

## Search Vehicle

### Abstract Code

View total number of cars available for purchase;

```
SELECT count(*) AS countOfCars FROM Vehicle V WHERE NOT EXIST (SELECT VIN FROM Buy as B WHERE V.VIN=B.VIN);
```

If user enters a keyword:

find entries that match the keyword and display details;

```
SELECT * FROM Vehicle AS V, Vehicle-Color AS VC, VehicleType AS VT, Manufacturer as M WHERE V.VIN=VC.VIN AND V.TypeId=VT.TypeId AND M.ManufactureId=V.ManufactureId AND (V.ModelName LIKE '%$keyword%' OR M.ManufacturerName LIKE '%$keyword%' OR V.ModelYear LIKE '%$keyword%' OR V.Description LIKE '%$keyword%') AND V.VIN NOT IN (SELECT VIN FROM Buy;
```

if no entries match:

display error

else

Populate search criteria drop downs.

While no options are selected do nothing

If option is selected then **list vehicles (use List Vehicle)** that match selected criteria:

Display VIN, vehicle type, model year, manufacturer, model, colours, mileage and sales price of all matched vehicles in search result

o If option = VehicleType

```
SELECT * FROM Vehicle AS V, Vehicle-Color AS VC, VehicleType AS VT, Manufacturer as M WHERE V.VIN=VC.VIN AND V.TypeId=VT.TypeId AND M.ManufactureId=V.ManufactureId AND VT.VehicleType='$option';
```

o If option = Manufacturer

```
SELECT * FROM Vehicle AS V, Vehicle-Color AS VC, VehicleType AS VT, Manufacturer as M WHERE V.VIN=VC.VIN AND V.TypeId=VT.TypeId AND M.ManufactureId=V.ManufactureId AND M.ManufacturerName ='$option';
```

o If option = ModelYear

```
SELECT * FROM Vehicle AS V, Vehicle-Color AS VC, VehicleType AS VT, Manufacturer as M WHERE V.VIN=VC.VIN AND V.TypeId=VT.TypeId AND
```

## Phase 2 Report | CS6400 – Fall 2019 | Team 60

```
M.ManufactureId=V.ManufactureId AND V.ModelYear
='$option';
```

- o If option = Color

```
SELECT * FROM Vehicle AS V, Vehicle-Color AS VC,
VehicleType AS VT, Manufacturer as M WHERE
V.VIN=VC.VIN AND V.TypeId=VT.TypeId AND
M.ManufactureId=V.ManufactureId AND
VC.Color='$option';
```

If a car is selected from the list :

Display VIN, vehicle type, model year, manufacturer, model name, colors, mileage, sales price and description of car (Use **view vehicle details**)

### Lookup Customer

#### Abstract Code

Privileged user clicked on **Sell Car** button on **Public Search Page**.

This loads **Sales Order** form.

Upon clicking **Lookup Customer** button on **Sales Order** form

- o Search for the customer that matches given uniqueID

```
SELECT CustomerID FROM Individual as I, Business as B
WHERE (B.TaxIDNumber='$uniqueID' OR
I.DriverLicenseNumber='$uniqueID');
```

- o If the customer is found:  
Select the customer

- o Else:  
Add a new customer (Use **Add Customer**)

Click on **Save** button to save the changes.

### Update Sales Date

#### Abstract Code

On **Sales Order** form, update customer details (Use **Lookup Customer**).

Add Sales Date on **Sales Order** form.

Click on **Save** button to save the changes.

```
UPDATE Sell SET SalesDate=GETDATE() WHERE Sell.CustomerId='$customerId';
```

### Update Loan Details

#### Abstract Code

## Phase 2 Report | CS6400 – Fall 2019 | Team 60

On **Sales Order** form, check if customer has applied for loan

If customer has applied for loan:

Enter loan details on **Sales Order** form. Enter Loan Term, Down Payment, Interest Rate, Loan ID, Monthly Payment and Start Month.

Else:

Do nothing

Click on **Save** button to save the changes.

```
UPDATE Loan SET LoanTerm='$loanterm', DownPayment= '$downpayment',  
InterestRate= '$InterestRate', MonthlyPayment= '$monthlypayment', StartMonth=  
'$startmonth' WHERE VIN=(SELECT VIN FROM Buy WHERE  
CustomerID='$customerId');
```

### View Vehicle Details

#### Abstract Code

After login when user searches for vehicle by using either of vehicle type, manufacturer, model year, color or any keyword.

if the vehicle matches to any of the vehicle's field **and parts**

List out all the vehicles matching the search criteria used by a user in ascending order of VINs.

Display following attributes for each of the vehicle:

VIN, vehicle type, model year, manufacturer, model, color(s), Mileage, sales price.

```
SELECT vin, vehicle_type, model_year, manufacturer_name, milage,  
cost FROM `Vehicle` WHERE vin='$InputVIN' OR  
vehicle_type='$InputVehicleType' OR model_year='$InputModelYear' OR  
manufacturer_name='$InputManufacturerName' OR milage='$InputMilage'  
OR cost='$InputCost'  
ORDER BY vin ASC;
```

Provide user an option for sorted list of vehicles as per users' choice of attribute.

If user selects any specific vehicle result:

Open details page for of the vehicle

Under the details page, display following vehicle details:

VIN, vehicle type, model year, model name, manufacturer, color(s), mileage, sales price, description of the car.

```
SELECT vin, vehicle_type, model_year, manufacturer_name, milage,  
cost, description FROM `Vehicle` WHERE vin='$SelectedVIN'
```

### List Parts

#### Abstract Code

Show parts details of a vehicle for an Inventory clerk in view vehicle details page.

If the car is newly added:

Show \$0 total for parts.

If user clicks on Parts section in the vehicle details page:

For each part show details like part number, description, vendor, purchase order, cost and status.

```
SELECT partNumber, cost, description, vendorName, FROM  
'Vehicle' WHERE Vehicle.vin='$SelectedVIN'
```

### Update Part Status

#### Abstract Code

If user clicks onto view parts section on view vehicle details page.

For each part show details like part number, description, vendor, purchase order, cost and status.

```
Select partNumber, description, c.vendorName, c.orderNumber, cost, c.part_status  
from Part as p INNER JOIN Contains as c ON p.partNumber=c.partNumber  
INNER JOIN PartOrder as po on c.orderNumber = po.orderNumber  
where po.vin = '$SelectedVIN'
```

If user clicks on edit status button:

User should be able to edit the status and save it.

```
Update Contains  
Set part_status = '$partStatus'  
where orderNumber = '$orderNumber' and partNumber = '$partNumber'
```

## Login

### Abstract Code

- User enters username ('username'), password ('password') input fields.
  - If data validation is successful for both *username* and *password* input fields, then
    - When **Enter** button is clicked:

```
SELECT username, password FROM PrivilegedUser
WHERE username='$currentUser';
```

- If User record is found but user.password != "password":
      - Go back to **Login** form, with error message.
    - Else:
      - Store login information as session variable 'username'
      - Go to **Main Page**
  - Else email and password input fields are invalid, display **Login** form, with error message.

## Add Vehicle

### Abstract Code

- Privileged user will click on the add vehicle button.
  - If the *session* for the privileged user is live:
    - Fill in the vehicle details in the add vehicle form.
    - Look up for the customer to be linked to the vehicle.
    - If the customer is found in the lookup :
      - Select the customer.

```
SELECT email_address from CUSTOMER where
email=$email;
```

Else:

## Phase 2 Report | CS6400 – Fall 2019 | Team 60

- Fill all the details for the customer.

```
INSERT into CUSTOMER (email_address, phone_number,  
street, postalcode, city, state) VALUES ( $emailaddress,  
$phonenumber, $street, $postalcode, $city, $state )
```

- Look up the customer list to select the newly added customer.
- Add the date of purchase
- The date of purchase will be validated if the date is not from the future.
- click on the submit button

```
INSERT into VEHICLE (vin, vehicletype, modelName,  
modelYear, manufactureid, mileage, description, cost)  
VALUES ($VehicleNumber, $vehicletype, $modelName,  
$modelYear, $manufactureid, $mileage, $description, $cost)
```

```
INSERT into VEHICLE-COLOR (vin, color) VALUES  
($vin, $color);
```

- Else:
  - • Take the privileged user to the **Login Page**.

## List Vehicle

### Abstract Code

Customer will view the **Public Page**

Customer will select a filter out of the valid filters from a drop down

If there any vehicles which match the filter

```
SELECT * FROM VEHICLE WHERE column_name = <filter>;
```

Else

Print an appropriate *error* message.

## View Monthly Sales

### Abstract Code

- On Display Reports page, when *View Monthly Sales is clicked*
- Validate if user has access to reports
- Initialise Monthly\_Sales\_Report and Monthly\_Sales\_DrillDown\_Report
- Find all the vehicles which have a non empty SalesDate column

```
SELECT vin, vehicle_type, model_year,model_name,manufacturer,  
mileage, sales_price, description, s.sales_date, s.sales  
FROM Vehicle as v, Sell as s  
Where v.vin = s.vin
```

- Group all these vehicles by year and then by month and sort in reverse chronological order
- Initialise total\_sales\_income, total\_vehicles, total\_net\_income for every month, salespeople\_list
  - For every vehicle in every month
  - Find the PurchasePrice, SalesPrice and sum of SumOfPartsCost from all parts orders for this vehicle
  - add SalesPrice to total\_sales\_income

```
Select SUM(order_cost) as total_parts_costs
From PartOrder as po , (Select SUM(p.cost) as order_cost, c.orderNumber
                        From Contains as c, Part as p
                        Where c.partNumber = p.partNumber
                        GROUP BY c.orderNumber ) as order_costs
Where po.orderNumber = order_costs.orderNumber and po.vin =
'$SelectedVIN'
```

- calculate net income using SalesPrice - (PurchasePrice + sum of SumOfPartsCost)
- add net\_income to total\_net\_income
- Find the corresponding SalesPerson and add id to salespeople\_list
- Display total\_sales\_income, total\_vehicles, total\_net\_income and add to Monthly\_Sales\_Report
- For every month
- Find details of PrivilegedUser using ids from salespeople\_list
- From the vehicles table, group them using no\_of\_vehicles sold and by the total\_sales of vehicles
- For every user in salespeople\_list
- Display FirstName, LastName, no\_of\_vehicles and total\_sales and update Monthly\_Sales\_DrillDown\_Report
- Sort by no\_of\_vehicles descending and update Monthly\_Sales\_DrillDown\_Report
- Sort total\_sales descending and update Monthly\_Sales\_DrillDown\_Report
- Display Monthly\_Sales\_DrillDown\_Report
- Display Monthly\_Sales\_Report

### View Vehicle Details

#### Abstract code

- After login when user searches for vehicle by using either of vehicle type, manufacturer, model year, color or any keyword.
  - If the vehicle matches to any of the vehicle's field:



## Phase 2 Report | CS6400 – Fall 2019 | Team 60

- List out all the vehicles matching the search criteria used by a user in ascending order of vin

```
SELECT vin, vehicle_type, model_year,model_name,manufacturer,
mileage, sales_price, description, vc.color
FROM Vehicle as v, Vehicle-Color as vc
WHERE v.vin=vc.vin and v.column_name = <filter>
ORDER BY vin
```

- Display following attributes for each of the vehicle:
- Provide user an option for sorted list of vehicle as per users choice of attribute.
- If user selects any specific vehicle result:
- Open details page for one of the vehicle
- Under the details page, display following vehicle details:
- VIN, vehicle type, model year, model name, manufacturer, color(s), mileage, sales price, description of the car.

### View Vehicle Time in Inventory

#### Abstract Code

- On **Display Reports** page, when *View Vehicle Time In Inventory* is clicked
- Validate if user has access to reports
- Find all the vehicles in the inventory -> vehicle\_list

```
SELECT vehicle_type, AVG(vehicle_time.vt)
FROM Vehicle as v, (select vin, IF (s.sales_date =null,TIMEDIFF(
GETDATE(),b.purchase_date), TIMEDIFF(s.sales_date, b.purchase_date)
as vt) from buy as b,sell as s where b.vin=s.vin) as vehicle_time
where v.vin = vehicle_time.vin
GROUP BY vehicle_type
```

- Group all the vehicles by vehicle type - > vehicle\_list\_by\_type
  - For every vehicle\_type\_group in vehicle\_list\_by\_type
  - initialise total\_time = 0

## Phase 2 Report | CS6400 – Fall 2019 | Team 60

- initialise total\_vehicle\_count\_in\_type = size of vehicle\_type\_list
- If total\_vehicle\_count\_in\_type is 0 then Display “N/A”
- Else

For every vehicle in vehicle\_type\_group

- Calculate time spent in inventory time\_in\_inventory using purchaseDate and sellDate and update total\_time
- Calculate average\_time per vehicle\_type using total\_time and total\_vehicle\_count\_in\_type
- Display average\_time per vehicle\_type

### View Parts Statistics

#### Abstract Code

- On **Display Reports** page, when *View Parts Statistics* is clicked
- Validate if user has access to reports
- Initialise **view\_parts\_statistics** report.
- Create a report with zero rows and three columns vendor, total\_noOf\_parts and total\_dollars\_spent.

```
SELECT po.vendorName, sum(parts_count.cost) as total_cost,
count(parts_count.count) as total_count
FROM PartOrder as po, (Select count(*) as count, sum(cost) as order_cost,
c.orderNumber from Part as p, Contains as c Where
p.partNumber=c.partNumber and p.status='Installed' groupby
c.orderNumber) as parts_cost
Where po.orderNumber=parts_cost.orderNumber
GROUP BY po.vendorName
```

- Find all the vehicles in the inventory -> vehicle\_list
  - For every vehicle in the vehicle\_list
  - For every part order
  - For every part in the order
  - Find the corresponding vendor

## Phase 2 Report | CS6400 – Fall 2019 | Team 60

- Add or update the cost of part to value of column `total_dollars_spent` for the vendor
- Update the column `total_noOf_parts` by 1 for the vendor
- Display the **view\_parts\_statistics** report.

### View Seller History

#### Abstract Code

- On **Display Reports Page** when *View Seller History* is clicked
- Validate if user has access to reports
- Initialise a **seller\_history** report with zero rows and five columns `seller_detail`, `total_vehicles`, `average_purchase_price`, `average_parts` and `average_parts_cost`
  - For every customer
  - If he has sold at least one vehicle
  - Create a display row -> `seller_row`
  - Add first column to `seller_row` first name and last name of customer or company name in case customer is business
  - Find list of all the vehicles sold by them as `vehicle_list` and
  - Add second column to `seller_row` **total\_vehicles** -> size of `vehicle_list`

```
Select username, pu.first_name, pu.last_name, p.cost,
from PrivilegedUser as pu INNER JOIN Vehicle as v
  ON pu.username = v.username
  INNER JOIN PartOrder as po
  ON v.username, po.username
  INNER JOIN Contains as c
  ON po.orderNumber=c.orderNumber
  INNER JOIN Part as p
  ON c.orderNumber = p.orderNumber
```

- Calculate the `average_purchase_price` for all those vehicles and add third column to `seller_row` **average\_purchase\_price**

```
Select username, AVG(cost) as avg_purchase_price
from PrivilegedUser as pu INNER JOIN Vehicle as v
ON pu.username = v.username
GROUP BY username
```

- Find all the parts order
- Calculate **average\_parts** and **average\_parts\_cost**
- For every vehicle in the vehicle\_list
- Add fourth column to seller\_row **average\_parts**
- Add fifth column to seller\_row **average\_parts\_cost**
- If average\_parts\_cost > \$500:
- Highlight the row background as red
- Add seller\_row to the report
- Sort all rows by **total\_vehicles** descending
- Sort all rows by **average\_purchase\_price** ascending
- Display the **seller\_history** report.

### View Monthly Loan Income

#### Abstract Code

- On **Display Reports** page, when *View Monthly Loan* is clicked
- Validate if user has access to reports
- Initialise a **view\_monthly\_loan** report
- Initialise start month as start\_month = current\_month - 1 - 12
- Initialise end month as end\_month = current\_month - 1
  - For current\_month in range of start\_month to end\_month
  - initialise total\_monthly\_payment = 0
  - Find all the vehicles with sellDate in current\_month, find date and year
  - For every vehicle find the corresponding loan if exists
  - If startMonth of loan is one less than current month then
  - add MonthlyPayment to total\_monthly\_payment.
- Calculate 1% of total\_monthly\_payment -> Mr Burdell's share.
- Display year, current\_month, total\_monthly\_payment, Mr Burdell's share.

## Phase 2 Report | CS6400 – Fall 2019 | Team 60

- Display **view\_monthly\_loan** report.

```
Select Sum(monthly_payment)
from Vehicle as v , Loan as l
where v.vin=l.vin and GETDATE()-l.start_month>30
```