Name : Vivek Chavan

Class : 3rd sem ‘E’ div

Roll no : 551

USN : 01fe21bcs313

CREATE TABLE:

```
USE carrentalco;
 CREATE TABLE Car (
  CarID INT PRIMARY KEY,
  Make VARCHAR(255) NOT NULL,
  Model VARCHAR(255) NOT NULL,
  Year_of_Production INT NOT NULL,
  Engine_Size DECIMAL(10,2) NOT NULL,
  Fuel_Type VARCHAR(255) NOT NULL,
  Number_of_Passengers INT NOT NULL,
  Registration_Number VARCHAR(255) NOT NULL,
  Purchase_Price DECIMAL(10,2) NOT NULL,
  Purchase_Date DATE NOT NULL,
  Rent_Price DECIMAL(10,2) NOT NULL,
  Insurance_Details VARCHAR(255) NOT NULL
);
```

```
INSERT INTO Car (CarID, Make, Model, Year_of_Production, Engine_Size, Fuel_Type,
Number_of_Passengers, Registration_Number, Purchase_Price, Purchase_Date, Rent_Price,
Insurance_Details)
VALUES
```

```
  (1, 'Toyota', 'Camry', 2019, 2.5, 'Petrol', 5, 'ABC123', 20000, '2020-01-01', 100, 'XYZ Insurance'),
  (2, 'Honda', 'Civic', 2020, 1.5, 'Petrol', 5, 'DEF456', 22000, '2020-02-01', 95, 'ABC Insurance'),
  (3, 'Tesla', 'Model 3', 2021, 3.0, 'Electric', 5, 'GHI789', 55000, '2021-01-01', 150, 'PQR Insurance');

CREATE TABLE Garage (
  GarageID INT PRIMARY KEY,
  Garage_Name VARCHAR(255) NOT NULL,
  Address VARCHAR(255) NOT NULL,
  Range_of_Services VARCHAR(255) NOT NULL,
  Payment_Terms VARCHAR(255) NOT NULL
);

INSERT INTO Garage (GarageID, Garage_Name, Address, Range_of_Services, Payment_Terms)
VALUES
  (1, 'AutoCare', '123 Main Street', 'Maintenance, Repairs, Upgrades', 'Monthly'),
  (2, 'TechGarage', '456 Park Ave', 'Diagnostics, Repairs, Customization', 'Per Service'),
  (3, 'FastFix', '789 Broadway', 'Emergency Repairs, Maintenance', 'Upfront');

CREATE TABLE Expenditures (
  ExpenditureID INT PRIMARY KEY,
  Expense_Type VARCHAR(255) NOT NULL,
  Amount DECIMAL(10,2) NOT NULL,
  Date DATE NOT NULL,
  Description VARCHAR(255) NOT NULL
);
```

```
INSERT INTO Expenditures (ExpenditureID, Expense_Type, Amount, Date, Description)
VALUES
  (1, 'Repairs', 1000, '2022-01-01', 'Engine repair for Toyota Camry'),
  (2, 'Maintenance', 500, '2022-02-01', 'Regular maintenance for Honda Civic'),
  (3, 'Insurance', 800, '2022-03-01', 'Renewal of insurance for Tesla Model 3');

CREATE TABLE Revenues (
  RevenueID INT PRIMARY KEY,
  Revenue_Type VARCHAR(255) NOT NULL,
  Amount DECIMAL(10,2) NOT NULL,
  Date DATE NOT NULL,
  Description VARCHAR(255) NOT NULL
);

INSERT INTO Revenues (RevenueID, Revenue_Type, Amount, Date, Description)
VALUES
  (1, 'Rent', 1000, '2022-01-01', 'Rent for Toyota Camry'),
  (2, 'Sale', 5000, '2022-02-01', 'Sale of Honda Civic'),
  (3, 'Rent', 1500, '2022-03-01', 'Rent for Tesla Model 3');

CREATE TABLE Customer (
  CustomerID INT PRIMARY KEY,
  Name VARCHAR(255) NOT NULL,
  Address VARCHAR(255) NOT NULL,
  Telephone_Number VARCHAR(255) NOT NULL,
  Driving_License_Number VARCHAR(255) NOT NULL,
  Credit_Card_Details VARCHAR(255) NOT NULL,
  Booking_Details VARCHAR(255) NOT NULL,
```

```
  Payment_Method VARCHAR(255) NOT NULL
);

INSERT INTO Customer (CustomerID, Name, Address, Telephone_Number,
Driving_License_Number, Credit_Card_Details, Booking_Details, Payment_Method)
VALUES
  (1, 'John Doe', '123 Main St', '555-555-5555', 'DL1234567', 'VISA 1234', 'Booking for Toyota
Camry from 01/01/2022 to 01/02/2022', 'Credit Card'),
  (2, 'Jane Doe', '456 Park Ave', '555-555-5556', 'DL2345678', 'MASTERCARD 5678', 'Booking for
Honda Civic from 02/01/2022 to 02/02/2022', 'Cash'),
  (3, 'Jim Smith', '789 Broadway', '555-555-5557', 'DL3456789', 'AMEX 9012', 'Booking for Tesla
Model 3 from 03/01/2022 to 03/02/2022', 'Debit Card');

CREATE TABLE Car_Garage (
  CarID INT,
  GarageID INT,
  PRIMARY KEY (CarID, GarageID),
  FOREIGN KEY (CarID) REFERENCES Car(CarID),
  FOREIGN KEY (GarageID) REFERENCES Garage(GarageID)
);

INSERT INTO Car_Garage (CarID, GarageID)
VALUES
  (1, 1),
  (2, 2),
  (3, 3);

CREATE TABLE Customer_Car (
```

```
  CustomerID INT,
  CarID INT,
  PRIMARY KEY (CustomerID, CarID),
  FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
  FOREIGN KEY (CarID) REFERENCES Car(CarID)
);

INSERT INTO Customer_Car (CustomerID, CarID)
VALUES
  (1, 1),
  (2, 2),
  (3, 3);

select * from Car;
select * from Garage;
select * from Expenditures;
select * from Revenues;
select * from Customer;
select * from Car_Garage;
select * from Customer_Car;

select * from car;
select * from garage;
select * from expenditures;
select * from revenues;
select * from car_garage;
select * from customer_car;
```

```
-- Multiple join operations;

-- 1)Joining the Expenditures and Garage tables to find the expenses incurred by each garage:
select car.Make,car.Model,garage.Garage_Name from car
join car_garage on car.CarID = car_garage.CarID
join garage on car_garage.GarageID = garage.GarageID;

-- 2)Joining the Expenditures and Garage tables to find the expenses incurred by each garage:
SELECT customer.Name, SUM(revenues.Amount) AS 'Total_Revenue'
FROM revenues
JOIN Customer_Car ON revenues.RevenueID = Customer_Car.CustomerID
JOIN customer ON Customer_Car.CustomerID = customer.CustomerID
GROUP BY customer.Name;

-- 3)Joining the Expenditures and Garage tables to find the expenses incurred by each garage:
SELECT garage.Garage_Name, SUM(expenditures.Amount) AS 'Total_Expenditure'
FROM expenditures
JOIN garage ON expenditures.ExpenditureID = garage.GarageID
GROUP BY garage.Garage_Name;

-- 4)Joining the Car, Customer, and Revenues tables to find the revenue generated by each car:
SELECT car.Make, car.Model, SUM(revenues.Amount) AS 'Total_Revenues'
FROM car
JOIN Customer_Car ON car.CarID = Customer_Car.CarID
JOIN revenues ON Customer_Car.CustomerID = revenues.RevenueID
GROUP BY car.Make, car.Model;

-- 5)Joining the Car, Customer, and Revenues tables to find the most rented car model:
```

```
SELECT Car.Model, COUNT(*) as Rentals
FROM Car
JOIN Customer_Car
ON Car.CarID = Customer_Car.CarID
JOIN Revenues
ON Customer_Car.CustomerID = Revenues.RevenueID
GROUP BY Car.Model
ORDER BY Rentals DESC
LIMIT 1;


-- subqueries single row;

-- 1)Find car with highest engine size:
select Engine_Size from car
where Engine_Size = (select max(Engine_Size) from car);

-- 2)Find car with oldest year of production
select Year_of_Production from car
where Year_of_Production = (select min(Year_of_Production) from car);

-- 3)Find car with highest purchase price
select Purchase_Price from car
where Purchase_Price = (select max(Purchase_Price) from car);

-- 4)Find car with most recent purchase date
select Purchase_Date from car
where Purchase_Date = (select max(Purchase_Date) from car);
```

```
-- 5)Find car with highest rent price
select Rent_Price from car
where Rent_Price = (select max(Rent_Price) from car);

-- subqueries correlated

-- 1)To find the cars with highest purchase price for each make
select car.Make,car.Model,car.Purchase_Price from car
where car.Purchase_Price=(select max(Purchase_Price) from car c2
                    where c2.Make = car.Make);

-- 2)To find customer who has rented most cars
SELECT Customer.Name
FROM Customer
WHERE (
  SELECT COUNT(*)
  FROM Customer_Car
  WHERE Customer_Car.CustomerID = Customer.CustomerID
) = (
  SELECT MAX(cust_count)
  FROM (
    SELECT CustomerID, COUNT(*) as cust_count
    FROM Customer_Car
    GROUP BY CustomerID
  ) sub
);
```

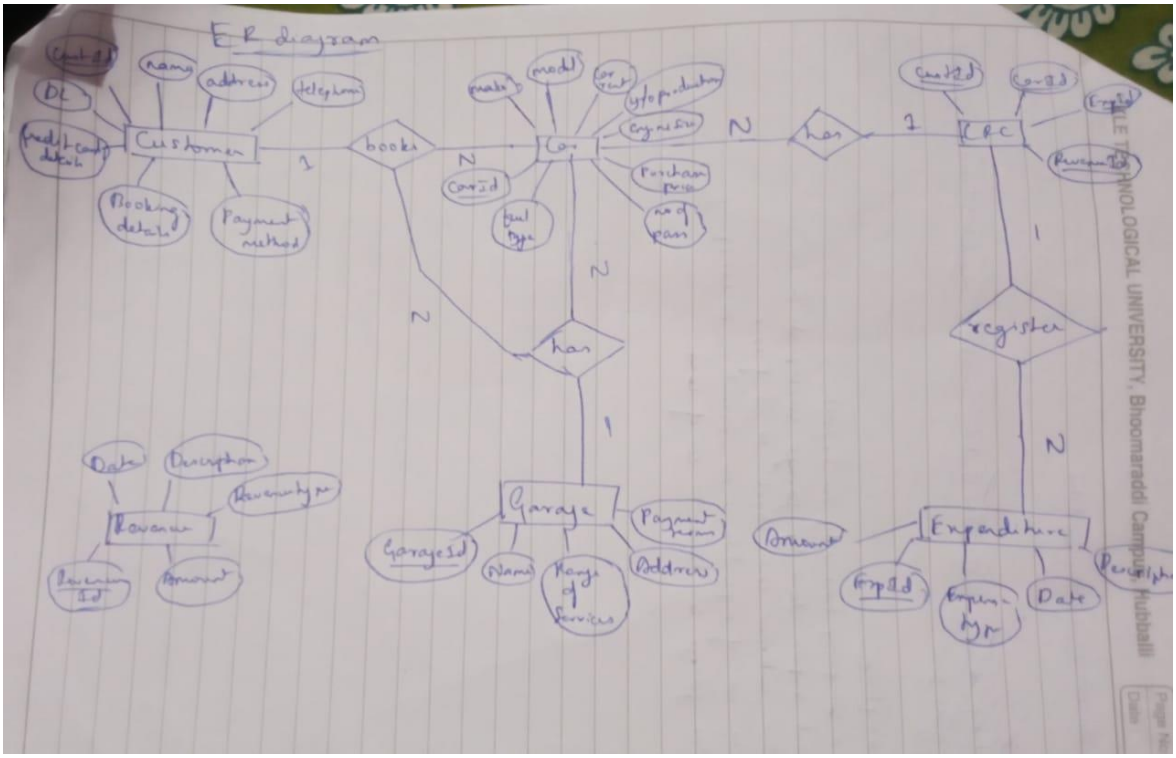
```
  -- 3)To find all customers who have rented most expensive car;
SELECT Customer.Name
FROM Customer
WHERE EXISTS (
  SELECT 1
  FROM Customer_Car
  JOIN Car ON Customer_Car.CarID = Car.CarID
  WHERE Customer_Car.CustomerID = Customer.CustomerID
  AND Car.Rent_Price = (
    SELECT MAX(Rent_Price)
    FROM Car
  )
);


  -- 4)To find no of expenditure of each type of expense;
SELECT Expense_Type, SUM(Amount)
FROM Expenditures
GROUP BY Expense_Type
HAVING SUM(Amount) = (
  SELECT MAX(sub.expense_total)
  FROM (
    SELECT Expense_Type, SUM(Amount) as expense_total
    FROM Expenditures
    GROUP BY Expense_Type
  ) sub
);
```

```
-- 5) To find the garages with the highest range of services:
SELECT Garage_Name, Range_of_Services
FROM Garage
WHERE Range_of_Services = (
  SELECT MAX(Range_of_Services)
  FROM Garage
);




-- arthematic and logicaloperatoes

-- 1)Find total purchase price of all cars
SELECT SUM(Purchase_Price) AS Total_Purchase_Price
FROM Car;

-- 2)Calculate avg rent price for all cars
SELECT AVG(Rent_Price) AS Average_Rent_Price
FROM Car;

-- 3)Find total amount spent on repairs
SELECT SUM(Amount) AS Total_Expenditure_On_Repairs
FROM Expenditures
WHERE Expense_Type = 'Repairs';

-- 4)Find all the cars whose production year is between 2015 to 2018
```

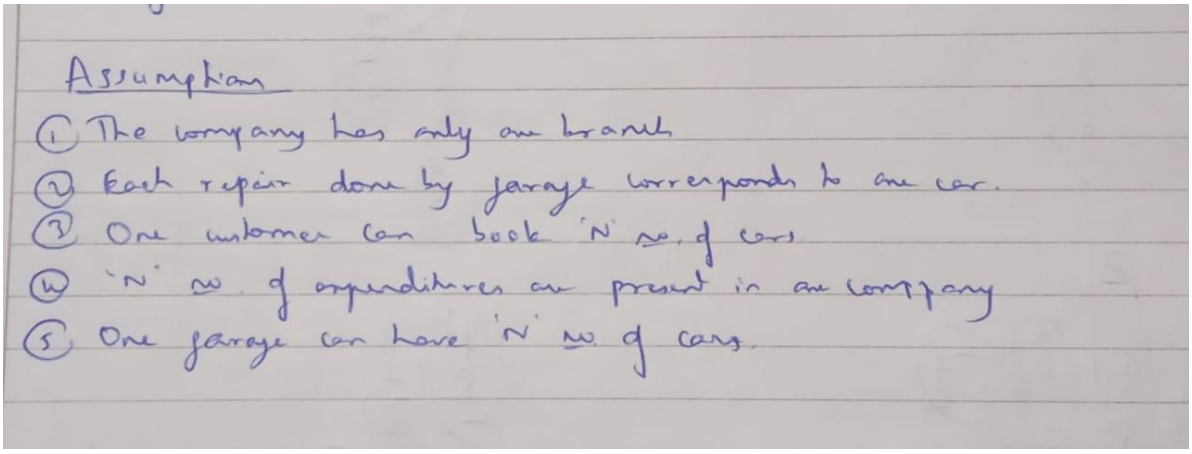
```
SELECT *
FROM Car
WHERE Year_of_Production < 2015 OR Year_of_Production > 2018;

-- 5)Find all the cars whose production year is not between 2015 to 2018
SELECT *
FROM Car
WHERE NOT (Year_of_Production BETWEEN 2015 AND 2018);

-- Clauses

-- 1)Find the cars and their rent price that have the make as toyota
SELECT Make, Rent_Price
FROM Car
WHERE Make = 'Toyota';

-- 2) find the cars where 5 passangers can be seated
select * from car where Number_of_Passengers='5';

-- 3)To retrieve all the cars sorted by their rent price in descending order:
SELECT * FROM Car
ORDER BY Rent_Price DESC;

-- 4)To retrieve all the revenues sorted by the amount in descending order:
SELECT * FROM Revenues
ORDER BY Amount DESC;

-- 5)To find the total expenditures by type
```

```
SELECT Expense_Type, SUM(Amount) AS Total_Expenditures
FROM Expenditures
GROUP BY Expense_Type;

-- views:
-- 1)What are the details of all car rentals made by customers?
SELECT * FROM customer_rentals_view;

-- 2)What is the total revenue generated from car rentals and sales?
SELECT SUM(Amount) AS Total_Revenue
FROM revenues_view
WHERE Revenue_Type = 'Rent' OR Revenue_Type = 'Sale';

-- 3)What are the details of all expenditures made for car repairs and maintenance?
SELECT * FROM expenses_view
WHERE Expense_Type = 'Repairs' OR Expense_Type = 'Maintenance';

-- 4)What is the total amount spent on insurance for all cars?
SELECT SUM(Amount) AS Total_Insurance_Expenditures
FROM expenses_view
WHERE Expense_Type = 'Insurance';

-- 5)What is the total number of passengers that can be accommodated by all cars in the fleet?
SELECT SUM(Number_of_Passengers) AS Total_Capacity
FROM cars_view;
```

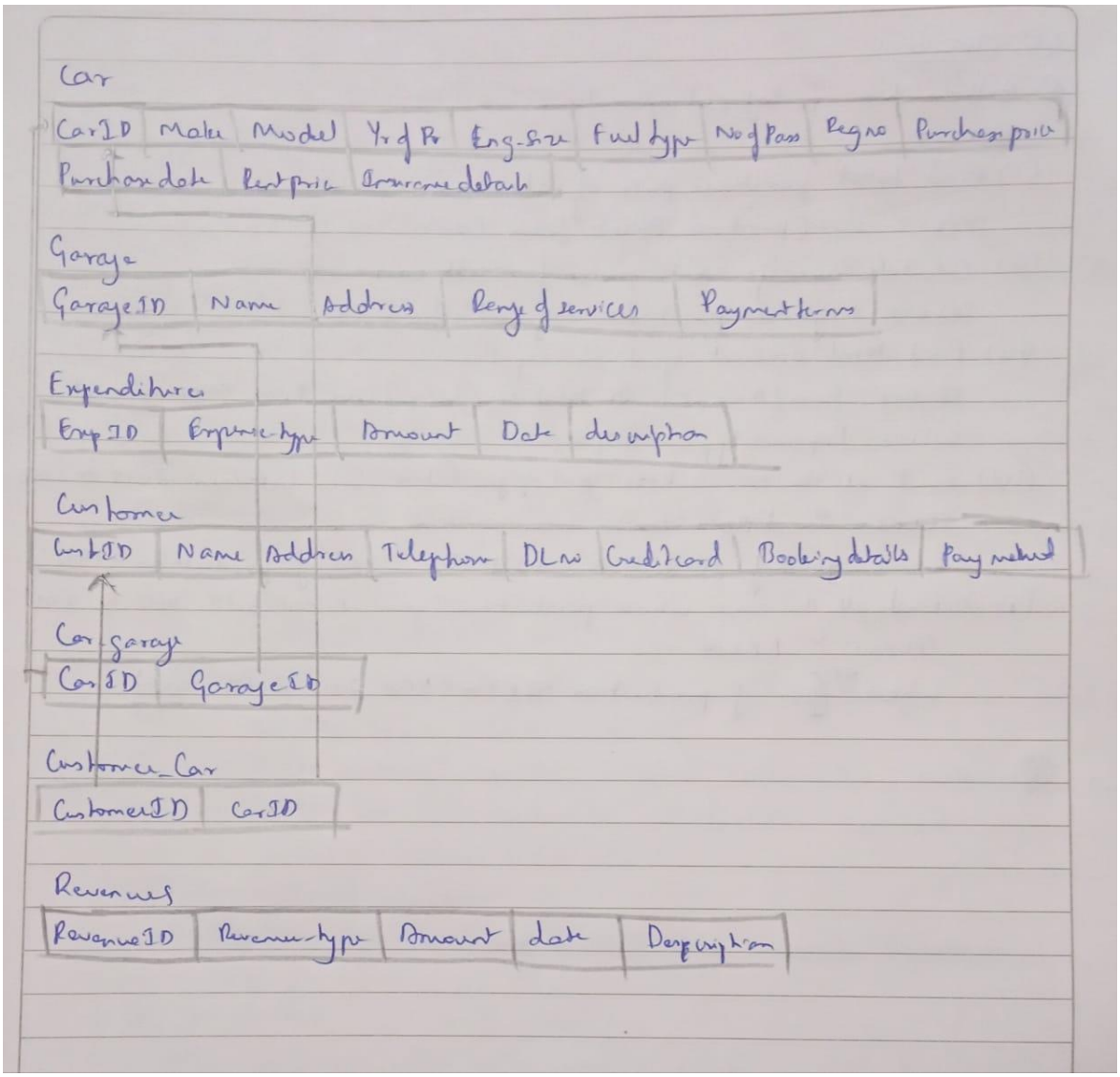
ER diagram :



Assumptions :



Relational Schema :


Car
CarID Make Model Yr of Pr Eng-Size Fuel type No of Pass Reg no Purchase price
Purchase date Rent price Insurance details
Garage
GarageID Name Address Range of services Payment terms
Expenditure
Exp ID Expense type Amount Date description
Customer
CustID Name Address Telephone DL no Credit card Booking details Pay method
Car_garage
CarID GarageID
Customer_Car
CustomerID CarID
Revenues
RevenueID Revenue-type Amount date Description