



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

SECD2523 - DATABASE

SEMESTER 1 2023/2024

P3 – Database Logical Design & SQL

GROUP 6

Group name: YYDS

Group member: a. SOO WEN CHUN A22EC0105

b. ENG JUN XIANG A22EC0049

c. KONG YEW YEONG A22EC0061

d. LIEW YONG ZHENG A23CS5008

Section: section 10

Lecturer: Dr ROZILAWATI BINTI DOLLAH

Table of Contents

1.0 Introduction.....	3
2.0 Overview of project	4
3.0 Database conceptual design	5
3.1 Updated business rule	5
3.2 Conceptual ERD	6
3.2.1 Conceptual ERD	6
3.2.2 Enhanced ERD.....	7
4.0 DB logical design.....	8
4.1 Logical ERD	8
4.2 Updated Data Dictionary	9
4.3 Normalization	14
5.0 Relational DB Schemas (after normalization)	16
6.0 SQL Statements.....	18
6.1 Create Table	18
6.2 Alter Table.....	20
6.3 Insert Value	22
6.4 Testing.....	25
7.0 Summary	27

1.0 Introduction

EdenShop's success in online retail is about excelling in several fundamental aspects, redefining the seller experience. Our platform provides an intuitive and efficient inventory management system that provides real-time updates and predictive analytics to simplify tracking and optimization of product inventory. Sellers can easily navigate to a more user-friendly interface, allowing them to focus on core business activities.

In addition, EdenShop provides sellers with valuable data analysis and opinions to help them fully understand market trends and customer behaviour. This knowledge becomes a strategic advantage, helping to make informed decisions and identify opportunities for growth within the platform. Security is also at the core of EdenShop, using strong measures such as secure payment gateways and encryption protocols to protect sensitive information and foster trust between sellers and customers.

Finally, our commitment to order accuracy and processing is evident through instant notifications, automated tracking, and status updates, ensuring a smooth, error-free fulfilment process. EdenShop is more than just a platform, it is a user-friendly ecosystem designed to support sellers in every aspect, promising increased efficiency, security, and unparalleled success in the competitive landscape of e-commerce.

2.0 Overview of project

In phase 3, we advance from conceptual design to the implementation of the EdenShop database. The project overview highlights our focus on refining business rules and converting them into simplified and enhanced versions of conceptual entity-relationship diagrams (ERDs). Turning to the logical design, we build the logical ERD by describing the functional dependencies of the update rules.

At the same time, the data dictionary is updated to provide detailed descriptions reflecting logical ERD changes. Normalisation is then applied to improve data storage efficiency. The final stage of this phase is to create a relational database schema that is consistent with the normalised structure. Crucially, SQL statements (DDL and DML) are used to implement the logical design in Oracle Apex, underscoring our commitment to delivering a well-structured and efficient database system suitable for EdenShop's ever-changing needs.

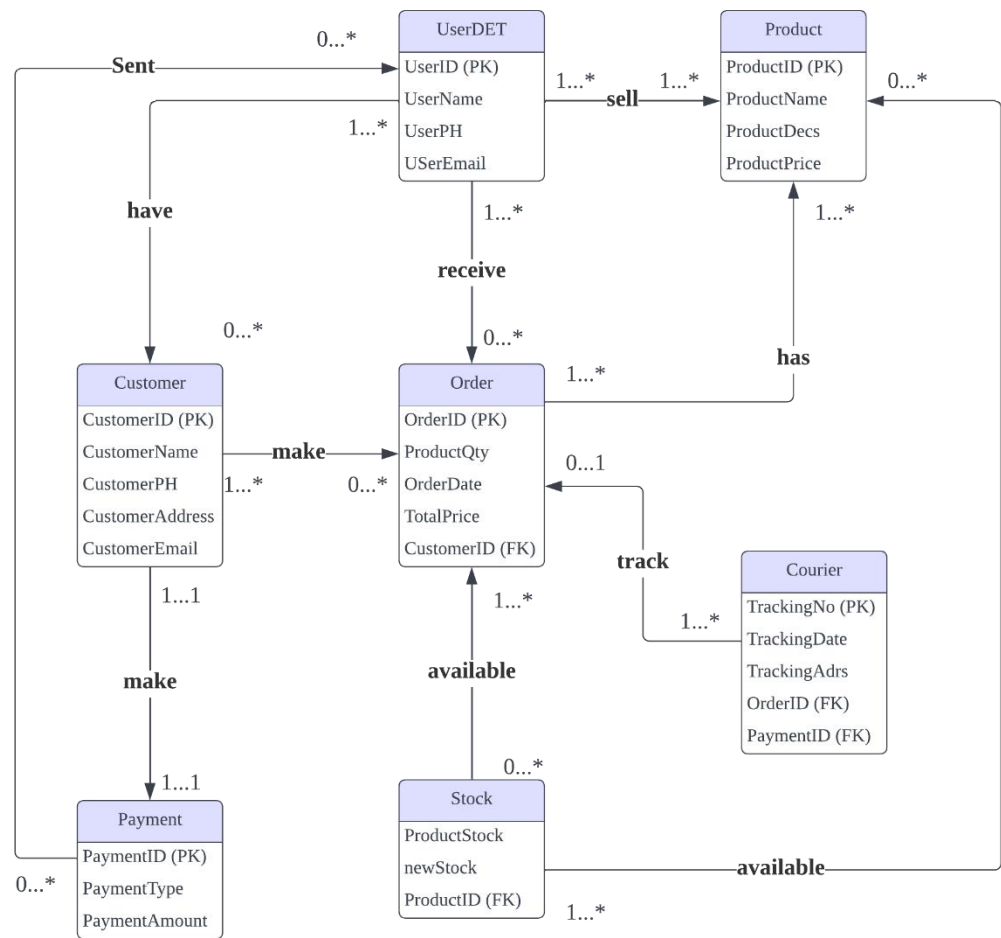
3.0 Database conceptual design

3.1 Updated business rule

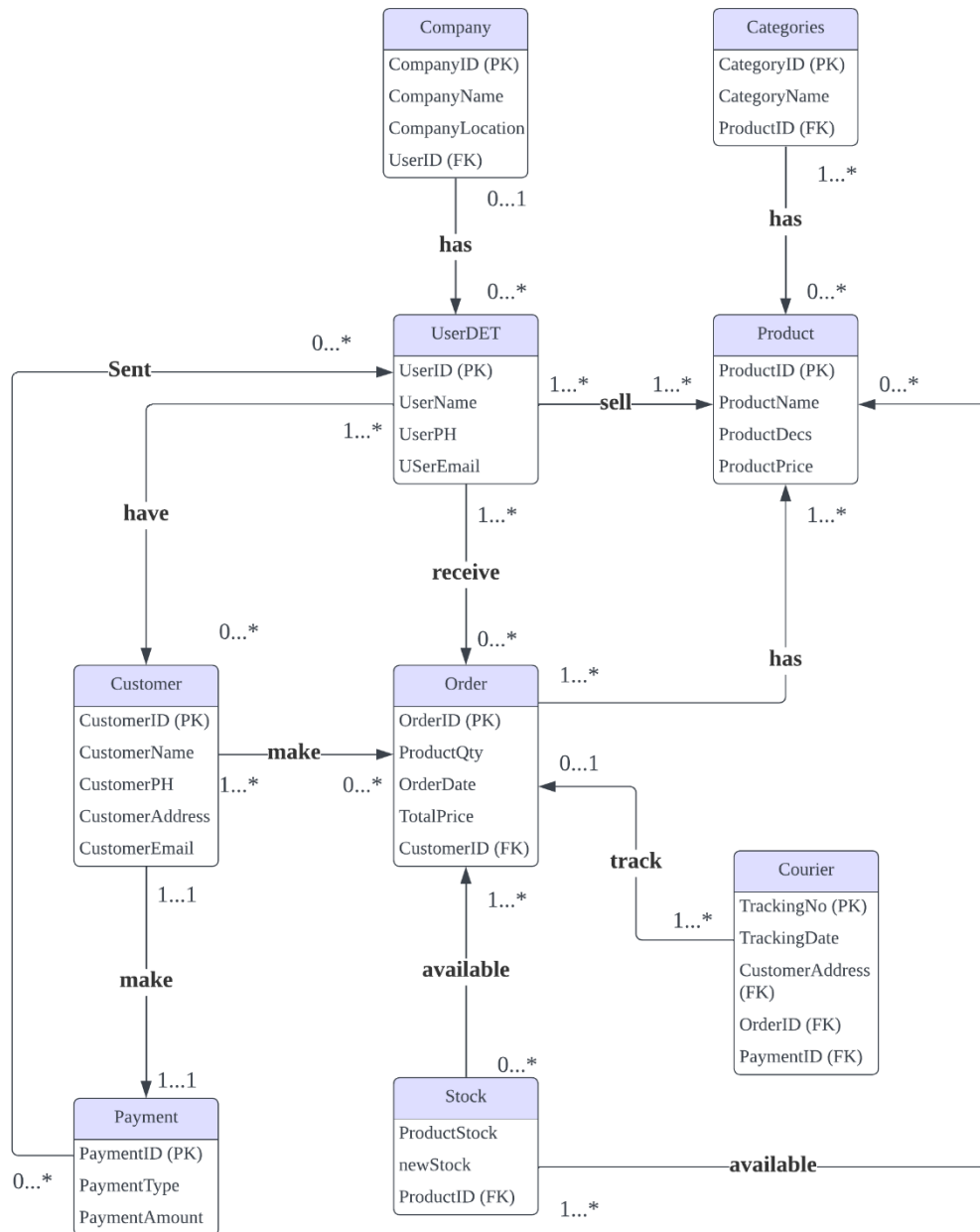
1. Each customer can place multiple order
2. Each order contains one or more quantity of products
3. Each customer can place multiple order from several users (sellers)
4. Each user (seller) has their own identifier
5. Each customer has their own identifier
6. Each user (seller) can sell multiple product
7. Each customer can only make one payment
8. Each tracking has its tracking number
9. Each user (seller) is allowed to check order history
10. Each user (seller) will be known the current available stock for products
11. User need to check the stock of products to ensure the order placement is processing well
12. User need to open notification to notice them when the stock is nearly empty or too much to prevent overstock and understock
13. Couriers need to provide tracking numbers once the parcel is delivered.
14. Couriers need to update the status and provide the live location of parcels.
15. Customers need to enter the order id or tracking number to track their purchases.
16. Once an order is placed, the bank needs to provide online payment details as a proof for the continuous delivery process.
17. The online payment details will be sent to users and the status of payment is updated.
18. Users need to identify the type of products for better sorting in categories.

3.2 Conceptual ERD

3.2.1 Conceptual ERD

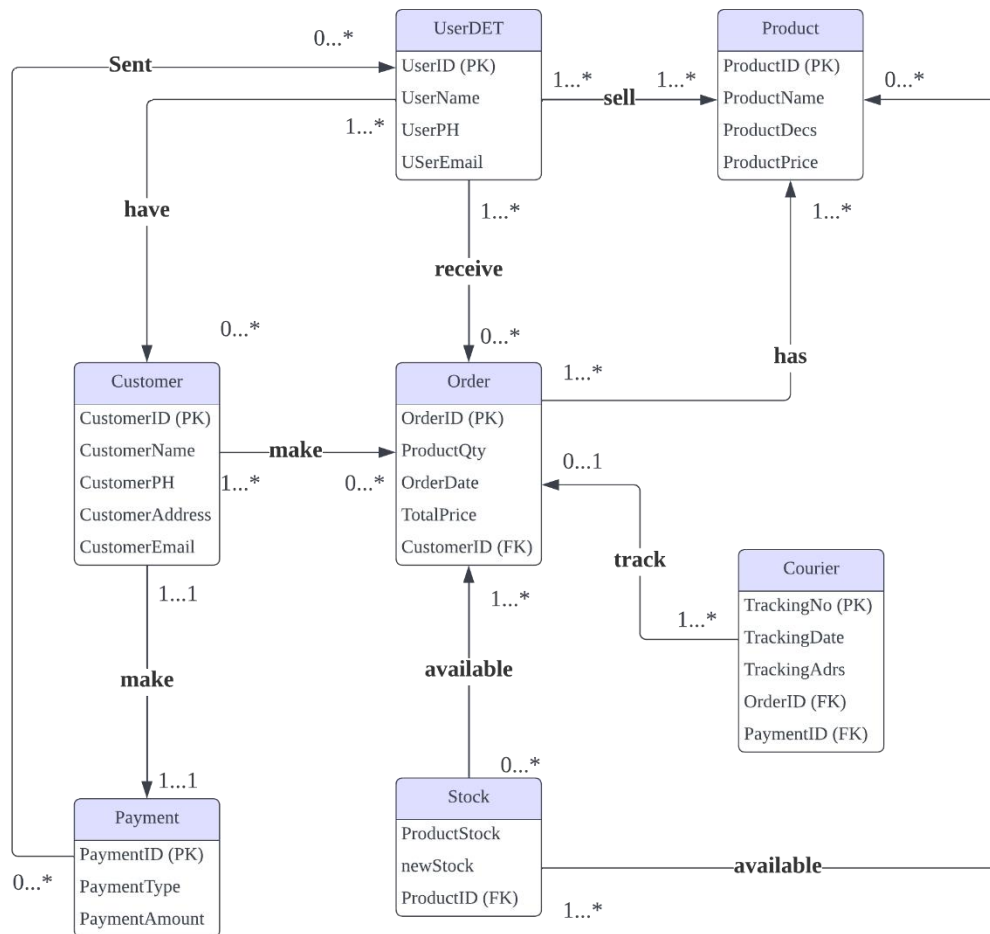


3.2.2 Enhanced ERD



4.0 DB logical design

4.1 Logical ERD



4.2 Updated Data Dictionary

Description of Entities

Entity	Description	Occurrence
Customer	Hold the data of customer	Customer's information is sent to users after the ordering process
User	Hold the data of user	Users proceed the orders made by customers and sent them to courier
Order	Hold the data of order made by customer	Once an order is placed by customers, a tracking number is generated to courier
Payment	Hold the data of payment made by customer	Payment is processed by customers after placing an order
Courier	Hold the data of courier	Courier provides the status of the parcel and its live location to the customers and users.
Stock	Hold the data of stock for each product	Stock is updated when the products are added or sold
Company	Hold the data of company from each user	The information of company is recorded to improve trustment
Categories	Hold the data of categories for each product	The products are displayed in respective category after filter the products

Relationship between entities

Entity	Multiplicity	Relationship	Multiplicity	Entity
UserDET	1...*	Have	0...*	Customer
	1...*	Sell	1...*	Product
	1...*	Receive	0...*	Order
Customer	1...*	Make	0...*	Order
	1...1	Make	1...1	Payment
Courier	1...*	Track	0...1	Order
Stock	0...*	Available	1...*	Order
	1...*	Available	1...*	Product
Order	1...*	Has	1...*	Product
Company	0...1	Has	0...*	Product
Categories	1...*	has	0...*	UserDET
Payment	0...1	sent	0...1	UserDET

Attributes in Entities

Entity	Attribute	Description	Data Type	Constraint
UserDET	UserID	User ID	VARCHAR2(16)	PRIMARY KEY
	UserName	User's name	VARCHAR2(30)	NOT NULL
	UserPH	User's phone number	VARCHAR2(12)	NOT NULL
	UserEmail	User's email	VARCHAR2(30)	NOT NULL
Customer	CustomerID	Customer ID	VARCHAR2(16)	PRIMARY KEY
	CustomerName	Customer's name	VARCHAR2(30)	NOT NULL
	CustomerPH	Customer's phone number	VARCHAR2(12)	NOT NULL
	CustomerAddress	Customer's address	VARCHAR2(50)	NOT NULL
	CustomerEmail	Customer's email	VARCHAR2(30)	NOT NULL
Product	ProductID	Product's ID	VARCHAR2(10)	PRIMARY KEY
	ProductName	Product's Name	VARCHAR2(20)	NOT NULL
	ProductDesc	Product's description	VARCHAR2(200)	NOT NULL
	ProductPrice	Product's price	DECIMAL (10,2)	NOT NULL
Orders	OrderID	Order ID	VARCHAR2(10)	PRIMARY KEY
	ProductQty	Order quantity	NUMBER(2)	NOT NULL
	OrderDate	Order date	DATE	NOT NULL
	TotalPrice	Total price of Order	DECIMAL (8,2)	NOT NULL
	CustomerID	Customer ID	VARCHAR2(16)	FOREIGN KEY reference Customer

Stock	ProductStock	Product Stock	NUMBER(2)	NOT NULL
	newStock	Number of new stocks added	NUMBER (3)	NOT NULL
	ProductID	Product ID	VARCHAR2(10)	FOREIGN KEY reference Product
Payment	PaymentID	Payment ID	VARCHAR2(12)	PRIMARY KEY
	PaymentType	Type of payment	CHAR (1)	NOT NULL
	PaymentAmount	Amount of payment	DECIMAL (8,2)	NOT NULL
Courier	TrackingNo	Tracking number	VARCHAR2(12)	PRIMARY KEY
	TrackingDate	Tracking date	DATE	NOT NULL
	TrackingAdrs	Tracking Address	VARCHAR2(50)	NOT NULL
	OrderID	Order ID	VARCHAR2(10)	FOREIGN KEY reference Order
	PaymentID	Payment ID	VARCHAR2(12)	FOREIGN KEY reference Payment
Company	CompanyID	Company ID	VARCHAR2(12)	PRIMARY KEY
	CompanyName	Company Name	VARCHAR2(20)	NOT NULL
	CompanyLocation	The location of company	VARCHAR2(50)	NOT NULL
	UserID	User ID	VARCHAR2(16)	FOREIGN KEY reference User
Categories	CategoryID	Category ID	VARCHAR2(12)	PRIMARY KEY

	CategoryName	The category of products	VARCHAR2(20)	NOT NULL
	ProductID	Product ID	VARCHAR2(10)	FOREIGN KEY reference Product

4.3 Normalization

1. UserDET (UserID, UserName, UserPH, UserEmail)

fd1: UserID → UserName, UserPH, UserEmail

1NF&2NF&3NF&BCNF:

UserDET (UserID, UserName, UserPH, UserEmail)

2. Customer (CustomerID, CustomerName, CustomerPH, CustomerAddress, CustomerEmail)

fd1: CustomerID → CustomerName, CustomerPH, CustomerAddress, CustomerEmail

1NF&2NF&3NF&BCNF:

Customer (CustomerID, CustomerName, CustomerPH, CustomerAddress, CustomerEmail)

3. Product (ProductID, ProductName, ProductDecs, ProductPrice)

fd1: ProductID → ProductName, ProductDecs, ProductPrice

1NF&2NF&3NF&BCNF:

Product (ProductID, ProductName, ProductDecs, ProductPrice)

4. Order (OrderID, ProductQty, OrderDate, TotalPrice, CustomerID)

fd1: OrderID → ProductQty, OrderDate, TotalPrice, CustomerID

1NF&2NF&3NF&BCNF:

Order (OrderID, ProductQty, OrderDate, TotalPrice, CustomerID)

5. Stock (ProductStock, newStock, ProductID)

fd1: ProductStock → newStock, ProductID

1NF&2NF&3NF&BCNF:

Stock (ProductStock, newStock, ProductID)

6. Payment (PaymentID, PaymentType, PaymentAmount)

fd1: PaymentID → PaymentType, PaymentAmount

1NF&2NF&3NF&BCNF:

Payment (PaymentID, PaymentType, PaymentAmount)

7. Courier (TrackingNo, TrackingDate, TrackingAdrs, OrderID, PaymentID)

fd1: TrackingNo → TrackingDate, TrackingAdrs, OrderID, PaymentID

1NF&2NF&3NF&BCNF:

Courier (TrackingNo, TrackingDate, TrackingAdrs, OrderID, PaymentID)

8. Company (CompanyID, CompanyName, CompanyLocation, UserID)

fd1: CompanyID → CompanyName, CompanyLocation, UserID

1NF&2NF&3NF&BCNF:

Company (CompanyID, CompanyName, CompanyLocation, UserID)

9. Categories (CategoryID, CategoryName, ProductID)

fd1: CategoryID → CategoryName, ProductID

1NF&2NF&3NF&BCNF:

Categories (CategoryID, CategoryName, ProductID)

5.0 Relational DB Schemas (after normalization)

1. UserDET (UserID, UserName, UserPH, UserEmail)
2. Customer (CustomerID, CustomerName, CustomerPH, CustomerAddress, CustomerEmail)
3. Product (ProductID, ProductName, ProductDecs, ProductPrice)
4. Order (OrderID, ProductQty, OrderDate, TotalPrice, CustomerID)
5. Stock (ProductStock, newStock, ProductID)
6. Payment (PaymentID, PaymentType, PaymentAmount)
7. Courier (TrackingNo, TrackingDate, TrackingAdrs, OrderID, PaymentID)
8. Company (CompanyID, CompanyName, CompanyLocation, UserID)
9. Categories (CategoryID, CategoryName, ProductID)

UserDET

UserID	UserName	UserPH	UserEmail
--------	----------	--------	-----------

Customer

CustomerID	CustomerName	CustomerPH	CustomerAddress	CustomerEmail
------------	--------------	------------	-----------------	---------------

Product

ProductID	ProductName	ProductDecs	ProductPrice
-----------	-------------	-------------	--------------

Order

OrderID	ProductQty	OrderDate	TotalPrice	CustomerID
---------	------------	-----------	------------	------------

Stock

ProductStock	newStock	ProductID
--------------	----------	-----------

Payment

PaymentID	PaymentType	PaymentAmount
-----------	-------------	---------------

Courier

TrackingNo	TrackingDate	TrackingAdrs	OrderID	PaymentID
------------	--------------	--------------	---------	-----------

Company

CompanyID	CompanyName	CompanyLocation	userID
-----------	-------------	-----------------	--------

Categories

CategoryID	CategoryName	ProductID
------------	--------------	-----------

6.0 SQL Statements

6.1 Create Table

```
1  -----CREATE TABLE-----
2  ----User----
3  CREATE TABLE UserDet(
4      userID varchar2(16),
5      userName varchar(30) NOT NULL,
6      userPH varchar2(12) NOT NULL,
7      userEmail varchar2(30) NOT NULL,
8      CONSTRAINT userID_PK PRIMARY KEY (userID)
9  );
10
11  ----Customer----
12  CREATE TABLE Customer(
13      CustomerID varchar2(16),
14      CustomerName varchar(30) NOT NULL,
15      CustomerPH varchar2(12) NOT NULL,
16      CustomerAddress varchar2(50) NOT NULL,
17      CustomerEmail varchar2(30) NOT NULL,
18      CONSTRAINT CustomerID_PK PRIMARY KEY (CustomerID)
19  );
20
21  ----Product----
22  CREATE TABLE Product(
23      ProductID varchar2(10),
24      ProductName varchar2(20) NOT NULL,
25      ProductDecs varchar2(200) NOT NULL,
26      ProductPrice DECIMAL(10,2) NOT NULL,
27      CONSTRAINT ProductID_PK PRIMARY KEY (ProductID)
28  );
```

```
29
30  ----Order----
31  CREATE TABLE Orders(
32      OrderID varchar2(10),
33      ProductQty NUMBER(2) NOT NULL,
34      OrderDate DATE NOT NULL,
35      TotalPrice DECIMAL(8,2) NOT NULL,
36      CustomerID varchar2(16) NOT NULL,
37      CONSTRAINT OrderID_PK PRIMARY KEY (OrderID)
38  );
39
40  ----Stock----
41  CREATE TABLE Stock(
42      ProductStock NUMBER(2) NOT NULL,
43      newStock NUMBER(3) NOT NULL,
44      ProductID varchar2(10) NOT NULL
45  );
46
47  ----Payment----
48  CREATE TABLE Payment(
49      PaymentID varchar2(12),
50      PaymentType CHAR(1) NOT NULL,
51      PaymentAmount DECIMAL(8,2) NOT NULL,
52      CONSTRAINT PaymentID_PK PRIMARY KEY (PaymentID)
53  );
54
```

```

55
56 ----Courier----
57 CREATE TABLE Courier(
58     TrackingNo varchar2(12),
59     TrackingDate DATE NOT NULL,
60     TrackingAdrs varchar2(50) NOT NULL,
61     OrderID varchar2(10) NOT NULL,
62     PaymentID varchar2(12) NOT NULL,
63     CONSTRAINT TrackingNo_PK PRIMARY KEY (TrackingNo)
64 );
65
66 ----Company----
67 CREATE TABLE Company(
68     CompanyID varchar2(12),
69     CompanyName varchar2(20) NOT NULL,
70     CompanyLocation varchar2(50) NOT NULL,
71     userID varchar2(16) NOT NULL,
72     CONSTRAINT CompanyID_PK PRIMARY KEY (CompanyID)
73 );
74
75 ----Categories----
76 CREATE TABLE Categories(
77     CategoryID varchar2(12),
78     CategoryName varchar2(20) NOT NULL,
79     ProductID varchar2(10) NOT NULL,
80     CONSTRAINT CategoryID_PK PRIMARY KEY (CategoryID)
81 );
82

```

6.2 Alter Table

```
81
82  -----ADD CONSTRAINT FOREIGN KEY-----
83  ---ORDERS---
84  ALTER TABLE Orders
85  ADD CONSTRAINT CustomerID_FK FOREIGN KEY (CustomerID)
86  REFERENCES Customer (CustomerID)
87
```

```
90
91  ---STOCK---
92  ALTER TABLE Stock
93  ADD CONSTRAINT ProductID_FK FOREIGN KEY (ProductID)
94  REFERENCES Product (ProductID);
95
96  ---COURIER---
97  ALTER TABLE Courier
98  ADD CONSTRAINT OrderID_2_FK FOREIGN KEY (OrderID)
99  REFERENCES Orders (OrderID)
100  ADD CONSTRAINT PaymentID_FK FOREIGN KEY (PaymentID)
101  REFERENCES Payment (PaymentID);
102
103  ---Company---
104  ALTER TABLE Company
105  ADD CONSTRAINT userID_2_FK FOREIGN KEY (userID)
106  REFERENCES UserDET (userID);
107
```

```
107
108
109  -----Categories-----
110  ALTER TABLE Categories
111  ADD CONSTRAINT ProductID_3_FK FOREIGN KEY (ProductID)
112  REFERENCES Product (ProductID);
113
```

15 minutes ago	ALTER TABLE Categories ADD CONSTRAINT ProductID_3_FK FOREIGN KEY (ProductID) REFERENCES Product (Pro	WKSP_DBPJT
17 minutes ago	ALTER TABLE Company ADD CONSTRAINT userID_2_FK FOREIGN KEY (userID) REFERENCES UserDET (userID);	WKSP_DBPJT
19 minutes ago	ALTER TABLE Courier ADD CONSTRAINT OrderID_2_FK FOREIGN KEY (OrderID) REFERENCES Orders (OrderID) AD	WKSP_DBPJT
22 minutes ago	ALTER TABLE Stock ADD CONSTRAINT ProductID_FK FOREIGN KEY (ProductID) REFERENCES Product (ProductID)	WKSP_DBPJT

22 minutes ago	ALTER TABLE Stock ADD CONSTRAINT ProductID_FK FOREIGN KEY (ProductID) REFERENCES Product (ProductID)
25 minutes ago	ALTER TABLE Stock ADD CONSTRAINT ProductID_FK FOREIGN KEY (ProductID) REFERENCES Product (ProductID)
27 minutes ago	ALTER TABLE Orders ADD CONSTRAINT CustomerID_FK FOREIGN KEY (CustomerID) REFERENCES Customer (Custom
29 minutes ago	CREATE TABLE Categories(CategoryID varchar2(12), CategoryName varchar2(20) NOT NULL, Pr
32 minutes ago	CREATE TABLE Company(CompanyID varchar2(12), CompanyName varchar2(20) NOT NULL, Company
32 minutes ago	CREATE TABLE Company(CompanyID varchar2(12), CompanyName varchar2(20) NOT NULL, Company
35 minutes ago	CREATE TABLE Courier(TrackingNo varchar2(12), TrackingDate DATE NOT NULL, TrackingAdrs
38 minutes ago	CREATE TABLE Payment(PaymentID varchar2(12), PaymentType CHAR(1) NOT NULL, PaymentAmoun
40 minutes ago	CREATE TABLE Stock(ProductQty NUMBER(2) NOT NULL, newStock NUMBER(3) NOT NULL, ProductI
43 minutes ago	CREATE TABLE Orders(OrderID varchar2(10), ProductQty NUMBER(2) NOT NULL, OrderDate DATE
48 minutes ago	CREATE TABLE Product(ProductID varchar2(10), ProductName varchar2(20) NOT NULL, Product
51 minutes ago	CREATE TABLE Customer(CustomerID varchar2(16), CustomerName varchar(30) NOT NULL, Custo
52 minutes ago	CREATE TABLE UserDet(userID varchar2(16), userName varchar(30) NOT NULL, userPH varchar

6.3 Insert Value

```
113
114 -----Insert Value-----
115 ----User----
116 INSERT INTO UserDET
117 VALUES('U0001', 'Kong', 0123456, 'kong@gmail.com');
118 INSERT INTO UserDET
119 VALUES('U0002', 'Soo', 0234567, 'soo@gmail.com');
120 INSERT INTO UserDET
121 VALUES('U0003', 'Eng', 0345678, 'eng@gmail.com');
122 INSERT INTO UserDET
123 VALUES('U0004', 'Liow', 0456789, 'liow@gmail.com');
124 INSERT INTO UserDET
125 VALUES('U0005', 'Ee', 0567890, 'ee@gmail.com');
126
127 ----Customer----
128 INSERT INTO Customer
129 VALUES('C0001', 'Winki', '1234567', 'Lot 1111', 'winki@gmail.com');
130 INSERT INTO Customer
131 VALUES('C0002', 'Keying', '2345678', 'Lot 2222', 'keying@gmail.com');
132 INSERT INTO Customer
133 VALUES('C0003', 'Xiaoxuan', '3456789', 'Lot 3333', 'xiaoxuan@gmail.com');
134 INSERT INTO Customer
135 VALUES('C0004', 'Yunyi', '4567890', 'Lot 4444', 'yunyi@gmail.com');
136 INSERT INTO Customer
137 VALUES('C0005', 'ZhiXuan', '5678901', 'Lot 5555', 'zhixuan@gmail.com');
138
```

```
139 ----Product----
140 INSERT INTO Product
141 VALUES('P0001', 'pen', 'This is a pen made from US', 35.50);
142 INSERT INTO Product
143 VALUES('P0002', 'shirt', 'This is a shirt made from Thailand', 68.99);
144 INSERT INTO Product
145 VALUES('P0003', 'shoes', 'This is a shoes made from UK', 400.00);
146 INSERT INTO Product
147 VALUES('P0004', 'Travel bag', 'This is a beg made from US', 60.00);
148 INSERT INTO Product
149 VALUES('P0005', 'socks', 'This is a socks made from China', 15.00);
150
```

```
150 ----Orders----
151 INSERT INTO Orders
152 VALUES('B0001', 1, SYSDATE, 15.00, 'C0001');
153 INSERT INTO Orders
154 VALUES('B0002', 1, SYSDATE, 400.00, 'C0004');
155 INSERT INTO Orders
156 VALUES('B0003', 3, SYSDATE, 180.00, 'C0003');
157 INSERT INTO Orders
158 VALUES('B0004', 1, SYSDATE, 688.99, 'C0002');
159 INSERT INTO Orders
160 VALUES('B0005', 2, SYSDATE, 70.00, 'C0005');
161
```

```
162 VALUES( '00003',2,SYSDATE,70100, '00003', '10003' );
163
164 ----Stock----
165 INSERT INTO Stock
166 VALUES(2,200,'P0001');
167 INSERT INTO Stock
168 VALUES(5,20,'P0002');
169 INSERT INTO Stock
170 VALUES(10,400,'P0003');
171 INSERT INTO Stock
172 VALUES(10,87,'P0004');
173 INSERT INTO Stock
174 VALUES(68,69,'P0005');
175
176 ----Payment----
177 INSERT INTO Payment
178 VALUES('X0001',1,15.00);
179 INSERT INTO Payment
180 VALUES('X0002',2,400.00);
181 INSERT INTO Payment
182 VALUES('X0003',3,180.00);
183 INSERT INTO Payment
184 VALUES('X0004',1,688.99);
185 INSERT INTO Payment
186 VALUES('X0005',2,70.00);
187
```

```

187
188 ----Company----
189 INSERT INTO Company
190 VALUES('CP0001','EdenShop','Lot Perak','U0001');
191 INSERT INTO Company
192 VALUES('CP0002','Vrify','Lot Selangor','U0002');
193 INSERT INTO Company
194 VALUES('CP0003','Sakei','Lot Johor','U0003');
195 INSERT INTO Company
196 VALUES('CP0004','Beggy','Lot Kedah','U0004');
197 INSERT INTO Company
198 VALUES('CP0005','Shopy','Lot Pahang','U0005');
199
200 ----Categories----
201 INSERT INTO Categories
202 VALUES('G0001','Stationery','P0001');
203 INSERT INTO Categories
204 VALUES('G0002','Shirt','P0002');
205 INSERT INTO Categories
206 VALUES('G0003','Sporting','P0003');
207 INSERT INTO Categories
208 VALUES('G0004','Sporting Accessories','P0005');
209 INSERT INTO Categories
210 VALUES('G0005','beg','P0004');
211

```

```

211
212 ----Courier----
213 INSERT INTO Courier
214 VALUES('T0001',SYSDATE,'Lot 1111','B0001','X0001');
215 INSERT INTO Courier
216 VALUES('T0002',SYSDATE,'Lot 4444','B0002','X0002');
217 INSERT INTO Courier
218 VALUES('T0003',SYSDATE,'Lot 3333','B0003','X0003');
219 INSERT INTO Courier
220 VALUES('T0004',SYSDATE,'Lot 2222','B0004','X0004');
221 INSERT INTO Courier
222 VALUES('T0005',SYSDATE,'Lot 5555','B0005','X0005');
223
224 ----Testing----

```


6.4 Testing

#Test 1 Display User

```
1 SELECT * from UserDET;
```

USERID	USERNAME	USERPH	USEREMAIL
U0001	Kong	123456	kong@gmail.com
U0003	Eng	545678	eng@gmail.com
U0004	Liew	456789	liew@gmail.com
U0005	Ee	567890	ee@gmail.com
U0002	Soo	234567	soo@gmail.com

5 rows returned in 0.02 seconds [Download](#)

#Test 2 Display customer's name and id

```
1 SELECT CustomerName, CustomerID
2 FROM Customer;
```

CUSTOMERNAME	CUSTOMERID
Yunyi	C0004
ZhiXuan	C0005
Winki	C0001
Xiaoxuan	C0003
Keying	C0002

5 rows returned in 0.00 seconds [Download](#)

#Test 3 Customer Name and the details of order made

```
1 SELECT C.customerName AS "Customer Name", O.OrderID AS "Order ID", O.OrderDate AS "Order Date", O.TotalPrice AS "Total Price"
2 FROM(
3     Orders O
4     JOIN Customer C ON O.customerID = C.CustomerID
5 )
6 ORDER BY O.TotalPrice;
```

Customer Name	Order ID	Order Date	Total Price
Winki	B0001	01/19/2024	15
ZhiXuan	B0005	01/19/2024	70
Xiaoxuan	B0003	01/19/2024	180
Yunyi	B0002	01/19/2024	400
Keying	B0004	01/19/2024	688.99

6 rows returned in 0.01 seconds [Download](#)

#Test 4 Filter and display the product with the price between 50.00 and 300.00

```
1 SELECT * from Product
2 where ProductPrice between 50.00 and 300.00;
```

PRODUCTID	PRODUCTNAME	PRODUCTDECS	PRODUCTPRICE
P0004	Travel bag	This is a bag made from US	60
P0002	shirt	This is a shirt made from Thailand	68.99

#Test 5 Display the product in respective categories

```
1 ---Display the product in respective categories---
2 SELECT P.ProductID AS "Product ID", P.ProductName AS "Product Name", P.ProductPrice AS "Product Price", C.CategoryName AS "Category Name"
3 FROM(
4     Product P
5     JOIN Categories C ON P.ProductID = C.ProductID
6 )
7 ORDER BY P.ProductPrice;
```

Product ID	Product Name	Product Price	Category Name
P0005	socks	15	Sporting Accessories
P0001	pen	35.5	Stationery
P0004	Travel bag	60	bag
P0002	shirt	68.99	Shirt
P0003	shoes	400	Sporting

5 rows returned in 0.01 seconds [Download](#)

#Test 6 Update Stock

```
1 SELECT ProductID AS "Product ID", ProductStock AS "Current Stock", newStock AS "New Stock Added", ProductStock + newStock AS "Stock Updated"
2 FROM Stock
3 ORDER BY ProductID;
```

Product ID	Current Stock	New Stock Added	Stock Updated
P0001	2	200	202
P0002	5	20	25
P0003	10	400	410
P0004	10	87	97
P0005	68	69	137

5 rows returned in 0.01 seconds [Download](#)

#Test 7 Filter and display the total price of orders which is between 10.00 and 200.00

```
1 SELECT * from Orders
2 where TotalPrice between 10.00 and 200.00;
```

ORDERID	PRODUCTQTY	ORDERDATE	TOTALPRICE	CUSTOMERID
B0003	3	01/19/2024	180	C0003
B0005	2	01/19/2024	70	C0005
B0001	1	01/19/2024	15	C0001

3 rows returned in 0.01 seconds [Download](#)

#Test 8 Display the company name and location of each users

```
1 SELECT U.userName AS "User Name", C.CompanyName AS "Company Name", C.CompanyID AS "Company ID", C.CompanyLocation AS "The location of Company"
2 FROM(
3     Company C
4     JOIN UserDet U ON C.userID = U.userID
5 )
6 ORDER BY C.CompanyID;
```

User Name	Company Name	Company ID	The Location of Company
Kong	EdenShop	CP0001	Lot Perak
Soo	Vnify	CP0002	Lot Selangor
Eng	Sakei	CP0003	Lot Johor
Liew	Beggy	CP0004	Lot Kedah
Ee	Shopy	CP0005	Lot Pahang

5 rows returned in 0.01 seconds [Download](#)

7.0 Summary

During phase 3, our focus was on the comprehensive design and implementation of the EdenShop database, ensuring a solid foundation for seamless operations. This phase begins with an introduction and then provides an overview of the project, laying the foundation for an in-depth exploration of the database conceptual design. This includes refining business rules to adapt to changing needs, developing conceptual entity relationship diagrams (ERDs), and enhanced ERDs that provide a visual representation of entity relationships.

Moving into the realm of logical design, phase 3 involves creating a logical ERD by establishing functional dependencies based on updated business rules. This process is complemented by an updated Data Dictionary, an important resource that provides detailed insights into entities, relationships, and attributes. Additionally, we delve into normalisation to ensure data integrity and efficiency from First Normal Form (1NF) to Boyce-Codd Normal Form (BCNF). The resulting relational database schema reflects our commitment to minimising data redundancy.

At the end of this phase, SQL statements (DDL and DML) are implemented to materialise the logical design into a tangible database using Oracle Apex. Our efforts at this stage are aligned with the overall goal of creating a fully functional, user-friendly system for EdenShop. By addressing the complexity of database design, we aim to optimise efficiency, enhance data accessibility, and ultimately contribute to the success of our stakeholders' online retail careers.