Part 1: Generate A Case Scenario & Draw An Erd (10 Marks)....... 2

a) With the help of any AI generative tool (e.g., ChatGPT),....... 2

b) Use draw.io software to draw the ERD using Crow foot notation with no attributes. If an optionality is required, please indicate it in the diagram....... 2

c) List at least TEN business rules for your environment....... 3

d) Number of words for the scenario should be at least 500 words....... 4

Part 2: Design A Relational Data Model (35 Marks)....... 6

a) Relational Database Model (15 marks)....... 6

b) Constraints (10 marks)....... 6

c) Indexes (10 marks)....... 6

Important Notes:....... 6

Part 3: Implement A Database (15 Marks)....... 7

a) Write a script named DBscript to create and populate a database using ORACLE DBMS....... 7

b) Populate the database - create data of your choice by using the following as a guide:....... 7

Part 4: Query A Database (Each Sub-Part Carries 4 Marks; Max Of 20 Marks)....... 8

Please Refer To Appendix 1 For A Full Example....... 8

a....... 8

b....... 8

c....... 8

d....... 8

e....... 8

Important Notes:....... 8

Summary Of Tasks And Deliverables....... 10

APPENDIX 1 (An example of Part 4 Question and Answer)....... 11

QUESTION....... 11

ANSWER....... 11

APPENDIX 2....... 12

Final assessment RUBRIC....... 12

Grade Scale:....... 12

Part 1: Scenario and Business Rules (10 Marks)....... 12

Part 2: Tables and Answers (35 Marks)....... 12

Part 3: Database Script (15 Marks)....... 12

Part 4: Scenarios and SQL Code (20 Marks)....... 13

Part 5: Presentation and Practical Tasks (20 Marks)....... 13

Part 1: Generate A Case Scenario & Draw An Erd (10 Marks)

a) With the help of any AI generative tool (e.g., ChatGPT),

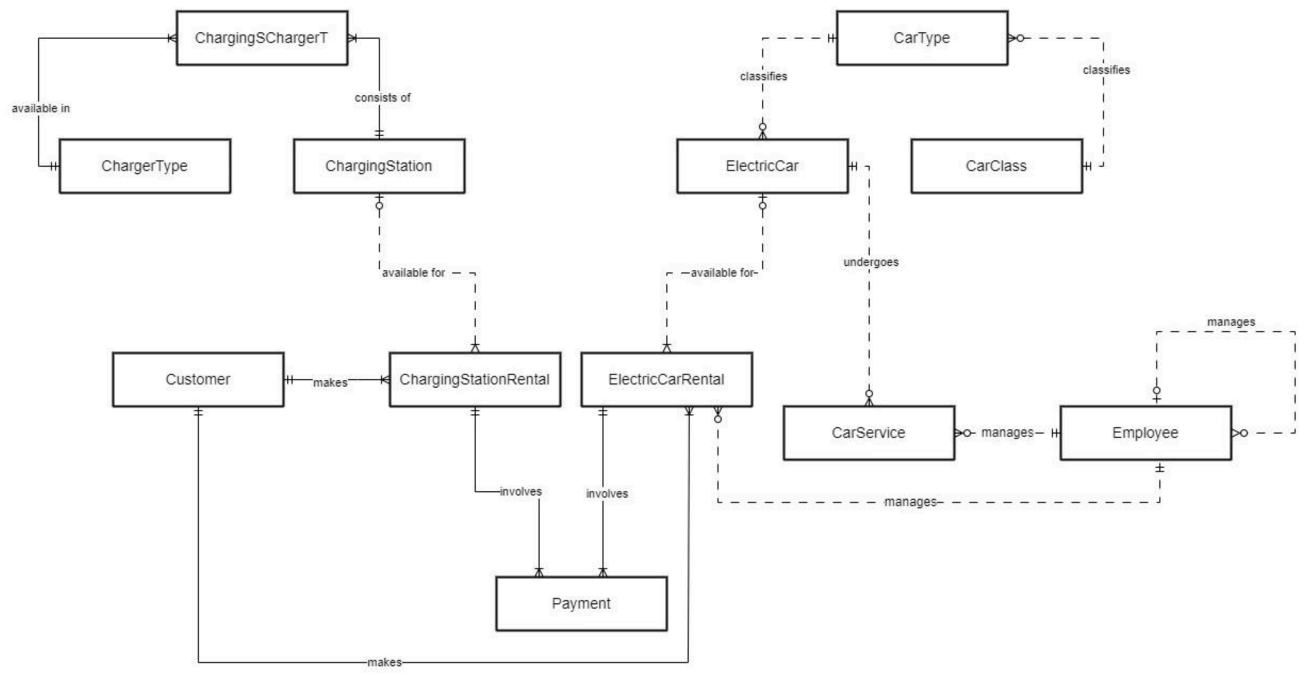
- You can choose ONE of the following environments.
  - Electric cars
  - Travel agency
  - Airbnb

Answer:

Electric Cars Case Scenario

b) Use draw.io software to draw the ERD using Crow foot notation with no attributes. If an optionality is required, please indicate it in the diagram.

Answer:



c) List at least TEN business rules for your environment.

Answer:

1. Each electric car belongs to only one car class.
2. Each car class can include many electric cars.
3. Each charging station can have multiple types of chargers.
4. Each charger type can be present in multiple charging stations.
5. Each employee has only one role.
6. Each non-manager employee is assigned to only one manager.
7. Each manager can supervise at most 50 employees.
8. All managers only report to the business owner.
9. Clerks only manage rentals of electric cars, not rentals of charging stations.
10. Mechanics only handle the electric car’s maintenance, not the charging stations.
11. Each customer can only rent one car and one charging station at any given time.
12. Deposits must be paid in full before the rental period begins.
13. The deposit fee is refundable upon the return of the vehicle in good condition.
14. Late returns are subject to additional fees, which are calculated based on the rental agreement.

d) The number of words for the scenario should be at least 500 words.

Answer:

Case Scenario: EcoDrive Electric Car Rental Management System

EcoDrive is a company that provides electric cars and charging stations to customers for rent.

Each electric car has a unique identifier (carVIN), registration number, make (e.g., Tesla), model (e.g., Model S), battery capacity (in kWh), mileage, and the kilometer reading of the last service. An electric car's type is determined by the make and model and is assigned a class.

Each car class has a unique identifier (classID), description, daily hire rate, deposit, and daily late return fee. A given car class will potentially cover many different car types. The car class is used to determine the daily hire rate and the deposit for the car rental. For example, a Class 1001 car has a daily hire rate of RM300/day, a deposit of RM500, and a daily late return fee of RM450, whereas a Class 1002 car has a daily hire rate of RM140/day, a deposit of RM350 and a daily late return fee of RM210.

The company also rents our charging stations. Each charging station has a unique identifier (stationID), location, and number of charging points. EcoDrive's charging stations can only be rented by their customers. The charging stations have different types of chargers (e.g., Type 1, Type 2, CHAdeMO) to accommodate different types of electric cars.

The date of purchase, purchase cost, and date of last maintenance are both recorded for electric cars and charging stations. All electric cars and charging stations are purchased separately.

The company has several employees with different roles such as manager, clerk, and mechanic. Each employee is identified using an employee ID. Their details including their family name, given name, contact number, address, Tax Identification Number (TIN), and salary are recorded in the system. Every non-manager employee is assigned to a manager and all managers report to the business owner. The mechanic handles the electric car's maintenance but the charging station's maintenance is handled by external professionals hired by the business owner.

Clerks manage the rental of vehicles but not the rental of charging stations. A new customer will have to provide their details such as the customer's family name, given name, contact number, address, and business name (if applicable), and will then be assigned a unique customer ID to be added to the system for easy future rentals. Customers are also required to provide their driver's license and their identification (MyKad for Malaysians and passport for non-Malaysians) before they are allowed to pick a preferred vehicle. After picking their preferred vehicle, the customer will have to specify their preferred rental start date and the preferred rental end date. The clerk will then assist the customer with deposits, payment, and the signing of a rental agreement.

The same goes for renting charging stations. New customers will have to provide their details such as the customer's family name, given name, contact number, address, and business name (if applicable), and they will then be assigned a unique customer ID to be added to the system for easy future rentals. Customers are also required to provide their driver's license and their identification (MyKad for Malaysians and passport for non-Malaysians).

Each Rental will have its unique rentalID, carVIN or stationID, time rented or preferred rental start date and preferred return date, or damage fee. Payment can be made in either cash, debit card, or credit card. Each payment has its own payment ID, date and time occurred, rental ID, payment type(deposit, rental), and payment method.

Part 2: Design A Relational Data Model (35 Marks)

- This part consists of THREE sub-parts; RDM, constraints, and indexes.
- With the help of the business rules stated in Part 1, do the following:

a) Relational Database Model (15 marks)

- Identify the primary & and alternate keys, and foreign keys (if any) for each table.
- List the attributes of each entity.

Answer:

Identify the primary & and alternate key, and foreign (if any) keys for each table. List the attributes of each entity.

ElectricCar *carVIN - PK* *regNum* *battCap* *mileage* *kmReadingLastService* *carPurchaseDate* *carPurchaseCost* *carLastMaintenanceDate* *carTypeID* *FOREIGN KEY (carTypeID) REFERENCES CarType(carTypeID)* *ALTERNATE KEY (regNum)*	***CarType*** *CarTypeID - PK* *carMake* *carModel* *classID* *FOREIGN KEY (ClassID) REFERENCES CarClass(ClassID)*
CarClass *classID - PK* *carDesc* *carDailyHireRate* *carDeposit* *dailyLateReturnFee*	***ChargingStation*** *stationID - PK* *stationLocation* *numChargingPoints* *stationPurchaseDate* *stationPurchaseCost* *stationLastMaintenanceDate* *stationRentalHourlyRate* *ALTERNATE KEY (stationLocation)*

ChargerType *chargerTypeID - PK* *chargerDesc* *ALTERNATE KEY (chargerDesc)*	***ChargingSChargerT*** *stationID - PK* *chargerTypeID - PK* *FOREIGN KEY (stationID) REFERENCES ChargingStation(stationID)* *FOREIGN KEY (chargerTypeID) REFERENCES ChargerType(chargerTypeID)*
Employee *empID - PK* *empFName* *empGName* *empContactNum* *empAddress* *TIN* *salary* *empRole* *managerID* *FOREIGN KEY (managerID) REFERENCES Employee(empID)* *ALTERNATE KEY (TIN)* *ALTERNATE KEY (empContactNum)*	***Customer*** *custID - PK* *custFName* *custGName* *custContactNum* *custAddress* *businessName* *driverLicenseNum* *IDType* *IDNum* *ALTERNATE KEY (driverLicenseNum)* *ALTERNATE KEY (IDNum)*
ElectricCarRental *carRentalID - PK* *carRentalStartDate* *carRentalEndDate* *carRentalCost* *carVIN* *custID* *empID* *damagefee* *FOREIGN KEY (carVIN) REFERENCES ElectricCar(carVIN)*	***ChargingStationRental*** *stationRentalID - PK* *stationRentalHours* *stationID* *custID* *FOREIGN KEY (stationID) REFERENCES ChargingStation(stationID)* *FOREIGN KEY (custID) REFERENCES Customer(custID)* *FOREIGN KEY (empID) REFERENCES Employee(empID)*

FOREIGN KEY (custID) REFERENCES Customer(custID) *FOREIGN KEY (empID) REFERENCES Employee(empID)* *CHECK (carRentalStartDate < carRentalEndDate)* *CHECK (carRentalStartDate < carRentalReturnDate)*	
RentalPayment *<u>paymentID - PK</u>* *paymentDateTime* *paymentType* *paymentMethod* *carRentalID* *stationRentalID* *FOREIGN KEY (carRentalID) REFERENCES ElectricCarRental(carRentalID)* *FOREIGN KEY (stationRentalID) REFERENCES ChargingStationRental(stationRentalID)*	***CarService*** *<u>serviceID - PK</u>* *carVIN* *serviceDate* *mechID* *FOREIGN KEY (carVIN) REFERENCES ElectricCar(carVIN)* *FOREIGN KEY (mechID) REFERENCES Employee(empID)*

b) Constraints (10 marks)

- Identify FIVE user constraints in the database and justify your choice.
- Please note that constraints related to NULL values and primary key/foreign key constraints are not considered user check constraints.

Answer:

No	Constraint type/description	Justification
1.	empRole CHECK constraint	To check if employee roles are valid as there are only 3 employee roles(manager, clerk, mechanic).
2.	paymentType CHECK constraint	To check if a payment type is valid as there are only two types of payments, deposit, and rental(rental includes any surcharge after the car is returned).
3.	paymentMethod CHECK constraint	To check if the payment method is valid as payments can only be made using cash, debit card, or credit card.
4.	IDType CHECK constraint	To check if ID types are valid customers must provide their identification (MyKad for Malaysians, passport for Non-Malaysians) before renting a vehicle or charging station.
5.	carRentalStartDate & carRentalEndDate CHECK constraint	To check if dates are valid as the preferred rental end date of the electric car should be after the preferred rental start date
6.	carRentalReturnDate & carRentalStartDate CHECK constraint	To check if the dates are valid the actual rental return date of the electric car should be after the preferred rental start date if the car has been returned
7.	empContactNum UNIQUE constraint	To ensure the employee contact number is unique
8.	TIN UNIQUE constraint	To ensure the employee Tax Identification Number is unique
9.	regNum UNIQUE constraint	To ensure the registration number/car plate of the electric car is unique
10.	custContactNum UNIQUE constraint	To ensure the customer contact number is unique

11.	IDNum UNIQUE constraint	To ensure the customer passport number(for Non-Malaysian) or identification card number(Malaysians) is unique

c) Indexes (10 marks)

- Identify FIVE indexes in the database and justify your choice. (one index for one table).

Answer:

No	Index/Description	Justification
1.	empRole (empRole) *Non-unique index*	Identifying or searching for an employee can be made easier by specifying their roles (manager, clerk, or mechanic).
2.	DomesticForeign (IDType) *Non-unique index*	Each customer has either a MyKad or a passport. This facilitates the searching of a customer by their type of identification.
3.	carStandards (carMake, carModel, classID) *Unique composite index*	Searching for a car's class (sedan, hatchback,...) can be made easier.
4.	paymentType (paymentType) *Non-unique index*	Searching for a customer's payment details can be made easier by specifying the payment type (deposit or rental).
5.	Maintenance (kmReadingLastService, carLastMaintenanceDate) *Non-unique index*	An employee will be able to find when an electric car was last serviced and determine when it is due for its next service more efficiently.

Important Notes:

- Ensure that the M:M relationships are properly mapped to RDM.

Part 3: Implement A Database (15 Marks)

- To develop and implement a customized database according to the output in Part 2.

a) Write a script named DBscript to create and populate a database using ORACLE DBMS.

- Ensure that both entity and referential integrities are accurately established. The script file should encompass the following requirements:
  - Implement FIVE user-defined check constraints in any five selected tables of your choice. Please note that constraints related to NULL values and primary key constraints are not considered user check constraints.
  - Define FIVE indexes for specific tables within the database. Include comments in the script file to explain why these tables require indexing.

b) Populate the database - create data of your choice by using the following as a guide:

- Each table must have a minimum of 15 rows except for each event table, which requires a minimum of 20 rows. However, where not possible, you may create a lesser number of records according to the scenario. (an event table consists of date or date and time; for example HIRE & ORDERS tables)
- Each record must consist of 'real' data that reflects integrity. For example,
  - If a field is defined as CHAR/VARCHAR2 data type, ensure that the data reflects the field name. For example, a field name title with data type VARCHAR2 may have data such as Database is fun! Instead of 101010101.
  - If a field is defined as a DATE data type, ensure that the date values reflect your scenario.
  - If there is a 1:M relationship in your ERD, ensure that the primary (parent) records can be linked to 1 or more secondary (child) records.
  - If your model has optionality, you need to reflect it in your data creation. In other words, some of the parent records may not have any child records.

Answer for a & b:

```
-------------------------------------------------------------------------------------------
--SEG1201 DATABASE FUNDAMENTALS Final Assessment
--   Group
-- Members:
-------------------------------------------------------------------------------------

ALTER SESSION SET NLS_DATE_FORMAT='DD-MON-YYYY';
DROP TABLE ElectricCar CASCADE CONSTRAINTS;
DROP TABLE CarType CASCADE CONSTRAINTS;
DROP TABLE CarClass CASCADE CONSTRAINTS;
DROP TABLE CarService CASCADE CONSTRAINTS;
DROP TABLE ChargingStation CASCADE CONSTRAINTS;
DROP TABLE ChargerType CASCADE CONSTRAINTS;
DROP TABLE ChargingSChargerT CASCADE CONSTRAINTS;
DROP TABLE Employee CASCADE CONSTRAINTS;
DROP TABLE Customer CASCADE CONSTRAINTS;
DROP TABLE ElectricCarRental CASCADE CONSTRAINTS;
DROP TABLE ChargingStationRental CASCADE CONSTRAINTS;
DROP TABLE RentalPayment CASCADE CONSTRAINTS;

------------------------------------------------------------------------------------
--create CarClass table for classes of electric cars available for rental
-- the following provides some explanation on selected attributes:
-- classDesc (Description of classes, eg. car with classID 1001 will have a description of "Sedan", classID 1002 =
"Compact")
-------------------------------------------------------------------------------------
CREATE TABLE CarClass
(classID INT PRIMARY KEY,
classDesc VARCHAR(50) NOT NULL,
carDailyHireRate DECIMAL(6,2) NOT NULL,
carDeposit DECIMAL(6,2) NOT NULL,
dailyLateReturnFee DECIMAL(6,2) NOT NULL
);
------------------------------------------------------------------------------------
--create CarType table for type of electric cars available for rental
-------------------------------------------------------------------------------------
CREATE TABLE CarType
(carTypeID INT PRIMARY KEY,
carMake VARCHAR(50) NOT NULL,
carModel VARCHAR(50) NOT NULL,
classID INT NOT NULL,
FOREIGN KEY (ClassID) REFERENCES CarClass(ClassID)
);

-----------------------------------------------------------------------------------
--create ChargingStation table for charging stations available for rental
-- the following provides some explanation on selected attributes:
-- numChargingPoints (number of charging points available in the station)
-- stationRentalHourlyRate (hourly rate of renting a charging station)
-----------------------------------------------------------------------------------
CREATE TABLE ChargingStation
(stationID INT PRIMARY KEY,
stationLocation VARCHAR(100),
numChargingPoints INT NOT NULL,
stationPurchaseDate DATE NOT NULL,
stationPurchaseCost DECIMAL(10, 2) NOT NULL,
stationLastMaintenanceDate DATE,
stationRentalHourlyRate DECIMAL (5,2) NOT NULL
);

-----------------------------------------------------------------------------------
--create ChargerType table for types of chargers available in charging stations
-- the following provides some explanation on selected attributes:
-- chargerDesc (charger description, eg charger with chargerTypeID 1 has chargerDesc of "Type 1", chargerTypeID 2 =
"Type 2", chargerTypeID 3 = "CHAdeMO")
-----------------------------------------------------------------------------------
CREATE TABLE ChargerType
(chargerTypeID INT PRIMARY KEY,
chargerDesc VARCHAR(50) NOT NULL
);

-----------------------------------------------------------------------------------
--create ChargingSChargerT table as a junction table that connects table ChargingStation and ChargerType
-----------------------------------------------------------------------------------
CREATE TABLE ChargingSChargerT (
stationID INT NOT NULL,
chargerTypeID INT NOT NULL,
PRIMARY KEY (stationID, ChargerTypeID),
FOREIGN KEY (stationID) REFERENCES ChargingStation(stationID),
FOREIGN KEY (chargerTypeID) REFERENCES ChargerType(chargerTypeID)
);

-----------------------------------------------------------------------------------
```

```sql
--create Employee table
-- the following provides some explanation on selected attributes:
-- empID (employee ID)
-- empFName (employee family name)
-- empGName (employee given name)
-- TIN (employee Tax Idendification Number)
-----------------------------------------------------------------------------------
CREATE TABLE Employee
(empID INT PRIMARY KEY,
empFName VARCHAR(50) NOT NULL,
empGName VARCHAR(50) NOT NULL,
empContactNum CHAR(10) UNIQUE NOT NULL,
empAddress VARCHAR(100),
TIN CHAR(12) UNIQUE NOT NULL,
salary DECIMAL(10, 2) NOT NULL,
empRole VARCHAR(20) NOT NULL,
managerID INT,
FOREIGN KEY (managerID) REFERENCES Employee(empID),
CHECK (empRole IN ('Manager', 'Clerk', 'Mechanic'))
);

-----------------------------------------------------------------------------------
--create ElectricCar table for electric cars available for rental
-- the following provides some explanation on selected attributes:
-- carVIN (Vehicle Identification Number of electric cars)
-- regNum (registration number/car plate)
-- battCap (battery capacity)
-- kmReadingLastService (the kilometre reading of the last service)
-- mechID (mechanic ID of the mechanic that is in charge of the car)
-----------------------------------------------------------------------------------
CREATE TABLE ElectricCar
(carVIN CHAR(17) PRIMARY KEY,
regNum CHAR(10) UNIQUE NOT NULL,
battCap DECIMAL(6,2) NOT NULL,
mileage DECIMAL(10,2) NOT NULL,
kmReadingLastService DECIMAL(10,2),
carPurchaseDate DATE NOT NULL,
carPurchaseCost DECIMAL(10, 2) NOT NULL,
carLastMaintenanceDate DATE,
carTypeID INT NOT NULL,
FOREIGN KEY (carTypeID) REFERENCES CarType(carTypeID)
);
```

```
-------------------------------------------------------------------------------------
--create CarService table
-- mechID (mechanic ID of the mechanic that is in charge of the car)
-------------------------------------------------------------------------------------
CREATE TABLE CarService (
   serviceID INT PRIMARY KEY,
   carVIN CHAR(17) NOT NULL,
   serviceDate DATE,
   mechID INT,
   FOREIGN KEY (carVIN) REFERENCES ElectricCar(carVIN),
   FOREIGN KEY (mechID) REFERENCES Employee(empID)
);

-------------------------------------------------------------------------------------
--create Customer table
-- the following provides some explanation on selected attributes:
-- custID (customer ID)
-- custFName (customer family name)
-- custGName (customer given name)
-- businessName (customer's business name if the customer is a business owner)
-- IDType (customer's Identification Type, MyKad for Malaysians, Passport for Non-Malaysians)
-- IDNum (customer's Identification Number, MyKad number for Malaysians, Passport number for Non-Malaysians)
-------------------------------------------------------------------------------------
CREATE TABLE Customer
(custID INT PRIMARY KEY,
custFName VARCHAR(50) NOT NULL,
custGName VARCHAR(50) NOT NULL,
custContactNum CHAR(10) UNIQUE NOT NULL,
custAddress VARCHAR(100),
businessName VARCHAR(50),
driverLicenseNum VARCHAR(20)UNIQUE NOT NULL,
IDType VARCHAR(10) NOT NULL,
IDNum VARCHAR(20) UNIQUE NOT NULL,
CHECK (IDType IN ('MyKad', 'Passport'))
);

-------------------------------------------------------------------------------------
--create ElectricCarRental table for managing the rental of electric cars
-- the following provides some explanation on selected attributes:
-- carRentalStartDate (preferred starting date of renting an electric car)
-- carRentalEndDate (preferred return date of renting an electric car)
-- carRentalReturnDate (actual return date of renting an electric car)
-----------------------------------------------------------------------------------
CREATE TABLE ElectricCarRental (
   carRentalID INT PRIMARY KEY,
   carRentalStartDate DATE NOT NULL,
   carRentalEndDate DATE NOT NULL,
   carRentalReturnDate DATE,
   carVIN CHAR(17) NOT NULL,
   custID INT NOT NULL,
   empID INT NOT NULL,
   damageFee DECIMAL(10, 2),
   CHECK (carRentalStartDate < carRentalEndDate),
   CHECK (carRentalReturnDate IS NULL OR carRentalStartDate < carRentalReturnDate),
   FOREIGN KEY (carVIN) REFERENCES ElectricCar(carVIN),
   FOREIGN KEY (custID) REFERENCES Customer(custID),
   FOREIGN KEY (empID) REFERENCES Employee(empID)
);

-----------------------------------------------------------------------------------
--create ChargingStationRental table for managing the rental of charging stations
-----------------------------------------------------------------------------------
CREATE TABLE ChargingStationRental (
   stationRentalID INT PRIMARY KEY,
   stationRentalHours NUMBER NOT NULL,
   stationID INT NOT NULL,
   custID INT NOT NULL,
   FOREIGN KEY (stationID) REFERENCES ChargingStation(stationID),
   FOREIGN KEY (custID) REFERENCES Customer(custID)
);

-----------------------------------------------------------------------------------
--create RentalPayment table
-- the following provides some explanation on selected attributes:
-- paymentType (type of payment, Deposit or Rental)
-- paymentMethod (Cash, Debit Card or Credit Card)
-----------------------------------------------------------------------------------
CREATE TABLE RentalPayment
(paymentID INT PRIMARY KEY,
paymentDateTime TIMESTAMP NOT NULL,
paymentType VARCHAR(20) NOT NULL,
paymentMethod VARCHAR(20) NOT NULL,
carRentalID INT NOT NULL,
stationRentalID INT,
FOREIGN KEY (carRentalID) REFERENCES ElectricCarRental(carRentalID),
FOREIGN KEY (stationRentalID) REFERENCES ChargingStationRental(stationRentalID),
CHECK (paymentType IN ('Deposit', 'Rental')),
CHECK (paymentMethod IN ('Cash', 'Debit Card', 'Credit Card'))
);

-----------------------------------------------------------------------------------
--create additional indexes
-- the following are some explanations as to why these indexes were created
-- empRole (filter employee roles; manager, clerk, or mechanic)
-- DomesticForeign (filter customers; local or internation)
-- paymentType (filter payment description; deposit or rental)
-----------------------------------------------------------------------------------
CREATE INDEX empRole ON Employee(empRole);
CREATE INDEX DomesticForeign ON Customer(IDType);
CREATE INDEX paymentType ON RentalPayment(paymentType);
CREATE UNIQUE INDEX carStandards ON CarType(carMake, carModel, classID);
CREATE INDEX Maintenance ON ElectricCar(kmReadingLastService, carLastMaintenanceDate);

-----------------------------------------------------------------------------------
--populating the database
--CarClass (classID, classDesc, carDailyHireRate, carDeposit, dailyLateReturnFee)
-----------------------------------------------------------------------------------
INSERT INTO CarClass VALUES (1001, 'Sedan', 300.00, 500.00, 450.00);
INSERT INTO CarClass VALUES (1002, 'Compact', 140.00, 350.00, 210.00);
INSERT INTO CarClass VALUES (1003, 'Sedan', 130.00, 400.00, 195.00);
INSERT INTO CarClass VALUES (1004, 'SUV', 320.00, 600.00, 480.00);
INSERT INTO CarClass VALUES (1005, 'Hatchback', 160.00, 350.00, 240.00);
INSERT INTO CarClass VALUES (1006, 'Compact', 300.00, 500.00, 450.00);
INSERT INTO CarClass VALUES (1007, 'Sedan', 650.00, 800.00, 975.00);
INSERT INTO CarClass VALUES (1008, 'Sport', 700.00, 1000.00, 1050.00);
INSERT INTO CarClass VALUES (1009, 'SUV', 350.00, 700.00, 525.00);
INSERT INTO CarClass VALUES (1010, 'SUV', 500.00, 700.00, 750.00);
INSERT INTO CarClass VALUES (1011, 'Hatchback', 120.00, 350.00, 180.00);
INSERT INTO CarClass VALUES (1012, 'Sedan', 320.00, 500.00, 480.00);
INSERT INTO CarClass VALUES (1013, 'Hatchback', 170.00, 350.00, 255.00);
INSERT INTO CarClass VALUES (1014, 'Sport', 900.00, 1000.00, 1350.00);
INSERT INTO CarClass VALUES (1015, 'Sedan', 180.00, 400.00, 270.00);

-----------------------------------------------------------------------------------
--CarType (carTypeID, carMake, carModel, classID)
```

```
-------------------------------------------------------------------------------------
INSERT INTO CarType VALUES (101, 'Tesla', 'Model S', 1001);
INSERT INTO CarType VALUES (102, 'Geely', 'Emgrand', 1003);
INSERT INTO CarType VALUES (103, 'Porsche', 'Taycan', 1008);
INSERT INTO CarType VALUES (104, 'Kia', 'EV6', 1004);
INSERT INTO CarType VALUES (105, 'BYD', 'Dolphin', 1011);
INSERT INTO CarType VALUES (106, 'BYD', 'Seal', 1015);
INSERT INTO CarType VALUES (107, 'Jaguar', 'I-PACE', 1010);
INSERT INTO CarType VALUES (108, 'Mini', 'Electric', 1006);
INSERT INTO CarType VALUES (109, 'Tesla', 'Model X', 1009);
INSERT INTO CarType VALUES (110, 'Renault', 'Zoe', 1005);
INSERT INTO CarType VALUES (111, 'Ora', 'Good Cat', 1002);
INSERT INTO CarType VALUES (112, 'Nissan', 'Leaf', 1013);
INSERT INTO CarType VALUES (113, 'Audi', 'RS e-tron GT', 1014);
INSERT INTO CarType VALUES (114, 'Hyundai', 'Ioniq 6', 1012);
INSERT INTO CarType VALUES (115, 'Mercedes', 'EQS', 1007);

-------------------------------------------------------------------------------------
--ChargingStation (stationID, stationLocation, numChargingPoints, stationPurchaseDate, stationPurchaseCost, stationLastMaintenanceDate, stationRentalHourlyRate)
-------------------------------------------------------------------------------------
INSERT INTO ChargingStation VALUES (201, 'Seksyen 51A', 2, '15-JAN-2022', 20000.00, '20-JUL-2024', 4.00);
INSERT INTO ChargingStation VALUES (202, 'Bandar Sunway', 4, '13-APR-2022', 40000.00, '30-JUN-2024', 5.20);
INSERT INTO ChargingStation VALUES (203, 'Parklane Hub', 4, '02-MAR-2022', 40000.00, '29-JUN-2024', 5.00);
INSERT INTO ChargingStation VALUES (204, 'UOA Park', 4, '21-JAN-2023', 40000.00, '03-JUL-2024', 5.00);
INSERT INTO ChargingStation VALUES (205, 'Sungai Kayu', 3, '08-OCT-2022', 30000.00, '20-JUN-2024', 4.00);
INSERT INTO ChargingStation VALUES (206, 'Taman Subang Ria', 2, '20-DEC-2022', 20000.00, '28-MAY-2024', 4.00);
INSERT INTO ChargingStation VALUES (207, 'Jalan Damansara', 4, '17-JUN-2023', 40000.00, '14-JUL-2024', 5.00);
INSERT INTO ChargingStation VALUES (208, 'Jalan SS7', 4, '14-JAN-2022', 40000.00, '02-JUL-2024', 4.00);
INSERT INTO ChargingStation VALUES (209, 'The Vertical', 2, '20-JUN-2022', 20000.00, '16-JUN-2024', 5.00);
INSERT INTO ChargingStation VALUES (210, 'The Sphere', 6, '09-JUL-2022', 60000.00, '19-JUL-2024', 5.00);
INSERT INTO ChargingStation VALUES (211, 'Jalan PJU', 2, '01-FEB-2022', 20000.00, '20-JUL-2024', 4.00);
INSERT INTO ChargingStation VALUES (212, 'Bukit Jalil', 2, '27-MAR-2023', 20000.00, '27-JUN-2024', 5.00);
INSERT INTO ChargingStation VALUES (213, 'Sungai Buloh', 2, '18-JAN-2022', 20000.00, '14-MAY-2024', 5.00);
INSERT INTO ChargingStation VALUES (214, 'Tropicana', 2, '20-JAN-2023', 20000.00, '05-JUL-2024', 5.00);
INSERT INTO ChargingStation VALUES (215, 'Jaya One', 2, '08-JAN-2023', 20000.00, '26-JUN-2024', 4.00);
INSERT INTO ChargingStation VALUES (216, 'Mid Valley', 4, '29-JAN-2023', 40000.00, '29-JUN-2024', 5.00);
INSERT INTO ChargingStation VALUES (217, 'Jalan Utama', 2, '02-JUN-2023', 20000.00, '20-JUL-2024', 4.00);
INSERT INTO ChargingStation VALUES (218, 'Jalan Puchong', 2, '02-JUL-2024', 20000.00, '14-JUL-2024', 4.00);
INSERT INTO ChargingStation VALUES (219, 'Sri Sentosa', 2, '08-FEB-2024', 20000.00, '26-JUN-2024', 4.00);
INSERT INTO ChargingStation VALUES (220, 'Sungei Wang', 2, '08-APR-2024', 20000.00, '26-JUN-2024', 5.00);
```

```
----------------------------------------------------------------------------------
--ChargerType (chargerTypeID, chargerDesc)
----------------------------------------------------------------------------------
INSERT INTO ChargerType VALUES (1, 'Type 1');
INSERT INTO ChargerType VALUES (2, 'Type 2');
INSERT INTO ChargerType VALUES (3, 'CHAdeMO');
INSERT INTO ChargerType VALUES (4, 'CCS');
INSERT INTO ChargerType VALUES (5, 'Tesla Supercharger');

----------------------------------------------------------------------------------
--ChargingSChargerT (stationID, chargerTypeID)
----------------------------------------------------------------------------------
INSERT INTO ChargingSChargerT VALUES (201, 4);
INSERT INTO ChargingSChargerT VALUES (202, 5);
INSERT INTO ChargingSChargerT VALUES (203, 2);
INSERT INTO ChargingSChargerT VALUES (203, 4);
INSERT INTO ChargingSChargerT VALUES (204, 4);
INSERT INTO ChargingSChargerT VALUES (205, 2);
INSERT INTO ChargingSChargerT VALUES (205, 3);
INSERT INTO ChargingSChargerT VALUES (205, 4);
INSERT INTO ChargingSChargerT VALUES (206, 4);
INSERT INTO ChargingSChargerT VALUES (207, 1);
INSERT INTO ChargingSChargerT VALUES (207, 4);
INSERT INTO ChargingSChargerT VALUES (208, 1);
INSERT INTO ChargingSChargerT VALUES (208, 2);
INSERT INTO ChargingSChargerT VALUES (209, 4);
INSERT INTO ChargingSChargerT VALUES (210, 4);
INSERT INTO ChargingSChargerT VALUES (211, 4);
INSERT INTO ChargingSChargerT VALUES (212, 3);
INSERT INTO ChargingSChargerT VALUES (212, 4);
INSERT INTO ChargingSChargerT VALUES (213, 2);
INSERT INTO ChargingSChargerT VALUES (214, 2);
INSERT INTO ChargingSChargerT VALUES (215, 2);
INSERT INTO ChargingSChargerT VALUES (215, 4);
INSERT INTO ChargingSChargerT VALUES (216, 4);
INSERT INTO ChargingSChargerT VALUES (217, 4);
INSERT INTO ChargingSChargerT VALUES (218, 2);
INSERT INTO ChargingSChargerT VALUES (219, 5);
INSERT INTO ChargingSChargerT VALUES (220, 2);

----------------------------------------------------------------------------------
--Employee (empID, empFName, empGName, empContactNum, empAddress, TIN, salary, empRole, managerID)
```

```sql
-------------------------------------------------------------------------------------
INSERT INTO Employee VALUES (1, 'Abdullah', 'Hafiz', '0164266950', '12 Jalan Seri Bintang, Taman Sri Hartamas, 50480 Kuala Lumpur', 'MYB543210987', 75000.00, 'Manager', NULL);
INSERT INTO Employee VALUES (10, 'Wong', 'Ling Ying', '0132412363', '27 Jalan Bahagia, Taman Sentosa, 80150 Johor Bahru', 'MYA104215252', 73500.00, 'Manager', NULL);
INSERT INTO Employee VALUES (2, 'Yusof', 'Syafiqah', '0116244032', '45 Jalan Mahsuri, Bandar Sunway, 46150 Petaling Jaya', 'MYB876543210', 52500.00, 'Clerk', 1);
INSERT INTO Employee VALUES (3, 'Tan', 'Li Yan', '0126345573', '8 Lorong Seri Indah, Bayan Lepas, 11900 Penang', 'MYC765432109', 64000.00, 'Mechanic', 1);
INSERT INTO Employee VALUES (4, 'Neutron', 'Jimmy', '0163522385', '23 Jalan Harmoni, Taman Desa Jaya, 81100 Johor Bahru', 'MYA654321098', 51000.00, 'Clerk', 10);
INSERT INTO Employee VALUES (5, 'Fletcher', 'Ferb', '0111631093', '17 Jalan Permai, Bukit Bintang, 55100 Kuala Lumpur', 'MYA321098765', 50000.00, 'Clerk', 1);
INSERT INTO Employee VALUES (6, 'Lee', 'Ming Hao', '0172962654', '6 Jalan Cemerlang, Taman Sri Nibong, 11900 Johor Bahru', 'MYC109876775', 62500.00, 'Mechanic', 10);
INSERT INTO Employee VALUES (7, 'Soon', 'Ting Wong', '0176733408', '32 Persiaran Damai, Kota Damansara, 47810 Petaling Jaya', 'MYA098764434', 67500.00, 'Mechanic', 10);
INSERT INTO Employee VALUES (8, 'Subramaniam', 'Rajesh', '0122574066', '14 Jalan Bestari, Taman Tun Dr Ismail, 60000 Kuala Lumpur', 'MYA232645239', 61000.00, 'Mechanic', 1);
INSERT INTO Employee VALUES (9, 'Lee', 'Chin Mei', '0166079303', '9 Lorong Seri Manis, Bayan Lepas, 11920 Penang', 'MYB555765253', 54500.00, 'Clerk', 1);
INSERT INTO Employee VALUES (11, 'Muthu', 'Tharma', '0126538394', '3 Jalan Perdana, Bukit Damansara, 50490 Kuala Lumpur', 'MYB098765432', 53000.00, 'Clerk', 10);
INSERT INTO Employee VALUES (12, 'Bong', 'Alysha', '0125156535', '11 Jalan Harmoni Indah, Bandar Utama, 47800 Petaling Jaya', 'MYC098765432', 66000.00, 'Mechanic', 10);
INSERT INTO Employee VALUES (13, 'Lim', 'Ah Kow', '0123976157', '14 Jalan Sutera, Taman Sri Sinar, 51200 Kuala Lumpur', 'MYC214449743', 61500.00, 'Mechanic', 10);
INSERT INTO Employee VALUES (14, 'Ahmad', 'Amir', '0128034752', '9 Lorong Baiduri, Bandar Sunway, 47500 Subang Jaya', 'MYB932775953', 65500.00, 'Mechanic', 1);
INSERT INTO Employee VALUES (15, 'Chong', 'Xuan Loo', '0168843085', '36 Jalan Ria, Bukit Jelutong, 40150 Shah Alam', 'MYA837545222', 64000.00, 'Mechanic', 1);

-------------------------------------------------------------------------------------
--ElectricCar (carVIN, regNum, battCap, mileage, kmReadingLastService, carPurchaseDate, carPurchaseCost, carLastMaintenanceDate, carTypeID)
-------------------------------------------------------------------------------------
INSERT INTO ElectricCar VALUES ('1HGCM82633A123456', 'VKR1939', 80.00, 20000.00, 15000.00, '10-MAY-2022', 300000.00, '15-JUN-2024', 101);
INSERT INTO ElectricCar VALUES ('1HGCM82633A654321', 'WUP3942', 40.00, 15000.00, 10000.00, '22-MAR-2022', 130000.00, '20-JUL-2024', 102);
INSERT INTO ElectricCar VALUES ('1HGCM82633A789012', 'WKW9253', 130.00, 30000.00, 25000.00, '15-JAN-2023', 700000.00, '10-JUL-2024', 103);
INSERT INTO ElectricCar VALUES ('1HGCM82633A890123', 'VPZ9285', 70.00, 25000.00, 20000.00, '25-JUL-2022', 320000.00, '05-JUN-2024', 104);
INSERT INTO ElectricCar VALUES ('1HGCM82633A901234', 'WTF2316', 46.00, 10000.00, 5000.00, '30-JUN-2023', 120000.00, '15-JUL-2024', 105);
INSERT INTO ElectricCar VALUES ('1HGCM82633A234567', 'WSV3942', 82.50, 18000.00, 13000.00, '10-MAR-2022', 180000.00, '10-MAY-2024', 106);
INSERT INTO ElectricCar VALUES ('1HGCM82633A345678', 'WPZ2845', 90.00, 22000.00, 17000.00, '20-AUG-2022', 500000.00, '15-APR-2024', 107);
INSERT INTO ElectricCar VALUES ('1HGCM82633A456789', 'VLW8295', 54.20, 27000.00, 22000.00, '08-AUG-2023', 300000.00, '30-JUN-2024', 108);
INSERT INTO ElectricCar VALUES ('1HGCM82633A567890', 'WOR5975', 100.00, 30000.00, 25000.00, '25-APR-2022', 350000.00, '25-MAY-2024', 109);
INSERT INTO ElectricCar VALUES ('1HGCM82633A678901', 'VAW2967', 52.00, 12000.00, 7000.00, '10-JUN-2023', 160000.00, '05-JUL-2024', 110);
INSERT INTO ElectricCar VALUES ('1HGCM82633A789123', 'WKW5586', 47.80, 17000.00, 12000.00, '15-MAY-2023', 140000.00, '10-APR-2024', 111);
INSERT INTO ElectricCar VALUES ('1HGCM82633A890234', 'WIT7644', 40.00, 19000.00, 14000.00, '18-JUL-2022', 170000.00, '18-MAY-2024', 112);
INSERT INTO ElectricCar VALUES ('1HGCM82633A901345', 'WWA9286', 170.00, 28000.00, 23000.00, '20-JAN-2023', 900000.00, '12-JUL-2024', 113);
INSERT INTO ElectricCar VALUES ('1HGCM82633A012456', 'VVO3456', 77.40, 26000.00, 21000.00, '15-SEP-2022', 320000.00, '08-JUN-2024', 114);
INSERT INTO ElectricCar VALUES ('1HGCM82633A123567', 'WOI8309', 120.00, 14000.00, 9000.00, '28-MAY-2023', 650000.00, '10-JUL-2024', 115);
INSERT INTO ElectricCar VALUES ('1HGCM82633A125930', 'WOA9372', 170.00, 28000.00, NULL, '02-JUL-2024', 900000.00, NULL, 113);
INSERT INTO ElectricCar VALUES ('1HGCM82633A109578', 'WAP3209', 82.50, 18000.00, NULL, '06-JUL-2024', 180000.00, NULL, 106);
INSERT INTO ElectricCar VALUES ('1HGCM82633A018573', 'VAT1023', 100.00, 30000.00, NULL, '14-JUL-2024', 350000.00, NULL, 109);
INSERT INTO ElectricCar VALUES ('1HGCM82633A098573', 'WPO3845', 54.20, 27000.00, NULL, '17-JUL-2024', 300000.00, NULL, 108);
INSERT INTO ElectricCar VALUES ('1HGCM82633A809273', 'WNZ4059', 52.00, 12000.00, NULL, '20-JUL-2024', 160000.00, NULL, 110);

-------------------------------------------------------------------------------------
--CarService (serviceID, carVIN, serviceDate, mechID)
-------------------------------------------------------------------------------------
INSERT INTO CarService VALUES (1, '1HGCM82633A123456', '18-DEC-2022', 3);
INSERT INTO CarService VALUES (2, '1HGCM82633A123456', '25-AUG-2023', 3);
INSERT INTO CarService VALUES (3, '1HGCM82633A123456', '15-JUN-2024', 3);
INSERT INTO CarService VALUES (4, '1HGCM82633A654321', '17-JAN-2023', 8);
INSERT INTO CarService VALUES (5, '1HGCM82633A654321', '20-JUL-2024', 8);
INSERT INTO CarService VALUES (6, '1HGCM82633A789012', '10-JUL-2024', 6);
INSERT INTO CarService VALUES (7, '1HGCM82633A890123', '29-MAR-2023', 7);
INSERT INTO CarService VALUES (8, '1HGCM82633A890123', '05-JUN-2024', 7);
INSERT INTO CarService VALUES (9, '1HGCM82633A901234', '15-JUL-2024', 8);
INSERT INTO CarService VALUES (10, '1HGCM82633A234567', '15-JAN-2023', 13);
INSERT INTO CarService VALUES (11, '1HGCM82633A234567', '10-MAY-2024', 13);
INSERT INTO CarService VALUES (12, '1HGCM82633A345678', '20-JUN-2023', 7);
INSERT INTO CarService VALUES (13, '1HGCM82633A345678', '15-APR-2024', 7);
INSERT INTO CarService VALUES (14, '1HGCM82633A456789', '19-OCT-2023', 12);
INSERT INTO CarService VALUES (15, '1HGCM82633A456789', '26-JAN-2024', 12);
INSERT INTO CarService VALUES (16, '1HGCM82633A456789', '03-APR-2024', 12);
INSERT INTO CarService VALUES (17, '1HGCM82633A456789', '30-JUN-2024', 12);
INSERT INTO CarService VALUES (18, '1HGCM82633A567890', '06-OCT-2023', 12);
INSERT INTO CarService VALUES (19, '1HGCM82633A567890', '25-MAY-2024', 12);
INSERT INTO CarService VALUES (20, '1HGCM82633A678901', '10-JAN-2024', 15);
INSERT INTO CarService VALUES (21, '1HGCM82633A678901', '05-JUL-2024', 15);
INSERT INTO CarService VALUES (22, '1HGCM82633A789123', '10-APR-2024', 14);
INSERT INTO CarService VALUES (23, '1HGCM82633A890234', '23-AUG-2023', 14);
INSERT INTO CarService VALUES (24, '1HGCM82633A890234', '18-MAY-2024', 14);
INSERT INTO CarService VALUES (25, '1HGCM82633A901345', '20-AUG-2023', 13);
INSERT INTO CarService VALUES (26, '1HGCM82633A901345', '12-JUL-2024', 13);
INSERT INTO CarService VALUES (27, '1HGCM82633A012456', '22-MAR-2023', 3);
INSERT INTO CarService VALUES (28, '1HGCM82633A012456', '08-JUN-2024', 3);
INSERT INTO CarService VALUES (29, '1HGCM82633A123567', '10-JUL-2024', 3);

------------------------------------------------------------------------------------
--Customer (custID, custFName, custGName, custContactNum, custAddress, businessName, driverLicenseNum,
IDType, IDNum)
------------------------------------------------------------------------------------
INSERT INTO Customer VALUES ('00001', 'Schmidt', 'Eva', '0125956789', '23, Jalan Merah, Taman Ria, 81200 Johor
Bahru, Johor', NULL, '25471893', 'Passport', 'C23456789') ;
INSERT INTO Customer VALUES ('00002', 'Nakamura', 'Yuki', '0162345678', '12, Lorong Impian, Taman Desa, 46000
Petaling Jaya, Selangor', 'VoltFlare Engines', '83746219', 'Passport', 'TN1234567') ;
INSERT INTO Customer VALUES ('00003', 'Tan', 'Cheng Wei', '0174567890', 'Unit 5B, Block A, Pangsapuri Murni, 11600
George Town, Penang', 'Azure Software Solutions', '49273618', 'MyKad', '010407124537') ;
INSERT INTO Customer VALUES ('00004', 'Kumar', 'Ravi', '0136789012', '18, Jalan Teratai, Taman Bukit, 75400 Melaka,
Melaka', NULL, '18564792', 'MyKad', '010813124520') ;
INSERT INTO Customer VALUES ('00005', 'Okoye', 'Nia', '0129876543', '9A, Jalan Merbuk, Taman Sejati, 14000 Bukit
Mertajam, Penang', NULL, '76039284', 'Passport', 'A12345678') ;
INSERT INTO Customer VALUES ('00006', 'Ng', 'Mei Ling', '0168765432', '6, Jalan Kemboja, Taman Sentosa, 81300
Skudai, Johor', NULL, '31485927', 'MyKad', '010669726455') ;
```

```
INSERT INTO Customer VALUES ('00007', 'Hakim', 'Amirul', '0171234567', 'Unit 10-2, Block C, Kondominium Avaria,
55000 Kuala Lumpur, Wilayah Persekutuan', NULL, '92837461', 'MyKad', '980620019874') ;
INSERT INTO Customer VALUES ('00008', 'Ivanova', 'Anastasia', '0135678901', '3, Lorong Dahlia, Taman Melur, 93000
Kuching, Sarawak', NULL, '47518263', 'Passport', '6489987654') ;
INSERT INTO Customer VALUES ('00009', 'Martinez', 'Diego', '0120987654', '22, Jalan Anggerik, Taman Jaya, 80000
Johor Bahru, Johor', NULL, '68134957', 'Passport', 'X1234567Z') ;
INSERT INTO Customer VALUES ('00010', 'Phuah', 'Christine', '0163456789', 'Unit 2A, Block B, Apartmen Harmoni,
86000 Kluang, Johor', 'Harmony Wellness Spa', '29384756', 'MyKad', '670212017384') ;
INSERT INTO Customer VALUES ('00011', 'Chew', 'Evelyn', '0172345678', '20A, Jalan Bunga Raya, Taman Mawar,
13700 Seberang Jaya, Penang', NULL, '54628193', 'MyKad', '950903014568') ;
INSERT INTO Customer VALUES ('00012', 'Nguyen', 'Liam', '0134567890', '7, Jalan Rusa, Taman Murni, 40100 Shah
Alam, Selangor', 'Golden Harvest Bakery', '81927564', 'Passport', 'E23456789') ;
INSERT INTO Customer VALUES ('00013', 'Suraya', 'Intan', '0126789012', '12, Jalan Kasturi, Taman Megah, 70200
Seremban, Negeri Sembilan', NULL, '36749182', 'MyKad', '820731015639') ;
INSERT INTO Customer VALUES ('00014', 'Ahmed', 'Lila', '0165432109', '19, Lorong Serindit, Taman Permai, 50350
Kuala Lumpur, Wilayah Persekutuanr', NULL, '75492831', 'Passport', 'A98765432') ;
INSERT INTO Customer VALUES ('00015', 'Yong', 'Jacob', '0170987654', 'Unit 7D, Block D, Pangsapuri Lestari, 68000
Ampang, Selangor', NULL, '62873915', 'MyKad', '990124018273') ;

--------------------------------------------------------------------------------------
--ElectricCarRental (carRentalID, carRentalStartDate, carRentalEndDate, carRentalReturnDate, carVIN, custID, empID,
damageFee)
--------------------------------------------------------------------------------------
INSERT INTO ElectricCarRental VALUES (1, '05-JAN-2024', '11-JAN-2024', '11-JAN-2024', '1HGCM82633A654321',
00002, 5, NULL);
INSERT INTO ElectricCarRental VALUES (2, '06-JAN-2024', '10-JAN-2024', '10-JAN-2024', '1HGCM82633A012456',
00012, 4, NULL);
INSERT INTO ElectricCarRental VALUES (3, '11-JAN-2024', '16-JAN-2024', '16-JAN-2024', '1HGCM82633A456789',
00002, 9, NULL);
INSERT INTO ElectricCarRental VALUES (4, '25-JAN-2024', '28-JAN-2024', '28-JAN-2024', '1HGCM82633A901234',
00005, 11, NULL);
INSERT INTO ElectricCarRental VALUES (5, '13-FEB-2024', '17-FEB-2024', '17-FEB-2024', '1HGCM82633A789012',
00011, 4, NULL);
INSERT INTO ElectricCarRental VALUES (6, '15-FEB-2024', '20-FEB-2024', '20-FEB-2024', '1HGCM82633A567890',
00009, 9, 1000.00);
INSERT INTO ElectricCarRental VALUES (7, '20-FEB-2024', '22-FEB-2024', '22-FEB-2024', '1HGCM82633A890123',
00004, 4, NULL);
INSERT INTO ElectricCarRental VALUES (8, '01-MAR-2024', '02-MAR-2024', '02-MAR-2024', '1HGCM82633A789123',
00015, 5, NULL);
INSERT INTO ElectricCarRental VALUES (9, '11-MAR-2024', '13-MAR-2024', '13-MAR-2024',  '1HGCM82633A901345',
00013, 4, NULL);
```

```
INSERT INTO ElectricCarRental VALUES (10, '15-MAR-2024', '17-MAR-2024', '17-MAR-2024', '1HGCM82633A345678', 00009, 5, NULL);
INSERT INTO ElectricCarRental VALUES (11, '25-MAR-2024', '29-MAR-2024', '30-MAR-2024', '1HGCM82633A123456', 00004, 11, NULL);
INSERT INTO ElectricCarRental VALUES (12, '01-APR-2024', '04-APR-2024', '04-APR-2024', '1HGCM82633A123567', 00010, 11, NULL);
INSERT INTO ElectricCarRental VALUES (13, '17-APR-2024', '20-APR-2024', '20-APR-2024', '1HGCM82633A234567', 00007, 9, NULL);
INSERT INTO ElectricCarRental VALUES (14, '19-MAY-2024', '22-MAY-2024', '22-MAY-2024', '1HGCM82633A123567', 00003, 9, NULL);
INSERT INTO ElectricCarRental VALUES (15, '19-JUN-2024', '20-JUN-2024', '20-JUN-2024', '1HGCM82633A901345', 00012, 11, NULL);
INSERT INTO ElectricCarRental VALUES (16, '02-JUL-2024', '05-JUL-2024', '05-JUL-2024', '1HGCM82633A123456', 00001, 2, NULL);
INSERT INTO ElectricCarRental VALUES (17, '03-JUL-2024', '05-JUL-2024', '05-JUL-2024', '1HGCM82633A345678', 00007, 2, NULL);
INSERT INTO ElectricCarRental VALUES (18, '07-JUL-2024', '10-JUL-2024', '12-JUL-2024', '1HGCM82633A890234', 00008, 11, 500.00);
INSERT INTO ElectricCarRental VALUES (19, '16-JUL-2024', '20-JUL-2024', '20-JUL-2024', '1HGCM82633A678901', 00014, 2, NULL);
INSERT INTO ElectricCarRental VALUES (20, '17-JUL-2024', '20-JUL-2024', NULL, '1HGCM82633A456789', 00006, 5, NULL);
------------------------------------------------------------------------------------
--ChargingStationRental (stationRentalID, stationRentalHours, stationID, custID)
------------------------------------------------------------------------------------
INSERT INTO ChargingStationRental VALUES (1, 6, 201, 00001);
INSERT INTO ChargingStationRental VALUES (2, 4, 203, 00005);
INSERT INTO ChargingStationRental VALUES (3, 2, 201, 00012);
INSERT INTO ChargingStationRental VALUES (4, 9, 204, 00009);
INSERT INTO ChargingStationRental VALUES (5, 6, 204, 00013);
INSERT INTO ChargingStationRental VALUES (6, 6, 207, 00008);
INSERT INTO ChargingStationRental VALUES (7, 6, 206, 00002);
INSERT INTO ChargingStationRental VALUES (8, 6, 202, 00009);
INSERT INTO ChargingStationRental VALUES (9, 5, 211, 00003);
INSERT INTO ChargingStationRental VALUES (10, 5, 215, 00010);
INSERT INTO ChargingStationRental VALUES (11, 6, 212, 00004);
INSERT INTO ChargingStationRental VALUES (12, 8, 210, 00011);
INSERT INTO ChargingStationRental VALUES (13, 6, 209, 00006);
INSERT INTO ChargingStationRental VALUES (14, 6, 209, 00015);
INSERT INTO ChargingStationRental VALUES (15, 6, 215, 00014);
INSERT INTO ChargingStationRental VALUES (16, 6, 205, 00007);
INSERT INTO ChargingStationRental VALUES (17, 5, 205, 00007);
INSERT INTO ChargingStationRental VALUES (18, 3, 208, 00004);
INSERT INTO ChargingStationRental VALUES (19, 6, 203, 00012);
INSERT INTO ChargingStationRental VALUES (20, 7, 205, 00005);
INSERT INTO ChargingStationRental VALUES (21, 8, 214, 00002);
INSERT INTO ChargingStationRental VALUES (22, 3, 206, 00005);



-----------------------------------------------------------------------------------
--RentalPayment (paymentID, paymentDateTime, paymentType, paymentMethod, carRentalID, stationRentalID)
-----------------------------------------------------------------------------------
INSERT INTO RentalPayment VALUES (1, '05-JAN-2024 11:55:01', 'Deposit', 'Cash', 1, NULL);
INSERT INTO RentalPayment VALUES (2, '06-JAN-2024 10:43:44', 'Deposit', 'Credit Card', 2, NULL);
INSERT INTO RentalPayment VALUES (3, '10-JAN-2024 11:12:11', 'Rental', 'Debit Card', 2, 3);
INSERT INTO RentalPayment VALUES (4, '11-JAN-2024 10:13:00', 'Rental', 'Credit Card', 1, 7);
INSERT INTO RentalPayment VALUES (5, '11-JAN-2024 12:02:52', 'Deposit', 'Debit Card', 3, NULL);
INSERT INTO RentalPayment VALUES (6, '16-JAN-2024 12:13:23', 'Rental', 'Cash', 3, 21);
INSERT INTO RentalPayment VALUES (7, '25-JAN-2024 11:43:20', 'Deposit', 'Cash', 4, NULL);
INSERT INTO RentalPayment VALUES (8, '28-JAN-2024 10:10:54', 'Rental', 'Credit Card', 4, 2);
INSERT INTO RentalPayment VALUES (9, '13-FEB-2024 12:34:51', 'Deposit', 'Debit Card', 5, NULL);
INSERT INTO RentalPayment VALUES (10, '15-FEB-2024 12:43:13', 'Deposit', 'Debit Card', 6, NULL);
INSERT INTO RentalPayment VALUES (11, '17-FEB-2024 12:11:16', 'Rental', 'Cash', 5, 12);
INSERT INTO RentalPayment VALUES (12, '20-FEB-2024 11:32:52', 'Rental', 'Credit Card', 6, 4);
INSERT INTO RentalPayment VALUES (13, '20-FEB-2024 12:46:00', 'Deposit', 'Credit Card', 7, NULL);
INSERT INTO RentalPayment VALUES (14, '22-FEB-2024 11:12:41', 'Rental', 'Debit Card', 7, 11);
INSERT INTO RentalPayment VALUES (15, '01-MAR-2024 11:12:45', 'Deposit', 'Credit Card', 8, NULL);
INSERT INTO RentalPayment VALUES (16, '02-MAR-2024 10:22:02', 'Rental', 'Credit Card', 8, 14);
INSERT INTO RentalPayment VALUES (17, '11-MAR-2024 12:00:34', 'Deposit', 'Cash', 9, NULL);
INSERT INTO RentalPayment VALUES (18, '13-MAR-2024 12:46:23', 'Rental', 'Debit Card', 9, 5);
INSERT INTO RentalPayment VALUES (19, '15-MAR-2024 11:50:20', 'Deposit', 'Debit Card', 10, NULL);
INSERT INTO RentalPayment VALUES (20, '17-MAR-2024 10:13:21', 'Rental', 'Cash', 10, 8);
INSERT INTO RentalPayment VALUES (21, '25-MAR-2024 10:50:21', 'Deposit', 'Cash', 11, NULL);
INSERT INTO RentalPayment VALUES (22, '30-MAR-2024 11:20:40', 'Rental', 'Debit Card', 11, 18);
INSERT INTO RentalPayment VALUES (23, '01-APR-2024 11:57:58', 'Deposit', 'Cash', 12, NULL);
INSERT INTO RentalPayment VALUES (24, '04-APR-2024 10:00:53', 'Rental', 'Debit Card', 12, 10);
INSERT INTO RentalPayment VALUES (25, '17-APR-2024 11:25:00', 'Deposit', 'Debit Card', 13, NULL);
INSERT INTO RentalPayment VALUES (26, '20-APR-2024 12:18:02', 'Rental', 'Cash', 13, 22);
INSERT INTO RentalPayment VALUES (27, '19-MAY-2024 12:26:20', 'Deposit', 'Credit Card', 14, NULL);
INSERT INTO RentalPayment VALUES (28, '22-MAY-2024 10:30:50', 'Rental', 'Debit Card', 14, 9);
INSERT INTO RentalPayment VALUES (29, '19-JUN-2024 11:00:25', 'Deposit', 'Debit Card', 15, NULL);
INSERT INTO RentalPayment VALUES (30, '20-JUN-2024 11:22:54', 'Rental', 'Debit Card', 15, 19);
INSERT INTO RentalPayment VALUES (31, '02-JUL-2024 10:21:25', 'Deposit', 'Credit Card', 16, NULL);
```

```sql
INSERT INTO RentalPayment VALUES (32, '03-JUL-2024 10:43:23', 'Deposit', 'Credit Card', 17, NULL);
INSERT INTO RentalPayment VALUES (33, '05-JUL-2024 10:53:05', 'Rental', 'Debit Card', 16, 1);
INSERT INTO RentalPayment VALUES (34, '05-JUL-2024 11:45:40', 'Rental', 'Debit Card', 17, 16);
INSERT INTO RentalPayment VALUES (35, '07-JUL-2024 10:31:01', 'Deposit', 'Debit Card', 18, NULL);
INSERT INTO RentalPayment VALUES (36, '12-JUL-2024 11:17:13', 'Rental', 'Debit Card', 18, 6);
INSERT INTO RentalPayment VALUES (37, '16-JUL-2024 12:00:23', 'Deposit', 'Cash', 19, NULL);
INSERT INTO RentalPayment VALUES (38, '17-JUL-2024 10:51:29', 'Deposit', 'Credit Card', 20, NULL);
INSERT INTO RentalPayment VALUES (39, '20-JUL-2024 11:25:11', 'Rental', 'Debit Card', 19, 15);
```

Part 4: Query A Database (Each Sub-Part Carries 4 Marks; Max Of 20 Marks)

- Write FOUR questions in a scenario format, and for each scenario, your SQL codes need to meet the criteria as stated in parts a - d.

Please Refer To Appendix 1 For A Full Example.

a.

- Write a user query containing two sub-queries of different levels. Additionally, incorporate one new date function that hasn't been covered in this course. Refer to the lab notes on sub-queries and explore the function on the ORACLE website for details (https://docs.oracle.com/database/121/SQLRF/functions.htm#SQLRF006)

Answer:

No	Query Description
a	EcoDrive wants to find out the total days that have elapsed since the last payment was made at the charging station. Include the paymentID, date and time the payment was made, and charging station location for that payment. Write a user query containing two sub-queries of different levels. SQL > SELECT paymentID, TO_CHAR(paymentDateTime, 'DD-MON-YYYY HH12:MI:SS') AS PaymentDateTime, stationLocation, (SELECT EXTRACT(DAY FROM (SYSTIMESTAMP - MAX(RentalPayment.paymentDateTime))) FROM RentalPayment) AS Days_Since_Last_Payment FROM RentalPayment, ChargingStationRental, ChargingStation WHERE (RentalPayment.stationRentalID = ChargingStationRental.stationRentalID) AND (ChargingStationRental.stationID = ChargingStation.stationID) AND (RentalPayment.stationRentalID IS NOT NULL) AND (paymentDateTime = (SELECT MAX(paymentDateTime) FROM RentalPayment))

PAYMENTID	PAYMENTDATETIME	STATIONLOCATION	DAYS_SINCE_LAST_PAYMENT
39	20-JUL-2024 11:25:11	Jaya One	7

b.

- Write a user query with a 5-table join, TWO user conditions (note: the join condition and user conditions are not the same), and a GROUP BY clause and HAVING subclause. Ensure that your scenario reflects the reason for such join, and not join those tables for the sake of joining.

Answer:

No	Query Description

b	The business owner wants to determine which car classes have the highest average daily hire rate and the number of car types within each class as they plan to remove the car classes where the average daily hire rate is less than 200 and keep the rest. This will assist the owner in understanding which car classes are more expensive and how many types fall into each class.
	
	SQL>
	SELECT cc.classDesc AS "Class Description", ROUND(AVG(cc.carDailyHireRate), 2) AS "Average Daily Hire Rate", COUNT(ct.carTypeID) AS "Number of Car Types"
	FROM CarClass cc, CarType ct, ElectricCar ec, ElectricCarRental ecr, Customer c
	WHERE cc.classID = ct.classID
	AND ct.carTypeID = ec.carTypeID
	AND ec.carVIN = ecr.carVIN
	AND ecr.custID = c.custID
	GROUP BY cc.classDesc
	HAVING AVG(cc.carDailyHireRate) > 200;

Class Description	Average Daily Hire Rate	Number of Car Types
Sport	833.33	3
Sedan	361.43	7
SUV	417.5	4
Compact	246.67	3

c.

- Write a user query with a UNARY join and TWO user conditions. One of the conditions uses the LIKE keyword.

Answer:

No	Query Description

c	The business owner has instructed the managers to report back with a list containing the names and contact numbers of the employees they supervise, who live either in Johor Bahru or Petaling Jaya, and have a salary below $65000. This is because the owner would like to provide them with transport allowances as work incentives.
	
	SQL>
	SELECT e.empGName AS "Given Name", e.empFName AS "Family Name", e.empContactNum AS "Contact Number"
	FROM Employee e
	WHERE (e.empRole = 'Clerk' OR e.empRole = 'Mechanic')
	AND (e.empAddress LIKE '%Johor Bahru%' OR e.empAddress LIKE '%Petaling Jaya%')
	AND e.salary < 65000;

Given Name	Family Name	Contact Number
Syafiqah	Yusof	0116244032
Jimmy	Neutron	0163522385
Ming Hao	Lee	0172962654

d.

- Write a user query using the set operator MINUS. Ensure that your scenario reflects the reason for such a requirement. Display only the relevant records and fields.

Answer:

No	Query Description

d	The business owner wants to identify the electric cars which have never been rented out by any customers. This will help in making informed decisions about potential future promotions.
	
	SQL>
	SELECT *
	FROM ElectricCar
	MINUS
	SELECT *
	FROM ElectricCar
	WHERE carVIN IN (SELECT carVIN FROM ElectricCarRental);

CARVIN	REGNUM	BATTCAP	MILEAGE	KMREADINGLASTSERVICE	CARPURCHASEDATE	CARPURCHASECOST	CARLASTMAINTENANCEDATE	CARTYPEID
1HGCM82633A018573	VAT1023	100	30000	-	14-JUL-2024	350000	-	109
1HGCM82633A098573	WPO3845	54.2	27000	-	17-JUL-2024	300000	-	108
1HGCM82633A109578	WAP3209	82.5	18000	-	06-JUL-2024	180000	-	106
1HGCM82633A125930	WOA9372	170	28000	-	02-JUL-2024	900000	-	113
1HGCM82633A809273	WNZ4059	52	12000	-	20-JUL-2024	160000	-	110

e.

- Answer the following question based on your chosen scenario
    - Case Scenario Electric cars
        - Provide a list of cars less than 1 year old and have the highest number of services within a year.
    - Case Scenario Travel agency
        - Calculate the total amount paid by customer Sally Smith for her most current booking.
    - Case Scenario Airbnb
        - Provide a list of Airbnb customers who have made at least 3 bookings within a year with a total sum paid of more than $3000

Answer:

No	Query Description

e	Case Scenario Electric cars
	Provide a list of cars less than 1 year old and have the highest number of services within a year.
	
	SQL>
	SELECT ec.carVIN, ec.regNum, ec.battCap, ec.mileage, ec.kmReadingLastService, ec.carPurchaseDate, ec.carPurchaseCost, ec.carLastMaintenanceDate, ec.carTypeID, COUNT(cs.serviceID) AS "Number of Services"
	FROM ElectricCar ec, CarService cs
	WHERE ec.carVIN = cs.carVIN
	AND ec.carPurchaseDate >= ADD_MONTHS(SYSDATE, -12)
	AND cs.serviceDate BETWEEN DATE '2023-07-21' AND DATE '2024-07-21'
	GROUP BY ec.carVIN, ec.regNum, ec.battCap, ec.mileage, ec.kmReadingLastService, ec.carPurchaseDate, ec.carPurchaseCost, ec.carLastMaintenanceDate, ec.carTypeID
	ORDER BY "Number of Services" DESC;

CARVIN	REGNUM	BATTCAP	MILEAGE	KMREADINGLASTSERVICE	CARPURCHASEDATE	CARPURCHASECOST	CARLASTMAINTENANCEDATE	CARTYPEID	Number of Services
1HGCM82633A456789	VLW8295	54.2	27000	22000	08-AUG-2023	300000	30-JUN-2024	108	4

Important Notes:

- The SQL answers must be in the form of a screenshot that consists of both the SQL codes and the complete results. Please submit the screenshot with a white background.

- The queries must work with all reasonable sets of test data. For example, do not assume that each guest only makes one booking.

- When a user query asks for information related to a guest name, your SE-LECT statement should contain that name. Do not look up the corresponding ID or number and use that value in your answer.

- SQL codes that are excessively complicated or messy (i.e., not in a logical sequence) but yield correct results will have marks deducted.

- For verification purposes, each SQL code must produce some results.

Summary Of Tasks And Deliverables

Part	Description	What to submit
1 (10 marks) Generate a case scenario & draw an ERD	Use any AI generative tool to create a scenario of your choice Business rules –these rules establish the constraints needed in the SQL script in Part 3. Draw the ERD using draw.io to produce the model; do not include the attributes in the diagram. (5 marks will be deducted from this part for not following this instruction)	i. Your chosen scenario ii. ERD in Crowfoot with no attributes in each of the entities iii. Minimum TEN business rules
2 (35 marks) Design a relational database	Design a database that consists of the following steps: a) Identify the primary & alternate keys, and foreign keys (if any) for each table. b) Identify 5 indexes in the database and justify your choice c) Identify 5 user constraints in the database and justify your choice	a) Fill Table 1 for part a) b) Fill Table 2 for part b) c) Fill Table 3 for part c)
3 (15 marks) Implement a database	Use ORACLE APEX software to create a script called DBscript.txt Put all the necessary comments for easy referencing	DBscript with the following content · database tables · user data · index & check constraints
4 (20 marks) Query a database	For each sub-part of Part 4, ensure that it has a scenario based on the question requirement, SQL code, and results. Refer to Appendix 1	· create your own questions in scenario format · answer the given question · provide SQL codes and results screenshot
5 (20 marks) Demonstrate a database system	Each member would need to present his/her contribution to this submission	30-minute presentation; each member is expected to demonstrate their work

APPENDIX 1 (An example of Part 4 Question and Answer)

QUESTION

Write ONE question in a scenario format, and for the scenario, your SQL code needs to meet the criteria as stated in part a.

a. Write a user query with TWO-table join and one user condition. Sort the report in descending order.
b. Write a user query with TWO-table join and two user conditions.

ANSWER

Scenario a:

- The travel agency wishes to promote their upcoming holiday packages. As such a report about their customer's names and the current year travel booking(s) is required to enable the agency to make targeted promotions.

Scenario b:

- The travel agency wishes to call their customers regarding a promotion. As such the contact details are needed for such a task. Only customers who have been to Europe in the year 2023 will be contacted.

```
SELECT cname,cphone,cemail
FROM customer,purchase
WHERE customer.cid = purchase.cid
AND  purchase.destination='Europe'
AND purchase_date=…..
```

```
SELECT guestname DESC, destination DESC
FROM guest,booking
WHERE guest.guest_id=booking.guest_id
AND TO_CHAR(bookdate,'YYYY')=2024;
```

APPENDIX 2

Final assessment RUBRIC

Grade Scale:

- **A**: 70 and above
- **B**: 60-69
- **C**: 50-59
- **D**: 40-49
- **Fail**: Below 40

Part 1: Scenario and Business Rules (10 Marks)

- **0-2 Marks:** Scenario is missing or out of context.
- **3-5 Marks:** The scenario is relevant, but the ERD (Entity-Relationship Diagram) is incorrect or missing. Fewer than 10 general business rules are provided.
- **6-7 Marks:** The scenario is relevant with a correct ERD and 10 or more general business rules.
- **8-10 Marks:** Scenario is relevant, ERD is correct, and more than 10 specific business rules are included.

Examples:

- General rule: One student takes many subjects in a semester.
- Specific rule: One student takes at most 5 subjects in a semester.

Part 2: Tables and Answers (35 Marks)

- **0-14 Marks:** No answers provided for Tables 1-3, or less than 40% of answers are correct. For instance, less than 40% correct answers in Table 1 and only 2 correct answers in Tables 2 and 3.
- **15-20 Marks:** More than 50% correct answers in Table 1, and 3 correct answers in Tables 2 and 3.
- **21-24 Marks:** More than 60% correct answers in Table 1, and 4 correct answers in Tables 2 and 3.
- **25-28 Marks:** More than 70% correct answers in Table 1, and 5 correct answers in Tables 2 and 3.
- **29-35 Marks:** Correct answers provided for all entries in Tables 1-3.

Part 3: Database Script (15 Marks)

- **0-3 Marks:** No script provided, or the DB script cannot be executed due to bugs, with two or fewer indexes and check constraints.
- **4-7 Marks:** Executable DB script with 60% of database tables, poor quality user data, and up to 3 indexes and 3 check constraints.
- **8-10 Marks:** Executable DB script with 70% of database tables, average quality user data, and up to 4 indexes and 4 check constraints.
- **11-13 Marks:** Executable DB script with 80% of database tables, good quality user data, and up to 5 indexes and 5 check constraints.
- **14-15 Marks:** Executable DB script with 100% of database tables, poor quality user data, and more than 5 indexes and 5 check constraints.

Part 4: Scenarios and SQL Code (20 Marks)

- **0-5 Marks:** Did not attempt or only one scenario and its SQL codes and results are correct.
- **6-9 Marks:** Two scenarios with correct SQL codes and results.
- **10-12 Marks:** Three scenarios with correct SQL codes and results.
- **13-15 Marks:** Four scenarios with correct SQL codes and results.
- **16-20 Marks:** Five scenarios with correct SQL codes and results.

Part 5: Presentation and Practical Tasks (20 Marks)

- **0-5 Marks:** Did not attend the presentation or could only complete two of the following tasks:
  - Run 2 SQL codes from Part 4.
  - Perform 2 ad-hoc queries during the presentation.
  - Explain the database design regarding anomalies, dependencies, and normalization.
  - Demonstrate the use of 2 indexes and 2 constraints.
- **6-10 Marks:** Completed three of the tasks listed above.
- **11-15 Marks:** Completed four of the tasks listed above.
- **16-20 Marks:** Completed all of the following tasks:
  - Run 5 SQL codes from Part 4.
  - Perform 6 ad-hoc queries during the presentation.
  - Explain the database design regarding anomalies, dependencies, and normalization.
  - Demonstrate the use of more than 6 indexes and constraints.