

# **CASE STUDY in ICTC0923 – Database Management System**

## **A Database Design for a Vehicle Rental System**

Submitted by:

Regie A. Sese

Regee D. Casaña

Jean Claudine D. Anilao

Alexis Emmanuel R. Cruz

Arnie James H. Manansala

Section:

BSCS-SD2B

Submitted to:

Marissa B. Ramos, MIT

Instructor - ICTC0923

Date Submitted:

July 6, 2022

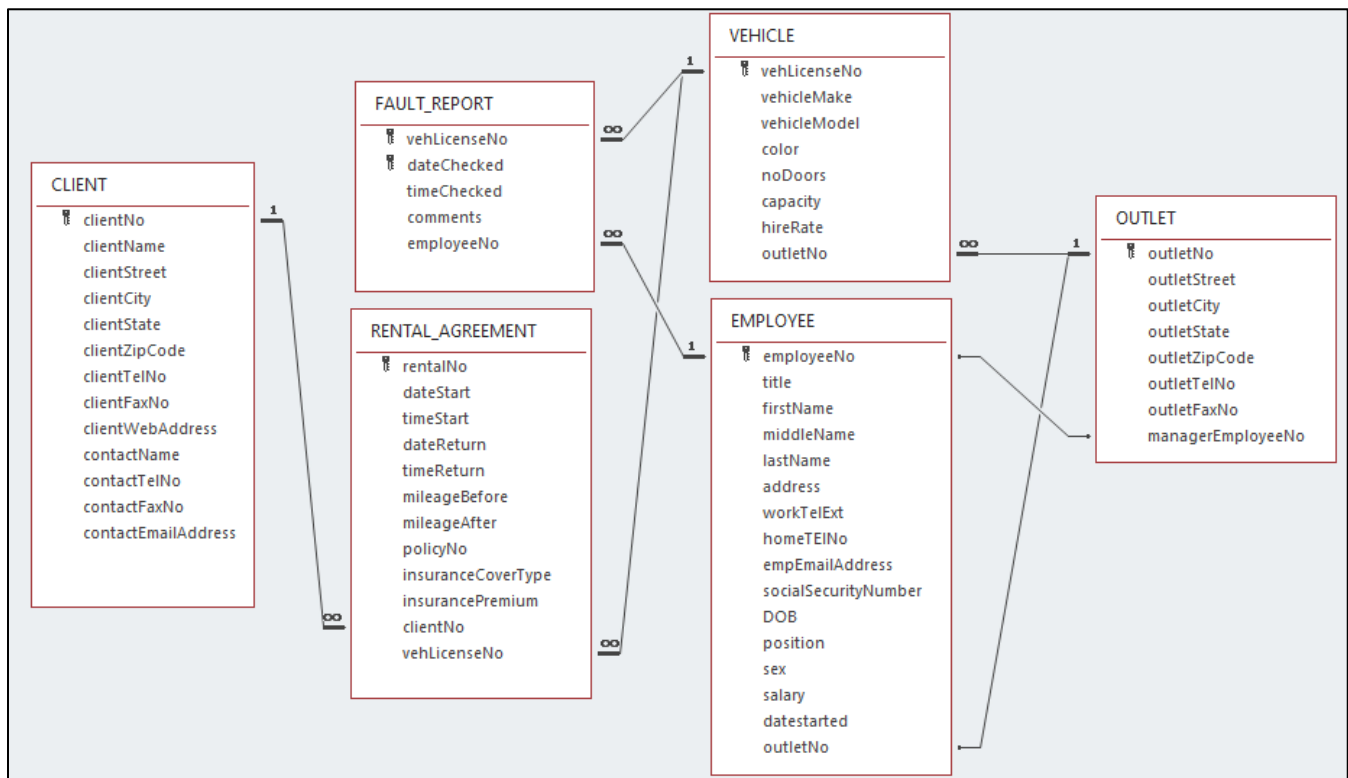
## Introduction

The Vehicle Rental System is a website that rents automobiles for a fee for short periods of time. Customers will be able to reserve their automobiles from anywhere thanks to the development of the Vehicle Rental System. By entering their information, users of this program will create a record of their information and access the website. A consumer who has registered on the website has the option to reserve the vehicle he needs. Customers are helped by this system, which offers to fill out the information as needed. It contains the kind of car they're attempting to rent as well as the dates before and after. Aside from the basic rental of a vehicle, car rental agencies typically offer additional products such as insurance. The user needs to input their personal information, and the rental agreement if they are going to rent a vehicle.

This system's goal is to provide a website where users may manually reserve a vehicle and specify the vehicle they need. It includes assembling many vehicle types on a single platform. A user can search for a car they want to rent based on their needs and reserve it by providing the dates they want to rent the vehicle from and the date they want to return it to the company. The vehicle Rental System is an access that primarily assist individuals who have a car that is briefly out of reach or out of service, such as out-of-town travelers or owners of destroyed or damaged vehicles awaiting repair or insurance compensatory damages. The main goal of our Vehicle Rental System is to keep track of information about the Car, Payment, Customer, Outlet, and Insurance. It keeps track of everything about Car, Insurance, Car. The system makes it possible for an individual to reserve/book a vehicle over the Internet, while the company's staff takes care of the transactions.

## Database Design

### A. Entity Relationship Diagram



The Entity Relationship Diagram of the Vehicle Rental System consists of a total of 6 tables. The following table names are as follows; Client, Outlet, Employee, FaultReport, Vehicle, and RentalAgreement.

Each table in the database stores different data. The client table is connected to the RentalAgreement table which determine which customer rented a vehicle. The FaultReport table is connected to Employee table to determine who is the employee in charge of checking the fault, and reporting it to the company. Also, the FaultReport table is connected to the Vehicle table which determine which vehicle is faulty and needs to repair. The RentalAgreement table is also connected to the Vehicle table to determine which vehicle the customer has selected to rent. The last table that is connected to Vehicle table is the Outlet table. It determines which outlet does the vehicle rented came from. Lastly, the Employee table is connected to the Outlet table to determine which employee is the manager in charge in the outlet.

## B. Data Dictionary

<b>Subject: Client</b>						
<b>Field Name</b>	<b>Data Type</b>	<b>Size</b>	<b>PK</b>	<b>FK</b>	<b>Null</b>	<b>Description</b>
clientNo	NUMBER	8	YES	NO	NO	Client's Number
clientName	VARCHAR2	50	NO	NO	NO	Client's name
clientStreet	VARCHAR2	40	NO	NO	NO	Client's Street
clientCity	VARCHAR2	15	NO	NO	NO	Client's City
clientState	VARCHAR2	15	NO	NO	NO	Client's State
clientZipCode	VARCHAR2	9	NO	NO	YES	Client's Zip Code
clientTelNo	VARCHAR2	20	NO	NO	NO	Client's Telephone Number
clientFaxNo	VARCHAR2	20	NO	NO	NO	Client's Fax. Number

ClientWebAddress	VARCHAR2	100	NO	NO	YES	Client's Web Address
contactName	VARCHAR2	50	NO	NO	NO	Contact Name
contactTelNo	VARCHAR2	20	NO	NO	YES	Contact Telephone Number
contactFaxNo	NUMBER	8	NO	NO	YES	Contact Fax Number
contactEmailAddress	VARCHAR2	50	NO	NO	YES	Contact Email Address

<b>Subject: Outlet</b>						
Field Name	Data Type	Size	PK	FK	Null	Description
outletNo	NUMBER	8	YES	NO	NO	Outlet Number
outletStreet	VARCHAR2	40	NO	NO	NO	Outlet Street
outletCity	VARCHAR2	15	NO	NO	NO	Outlet City
outletState	VARCHAR2	15	NO	NO	NO	Outlet State
outletZipCode	VARCHAR2	9	NO	NO	NO	Outlet Zip Code
outletTelNo	VARCHAR2	20	NO	NO	NO	Outlet Telephone Number
outletFaxNo	VARCHAR2	20	NO	NO	NO	Outlet Fax. Number

managerEmployeeNo	NUMBER	8	NO	YES	YES	Manger Employee Number
-------------------	--------	---	----	-----	-----	------------------------

<b>Subject:</b> Employee						
Field Name	Data Type	Size	PK	FK	Null	Description
employeeNo	NUMBER	8	YES	NO	NO	Employee Number
title	VARCHAR2	10	NO	NO	NO	Employee Title
firstName	VARCHAR2	30	NO	NO	NO	Employee's First Name
middleName	VARCHAR2	30	NO	NO	YES	Employee's Middle Name
lastName	VARCHAR2	30	NO	NO	NO	Employee's Last Name
address	VARCHAR2	80	NO	NO	NO	Employee's Address
workTelExt	VARCHAR2	4	NO	NO	NO	Work Telephone Extension
homeTelNo	VARCHAR2	20	NO	NO	NO	Home Telephone Number
empEmailAddress	VARCHAR2	100	NO	NO	YES	Employee's Email Address
socialSecurityNumber	VARCHAR2	12	NO	NO	NO	Employee's Social Security Number

DOB	DATE	None	NO	NO	NO	Employee's date of birth
position	VARCHAR2	20	NO	NO	NO	Employee's Position
sex	CHAR	None	NO	NO	NO	Employee's Sex
salary	NUMBER	8,2	NO	NO	NO	Employee's Salary
dateStarted	DATE	None	NO	NO	NO	The date the employee started
outletNo	NUMBER	8	NO	YES	NO	Employee's outlet Number

<b>Subject:</b> Vehicle						
Field Name	Data Type	Size	PK	FK	Null	Description
vehiLicenseNo	VARCHAR	10	YES	NO	NO	Vehicle License Number
vehicleMake	VARCHAR	15	NO	NO	NO	Vehicle Make
vehicleModel	VARCHAR	15	NO	NO	NO	Vehicle Model
color	VARCHAR	15	NO	NO	NO	Vehicle Color
noDoors	NUMBER	1	NO	NO	NO	Number of Doors
capacity	NUMBER	4	NO	NO	NO	Vehicle Capacity
hireRate	NUMBER	6, 2	NO	NO	NO	Vehicle Hire Rate

outletNo	NUMBER	8	NO	YES	NO	Vehicle Outlet Number
----------	--------	---	----	-----	----	-----------------------

<b>Subject:</b> FaultReport						
Field Name	Data Type	Size	PK	FK	Null	Description
vehLicenseNo	VARCHAR	10	YES	YES	NO	Vehicle License Number
dateChecked	DATE	None	YES	NO	NO	Date where the vehicle fault checked
timeChecked	NUMBER	4, 2	NO	NO	NO	Time where the vehicle fault checked
comments	VARCHAR	50	NO	NO	NO	Comments
employeeNo	NUMBER	8	NO	YES	NO	Employee Number of the one who check the vehicle.

<b>Subject:</b> RentalAgreement						
Field Name	Data Type	Size	PK	FK	Null	Description
rentalNo	NUMBER	8	YES	NO	NO	Rental Number
dateStart	DATE	DATE	NO	NO	NO	Date where the rent start
timeStart	NUMBER	4, 2	NO	NO	NO	Time where the rent start
dateReturn	DATE	None	NO	NO	YES	The date vehicle returned

timeReturn	NUMBER	4, 2	NO	NO	YES	The Time vehicle Returned
mileageBefore	NUMBER	8	NO	NO	NO	Mileage Before the vehicle rented
mileageAfter	NUMBER	8	NO	NO	YES	Mileage After returned the vehicle
policyNo	NUMBER	15	NO	NO	NO	Policy Number
insuranceCoverType	VARCHAR2	10	NO	NO	NO	The type of insurance it will cover
insurancePremium	NUMBER	5, 2	NO	NO	NO	Insurance Premium
clientNo	NUMBER	8	NO	YES	NO	The Client's Number who will rent
vehLicenseNo	VARCHAR2	10	NO	YES	NO	Vehicle License Number

## SQL Statements

### A. Data Definition Language

```

CREATE TABLE Client(
  clientNo NUMBER(8) NOT NULL,
  clientName VARCHAR2(50) NOT NULL UNIQUE,
  clientStreet VARCHAR2(40) NOT NULL,
  clientCity VARCHAR2(15) NOT NULL,
  clientState VARCHAR2(15) NOT NULL,
  clientZipCode VARCHAR2(9),
  clientTelNo VARCHAR2(20) NOT NULL UNIQUE,
  clientFaxNo VARCHAR2(20) NOT NULL UNIQUE,
  clientWebAddress VARCHAR2(100),
  contactName VARCHAR2(50) NOT NULL,
  contactTelNo VARCHAR2(20),
  contactFaxNo VARCHAR2(20),
  contactEmailAddress VARCHAR2(100),

```



PRIMARY KEY (clientNo)  
);

CREATE TABLE Outlet(  
outletNo NUMBER(8) NOT NULL,  
outletStreet VARCHAR2(40) NOT NULL,  
outletCity VARCHAR2(15) NOT NULL,  
outletState VARCHAR2(15) NOT NULL,  
outletZipCode VARCHAR2(9) NOT NULL,  
outletTelNo VARCHAR2(20) NOT NULL UNIQUE,  
outletFaxNo VARCHAR2(20) NOT NULL UNIQUE,  
managerEmployeeNo NUMBER(8),  
PRIMARY KEY (outletNo)  
);

CREATE TABLE Employee(  
employeeNo NUMBER(8) NOT NULL,  
title VARCHAR2(10) NOT NULL,  
firstName VARCHAR2(30) NOT NULL,  
middleName VARCHAR2(30),  
lastName VARCHAR2(30) NOT NULL,  
address VARCHAR2(80) NOT NULL,  
workTelExt VARCHAR2(4) NOT NULL,  
homeTelNo VARCHAR2(20) NOT NULL,  
empEmailAddress VARCHAR2(100),  
socialSecurityNumber VARCHAR2(12) NOT NULL UNIQUE,  
DOB DATE NOT NULL,  
position VARCHAR2(20) NOT NULL,  
sex CHAR NOT NULL,  
salary NUMBER(8,2) NOT NULL,  
dateStarted DATE NOT NULL,  
outletNo NUMBER(8) NOT NULL,  
PRIMARY KEY (employeeNo),  
CONSTRAINT outletnumber FOREIGN KEY (outletNo) REFERENCES Outlet(outletNo)  
);

CREATE TABLE Vehicle(  
vehLicenseNo VARCHAR2(10) NOT NULL,  
vehicleMake VARCHAR2(15) NOT NULL,  
vehicleModel VARCHAR2(15) NOT NULL,  
color VARCHAR2(15) NOT NULL,  
noDoors NUMBER(1) NOT NULL,  
capacity NUMBER(4) NOT NULL,  
hireRate NUMBER(6, 2) NOT NULL,  
outletNo NUMBER(8) NOT NULL,  
PRIMARY KEY (vehLicenseNo),  
CONSTRAINT outletnumber1 FOREIGN KEY (outletNo) REFERENCES Outlet(outletNo)  
);

```

CREATE TABLE FaultReport(
vehLicenseNo VARCHAR2(10) NOT NULL,
dateChecked DATE NOT NULL,
timeChecked NUMBER(4, 2) NOT NULL,
comments VARCHAR2(50) NOT NULL,
employeeNo NUMBER(8) NOT NULL,
PRIMARY KEY (vehLicenseNo, dateChecked),
CONSTRAINT vehiclelicensenummer FOREIGN KEY (vehLicenseNo) REFERENCES
Vehicle(vehLicenseNo),
CONSTRAINT employeenumber FOREIGN KEY (employeeNo) REFERENCES
Employee(employeeNo)
);

```

```

CREATE TABLE RentalAgreement(
rentalNo NUMBER(8) NOT NULL,
dateStart DATE NOT NULL,
timeStart NUMBER(4, 2) NOT NULL,
dateReturn DATE,
timeReturn NUMBER(4, 2),
mileageBefore NUMBER(8) NOT NULL,
mileageAfter NUMBER(8),
policyNo NUMBER(15) NOT NULL UNIQUE,
insuranceCoverType VARCHAR2(10) NOT NULL,
insurancePremium NUMBER(8, 2) NOT NULL,
clientNo NUMBER(8) NOT NULL,
vehLicenseNo VARCHAR2(10) NOT NULL,
PRIMARY KEY (rentalNo),
CONSTRAINT clientnumber FOREIGN KEY (clientNo) REFERENCES Client(clientNo),
CONSTRAINT vehiclelicensenummer1 FOREIGN KEY (vehLicenseNo) REFERENCES
Vehicle(vehLicenseNo)
);

```

## B. Data Manipulation Language / Views / PL/SQL Blocks / Stored Procedures

### //INSERTING RECORDS TO THE TABLES USING PROCEDURE AND PL/SQL BLOCKS

//Creating sequence sqclientID for auto incrementing the clientNo in the table Client

```

CREATE SEQUENCE sqclientID
MINVALUE 1
START WITH 1
INCREMENT BY 1;

```

//User input in inserting client information

```

DECLARE
varclientNo NUMBER;
varclientName VARCHAR2(50) := :Client_Name;
varclientSt VARCHAR2(50) := :Client_Street;
varclientCity VARCHAR2(15) := :Client_City;
varclientState VARCHAR2(15) := :Client_State;
varclientZipCode VARCHAR2(9) := :Client_ZipCode;

```

```

varclientTelNo VARCHAR2(20) := :Client_Telephone;
varclientFaxNo VARCHAR2(20) := :Client_FaxNumber;
varclientWebAdd VARCHAR(100) := :Client_WebAddress;
varcontactName VARCHAR2(50) := :Contact_Name;
varcontactTelNo VARCHAR2(20) := :Contact_Telephone;
varcontactFaxNo VARCHAR2(20) := :Contact_FaxNumber;
varcontactEmailAdd VARCHAR2(100) := :Contact_EmailAdd;
BEGIN
    INSERT INTO client VALUES (sqclientID.NEXTVAL, varclientName, varclientSt,
varclientCity, varclientState, varclientZipCode, varclientTelNo, varclientFaxNo, varclientWebAdd,
varcontactName, varcontactTelNo, varcontactFaxNo, varcontactEmailAdd);
END;

```

**//Creating sequence sqoutletID for auto incrementing the outletNo in the table outlet**

```

CREATE SEQUENCE sqoutletID
MINVALUE 1
START WITH 1
INCREMENT BY 1;

```

**//Creating stored procedure outletrec for table OUTLET**

```

CREATE OR REPLACE PROCEDURE outletrec(
    varoutletStreet VARCHAR,
    varoutletCity VARCHAR2,
    varoutletState VARCHAR2,
    varoutletZipCode VARCHAR2,
    varoutletTelNo VARCHAR2,
    varoutletFaxNo VARCHAR2,
    varmanagerEmployeeNo NUMBER
)
IS
BEGIN
    INSERT INTO OUTLET VALUES (sqoutletID.NEXTVAL, varoutletStreet, varoutletCity,
varoutletState, varoutletZipCode, varoutletTelNo, varoutletFaxNo, varmanagerEmployeeNo);
END;
/

```

**//Call procedure outletrec and insert records using the parameter**

```

BEGIN
    outletrec('Silangan', 'Balanga City', 'Bataan', '2100', '639123456788', '555-555-555', '1');
    outletrec('Looban', 'Alaminos', 'Pangasinan', '2404', '639123456782', '555-555-556', '2');
    outletrec('Santiago', 'Iba', 'Zambales', '2201', '639123456781', '555-555-557', '3');
END;

```

**//Creating sequence fsqempID for auto incrementing the EmployeeNo in the table outlet**

```

CREATE SEQUENCE sqempID
MINVALUE 1
START WITH 1
INCREMENT BY 1;

```

**//Creating procedure employeeRec for inserting records in employeeRec**

```
CREATE OR REPLACE PROCEDURE employeeRec(
    vartitle VARCHAR2,
    varfirstName VARCHAR2,
    varmiddleName VARCHAR2,
    varlastName VARCHAR2,
    varaddress VARCHAR2,
    varworkTelExt VARCHAR2,
    varhomeTelNo VARCHAR2,
    varempEmailAddress VARCHAR2,
    varsocialSecurityNumber VARCHAR2,
    varDOB DATE,
    varposition VARCHAR2,
    varsex CHAR,
    varsalary NUMBER,
    vardateStarted DATE,
    varoutletNo NUMBER)
IS
BEGIN
    INSERT INTO Employee VALUES (sqempID.NEXTVAL, vartitle, varfirstName,
    varmiddleName, varlastName, varaddress, varworkTelExt, varhomeTelNo, varempEmailAddress,
    varsocialSecurityNumber, varDOB, varposition, varsex, varsalary, vardateStarted, varoutletNo);
END;
```

**//Call procedure employeerec and insert values using the parameter**

```
BEGIN
    employeeRec('Manager', 'Juan', 'Garcia', 'Dela Cruz', 'Balanga City Bataan', '1234',
    '09123456781', 'Example@sample.com', '013-12-999', '01-JAN-1990', 'Admin', 'M', '100000', '01-
    JAN-2000', '1');
    employeeRec('Manager', 'Arnia', 'Sy', 'Gatdula', 'Alaminos Pangasinan',
    '1235', '09123456780', 'Example@sample.com', '013-12-998', '02-JAN-1990', 'Admin', 'F', '100001',
    '02-JAN-2000', '2');
    employeeRec('Manager', 'Pedro', 'Kabisote', 'Penduko', 'Iba Zambales',
    '1236', '09123456782', 'Example@sample.com', '013-12-997', '03-JAN-1990', 'Admin', 'M', '100002',
    '03-JAN-2000', '3');
END;
```

**//Alter table outlet for foreign key connection with employee table**

```
ALTER TABLE Outlet
ADD CONSTRAINT manageremployeenumber FOREIGN KEY (managerEmployeeNo)
REFERENCES Employee(employeeNo);
```

**//The table outlet is now connected to the employee table**

**//User input in inserting values in vehicle table**

```
DECLARE
    varvehicleLicenseNo VARCHAR2(10) := :Vehicle_LicenseNumber;
    varvehicleMake VARCHAR2(15) := :Make;
    varvehicleModel VARCHAR2(15) := :Model;
```

```

        varcolor VARCHAR2(15) := :Color;
        varnoDoors NUMBER(1) := :Doors_Number;
        varcapacity NUMBER(4) := :Capacity;
        varhireRate NUMBER(6,2) := :Hire_Rate;
        varoutletNo NUMBER(8) := :Outlet_Number;
BEGIN
        INSERT INTO VEHICLE VALUES(varvehicleLicenseNo, varvehicleMake,
varvehicleModel, varcolor, varnoDoors, varcapacity, varhireRate, varoutletNo);
END;

```

#### **//User input in inserting records in FaultReport table**

```

DECLARE
        varvehLicenseNo VARCHAR2(10) := :Vehicle_LicenseNumber;
        vardateChecked DATE := :Date_Checked;
        vartimeChecked NUMBER(4, 2) := :Time_Checked;
        varcomments VARCHAR2(50) := :Comments;
        varemployeeNo NUMBER(8) := :Employee_Number;
BEGIN
        INSERT INTO FAULTREPORT VALUES (varvehLicenseNo, vardateChecked,
vartimeChecked, varcomments, varemployeeNo);
END;

```

#### **//Creating sequence sqrentID for auto incrementing the RentalNo in the table RentalAgreement**

```

CREATE SEQUENCE sqrentID
MINVALUE 1
START WITH 1
INCREMENT BY 1;

```

#### **//User Input in inserting records in the table RentalAgreement**

```

DECLARE
        vardateStart DATE := :RentalDate_Start;
        vartimeStart NUMBER(4, 2) := :RentalTime_Start;
        vardateReturn DATE := null;
        vartimeReturn NUMBER(4,2) := null;
        varmileageBefore NUMBER(8) := :Mileage_BeforeRent;
        varmileageAfter NUMBER(8) := 0;
        varpolicyNo NUMBER(15) := :Policy_Number;
        varinsuranceCoverType VARCHAR2(10) := :Insurance_CoverType;
        varinsurancePremium NUMBER(8, 2) := :Insurance_Premium;
        varclientNo NUMBER(8) := :Client_Number;
        varvehLicenseNo VARCHAR2(10) := :Vehicle_LicenseNo;
BEGIN
        INSERT INTO RENTALAGREEMENT (rentalNo, dateStart, timeStart, dateReturn,
timeReturn, mileageBefore, mileageAfter, policyNo, insuranceCoverType, insurancePremium,
clientNo, vehLicenseNo)

```

```
VALUES (sqrentID.NEXTVAL, vardateStart, vartimeStart, vardateReturn, vartimeReturn,
varmileageBefore, varmileageAfter, varpolicyNo, varinsuranceCoverType, varinsurancePremium,
varclientNo, varvehLicenseNo);
END;
```

## //ADDITIONAL QUERIES

### //Creating a procedure that allows the user to return the vehicle rented

```
CREATE OR REPLACE PROCEDURE return_vehicle(
    varrentalID NUMBER,
    varreturndate DATE,
    varreturntime NUMBER,
    varmileageafter NUMBER
) IS
BEGIN
    UPDATE RENTALAGREEMENT
    SET dateReturn = varreturndate, timeReturn = varreturntime, mileageAfter =
varmileageafter
    WHERE rentalNo = varrentalID;
END;
```

### //Calling the procedure return\_vehicle and accept the user input for returning the vehicle

```
DECLARE
    varrentalID NUMBER := :Rental_ID;
    varreturndate DATE := :Date_Returned;
    varreturntime NUMBER(4,2) := :Time_Returned;
    varmileageafter NUMBER := :Mileage_After;
BEGIN
    return_vehicle(varrentalID, varreturndate, varreturntime, varmileageafter);
END;
```

### //Creating procedure that will greet the Client

```
CREATE OR REPLACE PROCEDURE client_greet(clientID NUMBER)
IS
    varclientno client.clientNo%type := clientID;
    varname client.clientName%type;
BEGIN
    SELECT clientName into varname from client where clientNo = varclientno;
    DBMS_OUTPUT.put_line('Hello '||varname||'! Welcome to Vehicle Rental System!');
END client_greet;
```

### //Calling the procedure client\_greet

```
DECLARE
    varclientID NUMBER := :Enter_ClientNo;
BEGIN
    client_greet(varclientID);
END;
```

#### **//Creating procedure that will greet the Employee**

```
CREATE OR REPLACE PROCEDURE Employee_greet(empID NUMBER)
IS
    varempno Employee.EmployeeNo%type := empID;
    vartitle Employee.title%type;
    varname Employee.lastName%type;
BEGIN
    SELECT lastName into varname from employee where employeeNo = varempno;
    SELECT title into vartitle from employee where employeeNo = varempno;
    DBMS_OUTPUT.put_line('Hello '||vartitle|| ' '||varname||'! Thank you for doing your job!');
END Employee_greet;
```

#### **//Calling the procedure Employee\_greet**

```
DECLARE
    varempID NUMBER := :Enter_EmployeeNo;
BEGIN
    Employee_greet(varempID);
END;
```

#### **//VIEWS**

##### **//Creating view for displaying table client and the date they rented and returned a vehicle**

```
CREATE OR REPLACE VIEW vwclientdisplay AS
SELECT c.clientName, (c.clientZipCode|| ' '||c.clientCity||, ' '||c.clientState)"Address",
c.clientwebAddress, c.clientTelNo, r.dateStart, r.dateReturn FROM CLIENT c FULL JOIN
RENTALAGREEMENT r
ON c.clientNo = r.clientNo;
```

##### **//Display the vwclientdisplay**

```
SELECT * FROM vwclientdisplay;
```

##### **//Creating view for displaying table Outlet and their manager**

```
CREATE OR REPLACE VIEW vwoutletdisplay
AS SELECT e.title, (e.firstname||' '||e.lastname) "Full Name", c.outletCity, c.outletState,
c.outletZipcode
FROM Outlet c FULL JOIN Employee e
ON c.managerEmployeeNo= e.EmployeeNo
```

##### **//Display the vwoutletdisplay**

```
SELECT * FROM vwoutletdisplay;
```

##### **//Creating view for displaying the managers in table employee**

```
CREATE OR REPLACE VIEW vwempdisplay
AS SELECT * FROM employee
WHERE title = 'Manager';
```

##### **//Display the vwempdisplay**

```
SELECT * FROM vwempdisplay;
```

**//Creating view for displaying table Vehicle and their outlet**

```
CREATE OR REPLACE VIEW vwvehicledisplay AS
SELECT (o.outletZipCode||' '||o.outletCity||', '||o.outletState)"Outlet Address", v.vehiclemake,
v.vehicleModel, v.vehLicenseNo
FROM VEHICLE v FULL JOIN OUTLET o
ON v.outletNo = o.outletNo;
```

**//Display the vwvehicledisplay**

```
SELECT * FROM vwvehicledisplay;
```

**//Creating view for displaying table FaultReport**

```
CREATE OR REPLACE VIEW vwfaultdisplay AS
SELECT f.vehLicenseNo, f.dateChecked, f.timeChecked, f.comments, (v.vehicleMake||' '
||v.vehicleModel)"Vehicle"
FROM FaultReport f, vehicle v
WHERE v.vehLicenseNo = f.vehLicenseNo;
```

**//Display the vwfaultdisplay**

```
SELECT * FROM vwfaultdisplay;
```

**//Creating view for displaying table RentalAgreement and the client name**

```
CREATE OR REPLACE VIEW vwRentdisplay AS
SELECT c.clientName, r.rentalNo, r.dateStart, r.timeStart, r.dateReturn, r.timeReturn,
r.mileageBefore, r.mileageAfter, r.policyNo, r.insuranceCoverType, r.insurancePremium,
r.clientNo, r.vehLicenseNo
FROM RentalAgreement r, client c
WHERE c.clientNo = r.clientNo;
```

**//Display the vwRentdisplay**

```
SELECT * FROM vwrentdisplay;
```