Introduction to Database Systems Final Project

Course Code: DBS211NDD

Authors:

- Mohsen Sabet Student ID: 113205165 Email: msabetkoohenjani@myseneca.ca
- Farzaneh Jahanian Student ID: 176014215 Email: Fjahanian@myseneca.ca

Title: Car Rental Management System (CRMS)

1. **Project Overview:**

The Car Rental Management System (CRMS) is a database solution designed for a car rental company to manage its operations more efficiently. This system will enable the company to handle reservations, customer information, vehicle details, rental rates, and more, resulting in improved customer service and operational efficiency.

2. **Business Rules:**

To create a successful Car Rental Management System, the following business rules must be considered:

1. A customer must be registered in the system with their personal information, including name, address, phone number, email address, and driver's license number.
2. A customer must be at least 21 years old to rent a car.
3. Each car in the fleet must have a unique identifier (e.g., Vehicle Identification Number or VIN), make, model, year, color, category, and rental rate.
4. Each car can only be rented from one location at a time and has its availability status tracked. While a customer is using a car, it is marked as "unavailable" and cannot be rented by other customers from any location. When a car is returned to a location, it is checked for maintenance and marked as "available" at that location if it is not under maintenance.
5. Each rental reservation must include the customer's ID, the car's VIN, the pick-up and return dates and times, and the rental location. The car must be available and assigned to the pickup location during the reservation period.
6. Rental rates vary based on the car category and rental duration.
7. Each rental transaction must include details such as the customer's ID, the car's VIN, the rental location, pick-up location, drop-off location, pick-up and return dates and times, and any additional charges (e.g., insurance, late fees). The car must be available at the time of the rental transaction.

8. A car can only be rented to one customer at a time. This constraint must be enforced by checking the car's availability status and existing reservations or rental transactions during the reservation and rental processes.
9. The system must keep track of car maintenance history, including service dates, performed services, and service provider information.
10. The system should support various user roles with different access levels, such as system administrators, branch managers, and customer service representatives.
11. A customer can be eligible for multiple promotions.
12. A promotion can be applied to multiple customers.
13. A car can be rented with different insurance packages, and an insurance package can be applied to multiple cars.
14. A customer can rent cars from multiple rental locations.
15. A rental location can serve multiple customers.
16. Each rental transaction must have a corresponding payment record, including payment ID, rental ID, payment method, amount, and payment date. A payment record can only be associated with one rental transaction, and each rental transaction can have exactly one payment record.

Entities and Attributes:

Customer:

- CustomerID (Primary Key)
- FirstName
- LastName
- Address
- PhoneNumber
- Email
- DateOfBirth
- DriversLicenseNumber

Car:

- VIN (Primary Key)
- Make
- Model
- Year
- Color
- Category
- RentalRate
- LocationID (Foreign Key)
- Status
- CurrentLocation

Location:

- LocationID (Primary Key)
- Address
- PhoneNumber

- Email

Reservation:

- ReservationID (Primary Key)
- CustomerID (Foreign Key)
- CarVIN (Foreign Key)
- LocationID (Foreign Key)
- PickUpDateTime
- ReturnDateTime
- ReservationStatus

Rental:

- RentalID (Primary Key)
- CustomerID (Foreign Key)
- CarVIN (Foreign Key)
- LocationID (Foreign Key)
- PickUpDateTime
- ReturnDateTime
- RentalRate
- AdditionalCharges
- RentalDuration

Maintenance:

- MaintenanceID (Primary Key)
- CarVIN (Foreign Key)
- ServiceDate
- ServicePerformed
- ServiceProvider

Users:

- UserID (Primary Key)
- Username
- Password
- UserRole

Promotion:

- PromotionID (Primary Key)
- PromotionName
- Description
- StartDate
- EndDate
- DiscountPercentage

InsurancePackage:

- InsuranceID (Primary Key)
- InsuranceName
- Description
- CoverageDetails
- Price

Payment:

- PaymentID (Primary Key)
- RentalID (Foreign Key)
- PaymentMethod
- Amount
- PaymentDate

CustomerPromotion:

• CustomerID (Foreign Key, Composite Primary Key)

• PromotionID (Foreign Key, Composite Primary Key)

CarInsurance:

• CarVIN (Foreign Key, Composite Primary Key)

• InsuranceID (Foreign Key, Composite Primary Key)

CustomerLocation:

• CustomerID (Foreign Key, Composite Primary Key)

• LocationID (Foreign Key, Composite Primary Key)

UserLocation:

• UserID (Foreign Key, Composite Primary Key)

• LocationID (Foreign Key, Composite Primary Key)

Relationships:

1. Customer to Reservation (One-to-Many)

    • A customer can have multiple reservations, but a reservation can only belong to one customer.

2. Car to Location (One-to-Many)

    • Each car can only be rented from one location at a time, but each location can have many cars.

3. Car to Reservation (One-to-One)

    • Each car can only be reserved by one customer at a time, and each reservation can only belong to one car. Once a car is reserved, it cannot be reserved again until the current reservation is complete and the car becomes available again.

4. Location to Reservation (One-to-Many)

    • A location can have zero to many reservations, but each reservation can only belong to one location.

5. Customer to Rental (One-to-Many)

    • A customer can have zero to multiple rentals, but a rental can only belong to one customer.

6. Car to Rental (One-to-Many)

    • A car can be rented multiple times, but each rental can only belong to one car.

7. Location to Rental (One-to-Many)

    • A location can have many rentals, but each rental can only belong to one location.

8. Car to Maintenance (One-to-Many)

    • A car can have zero to multiple maintenance records, but each maintenance record can only belong to one car.

9. User to Location (Many-to-Many)

    • Each user can be assigned to multiple locations, and each location can have many users.

10. Customer to Promotion (Many-to-Many)

    • A customer can be eligible for zero to multiple promotions, and a promotion can apply to multiple customers.

Normalization:

In this analysis, we have examined the given SQL script and established that it adheres to the normalization rules up to the Third Normal Form (3NF).

First Normal Form (1NF): The schema meets 1NF requirements, as each table has columns with unique names, all columns store atomic values, and the data types for each column are consistent across all rows. The order of data storage does not affect the overall design or functionality.

Second Normal Form (2NF): All tables are in 1NF, and all non-prime attributes in each table depend on the entire primary key, ensuring there are no partial dependencies. For example, the UserLocation table has a composite primary key (UserID, LocationID), and there are no other columns. Therefore, the table is in 2NF. Similar logic applies to other tables with composite primary keys, like CustomerPromotion, CarInsurance, and CustomerLocation.

Third Normal Form (3NF): All tables are in 2NF, and the schema is designed such that non-prime attributes depend only on primary key attributes, with no dependencies between non-prime attributes. This design ensures that the schema is in 3NF. For example, the Car table has a primary key of VIN, and all other attributes (Make, Model, Year, Color, Category, RentalRate, LocationID, Status, and CurrentLocation) depend solely on the primary key. There are no transitive dependencies among the non-prime attributes.

SQL script:

```
CREATE TABLE Location (
   LocationID INT PRIMARY KEY,
   Address VARCHAR(100) NOT NULL,
   PhoneNumber VARCHAR(20) NOT NULL,
   Email VARCHAR(100) NOT NULL
);

CREATE TABLE Users (
   UserID INT PRIMARY KEY,
   Username VARCHAR(50) NOT NULL,
   Password VARCHAR(50) NOT NULL,
   UserRole VARCHAR(50) NOT NULL
);

CREATE TABLE UserLocation (
   UserID INT NOT NULL,
   LocationID INT NOT NULL,
   PRIMARY KEY (UserID, LocationID),
   FOREIGN KEY (UserID) REFERENCES Users(UserID),
   FOREIGN KEY (LocationID) REFERENCES Location(LocationID)
);

CREATE TABLE Customer (
   CustomerID INT PRIMARY KEY,
   FirstName VARCHAR(50) NOT NULL,
   LastName VARCHAR(50) NOT NULL,
```

```
    Address VARCHAR(100) NOT NULL,
    PhoneNumber VARCHAR(20) NOT NULL,
    Email VARCHAR(100) NOT NULL,
    DateOfBirth DATE NOT NULL,
    DriversLicenseNumber VARCHAR(50) NOT NULL
);

CREATE TABLE Car (
    VIN VARCHAR(17) PRIMARY KEY,
    Make VARCHAR(50) NOT NULL,
    Model VARCHAR(50) NOT NULL,
    Year INT NOT NULL,
    Color VARCHAR(50) NOT NULL,
    Category VARCHAR(50) NOT NULL,
    RentalRate DECIMAL(6,2) NOT NULL,
    LocationID INT NOT NULL,
    Status VARCHAR(50) NOT NULL,
    CONSTRAINT FK_Car_Location FOREIGN KEY (LocationID)
        REFERENCES Location(LocationID)
);

CREATE TABLE Reservation (
    ReservationID INT PRIMARY KEY,
    CustomerID INT NOT NULL,
    CarVIN VARCHAR(17) NOT NULL,
    LocationID INT NOT NULL,
    PickUpDateTime TIMESTAMP NOT NULL,
    ReturnDateTime TIMESTAMP NOT NULL,
    CONSTRAINT FK_Reservation_Customer FOREIGN KEY (CustomerID)
```

```
    REFERENCES Customer(CustomerID),
  CONSTRAINT FK_Reservation_Car FOREIGN KEY (CarVIN)
    REFERENCES Car(VIN),
  CONSTRAINT FK_Reservation_Location FOREIGN KEY (LocationID)
    REFERENCES Location(LocationID)
);

CREATE TABLE Rental (
  RentalID INT PRIMARY KEY,
  CustomerID INT NOT NULL,
  CarVIN VARCHAR(17) NOT NULL,
  LocationID INT NOT NULL,
  PickUpDateTime TIMESTAMP NOT NULL,
  ReturnDateTime TIMESTAMP NOT NULL,
  AdditionalCharges DECIMAL(6,2) NOT NULL,
  CONSTRAINT FK_Rental_Customer FOREIGN KEY (CustomerID)
    REFERENCES Customer(CustomerID),
  CONSTRAINT FK_Rental_Car FOREIGN KEY (CarVIN)
    REFERENCES Car(VIN),
  CONSTRAINT FK_Rental_Location FOREIGN KEY (LocationID)
    REFERENCES Location(LocationID)
);

CREATE TABLE Maintenance (
  MaintenanceID INT PRIMARY KEY,
  CarVIN VARCHAR(17) NOT NULL,
  ServiceDate DATE NOT NULL,
  ServicePerformed VARCHAR(100) NOT NULL,
  ServiceProvider VARCHAR(50) NOT NULL,
```

```
    CONSTRAINT FK_Maintenance_Car FOREIGN KEY (CarVIN)
        REFERENCES Car(VIN)
);

CREATE TABLE Promotion (
    PromotionID INT PRIMARY KEY,
    PromotionName VARCHAR(50),
    Description VARCHAR(100),
    StartDate DATE,
    EndDate DATE,
    DiscountPercentage DECIMAL(5,2)
);

CREATE TABLE CustomerPromotion (
    CustomerID INT,
    PromotionID INT,
    PRIMARY KEY (CustomerID, PromotionID),
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
    FOREIGN KEY (PromotionID) REFERENCES Promotion(PromotionID)
);

CREATE TABLE InsurancePackage (
    InsuranceID INT PRIMARY KEY,
    InsuranceName VARCHAR(50),
    Description VARCHAR(100),
    CoverageDetails VARCHAR(200),
    Price DECIMAL(10,2)
);
CREATE TABLE CarInsurance (
```

```
    CarVIN VARCHAR(17),
    InsuranceID INT,
    PRIMARY KEY (CarVIN, InsuranceID),
    FOREIGN KEY (CarVIN) REFERENCES Car(VIN),
    FOREIGN KEY (InsuranceID) REFERENCES InsurancePackage(InsuranceID)
);

CREATE TABLE CustomerLocation (
    CustomerID INT,
    LocationID INT,
    PRIMARY KEY (CustomerID, LocationID),
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
    FOREIGN KEY (LocationID) REFERENCES Location(LocationID)
);

CREATE TABLE Payment (
    PaymentID INT PRIMARY KEY,
    RentalID INT,
    PaymentMethod VARCHAR(50),
    Amount DECIMAL(10,2),
    PaymentDate DATE,
    CONSTRAINT fk_payment_rental
        FOREIGN KEY (RentalID)
        REFERENCES Rental(RentalID)
);
```

Sample data script :

```
INSERT ALL
   INTO Location (LocationID, Address, PhoneNumber, Email) VALUES (1, '123 Main Street', '416-555-1234', 'mainstreetlocation@rental.com')
   INTO Location (LocationID, Address, PhoneNumber, Email) VALUES (2, '456 Yonge Street', '416-555-5678', 'yongelocation@rental.com')
   INTO Location (LocationID, Address, PhoneNumber, Email) VALUES (3, '789 Bay Street', '416-555-9012', 'baystreetlocation@rental.com')
   INTO Location (LocationID, Address, PhoneNumber, Email) VALUES (4, '321 Queen Street West', '416-555-3456', 'queenwestlocation@rental.com')
   INTO Location (LocationID, Address, PhoneNumber, Email) VALUES (5, '555 University Avenue', '416-555-7890', 'universitylocation@rental.com')
SELECT * FROM dual;

INSERT ALL
   INTO Users (UserID, Username, Password, UserRole) VALUES (1, 'admin', 'password', 'administrator')
   INTO Users (UserID, Username, Password, UserRole) VALUES (2, 'jane.smith', 'johndoe123', 'manager')
   INTO Users (UserID, Username, Password, UserRole) VALUES (3, 'jane.doe', 'janedoe123', 'manager')
   INTO Users (UserID, Username, Password, UserRole) VALUES (4, 'jack.smith', 'jacksmith123', 'customer rep')
   INTO Users (UserID, Username, Password, UserRole) VALUES (5, 'jill.johnson', 'jilljohnson123', 'customer rep')
SELECT * FROM dual;

INSERT ALL
   INTO UserLocation (UserID, LocationID) VALUES (1, 1)
   INTO UserLocation (UserID, LocationID) VALUES (2, 2)
   INTO UserLocation (UserID, LocationID) VALUES (3, 3)
   INTO UserLocation (UserID, LocationID) VALUES (4, 4)
   INTO UserLocation (UserID, LocationID) VALUES (5, 5)
```

```
SELECT * FROM dual;


INSERT ALL

   INTO Customer (CustomerID, FirstName, LastName, Address, PhoneNumber, Email, DateOfBirth, DriversLicenseNumber) VALUES (1, 'Alex', 'Johnson', '100 King St W', '416-555-1234', 'alex.johnson@gmail.com', TO_DATE('1988-06-15', 'YYYY-MM-DD'), '123456789')

   INTO Customer (CustomerID, FirstName, LastName, Address, PhoneNumber, Email, DateOfBirth, DriversLicenseNumber) VALUES (2, 'Emily', 'Wong', '200 Queen St E', '416-555-5678', 'emily.wong@hotmail.com', TO_DATE('1975-03-10', 'YYYY-MM-DD'), '987654321')

   INTO Customer (CustomerID, FirstName, LastName, Address, PhoneNumber, Email, DateOfBirth, DriversLicenseNumber) VALUES (3, 'Sam', 'Lee', '300 Bay St', '416-555-9012', 'sam.lee@yahoo.com', TO_DATE('1992-12-05', 'YYYY-MM-DD'), '654321987')

   INTO Customer (CustomerID, FirstName, LastName, Address, PhoneNumber, Email, DateOfBirth, DriversLicenseNumber) VALUES (4, 'Sophia', 'Garcia', '400 Yonge St', '416-555-3456', 'sophia.garcia@gmail.com', TO_DATE('1980-11-22', 'YYYY-MM-DD'), '789456123')

   INTO Customer (CustomerID, FirstName, LastName, Address, PhoneNumber, Email, DateOfBirth, DriversLicenseNumber) VALUES (5, 'Daniel', 'Kim', '500 University Ave', '416-555-7890', 'daniel.kim@hotmail.com', TO_DATE('1977-09-30', 'YYYY-MM-DD'), '456123789')

SELECT * FROM dual;


INSERT ALL

   INTO Car (VIN, Make, Model, Year, Color, Category, RentalRate, LocationID, Status) VALUES ('1G1YY26E385133258', 'Chevrolet', 'Corvette', 2018, 'Black', 'Sports Car', 150.00, 1, 'Available')

   INTO Car (VIN, Make, Model, Year, Color, Category, RentalRate, LocationID, Status) VALUES ('JN8AS5MV2DW106547', 'Nissan', 'Rogue', 2013, 'Silver', 'SUV', 75.00, 2, 'Available')

   INTO Car (VIN, Make, Model, Year, Color, Category, RentalRate, LocationID, Status) VALUES ('1FAHP2E85DG174263', 'Ford', 'Taurus', 2013, 'White', 'Sedan', 65.00, 3, 'Available')

   INTO Car (VIN, Make, Model, Year, Color, Category, RentalRate, LocationID, Status) VALUES ('KMHDN46D24U864675', 'Hyundai', 'Elantra', 2004, 'Red', 'Sedan', 40.00, 4, 'Not Available')

   INTO Car (VIN, Make, Model, Year, Color, Category, RentalRate, LocationID, Status) VALUES ('2C3CDXCT6EH185923', 'Dodge', 'Charger', 2014, 'Blue', 'Sedan', 80.00, 5, 'Available')

SELECT * FROM dual;
```

```
INSERT ALL

   INTO Reservation (ReservationID, CustomerID, CarVIN, LocationID, PickUpDateTime, ReturnDateTime)
VALUES (1, 1, '1G1YY26E385133258', 1, TO_DATE('2023-05-01 09:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_DATE('2023-05-03 17:00:00', 'YYYY-MM-DD HH24:MI:SS'))

   INTO Reservation (ReservationID, CustomerID, CarVIN, LocationID, PickUpDateTime, ReturnDateTime)
VALUES (2, 2, 'JN8AS5MV2DW106547', 2, TO_DATE('2023-05-05 10:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_DATE('2023-05-08 15:00:00', 'YYYY-MM-DD HH24:MI:SS'))

   INTO Reservation (ReservationID, CustomerID, CarVIN, LocationID, PickUpDateTime, ReturnDateTime)
VALUES (3, 3, '1FAHP2E85DG174263', 3, TO_DATE('2023-05-10 13:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_DATE('2023-05-12 12:00:00', 'YYYY-MM-DD HH24:MI:SS'))

   INTO Reservation (ReservationID, CustomerID, CarVIN, LocationID, PickUpDateTime, ReturnDateTime)
VALUES (4, 4, '2C3CDXCT6EH185923', 5, TO_DATE('2023-05-15 08:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_DATE('2023-05-19 11:00:00', 'YYYY-MM-DD HH24:MI:SS'))

   INTO Reservation (ReservationID, CustomerID, CarVIN, LocationID, PickUpDateTime, ReturnDateTime)
VALUES (5, 5, '1G1YY26E385133258', 1, TO_DATE('2023-05-22 12:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_DATE('2023-05-25 18:00:00', 'YYYY-MM-DD HH24:MI:SS'))

SELECT * FROM dual;
```

```
INSERT ALL

   INTO Rental (RentalID, CustomerID, CarVIN, LocationID, PickUpDateTime, ReturnDateTime,
AdditionalCharges) VALUES (1, 1, '1G1YY26E385133258', 1, TO_DATE('2023-05-01 09:00:00', 'YYYY-MM-
DD HH24:MI:SS'), TO_DATE('2023-05-03 17:00:00', 'YYYY-MM-DD HH24:MI:SS'), 0.00)

   INTO Rental (RentalID, CustomerID, CarVIN, LocationID, PickUpDateTime, ReturnDateTime,
AdditionalCharges) VALUES (2, 2, 'JN8AS5MV2DW106547', 2, TO_DATE('2023-05-05 10:00:00', 'YYYY-
MM-DD HH24:MI:SS'), TO_DATE('2023-05-08 15:00:00', 'YYYY-MM-DD HH24:MI:SS'), 0.00)

   INTO Rental (RentalID, CustomerID, CarVIN, LocationID, PickUpDateTime, ReturnDateTime,
AdditionalCharges) VALUES (3, 3, '1FAHP2E85DG174263', 3, TO_DATE('2023-05-10 13:00:00', 'YYYY-MM-
DD HH24:MI:SS'), TO_DATE('2023-05-12 12:00:00', 'YYYY-MM-DD HH24:MI:SS'), 50.00)

   INTO Rental (RentalID, CustomerID, CarVIN, LocationID, PickUpDateTime, ReturnDateTime,
AdditionalCharges) VALUES (4, 4, '2C3CDXCT6EH185923', 5, TO_DATE('2023-05-15 08:00:00', 'YYYY-MM-
DD HH24:MI:SS'), TO_DATE('2023-05-19 11:00:00', 'YYYY-MM-DD HH24:MI:SS'), 100.00)

   INTO Rental (RentalID, CustomerID, CarVIN, LocationID, PickUpDateTime, ReturnDateTime,
AdditionalCharges) VALUES (5, 5, '1G1YY26E385133258', 1, TO_DATE('2023-05-22 12:00:00', 'YYYY-MM-
DD HH24:MI:SS'), TO_DATE('2023-05-25 18:00:00', 'YYYY-MM-DD HH24:MI:SS'), 25.00)
```

```
SELECT * FROM dual;


INSERT ALL

   INTO Maintenance (MaintenanceID, CarVIN, ServiceDate, ServicePerformed, ServiceProvider) VALUES (1, '1G1YY26E385133258', TO_DATE('2022-10-01', 'YYYY-MM-DD'), 'Oil change', 'Joe''s Auto')

   INTO Maintenance (MaintenanceID, CarVIN, ServiceDate, ServicePerformed, ServiceProvider) VALUES (2, 'JN8AS5MV2DW106547', TO_DATE('2023-01-15', 'YYYY-MM-DD'), 'Brake pad replacement', 'Tony''s Auto')

   INTO Maintenance (MaintenanceID, CarVIN, ServiceDate, ServicePerformed, ServiceProvider) VALUES (3, '1FAHP2E85DG174263', TO_DATE('2023-02-05', 'YYYY-MM-DD'), 'Tire rotation', 'Steve''s Auto')

   INTO Maintenance (MaintenanceID, CarVIN, ServiceDate, ServicePerformed, ServiceProvider) VALUES (4, '2C3CDXCT6EH185923', TO_DATE('2023-03-20', 'YYYY-MM-DD'), 'Air filter replacement', 'Mike''s Auto')

   INTO Maintenance (MaintenanceID, CarVIN, ServiceDate, ServicePerformed, ServiceProvider) VALUES (5, '1G1YY26E385133258', TO_DATE('2023-04-10', 'YYYY-MM-DD'), 'Transmission fluid flush', 'Bob''s Auto')

SELECT * FROM dual;


INSERT ALL

   INTO Promotion (PromotionID, PromotionName, Description, StartDate, EndDate, DiscountPercentage) VALUES (1, 'Spring Sale', 'Get 20% off your rental rate this spring!', TO_DATE('2023-04-01', 'YYYY-MM-DD'), TO_DATE('2023-05-31', 'YYYY-MM-DD'), 20.00)

   INTO Promotion (PromotionID, PromotionName, Description, StartDate, EndDate, DiscountPercentage) VALUES (2, 'Summer Special', 'Rent for 7 days, get 2 days free!', TO_DATE('2023-06-01', 'YYYY-MM-DD'), TO_DATE('2023-08-31', 'YYYY-MM-DD'), 28.57)

   INTO Promotion (PromotionID, PromotionName, Description, StartDate, EndDate, DiscountPercentage) VALUES (3, 'Back to School', 'Get a free GPS rental with any car rental!', TO_DATE('2023-09-01', 'YYYY-MM-DD'), TO_DATE('2023-09-30', 'YYYY-MM-DD'), 0.00)

   INTO Promotion (PromotionID, PromotionName, Description, StartDate, EndDate, DiscountPercentage) VALUES (4, 'Fall Foliage', 'Rent for 4 days, get 1 day free!', TO_DATE('2023-10-01', 'YYYY-MM-DD'), TO_DATE('2023-11-30', 'YYYY-MM-DD'), 25.00)

   INTO Promotion (PromotionID, PromotionName, Description, StartDate, EndDate, DiscountPercentage) VALUES (5, 'Winter Wonderland', 'Get 10% off your rental rate this winter!', TO_DATE('2023-12-01', 'YYYY-MM-DD'), TO_DATE('2024-02-28', 'YYYY-MM-DD'), 10.00)

SELECT * FROM dual;
```

```
INSERT ALL
   INTO CustomerPromotion (CustomerID, PromotionID) VALUES (1, 1)
   INTO CustomerPromotion (CustomerID, PromotionID) VALUES (2, 2)
   INTO CustomerPromotion (CustomerID, PromotionID) VALUES (3, 3)
   INTO CustomerPromotion (CustomerID, PromotionID) VALUES (4, 4)
   INTO CustomerPromotion (CustomerID, PromotionID) VALUES (5, 5)
SELECT * FROM dual;

INSERT ALL
   INTO InsurancePackage (InsuranceID, InsuranceName, Description, CoverageDetails, Price) VALUES (1, 'Basic Coverage', 'Covers collision, theft, and damage up to $25K.', 'Collision, theft, and damage coverage up to $25K.', 15.00)
   INTO InsurancePackage (InsuranceID, InsuranceName, Description, CoverageDetails, Price) VALUES (2, 'Premium Coverage', 'Covers collision, theft, and damage up to $50K.', 'Collision, theft, and damage coverage up to $50K and liability up to $1M.', 25.00)
   INTO InsurancePackage (InsuranceID, InsuranceName, Description, CoverageDetails, Price) VALUES (3, 'Full Coverage', 'Covers collision, theft, and damage up to $100K.', 'Collision, theft, and damage coverage up to $100K, liability up to $2M, and medical up to $25K.', 35.00)
SELECT * FROM dual;

INSERT ALL
   INTO CarInsurance (CarVIN, InsuranceID) VALUES ('1G1YY26E385133258', 1)
   INTO CarInsurance (CarVIN, InsuranceID) VALUES ('1G1YY26E385133258', 2)
   INTO CarInsurance (CarVIN, InsuranceID) VALUES ('JN8AS5MV2DW106547', 2)
   INTO CarInsurance (CarVIN, InsuranceID) VALUES ('1FAHP2E85DG174263', 1)
   INTO CarInsurance (CarVIN, InsuranceID) VALUES ('2C3CDXCT6EH185923', 3)
SELECT * FROM dual;

INSERT ALL
   INTO CustomerLocation (CustomerID, LocationID) VALUES (1, 1)
```

```
    INTO CustomerLocation (CustomerID, LocationID) VALUES (2, 2)
    INTO CustomerLocation (CustomerID, LocationID) VALUES (3, 3)
    INTO CustomerLocation (CustomerID, LocationID) VALUES (4, 4)
    INTO CustomerLocation (CustomerID, LocationID) VALUES (5, 5)
SELECT * FROM dual;


INSERT ALL
        INTO Payment (PaymentID, RentalID, PaymentMethod, Amount, PaymentDate) VALUES (1, 1, 'Credit Card', 150.00, TO_DATE('2022-01-05', 'YYYY-MM-DD'))
        INTO Payment (PaymentID, RentalID, PaymentMethod, Amount, PaymentDate) VALUES (2, 2, 'Debit Card', 75.00, TO_DATE('2022-01-10', 'YYYY-MM-DD'))
        INTO Payment (PaymentID, RentalID, PaymentMethod, Amount, PaymentDate) VALUES (3, 3, 'Cash', 65.00, TO_DATE('2022-02-12', 'YYYY-MM-DD'))
        INTO Payment (PaymentID, RentalID, PaymentMethod, Amount, PaymentDate) VALUES (4, 4, 'Credit Card', 40.00, TO_DATE('2022-02-20', 'YYYY-MM-DD'))
        INTO Payment (PaymentID, RentalID, PaymentMethod, Amount, PaymentDate) VALUES (5, 5, 'Debit Card', 80.00, TO_DATE('2022-03-01', 'YYYY-MM-DD'))
SELECT * FROM dual;
```