SECD 2523 Database

Project Phase 3

Getmehired.io Assist System << HireMe >>

LECTURER: Dr Izyan Izzati

SECTION: 02

GROUP NAME: Vergil's Chair

TEAM MEMBERS:

1. Navveen Nair A/L Manoharan A22EC0092
2. Suren A/L Chandra Sekaran A22EC0275
3. Taneshwer Sanggar A23CS5052
4. Lee Jian Li A22EC0064
5. Liew Shan Kai A21EA0051

Table of content

Content	Page
1.0 Introduction ……..……...…	2
2.0 Overview of the project ……..	3
3.0 Database conceptual design	
3.1 Updated business rule ……...……	4
3.2 Conceptual ERD ……	5
3.3 Enhanced ERD ……..	6
4.0 Database logical design	
4.1 Logical ERD …….	7
4.2 Updated data dictionary ……	8
4.3 Normalization ……	12
5.0 Relational DB schemas (after normalization) …….	15
6.0 SQL statements (DDL & DML) …….	17
7.0 Summary …….……...….	45

1.0 Introduction

In today's fast-paced world of career advancement, having a compelling and professionally crafted Curriculum Vitae (CV) is crucial for job seekers in the competitive job market. To address the challenges faced by individuals in creating impactful CVs, we present ‘GetMeHired.io’ an innovative online platform dedicated to simplifying and enhancing the CV creation process.

"GetMeHired.io" aims to redefine the CV-building experience with an intuitive, step-by-step process, making it easy for users to navigate through each section of their professional profile. Users have access to a collection of professionally designed CV templates that can be purchased. The platform allows customization, enabling individuals to tailor their CVs to showcase their unique skills, experiences, and career aspirations.

2.0 Overview of the project

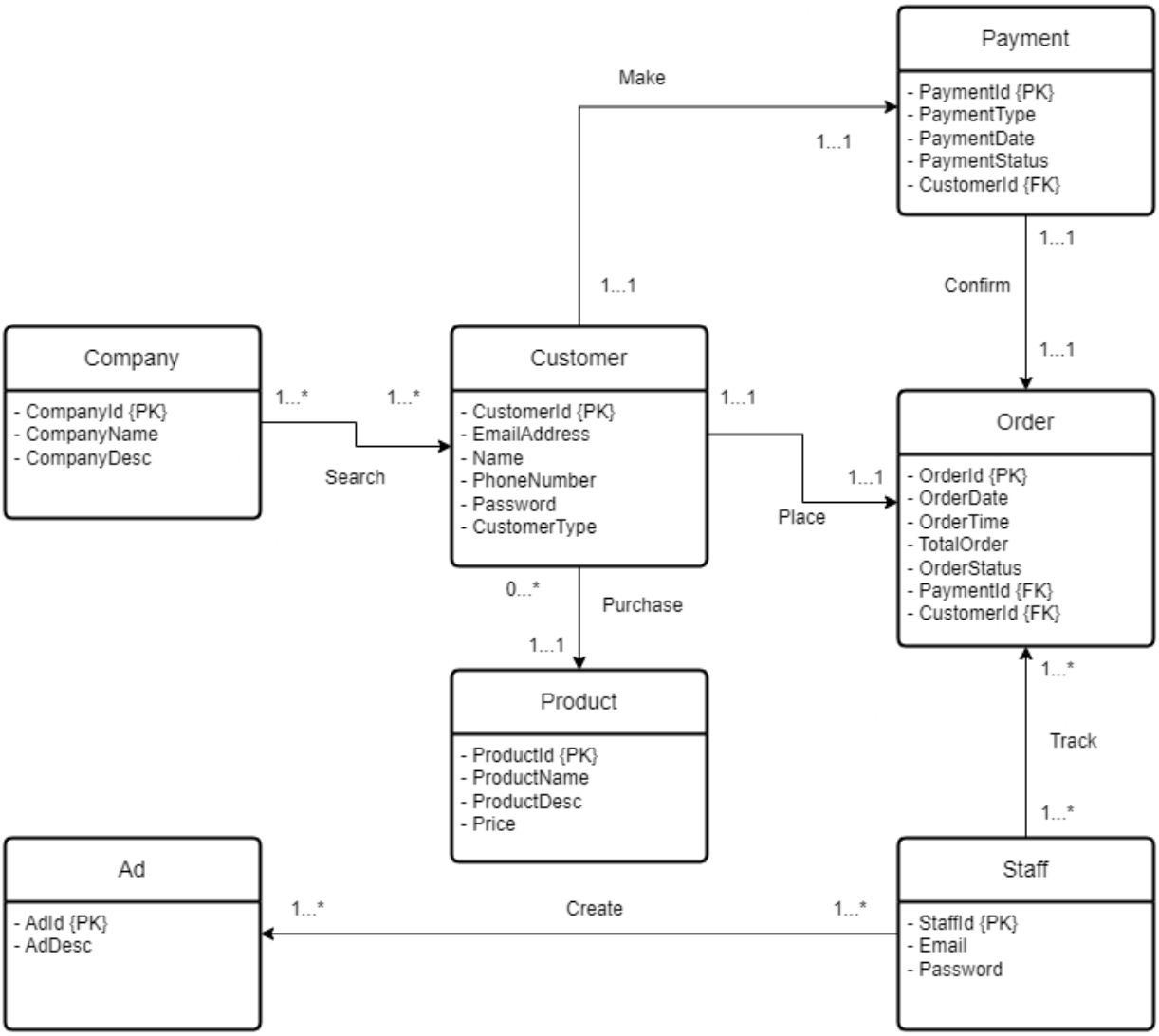
After completing phase 3 which is the database conceptual design phase, we have proceeded to phase 4 to implement the database logical design and produce SQL statements (both DDL and DML) for our GetMeHired.io website. In this report, we will transform the conceptual ERD in phase 3 into logical ERD followed by producing the relation schema. Normalization will also be done to minimize the duplicated data to enhance the data stored logically. After that, we will update the data dictionary for GetMeHired.io website by referring to the normalized relation. Lastly, the database for the website will be created using SQL.

3.0 Database conceptual design

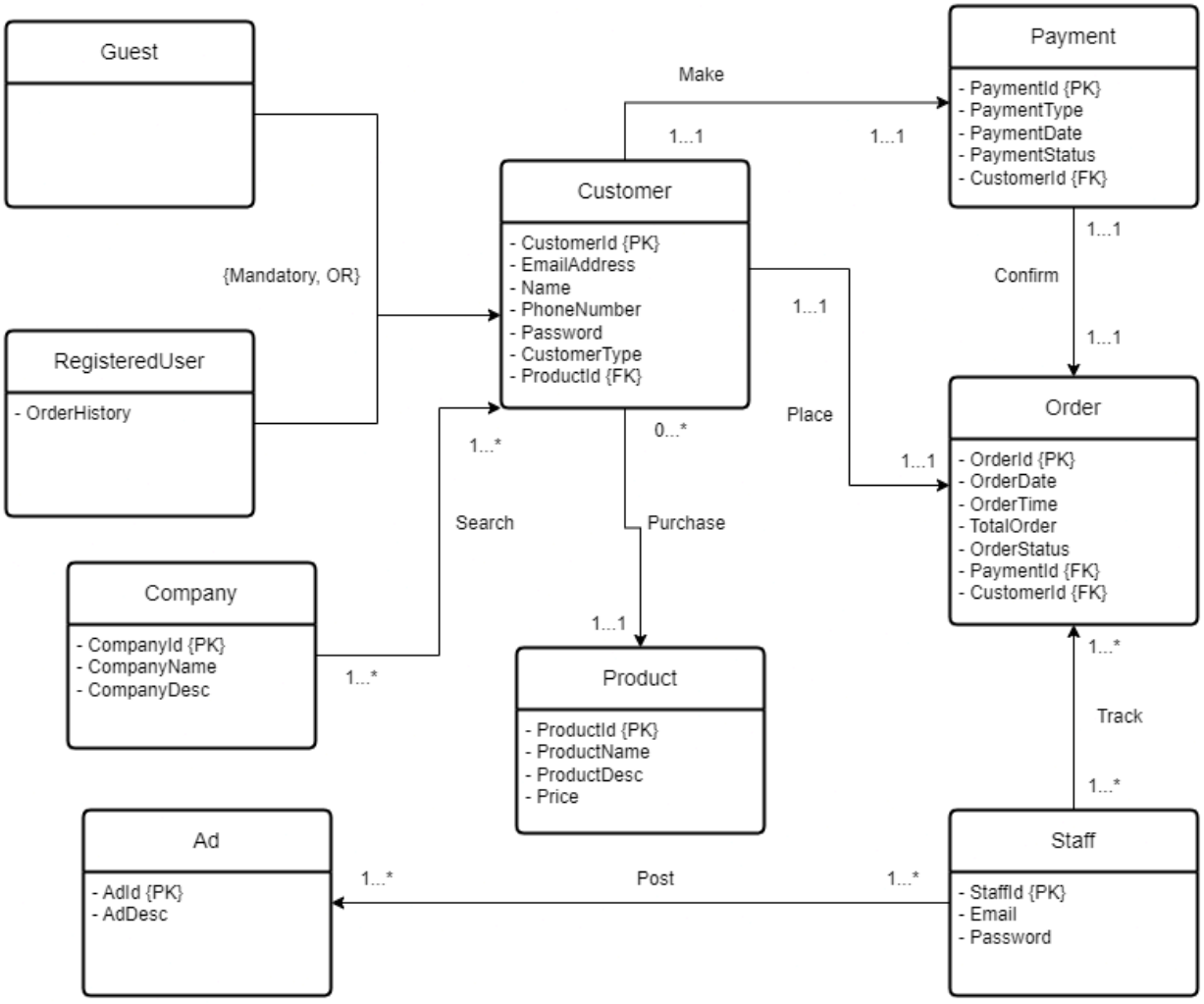
3.1 Updated business rule

1. The delivery system is working 24 hours a day, 7 days a week.
2. For the payment system, it will stop working from 12am-1am everyday due to the bank maintenance.
3. The payment status will be updated and sent to customers who have completed the purchase in less than 5 minutes via email delivery.
4. Customers are optional to register an account before purchasing a CV plan from the website.
5. Customers can preview each CV plan before placing an order.
6. For guests, customers must input their name and phone number in the system for checking.
7. Customers shall receive a final report of the CV from the system.
8. Staff must enter a Staff ID and password to access the system for tracking the orders. and email address for the delivery purposes upon checking out an order.
9. For registered users, customers can login into the account and perform the check out process directly without having to fill in the necessary details.
10. Customers must select a payment method from the available options to process the payment.
11. Customers shall receive the purchased CV templates via email.
12. Customers must fill the CV templates before submitting int

3.2 Conceptual ERD

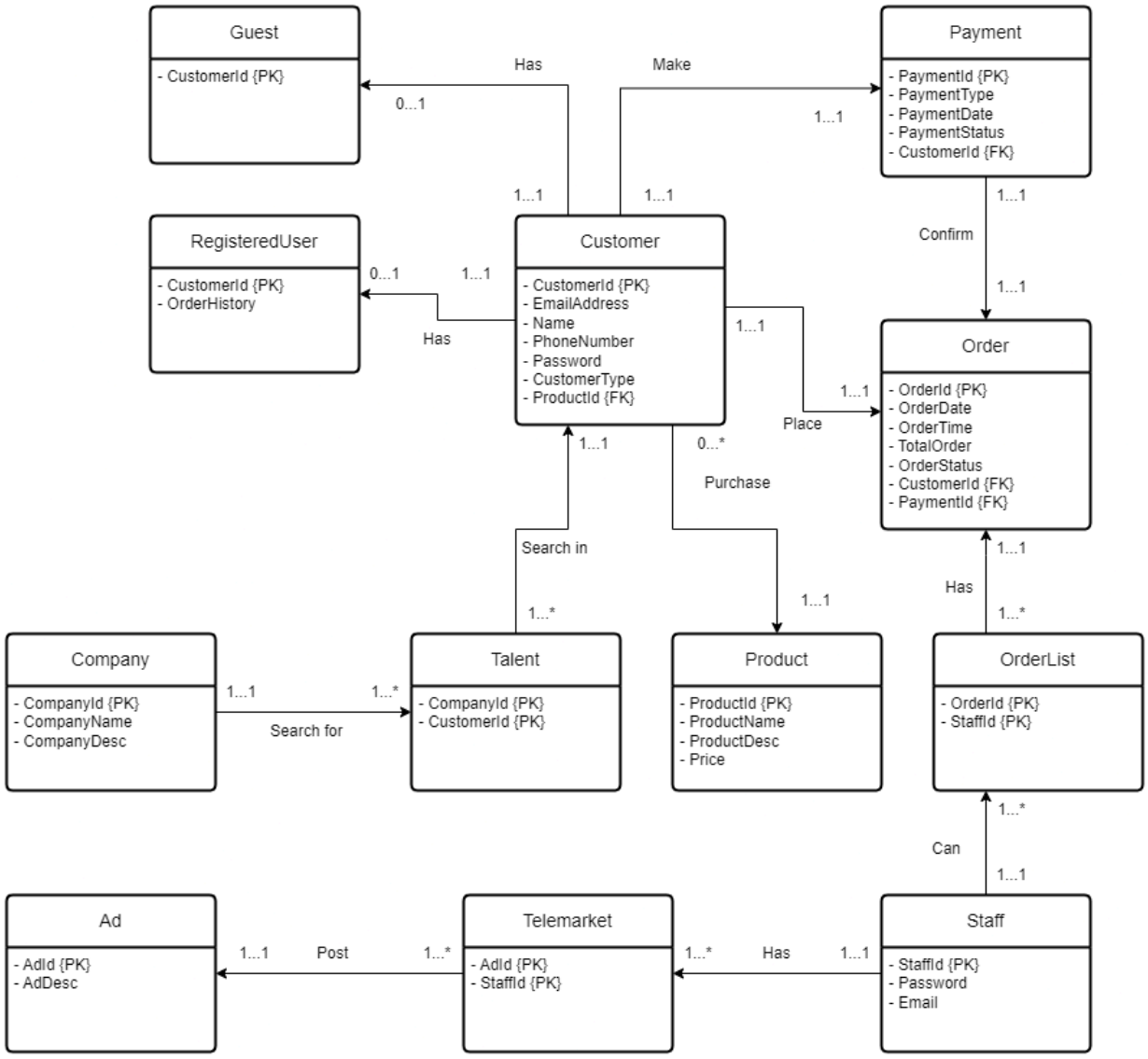


3.3 Enhanced ERD



4.0 Database logical design

4.1 Logical ERD



4.2 Updated data dictionary

4.2.1 Description of entities

Entity	Description	Occurrence
Customer	Holds the data of customer	Customer places order and makes payment for the order
Guest	Hold the data of new customer	Guest allows the new users to register themselves into the system
RegisteredUser	Hold the data of registered customer	Records order history of a customer.
Staff	Holds the data of staff	Staff monitors the sales and handles communication with potential customers
Ad	Hold the advertisement details	Staff create ads based on order.
Payment	Holds the payment details	Customer makes payment for their order
Order	Holds the order details	Customer places order for their desired product
Product	Holds the data about product	Customer can view details about the products
Company	Holds the data of the company	Company stores the details of companies that are available
Talent	Holds the customer talent detail	Company search for talented customers
Telemarket	Holds the staff's detail and advertisement ID that is going to be used once a customer makes a purchase for a specific product.	Telemarket stores advertisements to be shown for ads sessions after a purchase is done.
OrderList	Holds staff ID to give them the authority to check on order that is made by customer	Staff takes care of the order's that is undergoing processing to finalize an order

4.2.2 Description of relationship

Entity	Multiplicity	Relationship	Multiplicity	Entity
Customer	1…1	Place	1…1	Order
	1…1	Make	1…1	Payment
	0…*	Purchase	1…1	Product
	1…1	Has	0…1	Guest
	1…1	Has	0…1	RegisteredUser
Payment	1…1	Confirm	1…1	Order
Company	1…1	Search for	1…*	Talent
Talent	1…*	Search in	1…1	Customer
Staff	1…1	Can	1…*	OrderList
	1…1	Has	1…*	Telemarket
Telemarket	1…*	Post	1…1	Ad
Orderlist	1…*	Has	1…1	Order

4.2.3 Description of attributes

Entity	Attribute	Description	Data type	Constraint
Customer	CustomerId	Customer’s Id	VARCHAR2(15)	PRIMARY KEY
	Name	Customer’s name	VARCHAR2(30)	NOT NULL
	EmailAddress	Customer’s address	VARCHAR2(100)	NOT NULL
	PhoneNumber	Customer's	VARCHAR2(12)	NOT NULL

		phone number		
	Password	Login password	VARCHAR2(20)	NOT NULL
	CustomerType	Customer type	CHAR2(1)	NOT NULL
	ProductId	Customer purchased product ID	VARCHAR2(15)	FOREIGN KEY -REFERENCE PRODUCT
Guest	CustomerId	Customer's Id	VARCHAR2(15)	PRIMARY KEY
RegisteredUser	CustomerId	Customer's Id	VARCHAR2(15)	PRIMARY KEY
	OrderHistory	Order history of the customer	VARCHAR2(100)	NOT NULL
Company	CompanyId	Company's Id	VARCHAR2(15)	PRIMARY KEY
	CompanyName	Company Name	VARCHAR2(50)	NOT NULL
	CompanyDesc	Company description	VARCHAR2(200)	NOT NULL
Talent	CompanyId	Company's Id	VARCHAR2(15)	PRIMARY KEY FOREIGN KEY -REFERENCE COMPANY
	CustomerId	Customer's Id	VARCHAR2(15)	PRIMARY KEY FOREIGN KEY -REFERENCE CUSTOMER
Product	ProductId	Product's id	VARCHAR2(15)	PRIMARY KEY
	ProductName	Product's name	VARCHAR2(30)	NOT NULL
	ProductDesc	Product's	VARCHAR2(100)	NOT NULL

		description		
	Price	Product's price	DECIMAL(10,2)	NOT NULL
Payment	PaymentId	Payment Id	VARCHAR2(15)	PRIMARY KEY
	PaymentType	Payment method	CHAR(1)	NOT NULL
	PaymentDate	Date of payment	DATE	NOT NULL
	PaymentStatus	Payment status	CHAR(1)	NOT NULL
	CustomerId	Customer's Id	VARCHAR2(15)	FOREIGN KEY -REFERENCE CUSTOMER
Order	OrderId	Order's Id	VARCHAR2(15)	PRIMARY KEY
	OrderDate	Date of order	DATE	NOT NULL
	OrderTime	Time of order	TIMESTAMP	NOT NULL
	TotalOrder	Total price of order	DECIMAL(10,2)	NOT NULL
	OrderStatus	Order status	CHAR(1)	NOT NULL
	CustomerId	Customer's Id	VARCHAR2(15)	FOREIGN KEY -REFERENCE CUSTOMER
	PaymentId	Payment Id	VARCHAR2(15)	FOREIGN KEY -REFERENCE PAYMENT
Staff	StaffId	Staff's Id	VARCHAR2(15)	PRIMARY KEY
	Password	Staff's password	VARCHAR2(8)	NOT NULL
	Email	Staff's email	VARCHAR2(30)	NOT NULL
Ad	AdId	Advertisement Id	VARCHAR2(15)	PRIMARY KEY

	AdDesc	Advertisement description	VARCHAR2(100)	NOT NULL
Orderlist	OrderId	Order's ID	VARCHAR2(15)	PRIMARY KEY FOREIGN KEY -REFERENCE ORDER
	StaffId	Staff's ID	VARCHAR2(15)	PRIMARY KEY FOREIGN KEY -REFERENCE STAFF
Telemarket	Staffid	Staff's ID	VARCHAR2(15)	PRIMARY KEY FOREIGN KEY -REFERENCE STAFF
	AdId	Advertisement ID	VARCHAR2(15)	PRIMARY KEY FOREIGN KEY -REFERENCE AD

4.3 Normalization

1. Customer(CustomerId, EmailAddress, Name, PhoneNumber, Password, CustomerType, ProductId)

   1NF

   Customer(CustomerId, EmailAddress, Name, PhoneNumber, Password, CustomerType, ProductId)

   PK: CustomerId

   FK: ProductId reference Product(ProductId)

2. Guest(CustomerId)

   1NF

   Guest(CustomerId)

   PK: CustomerId

3. RegisteredUser(CustomerId, OrderHistory)

   1NF

   RegisteredUser(CustomerId, OrderHistory)

   PK: CustomerId

4. Company(CompanyId, CompanyName, CompanyDesc)

   1NF

   Company(CompanyId, CompanyName, CompanyDesc)

   PK: CompanyId

5. Talent(CompanyId, CustomerId)

   1NF

   Talent(CompanyId, CustomerId)

   PK: CompanyId, CustomerId

   FK: CompanyId reference Company(CompanyId)

   FK: CustomerId reference Customer(CustomerId)

6. Product(ProductId, ProductName, ProductDesc, Price)

   1NF

   Product(ProductId, ProductName, ProductDesc, Price)

   PK: ProductId

7. Payment(PaymentId, PaymentType, PaymentDate, PaymentStatus, CustomerId)

   1NF:

   Payment(PaymentId, PaymentType, PaymentDate, PaymentStatus, CustomerId)

   PK: PaymentId

FK: CustomerId reference Customer(CustomerId)

8. Order(OrderId, OrderDate, OrderTime, TotalOrder,OrderStatus, CustomerId, PaymentId)

   1NF:

   Order(OrderId, OrderDate, OrderTime, TotalOrder, OrderStatus, CustomerId, PaymentId)

   PK: OrderId

   FK: CustomerId reference Customer(CustomerId)

   PaymentId reference Payment(PaymentId)

9. Staff(StaffId, Password, Email)

   1NF

   Staff(StaffId, Email, Password)

   PK: StaffId, Email

10. Ad(AdId, AdDesc)

    1NF

    Ad(AdId, AdDesc)

    PK: AdId

11. Telemarket(StaffId, AdId)

    1NF

    Telemarket(StaffId, AdId)

    PK: StaffId, AdId

    FK: StaffId reference Staff(StaffId)

    FK: AdId reference Ad(AdId)

12. OrderList(OrderId, StaffId)

    1NF

    OrderList(OrderId, StaffId)

    PK: OrderId, StaffId

    FK: OrderId reference Order(OrderId)

FK: StaffId reference Staff(StaffId)

5.0 Relational DB schemas (after normalization)

These are the set of relation schemas in relational database schema

Customer(CustomerId, EmailAddress, Name, PhoneNumber, Password, CustomerType, ProductId)
PK: CustomerId
FK: ProductId reference Product(ProductId)

Guest(CustomerId)
PK: CustomerId

RegisteredUser(CustomerId, OrderHistory)
PK: CustomerId

Company(CompanyId, CompanyName, CompanyDesc)
PK: CompanyId

Talent(CompanyId, CustomerId)
PK: CompanyId, CustomerId
FK: CompanyId reference Company(CompanyId)
FK: CustomerId reference Customer(CustomerId)

Product(ProductId, ProductName, ProductDesc, Price)
PK: ProductId

Payment(PaymentId, PaymentType, PaymentDate, PaymentStatus, CustomerId)
PK: PaymentId
FK: CustomerId reference Customer(CustomerId)

Order(OrderId, OrderDate, OrderTime, TotalOrder, OrderStatus, CustomerId, PaymentId)

PK: OrderId

FK: CustomerId reference Customer(CustomerId)

PaymentId reference Payment(PaymentId)

Staff(StaffId, Email, Password)

PK: StaffId

Ad(AdId, AdDesc)

PK: AdId

Telemarket(StaffId, AdId)

PK: StaffId, AdId

FK: StaffId reference Staff(StaffId)

FK: AdId reference Ad(AdId)

OrderList(OrderId, StaffId)

PK: OrderId, StaffId

FK: OrderId reference Order(OrderId)

FK: StaffId reference Staff(StaffId)

6.0 SQL statements (DDL & DML)

```
-- Creating the PRODUCT table to store product information
CREATE TABLE PRODUCT(
   ProductID VARCHAR2(15) NOT NULL,
   ProductName VARCHAR2(30) NOT NULL,
   ProductDesc VARCHAR2(100) NOT NULL,
   Price DECIMAL(10,2) NOT NULL,
   CONSTRAINT ProductID_pk PRIMARY KEY (ProductID)
);

-- Creating the CUSTOMER table to store customer information
CREATE TABLE CUSTOMER(
   CustomerID VARCHAR2(15) NOT NULL,
   Name VARCHAR2(30) NOT NULL,
   PhoneNumber VARCHAR2(12) NOT NULL,
   EmailAddress VARCHAR2(100) NOT NULL,
   Password VARCHAR2(20) NOT NULL,
   CustomerType CHAR(1) NOT NULL,
   ProductID VARCHAR2(15) NOT NULL,
   CONSTRAINT ProductID_fk FOREIGN KEY(ProductID) REFERENCES
PRODUCT(ProductID),
   CONSTRAINT CustomerID_PK PRIMARY KEY (CustomerID)
);

-- Creating the PAYMENT table to store payment information
CREATE TABLE PAYMENT(
   PaymentID VARCHAR2(15) NOT NULL,
   PaymentType CHAR(1) NOT NULL,
   PaymentDate DATE NOT NULL,
   PaymentStatus CHAR(1) NOT NULL,
   CustomerID VARCHAR2(15) NOT NULL,
```

```
    CONSTRAINT PaymentID_pk PRIMARY KEY (PaymentID)
);

-- Creating the ORDERS table to store order information
CREATE TABLE ORDERS(
    OrderID VARCHAR2(15) NOT NULL,
    OrderDate DATE NOT NULL,
    OrderTime TIMESTAMP NOT NULL,
    OrderStatus CHAR(1) NOT NULL,
    CustomerID VARCHAR2(15) NOT NULL,
    PaymentID VARCHAR2(15) NOT NULL,
    CONSTRAINT OrderID_pk PRIMARY KEY(OrderID),
    CONSTRAINT CustomerID_5_fk FOREIGN KEY(CustomerID) REFERENCES
CUSTOMER(CustomerID),
    CONSTRAINT PaymentID_1_fk FOREIGN KEY(PaymentID) REFERENCES
PAYMENT(PaymentID)
);

-- Creating the STAFF table to store staff information
CREATE TABLE STAFF(
    StaffID VARCHAR2(15) NOT NULL,
    Password VARCHAR2(8) NOT NULL,
    Email VARCHAR2(30) NOT NULL,
    CONSTRAINT Staff_PK PRIMARY KEY (StaffID),
    CONSTRAINT Email_fk FOREIGN KEY(Email) REFERENCES STAFFLOGIN(Email)
);

-- Creating the COMPANY table to store company information
CREATE TABLE COMPANY(
    CompanyID VARCHAR2(15) NOT NULL,
    CompanyName VARCHAR2(50) NOT NULL,
```

```
    CompanyDesc VARCHAR2(200) NOT NULL,
    CONSTRAINT CompanyID_PK PRIMARY KEY (CompanyID)
);

-- Creating the GUEST table to store guest information
CREATE TABLE GUEST(
    CustomerID VARCHAR2(15) NOT NULL,
    CONSTRAINT CustomerID_6_PK PRIMARY KEY (CustomerID)
);

-- Creating the RegisteredUser table to store registered user information
CREATE TABLE RegisteredUser(
    CustomerID VARCHAR2(15) NOT NULL,
    OrderHistory VARCHAR2(100) NOT NULL,
    CONSTRAINT CustomerID_7_PK PRIMARY KEY (CustomerID)
);

-- Creating the AD table to store advertisement information
CREATE TABLE AD(
    AdID VARCHAR2(15) NOT NULL,
    AdDesc VARCHAR2(200) NOT NULL,
    CONSTRAINT AdID_1_PK PRIMARY KEY (AdID)
);

-- Creating the TALENT table to store talent information
CREATE TABLE TALENT(
    CompanyID VARCHAR2(15) NOT NULL,
    CustomerID VARCHAR2(15) NOT NULL,
    CONSTRAINT Talent_1_PK PRIMARY KEY (CustomerID,CompanyID),
    CONSTRAINT CompanyID_1_fk FOREIGN KEY(CompanyID) REFERENCES
COMPANY(CompanyID),
```

```
   CONSTRAINT CustomerID_8_fk FOREIGN KEY(CustomerID) REFERENCES
CUSTOMER(CustomerID)
);

-- Creating the TELEMARKET table
CREATE TABLE TELEMARKET(
   AdID VARCHAR2(15) NOT NULL,
   StaffID VARCHAR2(15) NOT NULL,
   CONSTRAINT Tlmrktpk_1_PK PRIMARY KEY (AdID, StaffID),
   CONSTRAINT AdID_2_FK FOREIGN KEY (AdID) REFERENCES AD(AdID),
   CONSTRAINT StaffID_3_FK FOREIGN KEY (StaffID) REFERENCES STAFF(StaffID)
);

-- Creating the ORDERLIST table
CREATE TABLE ORDERLIST(
   OrderID VARCHAR2(15) NOT NULL,
   StaffID VARCHAR2(15) NOT NULL,
   CONSTRAINT OrderListpk_1_PK PRIMARY KEY (OrderID, StaffID),
   CONSTRAINT OrderID_2_FK FOREIGN KEY (OrderID) REFERENCES
ORDERS(OrderID),
   CONSTRAINT StaffID_4_FK FOREIGN KEY (StaffID) REFERENCES STAFF(StaffID)
);

-- Add foreign key constraint to Payment Table
ALTER TABLE PAYMENT
ADD CONSTRAINT CustomerID_9_fk FOREIGN KEY(CustomerID) REFERENCES
CUSTOMER(CustomerID)

-- Inserting product data
INSERT INTO PRODUCT
VALUES('P01','CV Templates','2 ATS CV Templates with example, 3 Direct Email CV
```

```
templates-PPT, 3 Direct Email CV templates-Canva',
        99);
INSERT INTO PRODUCT
VALUES('P02','CV Writing Basic','1 Full Writing Top-Notch ATS CV, Powerful action verbs on
CV, Duration 5-10 days',
        159);
INSERT INTO PRODUCT
VALUES('P03','CV Writing Pro','1 Full Writing Top-Notch ATS CV, Full Access Library,
Duration 1-4 days',
        199);
INSERT INTO PRODUCT
VALUES('P04','Webinar','Cover 4 Modules, Certification of Completion, Full Access Member
Area', 250);

-- Inserting customer data
INSERT INTO CUSTOMER
VALUES('ID001','SUREN','0189884113','suren@example.com','password1234567890','r','P01');
INSERT INTO CUSTOMER
VALUES('ID002','ALI','0195667345','ali@example.com','pass9876543210','g','P02');
INSERT INTO CUSTOMER
VALUES('ID003','ABU','0184567890','abu@example.com','securepassword','r','P03');
INSERT INTO CUSTOMER
VALUES('ID004','LISA','0165432344','lisa@example.com','strongpass','g','P04');
INSERT INTO CUSTOMER
VALUES('ID005','DEPP','0156790089','depp@example.com','myp@ssword','r','P01');
INSERT INTO CUSTOMER
VALUES('ID006','JOSEPH','0167899999','joseph@example.com','secretpass','g','P02');
INSERT INTO CUSTOMER
VALUES('ID007','JONATHAN','0165553212','jonathan@example.com','pass123','r','P03');
INSERT INTO CUSTOMER
VALUES('ID008','JACK','0178909345','jack@example.com','randompass','g','P04');
```

```
INSERT INTO CUSTOMER
VALUES('ID009','MAX','0123421999','max@example.com','p@ssword','r','P01');
INSERT INTO CUSTOMER
VALUES('ID010','RAJU','0189878654','raju@example.com','securepass123','g','P02');
INSERT INTO CUSTOMER
VALUES('ID011','CHONG','0167456789','chong@example.com','pass987','r','P03');
INSERT INTO CUSTOMER
VALUES('ID012','LEE','0107896532','lee@example.com','newpassword','g','P04');
INSERT INTO CUSTOMER
VALUES('ID013','AMAR','0176452112','amar@example.com','qwerty123','r','P01');
INSERT INTO CUSTOMER
VALUES('ID014','AQILAH','0153213345','aqilah@example.com','testpass','g','P02');
INSERT INTO CUSTOMER
VALUES('ID015','ABHI','0134567856','abhi@example.com','passwordtest','r','P03');
INSERT INTO CUSTOMER
VALUES('ID016','NAVVEEN','0134123498','navveen@example.com','examplepass','g','P04');
INSERT INTO CUSTOMER
VALUES('ID017','SHAN','0134568909','shan@example.com','pass1234','r','P01');
INSERT INTO CUSTOMER
VALUES('ID018','LIEW','0178900764','liew@example.com','mypass123','g','P02');
INSERT INTO CUSTOMER
VALUES('ID019','KUGHAN','0145608765','kughan@example.com','pass@word','r','P03');
INSERT INTO CUSTOMER
VALUES('ID020','NAVIN','0145678909','navin@example.com','p@ss123word','g','P04');

-- Inserting payment data
INSERT INTO PAYMENT
VALUES ('P1111','A',SYSDATE,'0','ID001');
INSERT INTO PAYMENT
VALUES ('P1112','B',SYSDATE,'1','ID002');
INSERT INTO PAYMENT
```

```
VALUES ('P1113','B',SYSDATE,'1','ID003');
INSERT INTO PAYMENT
VALUES ('P1114','A',SYSDATE,'1','ID004');
INSERT INTO PAYMENT
VALUES ('P1115','A',SYSDATE,'1','ID005');
INSERT INTO PAYMENT
VALUES ('P1116','B',SYSDATE,'1','ID006');
INSERT INTO PAYMENT
VALUES ('P1117','A',SYSDATE,'0','ID007');
INSERT INTO PAYMENT
VALUES ('P1118','B',SYSDATE,'1','ID008');
INSERT INTO PAYMENT
VALUES ('P1119','A',SYSDATE,'1','ID009');
INSERT INTO PAYMENT
VALUES ('P1120','B',SYSDATE,'0','ID010');
INSERT INTO PAYMENT
VALUES ('P1121','B',SYSDATE,'1','ID011');
INSERT INTO PAYMENT
VALUES ('P1122','B',SYSDATE,'1','ID012');
INSERT INTO PAYMENT
VALUES ('P1123','B',SYSDATE,'0','ID013');
INSERT INTO PAYMENT
VALUES ('P1124','A',SYSDATE,'0','ID014');
INSERT INTO PAYMENT
VALUES ('P1125','A',SYSDATE,'1','ID015');
INSERT INTO PAYMENT
VALUES ('P1126','A',SYSDATE,'0','ID016');
INSERT INTO PAYMENT
VALUES ('P1127','B',SYSDATE,'1','ID017');
INSERT INTO PAYMENT
VALUES ('P1128','A',SYSDATE,'0','ID018');
```

```
INSERT INTO PAYMENT
VALUES ('P1129','A',SYSDATE,'0','ID019');
INSERT INTO PAYMENT
VALUES ('P1130','B',SYSDATE,'1','ID020');

INSERT INTO ORDERS
VALUES
('ORDER01',SYSDATE,TO_TIMESTAMP('12:24:36','HH24:MI:SS'),'S','ID001','P1111');
INSERT INTO ORDERS
VALUES ('ORDER02',SYSDATE,CURRENT_TIMESTAMP,'S','ID002','P1112');
INSERT INTO ORDERS
VALUES
('ORDER03',SYSDATE,TO_TIMESTAMP('13:45:01','HH24:MI:SS'),'U','ID003','P1113');
INSERT INTO ORDERS
VALUES ('ORDER04',SYSDATE,CURRENT_TIMESTAMP,'S','ID004','P1114');
INSERT INTO ORDERS
VALUES
('ORDER05',SYSDATE,TO_TIMESTAMP('16:01:10','HH24:MI:SS'),'U','ID005','P1115');
INSERT INTO ORDERS
VALUES
('ORDER06',SYSDATE,TO_TIMESTAMP('18:37:50','HH24:MI:SS'),'S','ID006','P1116');
INSERT INTO ORDERS
VALUES
('ORDER07',SYSDATE,TO_TIMESTAMP('12:50:21','HH24:MI:SS'),'S','ID007','P1117');
INSERT INTO ORDERS
VALUES
('ORDER08',SYSDATE,TO_TIMESTAMP('08:05:31','HH24:MI:SS'),'U','ID008','P1118');
INSERT INTO ORDERS
VALUES
('ORDER09',SYSDATE,TO_TIMESTAMP('04:51:59','HH24:MI:SS'),'S','ID009','P1119');
INSERT INTO ORDERS
```

```
VALUES
('ORDER10',SYSDATE,TO_TIMESTAMP('10:10:10','HH24:MI:SS'),'S','ID010','P1120');
INSERT INTO ORDERS
VALUES ('ORDER11',SYSDATE,CURRENT_TIMESTAMP,'S','ID011','P1121');
INSERT INTO ORDERS
VALUES
('ORDER12',SYSDATE,TO_TIMESTAMP('13:28:30','HH24:MI:SS'),'S','ID012','P1122');
INSERT INTO ORDERS
VALUES
('ORDER13',SYSDATE,TO_TIMESTAMP('11:42:59','HH24:MI:SS'),'S','ID013','P1123');
INSERT INTO ORDERS
VALUES
('ORDER14',SYSDATE,TO_TIMESTAMP('09:55:35','HH24:MI:SS'),'S','ID014','P1124');
INSERT INTO ORDERS
VALUES
('ORDER15',SYSDATE,TO_TIMESTAMP('12:24:45','HH24:MI:SS'),'S','ID015','P1125');
INSERT INTO ORDERS
VALUES
('ORDER16',SYSDATE,TO_TIMESTAMP('20:22:24','HH24:MI:SS'),'S','ID016','P1126');
INSERT INTO ORDERS
VALUES
('ORDER17',SYSDATE,TO_TIMESTAMP('23:00:02','HH24:MI:SS'),'S','ID017','P1127');
INSERT INTO ORDERS
VALUES ('ORDER18',SYSDATE,CURRENT_TIMESTAMP,'S','ID018','P1128');
INSERT INTO ORDERS
VALUES
('ORDER19',SYSDATE,TO_TIMESTAMP('23:30:54','HH24:MI:SS'),'S','ID019','P1129');
INSERT INTO ORDERS
VALUES
('ORDER20',SYSDATE,TO_TIMESTAMP('09:42:50','HH24:MI:SS'),'S','ID020','P1130');
```

```
-- Inserting company data
INSERT INTO COMPANY (CompanyID, CompanyName, CompanyDesc)
VALUES ('C1', 'Tech Solutions Inc.', 'A leading technology company specializing in software development and IT solutions.');
INSERT INTO COMPANY (CompanyID, CompanyName, CompanyDesc)
VALUES ('C2', 'Global Conglomerate Corp.', 'A global conglomerate with diverse business interests in finance, manufacturing, and technology.');
INSERT INTO COMPANY (CompanyID, CompanyName, CompanyDesc)
VALUES ('C3', 'International Logistics Ltd.', 'An international logistics company providing comprehensive supply chain solutions.');
INSERT INTO COMPANY (CompanyID, CompanyName, CompanyDesc)
VALUES ('C4', 'Health Innovations Co.', 'A healthcare organization committed to advancing medical research and patient care.');
INSERT INTO COMPANY (CompanyID, CompanyName, CompanyDesc)
VALUES ('C5', 'Industrial Manufacturing Industries', 'A manufacturing giant known for producing high-quality industrial equipment and machinery.');
INSERT INTO COMPANY (CompanyID, CompanyName, CompanyDesc)
VALUES ('C6', 'Diversified Enterprises Group', 'A business conglomerate with interests in real estate, hospitality, and entertainment.');
INSERT INTO COMPANY (CompanyID, CompanyName, CompanyDesc)
VALUES ('C7', 'Dynamic Marketing Agency', 'A dynamic marketing agency specializing in digital marketing and brand strategy.');
INSERT INTO COMPANY (CompanyID, CompanyName, CompanyDesc)
VALUES ('C8', 'Innovative Solutions Corp.', 'An innovative technology solutions provider offering cutting-edge software and services.');
INSERT INTO COMPANY (CompanyID, CompanyName, CompanyDesc)
VALUES ('C9', 'Research and Development Innovations', 'A research and development company focused on creating groundbreaking innovations in various industries.');
INSERT INTO COMPANY (CompanyID, CompanyName, CompanyDesc)
VALUES ('C10', 'Financial Services Ltd.', 'A financial institution providing a wide range of
```

banking and financial services.');
INSERT INTO COMPANY (CompanyID, CompanyName, CompanyDesc)
VALUES ('C11', 'Venture Capital Ventures', 'A venture capital firm investing in promising startups and emerging technologies.');
INSERT INTO COMPANY (CompanyID, CompanyName, CompanyDesc)
VALUES ('C12', 'Diversified Business Group', 'A diversified business group with interests in energy, telecommunications, and construction.');
INSERT INTO COMPANY (CompanyID, CompanyName, CompanyDesc)
VALUES ('C13', 'Advanced Technologies Ltd.', 'A technology company at the forefront of developing advanced hardware and software solutions.');
INSERT INTO COMPANY (CompanyID, CompanyName, CompanyDesc)
VALUES ('C14', 'Consumer Goods Co.', 'A leading consumer goods company producing popular brands in food and household products.');
INSERT INTO COMPANY (CompanyID, CompanyName, CompanyDesc)
VALUES ('C15', 'Systems WIntegration Systems', 'A systems integration company specializing in designing and implementing robust IT infrastructure.');
INSERT INTO COMPANY (CompanyID, CompanyName, CompanyDesc)
VALUES ('C16', 'Innovation Hub Innovate', 'An innovation hub fostering creativity and collaboration among entrepreneurs and inventors.');
INSERT INTO COMPANY (CompanyID, CompanyName, CompanyDesc)
VALUES ('C17', 'Strategic Solutions Consulting', 'A consulting firm offering strategic business solutions and management consulting services.');
INSERT INTO COMPANY (CompanyID, CompanyName, CompanyDesc)
VALUES ('C18', 'Diversified Enterprises Enterprises', 'An enterprise with diverse business ventures, including retail, manufacturing, and services.');
INSERT INTO COMPANY (CompanyID, CompanyName, CompanyDesc)
VALUES ('C19', 'Industrial Conglomerate Industries', 'An industrial conglomerate with operations in manufacturing, construction, and engineering.');
INSERT INTO COMPANY (CompanyID, CompanyName, CompanyDesc)
VALUES('C20', 'Multinational Group Group', 'A multinational group with a wide range of subsidiaries operating in various industries .');

```
-- Inserting guest data
INSERT INTO GUEST
VALUES ('ID002');
INSERT INTO GUEST
VALUES ('ID003');
INSERT INTO GUEST
VALUES ('ID004');
INSERT INTO GUEST
VALUES ('ID005');
INSERT INTO GUEST
VALUES ('ID006');
INSERT INTO GUEST
VALUES ('ID007');
INSERT INTO GUEST
VALUES ('ID008');
INSERT INTO GUEST
VALUES ('ID009');
INSERT INTO GUEST
VALUES ('ID010');
INSERT INTO GUEST
VALUES ('ID011');
INSERT INTO GUEST
VALUES ('ID012');

-- Inserting registered user data
INSERT INTO RegisteredUser
VALUES ('ID001','CV Templates');
INSERT INTO RegisteredUser
VALUES ('ID002','CV Writing Basic');
INSERT INTO RegisteredUser
```

```
VALUES ('ID013','CV Writing Pro');
INSERT INTO RegisteredUser
VALUES ('ID014','CV Writing Pro');
INSERT INTO RegisteredUser
VALUES ('ID015','CV Template');
INSERT INTO RegisteredUser
VALUES ('ID016','CV Writing Basic');
INSERT INTO RegisteredUser
VALUES ('ID017','CV Template');
INSERT INTO RegisteredUser
VALUES ('ID018','CV Writing Pro');
INSERT INTO RegisteredUser
VALUES ('ID019','Webinar');
INSERT INTO RegisteredUser
VALUES ('ID020','Webinar');

-- Inserting advertisement data
INSERT INTO AD
VALUES('AD1','Nike Air Max. Elevate your run with Nike Air Max, featuring unparalleled comfort and style. Reach new heights in the ultimate running experience.');
INSERT INTO AD
VALUES('AD2','Apple IPhone 15. Unleash the power of innovation with the Apple iPhone 13. Cutting-edge technology meets sleek design for an unparalleled mobile experience.');
INSERT INTO AD
VALUES('AD3','Coca-Cola Zero. Enjoy the same great taste with zero sugar! Refresh yourself with the crisp and satisfying flavor of Coca-Cola Zero. Indulge guilt-free!');
INSERT INTO AD
VALUES('AD4','Samsung QLED TV. Immerse yourself in the brilliance of Samsung QLED TV. Stunning visuals, cinematic colors, and smart features redefine your entertainment experience.');
INSERT INTO AD
VALUES('AD5','Starbucks Pumpkin Spice Latte. Fall in love with the season! Starbucks
```

```
Pumpkin Spice Latte – a cozy blend of espresso, steamed milk, and pumpkin spice goodness.');
INSERT INTO AD
VALUES('AD6','Adidas Ultraboost Sneakers. Step into comfort and style with Adidas Ultraboost Sneakers. Responsive cushioning and a sleek design for an active lifestyle.');
INSERT INTO AD
VALUES('AD7','Tesla Model 3. Drive into the future with the Tesla Model 3. Sustainable, high-performance electric driving redefined. Experience the thrill of zero-emission travel.');
INSERT INTO AD
VALUES('AD8', 'Dyson V11 Vacuum. Revolutionize your cleaning routine with the Dyson V11 Vacuum. Unrivaled suction power and intelligent technology for a spotless home');
INSERT INTO AD
VALUES('AD9','Fitbit Charge 5. Elevate your fitness journey with Fitbit Charge 5. Advanced health tracking, built-in GPS, and stylish design – your path to wellness.');
INSERT INTO AD
VALUES('AD10','Amazon Echo Dot. Meet the compact and powerful Amazon Echo Dot. Your voice-controlled assistant for smart living. Play music, control devices, and much more!');

-- Inserting talent data
INSERT INTO TALENT (CompanyID, CustomerID)
VALUES('C1','ID001');
INSERT INTO TALENT (CompanyID, CustomerID)
VALUES('C2','ID005');
INSERT INTO TALENT (CompanyID, CustomerID)
VALUES('C3','ID003');
INSERT INTO TALENT (CompanyID, CustomerID)
VALUES('C4','ID002');
INSERT INTO TALENT (CompanyID, CustomerID)
VALUES('C5','ID010');
INSERT INTO TALENT (CompanyID, CustomerID)
VALUES('C6','ID009');
INSERT INTO TALENT (CompanyID, CustomerID)
```

```
VALUES('C7','ID008');
INSERT INTO TALENT (CompanyID, CustomerID)
VALUES('C8','ID007');
INSERT INTO TALENT (CompanyID, CustomerID)
VALUES('C9','ID011');
INSERT INTO TALENT (CompanyID, CustomerID)
VALUES('C10','ID013');

-- Inserting staff data
INSERT INTO STAFF
VALUES('S01','12345678','Ali@graduate.utm.my');
INSERT INTO STAFF
VALUES('S02','01234567','Kuga@graduate.utm.my');
INSERT INTO STAFF
VALUES('S03','abu12345','Abu@graduate.utm.my');
INSERT INTO STAFF
VALUES('S04','alif8978','Alif@graduate.utm.my');
INSERT INTO STAFF
VALUES('S05','lee12345','Lee@graduate.utm.my');
INSERT INTO STAFF
VALUES('S06','87654321','David@graduate.utm.my');
INSERT INTO STAFF
VALUES('S07','JD123456','Jack@graduate.utm.my');
INSERT INTO STAFF
VALUES('S08','Leo01234','Leo@graduate.utm.my');
INSERT INTO STAFF
VALUES('S09','yong5431','Yong@graduate.utm.my');
INSERT INTO STAFF
VALUES('S10','spidermn','Peter@graduate.utm.my');

-- Inserting data into TELEMARKET
```

```
INSERT INTO TELEMARKET
VALUES('AD1', 'S02');
INSERT INTO TELEMARKET
VALUES('AD2', 'S04');
INSERT INTO TELEMARKET
 VALUES('AD3', 'S06');
INSERT INTO TELEMARKET
 VALUES('AD4', 'S08');
INSERT INTO TELEMARKET
VALUES('AD5', 'S10');
INSERT INTO TELEMARKET
 VALUES('AD6', 'S01');
INSERT INTO TELEMARKET
 VALUES('AD7', 'S03');
INSERT INTO TELEMARKET
VALUES('AD8', 'S05');
INSERT INTO TELEMARKET
VALUES('AD9', 'S07');
INSERT INTO TELEMARKET
VALUES('AD10', 'S09');

-- Inserting data into ORDERLIST
INSERT INTO ORDERLIST
VALUES('ORDER01', 'S02');
INSERT INTO ORDERLIST
VALUES('ORDER02', 'S04');
INSERT INTO ORDERLIST
VALUES('ORDER03', 'S06');
INSERT INTO ORDERLIST
VALUES('ORDER04', 'S08');
INSERT INTO ORDERLIST
```

```
VALUES('ORDER05', 'S10');
INSERT INTO ORDERLIST
VALUES('ORDER06', 'S01');
INSERT INTO ORDERLIST
VALUES('ORDER07', 'S03');
INSERT INTO ORDERLIST
VALUES('ORDER08', 'S05');
INSERT INTO ORDERLIST
VALUES('ORDER09', 'S07');
INSERT INTO ORDERLIST
VALUES('ORDER10', 'S09');
INSERT INTO ORDERLIST
VALUES('ORDER11', 'S02');
INSERT INTO ORDERLIST
VALUES('ORDER12', 'S04');
INSERT INTO ORDERLIST
VALUES('ORDER13', 'S06');
INSERT INTO ORDERLIST
VALUES('ORDER14', 'S08');
INSERT INTO ORDERLIST
VALUES('ORDER15', 'S10');
INSERT INTO ORDERLIST
VALUES('ORDER16', 'S01');
INSERT INTO ORDERLIST
VALUES('ORDER17', 'S03');
INSERT INTO ORDERLIST
VALUES('ORDER18', 'S05');
INSERT INTO ORDERLIST
VALUES('ORDER19', 'S07');
INSERT INTO ORDERLIST
VALUES('ORDER20', 'S09');
```

```
-- Selecting all customer data
SELECT *
FROM CUSTOMER

-- Updating customer phone number
UPDATE CUSTOMER
SET PhoneNumber='0167458271'
WHERE CustomerID ='ID001'

-- Selecting updated customer data
SELECT *
FROM CUSTOMER

-- Selecting all order data
SELECT *
FROM ORDERLIST

DELETE FROM ORDERLIST WHERE OrderID='ORDER20'

SELECT *
FROM ORDERLIST

-- Selecting specific customer data
SELECT CustomerID, Name , PhoneNumber , EmailAddress
FROM CUSTOMER
WHERE CustomerID LIKE 'ID004';

-- Selecting order data based on a list of payment IDs
SELECT OrderID,OrderDate,OrderTime ,OrderStatus,CustomerID,PaymentID
FROM ORDERS
```

```
WHERE PaymentID IN
('P1111','P1112','P1113','P1114','P1115','P1116','P1117','P1118','P1119','P1120');

SELECT *
FROM CUSTOMER NATURAL JOIN TALENT

SELECT c.CustomerID, Name, CustomerType, g.OrderHistory
FROM CUSTOMER c JOIN REGISTEREDUSER g
ON c.CustomerID=g.CustomerID

--Three-Way Joins with the ON Clause
SELECT o.OrderID, o.OrderDate, c.CustomerID, c.Name AS CustomerName, p.ProductName,
p.Price
FROM ORDERS o JOIN CUSTOMER c
ON o.CustomerID = c.CustomerID
JOIN PRODUCT p
ON c.ProductID = p.ProductID;


--Left outer join
SELECT c.CustomerID, c.Name AS CustomerName, p.PaymentID, p.PaymentType,
p.PaymentDate
FROM CUSTOMER c
LEFT OUTER JOIN PAYMENT p
ON c.CustomerID = p.CustomerID;
```

CUSTOMERID	NAME	PHONENUMBER	EMAILADDRESS	PASSWORD	CUSTOMERTYPE	PRODUCTID
ID001	SUREN	0189884113	suren@example.com	password1234567890	r	P01
ID002	ALI	0195667345	ali@example.com	pass9876543210	g	P02
ID003	ABU	0184567890	abu@example.com	securepassword	g	P03
ID004	LISA	0165432344	lisa@example.com	strongpass	g	P04
ID005	DEPP	0156790089	depp@example.com	myp@ssword	g	P01
ID006	JOSEPH	0167899999	joseph@example.com	secretpass	g	P02
ID007	JONATHAN	0165553212	jonathan@example.com	pass123	g	P03
ID008	JACK	0178909345	jack@example.com	randompass	g	P04
ID009	MAX	0123421999	max@example.com	p@ssword	g	P01
ID010	RAJU	0189878654	raju@example.com	securepass123	g	P02
ID011	CHONG	0167456789	chong@example.com	pass987	g	P03
ID012	LEE	0107896532	lee@example.com	newpassword	g	P04
ID013	AMAR	0176452112	amar@example.com	qwerty123	r	P01
ID014	AQILAH	0153213345	aqilah@example.com	testpass	r	P02
ID015	ABHI	0134567856	abhi@example.com	passwordtest	r	P03
ID016	NAVVEEN	0134123498	navveen@example.com	examplepass	r	P04
ID017	SHAN	0134568909	shan@example.com	pass1234	r	P01
ID018	LIEW	0178900764	liew@example.com	mypass123	r	P02
ID019	KUGHAN	0145608765	kughan@example.com	pass@word	r	P03
ID020	NAVIN	0145678909	navin@example.com	p@ss123word	r	P04

*SELECT *FROM CUSTOMER*

CUSTOMERID	NAME	PHONENUMBER	EMAILADDRESS	PASSWORD	CUSTOMERTYPE	PRODUCTID
ID001	SUREN	0167458271	suren@example.com	password1234567890	r	P01
ID002	ALI	0195667345	ali@example.com	pass9876543210	g	P02
ID003	ABU	0184567890	abu@example.com	securepassword	g	P03
ID004	LISA	0165432344	lisa@example.com	strongpass	g	P04
ID005	DEPP	0156790089	depp@example.com	myp@ssword	g	P01
ID006	JOSEPH	0167899999	joseph@example.com	secretpass	g	P02
ID007	JONATHAN	0165553212	jonathan@example.com	pass123	g	P03
ID008	JACK	0178909345	jack@example.com	randompass	g	P04
ID009	MAX	0123421999	max@example.com	p@ssword	g	P01
ID010	RAJU	0189878654	raju@example.com	securepass123	g	P02
ID011	CHONG	0167456789	chong@example.com	pass987	g	P03
ID012	LEE	0107896532	lee@example.com	newpassword	g	P04
ID013	AMAR	0176452112	amar@example.com	qwerty123	r	P01
ID014	AQILAH	0153213345	aqilah@example.com	testpass	r	P02
ID015	ABHI	0134567856	abhi@example.com	passwordtest	r	P03
ID016	NAVVEEN	0134123498	navveen@example.com	examplepass	r	P04
ID017	SHAN	0134568909	shan@example.com	pass1234	r	P01
ID018	LIEW	0178900764	liew@example.com	mypass123	r	P02
ID019	KUGHAN	0145608765	kughan@example.com	pass@word	r	P03
ID020	NAVIN	0145678909	navin@example.com	p@ss123word	r	P04

UPDATE CUSTOMER SET PhoneNumber='0167458271' WHERE CustomerID ='ID001';

ORDERID	STAFFID
ORDER01	S02
ORDER02	S04
ORDER03	S06
ORDER04	S08
ORDER05	S10
ORDER06	S01
ORDER07	S03
ORDER08	S05
ORDER09	S07
ORDER10	S09
ORDER11	S02
ORDER12	S04
ORDER13	S06
ORDER14	S08
ORDER15	S10
ORDER16	S01
ORDER17	S03
ORDER18	S05
ORDER19	S07
ORDER20	S09

*SELECT **

FROM ORDERLIST

ORDERID	STAFFID
ORDER01	S02
ORDER02	S04
ORDER03	S06
ORDER04	S08
ORDER05	S10
ORDER06	S01
ORDER07	S03
ORDER08	S05
ORDER09	S07
ORDER10	S09
ORDER11	S02
ORDER12	S04
ORDER13	S06
ORDER14	S08
ORDER15	S10
ORDER16	S01
ORDER17	S03
ORDER18	S05
ORDER19	S07

DELETE FROM ORDERLIST WHERE OrderID='ORDER20'

CUSTOMERID	NAME	PHONENUMBER	EMAILADDRESS
ID004	LISA	0165432344	lisa@example.com

SELECT CustomerID, Name , PhoneNumber , EmailAddress

FROM CUSTOMER

WHERE CustomerID LIKE 'ID004';

PAYMENTID	PAYMENTTYPE	PAYMENTDATE	PAYMENTSTATUS	CUSTOMERID
P1112	B	20-JAN-24	1	ID002
P1113	B	20-JAN-24	1	ID003
P1114	A	20-JAN-24	1	ID004
P1115	A	20-JAN-24	1	ID005
P1116	B	20-JAN-24	1	ID006
P1118	B	20-JAN-24	1	ID008
P1119	A	20-JAN-24	1	ID009
P1121	B	20-JAN-24	1	ID011
P1122	B	20-JAN-24	1	ID012
P1125	A	20-JAN-24	1	ID015
P1127	B	20-JAN-24	1	ID017
P1130	B	20-JAN-24	1	ID020

SELECT OrderID,OrderDate,OrderTime ,OrderStatus,CustomerID,PaymentID

FROM ORDERS

WHERE PaymentID IN

('P1111','P1112','P1113','P1114','P1115','P1116','P1117','P1118','P1119','P1120');

CUSTOMERID	NAME	PHONENUMBER	EMAILADDRESS	PASSWORD	CUSTOMERTYPE	PRODUCTID	COMPANYID
ID001	SUREN	0167458271	suren@example.com	password1234567890	r	P01	C1
ID002	ALI	0195667345	ali@example.com	pass9876543210	g	P02	C4
ID003	ABU	0184567890	abu@example.com	securepassword	g	P03	C3
ID005	DEPP	0156790089	depp@example.com	myp@ssword	g	P01	C2
ID007	JONATHAN	0165553212	jonathan@example.com	pass123	g	P03	C8
ID008	JACK	0178909345	jack@example.com	randompass	g	P04	C7
ID009	MAX	0123421999	max@example.com	p@ssword	g	P01	C6
ID010	RAJU	0189878654	raju@example.com	securepass123	g	P02	C5
ID011	CHONG	0167456789	chong@example.com	pass987	g	P03	C9
ID013	AMAR	0176452112	amar@example.com	qwerty123	r	P01	C10

*SELECT **

FROM CUSTOMER NATURAL JOIN TALENT

CUSTOMERID	NAME	CUSTOMERTYPE	ORDERHISTORY
ID001	SUREN	r	CV Templates
ID013	AMAR	r	CV Writing Pro
ID014	AQILAH	r	CV Writing Pro
ID015	ABHI	r	CV Template
ID016	NAVVEEN	r	CV Writing Basic
ID017	SHAN	r	CV Template
ID018	LIEW	r	CV Writing Pro
ID019	KUGHAN	r	Webinar
ID020	NAVIN	r	Webinar

SELECT c.CustomerID, Name, CustomerType, g.OrderHistory

FROM CUSTOMER c JOIN REGISTEREDUSER g

ON c.CustomerID=g.CustomerID

ORDERID	ORDERDATE	CUSTOMERID	CUSTOMERNAME	PRODUCTNAME	PRICE
ORDER01	20-JAN-24	ID001	SUREN	CV Templates	99
ORDER02	20-JAN-24	ID002	ALI	CV Writing Basic	159
ORDER03	20-JAN-24	ID003	ABU	CV Writing Pro	199
ORDER04	20-JAN-24	ID004	LISA	Webinar	250
ORDER05	20-JAN-24	ID005	DEPP	CV Templates	99
ORDER06	20-JAN-24	ID006	JOSEPH	CV Writing Basic	159
ORDER07	20-JAN-24	ID007	JONATHAN	CV Writing Pro	199
ORDER08	20-JAN-24	ID008	JACK	Webinar	250
ORDER09	20-JAN-24	ID009	MAX	CV Templates	99
ORDER10	20-JAN-24	ID010	RAJU	CV Writing Basic	159
ORDER11	20-JAN-24	ID011	CHONG	CV Writing Pro	199
ORDER12	20-JAN-24	ID012	LEE	Webinar	250
ORDER13	20-JAN-24	ID013	AMAR	CV Templates	99
ORDER14	20-JAN-24	ID014	AQILAH	CV Writing Basic	159
ORDER15	20-JAN-24	ID015	ABHI	CV Writing Pro	199
ORDER16	20-JAN-24	ID016	NAVVEEN	Webinar	250
ORDER17	20-JAN-24	ID017	SHAN	CV Templates	99
ORDER18	20-JAN-24	ID018	LIEW	CV Writing Basic	159
ORDER19	20-JAN-24	ID019	KUGHAN	CV Writing Pro	199
ORDER20	20-JAN-24	ID020	NAVIN	Webinar	250

```
SELECT o.OrderID, o.OrderDate, c.CustomerID, c.Name AS CustomerName,
p.ProductName, p.Price
FROM ORDERS o JOIN CUSTOMER c
ON o.CustomerID = c.CustomerID
JOIN PRODUCT p
ON c.ProductID = p.ProductID;
```

CUSTOMERID	CUSTOMERNAME	PAYMENTID	PAYMENTTYPE	PAYMENTDATE
ID001	SUREN	P1111	A	20-JAN-24
ID002	ALI	P1112	B	20-JAN-24
ID003	ABU	P1113	B	20-JAN-24
ID004	LISA	P1114	A	20-JAN-24
ID005	DEPP	P1115	A	20-JAN-24
ID006	JOSEPH	P1116	B	20-JAN-24
ID007	JONATHAN	P1117	A	20-JAN-24
ID008	JACK	P1118	B	20-JAN-24
ID009	MAX	P1119	A	20-JAN-24
ID010	RAJU	P1120	B	20-JAN-24
ID011	CHONG	P1121	B	20-JAN-24
ID012	LEE	P1122	B	20-JAN-24
ID013	AMAR	P1123	B	20-JAN-24
ID014	AQILAH	P1124	A	20-JAN-24
ID015	ABHI	P1125	A	20-JAN-24
ID016	NAVVEEN	P1126	A	20-JAN-24
ID017	SHAN	P1127	B	20-JAN-24
ID018	LIEW	P1128	A	20-JAN-24
ID019	KUGHAN	P1129	A	20-JAN-24
ID020	NAVIN	P1130	B	20-JAN-24

```
SELECT c.CustomerID, c.Name AS CustomerName, p.PaymentID, p.PaymentType,
p.PaymentDate
FROM CUSTOMER c
LEFT OUTER JOIN PAYMENT p
ON c.CustomerID = p.CustomerID;
```

7.0 Summary

In this phase, we did the database logical design and the SQL implementation of GetMeHired.io. In this task, we transformed the conceptual entity relationship diagram (ERD) in phase 3 into logical ERD by determining the functional dependencies between the relationships based on the updated business rules. We generate the relations schemas from the logical ERD produced followed by performing the normalization to reduce the redundancy of the data and make the website easy to access and manipulate the website efficiently without compromising the data integrity which can benefit future work.

In conclusion, we hoped that throughout this phase, we could produce a functional and user-friendly system to help out those who need assistance in creating CVs.