



Sri Lanka Institute of Information Technology

IM2675-Database Management Systems

Group Assignment

Diaper Deal CRM: Database System Design

Submitted by:

Student Id	Student Name
BM23661042	Shamla M T B F
BM23673236	Fonseka R T C
BM23598096	Dissanayake D M P C
BM23742154	Silva S.H.P.
BM23610866	Elton S M
BM23741850	Knowlton D R S

Submitted to:

Ms. Thisuru Dias

Date of Submission:

12th October 2025

1. Table of Contents

1.	About The Company and The System	3
2.	Problem Identification & Objectives of the CRM System	4
3.	Entity Relationship Diagram.....	5
4.	Relational Schema	7
5.	SQL Commands.....	8
1.	Database Creation	8
2.	Database Table Creation	8
3.	Database Data Entry	13
6.	Sample SQL Queries (10 Queries)	20

1. About The Company and The System

Diaper Deal is a small-scale Sri Lankan business that offers baby care and hygiene products, including diapers, and other related essentials. The company mainly serves parents and caregivers across the country by offering a wide range of affordable and quality products suitable for infants and toddlers. At present, Diaper Deal handles most of its operations manually, including customer information, order management, and product details. This traditional method of managing data through spreadsheets and written records often causes inefficiencies such as data duplication, communication delays, and difficulty in maintaining accurate customer histories.

As a growing business, Diaper Deal requires a more organized system to manage its customer relationships and operations effectively. Therefore, our group proposed the development of a **Customer Relationship Management (CRM) System** that will help the company centralize customer data, monitor interactions, support marketing campaigns, and enhance overall customer satisfaction and loyalty.

The requirements for the CRM system of Diaper Deal.

Entity	Attributes
Customer	Customer ID, Fname, Lname, Email, Address
Customer Phone	Customer ID, Phone Number
Products	Product ID, Product Name, Price
Orders	Order ID, Customer ID, Order Date, Total Amount
Order Details	Order Details ID, Order ID, Product ID, Quantity, Total Amount
Invoice	Invoice ID, Order ID, Amount, Quantity
Payments	Payment ID, Invoice ID, Amount, PayDate
Tax	Tax ID, Invoice ID, Tax Type, Rate, Amount
Delivery	Delivery ID, Order ID, Delivery Date, Delivery Address
Feedback Rating	Feedback ID, Customer ID, Rating, Feedback Date
Loyalty Point	Loyalty ID, Customer ID, Points Earned, Points Redeemed
Marketing Campaign	Campaign ID, Campaign Type, Budget, Start Date, End Date
Campaign Targets	Campaign ID, Customer ID

2. Problem Identification & Objectives of the CRM System

Problem Identification

Diaper Deal currently manages its customer data, order records, and product information manually using spreadsheets and paper-based methods. This approach presents several challenges;

- Manual entry increases the risk of errors and duplication in customer and order records.
- It is difficult to track customer interactions, purchase history, and feedback systematically.
- Without organized data, analyzing customer behavior or running targeted campaigns is challenging.
- Responding to customer queries and managing promotions is slower due to scattered information.

These challenges delay the company's ability to provide timely service, improve customer satisfaction, and make informed business decisions.

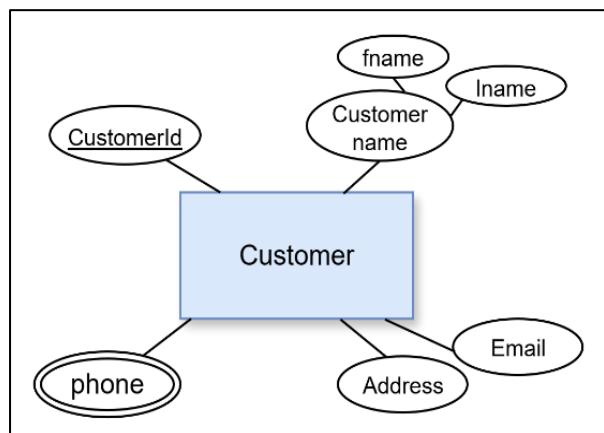
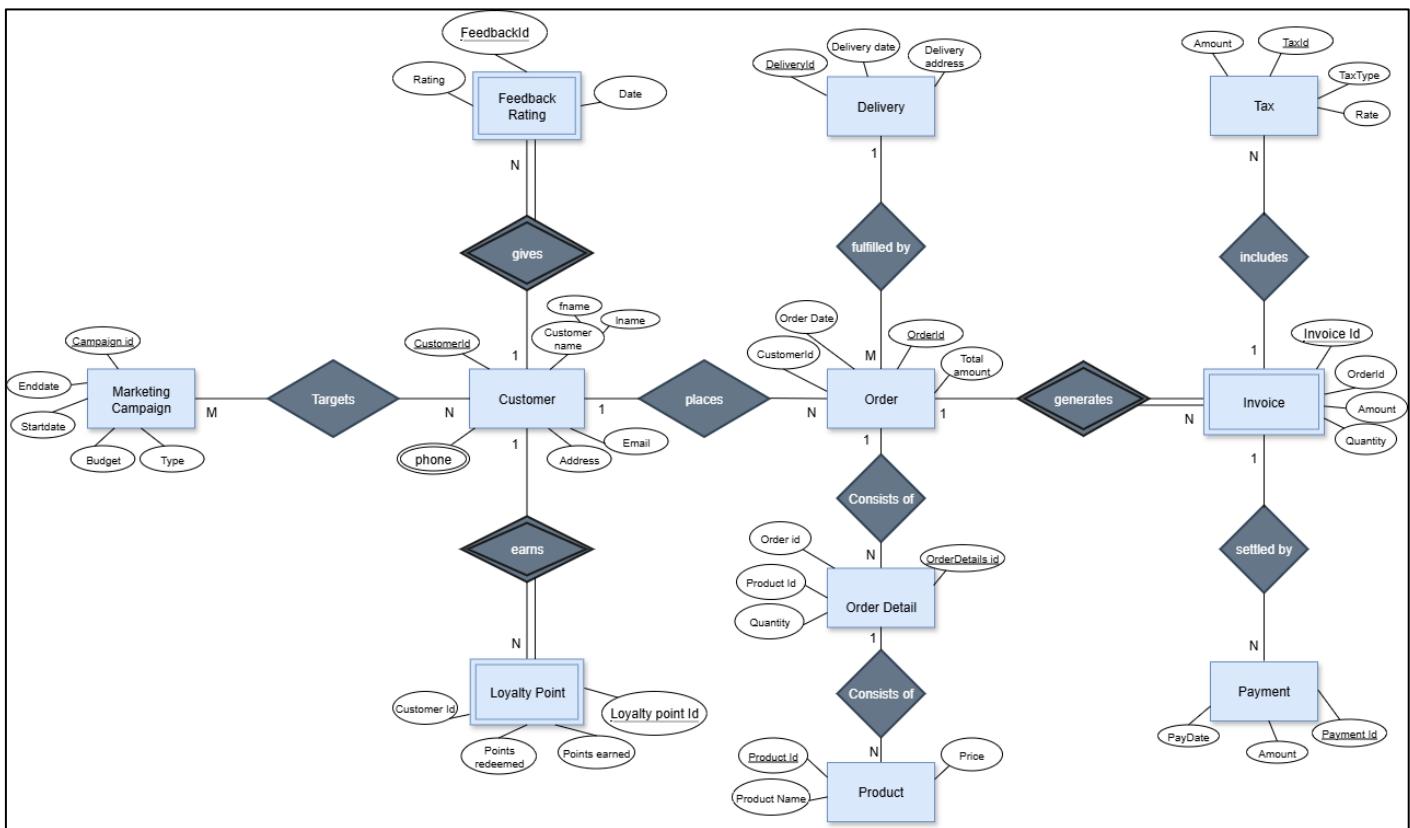
Objectives of the CRM System

The proposed CRM system aims to overcome these challenges by;

- Storing all customer information, orders, and interactions in a structured database.
- Tracking communications and feedback to provide timely and personalized responses.
- Recording campaign details, analyzing customer engagement, and identifying potential leads.
- Reducing manual work, minimizing errors, and providing accurate reports for decision making.
- By understanding customer preferences and purchase patterns, the company can strengthen long term relationships.

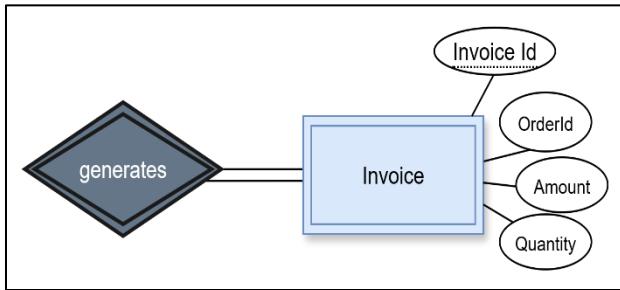
By implementing this CRM system, Diaper Deal will be able to streamline operations, enhance customer experience, and make data driven business decisions effectively.

3. Entity Relationship Diagram

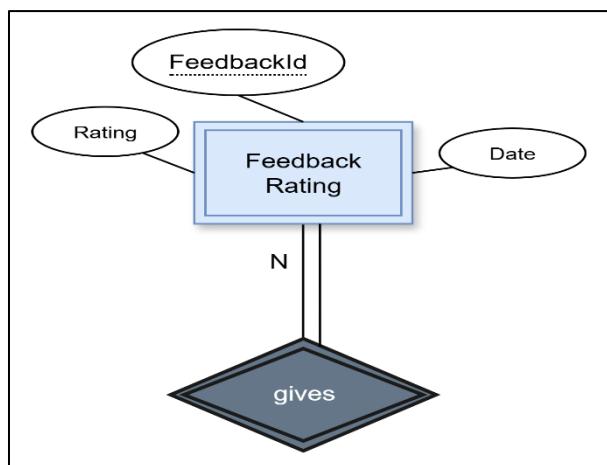


The customer entity has a *composite attribute* **Customer name** which can be further broken down into smaller components- **fname** and **lname**. This provides more detailed information about the customer name.

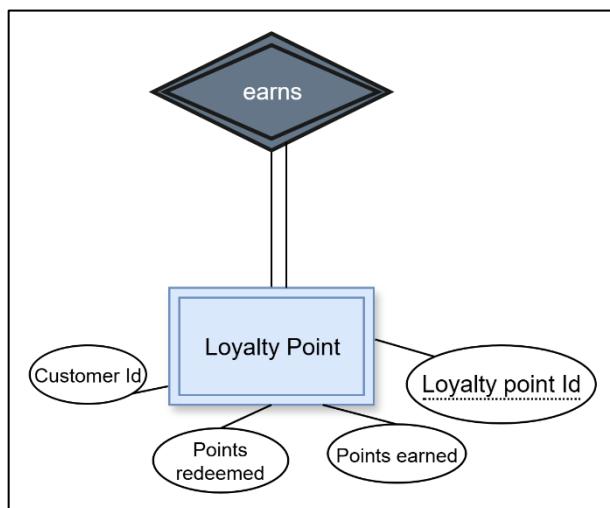
Moreover the attribute **phone** is a *multivalued attribute* which means it can have more than one value, as customers can have more than one phone number.



The entity **Invoice** is a *weak entity* that depends on **Order** for its identification through the *identifying relationship* generates, this is because each invoice is generated based on a specific order.

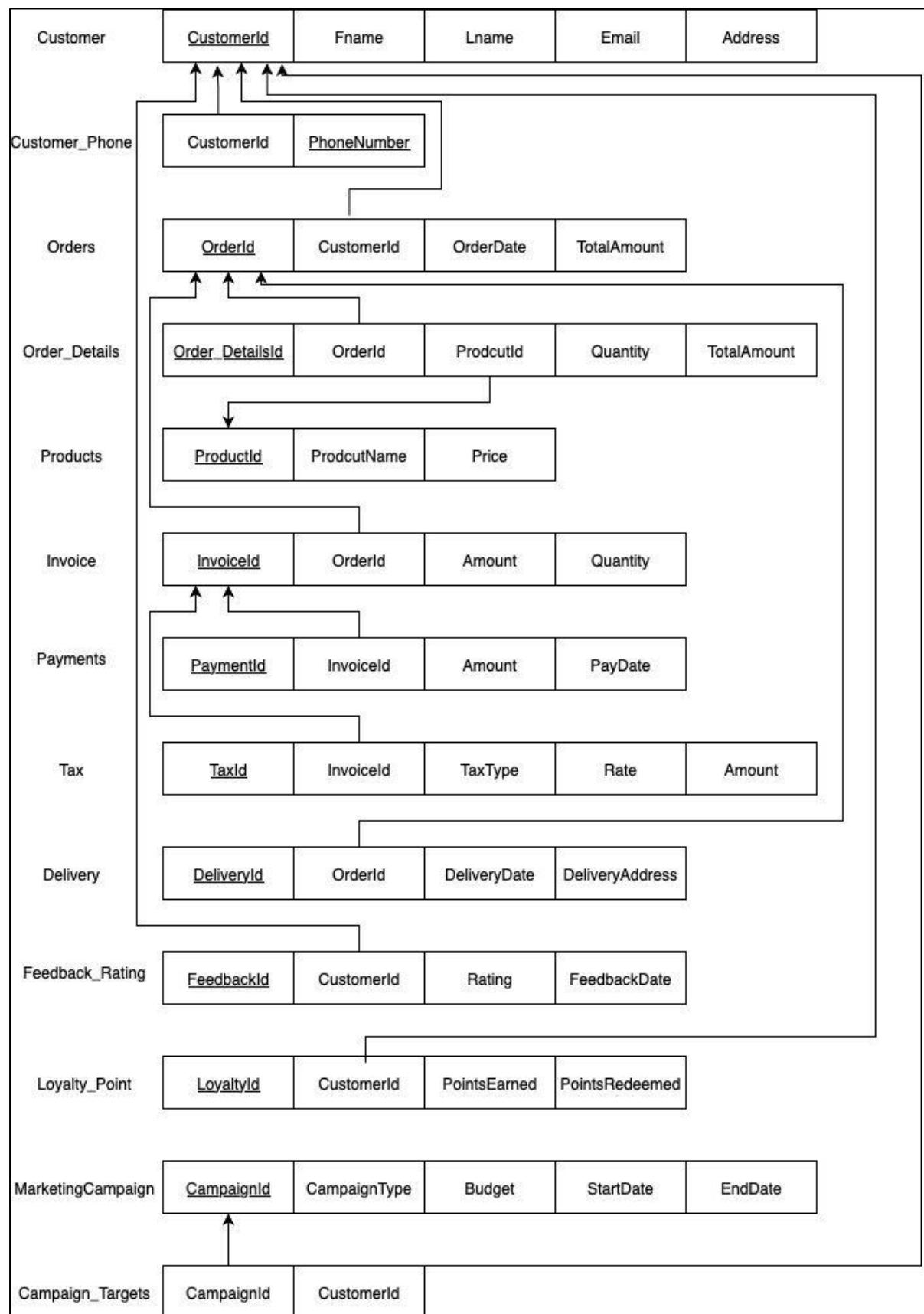


The entity **Feedback Rating** is a *weak entity* dependent on the strong entity **Customer**, it is in an *identifying relationship* through gives, this specifies that each feedback rating is given by a customer.



The *weak entity* **Loyalty Point** depends on the strong entity **Customer** and is related to customer through the *identifying relationship* earns as loyalty point is earned by a customer.

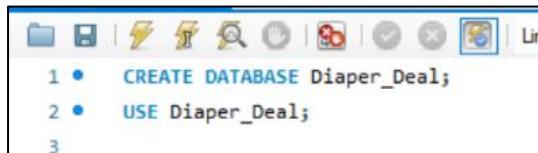
4. Relational Schema



5. SQL Commands

1. Database Creation

```
CREATE DATABASE Diaper_Deal;  
USE Diaper_Deal;
```



A screenshot of a SQL management tool interface. The toolbar at the top includes icons for file, database, search, and other common functions. Below the toolbar, the code for creating and selecting a database is displayed in a text area:

```
1 • CREATE DATABASE Diaper_Deal;  
2 • USE Diaper_Deal;  
3
```

2. Database Table Creation

CUSTOMER TABLE

```
CREATE TABLE Customer (  
    CustomerId VARCHAR(50) PRIMARY KEY,  
    FName VARCHAR(50) NOT NULL,  
    LName VARCHAR(50) NOT NULL,  
    Email VARCHAR(60) NOT NULL,  
    Address VARCHAR(120) NOT NULL  
);
```

```
4 • CREATE TABLE Customer (  
5     CustomerId VARCHAR(50) PRIMARY KEY,  
6     FName VARCHAR(50) NOT NULL,  
7     LName VARCHAR(50) NOT NULL,  
8     Email VARCHAR(60) NOT NULL,  
9     Address VARCHAR(120) NOT NULL  
10    );
```

CUSTOMER PHONE TABLE

```
CREATE TABLE Customer_Phone (  
    CustomerId VARCHAR(50) NOT NULL,  
    PhoneNumber VARCHAR(15) NOT NULL,  
    CONSTRAINT Customer_Phone_pk PRIMARY KEY (CustomerId, PhoneNumber),  
    CONSTRAINT Customer_Phone_fk FOREIGN KEY (CustomerId) REFERENCES Customer(CustomerId)  
);
```

```
12 • CREATE TABLE Customer_Phone (  
13     CustomerId VARCHAR(50) NOT NULL,  
14     PhoneNumber VARCHAR(15) NOT NULL,  
15     CONSTRAINT Customer_Phone_pk PRIMARY KEY (CustomerId, PhoneNumber),  
16     CONSTRAINT Customer_Phone_fk FOREIGN KEY (CustomerId) REFERENCES Customer(CustomerId)  
17    );
```

PRODUCTS TABLE

```
CREATE TABLE Products (
    ProductId VARCHAR(50) PRIMARY KEY,
    ProductName VARCHAR(50) NOT NULL,
    Price DECIMAL(10,2) NOT NULL
);
```

```
18
19 • ⏪ CREATE TABLE Products (
20     ProductId VARCHAR(50) PRIMARY KEY,
21     ProductName VARCHAR(50) NOT NULL,
22     Price DECIMAL(10,2) NOT NULL
23 );
```

ORDERS TABLE

```
CREATE TABLE Orders (
    OrderId VARCHAR(50) PRIMARY KEY,
    CustomerId VARCHAR(50) NOT NULL,
    OrderDate DATE NOT NULL,
    TotalAmount DECIMAL(10,2),
    CONSTRAINT Orders_fk FOREIGN KEY (CustomerId) REFERENCES Customer(CustomerId)
);
```

```
25 • ⏪ CREATE TABLE Orders (
26     OrderId VARCHAR(50) PRIMARY KEY,
27     CustomerId VARCHAR(50) NOT NULL,
28     OrderDate DATE NOT NULL,
29     TotalAmount DECIMAL(10,2),
30     CONSTRAINT Orders_fk FOREIGN KEY (CustomerId) REFERENCES Customer(CustomerId)
31 );
```

ORDER DETAILS TABLE

```
CREATE TABLE Order_Details (
    Order_DetailsId VARCHAR(50) PRIMARY KEY,
    OrderID VARCHAR(50) NOT NULL,
    ProductId VARCHAR(50) NOT NULL,
    Quantity INT NOT NULL,
    TotalAmount DECIMAL(10,2) NOT NULL,
    CONSTRAINT OrderDetails_fk1 FOREIGN KEY (OrderId) REFERENCES Orders(OrderId),
    CONSTRAINT OrderDetails_fk2 FOREIGN KEY (ProductId) REFERENCES Products(ProductId)
);
```

```
33 • ⏪ CREATE TABLE Order_Details (
34     Order_DetailsId VARCHAR(50) PRIMARY KEY,
35     OrderID VARCHAR(50) NOT NULL,
36     ProductId VARCHAR(50) NOT NULL,
37     Quantity INT NOT NULL,
38     TotalAmount DECIMAL(10,2) NOT NULL,
39     CONSTRAINT OrderDetails_fk1 FOREIGN KEY (OrderId) REFERENCES Orders(OrderId),
40     CONSTRAINT OrderDetails_fk2 FOREIGN KEY (ProductId) REFERENCES Products(ProductId)
41 );
```

INVOICE TABLE

```
CREATE TABLE Invoice (
    InvoiceId VARCHAR(50) PRIMARY KEY,
    OrderId VARCHAR(50) NOT NULL,
    Amount DECIMAL(10,2) NOT NULL,
    Quantity INT NOT NULL,
    CONSTRAINT Invoice_fk1 FOREIGN KEY (OrderId) REFERENCES Orders(OrderId)
);
```

```
43 • CREATE TABLE Invoice (
44     InvoiceId VARCHAR(50) PRIMARY KEY,
45     OrderId VARCHAR(50) NOT NULL,
46     Amount DECIMAL(10,2) NOT NULL,
47     Quantity INT NOT NULL,
48     CONSTRAINT Invoice_fk1 FOREIGN KEY (OrderId) REFERENCES Orders(OrderId)
49 );
```

PAYMENT TABLE

```
CREATE TABLE Payment (
    PaymentId VARCHAR(50) PRIMARY KEY,
    Amount DECIMAL(10,2) NOT NULL,
    InvoiceId VARCHAR(50) NOT NULL,
    PayDate DATE NOT NULL,
    CONSTRAINT Payment_fk1 FOREIGN KEY (InvoiceID) REFERENCES Invoice(InvoiceId)
);
```

```
51 • CREATE TABLE Payment (
52     PaymentId VARCHAR(50) PRIMARY KEY,
53     Amount DECIMAL(10,2) NOT NULL,
54     InvoiceId VARCHAR(50) NOT NULL,
55     PayDate DATE NOT NULL,
56     CONSTRAINT Payment_fk1 FOREIGN KEY (InvoiceID) REFERENCES Invoice(InvoiceId)
57 );
```

TAX TABLE

```
CREATE TABLE Tax (
    TaxId VARCHAR(50) PRIMARY KEY,
    InvoiceId VARCHAR(50) NOT NULL,
    TaxType VARCHAR(20) NOT NULL,
    Rate DECIMAL(4,2) NOT NULL,
    Amount DECIMAL(10,2) NOT NULL,
    CONSTRAINT Tax_fk1 FOREIGN KEY (InvoiceID) REFERENCES Invoice(InvoiceId)
);
```

```
59 • CREATE TABLE Tax (
60     TaxId VARCHAR(50) PRIMARY KEY,
61     InvoiceId VARCHAR(50) NOT NULL,
62     TaxType VARCHAR(20) NOT NULL,
63     Rate DECIMAL(4,2) NOT NULL,
64     Amount DECIMAL(10,2) NOT NULL,
65     CONSTRAINT Tax_fk1 FOREIGN KEY (InvoiceID) REFERENCES Invoice(InvoiceId)
66 );
```

DELIVERY TABLE

```
CREATE TABLE Delivery (
    DeliveryId VARCHAR(50) PRIMARY KEY,
    OrderId VARCHAR(50) NOT NULL,
    DeliveryDate DATE NOT NULL,
    DeliveryAddress VARCHAR(120) NOT NULL,
    CONSTRAINT Delivery_fk FOREIGN KEY (OrderId) REFERENCES Orders(OrderId)
);
```

```
68 • Ⓜ CREATE TABLE Delivery (
69     DeliveryId VARCHAR(50) PRIMARY KEY,
70     OrderId VARCHAR(50) NOT NULL,
71     DeliveryDate DATE NOT NULL,
72     DeliveryAddress VARCHAR(120) NOT NULL,
73     CONSTRAINT Delivery_fk FOREIGN KEY (OrderId) REFERENCES Orders(OrderId)
74 );
```

FEEDBACK RATING TABLE

```
CREATE TABLE Feedback_Rating (
    FeedbackId VARCHAR(50) PRIMARY KEY,
    CustomerId VARCHAR(50) NOT NULL,
    Rating INT NOT NULL CHECK (Rating BETWEEN 1 AND 5),
    FeedbackDate DATE NOT NULL,
    CONSTRAINT Feedback_fk FOREIGN KEY (CustomerId) REFERENCES Customer(CustomerId)
);
```

```
76 • Ⓜ CREATE TABLE Feedback_Rating (
77     FeedbackId VARCHAR(50) PRIMARY KEY,
78     CustomerId VARCHAR(50) NOT NULL,
79     Rating INT NOT NULL CHECK (Rating BETWEEN 1 AND 5),
80     FeedbackDate DATE NOT NULL,
81     CONSTRAINT Feedback_fk FOREIGN KEY (CustomerId) REFERENCES Customer(CustomerId)
82 );
```

LOYALTY POINT TABLE

```
CREATE TABLE Loyalty_Point (
    LoyaltyId VARCHAR(50) PRIMARY KEY,
    CustomerId VARCHAR(50) NOT NULL,
    PointsEarned INT NOT NULL,
    PointsRedeemed INT NOT NULL,
    CONSTRAINT Loyalty_Point_fk FOREIGN KEY (CustomerId) REFERENCES Customer(CustomerId)
);
```

```
84 • Ⓜ CREATE TABLE Loyalty_Point (
85     LoyaltyId VARCHAR(50) PRIMARY KEY,
86     CustomerId VARCHAR(50) NOT NULL,
87     PointsEarned INT NOT NULL,
88     PointsRedeemed INT NOT NULL,
89     CONSTRAINT Loyalty_Point_fk FOREIGN KEY (CustomerId) REFERENCES Customer(CustomerId)
90 );
```

MARKETING CAMPAIGN TABLE

```
CREATE TABLE MarketingCampaign (
    CampaignId VARCHAR(50) PRIMARY KEY,
    CampaignType VARCHAR(50) NOT NULL,
    Budget DECIMAL(10,2),
    StartDate DATE NOT NULL,
    EndDate DATE NOT NULL
);
```

```
92 • ⏪ CREATE TABLE MarketingCampaign (
93     CampaignId VARCHAR(50) PRIMARY KEY,
94     CampaignType VARCHAR(50) NOT NULL,
95     Budget DECIMAL(10,2),
96     StartDate DATE NOT NULL,
97     EndDate DATE NOT NULL
98 );
```

CAMPAIGN TARGETS TABLE

```
CREATE TABLE Campaign_TarGets (
    CampaignID VARCHAR(50),
    CustomerID VARCHAR(50),
    CONSTRAINT Campaign_TarGets_pk1 PRIMARY KEY (CampaignID, CustomerID),
    CONSTRAINT Campaign_TarGets_fk1 FOREIGN KEY (CampaignID) REFERENCES
MarketingCampaign(CampaignID),
    CONSTRAINT Campaign_TarGets_fk2 FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
);
```

```
100 • ⏪ CREATE TABLE Campaign_TarGets (
101     CampaignID VARCHAR(50),
102     CustomerID VARCHAR(50),
103     CONSTRAINT Campaign_TarGets_pk1 PRIMARY KEY (CampaignID, CustomerID),
104     CONSTRAINT Campaign_TarGets_fk1 FOREIGN KEY (CampaignID) REFERENCES MarketingCampaign(CampaignID),
105     CONSTRAINT Campaign_TarGets_fk2 FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
106 );
```

3. Database Data Entry

CUSTOMER DATA

```
INSERT INTO Customer (CustomerId, Fname, Lname, Email, Address) VALUES
('DD-CUST-25-0001', 'Supuni', 'Adikari', 'supunikavinda21@gmail.com', 'No. 12/6, Lake View Apartments, Tudella, Ja-Ela'),
('DD-CUST-25-0002', 'Tharini', 'Herath', 'thariniamanda1121@gmail.com', "Sisila", Kandana Road, Ja-Ela'),
('DD-CUST-25-0003', 'Hemashi', 'Siriwardana', 'Hemashigeethma0101@gmail.com', 'No. 77/4, St. Mary's Lane, Negombo Town'),
('DD-CUST-25-0004', 'Amiru', 'Gunesekara', 'amirupasandul2002@gmail.com', "Sunset Villa", Beach Road, Periyamulla),
('DD-CUST-25-0005', 'Adithya', 'Subharathi', 'adiliyanage246@gmail.com', 'No. 84/1, Beach Road, Ettukala, Negombo'),
('DD-CUST-25-0006', 'Madusara', 'Anuhas', 'madusarapiumantha@gmail.com', 'No. 33/5, Riverside Homes, Kurunegala Road, Kuliyapitiya'),
('DD-CUST-25-0007', 'Heshan', 'Sriyananda', 'sriyanandaheshan@gmail.com', 'No. 56/2, Riverside Apartments, Periyamulla, Negombo'),
('DD-CUST-25-0008', 'Rangi', 'Liyanage', 'rangi.liyanage16@gmail.com', "Sagara Villa", Colombo Road, Seeduwa'),
('DD-CUST-25-0009', 'Onel', 'Hettiarachchi', 'onelvh@gmail.com', '171/C3 , Palangathure, Negombo'),
('DD-CUST-25-0010', 'Hareesha', 'Jayarathna', 'hareeshajayarathna@gmail.com', '428 Morawatta Rd, Kandana');
```

```
108 • INSERT INTO Customer (CustomerId, Fname, Lname, Email, Address) VALUES
109   ('DD-CUST-25-0001', 'Supuni', 'Adikari', 'supunikavinda21@gmail.com', 'No. 12/6, Lake View Apartments, Tudella, Ja-Ela'),
110   ('DD-CUST-25-0002', 'Tharini', 'Herath', 'thariniamanda1121@gmail.com', "Sisila", Kandana Road, Ja-Ela'),
111   ('DD-CUST-25-0003', 'Hemashi', 'Siriwardana', 'Hemashigeethma0101@gmail.com', 'No. 77/4, St. Mary's Lane, Negombo Town'),
112   ('DD-CUST-25-0004', 'Amiru', 'Gunesekara', 'amirupasandul2002@gmail.com', "Sunset Villa", Beach Road, Periyamulla),
113   ('DD-CUST-25-0005', 'Adithya', 'Subharathi', 'adiliyanage246@gmail.com', 'No. 84/1, Beach Road, Ettukala, Negombo'),
114   ('DD-CUST-25-0006', 'Madusara', 'Anuhas', 'madusarapiumantha@gmail.com', 'No. 33/5, Riverside Homes, Kurunegala Road, Kuliyapitiya'),
115   ('DD-CUST-25-0007', 'Heshan', 'Sriyananda', 'sriyanandaheshan@gmail.com', 'No. 56/2, Riverside Apartments, Periyamulla, Negombo'),
116   ('DD-CUST-25-0008', 'Rangi', 'Liyanage', 'rangi.liyanage16@gmail.com', "Sagara Villa", Colombo Road, Seeduwa'),
117   ('DD-CUST-25-0009', 'Onel', 'Hettiarachchi', 'onelvh@gmail.com', '171/C3 , Palangathure, Negombo'),
118   ('DD-CUST-25-0010', 'Hareesha', 'Jayarathna', 'hareeshajayarathna@gmail.com', '428 Morawatta Rd, Kandana');
```

CUSTOMER PHONE DATA

```
INSERT INTO Customer_Phone (CustomerId, PhoneNumber) VALUES
('DD-CUST-25-0001', '750778590'),
('DD-CUST-25-0002', '754438491'),
('DD-CUST-25-0003', '717892407'),
('DD-CUST-25-0004', '775912544'),
('DD-CUST-25-0005', '767490820'),
('DD-CUST-25-0006', '760090746'),
('DD-CUST-25-0007', '775044983'),
('DD-CUST-25-0008', '770330254'),
('DD-CUST-25-0009', '728056547'),
('DD-CUST-25-0010', '751410883');
```

```
120 • INSERT INTO Customer_Phone (CustomerId, PhoneNumber) VALUES
121   ('DD-CUST-25-0001', '750778590'),
122   ('DD-CUST-25-0002', '754438491'),
123   ('DD-CUST-25-0003', '717892407'),
124   ('DD-CUST-25-0004', '775912544'),
125   ('DD-CUST-25-0005', '767490820'),
126   ('DD-CUST-25-0006', '760090746'),
127   ('DD-CUST-25-0007', '775044983'),
128   ('DD-CUST-25-0008', '770330254'),
129   ('DD-CUST-25-0009', '728056547'),
130   ('DD-CUST-25-0010', '751410883');
```

PRODUCTS DATA

```
INSERT INTO Products (ProductId, ProductName, Price) VALUES  
('P1001', 'Velona Cuddles Baby Diaper', 3230),  
('P1002', 'Marvel Baby Diaper', 3600),  
('P1003', 'Adfit Adult', 3150),  
('P1004', 'Friends Adult Small', 1800),  
('P1005', 'Velona Cuddles Baby Pants', 6510),  
('P1006', 'Velona Cuddles Baby Diaper', 3590),  
('P1007', 'Manytime Adult Medium', 2100),  
('P1008', 'Carewell Adult Medium', 2100),  
('P1009', 'Manytime Adult Diaper', 1900),  
('P1010', 'Pampers Medium', 2600);
```

```
132 • INSERT INTO Products (ProductId, ProductName, Price) VALUES  
133 ('P1001', 'Velona Cuddles Baby Diaper', 3230),  
134 ('P1002', 'Marvel Baby Diaper', 3600),  
135 ('P1003', 'Adfit Adult', 3150),  
136 ('P1004', 'Friends Adult Small', 1800),  
137 ('P1005', 'Velona Cuddles Baby Pants', 6510),  
138 ('P1006', 'Velona Cuddles Baby Diaper', 3590),  
139 ('P1007', 'Manytime Adult Medium', 2100),  
140 ('P1008', 'Carewell Adult Medium', 2100),  
141 ('P1009', 'Manytime Adult Diaper', 1900),  
142 ('P1010', 'Pampers Medium', 2600);
```

ORDERS DATA

```
INSERT INTO Orders (OrderId, CustomerId, OrderDate, TotalAmount) VALUES  
('D14030', 'DD-CUST-25-0001', '2025-09-03', 6300),  
('D14031', 'DD-CUST-25-0002', '2025-10-11', 2100),  
('D14032', 'DD-CUST-25-0003', '2025-11-22', 6510),  
('D14033', 'DD-CUST-25-0004', '2025-12-09', 9690),  
('D14034', 'DD-CUST-25-0005', '2025-10-25', 3600),  
('D14035', 'DD-CUST-25-0006', '2025-11-05', 7180),  
('D14036', 'DD-CUST-25-0007', '2025-09-19', 7800),  
('D14037', 'DD-CUST-25-0008', '2025-12-17', 7600),  
('D14038', 'DD-CUST-25-0009', '2025-10-07', 3600),  
('D14039', 'DD-CUST-25-0010', '2025-11-28', 4200);
```

```
144 • INSERT INTO Orders (OrderId, CustomerId, OrderDate, TotalAmount) VALUES  
145 ('D14030', 'DD-CUST-25-0001', '2025-09-03', 6300),  
146 ('D14031', 'DD-CUST-25-0002', '2025-10-11', 2100),  
147 ('D14032', 'DD-CUST-25-0003', '2025-11-22', 6510),  
148 ('D14033', 'DD-CUST-25-0004', '2025-12-09', 9690),  
149 ('D14034', 'DD-CUST-25-0005', '2025-10-25', 3600),  
150 ('D14035', 'DD-CUST-25-0006', '2025-11-05', 7180),  
151 ('D14036', 'DD-CUST-25-0007', '2025-09-19', 7800),  
152 ('D14037', 'DD-CUST-25-0008', '2025-12-17', 7600),  
153 ('D14038', 'DD-CUST-25-0009', '2025-10-07', 3600),  
154 ('D14039', 'DD-CUST-25-0010', '2025-11-28', 4200);
```

ORDER DETAILS DATA

```
INSERT INTO Order_Details (Order_DetailsId, OrderID, ProductId, Quantity, TotalAmount) VALUES  
('OD2501', 'D14030', 'P1003', 2, 6300),  
('OD2502', 'D14031', 'P1007', 1, 2100),  
('OD2503', 'D14032', 'P1005', 1, 6510),  
('OD2504', 'D14033', 'P1001', 3, 9690),  
('OD2505', 'D14034', 'P1002', 1, 3600),  
('OD2506', 'D14035', 'P1006', 2, 7180),  
('OD2507', 'D14036', 'P1010', 3, 7800),  
('OD2508', 'D14037', 'P1009', 4, 7600),  
('OD2509', 'D14038', 'P1004', 2, 3600),  
('OD2510', 'D14039', 'P1008', 2, 4200);
```

```
156 • INSERT INTO Order_Details (Order_DetailsId, OrderID, ProductId, Quantity, TotalAmount) VALUES  
157 ('OD2501', 'D14030', 'P1003', 2, 6300),  
158 ('OD2502', 'D14031', 'P1007', 1, 2100),  
159 ('OD2503', 'D14032', 'P1005', 1, 6510),  
160 ('OD2504', 'D14033', 'P1001', 3, 9690),  
161 ('OD2505', 'D14034', 'P1002', 1, 3600),  
162 ('OD2506', 'D14035', 'P1006', 2, 7180),  
163 ('OD2507', 'D14036', 'P1010', 3, 7800),  
164 ('OD2508', 'D14037', 'P1009', 4, 7600),  
165 ('OD2509', 'D14038', 'P1004', 2, 3600),  
166 ('OD2510', 'D14039', 'P1008', 2, 4200);
```

INVOICE DATA

```
INSERT INTO Invoice (InvoiceId, OrderId, Amount, Quantity)  
VALUES  
('INV001', 'D14030', 6300, 2),  
('INV002', 'D14031', 2100, 1),  
('INV003', 'D14032', 6510, 1),  
('INV004', 'D14033', 9690, 3),  
('INV005', 'D14034', 3600, 1),  
('INV006', 'D14035', 7180, 2),  
('INV007', 'D14036', 7800, 3),  
('INV008', 'D14037', 7600, 4),  
('INV009', 'D14038', 3600, 2),  
('INV010', 'D14039', 4200, 2);
```

```
168 • INSERT INTO Invoice (InvoiceId, OrderId, Amount, Quantity) VALUES  
169 ('INV001', 'D14030', 6300, 2),  
170 ('INV002', 'D14031', 2100, 1),  
171 ('INV003', 'D14032', 6510, 1),  
172 ('INV004', 'D14033', 9690, 3),  
173 ('INV005', 'D14034', 3600, 1),  
174 ('INV006', 'D14035', 7180, 2),  
175 ('INV007', 'D14036', 7800, 3),  
176 ('INV008', 'D14037', 7600, 4),  
177 ('INV009', 'D14038', 3600, 2),  
178 ('INV010', 'D14039', 4200, 2);
```

PAYMENTS DATA

```
INSERT INTO Payment (PaymentId, Amount, InvoiceId, PayDate) VALUES
('PAY001', 6300, 'INV002', '2025-09-14'),
('PAY002', 2100, 'INV005', '2025-10-18'),
('PAY003', 6510, 'INV003', '2025-11-08'),
('PAY004', 9690, 'INV001', '2025-10-01'),
('PAY005', 3600, 'INV009', '2025-10-27'),
('PAY006', 7180, 'INV006', '2025-11-29'),
('PAY007', 7800, 'INV010', '2025-11-13'),
('PAY008', 7600, 'INV007', '2025-09-26'),
('PAY009', 3600, 'INV004', '2025-12-20'),
('PAY010', 4200, 'INV008', '2025-12-04');
```

```
180 • INSERT INTO Payment (PaymentId, Amount, InvoiceId, PayDate) VALUES
181 ('PAY001', 6300, 'INV002', '2025-09-14'),
182 ('PAY002', 2100, 'INV005', '2025-10-18'),
183 ('PAY003', 6510, 'INV003', '2025-11-08'),
184 ('PAY004', 9690, 'INV001', '2025-10-01'),
185 ('PAY005', 3600, 'INV009', '2025-10-27'),
186 ('PAY006', 7180, 'INV006', '2025-11-29'),
187 ('PAY007', 7800, 'INV010', '2025-11-13'),
188 ('PAY008', 7600, 'INV007', '2025-09-26'),
189 ('PAY009', 3600, 'INV004', '2025-12-20'),
190 ('PAY010', 4200, 'INV008', '2025-12-04');
```

TAX DATA

```
INSERT INTO Tax (TaxId, InvoiceId, TaxType, Rate, Amount) VALUES
('TX1001', 'INV003', 'VAT', 0.18, 6300),
('TX1002', 'INV004', 'VAT', 0.18, 2100),
('TX1003', 'INV005', 'VAT', 0.18, 6510),
('TX1004', 'INV001', 'VAT', 0.18, 9690),
('TX1005', 'INV008', 'VAT', 0.18, 3600),
('TX1006', 'INV006', 'VAT', 0.18, 7180),
('TX1007', 'INV009', 'VAT', 0.18, 7800),
('TX1008', 'INV010', 'VAT', 0.18, 7600),
('TX1009', 'INV002', 'VAT', 0.18, 3600),
('TX1010', 'INV007', 'VAT', 0.18, 4200);
```

```
192 • INSERT INTO Tax (TaxId, InvoiceId, TaxType, Rate, Amount) VALUES
193 ('TX1001', 'INV003', 'VAT', 0.18, 6300),
194 ('TX1002', 'INV004', 'VAT', 0.18, 2100),
195 ('TX1003', 'INV005', 'VAT', 0.18, 6510),
196 ('TX1004', 'INV001', 'VAT', 0.18, 9690),
197 ('TX1005', 'INV008', 'VAT', 0.18, 3600),
198 ('TX1006', 'INV006', 'VAT', 0.18, 7180),
199 ('TX1007', 'INV009', 'VAT', 0.18, 7800),
200 ('TX1008', 'INV010', 'VAT', 0.18, 7600),
201 ('TX1009', 'INV002', 'VAT', 0.18, 3600),
202 ('TX1010', 'INV007', 'VAT', 0.18, 4200);
```

DELIVERY DATA

```
INSERT INTO Delivery (DeliveryId, OrderId, DeliveryDate, DeliveryAddress) VALUES
('#122040', 'D14030', '2025-10-02', 'No. 12/6, Lake View Apartments, Tudella, Ja-Ela'),
('#122041', 'D14031', '2025-09-15', 'Sisila, Kandana Road, Ja-Ela'),
('#122042', 'D14032', '2025-11-08', 'No. 77/4, St. Mary's Lane, Negombo Town'),
('#122043', 'D14033', '2025-12-21', 'Sunset Villa, Beach Road, Periyamulla'),
('#122044', 'D14034', '2025-10-19', 'No. 84/1, Beach Road, Ettukala, Negombo'),
('#122045', 'D14035', '2025-11-30', 'No. 33/5, Riverside Homes, Kurunegala Road, Kuliyapitiya'),
('#122046', 'D14036', '2025-09-27', 'No. 56/2, Riverside Apartments, Periyamulla, Negombo'),
('#122047', 'D14037', '2025-12-05', 'Sagara Villa, Colombo Road, Seeduwa'),
('#122048', 'D14038', '2025-10-28', '171/C3, Palangathure, Negombo'),
('#122049', 'D14039', '2025-11-14', '428 Morawatta Rd, Kandana');
```

```
204 • INSERT INTO Delivery (DeliveryId, OrderId, DeliveryDate, DeliveryAddress) VALUES
205 ('#122040', 'D14030', '2025-10-02', 'No. 12/6, Lake View Apartments, Tudella, Ja-Ela'),
206 ('#122041', 'D14031', '2025-09-15', 'Sisila, Kandana Road, Ja-Ela'),
207 ('#122042', 'D14032', '2025-11-08', 'No. 77/4, St. Mary's Lane, Negombo Town'),
208 ('#122043', 'D14033', '2025-12-21', 'Sunset Villa, Beach Road, Periyamulla'),
209 ('#122044', 'D14034', '2025-10-19', 'No. 84/1, Beach Road, Ettukala, Negombo'),
210 ('#122045', 'D14035', '2025-11-30', 'No. 33/5, Riverside Homes, Kurunegala Road, Kuliyapitiya'),
211 ('#122046', 'D14036', '2025-09-27', 'No. 56/2, Riverside Apartments, Periyamulla, Negombo'),
212 ('#122047', 'D14037', '2025-12-05', 'Sagara Villa, Colombo Road, Seeduwa'),
213 ('#122048', 'D14038', '2025-10-28', '171/C3, Palangathure, Negombo'),
214 ('#122049', 'D14039', '2025-11-14', '428 Morawatta Rd, Kandana');
```

FEEDBACK RATING DATA

```
INSERT INTO Feedback_Rating (FeedbackId, CustomerId, Rating, FeedbackDate) VALUES
('FBD-25-0001', 'DD-CUST-25-0001', 4, '2025-09-01'),
('FBD-25-0002', 'DD-CUST-25-0002', 5, '2025-09-03'),
('FBD-25-0003', 'DD-CUST-25-0003', 3, '2025-09-05'),
('FBD-25-0004', 'DD-CUST-25-0004', 4, '2025-09-06'),
('FBD-25-0005', 'DD-CUST-25-0005', 2, '2025-09-08'),
('FBD-25-0006', 'DD-CUST-25-0006', 5, '2025-09-09'),
('FBD-25-0007', 'DD-CUST-25-0007', 3, '2025-09-11'),
('FBD-25-0008', 'DD-CUST-25-0008', 4, '2025-09-12'),
('FBD-25-0009', 'DD-CUST-25-0009', 5, '2025-09-14'),
('FBD-25-0010', 'DD-CUST-25-0010', 3, '2025-09-15');
```

```
216 • INSERT INTO Feedback_Rating (FeedbackId, CustomerId, Rating, FeedbackDate) VALUES
217 ('FBD-25-0001', 'DD-CUST-25-0001', 4, '2025-09-01'),
218 ('FBD-25-0002', 'DD-CUST-25-0002', 5, '2025-09-03'),
219 ('FBD-25-0003', 'DD-CUST-25-0003', 3, '2025-09-05'),
220 ('FBD-25-0004', 'DD-CUST-25-0004', 4, '2025-09-06'),
221 ('FBD-25-0005', 'DD-CUST-25-0005', 2, '2025-09-08'),
222 ('FBD-25-0006', 'DD-CUST-25-0006', 5, '2025-09-09'),
223 ('FBD-25-0007', 'DD-CUST-25-0007', 3, '2025-09-11'),
224 ('FBD-25-0008', 'DD-CUST-25-0008', 4, '2025-09-12'),
225 ('FBD-25-0009', 'DD-CUST-25-0009', 5, '2025-09-14'),
226 ('FBD-25-0010', 'DD-CUST-25-0010', 3, '2025-09-15');
```

LOYALTY POINT DATA

```
INSERT INTO Loyalty_Point (LoyaltyId, CustomerId, PointsEarned, PointsRedeemed) VALUES
('LOY-25-7421', 'DD-CUST-25-0001', 500, 200),
('LOY-G-25-3905', 'DD-CUST-25-0002', 750, 150),
('LOY-25-8612', 'DD-CUST-25-0003', 620, 100),
('LOY-G-25-2549', 'DD-CUST-25-0004', 430, 130),
('LOY-25-1038', 'DD-CUST-25-0005', 900, 400),
('LOY-25-9754', 'DD-CUST-25-0006', 350, 50),
('LOY-25-6183', 'DD-CUST-25-0007', 800, 300),
('LOY-25-4307', 'DD-CUST-25-0008', 600, 200),
('LOY-25-5826', 'DD-CUST-25-0009', 700, 250),
('LOY-G-25-9071', 'DD-CUST-25-0010', 550, 100);
```

```
228 • INSERT INTO Loyalty_Point (LoyaltyId, CustomerId, PointsEarned, PointsRedeemed) VALUES
229   ('LOY-25-7421', 'DD-CUST-25-0001', 500, 200),
230   ('LOY-G-25-3905', 'DD-CUST-25-0002', 750, 150),
231   ('LOY-25-8612', 'DD-CUST-25-0003', 620, 100),
232   ('LOY-G-25-2549', 'DD-CUST-25-0004', 430, 130),
233   ('LOY-25-1038', 'DD-CUST-25-0005', 900, 400),
234   ('LOY-25-9754', 'DD-CUST-25-0006', 350, 50),
235   ('LOY-25-6183', 'DD-CUST-25-0007', 800, 300),
236   ('LOY-25-4307', 'DD-CUST-25-0008', 600, 200),
237   ('LOY-25-5826', 'DD-CUST-25-0009', 700, 250),
238   ('LOY-G-25-9071', 'DD-CUST-25-0010', 550, 100);
```

MARKETING CAMPAIGN DATA

```
INSERT INTO MarketingCampaign (CampaignId, CampaignType, Budget, StartDate, EndDate) VALUES
('CDD-25-7421', 'Mom & Baby Workshop', 5000.00, '2025-01-10', '2025-01-13'),
('CDD-25-3905', 'Instagram Story Contest', 2000.00, '2024-05-06', '2024-07-25'),
('CDD-25-8612', 'Baby Care Tips Newsletter', 1500.00, '2024-01-10', '2025-08-10'),
('CDD-25-2549', 'Diaper Discount Bundle', 650.00, '2022-03-10', '2025-12-25'),
('CDD-25-1038', 'Diaper Giveaway', 500.00, '2025-08-05', '2025-08-15'),
('CDD-25-9754', 'Mom Referral Program', 1000.00, '2024-11-11', '2025-12-11'),
('CDD-25-6183', 'Mother's Day Special Offer', 1000.00, '2025-05-11', '2025-05-11'),
('CDD-25-4307', 'Children's Day Special Offer', 1500.00, '2025-10-01', '2025-10-01'),
('CDD-25-5826', 'Adults Health Week', 5000.00, '2025-01-15', '2025-01-20'),
('CDD-25-9071', 'Annual diaper giveaway (CSR)', 750.00, '2024-01-01', '2025-12-31');
```

```
240 • INSERT INTO MarketingCampaign (CampaignId, CampaignType, Budget, StartDate, EndDate) VALUES
241   ('CDD-25-7421', 'Mom & Baby Workshop', 5000.00, '2025-01-10', '2025-01-13'),
242   ('CDD-25-3905', 'Instagram Story Contest', 2000.00, '2024-05-06', '2024-07-25'),
243   ('CDD-25-8612', 'Baby Care Tips Newsletter', 1500.00, '2024-01-10', '2025-08-10'),
244   ('CDD-25-2549', 'Diaper Discount Bundle', 650.00, '2022-03-10', '2025-12-25'),
245   ('CDD-25-1038', 'Diaper Giveaway', 500.00, '2025-08-05', '2025-08-15'),
246   ('CDD-25-9754', 'Mom Referral Program', 1000.00, '2024-11-11', '2025-12-11'),
247   ('CDD-25-6183', 'Mother's Day Special Offer', 1000.00, '2025-05-11', '2025-05-11'),
248   ('CDD-25-4307', 'Children's Day Special Offer', 1500.00, '2025-10-01', '2025-10-01'),
249   ('CDD-25-5826', 'Adults Health Week', 5000.00, '2025-01-15', '2025-01-20'),
250   ('CDD-25-9071', 'Annual diaper giveaway (CSR)', 750.00, '2024-01-01', '2025-12-31');
```

CAMPAIGN TARGETS DATA

```
INSERT INTO Campaign_TarGets (CampaignID, CustomerID) VALUES  
('CDD-25-7421', 'DD-CUST-25-0001'),  
('CDD-25-3905', 'DD-CUST-25-0002'),  
('CDD-25-8612', 'DD-CUST-25-0003'),  
('CDD-25-2549', 'DD-CUST-25-0004'),  
('CDD-25-1038', 'DD-CUST-25-0005'),  
('CDD-25-9754', 'DD-CUST-25-0006'),  
('CDD-25-6183', 'DD-CUST-25-0007'),  
('CDD-25-4307', 'DD-CUST-25-0008'),  
('CDD-25-5826', 'DD-CUST-25-0009'),  
('CDD-25-9071', 'DD-CUST-25-0010');
```

```
252 • INSERT INTO Campaign_TarGets (CampaignID, CustomerID) VALUES  
253 ('CDD-25-7421', 'DD-CUST-25-0001'),  
254 ('CDD-25-3905', 'DD-CUST-25-0002'),  
255 ('CDD-25-8612', 'DD-CUST-25-0003'),  
256 ('CDD-25-2549', 'DD-CUST-25-0004'),  
257 ('CDD-25-1038', 'DD-CUST-25-0005'),  
258 ('CDD-25-9754', 'DD-CUST-25-0006'),  
259 ('CDD-25-6183', 'DD-CUST-25-0007'),  
260 ('CDD-25-4307', 'DD-CUST-25-0008'),  
261 ('CDD-25-5826', 'DD-CUST-25-0009'),  
262 ('CDD-25-9071', 'DD-CUST-25-0010');
```

6. Sample SQL Queries (10 Queries)

1. Retrieve products with a price between 2000 and 5000.

```
SELECT ProductId, ProductName, Price  
FROM Products  
WHERE PRICE BETWEEN 2000 AND 5000;
```

The screenshot shows the SQL Server Management Studio interface. The query window contains the following code:

```
310  
311 |> SELECT ProductId, ProductName, Price  
312 |> FROM Products  
313 |> WHERE PRICE BETWEEN 2000 AND 5000;
```

The results pane displays a table with the following data:

	ProductId	ProductName	Price
1	P1001	Velona Cuddles Baby Diaper	3230.00
2	P1002	Marvel Baby Diaper	3600.00
3	P1003	Adfit Adult	3150.00
4	P1006	Velona Cuddles Baby Diaper	3590.00
5	P1007	Manytime Adult Medium	2100.00
6	P1008	Carewell Adult Medium	2100.00
7	P1010	Pampers Medium	2600.00

2. Display the customers whose phone number does not start with '077' AND '75'.

```
SELECT * FROM Customer_phone  
WHERE PhoneNumber NOT LIKE '77%' AND PhoneNumber NOT LIKE '75%';
```

The screenshot shows the SQL Server Management Studio interface. The query window contains the following code:

```
309  
310 |> SELECT * FROM Customer_Phone  
311 |> WHERE PhoneNumber NOT LIKE '77%' AND PhoneNumber NOT LIKE '75%';
```

The results pane displays a table with the following data:

	CustomerId	PhoneNumber
1	DD-CUST-25-0003	717892407
2	DD-CUST-25-0005	767490820
3	DD-CUST-25-0006	760090746
4	DD-CUST-25-0009	728056547

3. Retrieve customers who are from Negombo and who had a delivery in October.

```
SELECT* FROM Delivery  
WHERE DeliveryAddress LIKE '%Negombo%' AND MONTH(DeliveryDate) =10;
```

The screenshot shows the SQL Server Management Studio interface. The code window contains the following query:

```
309  
310  SELECT* FROM Delivery  
311 WHERE DeliveryAddress LIKE '%Negombo%' AND MONTH(DeliveryDate) = 10;
```

The results pane displays the following table:

	DeliveryId	OrderId	DeliveryDate	DeliveryAddress
1	#122044	D14034	2025-10-19	No. 84/1, Beach Road, Ettukala, Negombo
2	#122048	D14038	2025-10-28	171/C3, Palangathure, Negombo

4. Find the highest spending customer.

```
SELECT c.CustomerId, c.fname, c.lname, o.TotalAmount  
FROM Customer c, Order o  
WHERE c.CustomerId=o.CustomerId  
AND o.TotalAmount=  
(Select MAX(TotalAmount) FROM Order);
```

The screenshot shows the SQL Server Management Studio interface. The code window contains the following query:

```
500  
309  
310  SELECT c.CustomerId, c.fname, c.lname, o.TotalAmount  
311  FROM Customer c, Orders o  
312  WHERE c.CustomerId=o.CustomerId  
313  AND o.TotalAmount=  
314  (Select MAX(TotalAmount) FROM Orders);  
315  
316  
317
```

The results pane displays the following table:

	CustomerId	fname	lname	TotalAmount
1	DD-CUST-25-0004	Amiru	Gunesekara	9690.00

5. Display each products name and the total quantity sold for that product, order it by most sold to least sold.

```
SELECT p.ProductName, SUM(od.Quantity) AS TotalSold  
FROM Order_Details od  
JOIN Products p ON od.ProductId = p.ProductId  
GROUP BY p.ProductName  
ORDER BY TotalSold DESC;
```

The screenshot shows a code editor with numbered lines 328 through 335. Line 329 contains a SELECT statement that joins Order_Details and Products tables, groups by ProductName, and orders by TotalSold in descending order. Below the code is a results grid showing the total sales for nine different products. The results grid has columns for ProductName and TotalSold, with data rows numbered 1 through 9.

	ProductName	TotalSold
1	Velona Cuddles Baby Diaper	5
2	Manytime Adult Diaper	4
3	Pampers Medium	3
4	Adfit Adult	2
5	Carewell Adult Medium	2
6	Friends Adult Small	2
7	Manytime Adult Medium	1
8	Marvel Baby Diaper	1
9	Velona Cuddles Baby Pants	1

6. Find the order with the highest priced product.

```
SELECT OrderId, ProductId, Quantity  
FROM Order_Details  
WHERE ProductId = (  
    SELECT ProductId  
    FROM Products  
    WHERE Price = (SELECT MAX(Price) FROM Products)  
);
```

The screenshot shows a code editor with numbered lines 339 through 348. Line 340 contains a SELECT statement that finds the OrderId, ProductId, and Quantity for the order with the highest price. It uses a subquery to determine the maximum price from the Products table and then filters the Order_Details table accordingly. Below the code is a results grid showing one row with OrderId D14032, ProductId P1005, and Quantity 1.

	OrderId	ProductId	Quantity
1	D14032	P1005	1

7. Display invoice ID, payment ID, and pay date for invoices with quantity ≥ 3 , sorted by the earliest payment date.

```
SELECT i.InvoiceId,p.PaymentId,p.PayDate  
FROM Invoice i, Payment p  
WHERE i.InvoiceId=p.InvoiceId  
AND i.Quantity>=3  
ORDER BY PayDate ASC;
```

The screenshot shows a SQL query window with the following content:

```
349    SELECT i.InvoiceId,p.PaymentId,p.PayDate  
350    FROM Invoice i, Payment p  
351    WHERE i.InvoiceId=p.InvoiceId  
352    AND i.Quantity>=3  
353    ORDER BY PayDate ASC;  
354
```

Below the query window is a results grid displaying three rows of data:

	Invoiceld	PaymentId	PayDate
1	INV007	PAY008	2025-09-26
2	INV008	PAY010	2025-12-04
3	INV004	PAY009	2025-12-20

8. Identify names of customers who have given a rating above 3.

```
SELECT c.Fname, c.Lname, f.Rating  
FROM Feedback_Rating f, Customer c  
WHERE c.CustomerId = f.CustomerId  
AND Rating>3;
```

The screenshot shows a SQL query window with the following content:

```
309    SELECT c.Fname, c.Lname, f.Rating  
310    FROM Feedback_Rating f, Customer c  
311    WHERE c.CustomerId = f.CustomerId  
312    AND Rating>3;  
313
```

Below the query window is a results grid displaying six rows of data:

	Fname	Lname	Rating
1	Supuni	Adikari	4
2	Tharini	Herath	5
3	Amiru	Gunesekara	4
4	Madusara	Anuhas	5
5	Rangi	Liyanage	4
6	Onel	Hettiarachchi	5

9. Display the orders sorted by order id excluding any null values if the average quantity is above 2.

```
SELECT OrderId, AVG(Quantity) AS Average_Quantity  
FROM Order_Details  
WHERE Quantity IS NOT NULL  
GROUP BY OrderId  
HAVING AVG(Quantity) > 2  
ORDER BY Orderid ASC;
```

The screenshot shows the SQL query in the top pane and its results in the bottom pane. The results table has columns 'OrderId' and 'Average_Quantity', with data rows for D14033 (3), D14036 (3), and D14037 (4).

OrderId	Average_Quantity
1	D14033 3
2	D14036 3
3	D14037 4

10. List down the customers with their total spending where total spending is less than the average spending.

```
SELECT C.Fname,C.Lname, O.TotalAmount  
FROM Customer c, Orders O  
WHERE C.CustomerId=O.CustomerId AND TotalAmount<  
(SELECT AVG(TotalAmount) FROM Orders);
```

The screenshot shows the SQL query in the top pane and its results in the bottom pane. The results table has columns 'Fname', 'Lname', and 'TotalAmount', with data rows for Tharini (Herath, 2100.00), Adithya (Subharathi, 3600.00), Onel (Hettiarachchi, 3600.00), and Hareesha (Jayarathna, 4200.00).

	Fname	Lname	TotalAmount
1	Tharini	Herath	2100.00
2	Adithya	Subharathi	3600.00
3	Onel	Hettiarachchi	3600.00
4	Hareesha	Jayarathna	4200.00