ASSIGNMENT 2 FRONT SHEET

Qualification	**TEC Level 5 HND Diploma in Computing**		
Unit number and title	**Unit 04: Database Design & Development**		
Submission date	October, 31st 2022	**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	
Student Name	Tran Nguyet Can	**Student ID**	GCC210146
Class	GCC1002	**Assessor name**	Nguyen Hung Dung
Student declaration I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.			
		Student's signature	

Grading grid

P2	P3	P4	P5	M2	M3	M4	M5	D2	D3

☼ Summative Feedback:	☼ Resubmission Feedback:	
Grade:	**Assessor Signature:**	**Date:**
Signature & Date:		

Submission Format:

Format:

- This assignment is an Individual assignment and specifically including 2 documents:
  (1) sql file of your code and represent your code to your tutor
  (2) *a report document*
- You must use font Calibri size 12, set number of the pages and use multiple line spacing at 1.3. Margins must be: left: 1.25 cm; right: 1 cm; top: 1 cm and bottom: 1 cm. The reference follows Harvard referencing system. The recommended word limit is 2.000-2.500 words. You will not be penalized for exceeding the total word limit. The cover page of the report has to be the Assignment front sheet 2.

Submission

- Students are compulsory to submit the assignment in due date and in a way requested by the Tutor.
- The form of submission will be a soft copy posted on http://cms.greenwich.edu.vn/.
- Remember to convert the word file into PDF file before the submission on CMS.

Note:

- The individual Assignment *must* be your own work, and not copied by or from another student.
- If you use ideas, quotes or data (such as diagrams) from books, journals or other sources, you must reference your sources, using the Harvard style.
- Make sure that you understand and follow the guidelines to avoid plagiarism. Failure to comply this requirement will result in a failed assignment.

Unit Learning Outcomes:

LO2 Develop a fully functional relational database system, based on an existing system design.

LO3 Test the system against user and system requirements.

LO4 Produce technical and user documentation

Assignment Brief and Guidance:

Assignment scenario

You are employed as a Database Developer for a large IT consultancy company. The company has been approached by **FPT Shop** which is expanding due to the growth of the number of stores. **FPT Shop** is currently facing difficulties in dealing with managing the database from all shops on over country. It decided to develop a new

database so that: **users can register with their phone numbers as IDs and order or rate, comment for their bought devices, shop managers can take care for their stores and director board can view all data from all shops.**

You are tasked to select one of those systems to develop database for FPT Shop. Your tasks are to:

- Work with FPT Shop to find out about current requirements for each system
- Analyse the requirements and produce clear statements of user and system requirements.
- Design a relational database system using appropriate design tools and techniques
- Develop a fully functional relational database system, based on an existing system design.
- Test the system against user and system requirements.
- Produce technical and user documentation

Part 2 (Assignment 2)

Once the designs have been accepted by your manager you have been asked to:

Develop the database system using evidence of user interface, output and data validations and querying across multiple tables.

You want to include more than just the basics so you will implement a fully functional database system which will include system security and database maintenance features.

You have decided to implement a query language into the relational database system. The developed system will be demonstrated to your manager.

Your manager has asked you to include in the report:

(1) Assessing whether meaningful data has been extracted through the use of query tools to produce appropriate management information.

(2) Evaluating the effectiveness of the database solution in relation to user and system requirements, and suggest improvements.

(3) Once the system has been developed, you will test the system and your manager will complete a witness statement indicating how your tests are performing against user and system requirements.

(4) You will produce a brief report assessing the effectiveness of the testing, including an explanation of the choice of test data used.

(5) Lastly you will produce technical and user documentation which will be given to the company.

You want to provide some graphical representations for ease of reference in the technical guide, so you have decided to produce a technical and user documentation for a fully functional system, including diagrams showing movement of data through the system, and flowcharts describing how the system works.

Learning Outcomes and Assessment Criteria (Assignment 2):			
Learning Outcome	Pass	Merit	Distinction

LO2	**P2** Develop the database system with evidence of user interface, output and data validations, and querying across multiple tables. **P3** Implement a query language into the relational database system.	**M2** Implement a fully functional database system which includes system security and database maintenance. **M3** Assess whether meaningful data has been extracted through the use of query tools to produce appropriate management information.	**D2** Evaluate the effectiveness of the database solution in relation to user and system requirements, and suggest improvements.
LO3	**P4** Test the system against user and system requirements.	**M4** Assess the effectiveness of the testing, including an explanation of the choice of test data used.	
LO4	**P5** Produce technical and user documentation.	**M5** Produce technical and user documentation for a fully functional system, including ER Diagram and normalization statements and describing how the system works.	**D3** Assess any future improvements that may be required to ensure the continued effectiveness of the database system.

Table of Contents

Assignment 2

Chapter 1: Implementation

1. Code snippets to create each table

1.1 Create database “SELL_MANAGE1”

```
Create database SELL_MANAGE1
```

1.2 Create Table Customer

```
Create table Customer
(
	CustomerID char(10) primary key NOT NULL,
	CustomerName varchar(50) NOT NULL,
	Phone nchar(15) NOT NULL,
	Address varchar(100) NOT NULL
)
```

1.3 Create Table Orders

```
Create Table Orders
(
	OrderID int primary key NOT NULL,
	CustomerID char(10) references Customer(CustomerID) NOT NULL,
	StaffID int references Staff(StaffID) NOT NULL,
	OrderDate datetime NOT NULL,
	Delivery datetime NOT NULL
)
```

1.4 Create Table Staff

```
Create Table Staff
(
	StaffID int primary key NOT NULL,
	StaffName varchar(50) NOT NULL,
	Position varchar NOT NULL
)
```

1.5 Create Table OrderDetail

```
Create Table OrderDetail
(
	orderID int references Orders(orderID) NOT NULL,
	ProductID char(10) references Product(ProductID) NOT NULL,
	buying_qty int NOT NULL,
	Primary Key (orderID, ProductID)
)
```

1.6 Create Table Product

```
Create Table Product
(
	ProductID char(10) Primary key NOT NULL,
	ProductName varchar(50) NOT NULL,
	Stock int NOT NULL,
	Description varchar(20) NOT NULL,
	CategoryID char(10) references Category(CategoryID),
	Price int NOT NULL
)
```

1.7 Create Table Feedback

```
Create Table Feedback
(
	FeedbackID int primary key NOT NULL,
	FeedbackDate varchar(10) NOT NULL,
	Content varchar(200) NOT NULL,
	CustomerID char(10) NOT NULL references Customer(CustomerID),
	ProductID char(10) NOT NULL references Product(ProductID)
)
```

1.8 Create Table Category

```
Create Table Category
(
	CategoryID char(10) Primary key NOT NULL,
	CategoryName varchar(50) NOT NULL
)
```

2. Code snippets to insert some sample data for each table

2.1 Customer Table

```
Insert into Customer Values (01, 'Tran Minh Hoang', 0987654311, 'District 1, HCM City')
Insert into Customer Values (02, 'Tran Moc Phong', 0833569320, 'Thu Duc, HCM City')
Insert into Customer Values (03, 'Tran Lam Anh', 0137899123, 'District 10, HCM City')
Insert into Customer Values (04, 'Le Thanh Ngoan', 0909191230, 'OMon District , HCM City')
Insert into Customer Values (05, 'Nguyen Thi Hien', 0883362713, 'Rach Gia, Kien Giang')
Insert into Customer Values (06, 'Nguyen Hoang Khang', 0732368971, 'Sa Dec, Dong Thap')
```

106 %

Results | Messages

	CustomerID	CustomerName	Phone	Address
1	1	Tran Minh Hoang	987654311	District 1, HCM City
2	2	Tran Moc Phong	833569320	Thu Duc, HCM City
3	3	Tran Lam Anh	137899123	District 10, HCM City
4	4	Le Thanh Ngoan	909191230	OMon District , HCM City
5	5	Nguyen Thi Hien	883362713	Rach Gia, Kien Giang

2.2 Orders Table

```
Insert into Orders values (112, 02, 100, '2019/10/22', '2019/10/30')
Insert into Orders values (113, 03, 103, '2020/03/12', '2020/03/15')
Insert into Orders values (114, 02, 101, '2020/04/26', '2020/04/30')
Insert into Orders values (115, 01, 105, '2021/01/05', '2021/01/15')
Insert into Orders values (116, 04, 104, '2022/02/02', '2022/02/10')
Insert into Orders values (117, 05, 102, '2022/01/12', '2022/01/22')
```

106 %

Results | Messages

	OrderID	CustomerID	StaffID	OrderDate	Delivery
1	112	2	100	2019-10-22 00:00:00.000	2019-10-30 00:00:00.000
2	113	3	103	2020-03-12 00:00:00.000	2020-03-15 00:00:00.000
3	114	2	101	2020-04-26 00:00:00.000	2020-04-30 00:00:00.000
4	115	1	105	2021-01-05 00:00:00.000	2021-01-15 00:00:00.000
5	116	4	104	2022-02-02 00:00:00.000	2022-02-10 00:00:00.000
6	117	5	102	2022-01-12 00:00:00.000	2022-01-22 00:00:00.000

2.3 Staff Table

```
Insert into Staff values (100, 'Vuong Quang Minh', 'Cashier Staff')
Insert into Staff values (101, 'Ngo Huynh Phuc', 'Marketing Staff')
Insert into Staff values (102, 'Ta Minh The', 'Telesale Staff')
Insert into Staff values (103, 'Nguyen Ha Nhat Linh', 'Sales Staff')
Insert into Staff values (104, 'Huynh Thai Thinh', 'Business Staff')
Insert into Staff values (105, 'Vo Tan Nghia', 'Warehouse Staff')
```

81 %

Results | Messages

	StaffID	StaffName	Position
1	100	Vuong Quang Minh	Cashier Staff
2	101	Ngo Huynh Phuc	Marketing Staff
3	102	Ta Minh The	Telesale Staff
4	103	Nguyen Ha Nhat ...	Sales Staff
5	104	Huynh Thai Thinh	Business Staff
6	105	Vo Tan Nghia	Warehouse ...

2.4 OrderDetail Table

```
56  Insert into OrderDetail values (112,'P91', 2)
57  Insert into OrderDetail values (113,'P92', 2)
58  Insert into OrderDetail values (114,'P93', 3)
59  Insert into OrderDetail values (115,'P94', 1)
60  Insert into OrderDetail values (116,'P95', 1)
61  Insert into OrderDetail values (117,'P96', 5)
```

106 %

Results | Messages

	orderID	ProductID	buying_qty
1	112	P91	2
2	113	P92	2
3	114	P93	3
4	115	P94	1
5	116	P95	1
6	117	P96	5

2.5 Product Table

```
78  Insert into Product values ('P91', 'Laptop HP Probook', 11, 'Intel Core I5 6200u', 'HP520', 6000000)
79  Insert into Product values ('P92', 'MacbookPro 2013', 9, 'Intel Core I5 2.5ghz', 'MB13', 8500000)
80  Insert into Product values ('P93', 'Laptop Acer E5_573G', 10, 'Intel Core I5 5200u', 'A573', 6500000)
81  Insert into Product values ('P94', 'Laptop Gaming Dell Inspiron 5577', 2, 'Intel Core I7 7700HQ', 'GM5577', 1380000)
82  Insert into Product values ('P95', 'Laptop Dell Vostro 3500', 12, 'Intel Core I5 1135G7', 'DV35', 1300000)
83  Insert into Product values ('P96', 'Laptop Acer Nitro5', 5, 'Intel Core I7 11800H', 'A517', 20000000)
84
```

81 %

Results | Messages

	ProductID	ProductName	Stock	Description	CategoryID	Price
1	P91	Laptop HP Probook	11	Intel Core I5 6200u	HP520	6000000
2	P92	MacbookPro 2013	9	Intel Core I5 2.5ghz	MB13	8500000
3	P93	Laptop Acer E5_573G	10	Intel Core I5 5200u	A573	6500000
4	P94	Laptop Gaming Dell Inspiron 5577	2	Intel Core I7 7700HQ	GM5577	1380000
5	P95	Laptop Dell Vostro 3500	12	Intel Core I5 1135G7	DV35	1300000
6	P96	Laptop Acer Nitro5	5	Intel Core I7 11800H	A517	20000000

2.6 Feedback Table

```
93  Insert into Feedback values (12, '2019/11/01', 'pretty good', 02, 'P91')
94  Insert into Feedback values (13, '2020/03/17', 'great', 03, 'P92')
95  Insert into Feedback values (14, '2020/05/01', 'quality', 02, 'P93')
96  Insert into Feedback values (15, '2021/01/17','smooth game play', 01, 'P94')
97  Insert into Feedback values (16, '2022/02/15', 'Run the program fast', 04, 'P95')
98  Insert into Feedback values (17, '2022/01/25', 'Excellent', 05, 'P96')
```

81 %

Results | Messages

	FeedbackID	FeedbackDate	Content	CustomerID	ProductID
1	12	2019/11/01	pretty good	2	P91
2	13	2020/03/17	great	3	P92
3	14	2020/05/01	quality	2	P93
4	15	2021/01/17	smooth game play	1	P94
5	16	2022/02/15	Run the program fast	4	P95
6	17	2022/01/25	Excellent	5	P96

2.7 Category Table

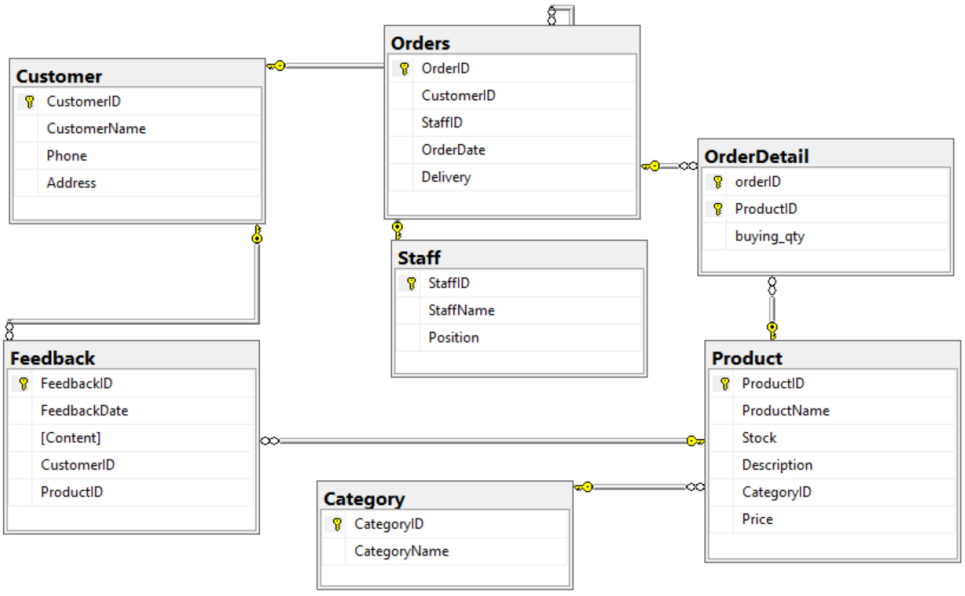
```
105 Insert into Category values ('HP520', 'Onboard HD Graphics 520')
106 Insert into Category values ('MB13', 'Onboard HD Graphics HD 4000')
107 Insert into Category values ('A573', 'GeForce GT920m')
108 Insert into Category values ('GM5577', 'GeForce GTX1050')
109 Insert into Category values ('DV35', 'GeForce MX230')
110 Insert into Category values ('A517', 'GeForce RTX3050')
```

81 %

Results | Messages

	CategoryID	CategoryName
1	A517	GeForce RTX3050
2	A573	GeForce GT920m
3	DV35	GeForce MX230
4	GM5577	GeForce GTX1050
5	HP520	Onboard HD Graphics 520
6	MB13	Onboard HD Graphics HD 4000

3. Generated Database Diagram of your Implementation



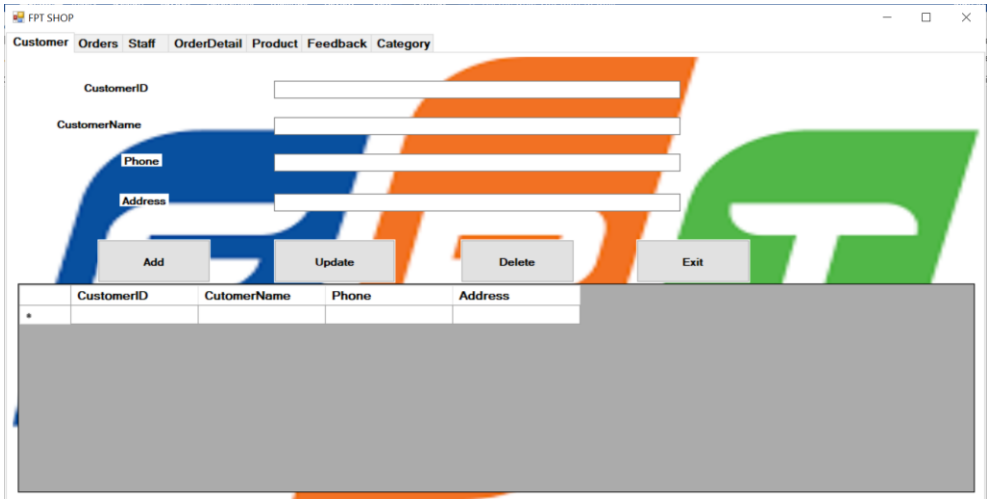
4. Explanations about any changes comparing to your design

In ASM2 there was a change, I removed "OrderTotal" from the Orders table because the value of the item when ordered and when it is sold can actually be different, I do not think that calculating the total value of items when ordering without a specific price

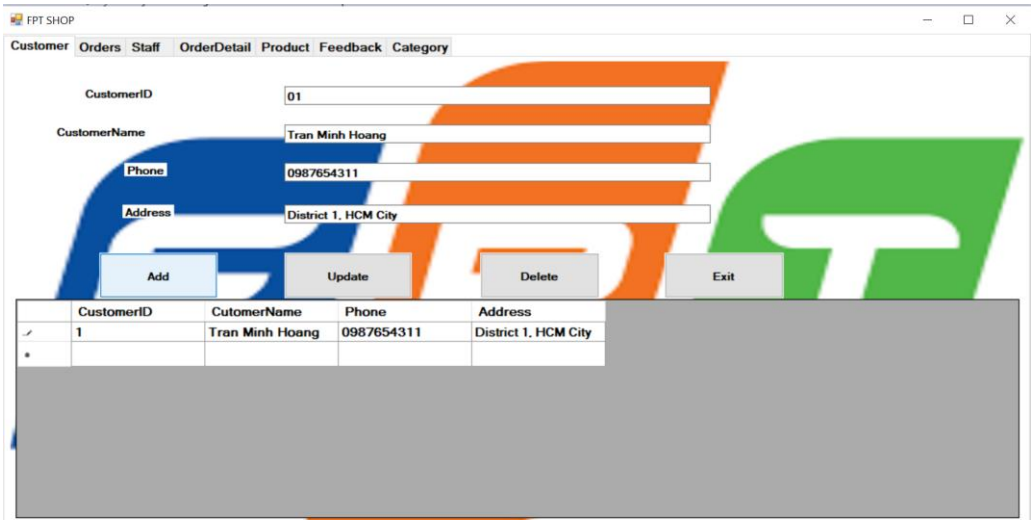
Chapter 2: User Interfaces and Queries

1. Customer Interface

1.1 Add Function



Users only need to fill in the customer information in the customer information interface, then press the Add button, the customer information will be added to the database.

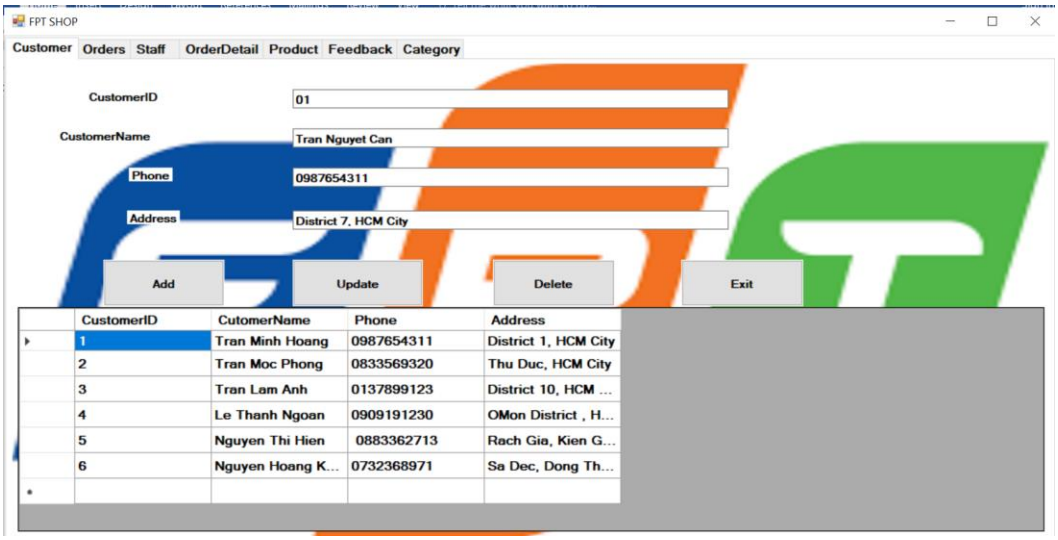


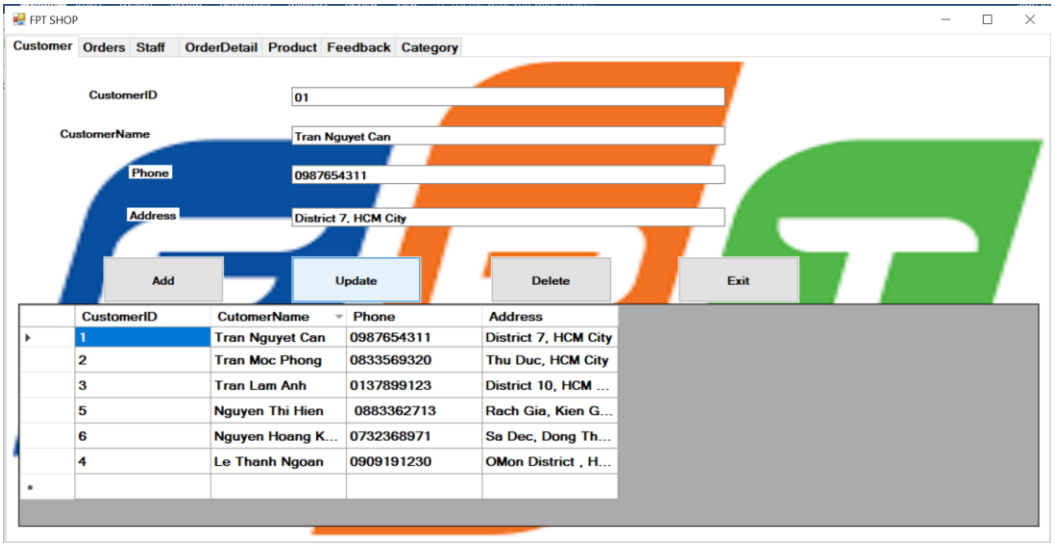
The following query to Add:

```
CREATE PROC Insert_Customer
@CusID char(10), @CusName varchar(30), @Phone nchar(15), @Address varchar(100)
AS
	INSERT INTO CUSTOMER VALUES (@CusID, @CusName, @Phone, @Address)

	Exec Insert_Customer '01', 'Tran Minh Hoang', '0987654311', 'District 1, HCM City'
```

1.2 Update Function

If the user wants to update the customer information, he or she must first input the customer information box and then the updated information. The user then selects the Update button.



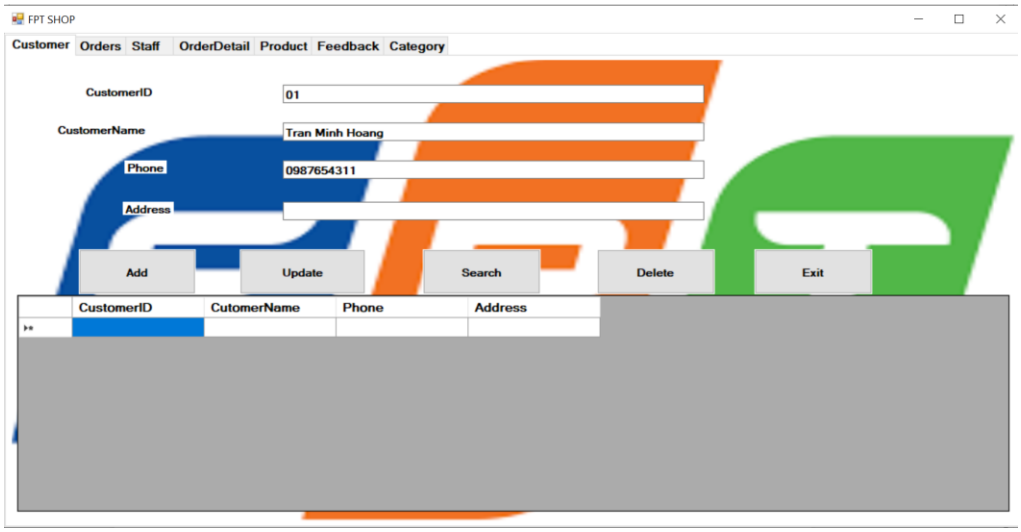


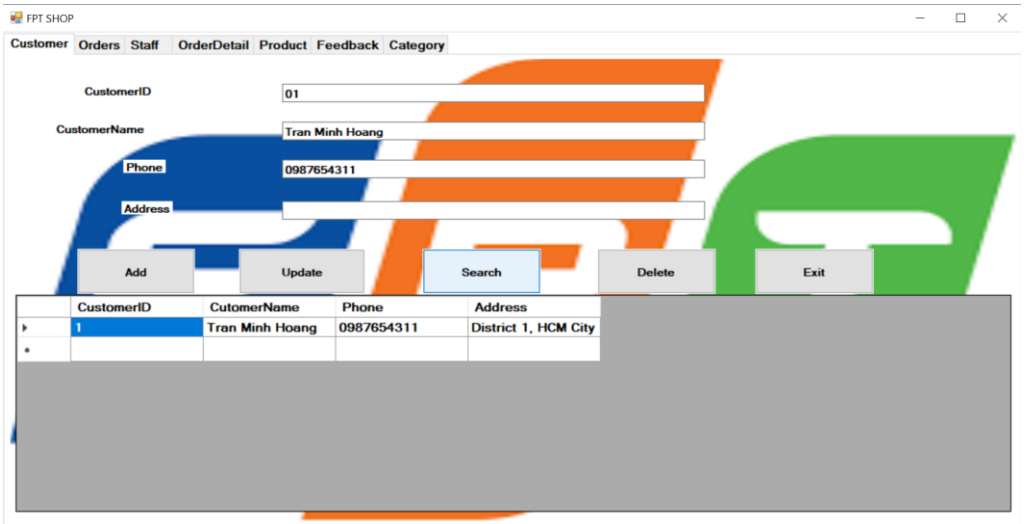
The following query to update:

```
CREATE PROC UpDateCus
(@CusID char(10), @CusName varchar(50), @Address varchar(100))
AS
	BEGIN
		UPDATE [dbo].[Customer] SET CustomerName = @CusName, Address = @Address WHERE @CusID
= CustomerID
	END

EXEC dbo.UpDateCus 01, 'Tran Nguyet Can','District 7, HCM City'
```

1.3 Search Funtion

Whenever a user wants to search the data for any customer information. The user merely needs to enter the customer's information, including the customer ID, name, and phone number, in order to conduct a search then click Search button.





When a user searches for a Customer in a table called Customer, I create procedure search information about that Customer from the data.

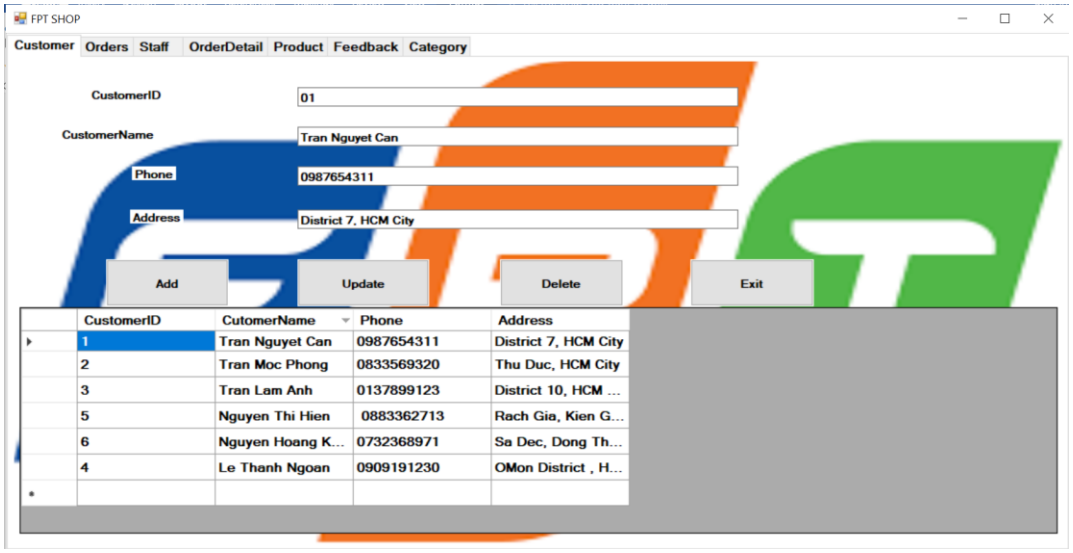
The following query to Search:

```
CREATE PROC SearchCus
@CusID char(10),@CusName varchar(50),@Phone nchar(15)
AS
	BEGIN
```

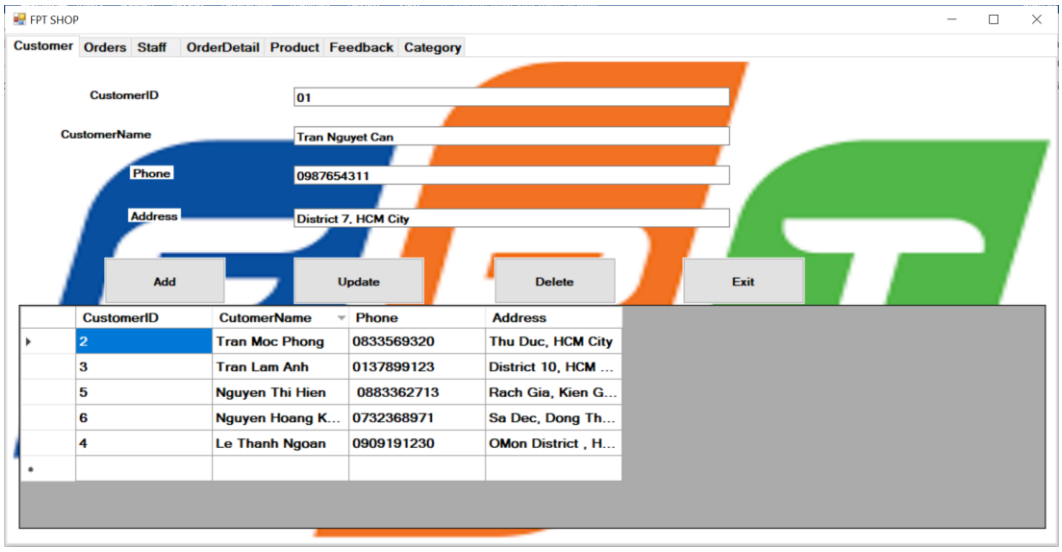
```
        SELECT * FROM Customer WHERE CustomerID = @CusID and CustomerName = @CusName and
Phone = @Phone
    END

    EXEC SearchCus 01, 'Tran Minh Hoang', 0987654311
```

1.4 Delete Function

When the user needs to delete customer information from the system, the user must click on the ID of the customer that the user wants to delete in the data and then press the Delete button.



I have created a trigger to delete data from the database when a user deletes a certain customer. When a user deletes any customer data on the Customer table, the user deletes that customer data in other tables as well.



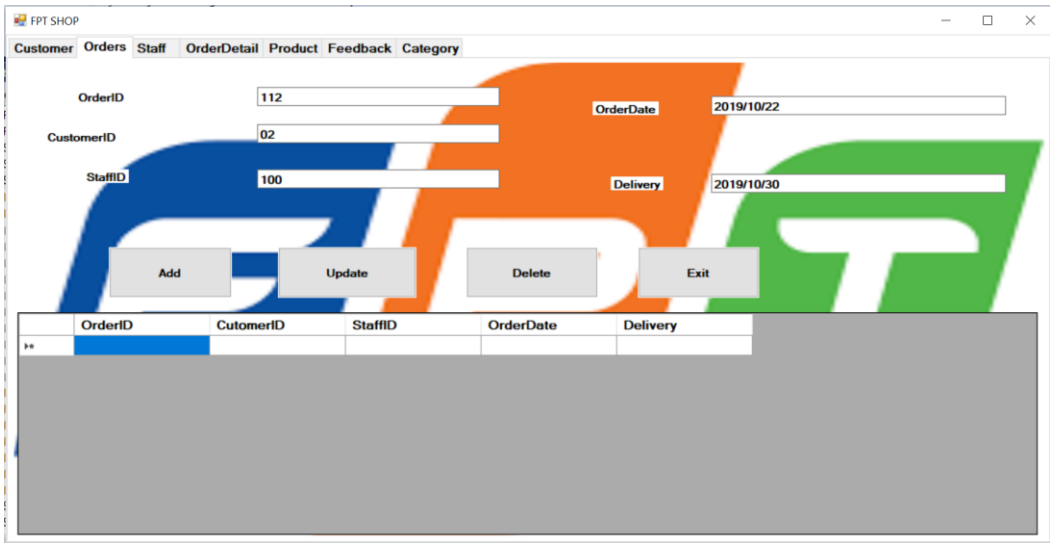
The following query to delete:

```
CREATE TRIGGER DeteleCus
ON [dbo].[Customer]
INSTEAD OF DELETE
AS
	BEGIN
		DELETE [dbo]. [OrderDetail] WHERE [orderID] in
		(SELECT[orderID] FROM [dbo].[Customer] AS c inner join [dbo].[Orders] AS o
		ON (c.CustomerID = o.CustomerID)
		WHERE c.[CustomerID] in (SELECT [CustomerID]  FROM deleted))
		DELETE [dbo].[Orders] WHERE [CustomerID] in (SELECT [CustomerID]  FROM deleted)
		DELETE [dbo].[Feedback] WHERE [CustomerID] in (SELECT [CustomerID] FROM deleted)
		DELETE [dbo].[Customer] WHERE [CustomerID] in (SELECT [CustomerID] FROM deleted)
	END

DELETE FROM [dbo].[Customer] WHERE [CustomerID] = 01
```

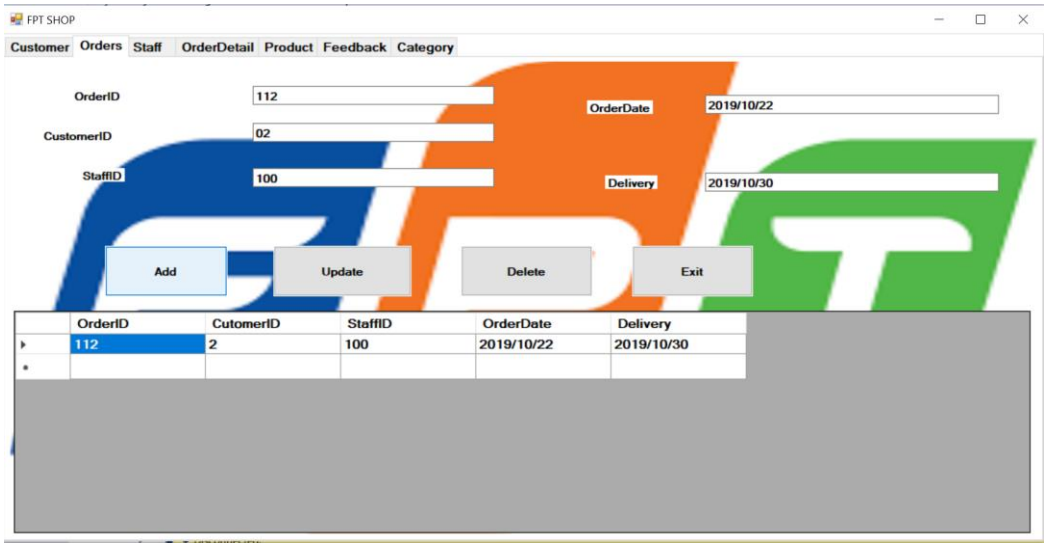
2. Order Interface

This is the Orders interface

2.1 Add Function



The user needs to fill in the Orders information in the Orders table, then click the Add button, the data of the Orders will be added to the database.



The following query to Add:

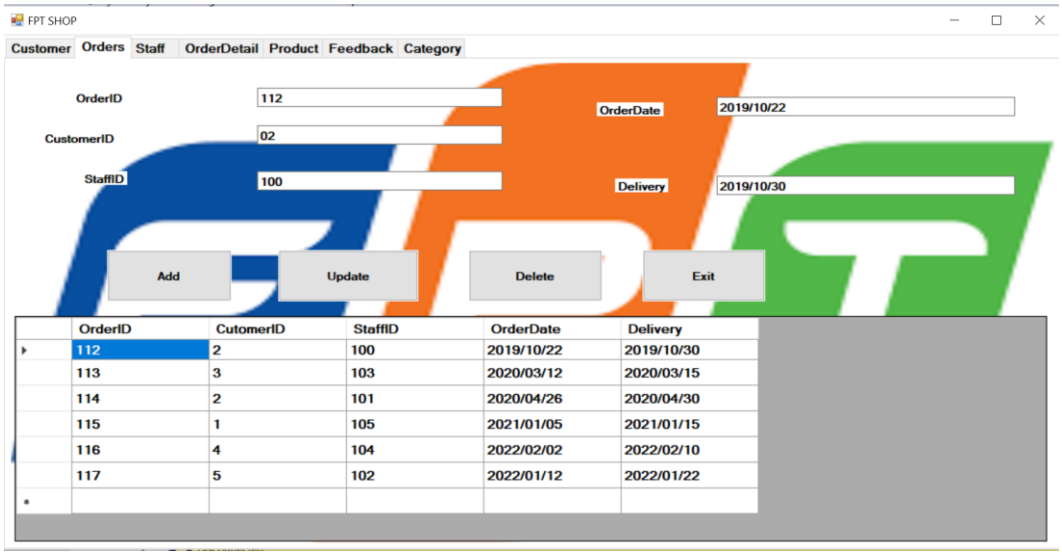
```
CREATE TRIGGER Insert_Orders
ON Orders
FOR INSERT
AS
	IF ((SELECT LEFT(OrderID,1) FROM inserted) <>'1')
	BEGIN
		PRINT ('ID must begin with the number 1')
```

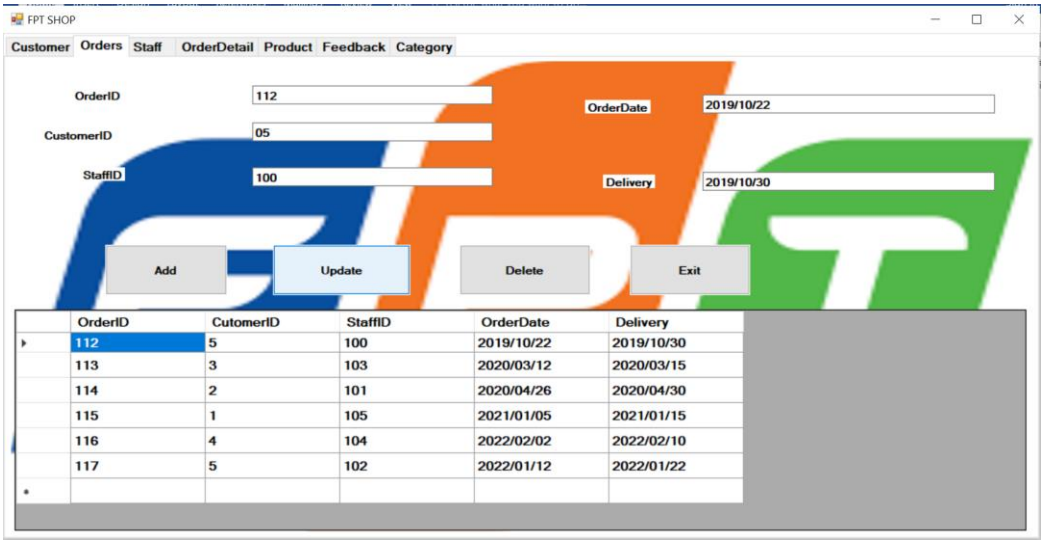
```
            ROLLBACK TRAN
        END
```

```
INSERT INTO Orders VALUES (112, 02, 100, '2019/10/22', '2019/10/30')
```

2.2 Update Function

If the user wants to update the information of the order, they must first enter the order information box and then update the information they want. The user then selects the Update button.





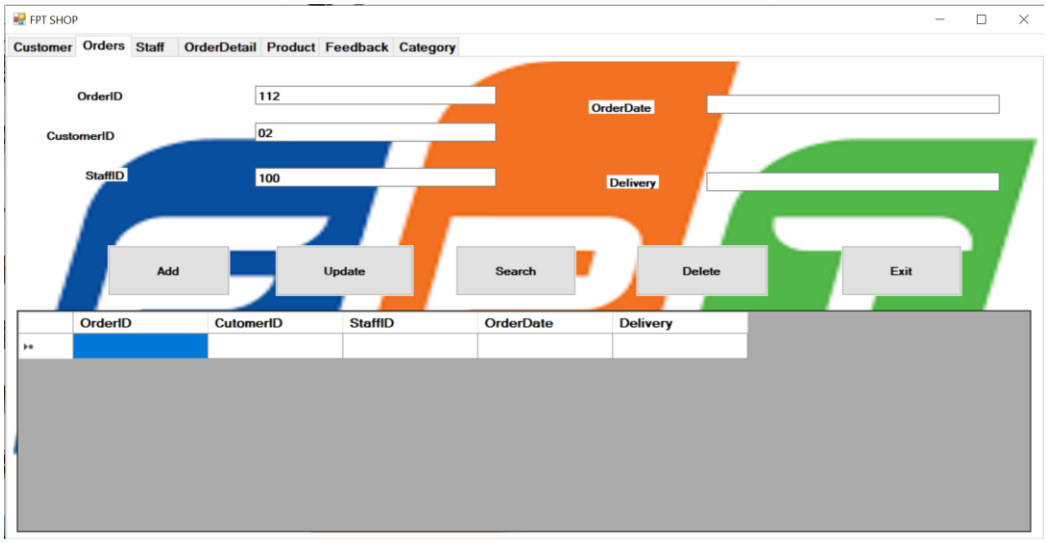
I create a process that will update the information when the user chooses the information and completes the Orders information the user wishes to update.

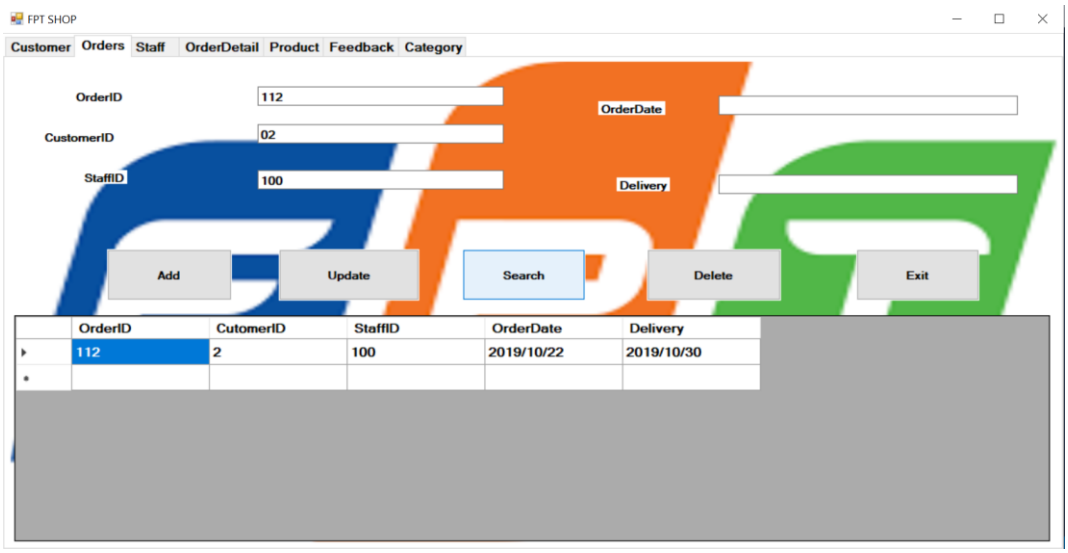
The following query to Update:

```
CREATE PROC UpDateOrders
(@OrderID int, @CusID char(10))
AS
	BEGIN
		UPDATE [dbo].[Orders] SET @CusID = CustomerID WHERE @OrderID = OrderID
	END

	EXEC dbo.UpDateOrders 112, 05
```

2.3 Search Function

Whenever a user wants to search the data for any orders information. The user merely needs to enter the orders information, including the Order ID, Customer ID and Staff ID , in order to conduct a search then click Search button.





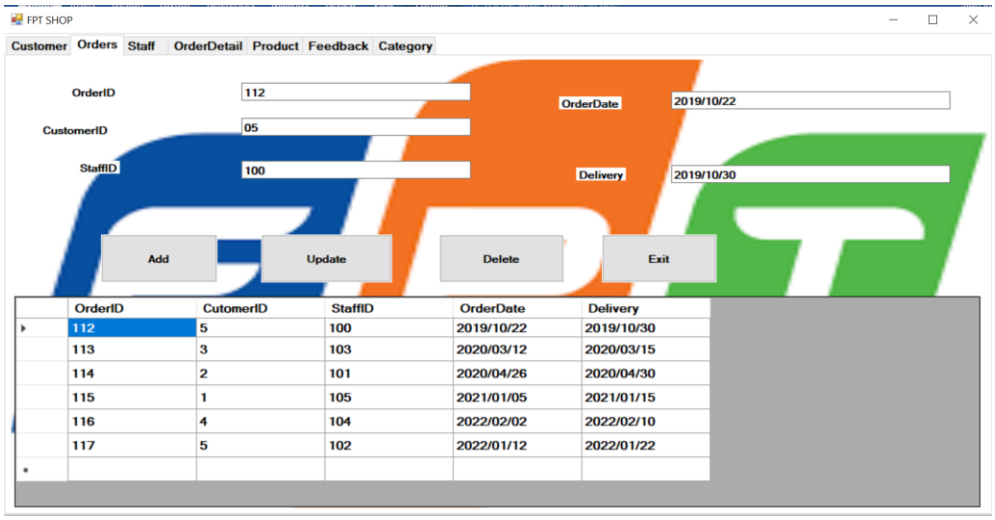
When a user searches for a Orders in a table called Orders, I create procedure search information about that Orders from the data.

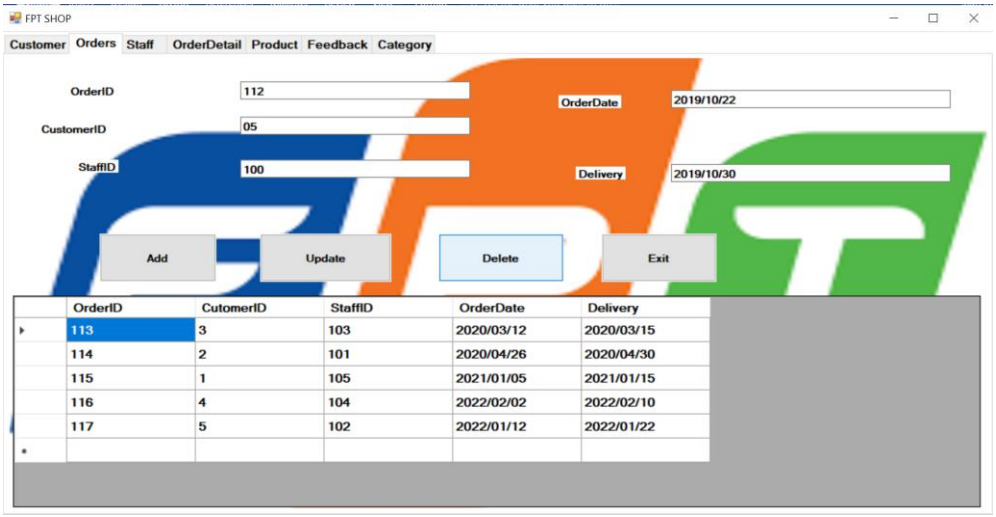
The following query to Search:

```
CREATE PROC SearchOrders
@OrID int, @CusID char(10),@StaffID int
AS
	BEGIN
		SELECT * FROM Orders WHERE OrderID = @OrID and CustomerID = @CusID and StaffID =
@StaffID
	END

	EXEC SearchOrders 112, 02, 100
```

2.4 Delete Function

If a user needs to delete a specific Orders, they must click on the Order ID in the data and then select the Delete button.





I have create a trigger to delete Orders information from the database when a user deletes a certain Orders. When a user wants to remove an order ID, all associated data is also removed from the database.

The following query to Delete:

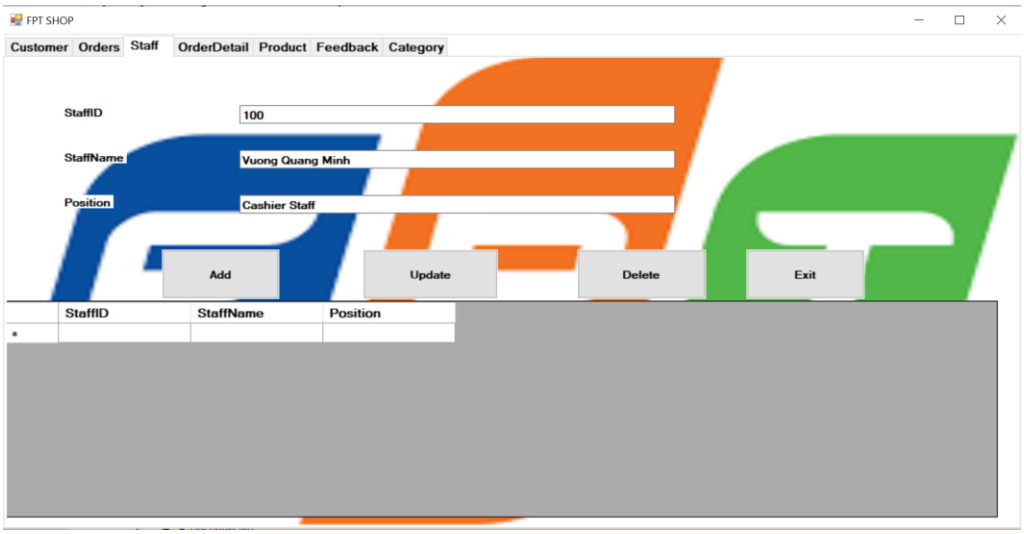
```
CREATE TRIGGER DeleteOrders
ON Orders
INSTEAD OF DELETE
AS
	DELETE FROM OrderDetail WHERE OrderID in (SELECT OrderID FROM deleted)
	DELETE FROM Orders WHERE OrderID in (SELECT OrderID FROM deleted)
GO
```

```
DELETE FROM Orders WHERE OrderID = 112
```

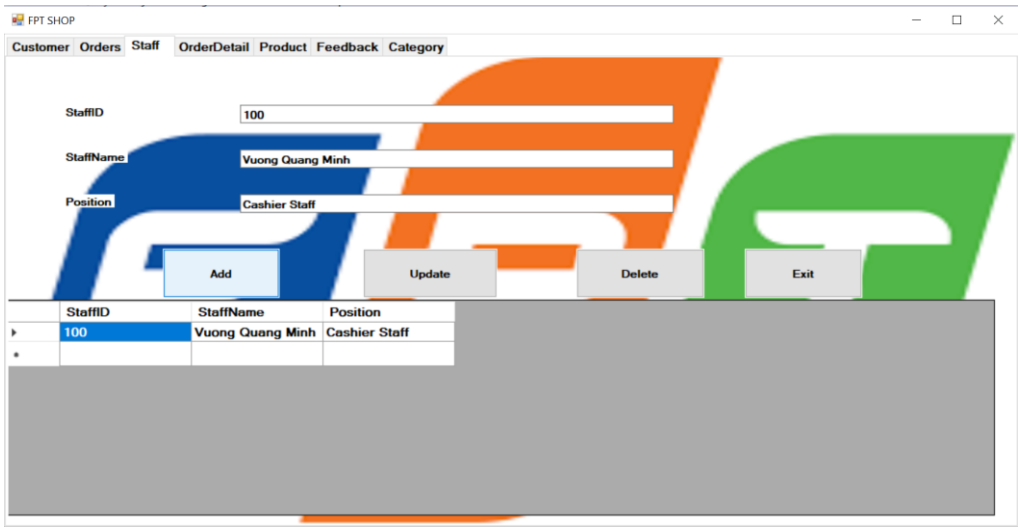
3. Staff Interface

This is the Staff interface

3.1 Add Function



Users only need to fill in the staff's information in the staff information interface, then press the Add button, the staff information will be added to the database.



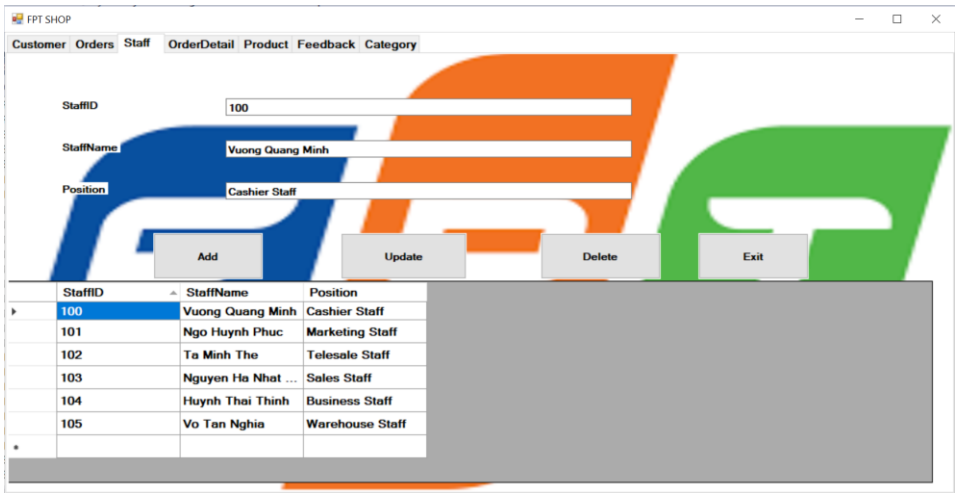
The following query to Add:

```
CREATE TRIGGER InsertStaff
```

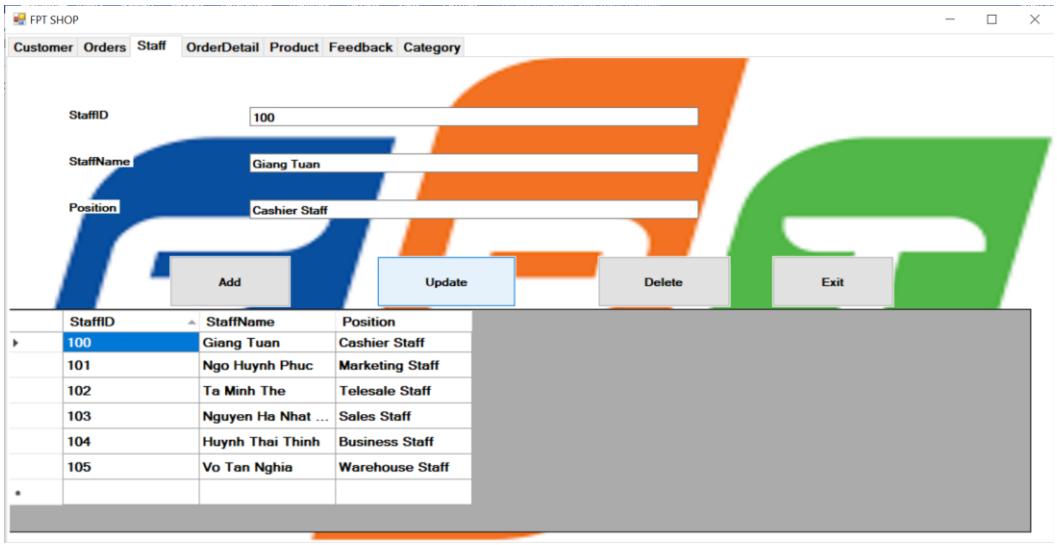
```
ON Staff
FOR INSERT
AS
            IF ((SELECT LEFT(StaffID,1) FROM inserted) <>'1')
      BEGIN
            PRINT ('ID must begin with the number 1')
      ROLLBACK TRAN
      END

INSERT INTO Staff VALUES (200, 'Vuong Quang Minh','Cashier Staff')
```

3.2 Update Function

If the user wants to update the information of the staff, they must first enter the information for the staff and then update the information they want to update. The user then selects the Update button.



I create a process that will update the customer information when the user selects the customer table and fills in the staff information that the user wants to update.

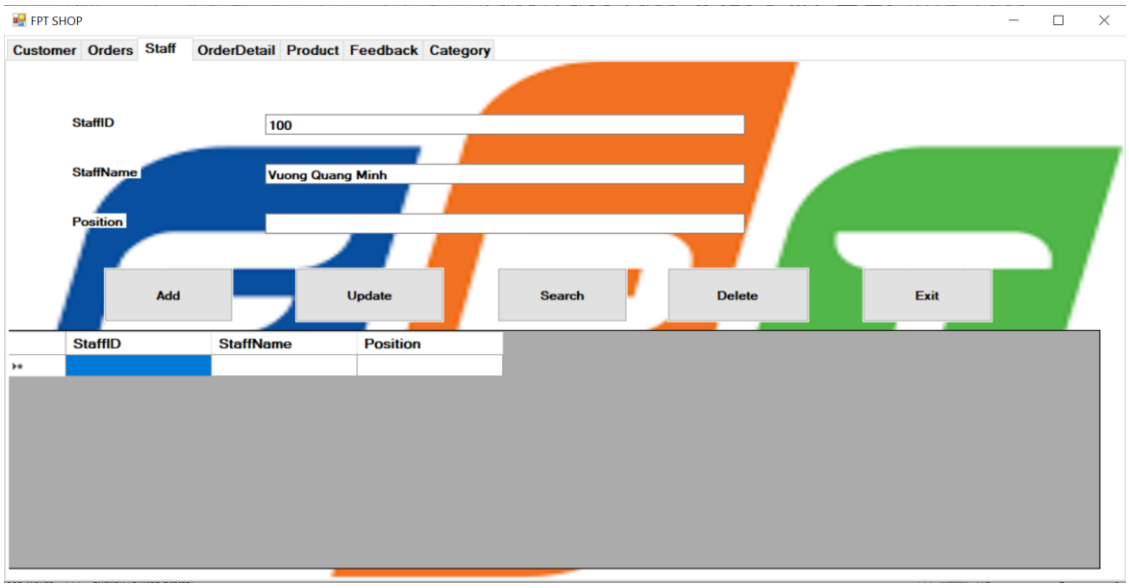


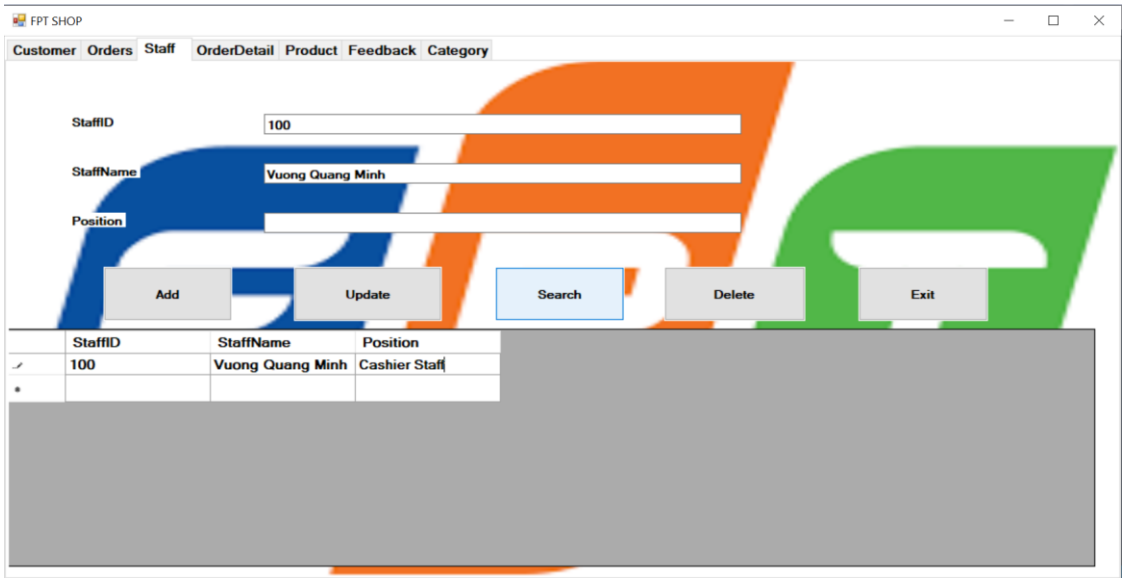
The following query to Update:

```
CREATE PROC UpDateStaff
(@StaffID int, @StaffName varchar(50))
AS
	BEGIN
		UPDATE [dbo].[Staff] SET [StaffName] = @StaffName WHERE @StaffID = [StaffID]
	END

EXEC dbo.UpDateStaff 100, 'Giang Tuan'
```

3.3 Search Function

Whenever a user wants to search the data for any staff information. The user merely needs to enter the staff information, including the staff ID and staff name in order to conduct a search then click Search button.





When a user searches for a Staff in a table called Staff, I create procedure search information about that Staff from the data.

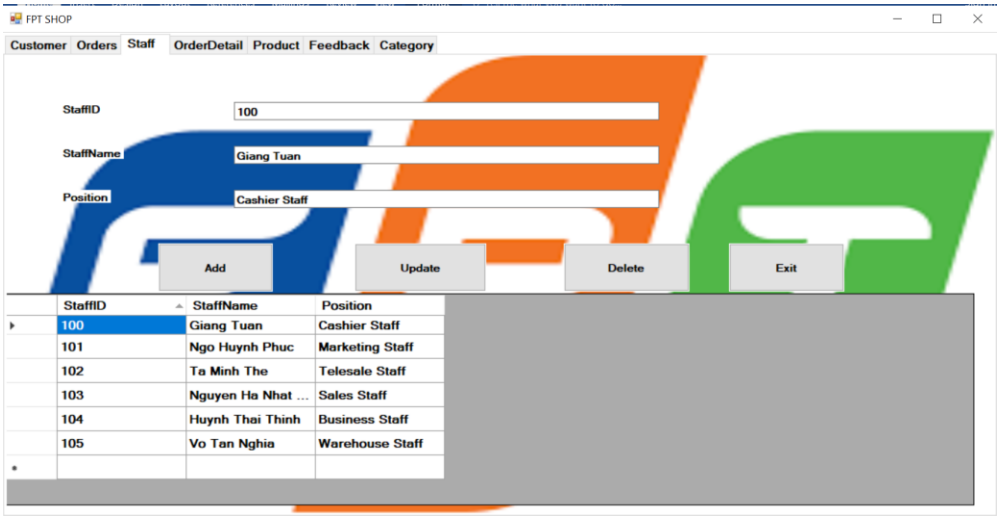
The following query to Search:

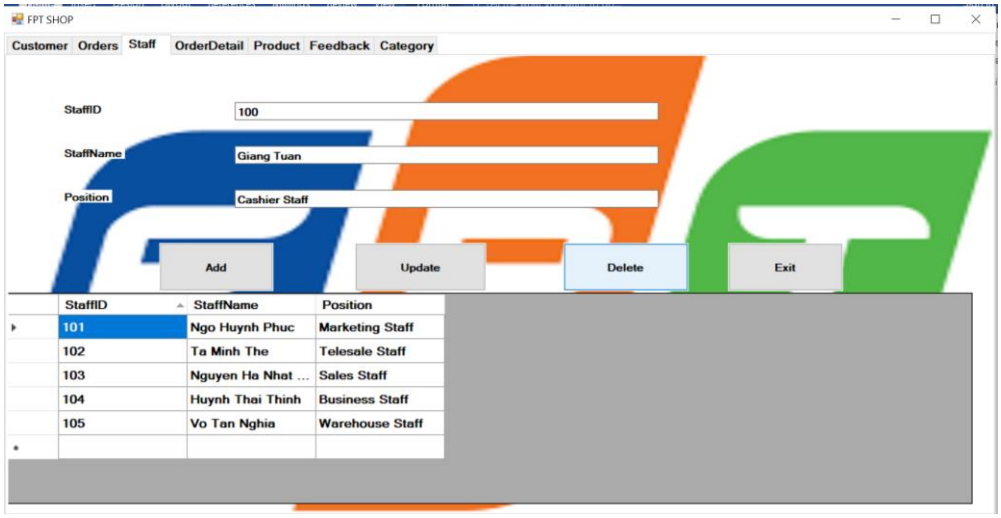
```
CREATE PROC SearchStaff
@StaffID int, @StaffName varchar(50)
AS
	BEGIN
		SELECT * FROM Staff WHERE StaffID = @StaffID and @StaffName = @StaffName
```

```
END

EXEC SearchStaff 100, 'Vuong Quang Minh'
```

3.4 Delete Function

When a user needs to remove staff information from the system, after user selecting the customer ID from the data the user will click the Delete button





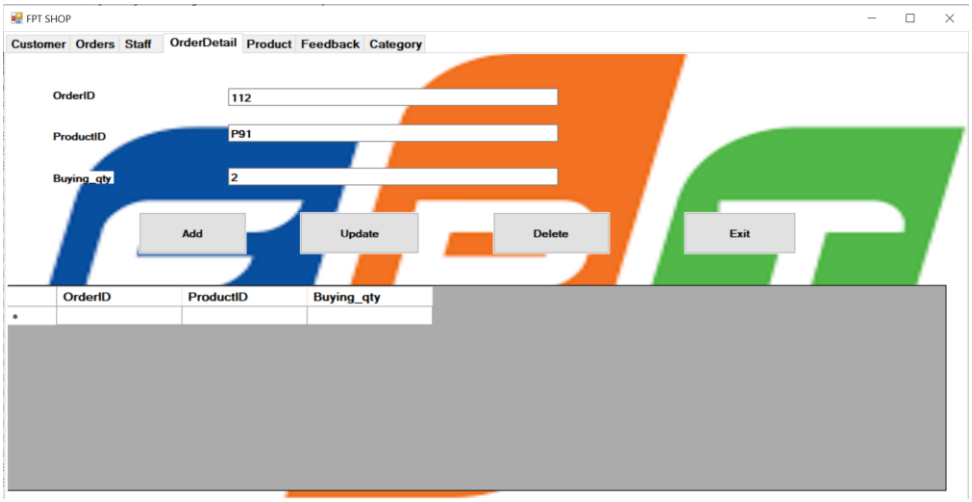
I have created a trigger to remove that Staff information from the database when a user deletes a certain Staff. All connected data is likewise deleted from the database when a user want to remove a staff member ID.

The following query to Delete:

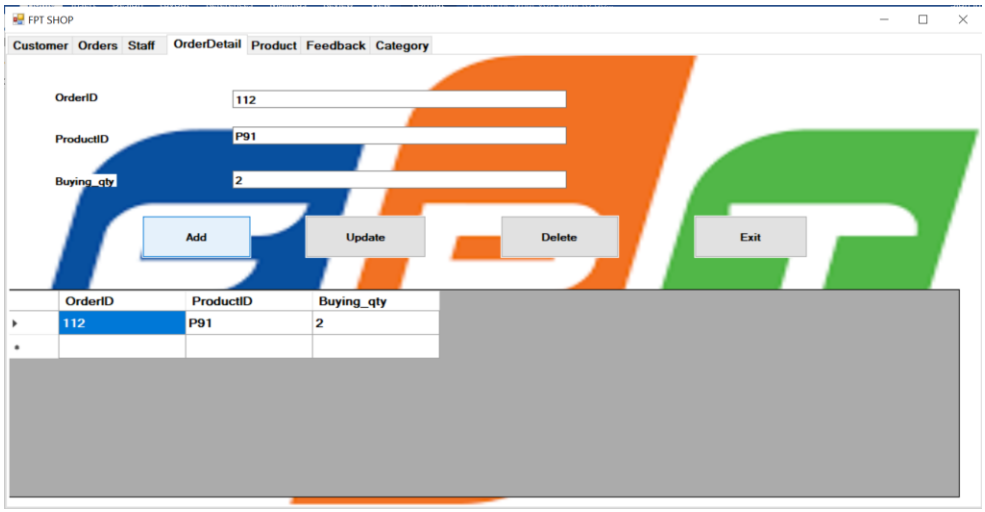
```
CREATE TRIGGER DeleteStaff
ON Staff
INSTEAD OF DELETE
AS
	DELETE FROM Orders WHERE [StaffID] in (SELECT StaffID FROM deleted)
	DELETE FROM Staff WHERE [StaffID] in (SELECT StaffID FROM deleted)
GO

DELETE FROM Staff WHERE StaffID = 100
```

4. OrderDetail Interface

4.1 Add Function



The user needs to fill in the information that the OrderDetail table has into the OrderDetail interface, then press the Add button, the OrderDetail information will be added to the database.

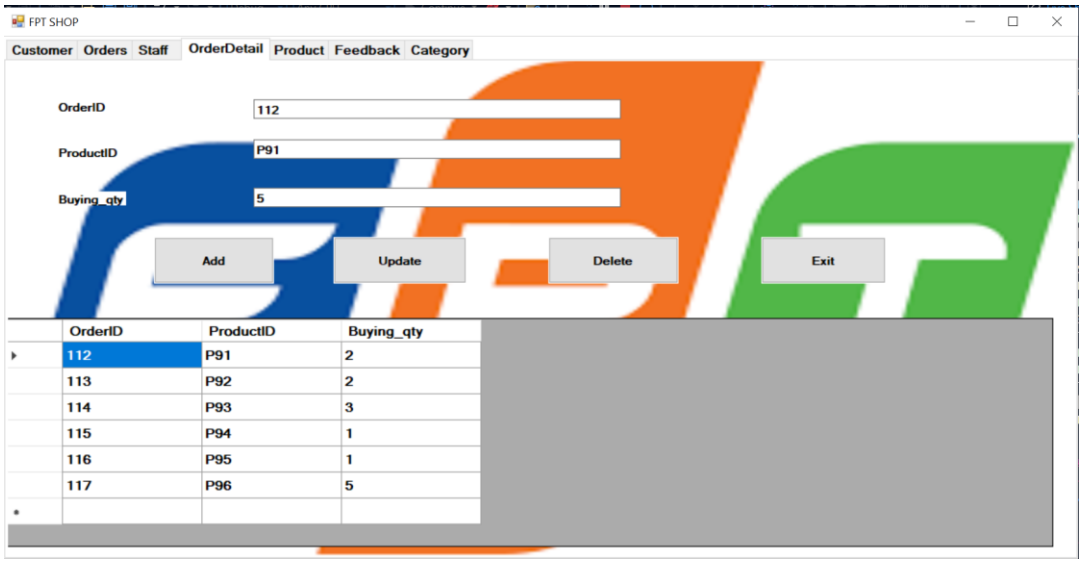


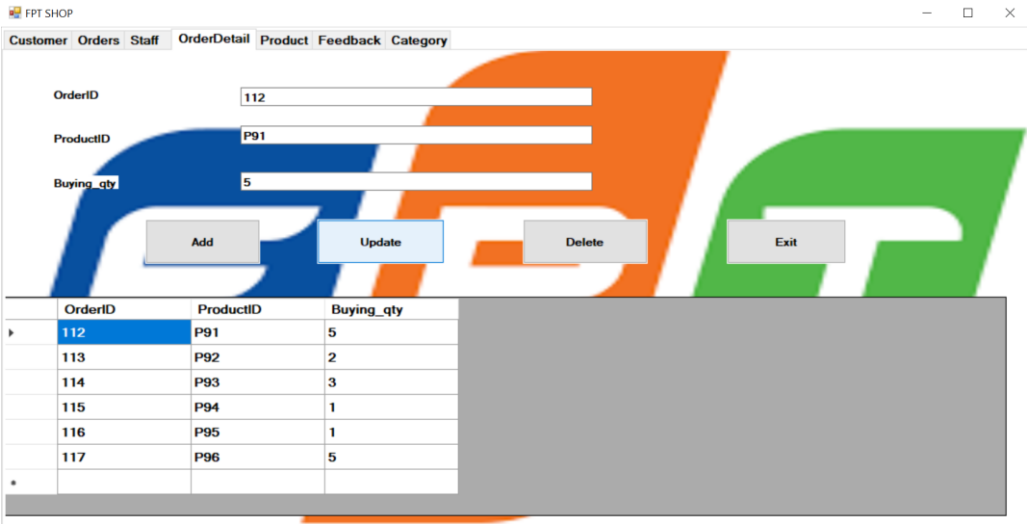
The following query to Add:

```
CREATE TRIGGER InsertOrderDetail
ON OrderDetail
FOR INSERT
AS
                IF ((SELECT LEFT(orderID,1) FROM inserted) <>'1')
            BEGIN
                PRINT ('ID must begin with the number 1')
                ROLLBACK TRAN
            END

INSERT INTO OrderDetail VALUES (112,'P91', 2)
```

4.2 Update Function

If the user wants to update the information of the OrderDetail, they must first enter the information for the OrderDetail and then update the information he wants to update. The user then selects the Update button.





I create a process that will update the OrderDetail information when the user selects the OrderDetail table and fills in the OrderDetail information that the user wants to update.

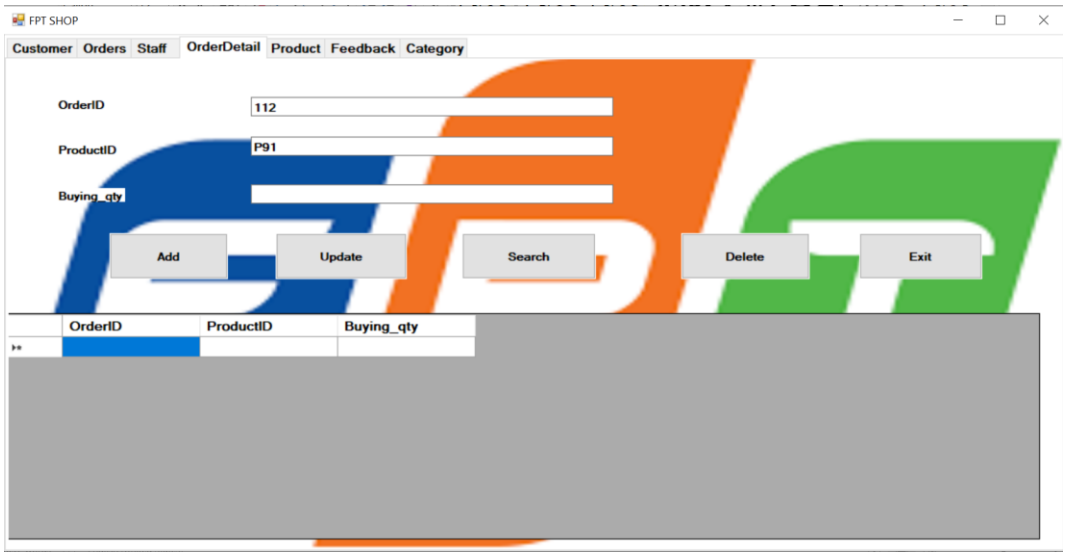
The following query to Update:

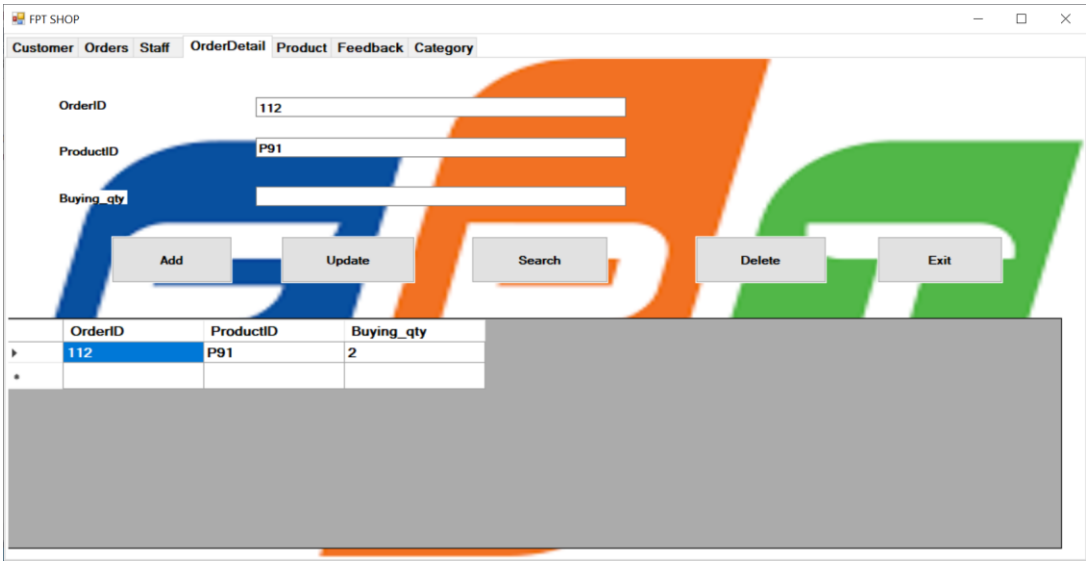
```
CREATE PROC UpDateOrderDetail
(@OrID int, @ProID char(10), @Buy_qty int)
AS
	BEGIN
		UPDATE [dbo].[OrderDetail] SET [buying_qty] = @Buy_qty  WHERE  [orderID] = @OrID and
ProductID = @ProID
	END
```

```
Exec dbo.UpDateOrderDetail 112, P91,5
```

4.3 Search Function

Whenever a user wants to search the data for any OrderDetail information. The user merely needs to enter the staff information, including the Order ID and Product ID in order to conduct a search then click Search button.





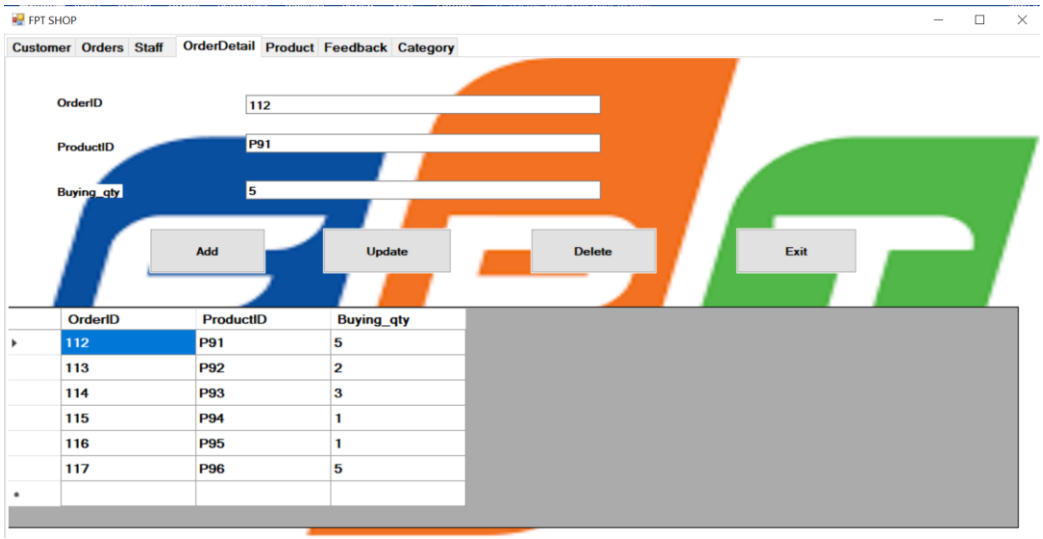
When a user searches for a OrderDetail in a table called OrderDetail, I create procedure search information about that OrderDetail from the data.

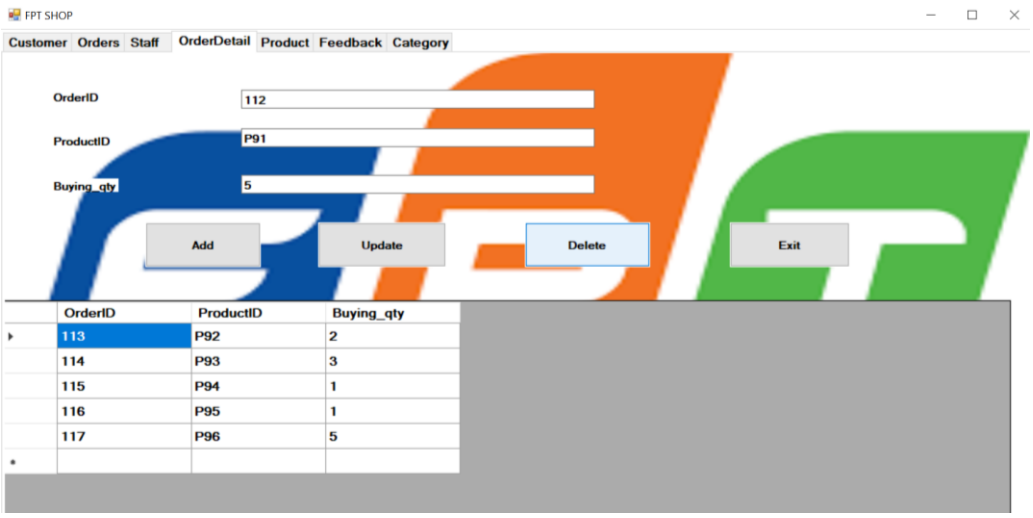
The following query to Search:

```
CREATE PROC SearchOrderDetail
@OrderID int, @ProID char(10)
AS
	BEGIN
		SELECT * FROM OrderDetail WHERE OrderID = @OrderID and ProductID = @ProID
	END

	EXEC SearchOrderDetail 112,'P91'
```

4.4 Delete Function

When user need to delete OrderDetail information from system, user selects OrderDetail ID from data, then user clicks Delete button





I have created a trigger to remove that OrderDetail information from the database when a user delete a certain OrderDetail. All connected data is likewise deleted from the database when a user want to remove OrderDetail.

The following query to Delete:

```
DELETE FROM OrderDetail WHERE orderID = 117 and ProductID = 'P96'
```

References

greenwich, 2022. *greenwich.* [Online]
Available at: https://flm.greenwich.edu.vn/gui/role/student/SyllabusDetails?sylID=2597
[Accessed 9 Dec 2021].