
LEFT JOIN Vehicle v
ON r.vin = v.vin
LEFT JOIN VehicleColor vc
ON r.vin = vc.vin
WHERE r.vin = '$vin';
```

      - Display the result
      - If r.complete_date = null or r.complete_date < current date:
        - *Add Repair* button is shown
      - Else if r.complete_date = null:
        - Edit *Repair button* is shown
      - User clicks ***Add*** Repair button - go to **Add Repair** task with '$vin'
      - User clicks ***Edit*** Repair button - go to **Edit Repair** task with '$vin' and '$start_date'

- If user clicks ***Cancel*** button:
  - Go back to **Navigation Panel**

Add Repair

Abstract Code

- User clicks ***Lookup/Add Customer*** button:
  - go to **Lookup/Add Customer** task, get the customer_id('$customer_id')
- User enters *vin*('$vin'), *odometer*('$odometer'), *description*('$description') on **Repair Form**, current date automatically displays as the *start_date*('$start_date')
- User clicks Add button:
  - If data validation succeeds and '$customer_id' is not Null, then:

```
INSERT INTO
Repair (vin, start_date, odometer, description, service_writer_username, customer_id)
VALUES
('$vin', '$start_date', '$odometer', '$description', USERNAME, '$customer_id');
```

  - Else:
    - Display an error message: “No customer selected or data verification failed!”, allowing user to edit again
- User clicks ***Cancel*** button:
  - Go back to **View Repair** task

Edit Repair

Abstract Code

- Get the origin labor_charge of the repair as '$old_labor_charge'

```
SELECT labor_charge as old_labor_charge
FROM Repair
WHERE vin = ‘$vin’ and start_date = ‘$start_date’;
```

- User clicks ***Add Parts*** button:
  - go to **Add Parts** task
- User clicks ***Complete*** button
  - the current date is stored as the complete date ('$complete_date')

```
UPDATE Repair
SET
complete_date = '$complete_date'
WHERE vin = '$vin' and start_date = '$start_date';
```

- User enters *larbor_charge*('$labor_charge') in the input field
- User clicks ***Save*** button:
  - If data validation failed, then display an error message: “Data verification failed”
  - Else if USER_TYPE = 'Service writer'
    - If '$old_labor_charge' > '$labor_charge', display an error message: “Labor charges cannot lower than before”
    - Else:

```
UPDATE Repair
SET
complete_date = '$complete_date', labor_charge = '$labor_charge'
WHERE vin = '$vin' and start_date = '$start_date';
```

  - Else if USER_TYPE = 'Owner'

```
UPDATE Repair
SET
complete_date = '$complete_date', labor_charge = '$labor_charge'
WHERE vin = '$vin' and start_date = '$start_date';
```

- User clicks ***Cancel*** button:
  - Go back to **View Repair** task

Add Parts

Abstract Code

- User enters *quantity*('$quantity'), *vendor name*('$vendor_name'), *part number*('$part_number'), *price*('$price') in the input field on **Part Form**
- User clicks ***Add*** button:
  - If data validation failed, then display an error message: "Data verification failed"
  - Else:

```
INSERT INTO
Part (vin, start_date, part_number, vendor_name, price, quantity)
VALUES
('$vin', '$start_date', '$part_number', '$vendor_name', '$price', '$quantity');
```

```
SELECT labor_charge
FROM Repair
WHERE vin = '$vin' and start_date = '$start_date';
```

```
UPDATE Repair
SET
larbor_charge = '$labor_charge' + '$price'
WHERE vin = '$vin' and start_date = '$start_date';
```

- User clicks ***Cancel*** button:
  - Go back to **View Repair task**

View Sales by Color

Abstract Code

- First get all the sold vehicles vins:
  - '$sold_vins' =

```
SELECT vin, purchase_date
FROM Sale;
```

- Filter out the vehicles that have multiple colors:
  - '$vins' =

```
SELECT vin, count(color) as cnt
FROM VehicleColor
WHERE vin IN  '$sold_vins'
GROUP BY vin;
```

  - vins = vins.filter(vin => vin.cnt == 1)
- Get their count by color respectively:
  - result =

```
SELECT color, count(color) as cnt
FROM VehicleColor
WHERE vin in '$vins'
GROUP BY color;
```

- Get the count based on purchase_date:
  - past30Days = result.fitler(purchase_date >= currentDate - 30)
  - pastYear = result.filter(purchase_date >= currentDate - 365)
  - allTime = result
- Get other colors that does not have any sell:
  - allColors =

```
SELECT color
FROM Color;
```

    - merge colors that does not have any sell to our past30Days, pastYear, allTime list
- Display the result
- If user clicks ***Cancel*** button:
    - Go back to **Navigation Panel**

View Sales by Type

Abstract Code

- First get all the sold vehicles vins:
    - '$sold_vins' =

```
SELECT vin, purchase_date
FROM Sale;
```

- Get the number of each vehicle type based on '$sold_vins':
    - result =

```
SELECT vehicle_type, count(vin)
FROM Vehicle
WHERE vin
IN '$sold_vins'
GROUP BY vehicle_type;
```

- Get the count by past30Days, pastYear, allTime respectively:
    - past30Days = result.filter(purchase_date >= currentDate - 30)
    - pastYear = result.filter(purchase_date >= currentDate - 365)
    - allTime = result
- If a vehicle type does not have any sell, the report should 0 at that column:
    - iterate through past30Days, pastYear and allTime list
    - if any of [“car”, “convertible”, “trunk”, “vanMinivan ”, “suv”] didn’t show, add 0 to it
- Display the result
- If user clicks ***Cancel*** button:
    - Go back to **Navigation Panel**

View Sales by Manufacturer

Abstract Code

- First Get all the sold vehicles vins:
    - '$sold_vins' =

```
SELECT vin, purchase_date
FROM Sale;
```

- Get the number of each vehicle type based on '$sold_vins':
    - result =

```
SELECT manufacturer, count(vin)
FROM Vehicle
WHERE vin
IN '$sold_vins'
GROUP BY manufacturer;
```

- Get the count by past30Days, pastYear, allTime respectively:
    - past30Days = result.filter(purchase_date >= currentDate - 30)
    - pastYear = result.filter(purchase_date >= currentDate - 365)
    - allTime = result
- Display the result
- If user clicks ***Cancel*** button:
    - Go back to **Navigation Panel**

View Gross Customer Income

Abstract Code

```
SELECT
customer_id,
sum(income) as gross_income,
min(date) as first_date,
max(date) as recent_date
FROM
((SELECT
t2.id as customer_id,
t1.purchase_date as date,
t1.sold_price as income
FROM Sale as t1
LEFT JOIN Customer as t2
ON t1.customer_id = t2.id) as t6
union
(SELECT
t2.id as customer_id,
t1.start_date as date,
t1.labor_charge+t3.parts_cost as income
FROM Repair as t3
LEFT JOIN Customer as t4
ON t3.customer_id = t2.id
) as t7
GROUP BY customer_id
ORDER BY gross_income DESC, last_date DESC
LIMIT 15;
```

- The result is stored in the variable '$gross_customer_income'. Then uses the list of customer_id '$customer_list' in SQL:

```
SELECT customer_id,
count(1) as number_of_sales
FROM Sales
GROUP BY customer_id
HAVING customer_id
IN '$customer_list';
```

  -

```
SELECT customer_id,
count(1) as number_of_pairs
FROM Repair
GROUP BY customer_id
HAVING customer_id
IN '$customer_list';
```

  -

```
SELECT customer_id,
concat(first_name, " ", last_name) as name
FROM Individual
WHERE customer_id
IN  '$customer_list';
```

  -

```
SELECT customer_id,
business_name as name
FROM Business
WHERE customer_id
```

```
IN  '$customer_list';
```

- Merge the results together on customer_id, then display customer_id, name, first_date, recent_date, number_of_sales, number_of_repair, gross_income
- If user clicks ***More details*** button, then variable '$customer_id' containing the customer_id next to the button is passed to SQL:

```
SELECT t1.purchase_date as sale_date,
t1.sold_price as sold_price, t1.vin as VIN,
t2.model_year as year,
t2.manufacturer as manufacturer,
t2.model_name as model,
concat(t3.first_name, " ", t3.last_name) as salesperson_name
FROM Sales as t1
INNER JOIN Vehicle t2
ON t1.VIN = t2.VIN
INNER JOIN User t3
ON t1.salespeople_username = t3.username
WHERE t1.customer_id = '$customer_id'
ORDER BY sale_date DESC, VIN ASC;
```

  - Display the result in the **vehicle sales section**

```
SELECT t1.start_date as start_date,
t1.complete_date as end_date,
t1.vin as VIN,
t1.odometer as odometer_reading,
t1.parts_cost  as parts_cost,
t1.larbor_charge as larbor_cost,
t1.parts_cost + t1.larbor_charge as total_cost,
concat(t2.first_name, " ", t2.last_name) as service_writer_name
FROM Repair t1
INNER JOIN User t2
ON t1.service_writer_username = t2.username
WHERE t1.customer_id =  '$customer_id'
ORDER BY start_date DESC, end_date DESC NULLS FIRST, VIN ASC;
```

  - Display the result in the **repairs section**
- If user clicks ***Cancel*** button:
  - Go back to **Navigation Panel**

View Repair Report

Abstract Code

```
SELECT t3.name as manufacturer,
count(1) as count_of_repairs,
sum(t1.parts_cost) as sum_parts_cost,
sum(t1.labor_charge) as sum_labor_cost,
sum(t1.parts_cost)+sum(t1.labor_charge) as sum_repair_cost
FROM Repair t1
INNER JOIN Vehicle t2
ON t1.vin = t2.vin
RIGHT JOIN Manufacturer t3
ON t2.manufacturer = t3.name
GROUP BY manufacturer
ORDER BY manufacturer ASC;
```

- The result is displayed in the first part
- If user clicks ***More details*** button next to a manufacturer, then variable '$manufacturer' is passed to SQL:

```
SELECT t2.vehicle_type as vehicle_type,
```

```
count(1) as number_of_repairs,
sum(t1.parts_cost) as parts_cost,
sum(t1.labor_charge) as labor_cost,
sum(t1.parts_cost)+sum(t1.labor_charge) as total_cost
FROM Repair t1
INNER JOIN Vehicle t2
ON t1.vin = t2.vin
GROUP BY vehicle_type
HAVING t2.manufacturer = '$manufacturer'
ORDER BY number_of_repairs DESC;
```

```
SELECT t2.vehicle_type as vehicle_type,
t2.model_name as model,
count(1) as number_of_repairs,
sum(t1.parts_cost) as parts_cost,
sum(t1.labor_charge) as labor_cost,
sum(t1.parts_cost)+sum(t1.labor_charge) as total_cost
FROM Repair t1
INNER JOIN Vehicle t2
ON t1.vin = t2.vin
GROUP BY vehicle_type, model
HAVING t2.manufacturer = '$manufacturer'
ORDER BY number_of_repairs DESC;
```

- The drill_down list displays the first result followed by the corresponding models statistics from the second result

- If user clicks ***Cancel*** button:
  - Go back to **Navigation Panel**

View Below Cost Sales

Abstract Code

```
SELECT t1.vin as VIN,
t2.purchase_date as date,
t1.invoice_price as invoice_price,
t2.sold_price as sold_price,
CONCAT(ROUND(t2.sold_price/t1.invoice_price*100,2),'%') as price_ratio,
concat(ifnull(t6.business_name,''), ifnull( t5.first_name,''), " ", ifnull(t5.last_name,'')) as customer_name,
concat(t3.first_name, " ", t3.last_name) as salesperson_name
FROM Vehicle t1
INNER JOIN Sales t2
ON t1.vin = t2.vin and t1.invoice_price > t2.sold_price
INNER JOIN User t3
ON t2.salespeople_username = t3.username
INNER JOIN Customer t4
ON t2.customer_id = t4.id
LEFT JOIN Individual t5
ON t4.id = t5.customer_id
LEFT JOIN Business t6
ON t4.id = t6.customer_id
ORDER BY date DESC, price_ratio DESC;
```

- Display the result with rows whose price_ratio is not above 95% highlighted to red
- If user clicks ***Cancel*** button:
  - Go back to **Navigation Panel**

View Average Time in Inventory

Abstract Code

```
SELECT t1.vehicle_type as vehicle_type,
avg(datediff(Day, t2.purchase_date, t1.added_date)) as average_time
FROM Vehicle t1
INNER JOIN Sale t2
ON t1.vin = t2.vin
GROUP BY vehicle_type;
```

- The result is displayed in the report, with "N/A" displayed for vehicle types with no sales history
- If user clicks ***Cancel*** button:
    - Go back to **Navigation Panel**

View Parts Statistics

Abstract Code

```
SELECT vender_name, sum(quantity) as number_of_parts, sum(price*quantity) as total_spent
FROM Part
GROUP BY vendor_name;
```

- The result is displayed in the report
- If user clicks ***Cancel button***:
    - Go back to **Navigation Panel**

View Monthly Sales

Abstract Code

```
SELECT year(t1.purchase_date) as year,
month(t1.purchase_date) as month,
count(1) as number_of_vehicles,
sum(t1.sold_price) as total_sales_income,
sum(t1.sold_price-t2.invoice_price) as total_net_income,
CONCAT(ROUND(sum(t1.sold_price)/sum(t2.invoice_price)*100,2),'%') as price_ratio
FROM Sale t1
INNER JOIN Vehicle t2
ON t1.vin = t2.vin
ORDER BY year DESC, month DESC;
```

- The result is displayed on the **summary page**
- If user clicks ***Top Salesperson*** button next to a month, then:
    - Variables '$year' and '$month' are passed to SQL:

```
SELECT concat(t1.first_name, " ", t1.last_name) as name,
count(1) as vehicles_sold,
sum(t1.sold_price) as total_sales
FROM Sale t1
INNER JOIN User t2
ON t1.salespeople_username = t2.username
WHERE year(t1.purchase_date) = '$year' and month(t1.purchase_date) = '$month'
GROUP BY t1.salespeople_username
ORDER BY vehicles_sold DESC, total_sales DESC
LIMIT 1;
```

- The result is displayed on the drill down report
- If user clicks ***Cancel*** button:

- Go back to **Navigation Panel**