**Project**

**Student Name:** Saransh Arora **UID:** 24MCI10041

**Branch:** MCA(AIML) **Section/Group:** 1(B)

**Semester:** 1 **Date of Performance:** 20-10-2024

**Subject Name:** PL/SQL LAB **Subject Code:** 24CAP-602

**Project Title:- Car Rental System**

**Objective**

To design and implement a relational database for managing a car rental business. This system will allow management of customers, vehicles, rentals, payments, and reservations, providing efficient data storage, retrieval, and updates.

**Database Schema**

**1. Customers Table**

Stores customer information, including unique ID, name, contact, and address.

CREATE TABLE Customers (

customer\_id INT PRIMARY KEY,

customer\_name VARCHAR(100) NOT NULL,

contact\_number VARCHAR(15),

email VARCHAR(100) UNIQUE,

address VARCHAR(200)

);

**2. Vehicles Table**

Stores details of each vehicle available for rent, including ID, model, brand, year, daily rate, and status.

CREATE TABLE Vehicles (

vehicle\_id INT PRIMARY KEY,

model VARCHAR(50) NOT NULL,

brand VARCHAR(50) NOT NULL,

year\_of\_manufacture INT,

daily\_rate DECIMAL(10, 2),

status VARCHAR(15) DEFAULT 'Available'

);

**3. Rentals Table**

Records the details of each rental transaction, linking customers and vehicles. Includes rental date, return date, and total amount.

CREATE TABLE Rentals (

rental\_id INT PRIMARY KEY,

customer\_id INT,

vehicle\_id INT,

rental\_date DATE,

return\_date DATE,

total\_amount DECIMAL(10, 2),

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id),

FOREIGN KEY (vehicle\_id) REFERENCES Vehicles(vehicle\_id)

);

**4. Payments Table**

Stores payment information for each rental, including payment date and amount.

CREATE TABLE Payments (

payment\_id INT PRIMARY KEY,

rental\_id INT,

payment\_date DATE,

amount\_paid DECIMAL(10, 2),

FOREIGN KEY (rental\_id) REFERENCES Rentals(rental\_id)

);

**5. Reservations Table**

Records reservations made by customers, which may later be converted to rentals.

CREATE TABLE Reservations (

reservation\_id INT PRIMARY KEY,

customer\_id INT,

vehicle\_id INT,

reservation\_date DATE,

status VARCHAR(15) DEFAULT 'Pending',

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id),

FOREIGN KEY (vehicle\_id) REFERENCES Vehicles(vehicle\_id)

);

**Sample Data Insertion**

INSERT INTO Customers (customer\_id, customer\_name, contact\_number, email, address)

VALUES

(1, 'John Doe', '1234567890', 'johndoe@example.com', '123 Elm Street'),

(2, 'Jane Smith', '0987654321', 'janesmith@example.com', '456 Oak Avenue');

INSERT INTO Vehicles (vehicle\_id, model, brand, year\_of\_manufacture, daily\_rate)

VALUES

(101, 'Civic', 'Honda', 2021, 50.00),

(102, 'Model S', 'Tesla', 2022, 100.00);

INSERT INTO Rentals (rental\_id, customer\_id, vehicle\_id, rental\_date, return\_date, total\_amount)

VALUES

(1, 1, 101, '2023-01-01', '2023-01-05', 250.00),

(2, 2, 102, '2023-02-01', '2023-02-03', 200.00);

INSERT INTO Payments (payment\_id, rental\_id, payment\_date, amount\_paid)

VALUES

(1, 1, '2023-01-05', 250.00),

(2, 2, '2023-02-03', 200.00);

INSERT INTO Reservations (reservation\_id, customer\_id, vehicle\_id, reservation\_date, status)

VALUES

(1, 1, 102, '2023-02-10', 'Pending');

**Queries and Stored Procedures**

**1. Query: Retrieve All Available Vehicles**

List all vehicles that are available for rental.

SELECT \* FROM Vehicles

WHERE status = 'Available';

**2. Query: Display Rental History of a Customer**

Show all rentals made by a specific customer.

SELECT r.rental\_id, v.model, v.brand, r.rental\_date, r.return\_date, r.total\_amount

FROM Rentals r

JOIN Vehicles v ON r.vehicle\_id = v.vehicle\_id

WHERE r.customer\_id = 1;

**3. Stored Procedure: Reserve a Vehicle**

Create a stored procedure to reserve a vehicle for a customer.

CREATE PROCEDURE Reserve\_Vehicle (

IN p\_customer\_id INT,

IN p\_vehicle\_id INT

)

BEGIN

INSERT INTO Reservations (customer\_id, vehicle\_id, reservation\_date, status)

VALUES (p\_customer\_id, p\_vehicle\_id, CURDATE(), 'Reserved');

UPDATE Vehicles

SET status = 'Reserved'

WHERE vehicle\_id = p\_vehicle\_id;

END;

**4. Trigger: Update Vehicle Status on Return**

Automatically update the vehicle status to "Available" when a rental return date is recorded.

CREATE TRIGGER Update\_Vehicle\_Status\_After\_Return

AFTER UPDATE ON Rentals

FOR EACH ROW

BEGIN

IF NEW.return\_date IS NOT NULL THEN

UPDATE Vehicles

SET status = 'Available'

WHERE vehicle\_id = NEW.vehicle\_id;

END IF;

END;

**5. Query: Display Total Revenue from Rentals**

Calculate the total revenue generated from all rentals.

SELECT SUM(total\_amount) AS Total\_Revenue FROM Rentals;

**6. Stored Procedure: Finalize Rental Payment**

Record a payment for a rental and finalize the rental process.

CREATE PROCEDURE Finalize\_Rental\_Payment (

IN p\_rental\_id INT,

IN p\_amount\_paid DECIMAL(10, 2)

)

BEGIN

INSERT INTO Payments (rental\_id, payment\_date, amount\_paid)

VALUES (p\_rental\_id, CURDATE(), p\_amount\_paid);

UPDATE Rentals

SET return\_date = CURDATE()

WHERE rental\_id = p\_rental\_id;

END;

**Visualization of Clean Data**

You can create tables or use tools like Excel or SQL visualization tools to display:

* **Total Rentals per Customer**: Count of rentals per customer.
* **Vehicle Availability Status**: Display the number of available and reserved vehicles.
* **Monthly Revenue**: Total revenue generated each month.