

ASSIGNMENT 2 FRONT SHEET

Qualification	TEC Level 5 HND Diploma in Computing		
Unit number and title	Unit 04: Database Design & Development		
Submission date	October, 31 st 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

★ 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

Learning Outcome	Pass	Merit	Distinction
------------------	------	-------	-------------

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

Table of Contents

Chapter 2: User Interfaces and Queries.....	6
5. Product Interface	6
6. Feedback Interface.....	12
7. Category Interface	17
8. Allow to extract useful information as described the previous assignment.	22
Chapter 3: Testing	24
References	34

Assignment 2

Chapter 2: User Interfaces and Queries

5. Product Interface

5.1 Add Function

Users must enter the product details in the product information interface before pressing the Add button, which adds the details to the database.

FPT SHOP

Customer	Orders	Staff	OrderDetail	Product	Feedback	Category																		
				ProductID: P91 Description: Intel Core i5 6200u ProductName: Laptop HP Probook CategoryID: HP520 Stock: 11 Price: 6000000																				
				<input type="button" value="Add"/> <input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="Exit"/>																				
*				<table border="1"> <thead> <tr> <th>ProductID</th> <th>ProductName</th> <th>Stock</th> <th>Description</th> <th>CategoryID</th> <th>Price</th> </tr> </thead> <tbody> <tr> <td>P91</td> <td>Laptop HP Probook</td> <td>11</td> <td>Intel Core i5 6200u</td> <td>HP520</td> <td>6000000</td> </tr> <tr> <td>*</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	ProductID	ProductName	Stock	Description	CategoryID	Price	P91	Laptop HP Probook	11	Intel Core i5 6200u	HP520	6000000	*							
ProductID	ProductName	Stock	Description	CategoryID	Price																			
P91	Laptop HP Probook	11	Intel Core i5 6200u	HP520	6000000																			
*																								

FPT SHOP

Customer	Orders	Staff	OrderDetail	Product	Feedback	Category																		
				ProductID: P91 Description: Intel Core i5 6200u ProductName: Laptop HP Probook CategoryID: HP520 Stock: 11 Price: 6000000																				
				<input type="button" value="Add"/> <input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="Exit"/>																				
*				<table border="1"> <thead> <tr> <th>ProductID</th> <th>ProductName</th> <th>Stock</th> <th>Description</th> <th>CategoryID</th> <th>Price</th> </tr> </thead> <tbody> <tr> <td>P91</td> <td>Laptop HP Probook</td> <td>11</td> <td>Intel Core i5 6200u</td> <td>HP520</td> <td>6000000</td> </tr> <tr> <td>*</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	ProductID	ProductName	Stock	Description	CategoryID	Price	P91	Laptop HP Probook	11	Intel Core i5 6200u	HP520	6000000	*							
ProductID	ProductName	Stock	Description	CategoryID	Price																			
P91	Laptop HP Probook	11	Intel Core i5 6200u	HP520	6000000																			
*																								

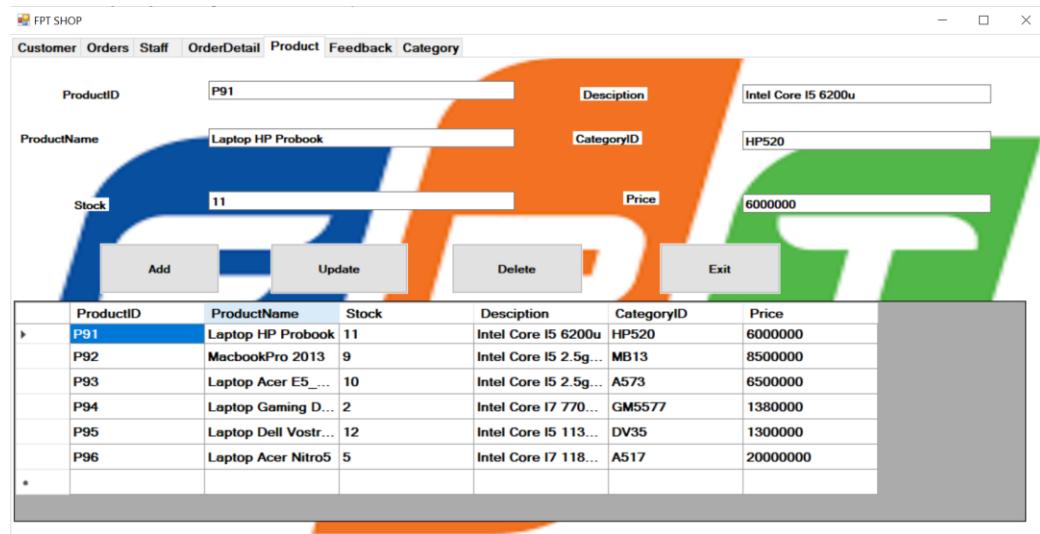
The following query to Add:

```

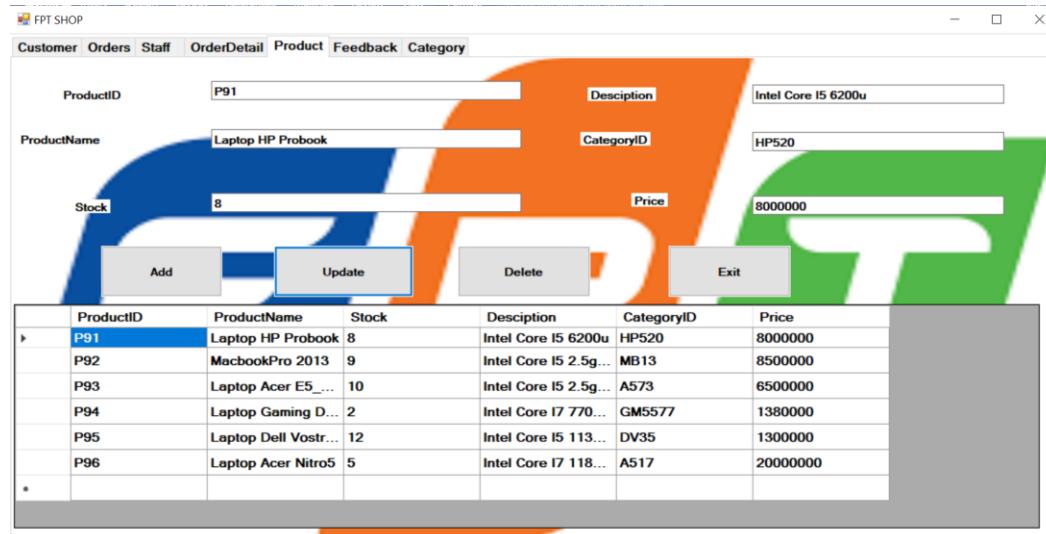
CREATE TRIGGER InsertPro
ON Product
FOR INSERT
AS
    IF ((select LEFT(ProductID,1) from inserted) <> 'P')
        BEGIN
            PRINT ('Product IDs starting with P')
            ROLLBACK TRAN
        END
    Insert into Product values ('P91', 'Laptop HP Probook', 11, 'Intel Core I5 6200u', 'HP520', 6000000)
    
```

5.2 Update Function

Users must enter the product information first, then update the product information they want to update. Then the user chooses to click the Update button.



ProductID	ProductName	Stock	Description	CategoryID	Price
P91	Laptop HP Probook	11	Intel Core i5 6200u	HP520	6000000
P92	MacbookPro 2013	9	Intel Core i5 2.5g...	MB13	8500000
P93	Laptop Acer E5_...	10	Intel Core i5 2.5g...	A573	6500000
P94	Laptop Gaming D...	2	Intel Core i7 770...	GM5577	1380000
P95	Laptop Dell Vostri...	12	Intel Core i5 113...	DV35	1300000
P96	Laptop Acer Nitro5	5	Intel Core i7 118...	A517	20000000



ProductID	ProductName	Stock	Description	CategoryID	Price
P91	Laptop HP Probook	8	Intel Core i5 6200u	HP520	8000000
P92	MacbookPro 2013	9	Intel Core i5 2.5g...	MB13	8500000
P93	Laptop Acer E5_...	10	Intel Core i5 2.5g...	A573	6500000
P94	Laptop Gaming D...	2	Intel Core i7 770...	GM5577	1380000
P95	Laptop Dell Vostri...	12	Intel Core i5 113...	DV35	1300000
P96	Laptop Acer Nitro5	5	Intel Core i7 118...	A517	20000000

I create a flow that will update the product information. When the user selects the product table and enters the product information they want to update.

The following query to Update

```
CREATE PROC UpDatePro
(@ProID char(10), @Stock int, @Price int)
```

AS

BEGIN

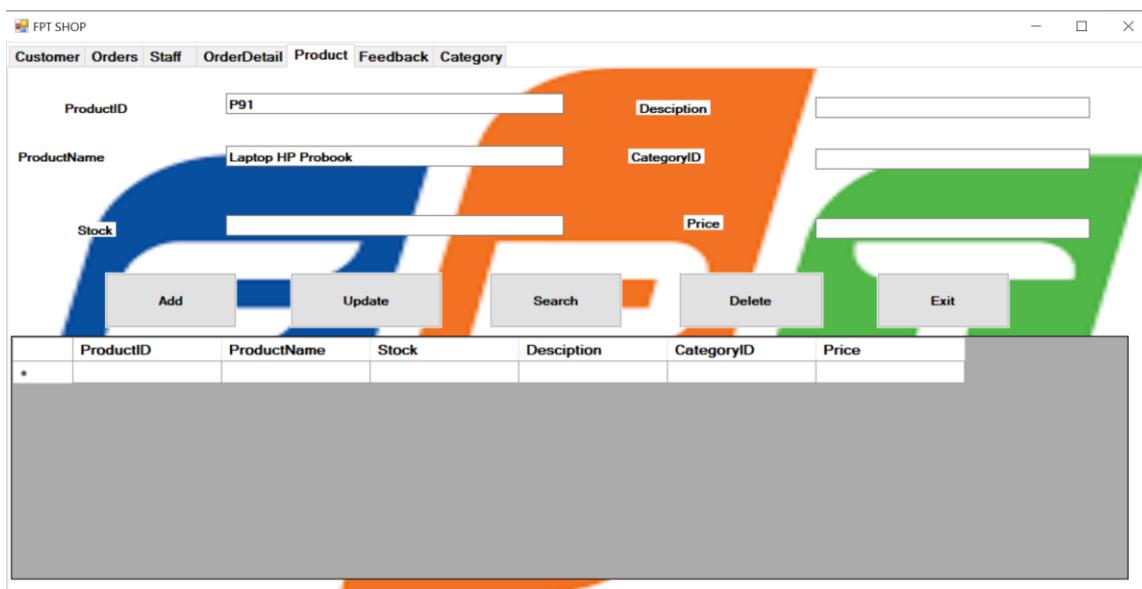
```
UPDATE [dbo].[Product] SET [Stock] = @Stock, [Price] = @Price WHERE @ProID =
[ProductID]
```

END

```
EXEC dbo.UpDatePro 'P91', 8, 8000000
```

5.3 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.



FPT SHOP

Customer Orders Staff OrderDetail Product Feedback Category

ProductID	P91	Description			
ProductName	Laptop HP Probook	CategoryID			
Stock		Price			
Add	Update	Search	Delete	Exit	
ProductID	ProductName	Stock	Description	CategoryID	Price
P91	Laptop HP Probook	11	Intel Core i5 6200u	HP520	6000000
*					

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

The following query to Search:

```
CREATE PROC SearchProduct
@ProID char(10), @ProName varchar(50)
AS
BEGIN
    SELECT * FROM Product WHERE ProductID = @ProID and ProductName = @ProName
END

EXEC SearchProduct 'P91', 'Laptop HP Probook'
```

5.4 Delete Function

When a user needs to remove product data from the system, they must first pick the product ID from the data and then click the Delete button.

FPT SHOP

Customer	Orders	Staff	OrderDetail	Product	Feedback	Category																																										
				ProductID: P91 Description: Intel Core i5 6200u ProductName: Laptop HP Probook CategoryID: HP520 Stock: 8 Price: 8000000																																												
<input type="button" value="Add"/> <input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="Exit"/>																																																
<table border="1"> <thead> <tr> <th>ProductID</th> <th>ProductName</th> <th>Stock</th> <th>Description</th> <th>CategoryID</th> <th>Price</th> </tr> </thead> <tbody> <tr> <td>P91</td> <td>Laptop HP Probook</td> <td>8</td> <td>Intel Core i5 6200u</td> <td>HP520</td> <td>8000000</td> </tr> <tr> <td>P92</td> <td>MacbookPro 2013</td> <td>9</td> <td>Intel Core i5 2.5g...</td> <td>MB13</td> <td>8500000</td> </tr> <tr> <td>P93</td> <td>Laptop Acer E5_...</td> <td>10</td> <td>Intel Core i5 2.5g...</td> <td>A573</td> <td>6500000</td> </tr> <tr> <td>P94</td> <td>Laptop Gaming D...</td> <td>2</td> <td>Intel Core i7 770...</td> <td>GM5577</td> <td>1380000</td> </tr> <tr> <td>P95</td> <td>Laptop Dell Vostr...</td> <td>12</td> <td>Intel Core i5 113...</td> <td>DV35</td> <td>1300000</td> </tr> <tr> <td>P96</td> <td>Laptop Acer Nitro5</td> <td>5</td> <td>Intel Core i7 118...</td> <td>A517</td> <td>20000000</td> </tr> </tbody> </table>							ProductID	ProductName	Stock	Description	CategoryID	Price	P91	Laptop HP Probook	8	Intel Core i5 6200u	HP520	8000000	P92	MacbookPro 2013	9	Intel Core i5 2.5g...	MB13	8500000	P93	Laptop Acer E5_...	10	Intel Core i5 2.5g...	A573	6500000	P94	Laptop Gaming D...	2	Intel Core i7 770...	GM5577	1380000	P95	Laptop Dell Vostr...	12	Intel Core i5 113...	DV35	1300000	P96	Laptop Acer Nitro5	5	Intel Core i7 118...	A517	20000000
ProductID	ProductName	Stock	Description	CategoryID	Price																																											
P91	Laptop HP Probook	8	Intel Core i5 6200u	HP520	8000000																																											
P92	MacbookPro 2013	9	Intel Core i5 2.5g...	MB13	8500000																																											
P93	Laptop Acer E5_...	10	Intel Core i5 2.5g...	A573	6500000																																											
P94	Laptop Gaming D...	2	Intel Core i7 770...	GM5577	1380000																																											
P95	Laptop Dell Vostr...	12	Intel Core i5 113...	DV35	1300000																																											
P96	Laptop Acer Nitro5	5	Intel Core i7 118...	A517	20000000																																											

FPT SHOP

Customer	Orders	Staff	OrderDetail	Product	Feedback	Category																																				
				ProductID: P91 Description: Intel Core i5 6200u ProductName: Laptop HP Probook CategoryID: HP520 Stock: 8 Price: 8000000																																						
<input type="button" value="Add"/> <input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="Exit"/>																																										
<table border="1"> <thead> <tr> <th>ProductID</th> <th>ProductName</th> <th>Stock</th> <th>Description</th> <th>CategoryID</th> <th>Price</th> </tr> </thead> <tbody> <tr> <td>P92</td> <td>MacbookPro 2013</td> <td>9</td> <td>Intel Core i5 2.5g...</td> <td>MB13</td> <td>8500000</td> </tr> <tr> <td>P93</td> <td>Laptop Acer E5_...</td> <td>10</td> <td>Intel Core i5 2.5g...</td> <td>A573</td> <td>6500000</td> </tr> <tr> <td>P94</td> <td>Laptop Gaming D...</td> <td>2</td> <td>Intel Core i7 770...</td> <td>GM5577</td> <td>1380000</td> </tr> <tr> <td>P95</td> <td>Laptop Dell Vostr...</td> <td>12</td> <td>Intel Core i5 113...</td> <td>DV35</td> <td>1300000</td> </tr> <tr> <td>P96</td> <td>Laptop Acer Nitro5</td> <td>5</td> <td>Intel Core i7 118...</td> <td>A517</td> <td>20000000</td> </tr> </tbody> </table>							ProductID	ProductName	Stock	Description	CategoryID	Price	P92	MacbookPro 2013	9	Intel Core i5 2.5g...	MB13	8500000	P93	Laptop Acer E5_...	10	Intel Core i5 2.5g...	A573	6500000	P94	Laptop Gaming D...	2	Intel Core i7 770...	GM5577	1380000	P95	Laptop Dell Vostr...	12	Intel Core i5 113...	DV35	1300000	P96	Laptop Acer Nitro5	5	Intel Core i7 118...	A517	20000000
ProductID	ProductName	Stock	Description	CategoryID	Price																																					
P92	MacbookPro 2013	9	Intel Core i5 2.5g...	MB13	8500000																																					
P93	Laptop Acer E5_...	10	Intel Core i5 2.5g...	A573	6500000																																					
P94	Laptop Gaming D...	2	Intel Core i7 770...	GM5577	1380000																																					
P95	Laptop Dell Vostr...	12	Intel Core i5 113...	DV35	1300000																																					
P96	Laptop Acer Nitro5	5	Intel Core i7 118...	A517	20000000																																					

I have created a trigger to remove that Product's information from the database when a user deletes certain Product. When a user chooses to delete a Product, all related data is also removed from the database.

The following query to Delete:

```

CREATE TRIGGER DeletePro
ON Product
INSTEAD OF DELETE
AS
    DELETE FROM OrderDetail WHERE [ProductID] in (SELECT ProductID FROM deleted)
    DELETE FROM Feedback WHERE [ProductID] in (SELECT ProductID FROM deleted)
    DELETE FROM Product WHERE [ProductID] in (SELECT ProductID FROM deleted)

```

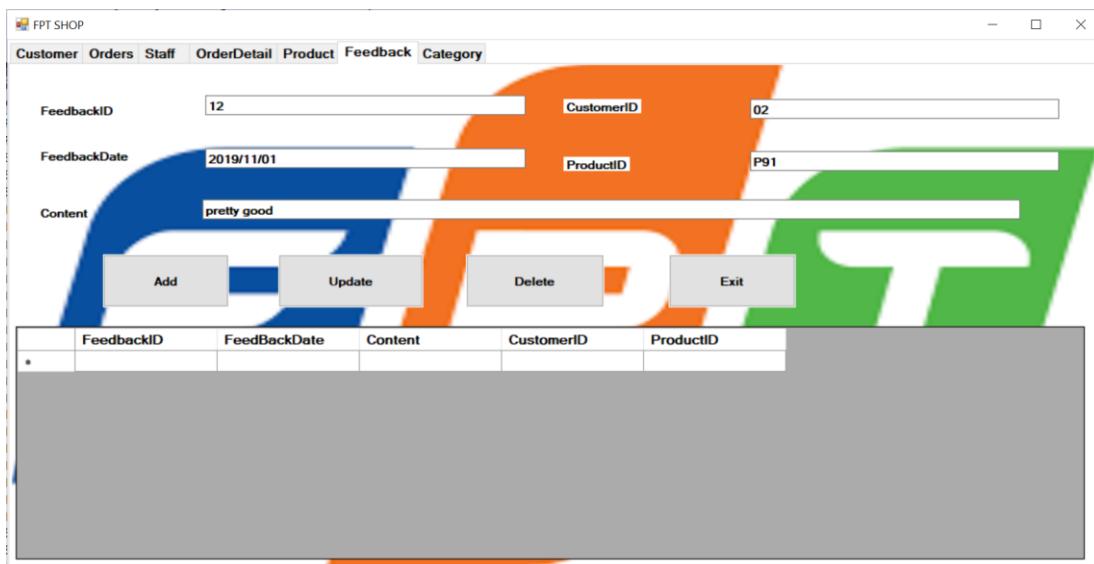
```
DELETE FROM Product WHERE ProductID = 'P91'
```

6. Feedback Interface

This is the Feedback interface

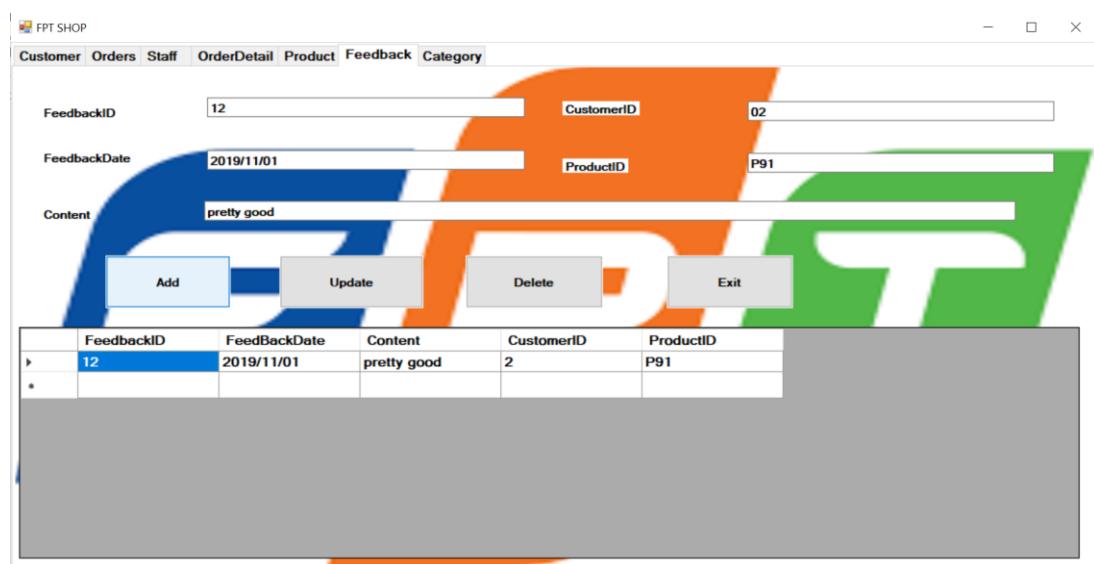
6.1 Add Function

When the user finishes filling in the information from the Feedback panel into the Feedback interface and clicks the Add button, the Feedback will be added to the database.



The screenshot shows a Windows application window titled "FTP SHOP". The menu bar includes Customer, Orders, Staff, OrderDetail, Product, Feedback, and Category. The "Feedback" tab is selected. The main area contains four input fields: FeedbackID (12), CustomerID (02), FeedbackDate (2019/11/01), ProductID (P91), and a Content field containing "pretty good". Below these fields are four buttons: Add, Update, Delete, and Exit. The "Add" button is highlighted with a blue glow. At the bottom is a table with columns: FeedbackID, FeedBackDate, Content, CustomerID, and ProductID. A single row is present with values corresponding to the input fields above.

	FeedbackID	FeedBackDate	Content	CustomerID	ProductID
*	12	2019/11/01	pretty good	02	P91



The screenshot shows the same "FTP SHOP" application window after the feedback has been added. The "Add" button is no longer highlighted. The table at the bottom now shows two rows. The first row corresponds to the previous screenshot. A second row is added, showing the newly added feedback entry with FeedbackID 12, FeedBackDate 2019/11/01, Content "pretty good", CustomerID 2, and ProductID P91.

	FeedbackID	FeedBackDate	Content	CustomerID	ProductID
*	12	2019/11/01	pretty good	02	P91
>	12	2019/11/01	pretty good	2	P91

The following query to Add:

```

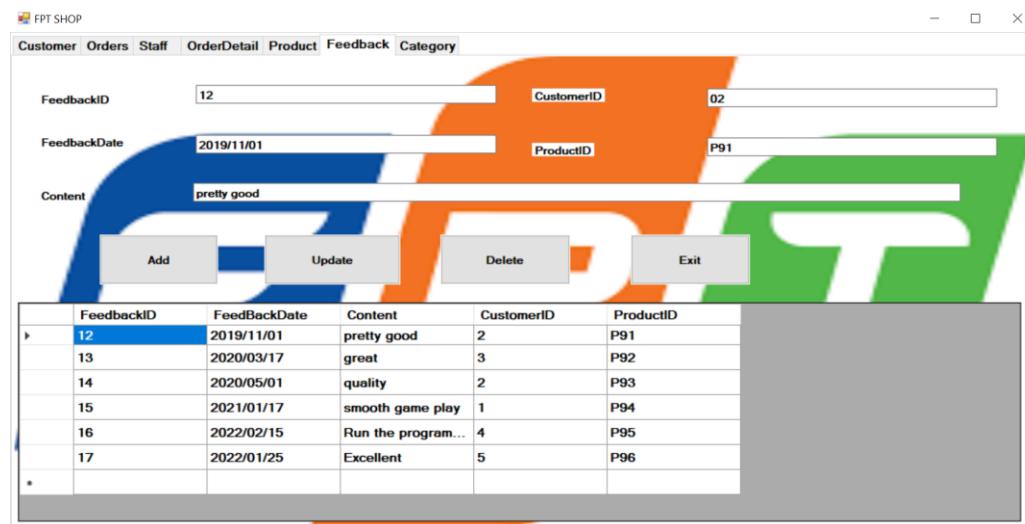
CREATE TRIGGER InsertFeed
ON Feedback
FOR INSERT
AS
    IF ((SELECT LEFT(FeedbackID,1) FROM inserted) <> '1')
    BEGIN
        PRINT ('ID must begin with the number 1')
        ROLLBACK TRAN
    END

INSERT INTO Feedback VALUES (12, '2019/11/01', 'pretty good', 02, 'P91')

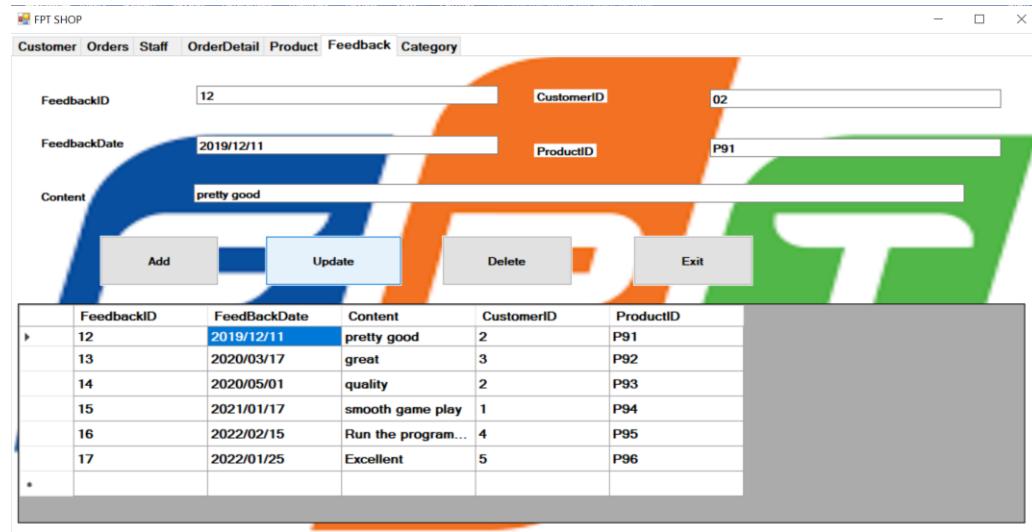
```

6.2 Update Function

If a user wishes to update feedback information, they must first enter feedback information and then update the desired information. The user then choose to click Update button.



I create a feedback update process. User needs to enter information for Feedback panel and then update it



The screenshot shows a Windows application window titled "FPT SHOP". The menu bar includes Customer, Orders, Staff, OrderDetail, Product, Feedback, and Category. The "Feedback" tab is selected. The main area contains four input fields: "FeedbackID" (12), "CustomerID" (02), "FeedbackDate" (2019/12/11), and "ProductID" (P91). Below these is a text area labeled "Content" containing "pretty good". At the bottom are four buttons: Add, Update (highlighted in blue), Delete, and Exit. A large watermark for "FPT" is overlaid across the entire window.

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

The following query to Update:

```

CREATE PROC UpDateFeed
(@FeedID char(10), @FeedDate varchar(10))
AS
BEGIN
    UPDATE [dbo].[Feedback] SET [FeedbackDate] = @FeedDate WHERE [FeedbackID] = @FeedID
END

EXEC dbo.UpDateFeed 12, '2019/12/11'

```

6.3 Search Function

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

FPT SHOP

Customer	Orders	Staff	OrderDetail	Product	Feedback	Category
					12	CustomerID
					2019/11/01	ProductID
					Content	
					Add	Update
					Search	Delete
					Exit	

FPT SHOP

Customer	Orders	Staff	OrderDetail	Product	Feedback	Category
					12	CustomerID
					2019/11/01	ProductID
					Content	
					Add	Update
					Search	Delete
					Exit	

FeedbackID	FeedBackDate	Content	CustomerID	ProductID
12	2019/11/01	pretty good	02	P91
*				

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

The following query to Search:

```

CREATE PROC SearchFeed
@FeedID int, @FeedDate varchar(10)
AS
BEGIN
    SELECT * FROM Feedback WHERE FeedbackID = @FeedID and FeedbackDate = @FeedDate

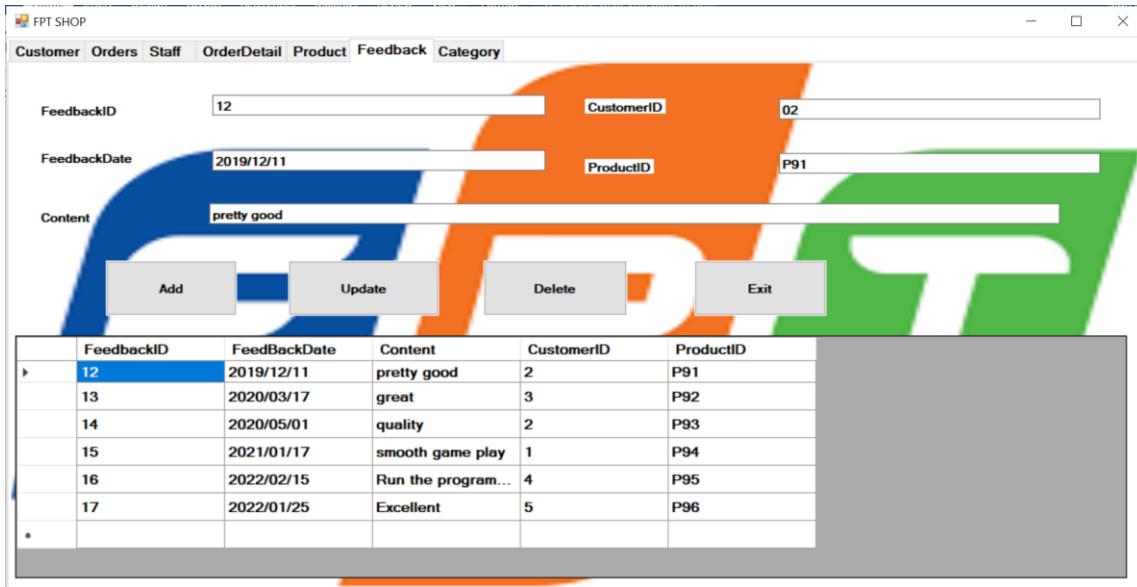
```

END

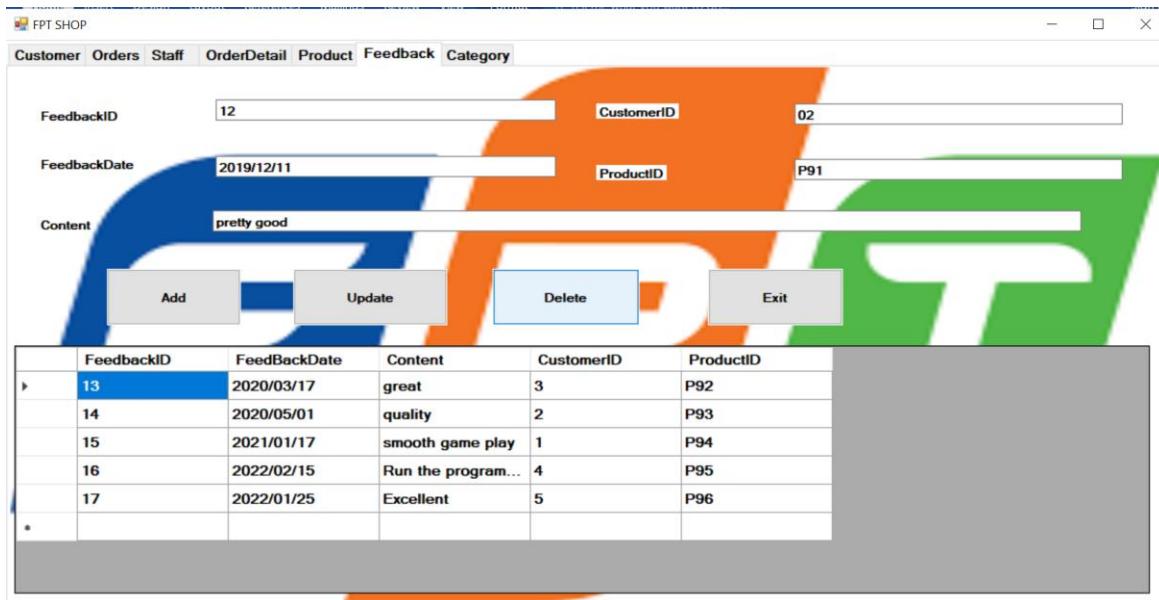
EXEC SearchFeed 12, '2019/11/01'

6.4 Delete Function

When the user needs to delete the Feedback information from the system. User need to select the ID of Feedback that user wants from the data, then user will click on Delete button



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



	FeedbackID	FeedBackDate	Content	CustomerID	ProductID
▶	13	2020/03/17	great	3	P92
	14	2020/05/01	quality	2	P93
	15	2021/01/17	smooth game play	1	P94
	16	2022/02/15	Run the program...	4	P95
	17	2022/01/25	Excellent	5	P96

I created a trigger to remove that feedback from the database when a certain number of feedback are deleted by the user. All connected data is also deleted from the database when a certain feedback is deleted by the user.

The following query to Delete:

```
Select * From Feedback
CREATE TRIGGER DeleteFeed
ON Feedback
FOR DELETE
AS
    Delete From Feedback Where FeedbackID in (Select FeedbackID From deleted)
Go

DELETE FROM Feedback WHERE FeedbackID= 12
```

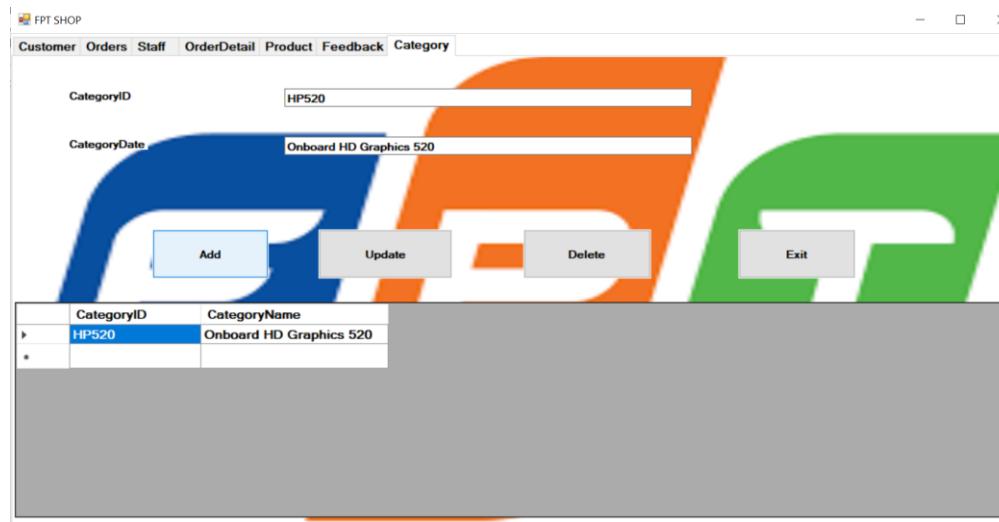
7. Category Interface

This is the Category interface

7.1 Add Function

Users must input Category information from the Category table into the Category interface before pressing the Add button to add the information to the database.





The following query to Add:

```

CREATE PROC Insertcate
@cateid char(10), @catename varchar(50)
AS
    INSERT INTO Category VALUES ( @cateid, @catename)

    Exec Insertcate 'HP520', 'Onboard HD Graphics 520'

```

7.2 Update Function

Users who want to update the information of the Category must first enter the information for the Category and then update the information they want to update. The user then selects the Update button.

FPT SHOP

CategoryID	CategoryName
HP520	Onboard HD Graphics 50000
MB13	Onboard HD Graphics HD...
A573	GeForce GT920m
GM5577	GeForce GTX1050
DV35	GeForce MX230
A517	GeForce RTX3050
*	

Add Update Delete Exit

FPT SHOP

CategoryID	CategoryName
HP520	Onboard HD Graphics 50000
MB13	Onboard HD Graphics HD...
A573	GeForce GT920m
GM5577	GeForce GTX1050
DV35	GeForce MX230
A517	GeForce RTX3050
*	

Add Update Delete Exit

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

The following query to Update:

```

CREATE PROC UpDateCat
(@CatID char(10), @CatName varchar(50))
AS
BEGIN
    UPDATE [dbo].[Category] SET [CategoryName] = @CatName WHERE @CatID = [CategoryID]

```

END

```
UPDATE [dbo].[Category] SET [CategoryName] = 'Onboard HD Graphics 5000' WHERE @CateID = 'HP520'
```

7.3 Search Function

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



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

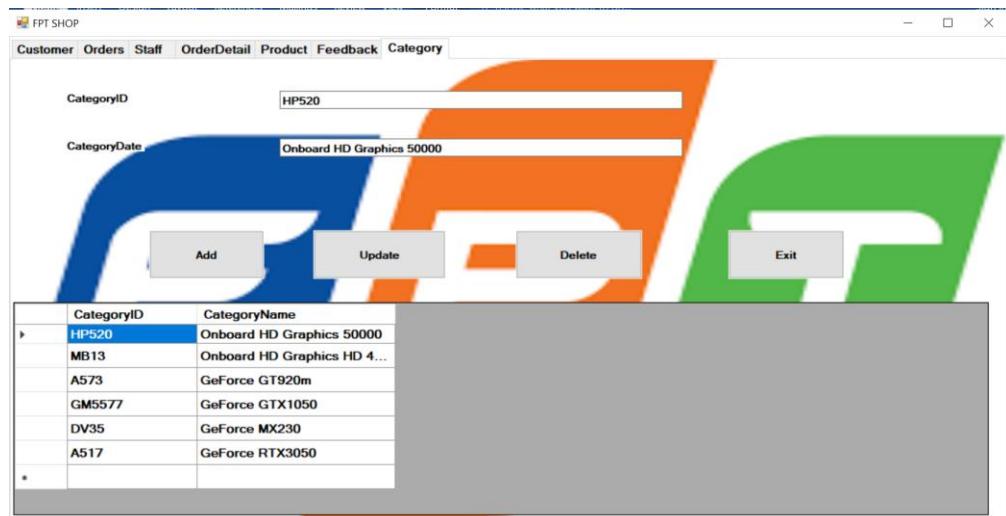
The following query to Search:

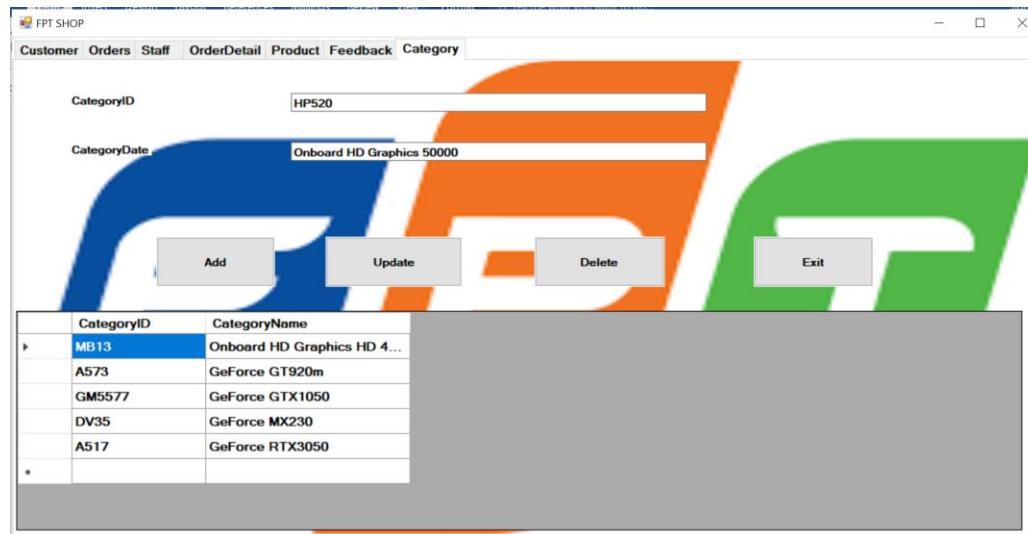
```
CREATE PROC SearchCate
@CateID char(10)
AS
BEGIN
    SELECT * FROM Category WHERE CategoryID = @CateID
END

EXEC SearchCate 'HP520'
```

7.4 Delete Function

When user needs to delete category information from the system, the user just needs to enter the category information that the user wants to delete from the data then the user will click the Delete button.





The following query to Delete:

```

CREATE TRIGGER DeleteCategory
ON Category
FOR DELETE
AS
    DELETE FROM Product WHERE CategoryID IN (SELECT CategoryID FROM deleted)
    DELETE FROM Category WHERE CategoryID IN (SELECT CategoryID FROM deleted)

```

GO

```
DELETE FROM Category WHERE CategoryID = 'HP520'
```

8. Allow to extract useful information as described the previous assignment.

8.1 The product with the highest quantity on the orders

```

SELECT ProductName, OrderID, Buying_qty
FROM OrderDetail AS od INNER JOIN Product AS p ON (od.ProductID = p.ProductID)
WHERE buying_qty = (SELECT MAX(buying_qty) FROM OrderDetail)

```

	ProductName	OrderID	Buying_qty
1	Laptop Acer Nitro5	117	5

8.2 The product with the lowest quantity on the orders

```

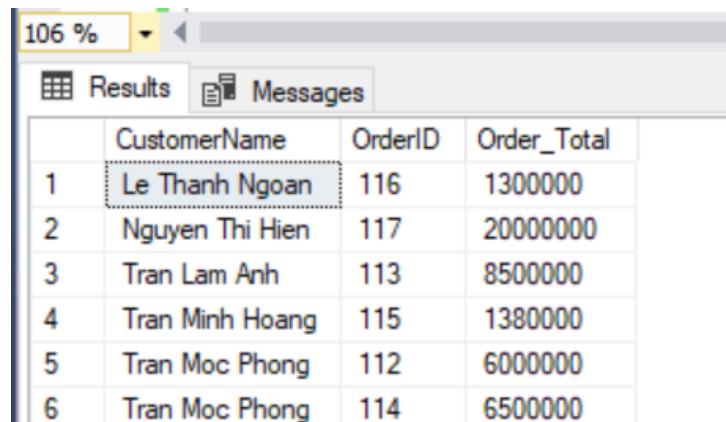
SELECT ProductName, OrderID, Buying_qty
FROM OrderDetail AS od INNER JOIN Product AS p ON (od.ProductID = p.ProductID)
WHERE buying_qty = (SELECT MIN(buying_qty) FROM OrderDetail)
    
```

	ProductName	OrderID	Buying_qty
1	Laptop Gaming Dell Inspiron 5577	115	1
2	Laptop Dell Vostro 3500	116	1

8.3 Total amount customers have ordered

```

SELECT CustomerName, o.OrderID, SUM(Price) as [Order_Total]
FROM Product AS p INNER JOIN OrderDetail AS os ON (os.ProductID = p.ProductID) INNER JOIN
orders AS o ON (os.orderID = o.OrderID) INNER JOIN Customer AS c ON (o.CustomerID =
c.CustomerID)
GROUP BY o.OrderID, CustomerName
    
```



	CustomerName	OrderID	Order_Total
1	Le Thanh Ngoan	116	1300000
2	Nguyen Thi Hien	117	20000000
3	Tran Lam Anh	113	8500000
4	Tran Minh Hoang	115	1380000
5	Tran Moc Phong	112	6000000
6	Tran Moc Phong	114	6500000

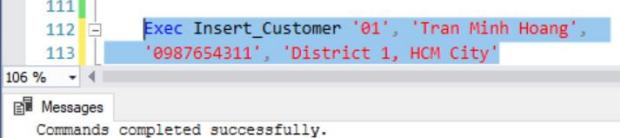
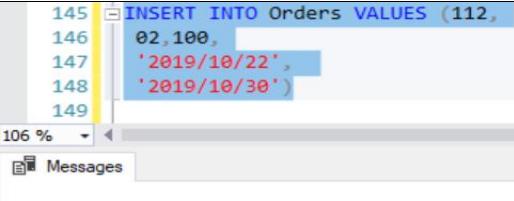
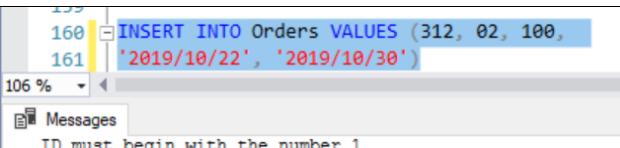
8.4 Display the name of the customer who ordered

```

SELECT CustomerName, ProductName, buying_qty
FROM [dbo].[Customer] c, [dbo].[Orders] o, [dbo].[OrderDetail] d, [dbo].[Product] p
WHERE c.CustomerID = o.CustomerID and o.OrderID = d.orderID and d.ProductID = p.ProductID
    
```

	CustomerName	ProductName	buying_qty
1	Tran Moc Phong	Laptop HP Probook	2
2	Tran Lam Anh	MacbookPro 2013	2
3	Tran Moc Phong	Laptop Acer E5_573G	3
4	Tran Minh Hoang	Laptop Gaming Dell Inspiron 5577	1
5	Le Thanh Ngoan	Laptop Dell Vostro 3500	1
6	Nguyen Thi Hien	Laptop Acer Nitro5	5

Chapter 3: Testing

No	Test Case	Function	Test Data	Execute
1	Verify that customer's information will save into customer table when user enters valid information	Adding Customer	<ul style="list-style-type: none"> -CustomerID: 01 -CustomerName: Tran Minh Hoang -Phone: 0987654311 -Address: District 1, HCM City 	 <pre>111 112 Exec Insert_Customer '01', 'Tran Minh Hoang', 113 '0987654311', 'District 1, HCM City' 106 % Messages Commands completed successfully.</pre>
2	Verify that orders information will save into orders table when user enters valid information	Adding Orders	<ul style="list-style-type: none"> -OrderID: 112 -CustomerID: 02 -StaffID: 100 -OrderDate: 2019/10/22 -Delivery: 2019/10/30 	 <pre>145 INSERT INTO Orders VALUES (112, 146 02,100, 147 '2019/10/22', 148 '2019/10/30') 149 106 % Messages (1 row affected)</pre>
3	Verify that the error message is displayed 'ID must begin with the number 1' and do not allow the user to enter the	Adding Orders	<ul style="list-style-type: none"> -OrderID: 112 -CustomerID: 02 -StaffID: 100 -OrderDate: 2019/10/22 -Delivery: 2019/10/30 	 <pre>155 156 INSERT INTO Orders VALUES (312, 02, 100, 157 '2019/10/22', '2019/10/30') 158 106 % Messages ID must begin with the number 1</pre>

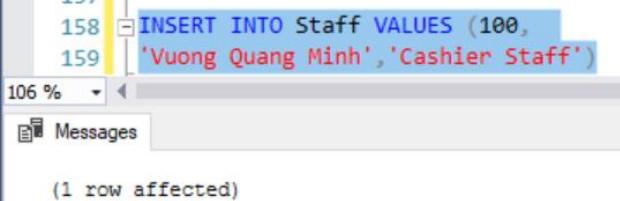
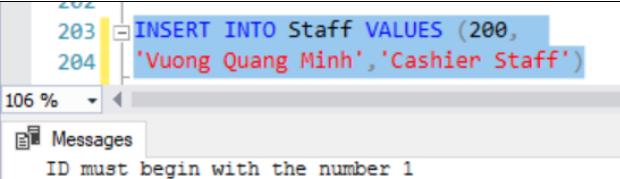
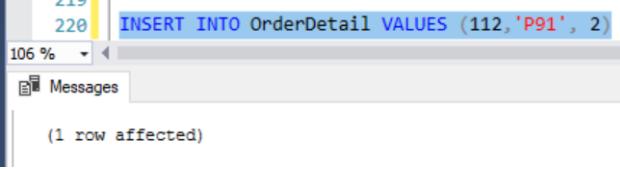
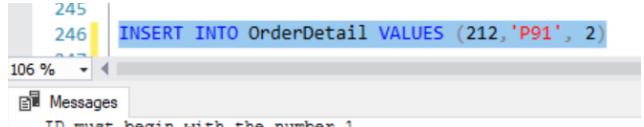
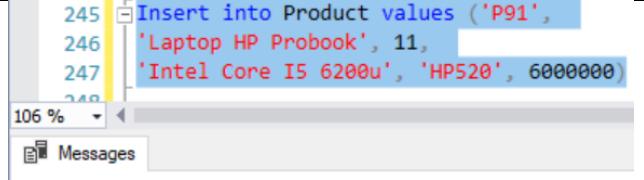
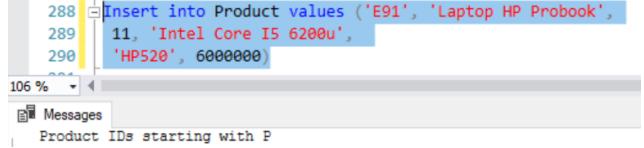
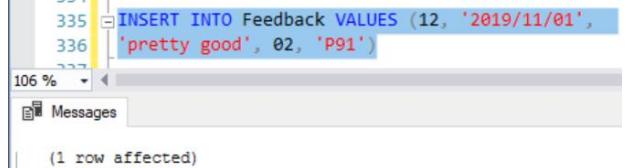
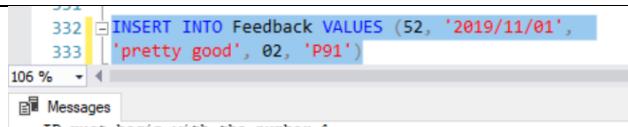
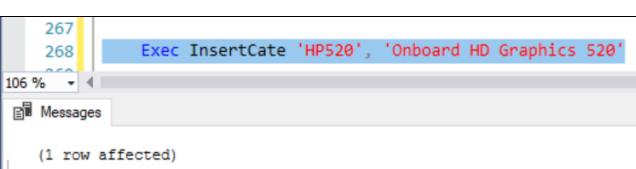
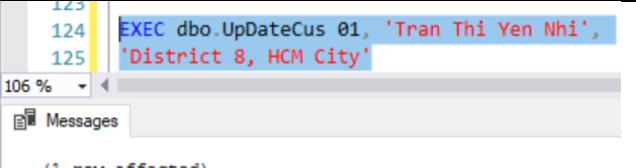
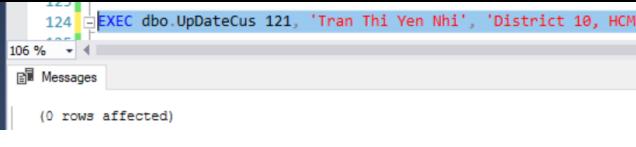
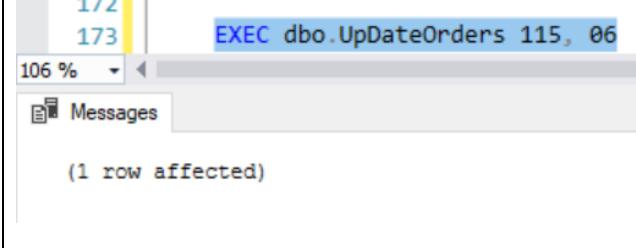
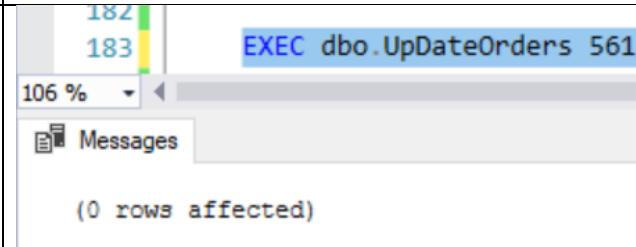
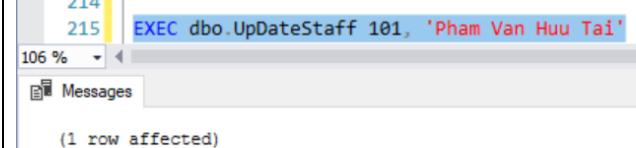
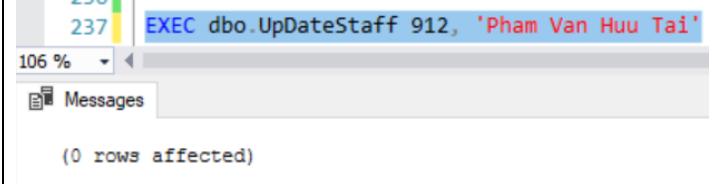
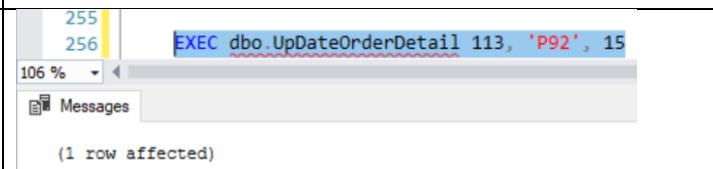
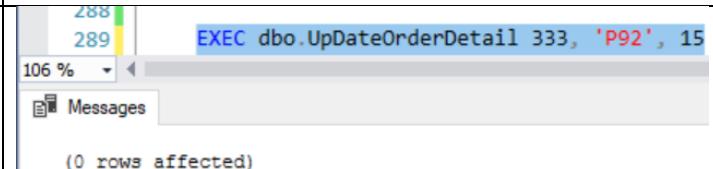
	wrong Orders ID that is already in the data.			
4	Verify that Staff's information will save into staff table when user enters valid information	Adding Staff	<ul style="list-style-type: none"> -StaffID: 100 -StaffName: Vuong Quang Minh -Position: Cashier Staff 	
5	Verify that the error message is displayed 'ID must begin with the number 1' and do not allow the user to enter the wrong Staff ID that is already in the data.	Adding Staff	<ul style="list-style-type: none"> -StaffID: 100 -StaffName: Vuong Quang Minh -Position: Cashier Staff 	
6	Verify that OrderDetail information will save into orderdetail	Adding OrderDetail	<ul style="list-style-type: none"> - OrderID: 112 - ProductID: P91 - buying_qty: 2 	

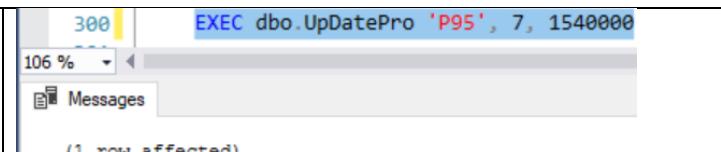
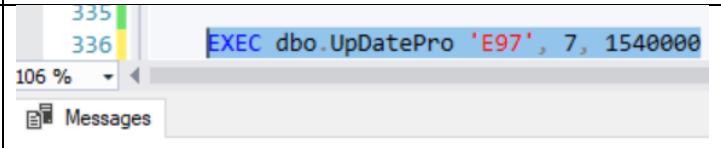
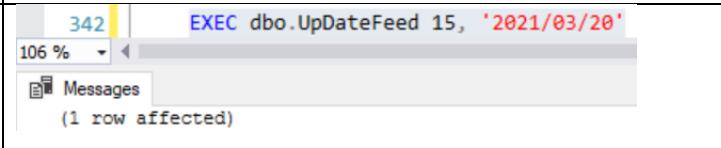
	table when user enters valid information			
7	Verify that the error message is displayed 'ID must begin with the number 1' and do not allow the user to enter incorrect Product information already in the data.	Adding OrderDetail	- OrderID: 112 -ProductID: P91 -buying_qty: 2	
8	Verify that Product's information will save into product table when user enters valid information	Adding Product	-ProductID: P91 -ProductName: Laptop Probook -Stock: 11 -Description: Intel Core I5 6200u -CategoryID: HP520 -Price: 6000000	
9	Verify that the error message is displayed	Adding Product	-ProductID: P91	

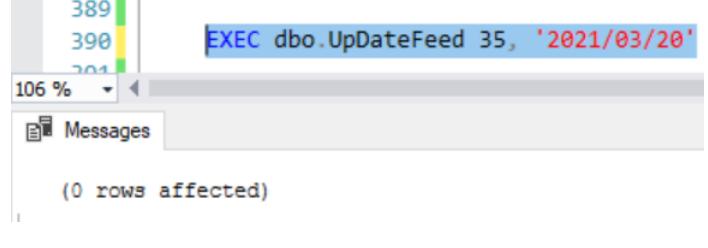
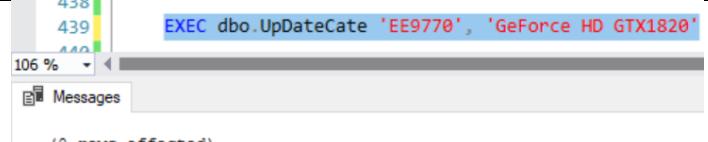
	'Product IDs starting with P' and do not allow the user to enter incorrect Product information already in the data.		-ProductName: Laptop HP Probook -Stock: 11 -Description: Intel Core I5 6200u -CategoryID: HP520 -Price: 6000000	
10	Verify that Feedback's information will save into feedback table when user enters valid information	Adding Feedback	-FeedbackID: 12 -FeedbackDate: 2019/11/01 -Content: pretty good -CustomerID: 02 -ProductID: P91	 <pre>335 [] INSERT INTO Feedback VALUES (12, '2019/11/01', 336 [] 'pretty good', 02, 'P91') 106 % <--> Messages (1 row affected)</pre>
11	Verify that the error message is displayed 'ID must begin with the number 1' and do not allow the user to enter incorrect Feedback information	Adding Feedback	-FeedbackID: 12 -FeedbackDate: 2019/11/01 -Content: pretty good -CustomerID: 02 -ProductID: P91	 <pre>332 [] INSERT INTO Feedback VALUES (52, '2019/11/01', 333 [] 'pretty good', 02, 'P91') 106 % <--> Messages ID must begin with the number 1</pre>

	already in the data.			
12	Verify that Category's information will save into category table when user enters valid information	Adding Category	-CategoryID: HP520 -CategoryName: Onboard HD Graphics 520	
13	Verify that customer's information will update into the Customer table when the user enters valid information	Update Customer	-CustomerID: 01 -CustomerName: Tran Thi Yen Nhi -Phone: 0987654311 -Address: District 8, HCM City	
14	Verify that customer information will not update to the Customers table when the user enters an invalid ID	Update Customer	-CustomerID: 121 -CustomerName: Tran Thi Yen Nhi -Phone: 0987654311 -Address: District 10, HCM City	

15	Verify that Orders information will update into the Orders table when the user enters valid information	Update Orders	-OrderID: 115 -CustomerID: 06 -StaffID: 100 -OrderDate: 2021/01/05 -Delivery: 2021/01/15	
16	Verify that orders information will not update to the Orders table when the user enters an invalid ID	Update Orders	-OrderID: 561 -CustomerID: 06 -StaffID: 100 -OrderDate: 2021/01/05 -Delivery: 2021/01/15	
17	Verify that Staff's information will update into the Staff table when the user enters valid information	Update Staff	-StaffID: 101 -StaffName: Pham Van Huu Tai -Position: Marketing Staff	

18	Verify that Staff information will not update to the Staff table when the user enters an invalid ID	Update Staff	-StaffID: 912 -StaffName: Pham Van Huu Tai -Position: Marketing Staff	
19	Verify that OrderDetail information will update into the OrderDetail table when the user enters valid information	Update OrderDetail	- OrderID: 113 -ProductID: P92 -buying_qty: 15	
20	Verify that Orderdetail information will not update to the OrderDetail table when the user enters an invalid ID	Update OrderDetail	- OrderID: 333 -ProductID: P92 -buying_qty: 15	

21	Verify that Product's information will update into the Product table when the user enters valid information	Update Product	-ProductID: P95 -ProductName: Laptop Dell Vostro 3500 -Stock: 7 -Description: Intel Core I5 1135G7 -CategoryID: DV35 -Price: 1540000	
22	Verify that Product information will not update to the Product table when the user enters an invalid ID	Update Product	-ProductID: E97 -ProductName: Laptop Dell Vostro 3500 -Stock: 7 -Description: Intel Core I5 1135G7 -CategoryID: DV35 -Price: 1540000	
23	Verify that Feedback's information will update into the Feedback table when the user enters valid information	Update Feedback	-FeedbackID: 15 -FeedbackDate: 2021/03/20 -Content: pretty good -CustomerID: 01 -ProductID: P94	

24	Verify that Feedback information will not update to the Feedback table when the user enters an invalid ID	Update Feedback	-FeedbackID: 35 -FeedbackDate: 2021/03/20 -Content: pretty good -CustomerID: 01 -ProductID: P94	
25	Verify that Category information will update into the Category table when the user enters valid information	Update Category	-CategoryID: GM5577 -CategoryName: GeForce HD GTX1820	
26	Verify that Category information will not update to the Category table when the user enters an invalid ID	Update Category	-CategoryID: EE9770 -CategoryName: GeForce HD GTX1820	

References

greenwich, 2022. *greenwich*. [Online]

Available at: <https://flm.greenwich.edu.vn/gui/role/student/SyllabusDetails?syllID=2597>

[Accessed 9 Dec 2021].