`

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

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.

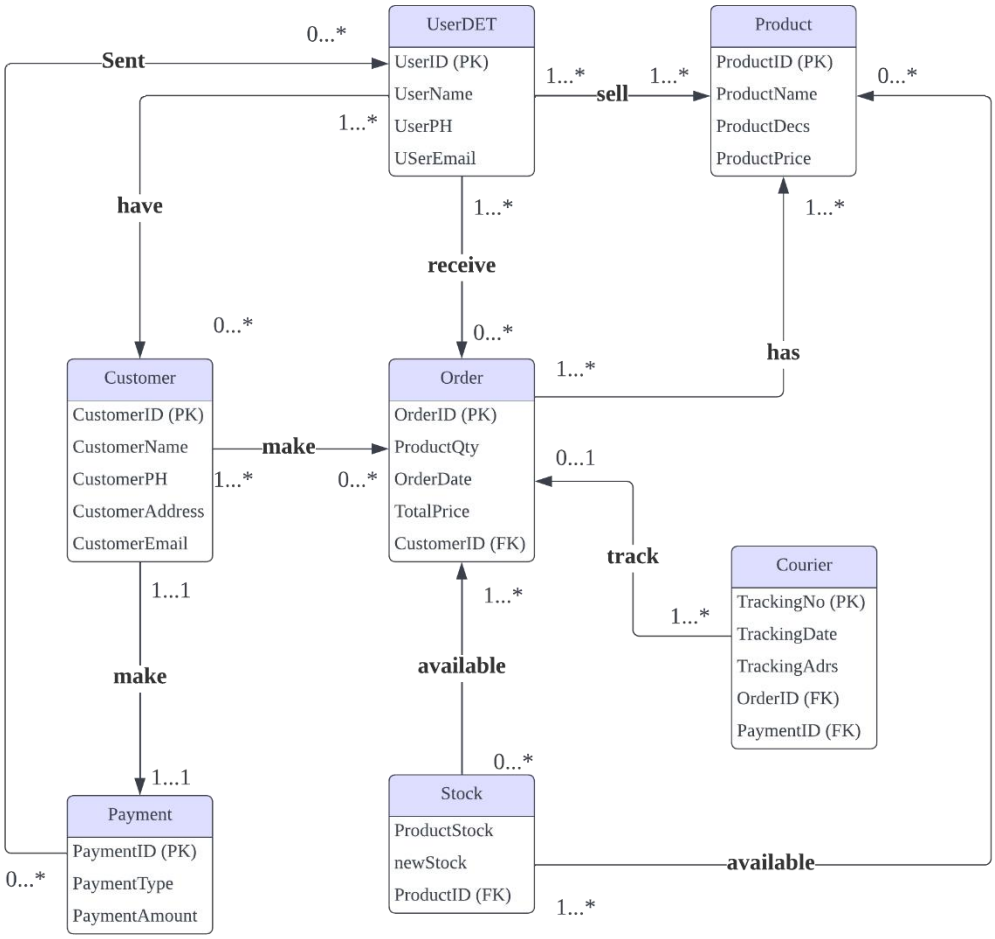
16. Once an order is placed, the bank needs to provide online payment details as a proof for the continuous delivery process.

17. The online payment details will be sent to users and the status of payment is updated.

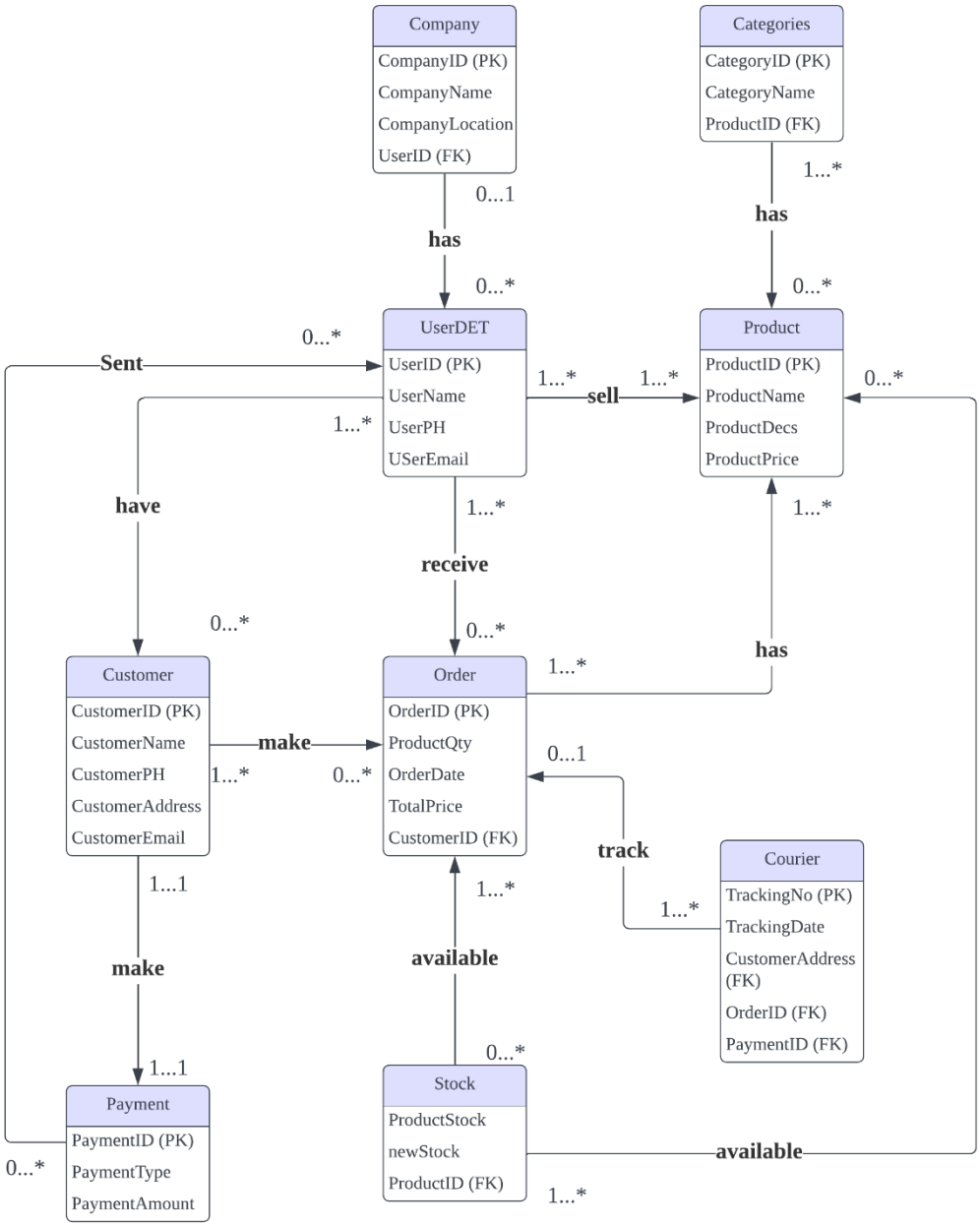
18. Users need to identify the type of products for better sorting in categories.

3.2 Conceptual ERD

3.2.1 Conceptual ERD

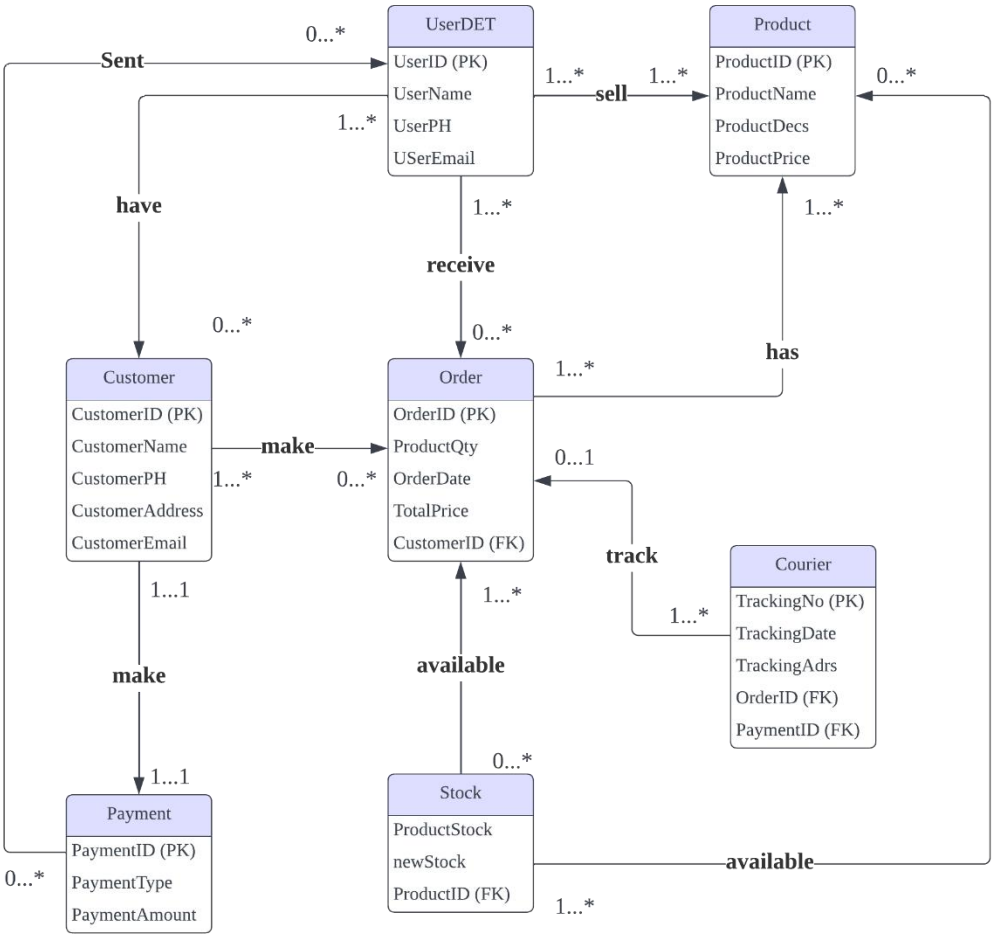


3.2.2 Enhanced ERD



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	Holde 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
```

```
----Courier----
CREATE TABLE Courier(
    TrackingNo varchar2(12),
    TrackingDate DATE NOT NULL,
    TrackingAdrs varchar2(50) NOT NULL,
    OrderID varchar2(10) NOT NULL,
    PaymentID varchar2(12) NOT NULL,
    CONSTRAINT TrackingNo_PK PRIMARY KEY (TrackingNo)
);

----Company----
CREATE TABLE Company(
    CompanyID varchar2(12),
    CompanyName varchar2(20) NOT NULL,
    CompanyLocation varchar2(50) NOT NULL,
    userID varchar2(16) NOT NULL,
    CONSTRAINT CompanyID_PK PRIMARY KEY (CompanyID)
);

----Categories----
CREATE TABLE Categories(
    CategoryID varchar2(12),
    CategoryName varchar2(20) NOT NULL,
    ProductID varchar2(10) NOT NULL,
    CONSTRAINT CategoryID_PK PRIMARY KEY (CategoryID)
);
```

6.2 Alter Table

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

```
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

```
----------------------Insert Value--------------------
----User-----
INSERT INTO UserDET
VALUES('U0001', 'Kong', 0123456, 'kong@gmail.com');
INSERT INTO UserDET
VALUES('U0002', 'Soo', 0234567, 'soo@gmail.com');
INSERT INTO UserDET
VALUES('U0003', 'Eng', 0345678, 'eng@gmail.com');
INSERT INTO UserDET
VALUES('U0004', 'Liow', 0456789, 'liow@gmail.com');
INSERT INTO UserDET
VALUES('U0005', 'Ee', 0567890, 'ee@gmail.com');
INSERT INTO UserDET
VALUES('U0006', 'James', 305462, 'james@gmail.com');
INSERT INTO UserDET
VALUES('U0007', 'Robert', 155115, 'robert@gmail.com');
INSERT INTO UserDET
VALUES('U0008', 'John', 146966, 'john@gmail.com');
INSERT INTO UserDET
VALUES('U0009', 'Michael', 144828, 'michael@gmail.com');
INSERT INTO UserDET
VALUES('U0010', 'David', 140379, 'david@gmail.com');
INSERT INTO UserDET
VALUES('U0011', 'William', 138599, 'william@gmail.com');
INSERT INTO UserDET
VALUES('U0012', 'Richard', 242395, 'richard@gmail.com');
INSERT INTO UserDET
VALUES('U0013', 'Joseph', 230004, 'joseph@gmail.com');
INSERT INTO UserDET
VALUES('U0014', 'Thomas', 213453, 'thomas@gmail.com');
INSERT INTO UserDET
VALUES('U0015', 'Charles', 203795, 'charles@gmail.com');
INSERT INTO UserDET
VALUES('U0016', 'Daniel', 190538, 'daniel@gmail.com');
INSERT INTO UserDET
VALUES('U0017', 'Matthew', 162015, 'matthew@gmail.com');
INSERT INTO UserDET
VALUES('U0018', 'Anthony', 014091, 'anthony@gmail.com');
INSERT INTO UserDET
VALUES('U0019', 'Mark', 013490, 'mark@gmail.com');
INSERT INTO UserDET
VALUES('U0020', 'Donald', 013089, 'donald@gmail.com');
----Customer----
INSERT INTO Customer
VALUES('C0001','Winki','1234567','Lot 1111','winki@gmail.com');
INSERT INTO Customer
VALUES('C0002','Keying','2345678','Lot 2222','keying@gmail.com');
INSERT INTO Customer
VALUES('C0003','Xiaoxuan','3456789','Lot 3333','xiaoxuan@gmail.com');
INSERT INTO Customer
VALUES('C0004','Yunyi','4567890','Lot 4444','yunyi@gmail.com');
INSERT INTO Customer
VALUES('C0005','ZhiXuan','5678901','Lot 5555','zhixuan@gmail.com');
INSERT INTO Customer
VALUES('C0006','Mary','426536','Lot 6666','mary@gmail.com');
INSERT INTO Customer
VALUES('C0007','Patricia','435762','Lot 7777','patricia@gmail.com');
INSERT INTO Customer
VALUES('C0008','Jennifer','435450','Lot 8888','jennifer@gmail.com');
INSERT INTO Customer
VALUES('C0009','Linda','434584','Lot 9999','linda@gmail.com');
INSERT INTO Customer
VALUES('C0010','Barbara','434584','Lot 1010','barbara@gmail.com');
```

```sql
INSERT INTO Customer
VALUES('C0011','Susan','110224','Lot 2020','susan@gmail.com');
INSERT INTO Customer
VALUES('C0012','Jessica','104763','Lot 3030','jessica@gmail.com');
INSERT INTO Customer
VALUES('C0013','Sarah','987732','Lot 4040','sarah@gmail.com');
INSERT INTO Customer
VALUES('C0014','Karen','986072','Lot 5050','karen@gmail.com');
INSERT INTO Customer
VALUES('C0015','Lisa','965306','Lot 6060','lisa@gmail.com');
INSERT INTO Customer
VALUES('C0016','Nancy','096058','Lot 7070','nancy@gmail.com');
INSERT INTO Customer
VALUES('C0017','Betty','088611','Lot 8080','betty@gmail.com');
INSERT INTO Customer
VALUES('C0018','Sandra','087357','Lot 9090','sandra@gmail.com');
INSERT INTO Customer
VALUES('C0019','Ashley','085892','Lot 1212','ashley@gmail.com');
INSERT INTO Customer
VALUES('C0020','Emily','083984','Lot 1313','emily@gmail.com');
```

```sql
----Product----
INSERT INTO Product
VALUES('P0001','pen','This is a pen made from US',35.50);
INSERT INTO Product
VALUES('P0002','shirt','This is a shirt made from Thailand',68.99);
INSERT INTO Product
VALUES('P0003','shoes','This is a shoes made from UK',400.00);
INSERT INTO Product
VALUES('P0004','Travel bag','This is a beg made from US',60.00);
INSERT INTO Product
VALUES('P0005','socks','This is a socks made from China',15.00);
INSERT INTO Product
VALUES('P0006','snack','This is a snack made from US',68.50);
INSERT INTO Product
VALUES('P0007','pant','This is a pant made from US',299.00);
INSERT INTO Product
VALUES('P0008','hoodie','This is a hoodie made from UK',79.00);
INSERT INTO Product
VALUES('P0009','snack','This is a snack made from Thailand',70.00);
INSERT INTO Product
VALUES('P0010','phone casing','This is a phone casing made from China',115.00);
```

```sql
INSERT INTO Product
VALUES('P0011','toy','This is a toy made from US',15.50);
INSERT INTO Product
VALUES('P0012','Matcha','This is a Matcha made from Thailand',99.99);
INSERT INTO Product
VALUES('P0013','shirt','This is a shirt made from China',149.00);
INSERT INTO Product
VALUES('P0014','pen','This is a pen made from US',55.00);
INSERT INTO Product
VALUES('P0015','glasses','This is a glasses made from UK',29.00);
INSERT INTO Product
VALUES('P0016','mouse','This is a mouse made from US',76.50);
INSERT INTO Product
VALUES('P0017','skirt','This is a skirt made from China',20.00);
INSERT INTO Product
VALUES('P0018','bottle','This is a bottle made from China',29.99);
INSERT INTO Product
VALUES('P0019','watch','This is a watch made from UK',300.00);
INSERT INTO Product
VALUES('P0020','tissue','This is a tissue made from Thailand',140.99);
```

```
241    ----Orders----
242    INSERT INTO Orders
243    VALUES('B0001',1,SYSDATE,15.00,'C0001');
244    INSERT INTO Orders
245    VALUES('B0002',1,SYSDATE,400.00,'C0004');
246    INSERT INTO Orders
247    VALUES('B0003',3,SYSDATE,180.00,'C0003');
248    INSERT INTO Orders
249    VALUES('B0004',1,SYSDATE,688.99,'C0002');
250    INSERT INTO Orders
251    VALUES('B0005',2,SYSDATE,70.00,'C0005');
252    INSERT INTO Orders
253    VALUES('B0006',2,SYSDATE,79.00,'C0006');
254    INSERT INTO Orders
255    VALUES('B0007',2,SYSDATE,299.00,'C0007');
256    INSERT INTO Orders
257    VALUES('B0008',3,SYSDATE,245.00,'C0008');
258    INSERT INTO Orders
259    VALUES('B0009',1,SYSDATE,15.00,'C0009');
260    INSERT INTO Orders
261    VALUES('B0010',1,SYSDATE,115.00,'C0010');
262    INSERT INTO Orders
263    VALUES('B0011',2,SYSDATE,68.00,'C0011');
264    INSERT INTO Orders
265    VALUES('B0012',3,SYSDATE,55.00,'C0012');
266    INSERT INTO Orders
267    VALUES('B0013',1,SYSDATE,29.00,'C0013');
268    INSERT INTO Orders
269    VALUES('B0014',3,SYSDATE,329.00,'C0014');
270    INSERT INTO Orders
271    VALUES('B0015',1,SYSDATE,149.00,'C0015');
272    INSERT INTO Orders
273    VALUES('B0016',1,SYSDATE,76.50,'C0016');
274    INSERT INTO Orders
275    VALUES('B0017',2,SYSDATE,20.00,'C0017');
276    INSERT INTO Orders
277    VALUES('B0018',3,SYSDATE,29.99,'C0018');
278    INSERT INTO Orders
279    VALUES('B0019',2,SYSDATE,300.00,'C0019');
280    INSERT INTO Orders
281    VALUES('B0020',1,SYSDATE,140.99,'C0020');
284    ----Stock----
285    INSERT INTO Stock
286    VALUES(2,200,'P0001');
287    INSERT INTO Stock
288    VALUES(5,20,'P0002');
289    INSERT INTO Stock
290    VALUES(10,400,'P0003');
291    INSERT INTO Stock
292    VALUES(10,87,'P0004');
293    INSERT INTO Stock
294    VALUES(68,69,'P0005');
295    INSERT INTO Stock
296    VALUES(5,250,'P0006');
297    INSERT INTO Stock
298    VALUES(8,56,'P0007');
299    INSERT INTO Stock
300    VALUES(10,320,'P0008');
301    INSERT INTO Stock
302    VALUES(3,33,'P0009');
303    INSERT INTO Stock
304    VALUES(68,100,'P0010');
```

```
INSERT INTO Stock
VALUES(37,40,'P0011');
INSERT INTO Stock
VALUES(58,85,'P0012');
INSERT INTO Stock
VALUES(10,123,'P0013');
INSERT INTO Stock
VALUES(22,69,'P0014');
INSERT INTO Stock
VALUES(15,150,'P0015');
INSERT INTO Stock
VALUES(5,31,'P0016');
INSERT INTO Stock
VALUES(21,77,'P0017');
INSERT INTO Stock
VALUES(25,12,'P0018');
INSERT INTO Stock
VALUES(98,99,'P0019');
INSERT INTO Stock
VALUES(6,80,'P0020');
```

```
----Payment----
INSERT INTO Payment
VALUES('X0001',1,15.00);
INSERT INTO Payment
VALUES('X0002',1,400.00);
INSERT INTO Payment
VALUES('X0003',3,180.00);
INSERT INTO Payment
VALUES('X0004',1,688.99);
INSERT INTO Payment
VALUES('X0005',2,70.00);
INSERT INTO Payment
VALUES('X0006',2,79.00);
INSERT INTO Payment
VALUES('X0007',2,299.00);
INSERT INTO Payment
VALUES('X0008',3,245.00);
INSERT INTO Payment
VALUES('X0009',1,15.00);
INSERT INTO Payment
VALUES('X0010',1,115.00);
INSERT INTO Payment
VALUES('X0011',2,68.00);
INSERT INTO Payment
VALUES('X0012',3,55.00);
INSERT INTO Payment
VALUES('X0013',1,29.00);
INSERT INTO Payment
VALUES('X0014',3,329.00);
INSERT INTO Payment
VALUES('X0015',1,149.00);
INSERT INTO Payment
VALUES('X0016',1,76.50);
INSERT INTO Payment
VALUES('X0017',2,20.00);
INSERT INTO Payment
VALUES('X0018',3,29.99);
INSERT INTO Payment
VALUES('X0019',2,300.00);
INSERT INTO Payment
VALUES('X0020',1,140.99);
```

```sql
----Company----
INSERT INTO Company
VALUES('CP0001','EdenShop','Lot Perak','U0001');
INSERT INTO Company
VALUES('CP0002','Vrify','Lot Selangor','U0002');
INSERT INTO Company
VALUES('CP0003','Sakei','Lot Johor','U0003');
INSERT INTO Company
VALUES('CP0004','Beggy','Lot Kedah','U0004');
INSERT INTO Company
VALUES('CP0005','Shopy','Lot Pahang','U0005');
INSERT INTO Company
VALUES('CP0006','BodyCandy','Lot Penang','U0006');
INSERT INTO Company
VALUES('CP0007','Rebag','Lot Terengganu','U0007');
INSERT INTO Company
VALUES('CP0008','Ridge','Lot Melaka','U0008');
INSERT INTO Company
VALUES('CP0009','VRAI','Lot Perlis','U0009');
INSERT INTO Company
VALUES('CP0010','Alex And Ani','Lot Negeri Sembilan','U0010');
INSERT INTO Company
VALUES('CP0011','Crown Caliber','Lot Kuala Lumpur','U0011');
INSERT INTO Company
VALUES('CP0012','PrincessPolly','Lot Sabah','U0012');
INSERT INTO Company
VALUES('CP0013','Diatone','Lot Sarawak','U0013');
INSERT INTO Company
VALUES('CP0014','Naturool','Lot Kedah','U0014');
INSERT INTO Company
VALUES('CP0015','Kinetics','Lot Penang','U0015');
INSERT INTO Company
VALUES('CP0016','MeltCosmetics','Lot Perak','U0016');
INSERT INTO Company
VALUES('CP0017','DailyClub','Lot Selangor','U0017');
INSERT INTO Company
VALUES('CP0018','Haute','Lot Johor','U0018');
INSERT INTO Company
VALUES('CP0019','SassyStyles','Lot Kuala Lumpur','U0019');
INSERT INTO Company
VALUES('CP0020','Trendsetter','Lot Perak','U0020');
----Categories----
INSERT INTO Categories
VALUES('G0001','Stationery','P0001');
INSERT INTO Categories
VALUES('G0002','Shirt','P0002');
INSERT INTO Categories
VALUES('G0003','Sporting','P0003');
INSERT INTO Categories
VALUES('G0004','Sporting Accessories','P0005');
INSERT INTO Categories
VALUES('G0005','beg','P0004');
INSERT INTO Categories
VALUES('G0006','Snack','P0006');
INSERT INTO Categories
VALUES('G0007','Pant','P0007');
INSERT INTO Categories
VALUES('G0008','Hoodie','P0008');
INSERT INTO Categories
VALUES('G0009','Snack','P0009');
INSERT INTO Categories
VALUES('G0010','Phone Accessories','P0010');
```

```
INSERT INTO Categories
VALUES('G0011','Toy','P0011');
INSERT INTO Categories
VALUES('G0012','Matcha','P0012');
INSERT INTO Categories
VALUES('G0013','Shirt','P0013');
INSERT INTO Categories
VALUES('G0014','Stationery','P0014');
INSERT INTO Categories
VALUES('G0015','Glasses','P0015');
INSERT INTO Categories
VALUES('G0016','Mouse','P0016');
INSERT INTO Categories
VALUES('G0017','Skirt','P0017');
INSERT INTO Categories
VALUES('G0018','Bottle','P0018');
INSERT INTO Categories
VALUES('G0019','Watch','P0019');
INSERT INTO Categories
VALUES('G0020','Daily','P0020');
```

```
----Courier----
INSERT INTO Courier
VALUES('T0001',SYSDATE,'Lot 1111','B0001','X0001');
INSERT INTO Courier
VALUES('T0002',SYSDATE,'Lot 4444','B0002','X0002');
INSERT INTO Courier
VALUES('T0003',SYSDATE,'Lot 3333','B0003','X0003');
INSERT INTO Courier
VALUES('T0004',SYSDATE,'Lot 2222','B0004','X0004');
INSERT INTO Courier
VALUES('T0005',SYSDATE,'Lot 5555','B0005','X0005');
INSERT INTO Courier
VALUES('T0006',SYSDATE,'Lot 6666','B0006','X0006');
INSERT INTO Courier
VALUES('T0007',SYSDATE,'Lot 7777','B0007','X0007');
INSERT INTO Courier
VALUES('T0008',SYSDATE,'Lot 8888','B0008','X0008');
INSERT INTO Courier
VALUES('T0009',SYSDATE,'Lot 9999','B0009','X0009');
INSERT INTO Courier
VALUES('T00010',SYSDATE,'Lot 1010','B0010','X0010');
INSERT INTO Courier
VALUES('T0011',SYSDATE,'Lot 2020','B0011','X0011');
INSERT INTO Courier
VALUES('T0012',SYSDATE,'Lot 3030','B0012','X0012');
INSERT INTO Courier
VALUES('T0013',SYSDATE,'Lot 4040','B0013','X0013');
INSERT INTO Courier
VALUES('T0014',SYSDATE,'Lot 5050','B0014','X0014');
INSERT INTO Courier
VALUES('T0015',SYSDATE,'Lot 6060','B0015','X0015');
INSERT INTO Courier
VALUES('T0016',SYSDATE,'Lot 7070','B0016','X0016');
INSERT INTO Courier
VALUES('T0017',SYSDATE,'Lot 8080','B0017','X0017');
INSERT INTO Courier
VALUES('T0018',SYSDATE,'Lot 9090','B0018','X0018');
INSERT INTO Courier
VALUES('T0019',SYSDATE,'Lot 1212','B0019','X0019');
INSERT INTO Courier
VALUES('T0020',SYSDATE,'Lot 1313','B0020','X0020');
```

6.4 Testing

#Test 1 Display User

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

Results | Explain | Describe | Saved SQL | History

USERID	USERNAME	USERPH	USEREMAIL
U0001	Kong	123456	kong@gmail.com
U0016	Daniel	190538	daniel@gmail.com
U0002	Soo	234567	soo@gmail.com
U0013	Joseph	230004	joseph@gmail.com
U0003	Eng	345678	eng@gmail.com
U0004	Liow	456789	liow@gmail.com
U0005	Ee	567890	ee@gmail.com
U0006	James	305462	james@gmail.com
U0008	John	146966	john@gmail.com
U0011	William	138599	william@gmail.com

More than 10 rows available. Increase rows selector to view more rows.

#Test 2 Display customer's name and id

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

Results | Explain | Describe | Saved SQL | History

CUSTOMERNAME	CUSTOMERID
Winki	C0001
Jennifer	C0008
Nancy	C0016
Betty	C0017
Lisa	C0015
Jessica	C0012
Ashley	C0019
Keying	C0002
Xiaoxuan	C0003
Patricia	C0007

More than 10 rows available. Increase rows selector to view more rows.

#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;
```

Results | Explain | Describe | Saved SQL | History

USERID	USERNAME	USERPH	USEREMAIL
U0001	Kong	123456	kong@gmail.com
U0016	Daniel	190538	daniel@gmail.com
U0002	Soo	234567	soo@gmail.com
U0013	Joseph	230004	joseph@gmail.com
U0003	Eng	345678	eng@gmail.com
U0004	Liow	456789	liow@gmail.com
U0005	Ee	567890	ee@gmail.com
U0006	James	305462	james@gmail.com
U0008	John	146966	john@gmail.com
U0011	William	138599	william@gmail.com

More than 10 rows available. Increase rows selector to view more rows.

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

```
SELECT * from Product
WHERE ProductPrice between 50.00 and 300.00;
```

Results | Explain | Describe | Saved SQL | History

PRODUCTID	PRODUCTNAME	PRODUCTDECS	PRODUCTPRICE
P0009	snack	This is a snack made from Thailand	70
P0012	Matcha	This is a Matcha made from Thailand	99.99
P0008	hoodie	This is a hoodie made from UK	79
P0002	shirt	This is a shirt made from Thailand	68.99
P0004	Travel bag	This is a beg made from US	60
P0006	snack	This is a snack made from US	68.5
P0014	pen	This is a pen made from US	55
P0007	pant	This is a pant made from US	299
P0013	shirt	This is a shirt made from China	149
P0019	watch	This is a watch made from UK	300

More than 10 rows available. Increase rows selector to view more rows.

#Test 5 Display the product in respective categories

```
SELECT P.ProductID AS "Product ID", P.ProductName AS "Product Name", P.ProductPrice AS "Product Price", C.CategoryName AS "Category Name"
FROM(
    Product P
    JOIN Categories C ON P.ProductID = C.ProductID
)
ORDER BY P.ProductPrice;
```

Results | Explain | Describe | Saved SQL | History

PRODUCTID	PRODUCTNAME	PRODUCTDECS	PRODUCTPRICE
P0009	snack	This is a snack made from Thailand	70
P0012	Matcha	This is a Matcha made from Thailand	99.99
P0008	hoodie	This is a hoodie made from UK	79
P0002	shirt	This is a shirt made from Thailand	68.99
P0004	Travel bag	This is a beg made from US	60
P0006	snack	This is a snack made from US	68.5
P0014	pen	This is a pen made from US	55
P0007	pant	This is a pant made from US	299
P0013	shirt	This is a shirt made from China	149
P0019	watch	This is a watch made from UK	300

More than 10 rows available. Increase rows selector to view more rows.

#Test 6 Update Stock

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

Results | Explain | Describe | Saved SQL | History

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
P0006	5	250	255
P0007	8	56	64
P0008	10	320	330
P0009	3	33	36
P0010	68	100	168

More than 10 rows available. Increase rows selector to view more rows.

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

```
SELECT * from Orders
WHERE TotalPrice between 10.00 and 200.00;
```

Results | Explain | Describe | Saved SQL | History

ORDERID	PRODUCTQTY	ORDERDATE	TOTALPRICE	CUSTOMERID
B0013	1	01/19/2024	29	C0013
B0001	1	01/19/2024	15	C0001
B0011	2	01/19/2024	68	C0011
B0006	2	01/19/2024	79	C0006
B0003	3	01/19/2024	180	C0003
B0005	2	01/19/2024	70	C0005
B0015	1	01/19/2024	149	C0015
B0016	1	01/20/2024	76.5	C0016
B0017	2	01/20/2024	20	C0017
B0018	3	01/20/2024	29.99	C0018

More than 10 rows available. Increase rows selector to view more rows.

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

```
---Display the company name and location of company for each users---
SELECT U.userName AS "User Name",
C.CompanyName AS "Comapny Name",
C.CompanyID AS "Company ID",
C.CompanyLocation AS "The Location of Company"
FROM(
    Company C
    JOIN UserDET U ON C.UserID = U.UserID
)
ORDER BY C.CompanyID;
```

User Name	Comapny Name	Company ID	The Location of Company
Kong	EdenShop	CP0001	Lot Perak
Soo	Vrify	CP0002	Lot Selangor
Eng	Sakei	CP0003	Lot Johor
Liow	Beggy	CP0004	Lot Kedah
Ee	Shopy	CP0005	Lot Pahang
James	BodyCandy	CP0006	Lot Penang
Robert	Rebag	CP0007	Lot Terengganu
John	Rider	CP0008	Lot Melaka

2024 Oracle Live SQL 23.4.2 running Oracle Database 19c EE

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.