# ASSIGNMENT 2 FRONT SHEET

| | | | |
|---|---|---|---|
| **Qualification** | **TEC Level 5 HND Diploma in Computing** | | |
| **Unit number and title** | **Unit 04: Database Design & Development** | | |
| **Submission date** | 04/05/2022 | **Date Received 1st submission** | 04/05/2022 |
| **Re-submission Date** | 04/05/2022 | **Date Received 2nd submission** | 04/05/2022 |
| **Student Name** | Bùi Hương Linh | **Student ID** | GBH200662 |
| **Class** | GCH1002 | **Assessor name** | Hàn Minh Phương |
| **Student declaration**<br><br>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: | | |

## Table of Contents

# Chapter 1 - Develop the database system

1. Final Mock-up of the application


**Figure 1: Interfaces Home page**

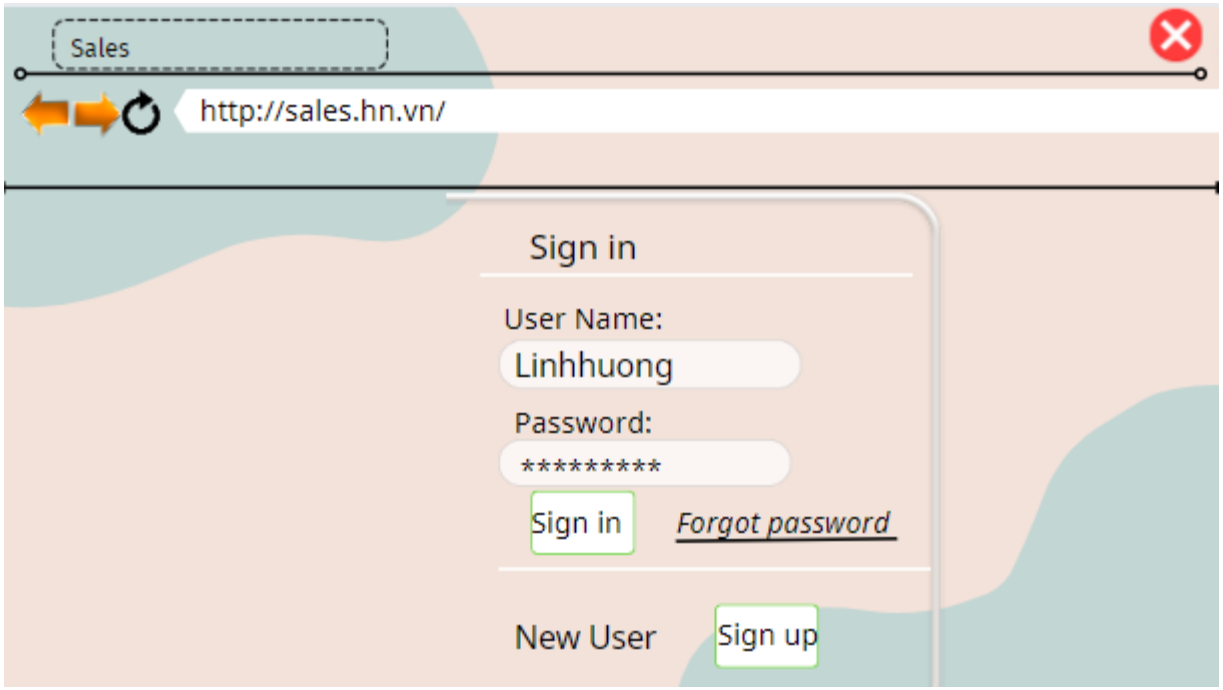



**Figure 2: Interfaces Login**

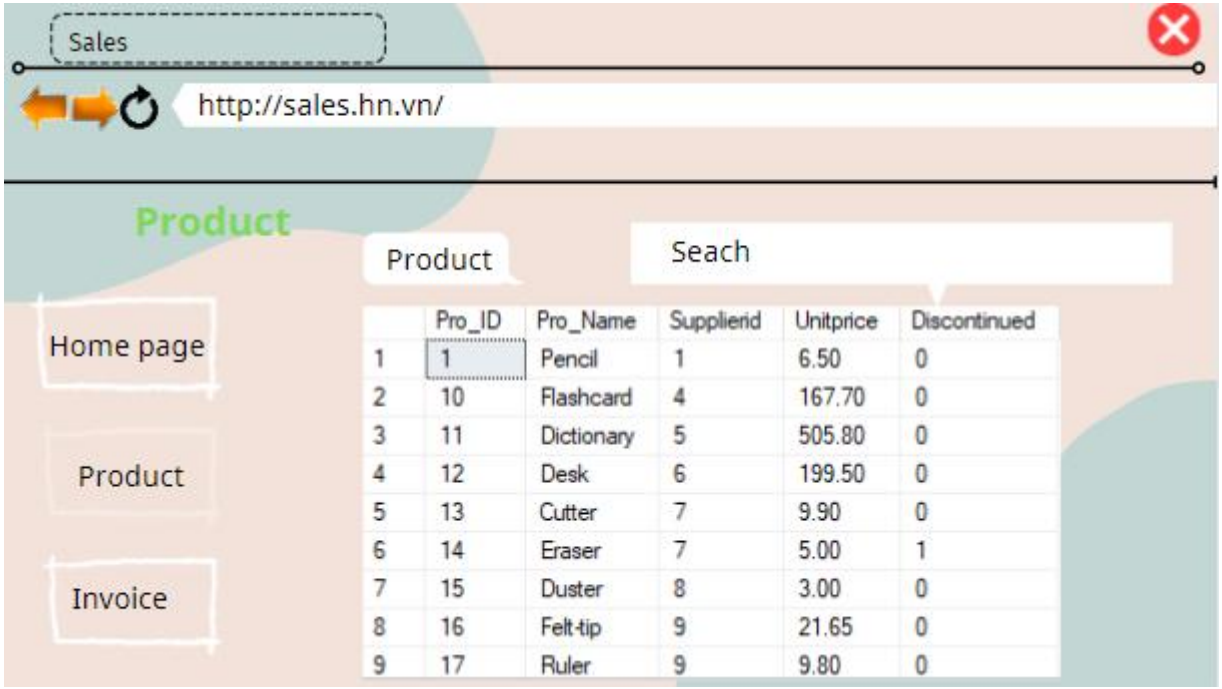
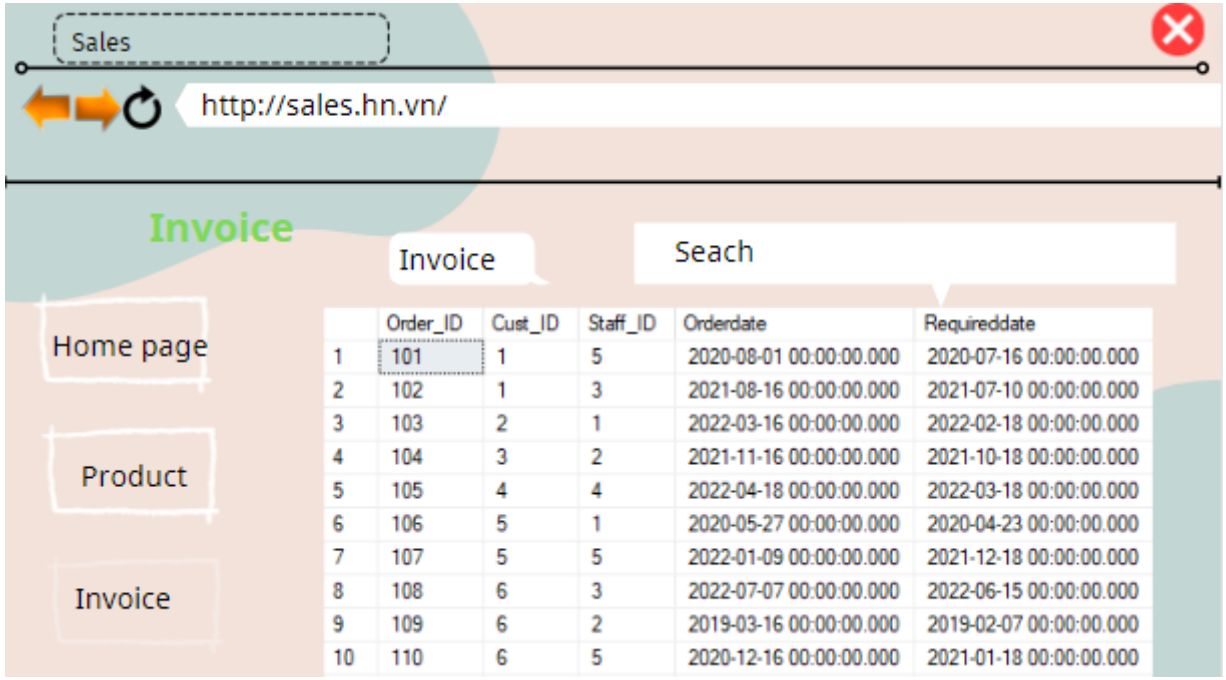



**Figure 3:Interface Product**




| | Order_ID | Cust_ID | Staff_ID | Orderdate | Requireddate |
|---|---|---|---|---|---|
| 1 | 101 | 1 | 5 | 2020-08-01 00:00:00.000 | 2020-07-16 00:00:00.000 |
| 2 | 102 | 1 | 3 | 2021-08-16 00:00:00.000 | 2021-07-10 00:00:00.000 |
| 3 | 103 | 2 | 1 | 2022-03-16 00:00:00.000 | 2022-02-18 00:00:00.000 |
| 4 | 104 | 3 | 2 | 2021-11-16 00:00:00.000 | 2021-10-18 00:00:00.000 |
| 5 | 105 | 4 | 4 | 2022-04-18 00:00:00.000 | 2022-03-18 00:00:00.000 |
| 6 | 106 | 5 | 1 | 2020-05-27 00:00:00.000 | 2020-04-23 00:00:00.000 |
| 7 | 107 | 5 | 5 | 2022-01-09 00:00:00.000 | 2021-12-18 00:00:00.000 |
| 8 | 108 | 6 | 3 | 2022-07-07 00:00:00.000 | 2022-06-15 00:00:00.000 |
| 9 | 109 | 6 | 2 | 2019-03-16 00:00:00.000 | 2019-02-07 00:00:00.000 |
| 10 | 110 | 6 | 5 | 2020-12-16 00:00:00.000 | 2021-01-18 00:00:00.000 |

**Figure 4: Interface Invoice**

2. Queries to create database with results

```
create database Sale_Managers
go

use Sale_Managers
go

CREATE TABLE Customers
(
  Cust_ID       INT          NOT NULL IDENTITY,
  Companyname   NVARCHAR(40) NOT NULL,
  Contactname   NVARCHAR(30) NOT NULL,
  Address       NVARCHAR(60) NOT NULL,
  Country       NVARCHAR(15) NOT NULL,
  Phone         NVARCHAR(24) NOT NULL,
  CONSTRAINT PK_Customers PRIMARY KEY(Cust_ID)
);
```

```
%
Messages
Commands completed successfully.

Completion time: 2022-05-04T22:24:52.5454966+07:00
```

**Figure 5: Create Database**

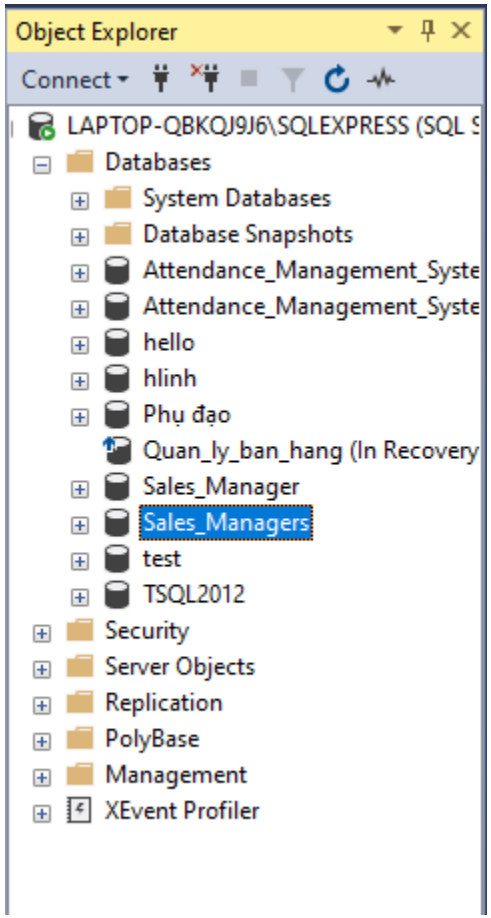



**Figure 6: Show database on toolbar**

```
CREATE TABLE Customers
(
  Cust_ID       INT          NOT NULL ,
  Companyname   NVARCHAR(40) NOT NULL,
  Contactname   NVARCHAR(30) NOT NULL,
  Address       NVARCHAR(60) NOT NULL,
  Country       NVARCHAR(15) NOT NULL,
  Phone         NVARCHAR(24) NOT NULL,
  CONSTRAINT PK_Customers PRIMARY KEY(Cust_ID)
);
```

100 %

Messages

```
Commands completed successfully.

Completion time: 2022-05-05T14:59:10.5865077+07:00
```

**Figure 7: Create table Customer**

```
CREATE TABLE Staff
(
  Staff_ID     nvarchar(10) NOT NULL,
  Lastname        NVARCHAR(20) NOT NULL,
  Firstname       NVARCHAR(10) NOT NULL,
  Birthdate       DATETIME     NOT NULL,
  Hiredate        DATETIME     NOT NULL,
  Address         NVARCHAR(60) NOT NULL,
  City            NVARCHAR(15) NOT NULL,
  Phone           NVARCHAR(24) NOT NULL,
  CONSTRAINT PK_Staff PRIMARY KEY(Staff_ID),
  CONSTRAINT CHK_Birthdate CHECK(Birthdate <= CURRENT_TIMESTAMP)
);
```

100 %

Messages

```
Commands completed successfully.

Completion time: 2022-05-05T15:00:01.9680205+07:00
```

**Figure 8: Create table Staff**

```
CREATE TABLE Product
(
  Pro_ID nvarchar(10) not null,
  Pro_Name  NVARCHAR(40) NOT NULL,
  Supplierid   INT          NOT NULL,
  Unitprice    MONEY        NOT NULL
    CONSTRAINT DFT_Product_Unitprice DEFAULT(0),
  Discontinued BIT          NOT NULL
    CONSTRAINT DFT_Product_Discontinued DEFAULT(0),
  CONSTRAINT PK_Product PRIMARY KEY(Pro_ID),
  CONSTRAINT CHK_Product_Unitprice CHECK(Unitprice >= 0)
);
```

100 %

Messages

```
Commands completed successfully.

Completion time: 2022-05-05T15:00:51.0121555+07:00
```

**Figure 9: Create table Product**

```
CREATE TABLE Invoice
(
  Order_ID        INT          NOT NULL ,
  Cust_ID         INT          NULL,
  Staff_ID     nvarchar(10) NOT NULL,
  Orderdate      DATETIME     NOT NULL,
  Requireddate   DATETIME     NOT NULL,
  CONSTRAINT PK_Invoice PRIMARY KEY(Order_ID),
  CONSTRAINT FK_Invoice_Customers FOREIGN KEY( Cust_ID )
    REFERENCES Customers( Cust_ID ),
  CONSTRAINT FK_Invoice_Staff FOREIGN KEY(Staff_ID)
    REFERENCES Staff(Staff_ID),

  );
```

100 %

Messages

```
Commands completed successfully.

Completion time: 2022-05-05T15:01:19.0093866+07:00
```

**Figure 10: Create table Invoice**

```
CREATE TABLE InvoiceDetails
(
Order_ID        INT          NOT NULL,
Pro_ID nvarchar(10) not null,
Unitprice MONEY         NOT NULL
  CONSTRAINT DFT_InvoiceDetails_Unitprice DEFAULT(0),
Qty       SMALLINT      NOT NULL
  CONSTRAINT DFT_InvoiceDetails_Qty DEFAULT(1),
Discount  NUMERIC(4, 3) NOT NULL
  CONSTRAINT DFT_OrderDetails_discount DEFAULT(0),
CONSTRAINT PK_InvoiceDetails PRIMARY KEY(Order_ID, Pro_ID),
CONSTRAINT FK_InvoiceDetails_Orders FOREIGN KEY(Order_ID)
  REFERENCES Invoice(Order_ID),
CONSTRAINT FK_InvoiceDetails_Product FOREIGN KEY(Pro_ID)
  REFERENCES Product( Pro_ID),
CONSTRAINT CHK_Discount  CHECK (discount BETWEEN 0 AND 1),
CONSTRAINT CHK_Qty  CHECK (qty > 0),
CONSTRAINT CHK_Unitprice CHECK (unitprice >= 0)
);
```

```
33 %
Messages
Commands completed successfully.

Completion time: 2022-05-05T15:02:04.9978379+07:00
```

**Figure 11: Create table InvoiceDetails**

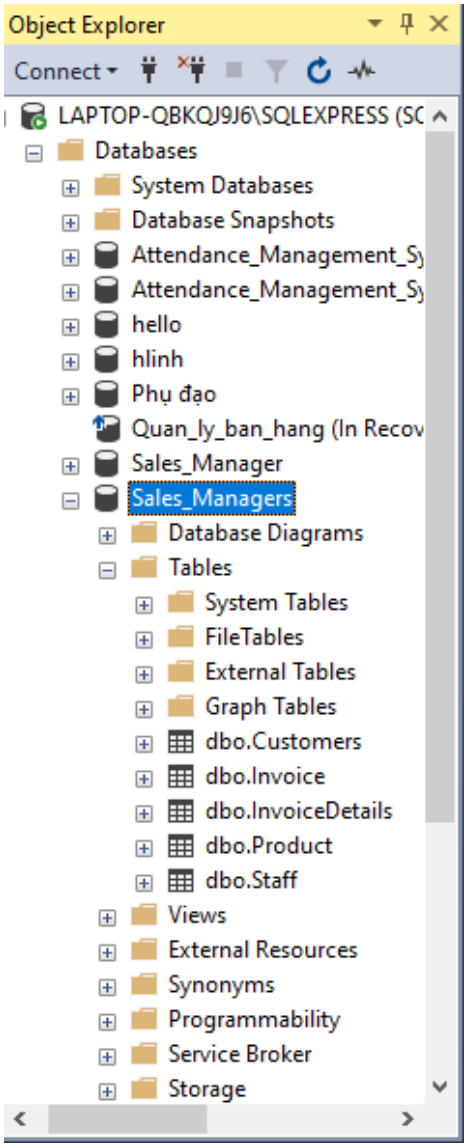



**Figure 12: Show all panels in toolbar**

# Chapter 2 – Produce queries

1. Queries to INSERT data with illustrations of final result

```
--Populate table Customer
INSERT INTO Customers (Cust_ID, Companyname, Contactname, Address, Country, Phone)
VALUES ('1',N'Allen, Michael', N'Sales Representative', 'Dong Da Str', ' Vietname', '0972208243')
INSERT INTO Customers (Cust_ID, Companyname, Contactname, Address, Country, Phone)
VALUES ('2', N'Customer KBUDE', N'Peoples, John', N'Mataderos  7890', 'China', '0972846198')
INSERT INTO Customers (Cust_ID, Companyname, Contactname, Address, Country, Phone)
VALUES ('3', N'Customer HFBZG', N'Arndt, Torsten',  N'7890 Hanover Sq.', N'London', '1714567890')
INSERT INTO Customers (Cust_ID, Companyname, Contactname, Address, Country, Phone)
VALUES ('4', N'Customer XHXJV', N'Poland, Carole',N'Forsterstr. 7890', 'Germany' , N'0621554458')
INSERT INTO Customers (Cust_ID, Companyname, Contactname, Address, Country, Phone)
VALUES ('5', N'Customer LVJSO', N'Smith, Denise', N'City Center Plaza 2345 Main St.',  N'USA', N'5035550126')
INSERT INTO Customers (Cust_ID, Companyname, Contactname, Address, Country, Phone)
VALUES ('6', N'Customer RTXGC', N'Raghav, Amritansh', N'6789, rue des Bouchers', N'France', N'05215665686')
INSERT INTO Customers (Cust_ID, Companyname, Contactname, Address, Country, Phone)
VALUES ('7', N'Customer XPNIK', N'Dressler, Marlies', N'Carrera 7890 con Ave. Bolívar #65-98 Llano Largo', N'Venezuela', N'2658510445')
INSERT INTO Customers (Cust_ID, Companyname, Contactname, Address, Country, Phone)
VALUES ('8', N'Customer GCJSG', N'Mallit, Ken', N'South House 1234 Queensbridge', N'UK', N'1718901234')
INSERT INTO Customers (Cust_ID, Companyname, Contactname, Address, Country, Phone)
VALUES ('9', N'Customer SNXOJ', N'Syamala, Manoj', N'Adenauerallee 7890', N'Germany', N'0711345678')
INSERT INTO Customers (Cust_ID, Companyname, Contactname, Address, Country, Phone)
VALUES ('10', N'Customer CCFIZ', N'Conn, Steve', N'ul. Filtrowa 6789', N'USA', N'2065550123')
```

91 %

Messages

```
(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)
```

**Figure 13: Queries to INSERT Customer**

Results | Messages

| | Cust_ID | Companyname | Contactname | Address | Country | Phone |
|---|---|---|---|---|---|---|
| 1 | 1 | Allen, Michael | Sales Representative | Dong Da Str | Vietname | 0972208243 |
| 2 | 2 | Customer KBUDE | Peoples, John | Mataderos 7890 | China | 0972846198 |
| 3 | 3 | Customer HFBZG | Arndt, Torsten | 7890 Hanover Sq. | London | 1714567890 |
| 4 | 4 | Customer XHXJV | Poland, Carole | Forsterstr. 7890 | Germany | 0621554458 |
| 5 | 5 | Customer LVJSO | Smith, Denise | City Center Plaza 2345 Main St. | USA | 5035550126 |
| 6 | 6 | Customer RTXGC | Raghav, Amritansh | 6789, rue des Bouchers | France | 05215665686 |
| 7 | 7 | Customer XPNIK | Dressler, Marlies | Carrera 7890 con Ave. Bolívar #65-98 Llano Largo | Venezuela | 2658510445 |
| 8 | 8 | Customer GCJSG | Mallit, Ken | South House 1234 Queensbridge | UK | 1718901234 |
| 9 | 9 | Customer SNXOJ | Syamala, Manoj | Adenauerallee 7890 | Germany | 0711345678 |
| 10 | 10 | Customer CCFIZ | Conn, Steve | ul. Filtrowa 6789 | USA | 2065550123 |

Query executed successfully. | LAPTOP-QBKQJ9J6\SQLEXPRESS ... | LAPTOP-QBKQJ9J6\Admin ... | Sales_Managers | 00:00:00 | 10 rows

**Figure 14: Show all Customer table data**

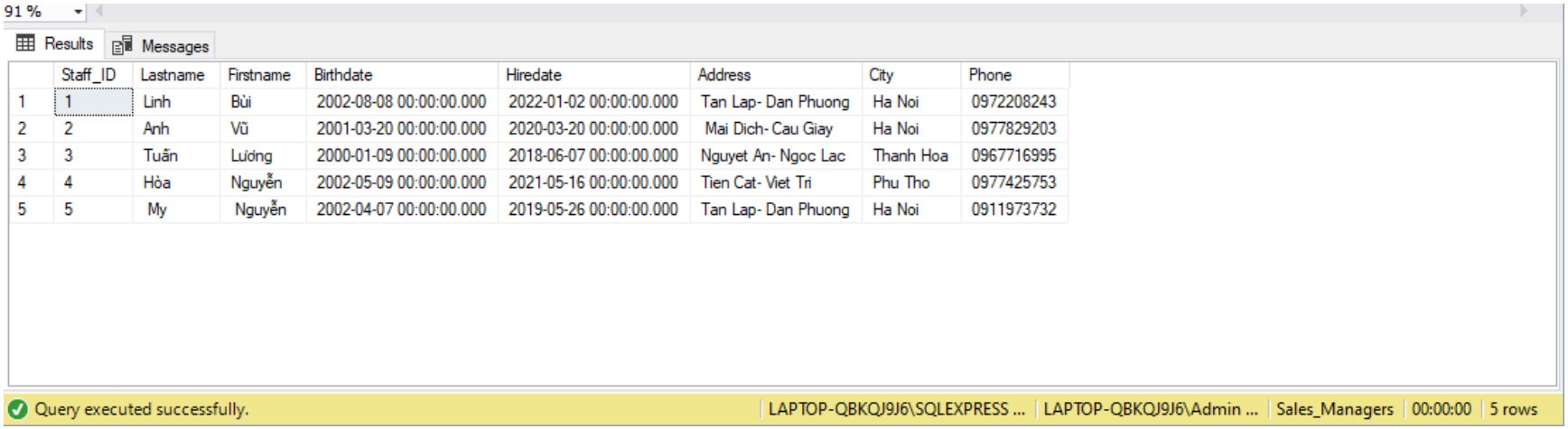
```
--Populate table Staff
INSERT INTO Staff (Staff_ID, Lastname, Firstname, Birthdate, Hiredate, Address, City, Phone)
VALUES ('1', N'Linh', N'Bùi', N'20020808 00:00:00.000',  N'20220102 00:00:00.000', 'Tan Lap- Dan Phuong', 'Ha Noi', '0972208243')
INSERT INTO Staff (Staff_ID, Lastname, Firstname, Birthdate, Hiredate, Address, City, Phone)
VALUES ('2', N'Anh', N'Vũ', N'20010320 00:00:00.000', N'20200320 00:00:00.000', ' Mai Dich- Cau Giay', 'Ha Noi','0977829203')
INSERT INTO Staff (Staff_ID, Lastname, Firstname, Birthdate, Hiredate, Address, City, Phone)
VALUES ('3', N'Tuấn' , N'Lương', N'20000109 00:00:00.000', N' 20180607 00:00:00.000', 'Nguyet An- Ngoc Lac', 'Thanh Hoa', '0967716995')
INSERT INTO Staff (Staff_ID, Lastname, Firstname, Birthdate, Hiredate, Address, City, Phone)
VALUES ('4', N'Hòa', N'Nguyễn', N'20020509 00:00:00.000', N'20210516 00:00:00.000', 'Tien Cat- Viet Tri' , 'Phu Tho', '0977425753')
INSERT INTO Staff (Staff_ID, Lastname, Firstname, Birthdate, Hiredate, Address, City, Phone)
VALUES ('5', N' My', N' Nguyễn', N'20020407 00:00:00.000', N'20190526 00:00:00.000', 'Tan Lap- Dan Phuong', 'Ha Noi', '0911973732')

--Populate table Product
INSERT INTO Product (Pno_ID, Pno_Name, Supplierid, Unitprice, Discontinued )
```

91 %

Messages

```
(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2022-05-05T15:05:14.0695777+07:00
```

**Figure 15: Queries to INSERT Staff**


91 %

Results | Messages

| | Staff_ID | Lastname | Firstname | Birthdate | Hiredate | Address | City | Phone |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | Linh | Bùi | 2002-08-08 00:00:00.000 | 2022-01-02 00:00:00.000 | Tan Lap- Dan Phuong | Ha Noi | 0972208243 |
| 2 | 2 | Anh | Vũ | 2001-03-20 00:00:00.000 | 2020-03-20 00:00:00.000 | Mai Dich- Cau Giay | Ha Noi | 0977829203 |
| 3 | 3 | Tuấn | Lương | 2000-01-09 00:00:00.000 | 2018-06-07 00:00:00.000 | Nguyet An- Ngoc Lac | Thanh Hoa | 0967716995 |
| 4 | 4 | Hòa | Nguyễn | 2002-05-09 00:00:00.000 | 2021-05-16 00:00:00.000 | Tien Cat- Viet Tri | Phu Tho | 0977425753 |
| 5 | 5 | My | Nguyễn | 2002-04-07 00:00:00.000 | 2019-05-26 00:00:00.000 | Tan Lap- Dan Phuong | Ha Noi | 0911973732 |

Query executed successfully. | LAPTOP-QBKQJ9J6\SQLEXPRESS ... | LAPTOP-QBKQJ9J6\Admin ... | Sales_Managers | 00:00:00 | 5 rows

**Figure 16: Show all Staff  table data**

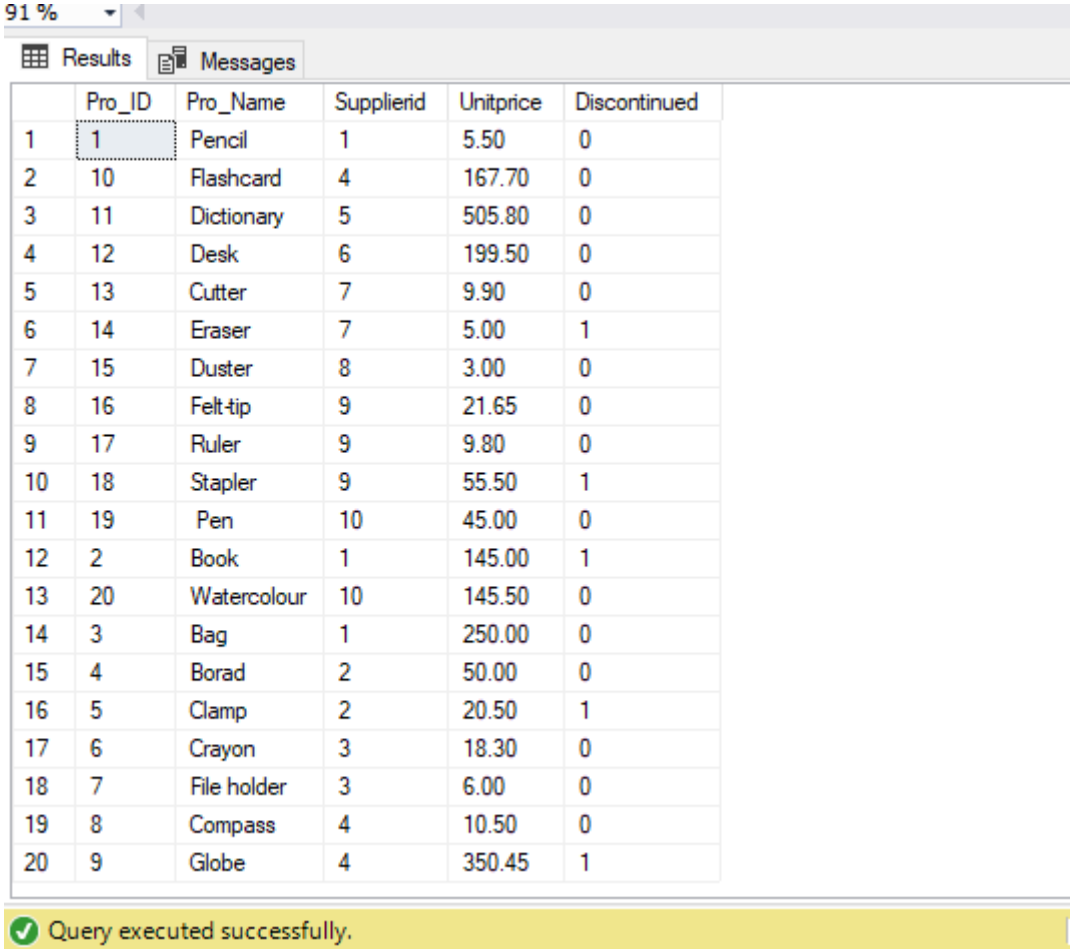
```
--Populate table Product
INSERT INTO Product (Pro_ID, Pro_Name, Supplierid, Unitprice, Discontinued )
VALUES  ('1', 'Pencil', 1, 5.500, 0)
INSERT INTO Product (Pro_ID, Pro_Name, Supplierid, Unitprice, Discontinued )
VALUES  ('2', 'Book', 1, 145.000, 1)
INSERT INTO Product (Pro_ID, Pro_Name, Supplierid, Unitprice, Discontinued )
VALUES  ('3', 'Bag', 1, 250.000, 0)
INSERT INTO Product (Pro_ID, Pro_Name, Supplierid, Unitprice, Discontinued )
VALUES  ('4', 'Borad', 2, 50.000, 0)
INSERT INTO Product (Pro_ID, Pro_Name, Supplierid, Unitprice, Discontinued )
VALUES  ('5', 'Clamp', 2, 20.500, 1)
INSERT INTO Product (Pro_ID, Pro_Name, Supplierid, Unitprice, Discontinued )
VALUES  ('6', 'Crayon', 3, 18.300, 0)
INSERT INTO Product (Pro_ID, Pro_Name, Supplierid, Unitprice, Discontinued )
VALUES  ('7', 'File holder', 3, 6.000, 0)
INSERT INTO Product (Pro_ID, Pro_Name, Supplierid, Unitprice, Discontinued )
VALUES  ('8', 'Compass', 4, 10.500, 0)
INSERT INTO Product (Pro_ID, Pro_Name, Supplierid, Unitprice, Discontinued )
VALUES  ('9', 'Globe', 4, 350.450, 1)
INSERT INTO Product (Pro_ID, Pro_Name, Supplierid, Unitprice, Discontinued )
```

91 %

Messages

```
(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)
```

**Figure 17: Queries to INSERT Product**



91 %

Results | Messages

| | Pro_ID | Pro_Name | Supplierid | Unitprice | Discontinued |
|---|---|---|---|---|---|
| 1 | 1 | Pencil | 1 | 5.50 | 0 |
| 2 | 10 | Flashcard | 4 | 167.70 | 0 |
| 3 | 11 | Dictionary | 5 | 505.80 | 0 |
| 4 | 12 | Desk | 6 | 199.50 | 0 |
| 5 | 13 | Cutter | 7 | 9.90 | 0 |
| 6 | 14 | Eraser | 7 | 5.00 | 1 |
| 7 | 15 | Duster | 8 | 3.00 | 0 |
| 8 | 16 | Felt-tip | 9 | 21.65 | 0 |
| 9 | 17 | Ruler | 9 | 9.80 | 0 |
| 10 | 18 | Stapler | 9 | 55.50 | 1 |
| 11 | 19 | Pen | 10 | 45.00 | 0 |
| 12 | 2 | Book | 1 | 145.00 | 1 |
| 13 | 20 | Watercolour | 10 | 145.50 | 0 |
| 14 | 3 | Bag | 1 | 250.00 | 0 |
| 15 | 4 | Borad | 2 | 50.00 | 0 |
| 16 | 5 | Clamp | 2 | 20.50 | 1 |
| 17 | 6 | Crayon | 3 | 18.30 | 0 |
| 18 | 7 | File holder | 3 | 6.00 | 0 |
| 19 | 8 | Compass | 4 | 10.50 | 0 |
| 20 | 9 | Globe | 4 | 350.45 | 1 |

Query executed successfully.

**Figure 18: Show all Product table data**

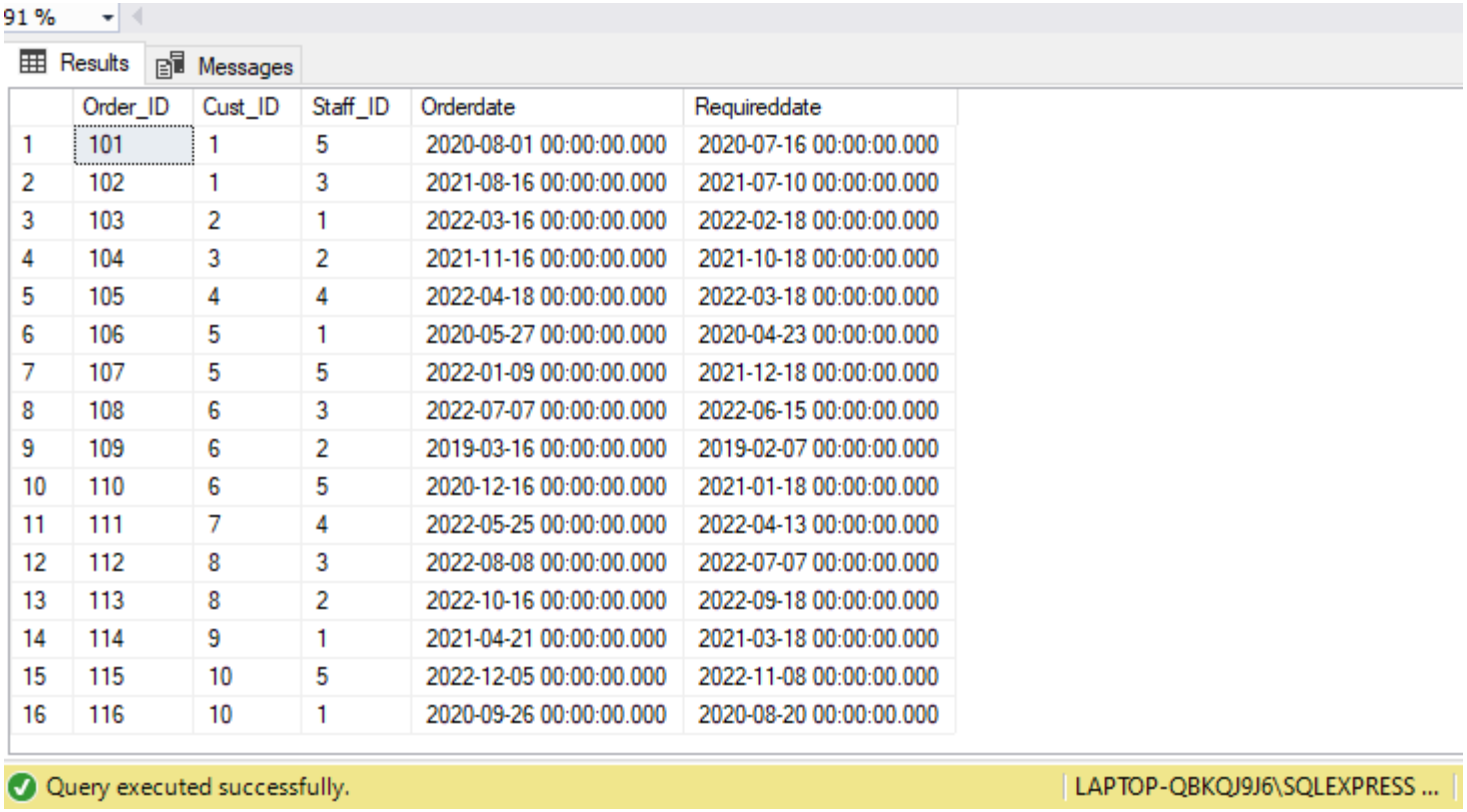
```
--Populate table Invoice
INSERT INTO Invoice(Order_ID, Cust_ID, Staff_ID, Orderdate, Requireddate)
VALUES  (101, 1, 5, '20200801 00:00:00.000', '20200716 00:00:00.000' )
INSERT INTO Invoice(Order_ID, Cust_ID, Staff_ID, Orderdate, Requireddate)
VALUES  (102, 1, 3, '20210816 00:00:00.000', '20210710 00:00:00.000' )
INSERT INTO Invoice(Order_ID, Cust_ID, Staff_ID, Orderdate, Requireddate)
VALUES  (103, 2, 1, '20220316 00:00:00.000', '20220218 00:00:00.000')
INSERT INTO Invoice(Order_ID, Cust_ID, Staff_ID, Orderdate, Requireddate)
VALUES  (104, 3, 2, '20211116 00:00:00.000', '20211018 00:00:00.000')
INSERT INTO Invoice(Order_ID, Cust_ID, Staff_ID, Orderdate, Requireddate)
VALUES  (105, 4, 4, '20220418 00:00:00.000', '20220318 00:00:00.000')
INSERT INTO Invoice(Order_ID, Cust_ID, Staff_ID, Orderdate, Requireddate)
VALUES  (106, 5, 1, '20200527 00:00:00.000', '20200423 00:00:00.000')
INSERT INTO Invoice(Order_ID, Cust_ID, Staff_ID, Orderdate, Requireddate)
VALUES  (107, 5, 5, '20220109 00:00:00.000', '20211218 00:00:00.000')
INSERT INTO Invoice(Order_ID, Cust_ID, Staff_ID, Orderdate, Requireddate)
VALUES  (108, 6, 3, '20220707 00:00:00.000', '20220615 00:00:00.000')
INSERT INTO Invoice(Order_ID, Cust_ID, Staff_ID, Orderdate, Requireddate)
VALUES  (109, 6, 2, '20190316 00:00:00.000', '20190207 00:00:00.000')
```

91 %

Messages

```
(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)
```

**Figure 19: Queries to INSERT Invoice**



91 %

Results | Messages

| | Order_ID | Cust_ID | Staff_ID | Orderdate | Requireddate |
|---|---|---|---|---|---|
| 1 | 101 | 1 | 5 | 2020-08-01 00:00:00.000 | 2020-07-16 00:00:00.000 |
| 2 | 102 | 1 | 3 | 2021-08-16 00:00:00.000 | 2021-07-10 00:00:00.000 |
| 3 | 103 | 2 | 1 | 2022-03-16 00:00:00.000 | 2022-02-18 00:00:00.000 |
| 4 | 104 | 3 | 2 | 2021-11-16 00:00:00.000 | 2021-10-18 00:00:00.000 |
| 5 | 105 | 4 | 4 | 2022-04-18 00:00:00.000 | 2022-03-18 00:00:00.000 |
| 6 | 106 | 5 | 1 | 2020-05-27 00:00:00.000 | 2020-04-23 00:00:00.000 |
| 7 | 107 | 5 | 5 | 2022-01-09 00:00:00.000 | 2021-12-18 00:00:00.000 |
| 8 | 108 | 6 | 3 | 2022-07-07 00:00:00.000 | 2022-06-15 00:00:00.000 |
| 9 | 109 | 6 | 2 | 2019-03-16 00:00:00.000 | 2019-02-07 00:00:00.000 |
| 10 | 110 | 6 | 5 | 2020-12-16 00:00:00.000 | 2021-01-18 00:00:00.000 |
| 11 | 111 | 7 | 4 | 2022-05-25 00:00:00.000 | 2022-04-13 00:00:00.000 |
| 12 | 112 | 8 | 3 | 2022-08-08 00:00:00.000 | 2022-07-07 00:00:00.000 |
| 13 | 113 | 8 | 2 | 2022-10-16 00:00:00.000 | 2022-09-18 00:00:00.000 |
| 14 | 114 | 9 | 1 | 2021-04-21 00:00:00.000 | 2021-03-18 00:00:00.000 |
| 15 | 115 | 10 | 5 | 2022-12-05 00:00:00.000 | 2022-11-08 00:00:00.000 |
| 16 | 116 | 10 | 1 | 2020-09-26 00:00:00.000 | 2020-08-20 00:00:00.000 |

Query executed successfully. | LAPTOP-QBKQJ9J6\SQLEXPRESS ...

**Figure 20: Show all Invoice table data**

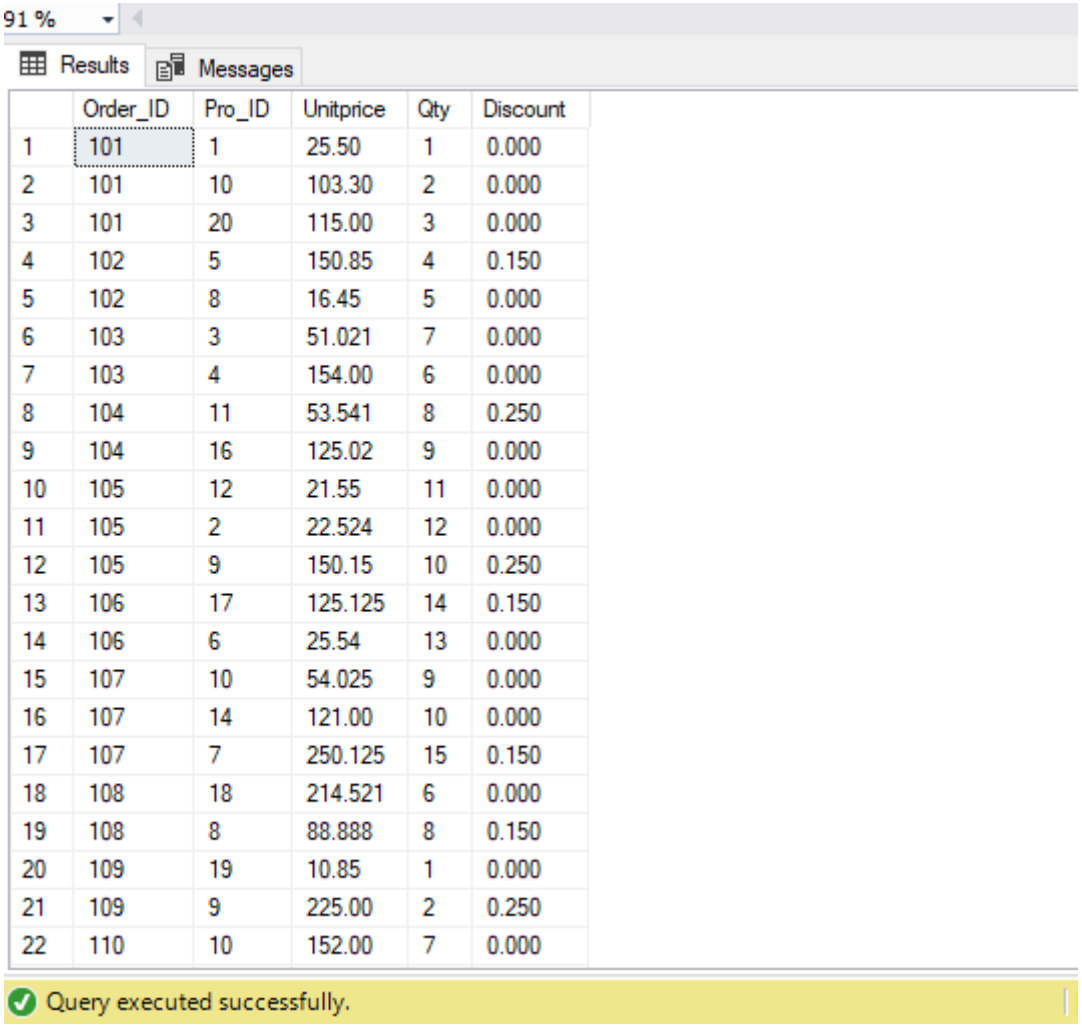
```
--Populate table InvoiceDetails
INSERT INTO InvoiceDetails (Order_ID, Pro_ID, Qty, Unitprice, Discount )
VALUES (101, 1, 1, 25.500, 0)
INSERT INTO InvoiceDetails (Order_ID, Pro_ID, Qty, Unitprice, Discount )
VALUES (101, 10, 2, 103.300, 0)
INSERT INTO InvoiceDetails (Order_ID, Pro_ID, Qty, Unitprice, Discount )
VALUES (101, 20, 3, 115.000, 0)
INSERT INTO InvoiceDetails (Order_ID, Pro_ID, Qty, Unitprice, Discount )
VALUES (102, 5, 4, 150.850, 0.15)
INSERT INTO InvoiceDetails (Order_ID, Pro_ID, Qty, Unitprice, Discount )
VALUES (102, 8, 5, 16.450, 0)
INSERT INTO InvoiceDetails (Order_ID, Pro_ID, Qty, Unitprice, Discount )
VALUES (103, 4, 6, 154.000, 0)
INSERT INTO InvoiceDetails (Order_ID, Pro_ID, Qty, Unitprice, Discount )
VALUES (103, 3, 7, 51.021, 0)
INSERT INTO InvoiceDetails (Order_ID, Pro_ID, Qty, Unitprice, Discount )
VALUES (104, 11, 8, 53.541, 0.25)
INSERT INTO InvoiceDetails (Order_ID, Pro_ID, Qty, Unitprice, Discount )
VALUES (104, 16, 9, 125.020, 0)
INSERT INTO InvoiceDetails (Order_ID, Pro_ID, Qty, Unitprice,Discount )
```

91 %

Messages

```
(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)
```

**Figure 21: Queries to INSERT InvoiceDetails**



91 %

Results | Messages

| | Order_ID | Pro_ID | Unitprice | Qty | Discount |
|---|---|---|---|---|---|
| 1 | 101 | 1 | 25.50 | 1 | 0.000 |
| 2 | 101 | 10 | 103.30 | 2 | 0.000 |
| 3 | 101 | 20 | 115.00 | 3 | 0.000 |
| 4 | 102 | 5 | 150.85 | 4 | 0.150 |
| 5 | 102 | 8 | 16.45 | 5 | 0.000 |
| 6 | 103 | 3 | 51.021 | 7 | 0.000 |
| 7 | 103 | 4 | 154.00 | 6 | 0.000 |
| 8 | 104 | 11 | 53.541 | 8 | 0.250 |
| 9 | 104 | 16 | 125.02 | 9 | 0.000 |
| 10 | 105 | 12 | 21.55 | 11 | 0.000 |
| 11 | 105 | 2 | 22.524 | 12 | 0.000 |
| 12 | 105 | 9 | 150.15 | 10 | 0.250 |
| 13 | 106 | 17 | 125.125 | 14 | 0.150 |
| 14 | 106 | 6 | 25.54 | 13 | 0.000 |
| 15 | 107 | 10 | 54.025 | 9 | 0.000 |
| 16 | 107 | 14 | 121.00 | 10 | 0.000 |
| 17 | 107 | 7 | 250.125 | 15 | 0.150 |
| 18 | 108 | 18 | 214.521 | 6 | 0.000 |
| 19 | 108 | 8 | 88.888 | 8 | 0.150 |
| 20 | 109 | 19 | 10.85 | 1 | 0.000 |
| 21 | 109 | 9 | 225.00 | 2 | 0.250 |
| 22 | 110 | 10 | 152.00 | 7 | 0.000 |

Query executed successfully.

**Figure 22: Show all InvoiceDetails table data**

2. Queries to UPDATE data with illustrations of final result

91 %

Results | Messages

| | Pro_ID | Pro_Name | Supplierid | Unitprice | Discontinued |
|---|---|---|---|---|---|
| 1 | 1 | Pencil | 1 | 5.50 | 0 |
| 2 | 10 | Flashcard | 4 | 167.70 | 0 |
| 3 | 11 | Dictionary | 5 | 505.80 | 0 |
| 4 | 12 | Desk | 6 | 199.50 | 0 |
| 5 | 13 | Cutter | 7 | 9.90 | 0 |
| 6 | 14 | Eraser | 7 | 5.00 | 1 |
| 7 | 15 | Duster | 8 | 3.00 | 0 |
| 8 | 16 | Felt-tip | 9 | 21.65 | 0 |
| 9 | 17 | Ruler | 9 | 9.80 | 0 |
| 10 | 18 | Stapler | 9 | 55.50 | 1 |
| 11 | 19 | Pen | 10 | 45.00 | 0 |
| 12 | 2 | Book | 1 | 145.00 | 1 |
| 13 | 20 | Watercolour | 10 | 145.50 | 0 |
| 14 | 3 | Bag | 1 | 250.00 | 0 |
| 15 | 4 | Borad | 2 | 50.00 | 0 |
| 16 | 5 | Clamp | 2 | 20.50 | 1 |
| 17 | 6 | Crayon | 3 | 18.30 | 0 |
| 18 | 7 | File holder | 3 | 6.00 | 0 |
| 19 | 8 | Compass | 4 | 10.50 | 0 |
| 20 | 9 | Globe | 4 | 350.45 | 1 |

Query executed successfully.

**Figure 23: Queries to Update**

```
-- Update
Update Product
set Unitprice = ' 6.500'
where Pro_ID = '1'
```

%

Messages

```
(1 row affected)

Completion time: 2022-05-05T15:20:29.6181672+07:00
```

**Figure 24: Result**

```
select * from Product
-- Update
Update Product
set Unitprice = ' 6.500'
where Pro_ID = '1'
```

91 %

Results | Messages

| | Pro_ID | Pro_Name | Supplierid | Unitprice | Discontinued |
|---|---|---|---|---|---|
| 1 | 1 | Pencil | 1 | 6.50 | 0 |
| 2 | 10 | Flashcard | 4 | 167.70 | 0 |
| 3 | 11 | Dictionary | 5 | 505.80 | 0 |
| 4 | 12 | Desk | 6 | 199.50 | 0 |
| 5 | 13 | Cutter | 7 | 9.90 | 0 |
| 6 | 14 | Eraser | 7 | 5.00 | 1 |
| 7 | 15 | Duster | 8 | 3.00 | 0 |
| 8 | 16 | Felt-tip | 9 | 21.65 | 0 |
| 9 | 17 | Ruler | 9 | 9.80 | 0 |
| 10 | 18 | Stapler | 9 | 55.50 | 1 |
| 11 | 19 | Pen | 10 | 45.00 | 0 |
| 12 | 2 | Book | 1 | 145.00 | 1 |

Query executed successfully.

**Figure 25: Display information when changes**

3. Queries to DELETE data with illustrations of final result

Results | Messages

| | Cust_ID | Companyname | Contactname | Address | Country | Phone |
|---|---|---|---|---|---|---|
| 1 | 1 | Allen, Michael | Sales Representative | Dong Da Str | Vietname | 0972208243 |
| 2 | 2 | Customer KBUDE | Peoples, John | Mataderos 7890 | China | 0972846198 |
| 3 | 3 | Customer HFBZG | Arndt, Torsten | 7890 Hanover Sq. | London | 1714567890 |
| 4 | 4 | Customer XHXJV | Poland, Carole | Forsterstr. 7890 | Germany | 0621554458 |
| 5 | 5 | Customer LVJSO | Smith, Denise | City Center Plaza 2345 Main St. | USA | 5035550126 |
| 6 | 6 | Customer RTXGC | Raghav, Amritansh | 6789, rue des Bouchers | France | 05215665686 |
| 7 | 7 | Customer XPNIK | Dressler, Marlies | Carrera 7890 con Ave. Bolívar #65-98 Llano Largo | Venezuela | 2658510445 |
| 8 | 8 | Customer GCJSG | Mallit, Ken | South House 1234 Queensbridge | UK | 1718901234 |
| 9 | 9 | Customer SNXOJ | Syamala, Manoj | Adenauerallee 7890 | Germany | 0711345678 |
| 10 | 10 | Customer CCFIZ | Conn, Steve | ul. Filtrowa 6789 | USA | 2065550123 |

Query executed successfully. | LAPTOP-QBKQJ9J6\SQLEXPRESS ... | LAPTOP-QBKQJ9J6\Admin ... | Sales_Managers | 00:00:00 | 10 rows

**Figure 26: Display information before**

```
select * from Customers
DELETE FROM Customers WHERE Companyname = 'Customer CCFIC';
```

91 %

Results | Messages

| | Cust_ID | Companyname | Contactname | Address | Country | Phone |
|---|---|---|---|---|---|---|
| 1 | 1 | Allen, Michael | Sales Representative | Dong Da Str | Vietname | 0972208243 |
| 2 | 2 | Customer KBUDE | Peoples, John | Mataderos 7890 | China | 0972846198 |
| 3 | 3 | Customer HFBZG | Arndt, Torsten | 7890 Hanover Sq. | London | 1714567890 |
| 4 | 4 | Customer XHXJV | Poland, Carole | Forsterstr. 7890 | Germany | 0621554458 |
| 5 | 5 | Customer LVJSO | Smith, Denise | City Center Plaza 2345 Main St. | USA | 5035550126 |
| 6 | 6 | Customer RTXGC | Raghav, Amritansh | 6789, rue des Bouchers | France | 05215665686 |
| 7 | 7 | Customer XPNIK | Dressler, Marlies | Carrera 7890 con Ave. Bolívar #65-98 Llano Largo | Venezuela | 2658510445 |
| 8 | 8 | Customer GCJSG | Mallit, Ken | South House 1234 Queensbridge | UK | 1718901234 |
| 9 | 9 | Customer SNXOJ | Syamala, Manoj | Adenauerallee 7890 | Germany | 0711345678 |

Query executed successfully. LAPTOP-QBKQJ9J6\SQLEXPRESS ...

**Figure 27: Display information when changes**

4. Queries to SELECT data with illustration of final result

```
select * from Invoice
```

91 %

Results | Messages

| | Order_ID | Cust_ID | Staff_ID | Orderdate | Requireddate |
|---|---|---|---|---|---|
| 1 | 101 | 1 | 5 | 2020-08-01 00:00:00.000 | 2020-07-16 00:00:00.000 |
| 2 | 102 | 1 | 3 | 2021-08-16 00:00:00.000 | 2021-07-10 00:00:00.000 |
| 3 | 103 | 2 | 1 | 2022-03-16 00:00:00.000 | 2022-02-18 00:00:00.000 |
| 4 | 104 | 3 | 2 | 2021-11-16 00:00:00.000 | 2021-10-18 00:00:00.000 |
| 5 | 105 | 4 | 4 | 2022-04-18 00:00:00.000 | 2022-03-18 00:00:00.000 |
| 6 | 106 | 5 | 1 | 2020-05-27 00:00:00.000 | 2020-04-23 00:00:00.000 |
| 7 | 107 | 5 | 5 | 2022-01-09 00:00:00.000 | 2021-12-18 00:00:00.000 |
| 8 | 108 | 6 | 3 | 2022-07-07 00:00:00.000 | 2022-06-15 00:00:00.000 |
| 9 | 109 | 6 | 2 | 2019-03-16 00:00:00.000 | 2019-02-07 00:00:00.000 |
| 10 | 110 | 6 | 5 | 2020-12-16 00:00:00.000 | 2021-01-18 00:00:00.000 |

**Figure 28: Select**

```
Select * from Customers
Select Cust_ID, Companyname, Phone from Customers
```

91 %

Results | Messages

| | Cust_ID | Companyname | Phone |
|---|---|---|---|
| 1 | 1 | Allen, Michael | 0972208243 |
| 2 | 2 | Customer KBUDE | 0972846198 |
| 3 | 3 | Customer HFBZG | 1714567890 |
| 4 | 4 | Customer XHXJV | 0621554458 |
| 5 | 5 | Customer LVJSO | 5035550126 |
| 6 | 6 | Customer RTXGC | 05215665686 |
| 7 | 7 | Customer XPNIK | 2658510445 |
| 8 | 8 | Customer GCJSG | 1718901234 |
| 9 | 9 | Customer SNXOJ | 0711345678 |
| 10 | 10 | Customer CCFIZ | 2065550123 |

Query executed successfully.

**Figure 29: Select ***

```
select Companyname
from Customers
where Companyname like 'Customer KBUDE'
```

91 %

Results | Messages

| | Companyname |
|---|---|
| 1 | Customer KBUDE |

**Figure 30: Select ... like**

```
select Lastname, Firstname , (year(getdate()) - year(Birthdate)) as Age
from Staff
where (year(getdate()) - year(Birthdate)) < 21
```

91 %

Results | Messages

| | Lastname | Firstname | Age |
|---|---|---|---|
| 1 | Linh | Bùi | 20 |
| 2 | Hòa | Nguyễn | 20 |
| 3 | My | Nguyễn | 20 |

**Figure 31: Select ... where**

```
select top (10) Pro_Name , SUM(Unitprice) as [Total sold]
from Product
group by Pro_Name
order by SUM(Unitprice) DESC
```

91 %

Results | Messages

| | Pro_Name | Total sold |
|---|---|---|
| 1 | Dictionary | 505.80 |
| 2 | Globe | 350.45 |
| 3 | Bag | 250.00 |
| 4 | Desk | 199.50 |
| 5 | Flashcard | 167.70 |
| 6 | Watercolour | 145.50 |
| 7 | Book | 145.00 |
| 8 | Stapler | 55.50 |
| 9 | Borad | 50.00 |
| 10 | Pen | 45.00 |

**Figure 32: Select ... top**

```
select Order_ID, SUM(Unitprice*Qty*(1-Discount)) as [value of order]
from InvoiceDetails
group by Order_ID
```

91 %

Results | Messages

| | Order_ID | value of order |
|---|---|---|
| 1 | 101 | 577.1000000 |
| 2 | 102 | 595.1400000 |
| 3 | 103 | 1281.1470000 |
| 4 | 104 | 1446.4260000 |
| 5 | 105 | 1633.4630000 |
| 6 | 106 | 1821.0075000 |
| 7 | 107 | 4885.3187500 |
| 8 | 108 | 1891.5644000 |
| 9 | 109 | 348.3500000 |
| 10 | 110 | 1227.1250000 |

**Figure 33: Select ... group by**

```
select month(orderdate), SUM(D.unitprice * D.qty * (1-D.discount)) as Revenu
from Invoice as I, InvoiceDetails as D
where D.Order_ID = I.Order_ID and YEAR(orderdate) = 2022
group by month(orderdate)

select * from Invoicedetails
```

91 %

Results | Messages

| | (No column name) | Revenu |
|---|---|---|
| 1 | 1 | 4885.3187500 |
| 2 | 3 | 1281.1470000 |
| 3 | 4 | 1633.4630000 |
| 4 | 5 | 180.6000000 |
| 5 | 7 | 1891.5644000 |
| 6 | 8 | 14.0000000 |
| 7 | 10 | 200.5060000 |
| 8 | 12 | 277.0000000 |

**Figure 34: Select ... group by**

```
SELECT C.companyname, SUM (D.unitprice*D.qty*(1-D.discount)) as [Total money spent in 2022]
FROM Customers as C, Invoice as I, InvoiceDetails as D
WHERE C.Cust_ID = I.Cust_ID and I.Order_ID = D.Order_ID
and C.companyname = 'Customer KBUDE'
and YEAR(I.orderdate) = 2022
GROUP BY C.companyname
```

91 %

Results | Messages

| | companyname | Total money spent in 2022 |
|---|---|---|
| 1 | Customer KBUDE | 1281.1470000 |

**Figure 35: Select ... group by**

```
select City, count(*) as [Total]
from Staff
group by City
having (City= 'Ha Noi' or City= 'Thanh Hoa')
```

91 %

Results | Messages

| | City | Total |
|---|---|---|
| 1 | Ha Noi | 3 |
| 2 | Thanh Hoa | 1 |

Figure 36: Select... having

```
CREATE TRIGGER trg_Products
ON Product
FOR DELETE
AS
 BEGIN
  IF EXISTS(SELECT *FROM Product WHERE Supplierid =1)
  BEGIN
   PRINT N'Không được xóa sản phẩm vẫn còn hàng';
   ROLLBACK TRANSACTION;
  END
 END
GO
```

```
91 %
Messages
Commands completed successfully.

Completion time: 2022-05-05T16:55:18.5797822+07:00
```

Figure 37: Trigger

# Chapter 3 – Test the system

1. Test cases

| Test case type | Description | Test step | Expected | Status (P/F) |
|---|---|---|---|---|
| Insert into | There is no need to specify column names because all column values are in table column order. | Using the following queries, I filled in the information in the order of the columns and inserted the data. | The new data values are stored in the database. | P |
| Update | It includes a SET clause, which defines the column and value as a pair of items. To limit the number of rows, it is necessary to use the where clause to locate the row to be modified in the table. | Change the data of the Unitprice column in Product for all rows of the table with the value of 5.500 in Pro_ID = '1' to the current 6.500 | Saw that the Unitprice column at Pro_ID = '1' was changed with the new values. | P |
| Delete | DELETE queries include a condition in the WHERE clause because if it is not used, all records in the table are deleted. | Enter the DELETE statement to delete Companyname = 'Customer CCFIC' in the Customer table. | Companyname = 'Customer CCFIC' has been deleted. | P |
| Select * from | The query returned the full set of rows, columns, or a subset of rows and columns from one or more system Manager tables. | FROM is used to list the necessary tables used in the query. | Executing the SELECT* FROM Customer (Cust_ID, Companyname, Phone) statement outputs all columns of data. | P |

| | | | | |
|---|---|---|---|---|
| Select top | The TOP(10) return clause inserted the first 10 rows from the result set based on Pro_Name. | TOP(10) has inserted the top 10 and is already sorted. | Will retrieve the first Pro_Name record on the table . | P |
| SQL having | Limit the sum results to limit the groups of returned rows to the group of returned rows, only if the TRUE condition for City is met to filter the results so that only addresses in Ha Noi and Thanh Hoa are returned about. | When using the HAVING clause, it must be combined with the Group by clause. Print to screen error. | Returns the result (2 rows affected) | P |
| SQL where | Filter record from tables to satisfy all system and user requiements | The operators between, >, like, can be used. | Data has been returned | P |
| SQL trigger | Delete the information of the Product table, then records the information of these changes. | Delete Products information into the table, and when the delete is complete, a successful message will be printed on the screen. All Trigger creation BEGIN and ending with END | Message: Commands completed successfully (1 rows affected ) | P |

2. Flowchart to show how the system works