DB6400 Project

Below are sql queries in each php file(totally 20 files) with the file name and description:

1. main.php (the public/login user search page)

```
<!-- get vehile types -->
   $query = WITH VINType (VIN, type) AS
   (SELECT VIN, type FROM car
   UNION SELECT VIN, type FROM truck
   UNION SELECT VIN, type FROM convertible
   UNION SELECT VIN, type FROM suv
   UNION SELECT VIN, type FROM Van)
   SELECT DISTINCT type
   FROM vehicle AS V INNER JOIN VINType ON V.VIN = VINType.VIN
   WHERE CASE WHEN '{$role}' != '{$CONST_manager}' and '{$role}' != '{$CONST_Owner}' THEN is_sold = 0 ELSE 0 = 0 END;

<!-- get manufacturers -->
   $query = SELECT DISTINCT M.name as m_name
   FROM vehicle AS V INNER JOIN manufacturer AS M ON V.mID = M.mID
   WHERE CASE WHEN '{$role}' != '{$CONST_manager}' and '{$role}' != '{$CONST_Owner}' THEN is_sold = 0 ELSE 0 = 0 END;

<!-- get model years -->
   $query = SELECT DISTINCT model_year
   FROM vehicle
   WHERE CASE WHEN '{$role}' != '{$CONST_manager}' and '{$role}' != '{$CONST_Owner}' THEN is_sold = 0 ELSE 0 = 0 END
   ORDER BY model_year;

<!-- get all colors -->
   $query = SELECT DISTINCT color
   FROM vehicle as V INNER JOIN vehiclecolor AS VC ON V.VIN = VC.VIN
   WHERE CASE WHEN '{$role}' != '{$CONST_manager}' or '{$role}' != '{$CONST_Owner}' THEN is_sold = 0 ELSE 0 = 0 END
   ORDER BY color;

          // search vehicle
      $query = WITH VINType (VIN, vehicle_type) AS
      (SELECT VIN, type FROM car
      UNION SELECT VIN, type FROM truck
      UNION SELECT VIN, type FROM convertible
      UNION SELECT VIN, type FROM suv
```

```
       UNION SELECT VIN, type FROM Van),
        vehicleColorList (VIN, colorlist) AS
       (SELECT V.VIN, GROUP_CONCAT(color) FROM vehicle AS V INNER JOIN vehiclecolor AS VC
on V.VIN = VC.VIN
       GROUP BY V.VIN)
        SELECT V.VIN, model_name, model_year, invoice_price, invoice_price * 1.25 as list_price,
vehicle_type, M.name as manufacturer_name, VCL.colorlist as color, description
         FROM vehicle AS V
         INNER JOIN VINType AS VT ON V.VIN = VT.VIN
         INNER JOIN Manufacturer AS M on V.mID = M.mID
         INNER JOIN VehicleColorList AS VCL on V.VIN = VCL.VIN
         WHERE
         CASE WHEN '{$VIN}' != '' THEN V.VIN = '{$VIN}' ELSE 1 = 1 END
         AND
         CASE WHEN '{$vehicle_type}' != '' THEN vehicle_type = '{$vehicle_type}' ELSE 1 = 1 END
         AND
         CASE WHEN '{$manufacturer}' != '' THEN M.name = '{$manufacturer}' ELSE 1 = 1 END
         AND
         CASE WHEN '{$model_year}' != '' THEN model_year = '{$model_year}' ELSE 1 = 1 END
         AND
         CASE WHEN '{$color}' != '' THEN colorlist LIKE '%{$color}%' ELSE 1 = 1 END
         AND
         CASE WHEN '{$min_list_price}' != '' THEN invoice_price * 1.25 >= '{$min_list_price}' ELSE 1
= 1 END
         AND
         CASE WHEN '{$max_list_price}' != '' THEN invoice_price * 1.25 <= '{$max_list_price}' ELSE
1 = 1 END
         AND
         CASE WHEN '{$keyword}' != '' THEN (M.name LIKE BINARY '%{$keyword}%' OR
model_year LIKE BINARY '%{$keyword}%' OR model_name LIKE BINARY '%{$keyword}%' OR
`description` LIKE BINARY '%{$keyword}%') ELSE 1 = 1 END
         AND
         CASE WHEN '{$is_sold}' != '2' THEN is_sold = '{$is_sold}' ELSE 1 = 1 END
         ORDER BY VIN ASC;
```

2.view_vehicle_detail.php (view vehicle details)

```
          //get vehicle type
       $query_type = WITH VINType (VIN, vehicle_type) AS
       (SELECT VIN, type FROM car
       UNION SELECT VIN, type FROM truck
       UNION SELECT VIN, type FROM convertible
       UNION SELECT VIN, type FROM suv
```

```
    UNION SELECT VIN, type FROM Van)
      SELECT vehicle_type FROM VINType AS VT WHERE VT.VIN = '{$VIN}';

    //get vehicle color
    $query_color = WITH VehicleColor (VIN, color) AS
    (SELECT V.VIN, GROUP_CONCAT(color) FROM vehicle AS V INNER JOIN vehiclecolor AS VC on V.VIN = VC.VIN
    GROUP BY V.VIN);

      // get vehicle details
      if login user is inventory clerk, manager or owner:
      show invoice price

      if vehicle_type is Car:
     $query = SELECT is_sold, color, V.VIN, door_num, model_year, model_name, M.name as manufacturer_name, invoice_price, invoice_price * 1.25 as list_price, description
      FROM vehicle AS V INNER JOIN manufacturer AS M on V.mID = M.mID
      INNER JOIN car ON V.VIN = car.VIN
      INNER JOIN VehicleColor as VC ON V.VIN = VC.VIN
      WHERE V.VIN = '{$VIN}';

    if vehicle_type is Truck:
     $query = SELECT is_sold, color, V.VIN, cargo_capacity, cargocover_type, rear_axle_num, model_year, model_name, M.name as manufacturer_name, invoice_price, invoice_price * 1.25 as list_price, description
      FROM vehicle AS V INNER JOIN manufacturer AS M on V.mID = M.mID
      INNER JOIN Truck ON V.VIN = Truck.VIN
      INNER JOIN VehicleColor as VC ON V.VIN = VC.VIN
      WHERE V.VIN = '{$VIN}';

    if vehicle_type is Convertible:
     $query = SELECT is_sold, color, V.VIN, roof_type, backseat_num, model_year, model_name, M.name as manufacturer_name, invoice_price, invoice_price * 1.25 as list_price, description
      FROM vehicle AS V INNER JOIN manufacturer AS M on V.mID = M.mID
      INNER JOIN Convertible ON V.VIN = Convertible.VIN
      INNER JOIN VehicleColor as VC ON V.VIN = VC.VIN
      WHERE V.VIN = '{$VIN}';

    if vehicle_type is Van:
     $query = SELECT is_sold, color, V.VIN, driverside_backdoor, model_year, model_name, M.name as manufacturer_name, invoice_price, invoice_price * 1.25 as list_price, description
      FROM vehicle AS V INNER JOIN manufacturer AS M on V.mID = M.mID
      INNER JOIN Van ON V.VIN = Van.VIN
```

```
        INNER JOIN VehicleColor as VC ON V.VIN = VC.VIN
        WHERE V.VIN = '{$VIN}';

    if vehicle_type is SUV:
      $query = SELECT is_sold, color, V.VIN, drivetrain_type, cupholder_num, model_year,
model_name, M.name as manufacturer_name,        invoice_price, invoice_price * 1.25 as
list_price, description
        FROM vehicle AS V INNER JOIN manufacturer AS M on V.mID = M.mID
        INNER JOIN SUV ON V.VIN = SUV.VIN
        INNER JOIN VehicleColor as VC ON V.VIN = VC.VIN
        WHERE V.VIN = '{$VIN}';

        // view the date added to inventory and who added it for manager/owner
        $query = SELECT add_date, first_name, last_name FROM vehicle as V INNER JOIN
loginuser AS LU ON V.username = LU.username;

        // view vehicle sale by manager/owner
                if customerType is Individual:
          $query = SELECT 1.25 * invoice_price as list_price, sold_price, sold_date,
CONCAT(LU.first_name, ' ', LU.last_name) as salesperson_name,
          CONCAT(I.first_name, ' ', I.last_name) as customer_name, email, phone,
street_address, city, state, postal_code
          FROM sale AS S INNER JOIN loginuser AS LU ON S.username = LU.username
          INNER JOIN vehicle as V ON S.VIN = V.VIN
          INNER JOIN customer AS C ON S.customerID = C.customerID
          INNER JOIN individual as I ON S.customerID = I.customerID
          WHERE S.VIN = '{$VIN}';

                if customerType is Business:
                $query = SELECT 1.25 * invoice_price as list_price, sold_price, sold_date,
CONCAT(LU.first_name, ' ', LU.last_name) as salesperson_name,
          business_name, title, CONCAT(B.contact_first_name, ' ', B.contact_last_name) as
contact_name, email, phone, street_address, city, state, postal_code
          FROM sale AS S INNER JOIN loginuser AS LU ON S.username = LU.username
          INNER JOIN vehicle as V ON S.VIN = V.VIN
          INNER JOIN customer AS C ON S.customerID = C.customerID
          INNER JOIN business as B ON S.customerID = B.customerID
          WHERE S.VIN = '{$VIN}';

        // view vehicle repair by manager/owner
           $query_customerName = WITH CustomerName (customerID, customer_name) AS
       (SELECT customerID, CONCAT(first_name, ' ', last_name) as name FROM individual
       UNION
       SELECT customerID, business_name as name FROM business)
```

```
        $query_PartsCost = ,PartsCost (repairID, parts_cost) AS
        (SELECT repairID, sum(part_quantity * part_price) AS parts_cost
        FROM part
        GROUP BY repairID)

      $query_select = SELECT customer_name, CONCAT(LU.first_name, ' ', LU.last_name) AS
servicewriter_name,
       start_date, complete_date, IFNULL(labor_charge, 0) as labor_charges, IFNULL(parts_cost,
0) as part_charges,
       (IFNULL(labor_charge, 0) + IFNULL(parts_cost, 0)) AS total_charges
       FROM Repair as R
       INNER JOIN CustomerName as CN ON R.customerID = CN.customerID
       INNER JOIN loginuser as LU ON R.username = LU.username
       LEFT JOIN PartsCost as PC ON R.repairID = PC.repairID
       WHERE R.VIN = '{$VIN}';
```

3. add_vehicle.php (add vehicle into inventory)

```
        if vehicle type is car:
        $query_add_type = "INSERT INTO car VALUES ('{$VIN}', 'Car', {$door_num})";

        if vehicle type is convertible:
        $query_add_type = "INSERT INTO convertible VALUES ('{$VIN}', 'Convertible',
'{$roof_type}', {$backseat_num})";

        if vehicle type is truck:
        $query_add_type = "INSERT INTO truck VALUES ('{$VIN}', 'Truck', {$cargo_capacity},
'{$cargocover_type}', {$rear_axle_num})";

        if vehicle type is van:
        $query_add_type = "INSERT INTO Van VALUES ('{$VIN}', 'Van', {$driverside_backdoor})";

        if vehicle type is SUV:
        $query_add_type = "INSERT INTO suv VALUES ('{$VIN}', 'SUV', '{$drivetrain_type}',
{$cupholder_num})";

        // add vehicle
        $query = INSERT INTO vehicle VALUES ('{$VIN}', '{$model_name}', {$model_year},
{$invoice_price}, FALSE, '{$description}', curdate(),
          (SELECT mID FROM manufacturer WHERE name = '{$manufacturer}'),
          '{$username}');
```

4. sale_form.php (add sale transactions)

```
        $query = INSERT INTO sale (username, customerID, VIN, sold_date, sold_price) VALUES
('{$username}','{$customerID}', '{$VIN}','{$sold_date}', '{$sold_price}');
```

5. lookup_customer.php (search customer in database)

```
        if customer is individual:
        $query = SELECT email, phone, street_address, first_name, last_name, C.customerID
        FROM Customer AS C INNER JOIN Individual AS I ON C.customerID = I.customerID
        WHERE driver_license = '{$driver_license}’;

        if customer is business:
        $query = SELECT email, phone, street_address, business_name, contact_first_name,
contact_last_name, title, B.customerID
        FROM Customer AS C INNER JOIN Business AS B ON C. customerID = B.customerID
        WHERE taxID = '{$taxID}’;
```

6. add_customer.php (add customer into database)

```
        $query = INSERT INTO customer (email, phone, street_address, city, state, postal_code)
VALUES ('{$email}', '{$phone}', '{$street_address}', '{$city}', '{$state}', '{$postal_code}');

        if customer is individual:
        $query_add_customer = "INSERT INTO individual (driver_license, first_name, last_name,
customerID) VALUES ('{$driver_license}',
        '{$first_name}', '{$last_name}', (SELECT max(customerID) FROM customer))";

        if customer is business:
        $query_add_customer = "INSERT INTO business (taxID, business_name, title,
contact_first_name, contact_last_name,  customerID) VALUES
        ('{$taxID}', '{$business_name}', '{$title}', '{$contact_first_name}',
'{$contact_last_name}', (SELECT max(customerID) FROM customer))";
```

7. repair_form.php (check vehicle detail and get existing repair information)

```
        // check vehicle detail
        $query = WITH VINType (VIN, vehicle_type) AS
     (SELECT VIN, type FROM car
     UNION SELECT VIN, type FROM truck
     UNION SELECT VIN, type FROM convertible
     UNION SELECT VIN, type FROM suv
     UNION SELECT VIN, type FROM van),
     VehicleColorList (VIN, colorlist) AS
     (SELECT V.VIN, GROUP_CONCAT(color) FROM vehicle AS V INNER JOIN vehiclecolor AS VC
 on V.VIN = VC.VIN
     GROUP BY V.VIN)
     SELECT V.VIN, model_name, model_year, vehicle_type, M.name as manufacturer_name,
 VCL.colorlist as color
     FROM vehicle AS V
     INNER JOIN VINType AS VT ON V.VIN = VT.VIN
     INNER JOIN Manufacturer AS M on V.mID = M.mID
     INNER JOIN VehicleColorList AS VCL on V.VIN = VCL.VIN
     WHERE V.VIN = '{$VIN}' and is_sold = 1;

        // get all existing repairs
        $query_repair = WITH CustomerName (customerID, customer_name) AS
     (SELECT customerID, CONCAT(first_name, ' ', last_name) as name FROM individual
     UNION
     SELECT customerID, business_name as name FROM Business)

     SELECT R.VIN, customer_name, CONCAT(LU.first_name, ' ', LU.last_name) AS
 servicewriter_name, start_date, odometer, complete_date, labor_charge, description, partID,
 vendor, part_price, part_quantity
     FROM Repair as R
     INNER JOIN CustomerName as CN ON R.customerID = CN.customerID
     INNER JOIN Loginuser as LU ON R.username = LU.username
     LEFT JOIN Part as P ON R.repairID = P.repairID
     WHERE R.VIN = '{$VIN}'
     ORDER BY complete_date ASC;
```

8. add_repair.php (add repair)

```
      $query_insert = INSERT INTO Repair (VIN, username, customerID, odometer,
labor_charge, start_date, description, complete_date)
                        VALUES ('{$VIN}', '{$username}', {$customerID}, {$odometer},
{$laborCharge}, '{$startDate}', '{$description}',
      '{$completeDate}’);
```

9. add_part.php (add parts to repair)

```
      $query_insert = INSERT INTO Part (repairID, partID, vendor, part_price, part_quantity)
                VALUES ((SELECT repairID FROM Repair as R WHERE R.VIN = '{$VIN}' and
R.start_date = '{$startDate}'), and'{$partID}','{$vendor}', {$partPrice}, {$partQuantity});
```

10. edit_repair.php (edit labor charge)

```
       $query_new = "UPDATE Repair as R
   SET labor_charge = {$laborCharge}
   WHERE VIN = '{$VIN}' and R.start_date = '{$startDate}' ";
```

11. complete_repair.php (complete the repair)

```
       $query_new = "UPDATE Repair as R
  SET complete_date = '{$completeDate}'
  WHERE VIN = '{$VIN}' and R.start_date = '{$startDate}' ";
```

12. report_sales_by_color.php (view report of sale by color)

```
       $query = WITH VehicleColorCount (VIN, color, color_cnt) AS
  (SELECT VIN, color, count(color) over (partition by VIN)
  FROM VehicleColor)

  , VehicleTrueColor (VIN, true_color) AS
  (SELECT DISTINCT VIN,
      CASE WHEN color_cnt > 1 THEN 'Multiple'
      ELSE color
      END AS true_color
  FROM VehicleColorCount)

  , SaleColorDays (VIN, color, days_sold, years_sold) AS
  (SELECT S.VIN,
      true_color,
      TIMESTAMPDIFF(DAY, sold_date, MAX(sold_date) OVER()) AS days_sold,
      TIMESTAMPDIFF(YEAR, sold_date, MAX(sold_date) OVER()) AS years_sold
  FROM Sale AS S INNER JOIN VehicleTrueColor AS VTC ON S.VIN = VTC.VIN)


  , SaleByColor(color, 30Days, 1Year, allTime) AS
  (SELECT color,
      sum(CASE WHEN days_sold < 30 THEN 1 ELSE 0 END) AS 30Days,
      sum(CASE WHEN years_sold = 0 THEN 1 ELSE 0 END) as 1Year,
      count(*) as allTime
  FROM SaleColorDays
  GROUP BY color)


  , AllColors (color) AS
  (SELECT color
  FROM Color
```

```
UNION ALL
SELECT 'Multiple')

SELECT AC.color as color,
    IFNULL(30days, 0) as 30Days,
    IFNULL(1Year, 0) as 1Year,
    IFNULL(allTime, 0) as allTime
FROM AllColors AS AC LEFT JOIN SaleByColor AS SBC
ON AC.color = SBC.color
ORDER BY AC.color ASC;
```

13. report_sales_by_type.php (view report of sale by vehicle type)

```
     $query = WITH VehicleType AS
(SELECT VIN, type FROM car
  UNION SELECT VIN, type FROM truck
  UNION SELECT VIN, type FROM convertible
  UNION SELECT VIN, type FROM Van
  UNION SELECT VIN, type FROM suv)

, SaleTypeTime (VIN, type, days_sold, years_sold) AS
(SELECT V.VIN, type,
TIMESTAMPDIFF(DAY, sold_date, MAX(sold_date) OVER()) AS days_sold,
TIMESTAMPDIFF(YEAR, sold_date, MAX(sold_date) OVER()) AS years_sold
FROM Sale AS S
INNER JOIN Vehicle AS V ON S.VIN = V.VIN
INNER JOIN VehicleType AS VT ON S.VIN = VT.VIN)

, SaleByType (type, 30days, oneYear, allTime) AS
(SELECT type,
  sum(CASE WHEN days_sold < 30 THEN 1 ELSE 0 END) as 30Days,
  sum(CASE WHEN years_sold = 0 THEN 1 ELSE 0 END) as oneYear,
  count(*) as allTime
FROM SaleTypeTime
GROUP BY type)

, AllTypes (type) AS
(SELECT DISTINCT type FROM car
UNION SELECT DISTINCT type FROM convertible
UNION SELECT DISTINCT type FROM Van
UNION SELECT DISTINCT type FROM truck
UNION SELECT DISTINCT type FROM suv)
```

```
   SELECT AT.type as type,
   IFNULL(30days, 0) AS 30Days,
   IFNULL(oneYear, 0) AS 1Year,
   IFNULL(allTime, 0) AS allTime
   FROM AllTypes AS AT LEFT JOIN SaleByType AS SBT ON AT.type = SBT.type
   ORDER BY AT.type;
```

14. report_sales_by_manufacturer.php (view report of sale by manufacturer)

```
          $query = WITH SaleManTime (VIN, manufacturer, days_sold, years_sold) AS
   (SELECT V.VIN, M.name,
   TIMESTAMPDIFF(DAY, sold_date, MAX(sold_date) OVER()) AS days_sold,
   TIMESTAMPDIFF(YEAR, sold_date, MAX(sold_date) OVER()) AS years_sold
   FROM Sale AS S
   INNER JOIN Vehicle AS V ON S.VIN = V.VIN
   INNER JOIN Manufacturer as M ON V.mID = M.mID)

   SELECT manufacturer,
      sum(CASE WHEN days_sold < 30 THEN 1 ELSE 0 END) as 30Days,
      sum(CASE WHEN years_sold = 0 THEN 1 ELSE 0 END) as 1Year,
      count(*) as allTime
   FROM SaleManTime
   GROUP BY manufacturer
   ORDER BY manufacturer;
```

15. report_gross_customer_income.php (view report of sales by gross customer income of top 15 customers and their sale details and/or repair details)

```
          $query = WITH PartsCostByID (repairID, parts_cost) AS
   (SELECT repairID, sum(part_quantity * part_price) AS parts_cost
   FROM Part
   GROUP BY repairID)

   , CustomerRepair (RepairCustomerID, repair_count, first_repair_date, last_repair_date,
repair_charge) AS
   (SELECT R.customerID, count(*),
         min(R.start_date), max(R.start_date),
         sum(IFNULL(labor_charge, 0) + IFNULL(parts_cost, 0))
   FROM Repair AS R LEFT JOIN PartsCostByID AS PCID ON R.repairID = PCID.repairID
   GROUP BY R.customerID)

   , CustomerSale (SaleCustomerID, sale_count, first_sale_date, last_sale_date, sale_charge) AS
   (SELECT customerID, count(*), min(sold_date), max(sold_date), sum(sold_price)
   FROM Sale
   GROUP BY customerID)

   , CustomerRepairSaleUnion AS
   (SELECT * FROM CustomerRepair as CR
   LEFT JOIN CustomerSale AS CS on CR.RepairCustomerID = CS.SaleCustomerID
   UNION
   SELECT * FROM CustomerRepair AS CR
   RIGHT JOIN CustomerSale AS CS on CR.RepairCustomerID = CS.SaleCustomerID)


   , CustomerExpense AS
   (SELECT
       CASE WHEN RepairCustomerID is NOT NULL THEN RepairCustomerID
       ELSE SaleCustomerID
       END AS customerID,
       IFNULL(repair_count, 0) AS repair_count,
       IFNULL(sale_count, 0) as sale_count,
       CASE
         WHEN first_repair_date IS NULL THEN first_sale_date
         WHEN first_sale_date IS NULL THEN first_repair_date
          ELSE LEAST(first_repair_date, first_sale_date)
       END AS first_date,

       CASE
         WHEN last_repair_date IS NULL THEN last_sale_date
         WHEN last_sale_date IS NULL THEN last_repair_date
          ELSE GREATEST(first_repair_date, first_sale_date)
       END AS last_date,

       IFNULL(repair_charge, 0) + IFNULL(sale_charge, 0) as total_charge
   FROM CustomerRepairSaleUnion
   )

   , CustomerExpenseSort AS
   (SELECT * FROM CustomerExpense
   ORDER BY total_charge DESC, last_date DESC
   LIMIT 15
   )

   , CustomerTypeName AS
   (SELECT customerID, 'Individual' AS customer_type, CONCAT(first_name, ' ', last_name) as customer_name FROM individual
   UNION
```

```
   SELECT customerID, 'Business' AS customer_type, business_name as customer_name From
business)


   SELECT CE.customerID, customer_type, customer_name, first_date, last_date, sale_count,
repair_count, total_charge
   FROM CustomerExpenseSort as CE INNER JOIN CustomerTypeName as CTN ON
CE.customerID = CTN.customerID
   ORDER BY total_charge DESC, last_date DESC;

          // view customer sale information
          $query = SELECT sold_date, sold_price, S.VIN, model_year, M.name as manufacturer,
model_name, LU.first_name, LU.last_name
      FROM Sale AS S INNER JOIN Vehicle AS V ON S.VIN = V.VIN
      INNER JOIN loginuser AS LU ON S.username = LU.username
      INNER JOIN manufacturer AS M ON V.mID = M.mID
      WHERE S.customerID = {$customerID}
      ORDER BY sold_date DESC, S.VIN ASC;

          // view customer repair information
          $query = WITH PartsCostByID (repairID, parts_cost) AS
      (SELECT repairID, sum(part_quantity * part_price) AS parts_cost
      FROM Part
      GROUP BY repairID)

      SELECT R.customerID AS customerID,
            start_date,
            R.VIN as VIN,
            odometer,
            complete_date,
            IFNULL(parts_cost, 0) as parts_cost,
            labor_charge,
            IFNULL(parts_cost, 0) + IFNULL(labor_charge, 0) AS total_cost,
            first_name,
            last_name
      FROM Repair AS R LEFT JOIN PartsCostByID AS PCID ON R.repairID = PCID.repairID
      INNER JOIN Vehicle AS V ON R.VIN = V.VIN
      INNER JOIN LoginUser as LU ON R.username = LU.username
      WHERE R.customerID = {$customerID}
      ORDER BY start_date DESC, complete_date IS NOT NULL, complete_date DESC, R.VIN ASC;
```

16. report_repair_manufacturer_type_model.php (view report of repair by manufacturer/type/model)

```
        // report of repair by manufacturer
        $query = WITH PartsCostByID (repairID, parts_cost) AS
  (SELECT repairID, sum(part_quantity * part_price) AS parts_cost
  FROM Part
  GROUP BY repairID)

  , ManufactuerRepair AS
  (SELECT M.name AS manufacturer,
      count(*) AS count_repairs,
      sum(IFNULL(parts_cost, 0)) as all_parts_cost,
      sum(IFNULL(labor_charge, 0)) as all_labor_cost,
      sum(IFNULL(parts_cost, 0) + IFNULL(labor_charge, 0)) as total_repairs_cost
  FROM Repair AS R LEFT JOIN PartsCostByID AS PCID ON R.repairID = PCID.repairID
  INNER JOIN Vehicle AS V ON R.VIN = V.VIN
  INNER JOIN Manufacturer as M ON V.mID = M.mID
  GROUP BY M.name)

  SELECT
    M.name as manufacturer_name,
    M.mID as manufacturer_id,
    IFNULL(count_repairs, 0) AS count_repairs,
    IFNULL(all_parts_cost, 0) AS all_parts_cost,
    IFNULL(all_labor_cost, 0) AS all_labor_cost,
    IFNULL(total_repairs_cost, 0) AS total_repairs_cost
  FROM Manufacturer AS M LEFT JOIN ManufactuerRepair AS MR ON M.name = MR.manufacturer
  ORDER BY M.name;

        //  report of repair details by type/model
        $query = WITH PartsCostByID (repairID, parts_cost) AS
    (SELECT repairID, sum(part_quantity * part_price) AS parts_cost
    FROM Part
    GROUP BY repairID)

    , VinType (VIN, vehicle_type) AS
    (SELECT VIN, type FROM car
    UNION SELECT VIN, type FROM truck
    UNION SELECT VIN, type FROM convertible
    UNION SELECT VIN, type FROM suv
    UNION SELECT VIN, type FROM Van)
    , RepairInfo (repairID, vehicle_type, model_name, parts_cost, labor_cost, total_cost) AS
    (SELECT R.repairID,
        vehicle_type,
        model_name,
        IFNULL(parts_cost, 0) AS parts_cost,
        IFNULL(labor_charge, 0) AS labor_cost,
        IFNULL(parts_cost, 0) + IFNULL(labor_charge, 0) AS total_cost
    FROM Repair AS R
    LEFT JOIN PartsCostByID AS PCID ON R.repairID = PCID.repairID
    INNER JOIN Vehicle AS V ON R.VIN = V.VIN
    INNER JOIN VinType AS VT ON R.VIN = VT.VIN
    WHERE V.mID = {$mID})

    , RepairbyGroup (repairID, vehicle_type, model_name, model_number_repairs,
model_parts_cost, model_labor_cost, model_repair_cost) AS
    (SELECT repairID, vehicle_type, model_name,
        count(*) as model_number_repairs,
        sum(parts_cost) as model_parts_cost,
        sum(labor_cost) as model_labor_cost,
        sum(parts_cost + labor_cost) as model_repair_cost
    FROM RepairInfo
    GROUP BY vehicle_type, model_name)

    , RepairTypeModel AS
    (SELECT vehicle_type, model_name,
        model_number_repairs, model_labor_cost, model_parts_cost, model_repair_cost,
        sum(model_number_repairs) OVER(PARTITION BY vehicle_type) as
type_number_repairs,
        sum(model_parts_cost) OVER(PARTITION BY vehicle_type) as type_parts_cost,
        sum(model_labor_cost) OVER(PARTITION BY vehicle_type) as type_labor_cost,
        sum(model_repair_cost) OVER (PARTITION by vehicle_type) as type_repair_cost
    FROM RepairbyGroup)

    SELECT vehicle_type, model_name,
       model_number_repairs, model_labor_cost, model_parts_cost, model_repair_cost,
       type_number_repairs, type_labor_cost, type_parts_cost, type_repair_cost
    FROM RepairTypeModel
    ORDER BY type_number_repairs DESC, vehicle_type DESC, model_number_repairs DESC;
```

17. report_below_cost_sales.php (view report of below cost sales)

```
      $query = WITH CustomerTypeName AS
```

```
   (SELECT customerID, 'Individual' AS customer_type, CONCAT(first_name, ' ', last_name) as customer_name FROM individual
   UNION
   SELECT customerID, 'Business' AS customer_type, business_name as customer_name From business)

   SELECT sold_date, invoice_price, sold_price, sold_price / invoice_price * 100.0 AS ratio, customer_name, LU.first_name, LU.last_name
   FROM Sale AS S INNER JOIN Vehicle AS V ON S.VIN = V.VIN
   INNER JOIN CustomerTypeName as CTN ON S.customerID = CTN.customerID
   INNER JOIN LoginUser as LU ON S.username = LU.username
   WHERE sold_price < invoice_price
   ORDER BY sold_date DESC, sold_price / invoice_price DESC;
```

18. report_average_time_in_inventory.php (view report of average time in inventory)

```
          $query = WITH VinType (VIN, vehicle_type) AS
   (SELECT VIN, type FROM car
   UNION SELECT VIN, type FROM truck
   UNION SELECT VIN, type FROM convertible
   UNION SELECT VIN, type FROM suv
   UNION SELECT VIN, type FROM van)

   , DaysInInventory (VIN, vehicle_type, days_in_inventroy) AS
   (SELECT S.VIN,
      vehicle_type,
      TIMESTAMPDIFF(DAY, add_date, sold_date) + 1 AS days_in_inventroy
   FROM Sale AS S
   INNER JOIN vehicle AS V ON S.VIN = V.VIN
   INNER JOIN VinType AS VT ON S.VIN = VT.VIN)

   , AvgDaysInventory (vehicle_type, avg_days_inventroy) AS
   (SELECT vehicle_type, avg(days_in_inventroy)
   FROM DaysInInventory
   GROUP BY vehicle_type)

   , AllTypes (vehicle_type) AS
   (SELECT DISTINCT type FROM car
   UNION SELECT DISTINCT type FROM convertible
   UNION SELECT DISTINCT type FROM Van
   UNION SELECT DISTINCT type FROM truck
   UNION SELECT DISTINCT type FROM suv)
   SELECT AT.vehicle_type AS vehicle_type, IFNULL(avg_days_inventroy, 'NA') as
avg_days_inventory
   FROM AllTypes as AT LEFT JOIN AvgDaysInventory AS ADI
   ON AT.vehicle_type = ADI.vehicle_type
   ORDER BY AT.vehicle_type ASC;
```

19. report_parts_statistics.php (view report of parts statistics)

```
          $query = SELECT vendor,
     sum(part_quantity) as number_parts,
     sum(part_price * part_quantity) as total_parts_expense
          FROM Part
          GROUP BY vendor
          ORDER BY sum(part_price * part_quantity) DESC;
```

20. report_monthly_sales.php (view report of monthly sales and top salesperson)

```
          // show monthly sales
          $query = WITH SaleByMonth (VIN, year, month, sold_price, invoice_price) AS
   (SELECT Sale.VIN,
    YEAR(sold_date) AS year,
    Month(sold_date) AS month,
    sold_price,
    invoice_price
   FROM Sale INNER JOIN Vehicle ON Sale.VIN = Vehicle.VIN)

   SELECT year,
    month,
    count(*) as number_sale,
    sum(sold_price) as total_sale_income,
    sum(sold_price - invoice_price) as total_net_income,
    ((sum(sold_price) / sum(invoice_price)) * 100.0) as ratio
   FROM SaleByMonth
   GROUP BY year, month
   ORDER BY year DESC, month DESC;

          // show top salesperson
          $query = WITH SaleBySomeMonth (saleperson_name, sold_price) AS
       (SELECT CONCAT(LU.first_name, ' ', LU.last_name) as saleperson_name, sold_price
       FROM Sale INNER JOIN LoginUser AS LU ON Sale.username = LU.username
       WHERE YEAR(sold_date) = {$year} AND MONTH(sold_date) = {$month})

      SELECT saleperson_name,
         count(*) as total_vehicles_sold,
         sum(sold_price) as total_sales
      FROM SaleBySomeMonth
      GROUP BY saleperson_name
      ORDER BY count(*) DESC, sum(sold_price) DESC;
```