**VIETNAM NATIONAL UNIVERSITY**

**UNIVERSITY OF ECONOMICS AND LAW**

**FACULTY OF INFORMATION SYSTEM**



**SUBJECT: DATABASE**

**REPORT**

**GROCERY STORE MANAGEMENT**

*Instructor guide: Master Nguyen Duy Nhat*

*Huynh Thanh Quang*

|  |  |
| --- | --- |
| Members: | Student code |
| Trần Khánh Duy | K184060780 |
| Hồ Thủy Tiên | K184060808 |
| Bùi Nhật Khanh | K184060791 |
| Nguyễn Hà Phương Trúc | K184060765 |

*Ho Chi Minh City, June 1st, 2020*

# **TABLE OF CONTENT**

[**TABLE OF CONTENT** 2](#_Toc42725373)

[**CHAPTER 1: PROJECT’S OVERVIEW** 4](#_Toc42725374)

[**1.1 Overview of the problem and introduction** 4](#_Toc42725375)

[**1.2 The reason for choosing the project** 4](#_Toc42725376)

[**1.3 The benefits of using Grocery Store Management Application** 4](#_Toc42725377)

[**1.4 Task description** 5](#_Toc42725378)

[**CHAPTER 2: USER REQUIREMENT ANALYSIS** 6](#_Toc42725379)

[**2.1 Entity** 6](#_Toc42725380)

[**2.2 User requirements** 6](#_Toc42725381)

[***2.2.1 For saving purpose*** 6](#_Toc42725382)

[***2.2.2 Information queries*** 6](#_Toc42725383)

[**CHAPTER 3: DATABASE DESIGNING** 8](#_Toc42725384)

[**3.1 Conceptual model** 8](#_Toc42725385)

[**3.2 ERD design** 9](#_Toc42725386)

[**3.3 Logical diagram** 11](#_Toc42725387)

[**3.4 Physical design** 11](#_Toc42725388)

[**3.5 Integrity constraint** 13](#_Toc42725389)

[**CHAPTER 4: SQL SOLVING** 14](#_Toc42725390)

[**4.1 Customer management** 14](#_Toc42725391)

[**4.2 Invoice management** 18](#_Toc42725392)

[**4.3 Product management** 22](#_Toc42725393)

[**4.4 Category management** 27](#_Toc42725394)

[**4.5 Supplier management** 30](#_Toc42725395)

[**4.6 Selling management** 34](#_Toc42725396)

[**4.7 Sign-in** 37](#_Toc42725397)

[**CHAPTER 5: APPLICATION** 40](#_Toc42725398)

[**5.1 Platform** 40](#_Toc42725399)

[**5.2 Business Processing** 40](#_Toc42725400)

[***5.2.1 Sign in*** 40](#_Toc42725401)

[***5.2.2 Create invoices*** 41](#_Toc42725402)

[***5.2.3 Manage the grocery*** 42](#_Toc42725403)

[***5.2.4 Manage user accounts*** 48](#_Toc42725404)

[***5.2.5 Log out*** 50](#_Toc42725405)

[**CHAPTER 6: EVALUATION THE PROJECT** 51](#_Toc42725406)

[**6.1 Advantage** 51](#_Toc42725407)

[**6.2 Disadvantage** 51](#_Toc42725408)

# **CHAPTER 1: PROJECT’S OVERVIEW**

## **1.1 Overview of the problem and introduction**

* Nowadays, most of the grocery stores in Vietnam still use a paper management system. This leads to a lot of inconvenience and difficulty for the owners when it comes to storage and evaluation of their business. While technology is growing, the traditional paper method might increase inadequate and making difficult for the owner management.
* The disadvantage of the traditional paper method is if the owners want to evaluate the revenue each month, it will cost a lot of time to measure.

## **1.2 The reason for choosing the project**

* First of all, selecting this topic comes from reality, we can create a grocery store management program that helps a lot of grocery owners in selling and controlling their business.
* Previously, owners could only measure their revenue manually each day. And by collecting all the paper each day they will have the result for each month and so on.
* Therefore, the grocery store management solution was born to serve the essential needs of the owners in managing their products and selling status.
* The application of IT has brought a breakthrough for grocery management, helping the owners to collect information about their invoices, products, and customers accurately and promptly. From there, the owners can make the right decisions for their business.
* This tracking system aims to improve the management quality of their business and avoid the mistakes of the traditional method.

## **1.3 The benefits of using Grocery Store Management Application**

* Supervising their grocery activities comprehensively, in real-time. Data is store in digital form, which easy to evaluate. The report figures are accurate.
* Provide information, revenue reports, and order payments.
* Saving time, effort, and optimize sale revenues.

## **1.4 Task description**

* ***Management information***
* Customers information.
* Managing revenue per month.
* ***Products management***
* Products information
* Suppliers information
* ***Selling management process***
* Enter the sale order of each customer
* Evaluate and classify at the end of each month
* ***Storage process***
* At the beginning of each month or when out of stock, increase the number of products.
* ***Management process***
* The mission is responsible for managing the information related to the date, customers, invoices, products, and supplier's information at the beginning of each month or changes each week.

# **CHAPTER 2: USER REQUIREMENT ANALYSIS**

## **2.1 Entity**

* From the research and survey process, setting out the professional requirements in the process of managing a grocery store, our group has identified some important information that needs to manage and develop into data entities are:
* Customer
* Invoice
* Product
* Supplier

## **2.2 User requirements**

### ***2.2.1 For saving purpose***

* Customer: Customer ID, Customer Name, Birthday, Address, Phone Number.
* Invoice: Invoice ID, Customer ID, Date, Price.
* Product: Product ID, Product Name, Product Category, Price, Number, Supplier ID.
* Supplier: Supplier ID, Supplier Name, Address, Phone Number.

### ***2.2.2 Information queries***

* ***Customer management***
* Function: manage all the information of the customer. Such as add, delete, and update customer information. Find customer information by some specific attribute.
* ***Invoice management***
* Function: manage all the information on each invoice. Such as add invoices, make sale order, and evaluate revenue.
* ***Product management***
* Function: manage all the information on each product. Such as add, delete, and update product information. Find product information by some specific attribute. Sort the product by category.
* ***Category management***
* Function: manage all the information on each category. Such as add, delete, and update category information.
* ***Supplier management***
* Function: manage all the information of the supplier. Such as add, delete, and update supplier information. Find supplier information by some specific attribute
* ***Selling management***
* Function: allow owners to pick up the product into sale order and make the payment. Tracking the change in the inventory after each sale order.
* ***Sign-in***
* Function: allow owners to sign-in the systems. Changing the fixed password.

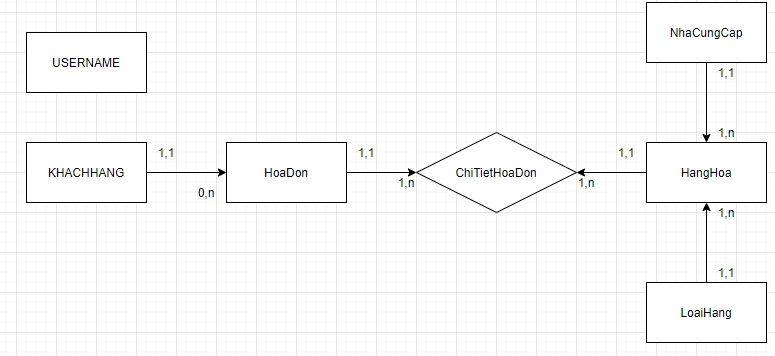
# **CHAPTER 3: DATABASE DESIGNING**

## **3.1 Conceptual model**



* Customer: saving information about customer identification, customer name, home address, date of birth, and phone number.
* Invoice: saving information about customer identification, invoice identification, date invoice created, the price of the invoice.
* Product: saving information about product identification, product name, product category, the price of the product, amount of the product, supplier identification.
* Supplier: saving information about supplier identification, supplier name, supplier address, and phone number.

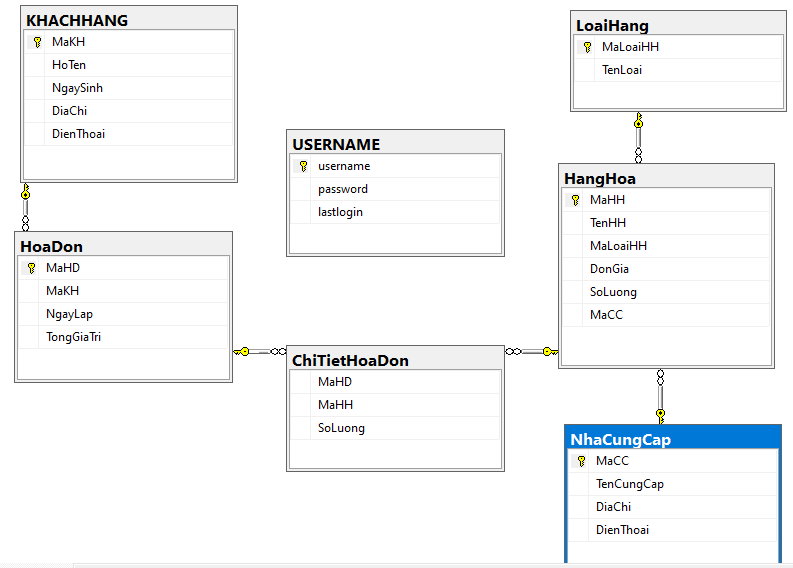
## **3.2 ERD design**



Based on information from management requirements and the conceptual model, our group has defined relationships between entities as follows:

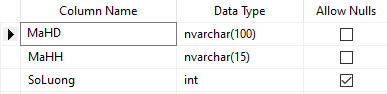
* ***KHACHHANG – HoaDon (1 - n)***
* The relationship between KHACHHANG and HoaDon is one to many, which means that each KHACHHANG has multiple HoaDon and each HoaDon only belongs to one KHACHHANG.
* ***HoaDon – ChiTietHoaDon (1 - n)***
* The relationship between HoaDon and ChiTietHoaDon is one to many, which means that each HoaDon has multiple ChiTietHoaDon and each ChiTietHoaDon only belongs to one HoaDon.
* ***HangHoa – ChiTietHoaDon (1 – n)***
* The relationship between HangHoa and ChiTietHoaDon is one to many, which means that each HangHoa has multiple ChiTietHoaDon and each ChiTietHoaDon only belongs to one HangHoa.
* ***LoaiHang – HangHoa (1 – n)***
* The relationship between LoaiHang and HangHoa is one to many, which means that each LoaiHang has multiple HangHoa and each HangHoa only belongs to one LoaiHang.
* ***NhaCungCap – HangHoa (1 – n)***
* The relationship between NhaCungCap and HangHoa is one to many, which means that each NhaCungCap has multiple HangHoa and each HangHoa only belongs to one NhaCungCap.

## **3.3 Logical diagram**



## **3.4 Physical design**

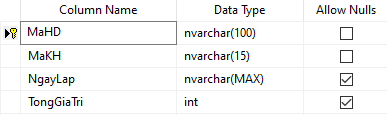
* ***Table ChiTietHoaDon***



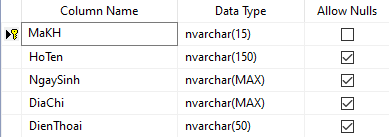
* ***Table HangHoa***



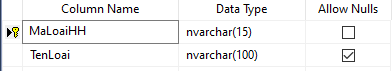
* ***Table HoaDon***



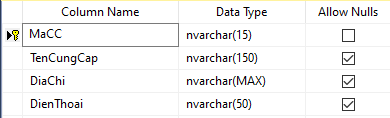
* ***Table KHACHHANG***



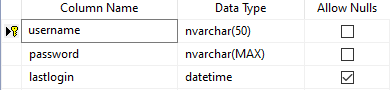
* ***Table LoaiHang***



* ***Table NhaCungCap***



* ***Table USERNAME***



## **3.5 Integrity constraint**

*Integrity Constraint (Rule) and check the error of integrity is a very important issue in the process of analyzing and exploiting and designing a database material.*

R0: On the relationship between ChiTietHoaDon and HangHoa, the SoLuong attribute in the ChiTietHoaDon table must lower than the SoLuong attribute in the HangHoa table.

Form specification:

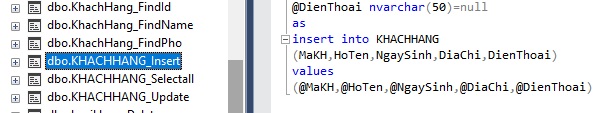
R0:

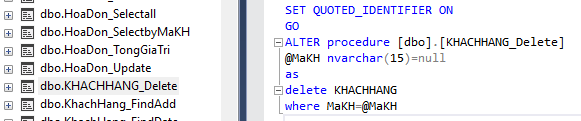
Influence table:

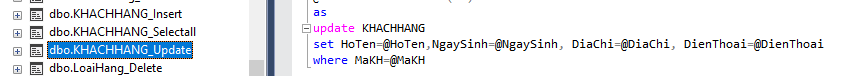
|  |  |  |  |
| --- | --- | --- | --- |
| R0 | Add | Delete | Fix |
| HangHoa | - | + | + (SoLuong) |
| ChiTietHoaDon | + | - |

# **CHAPTER 4: SQL SOLVING**

## **4.1 Customer management**

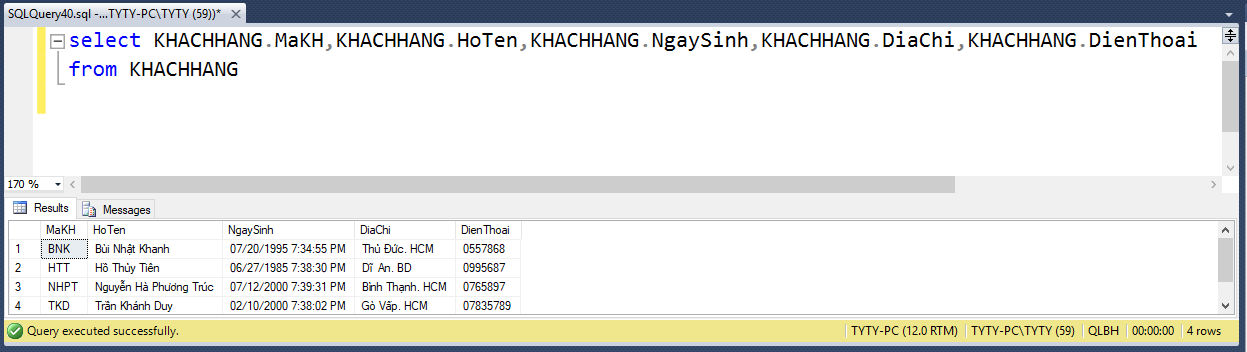
* ***Insert***
* The following SQL statement will insert a new record, insert data in the “MaKH”, “HoTen”, “NgaySinh”, “DiaChi”, “DienThoai” columns in “KHACHHANG” table.
* ***Delete***
* The following SQL statement will delete information of “MaKH” in “KHACHHANG” table.



* ***Update***
* The following SQL statement will update the information of “HoTen”, ”NgaySinh”, ”DiaChi”, “DienThoai” columns in “KHACHHANG” table.
* ***Display information***
* Display a table containing information of the Customers.

SELECT KHACHHANG.MaKH,KHACHHANG.HoTen,KHACHHANG.NgaySinh,  
KHACHHANG.DiaChi,KHACHHANG.DienThoai

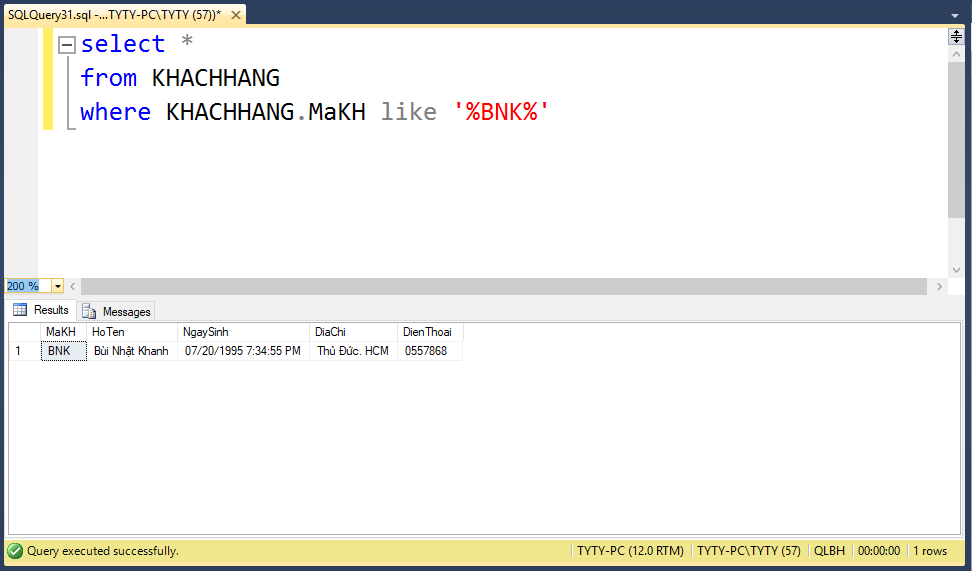
FROM KHACHHANG



* ***Search information***
* Find information to look up in the Customer table (based on Customer Code/ Full Name/ Date of Birth/ Address/ Phone Number ).
* Based on the Customer Code.

SELECT \* FROM KHACHHANG

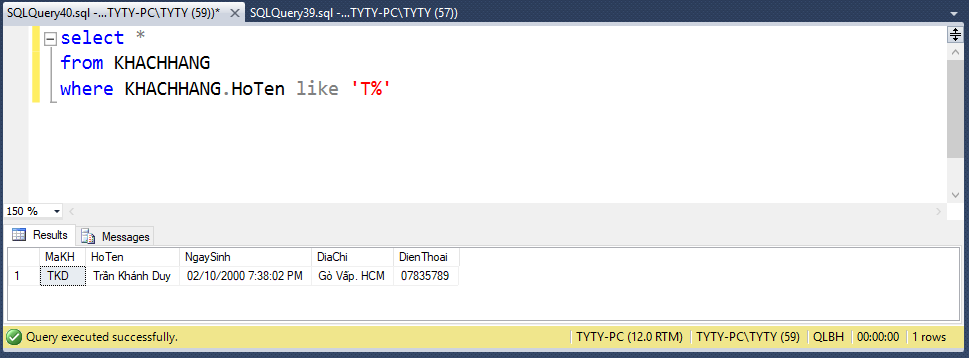
WHERE KHACHHANG.MaKH like '%'+@MaKH+'%'



* Based on Full Name.

SELECT \* FROM KHACHHANG

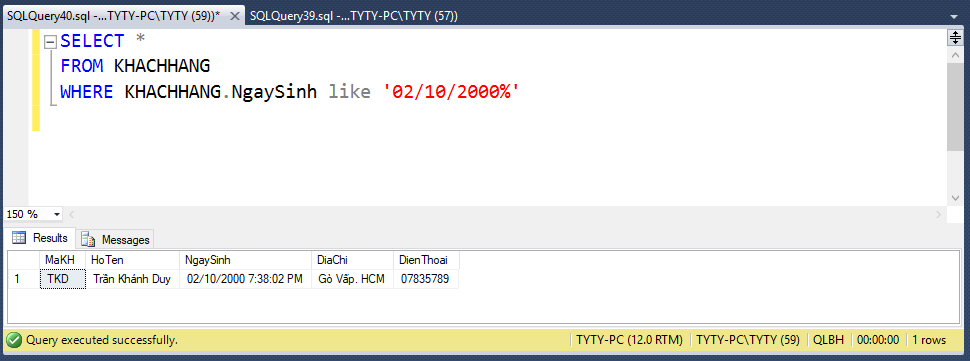
WHERE KHACHHANG.HoTen like '%'+@HoTen+'%'



* Based on Date of Birth.

SELECT \* FROM KHACHHANG

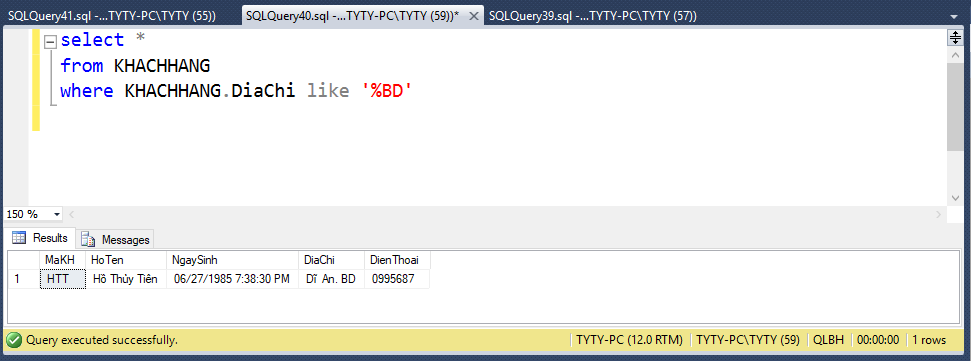
WHERE KHACHHANG.NgaySinh like '%'+@NgaySinh+'%'



* Based on Address.

SELECT \* FROM KHACHHANG

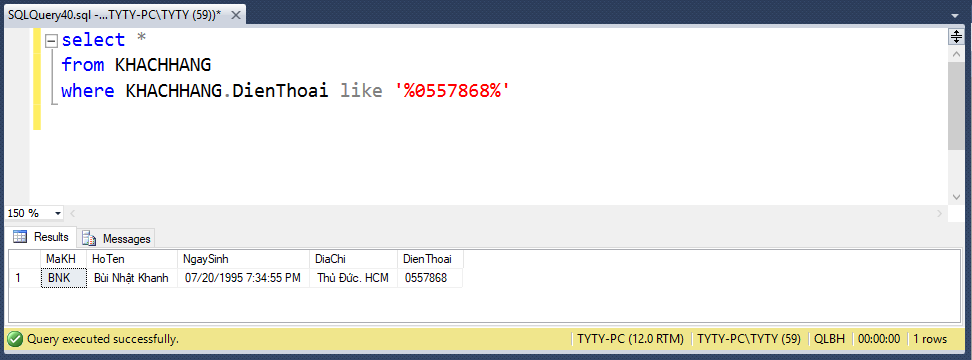
WHERE KHACHHANG.DiaChi like '%'+@DiaChi+'%'



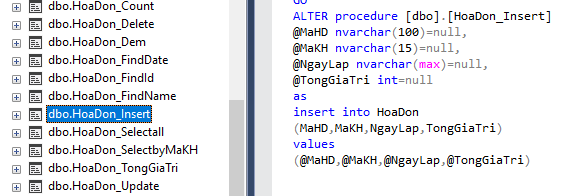
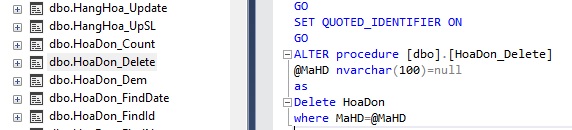
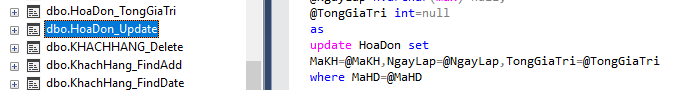
* Based on Phone Number.

SELECT \* FROM KHACHHANG

WHERE KHACHHANG.DienThoai like '%'+@DienThoai+'%'

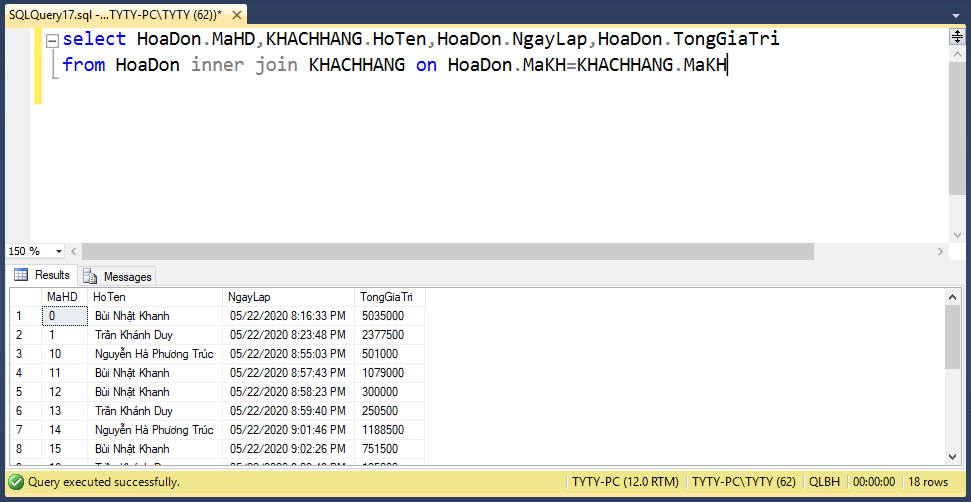


## **4.2 Invoice management**

* ***Insert***
* The following SQL statement will insert a new record, insert data in the “MaHD”, “MaKH”, “NgayLap” and “TongGiaTri” columns in “HoaDon” table.
* ***Delete***
* The following SQL statement will delete information of “MaHD” in “HoaDon” table.
* ***Update***
* The following SQL statement will update the information of “MaKH”, “NgayLap”, “TongGiaTri” columns in “HoaDon” table.
* ***Display information***
* Display a table describing the sales revenue of the store.

SELECT HoaDon.MaHD,KHACHHANG.HoTen,HoaDon.NgayLap,HoaDon.TongGiaTri

FROM HoaDon inner join KHACHHANG on HoaDon.MaKH=KHACHHANG.MaKH

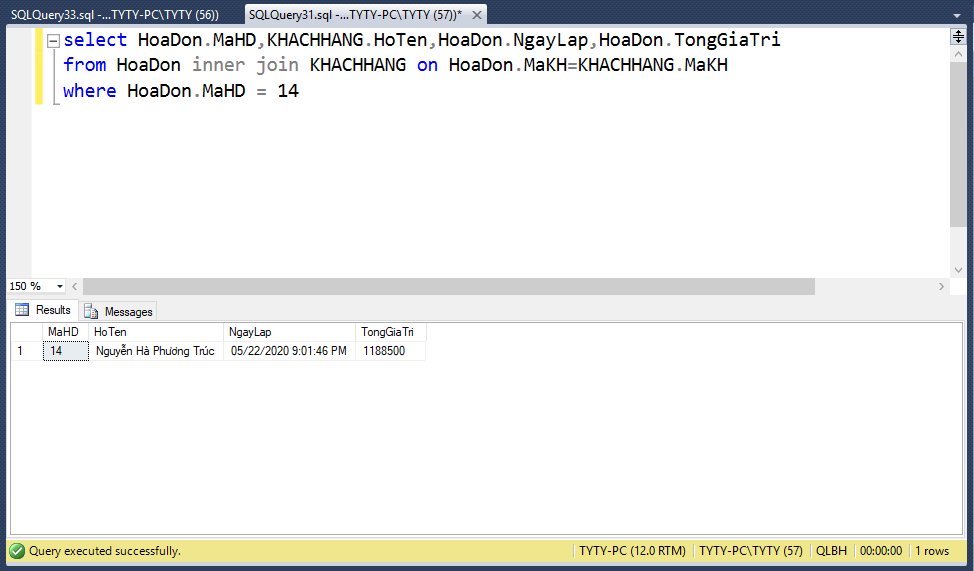


* ***Search information***
* Find information to look up in the Sales Revenue table ( based on Invoice or Customer or Invoice Date ).
* Based on the Invoice.

SELECT HoaDon.MaHD,KHACHHANG.HoTen,HoaDon.NgayLap,HoaDon.TongGiaTri

FROM HoaDon inner join KHACHHANG on HoaDon.MaKH=KHACHHANG.MaKH

WHERE HoaDon.MaHD = @MaHD



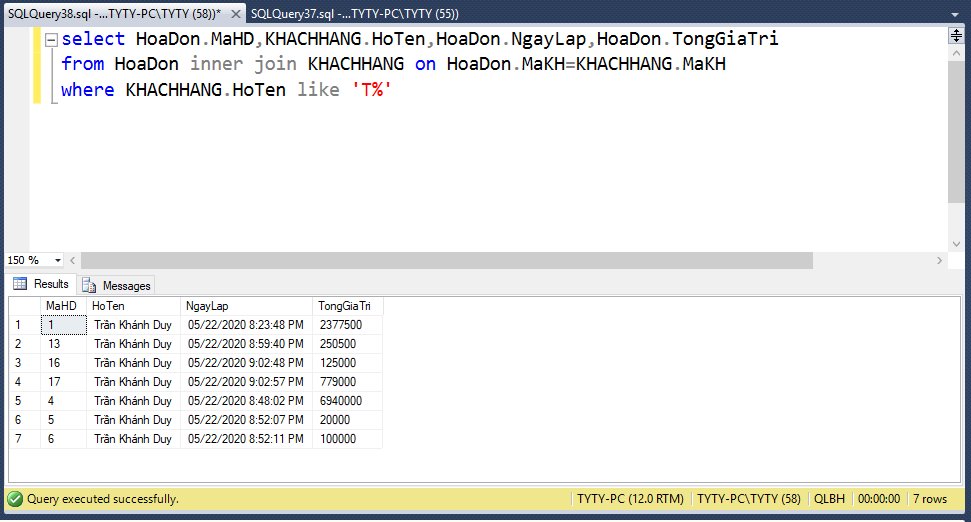
* Based on Customer.

SELECT HoaDon.MaHD,KHACHHANG.HoTen,HoaDon.NgayLap,

HoaDon.TongGiaTri

FROM HoaDon inner join KHACHHANG on HoaDon.MaKH=KHACHHANG.MaKH

WHERE KHACHHANG.HoTen like '%'+@HoTen+'%'



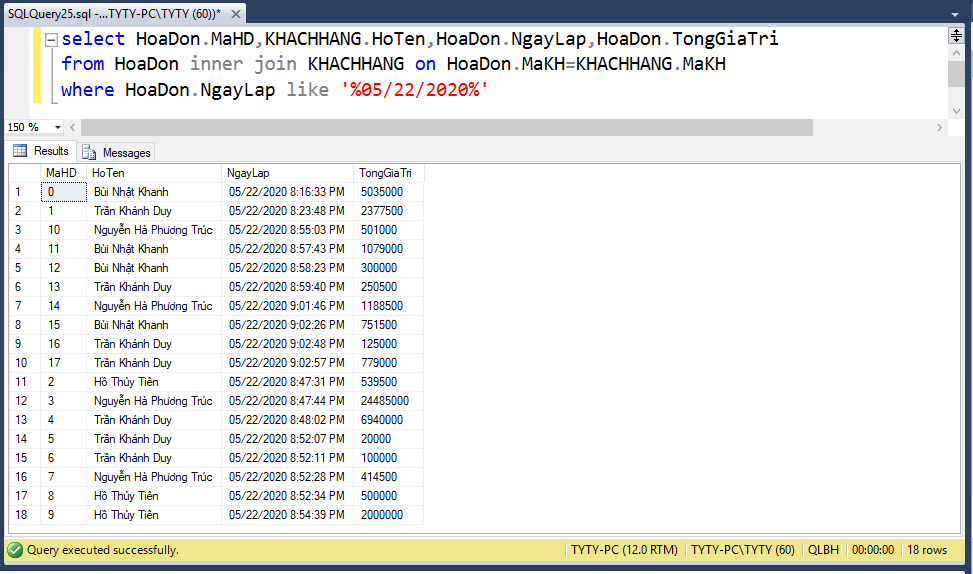
* Based on Invoice Date.

SELECT HoaDon.MaHD,KHACHHANG.HoTen,HoaDon.NgayLap,

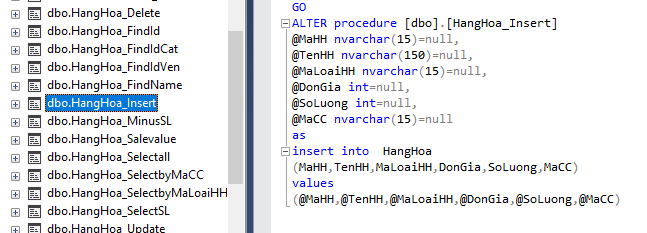
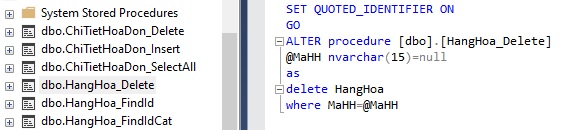
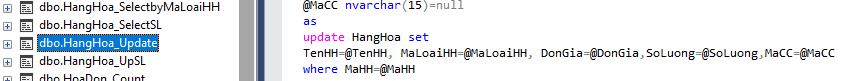
HoaDon.TongGiaTri

FROM HoaDon inner join KHACHHANG on HoaDon.MaKH=KHACHHANG.MaKH

WHERE HoaDon.NgayLap like '%'+@NgayLap+'%'



## **4.3 Product management**

* ***Insert***
* The following SQL statement will insert a new record, insert data in the “TenHH”, “MaHH”, “SoLuong”, “MaLoaiHH”, “MaCC” and “DonGia” columns in “HangHoa” table.
* ***Delete***
* The following SQL statement will delete information of “MaHH” in “HangHoa” table.
* ***Update***
* The following SQL statement will update a new record.
* ***Display information***
* Display a table containing information about the Goods.

SELECT q1.MaHH, q1.TenHH, LoaiHang.TenLoai, q1.DonGia, q1.SoLuong, q1.TenCungCap

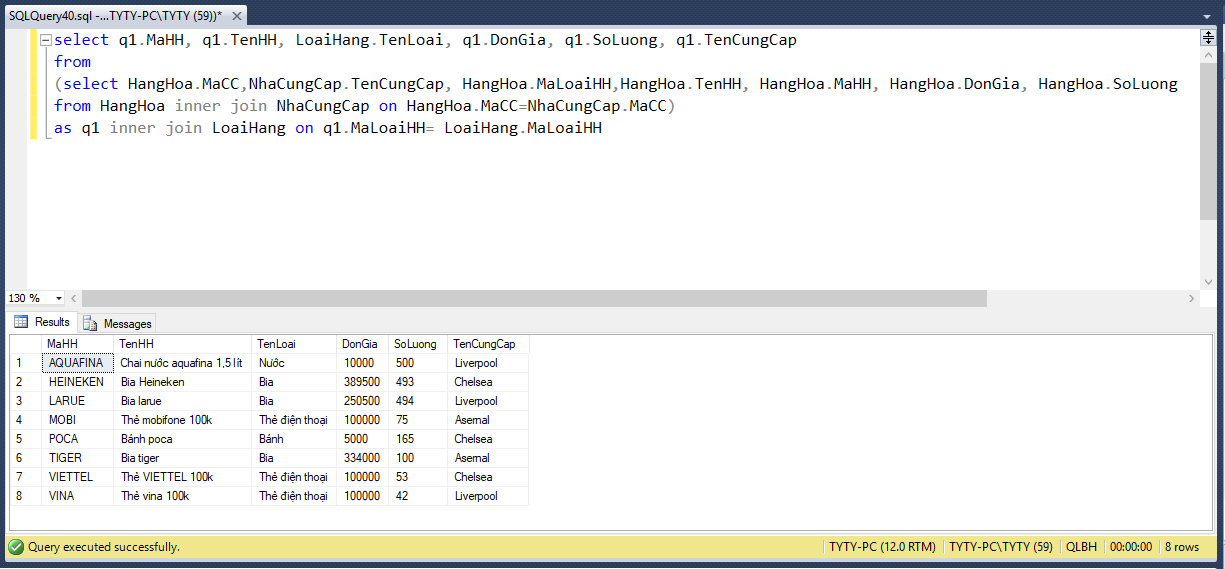
FROM

(SELECT HangHoa.MaCC,NhaCungCap.TenCungCap,HangHoa.MaLoaiHH,

HangHoa.TenHH, HangHoa.MaHH, HangHoa.DonGia, HangHoa.SoLuong

FROM HangHoa inner join NhaCungCap on HangHoa.MaCC=NhaCungCap.MaCC)

as q1 inner join LoaiHang on q1.MaLoaiHH= LoaiHang.MaLoaiHH



* ***Search information***
* Find information to look up in the Goods table ( based on Goods Code/ Goods Name/ Type of Goods/ Vendor Name).
* Based on Goods Code.

SELECT q1.MaHH, q1.TenHH, LoaiHang.TenLoai, q1.DonGia, q1.SoLuong, q1.TenCungCap

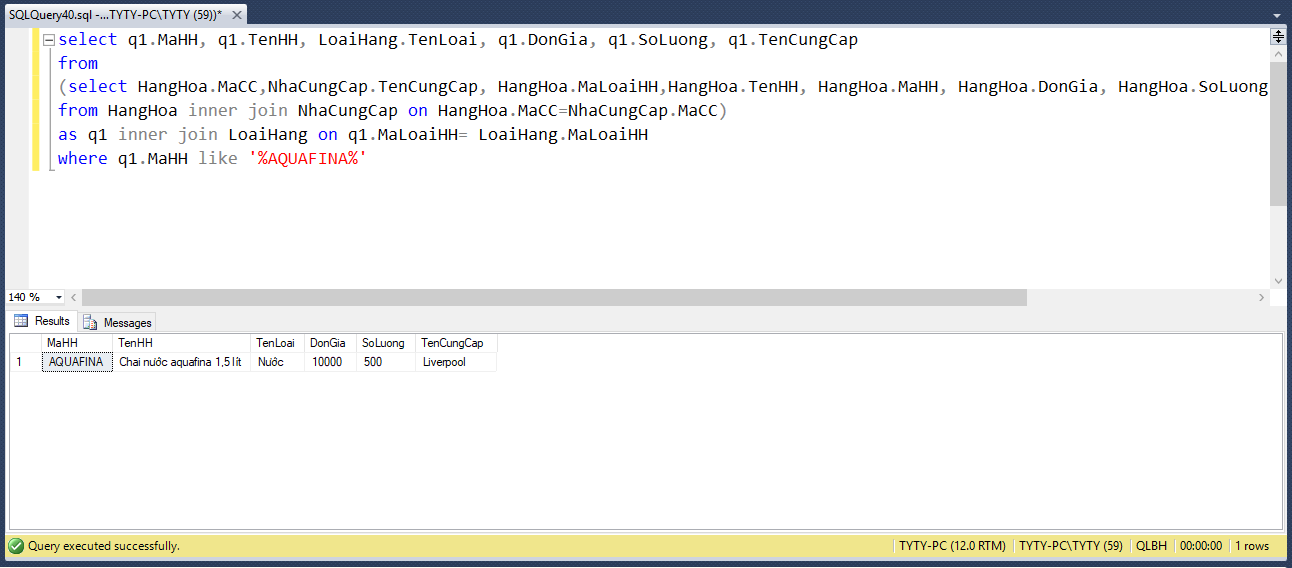
FROM

(SELECT HangHoa.MaCC,NhaCungCap.TenCungCap, HangHoa.MaLoaiHH,HangHoa.TenHH, HangHoa.MaHH, HangHoa.DonGia, HangHoa.SoLuong

FROM HangHoa inner join NhaCungCap on HangHoa.MaCC=NhaCungCap.MaCC)

as q1 inner join LoaiHang on q1.MaLoaiHH= LoaiHang.MaLoaiHH

WHERE q1.MaHH like '%'+@MaHH+'%'



* Based on Goods Name.

SELECT q1.MaHH, q1.TenHH, LoaiHang.TenLoai, q1.DonGia, q1.SoLuong, q1.TenCungCap

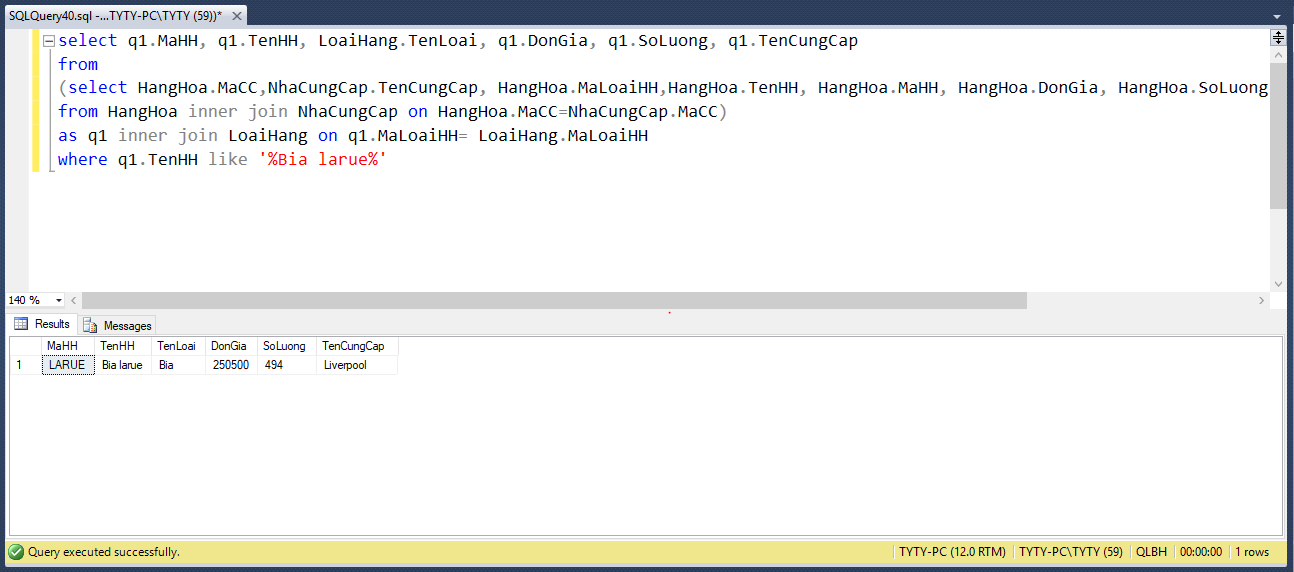
FROM

(SELECT HangHoa.MaCC,NhaCungCap.TenCungCap, HangHoa.MaLoaiHH,HangHoa.TenHH, HangHoa.MaHH, HangHoa.DonGia, HangHoa.SoLuong

FROM HangHoa inner join NhaCungCap on HangHoa.MaCC=NhaCungCap.MaCC)

as q1 inner join LoaiHang on q1.MaLoaiHH= LoaiHang.MaLoaiHH

WHERE q1.TenHH like '%'+@TenHH+'%'



* Based on Type of Goods.

SELECT q1.MaHH, q1.TenHH, LoaiHang.TenLoai, q1.DonGia, q1.SoLuong, q1.TenCungCap

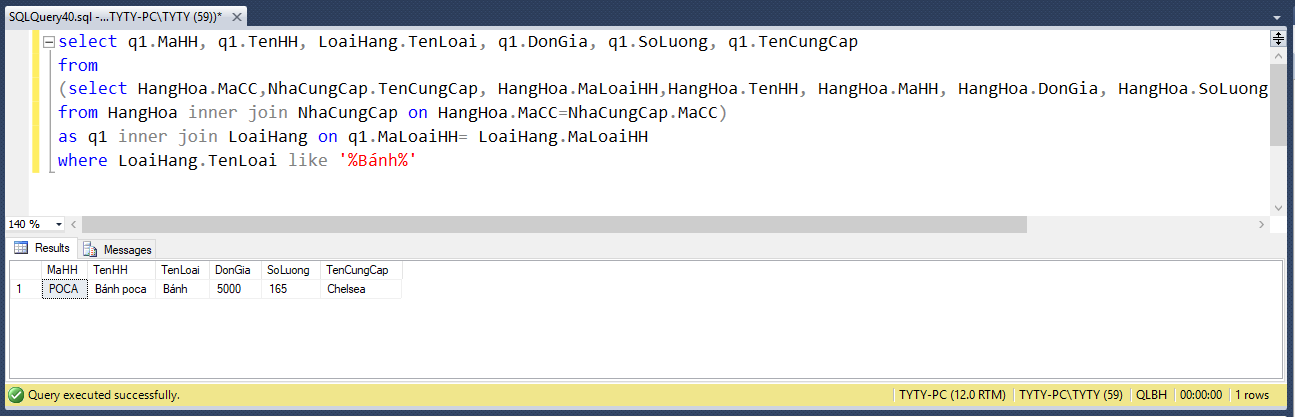
FROM

(SELECT HangHoa.MaCC,NhaCungCap.TenCungCap, HangHoa.MaLoaiHH,HangHoa.TenHH, HangHoa.MaHH, HangHoa.DonGia, HangHoa.SoLuong

FROM HangHoa inner join NhaCungCap on HangHoa.MaCC=NhaCungCap.MaCC)

as q1 inner join LoaiHang on q1.MaLoaiHH= LoaiHang.MaLoaiHH

WHERE LoaiHang.TenLoai like '%'+@TenLoai+'%'



* Based on Vendor Name.

SELECT q1.MaHH, q1.TenHH, LoaiHang.TenLoai, q1.DonGia, q1.SoLuong, q1.TenCungCap

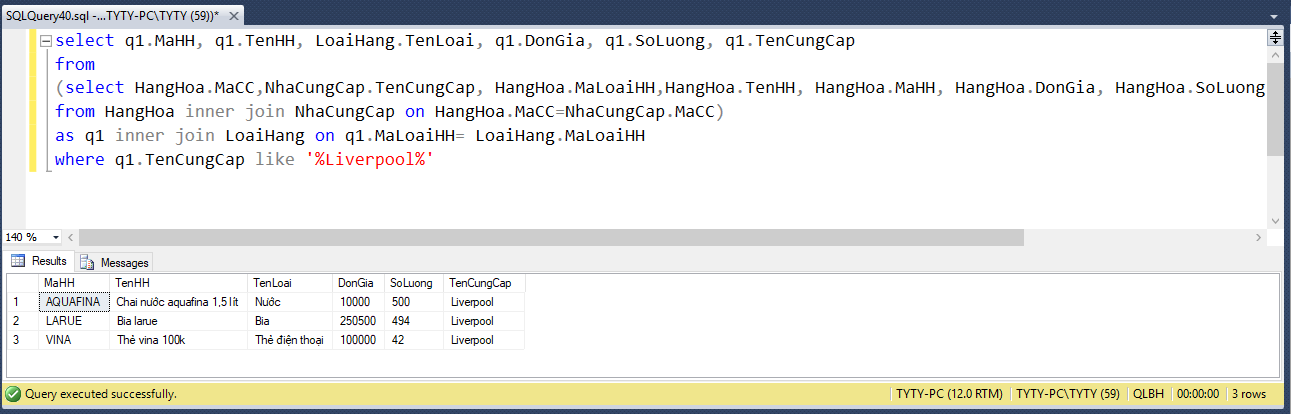
FROM

(SELECT HangHoa.MaCC,NhaCungCap.TenCungCap, HangHoa.MaLoaiHH,HangHoa.TenHH, HangHoa.MaHH, HangHoa.DonGia, HangHoa.SoLuong

FROM HangHoa inner join NhaCungCap on HangHoa.MaCC=NhaCungCap.MaCC)

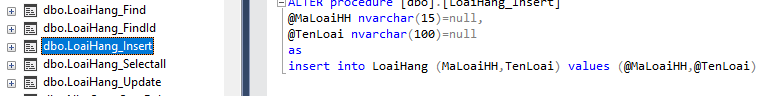
as q1 inner join LoaiHang on q1.MaLoaiHH= LoaiHang.MaLoaiHH

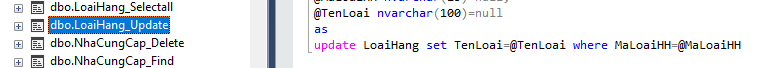
WHERE q1.TenCungCap like '%'+@TenCungCap+'%'



## **4.4 Category management**

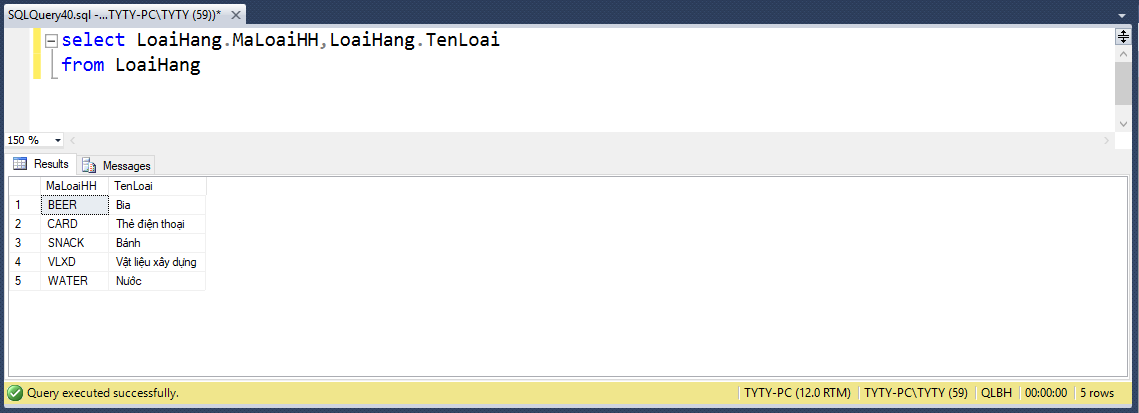
* ***Insert***
* The following SQL statement will insert a new record, insert data in the “MaLoaiHH”, “TenLoai” columns in “LoaiHang” table.



* ***Delete***
* The following SQL statement will delete information of “MaLoaiHH” in “LoaiHang” table.
* ***Update***
* The following SQL statement will update the information of “TenLoai” columns in “LoaiHang” table.
* ***Display information***
* Display a table containing information on the Category.

SELECT LoaiHang.MaLoaiHH,LoaiHang.TenLoai

FROM LoaiHang

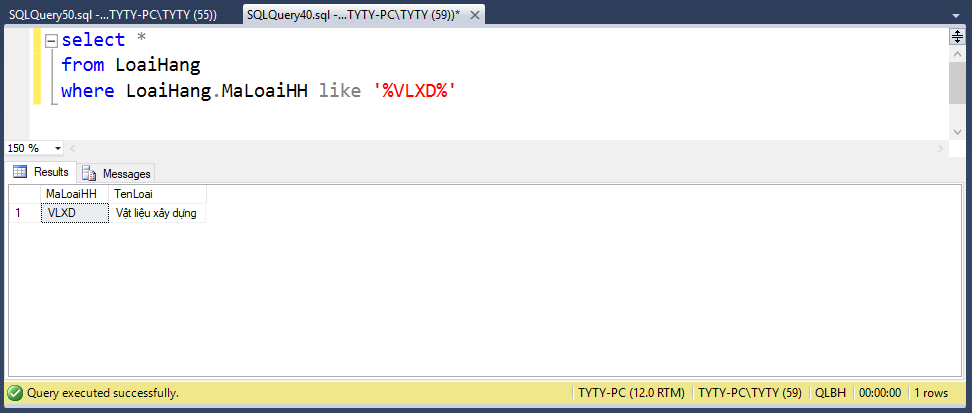


* ***Search information***
* Find information to look up in the Category table ( Based on Category Code/ Category Name ).
* Based on Category Code.

SELECT \*

FROM LoaiHang

WHERE LoaiHang.MaLoaiHH like '%'+@MaLoaiHH+'%'

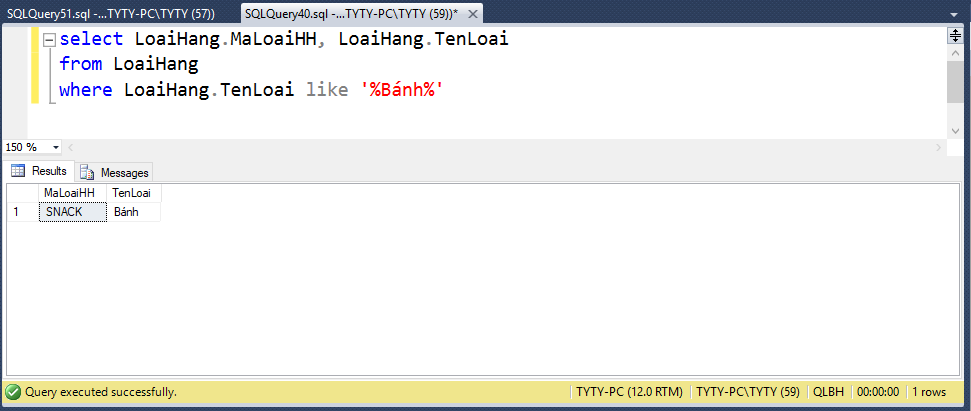


* Based on Category Name.

SELECT LoaiHang.MaLoaiHH, LoaiHang.TenLoai

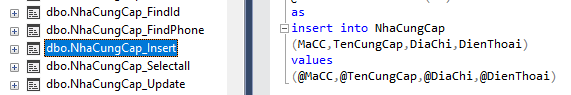
FROM LoaiHang

WHERE LoaiHang.TenLoai like '%'+@TenLoai+'%'

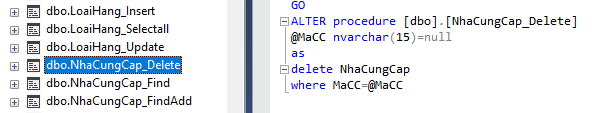


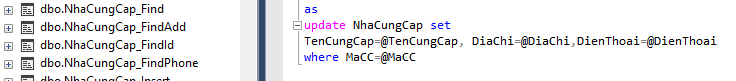
## **4.5 Supplier management**

* ***Insert***
* The following SQL statement will insert a new record, insert data in the “MaCC”, “TenCungCap”, “DiaChi” and “DienThoai” columns in “NhaCungCap” table.



* ***Delete***
* The following SQL statement will delete information of “MaCC” in “NhaCungCap” table.



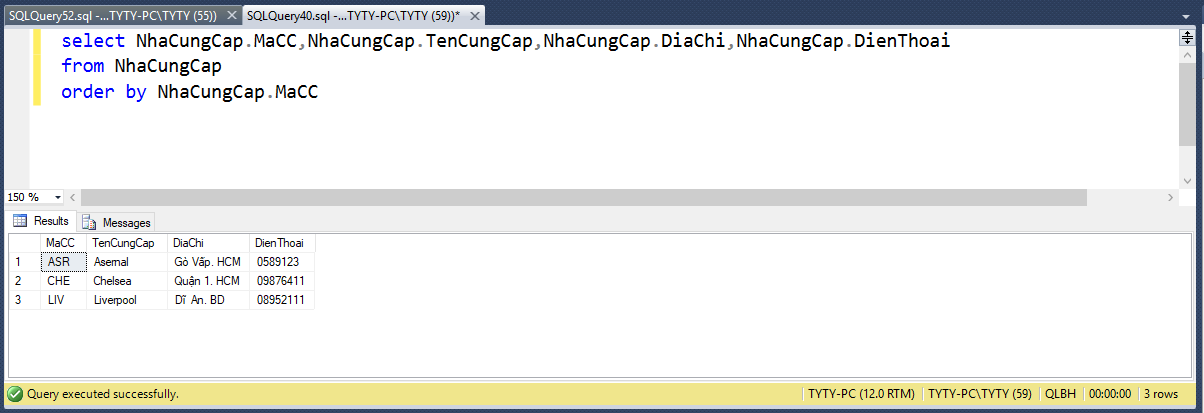
* ***Update***
* The following SQL statement will update the information of “TenCungCap”, “DiaChi”, “DienThoai” columns in “NhaCungCap” table.
* ***Display information***
* Display a table containing information of the Vendors.

SELECT NhaCungCap.MaCC,NhaCungCap.TenCungCap,NhaCungCap.DiaChi,

NhaCungCap.DienThoai

FROM NhaCungCap

ORDER BY NhaCungCap.MaCC

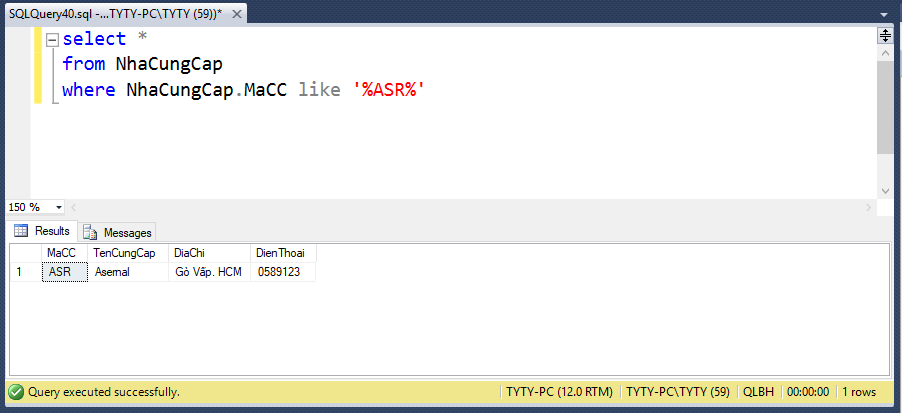


* ***Search information***
* Find information to look up in the Vendor table ( based on Vendor Code/ Vendor Name/ Address/ Phone Number ).
* Based on the Vendor Code.

SELECT \*

FROM NHACUNGCAP

WHERE NHACUNGCAP.MACC LIKE '%'+@MACC+'%'

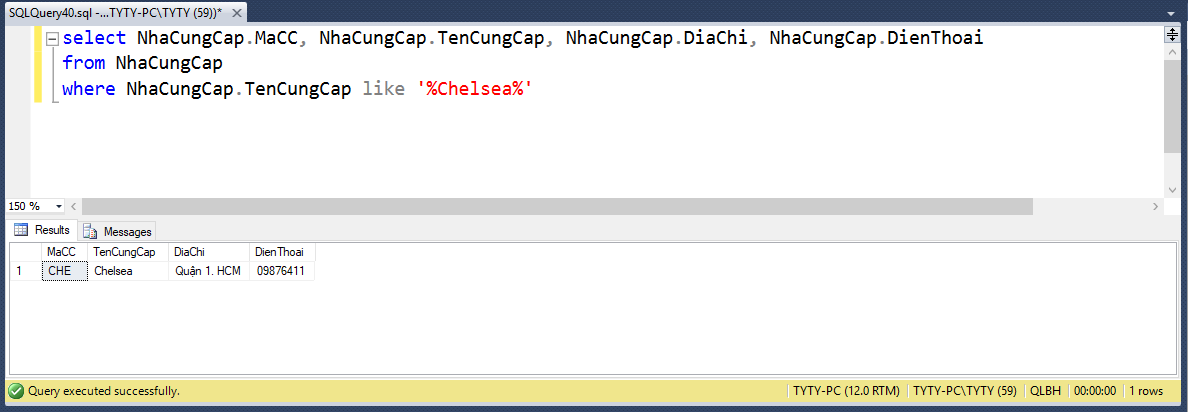


* Based on Vendor Name.

SELECT NhaCungCap.MaCC, NhaCungCap.TenCungCap, NhaCungCap.DiaChi, NhaCungCap.DienThoai

FROM NhaCungCap

WHERE NhaCungCap.TenCungCap like '%'+@TenCungCap+'%'

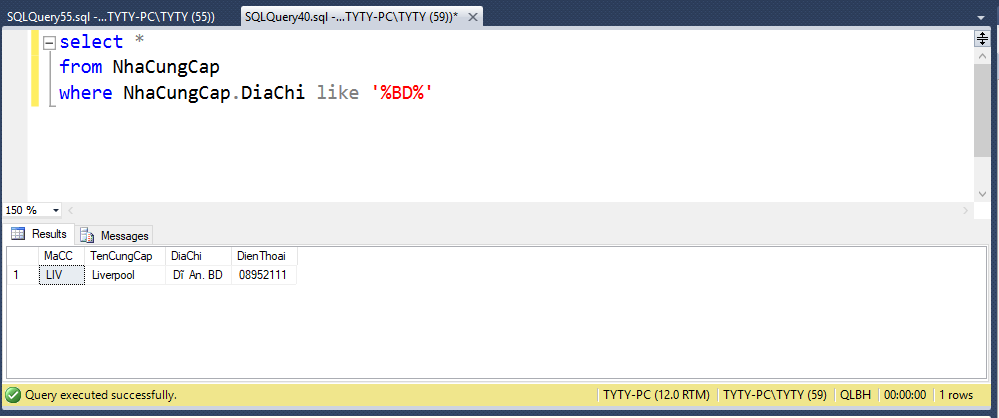


* Based on Address.

SELECT \*

FROM NhaCungCap

WHERE NhaCungCap.DiaChi like '%'+@DiaChi+'%'

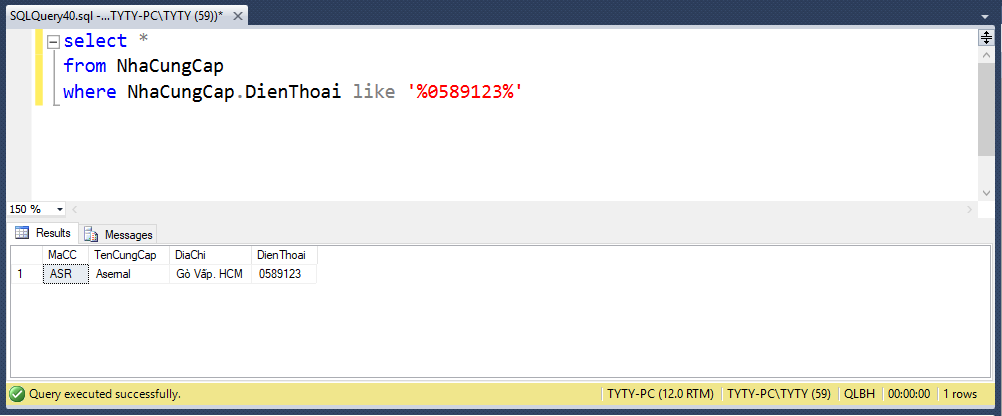


* Based on Phone Number.

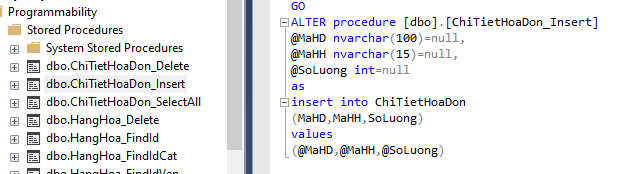
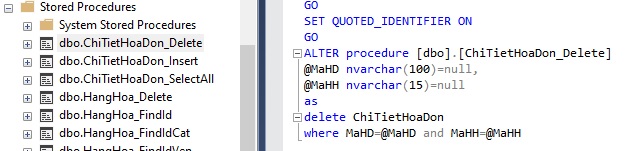
SELECT \*

FROM NhaCungCap

WHERE NhaCungCap.DienThoai like '%'+@DienThoai+'%'



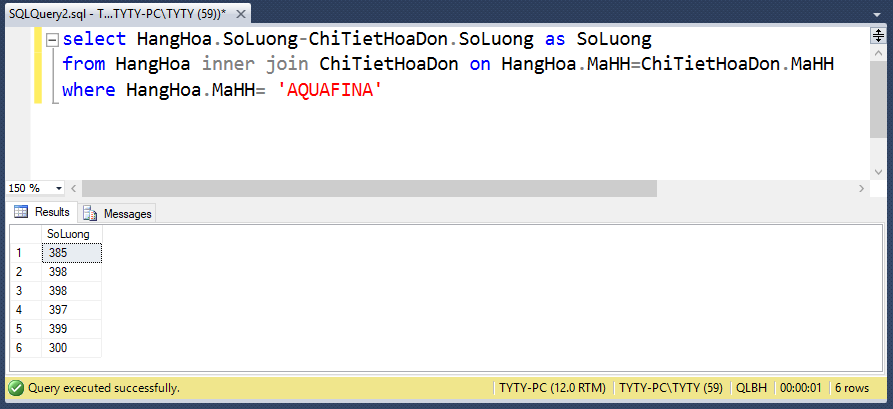
## **4.6 Selling management**

* ***Insert***
* The following SQL statement will insert a new record, insert data in the “MaHD”, “MaHH”, and “SoLuong” columns in “ChiTietHoaDon” table.
* ***Delete***
* The following SQL statement will delete information of “MaHD” and “MaHH” in “ChiTietHoaDon” table.
* ***Update after a complete sale order***
* Record how many goods are in stock after creating the invoice.

SELECT HangHoa.SoLuong-ChiTietHoaDon.SoLuong as SoLuong

FROM HangHoa inner join ChiTietHoaDon on HangHoa.MaHH=ChiTietHoaDon.MaHH

WHERE HangHoa.MaHH=@MaHH

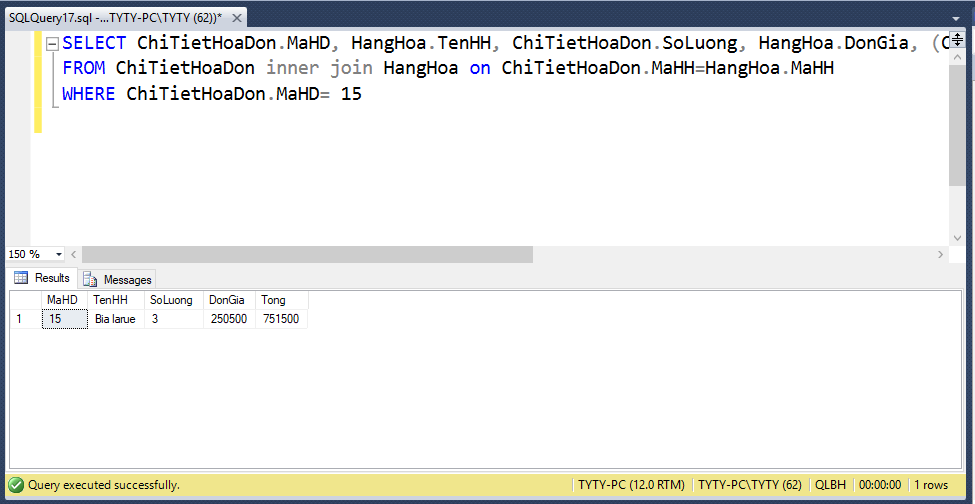


* ***Display invoice detail***
* List invoice details of any invoice based on Invoice code.

SELECT ChiTietHoaDon.MaHD, HangHoa.TenHH, ChiTietHoaDon.SoLuong, HangHoa.DonGia, (ChiTietHoaDon.SoLuong\*HangHoa.DonGia) as Tong

FROM ChiTietHoaDon inner join HangHoa on ChiTietHoaDon.MaHH=HangHoa.MaHH

WHERE [ChiTietHoaDon.MaHD=@MaHD](mailto:ChiTietHoaDon.MaHD=@MaHD)



* ***Display sold product***
* Display a table describing sales statistics of the store.

SELECT HangHoa.TenHH, sum(ChiTietHoaDon.SoLuong) as SLBan

FROM ChiTietHoaDon inner join HangHoa on ChiTietHoaDon.MaHH=HangHoa.MaHH

GROUP BY HangHoa.TenHH

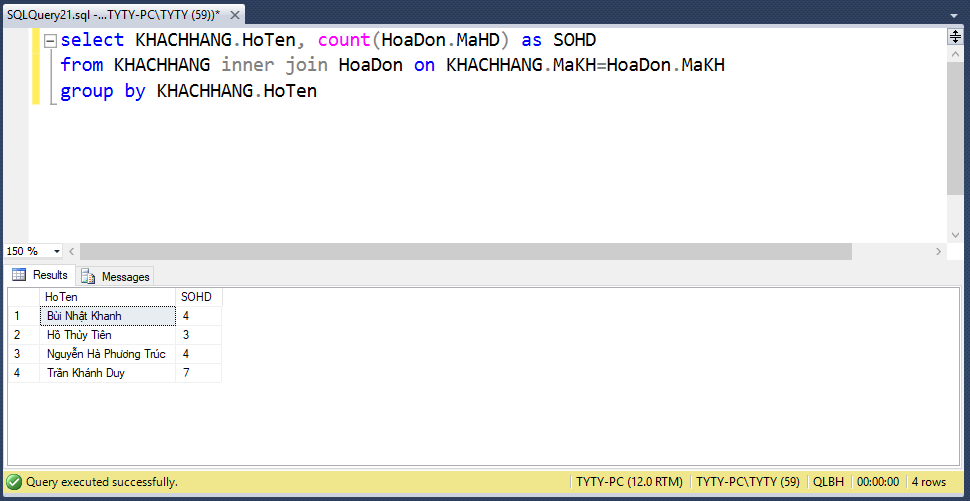


* ***Display a statistic of the invoice per customer***
* Display a statistic table of the invoices of the customers who bought goods from the store.

SELECT KHACHHANG.HoTen, count(HoaDon.MaHD) as SOHD

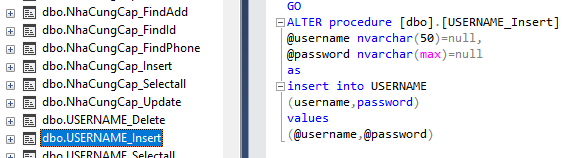
FROM KHACHHANG inner join HoaDon on KHACHHANG.MaKH=HoaDon.MaKH

GROUP BY KHACHHANG.HoTen



## **4.7 Sign-in**

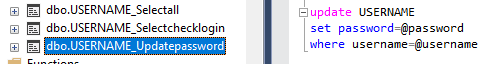
* ***Insert***
* The following SQL statement will insert a new record, insert data in the “username”, “password” columns in “USERNAME” table.



* ***Delete***
* The following SQL statement will delete information of “username” in “USERNAME” table.

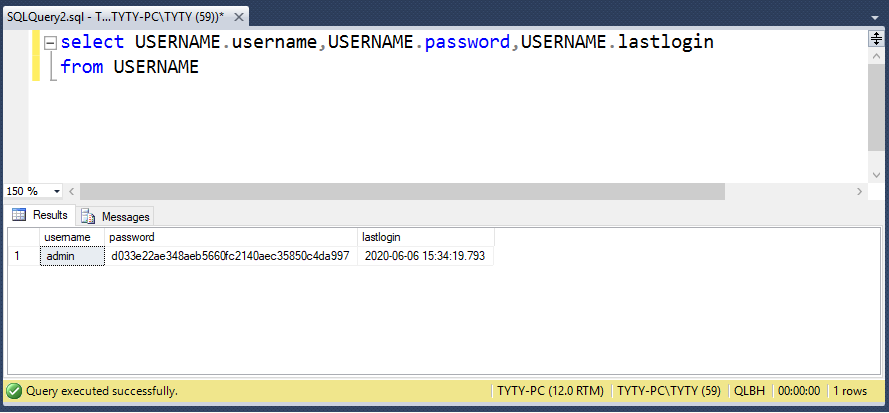


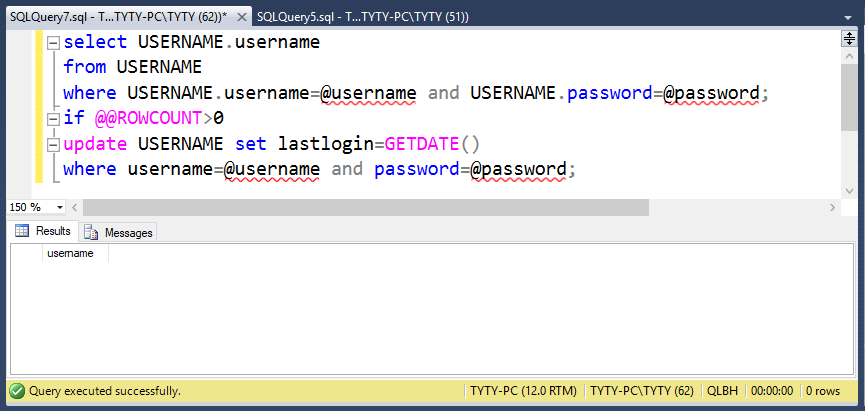
* ***Update***
* The following SQL statement will update the information of “password” columns in “USERNAME” table.



* ***Display information***
* View user account information ( including username, password, last log in).

SELECT USERNAME.username,USERNAME.password,USERNAME.lastlogin

FROM USERNAME

* ***Check log in***
* Check if the username and password entered are correct or incorrect.

# **CHAPTER 5: APPLICATION**

## **5.1 Platform**

* This application is used for the sales management of grocery. So, we decided to build this application with WinForm for these reasons:
* The internet is not needed in this situation.
* The data is not too complicated because grocery has no department division.
* The grocery meets the needs of facilities easily.

## **5.2 Business Processing**

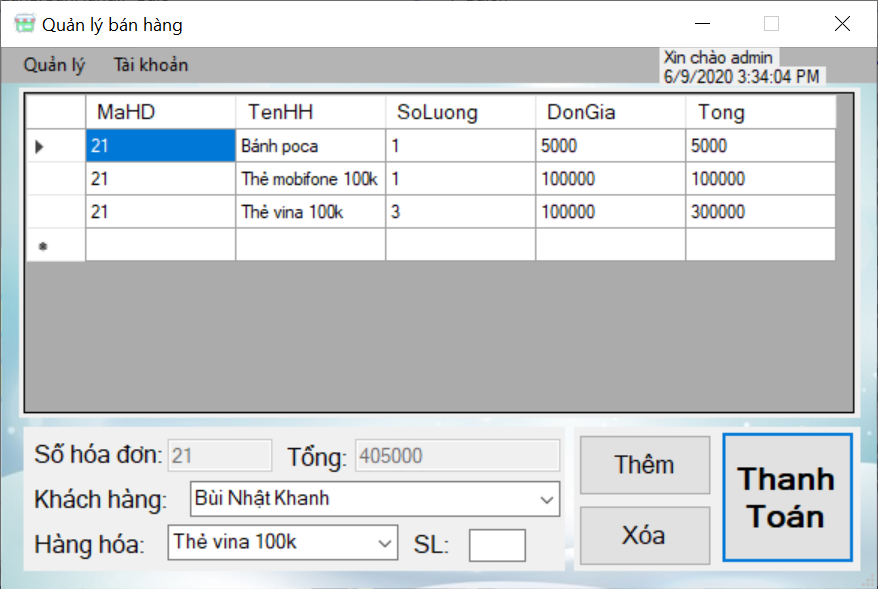
### ***5.2.1 Sign in***

* The grocery owner signs in with his/her username and password. Because the scale here is not big, new users will be added in the database manually. Otherwise, for the security issue, on the user display, there is only Sign in function.

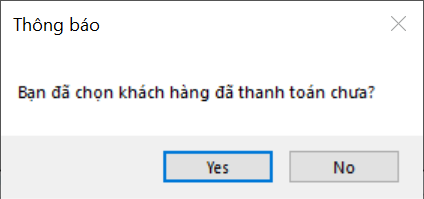


### ***5.2.2 Create invoices***

* The main function of grocery is sales. In this process, the owner can create invoices to make revenue recognition.
* The ID of billing will be created automatically. The first step is choosing the customer’s name in your list. Then, choose the products and quantities before adding information on sales into the invoice. The total invoice is displayed immediately after each product added.

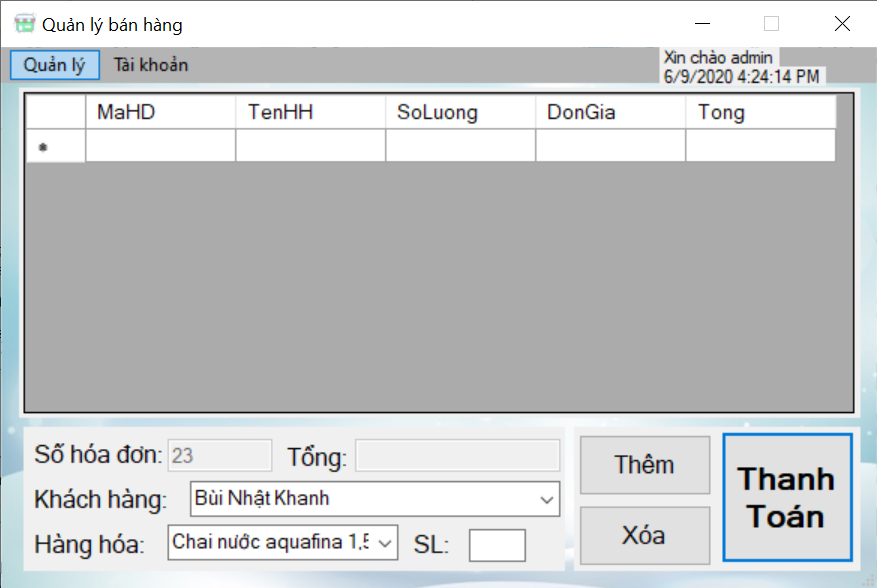


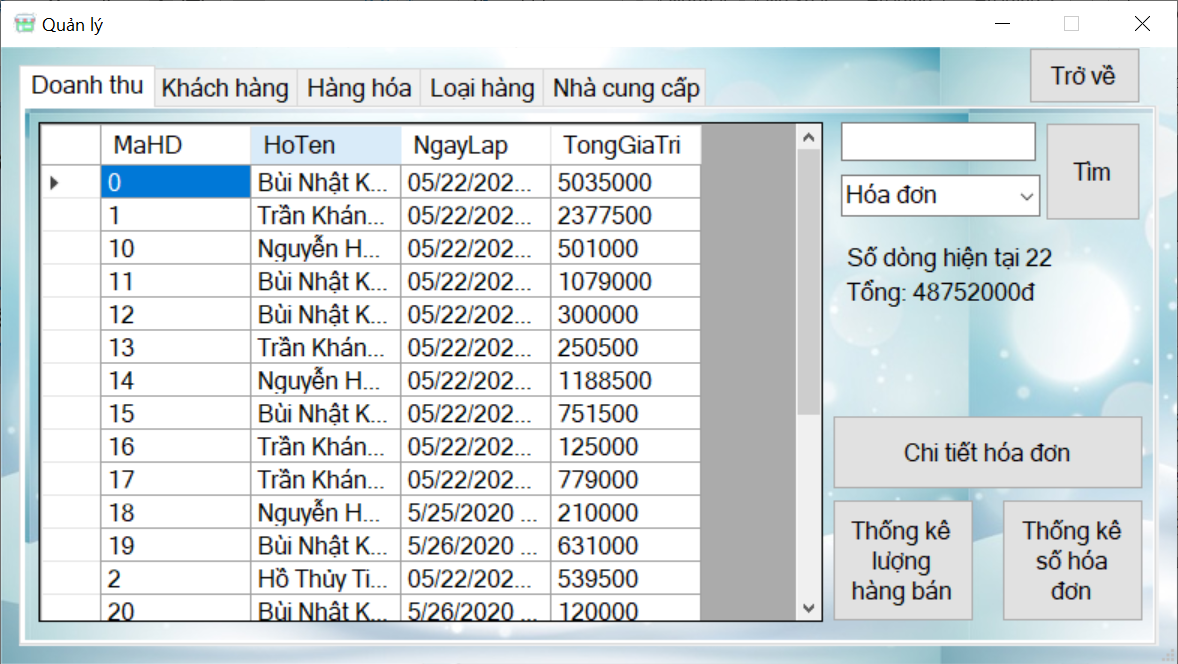
* If customers want to change the products or quantities, there is a function to delete whatever they want, and then add again the right ones.
* The system will record the invoice when the owner presses the button “Thanh toán”. Besides, to control some of the careless situations, the notification will appear to confirm the customer who will pay for this invoice.

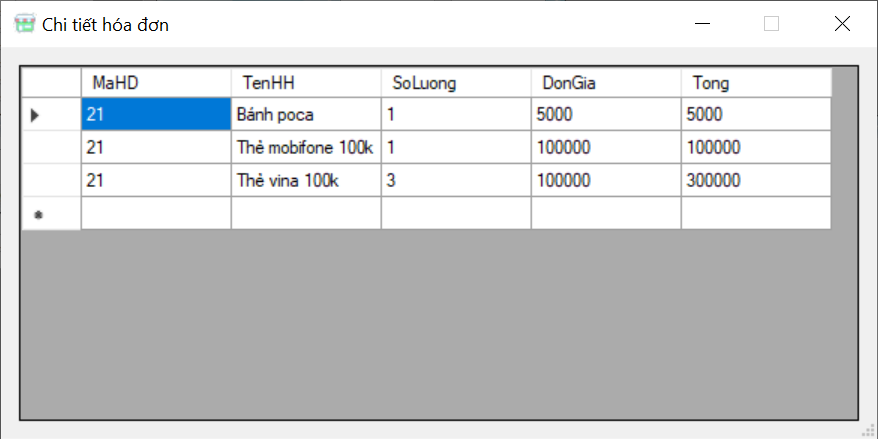


### ***5.2.3 Manage the grocery***

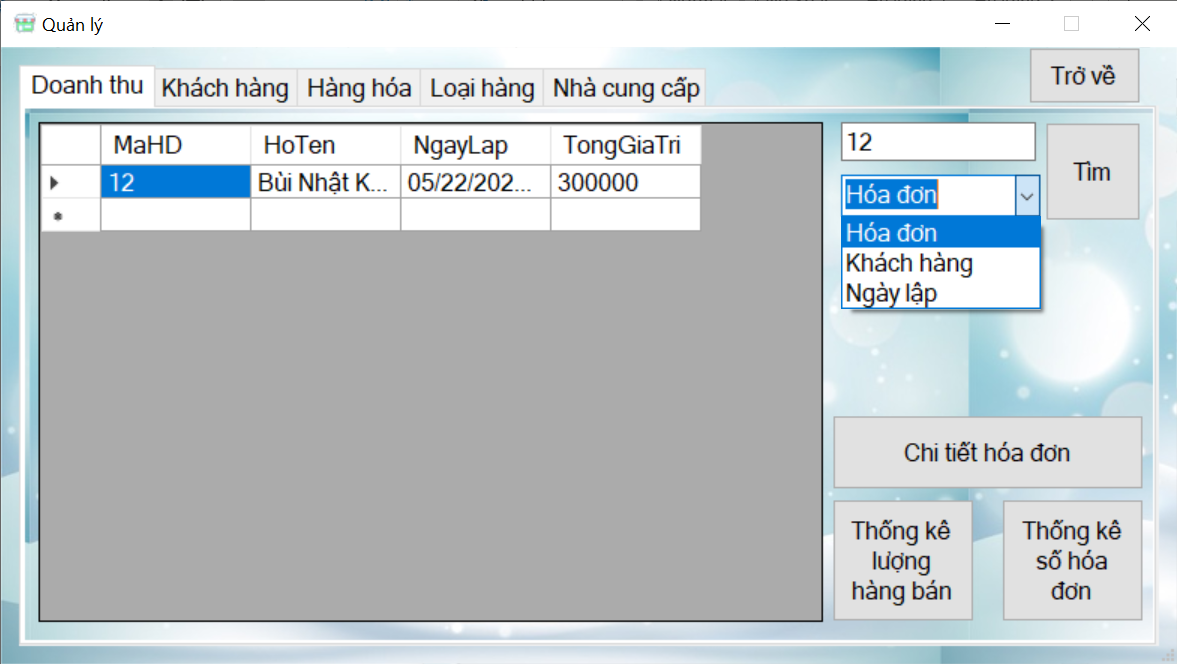
* To enter the management function, press the Tab Control named “Quản lý”. In this process, the owner can statistic, add, find, delete revenue, information of customers, inventory, and suppliers.



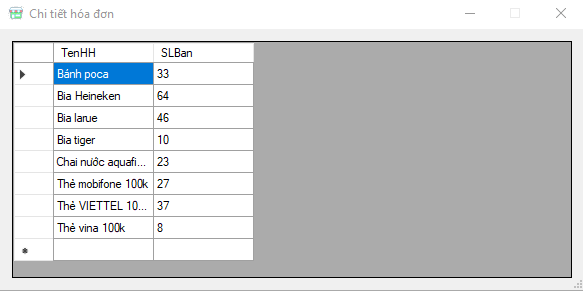
* ***Revenue***
* In the Tab Control “Doanh thu”, all of the invoices are displayed. This application also exhibits the revenue until then and how many invoices were sold.
* On the other hand, you can show invoice details such as number, unit price, etc. when clicking on “Chi tiết hóa đơn”.



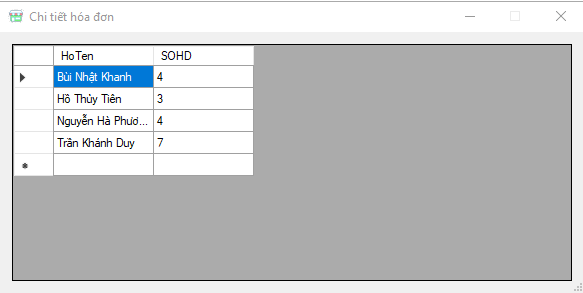
* For a specific purpose, the owner can find out which invoice he/ she wants with a key search like ID, customer, or date.



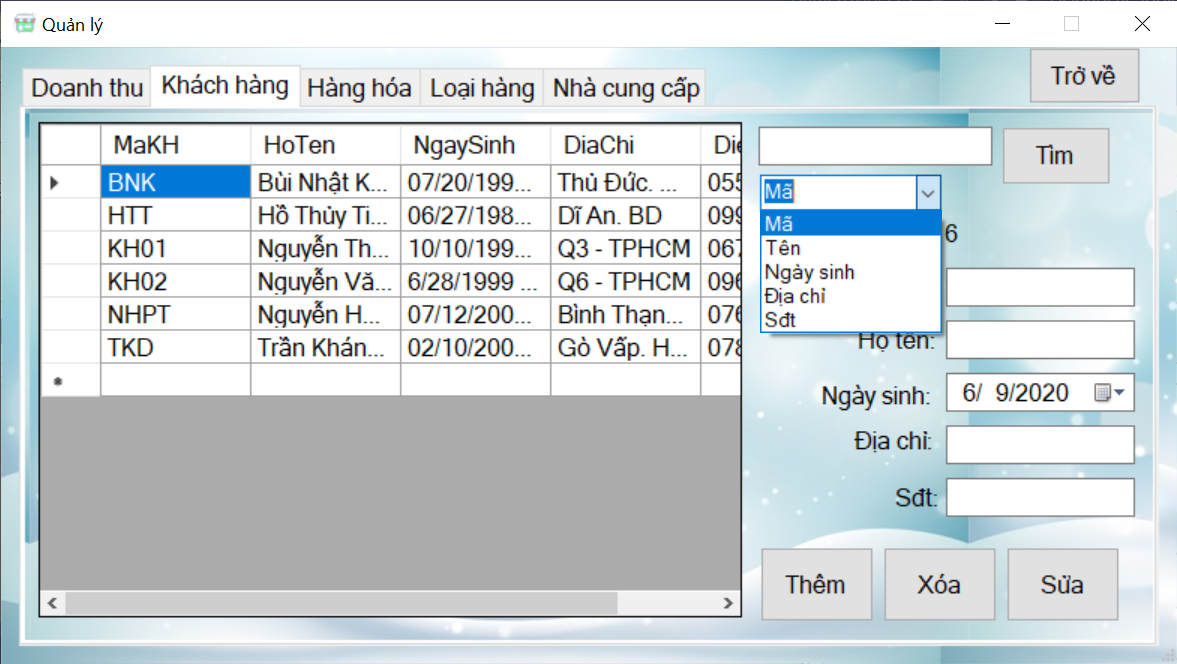
* Another function is statistic the quantities of all products which the grocery has sold since the grand opening. The sales situation will be shown clearly and can help the owner give right and a timely decision like promotion, how many units need to import, so on.



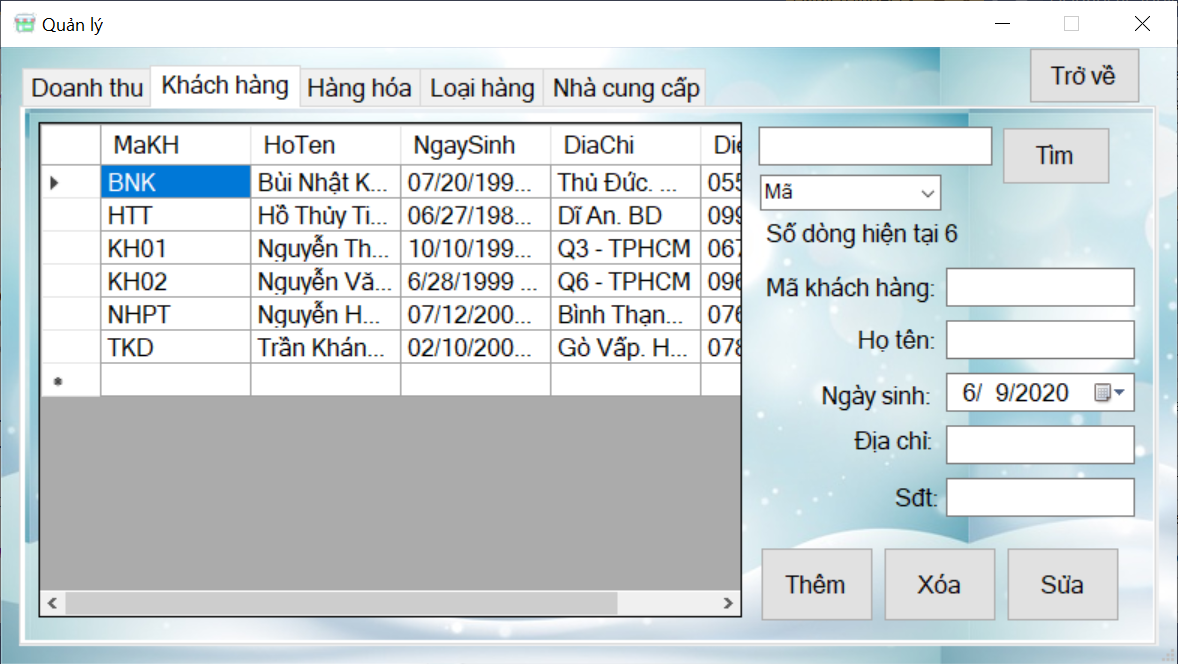
* The statistic of the number of invoices is the same purpose as quantities of product.



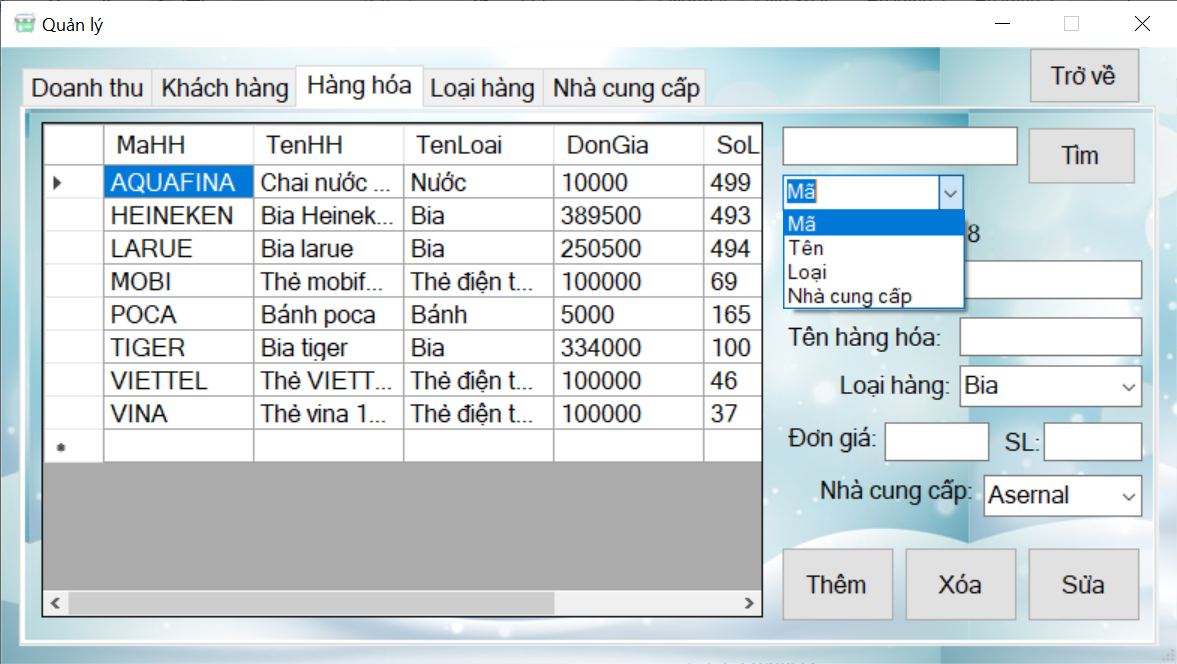
* ***Customers***
* The same as Revenue, this process also has a display, search information of customers with ID, name, phone number, address, etc.



* Furthermore, the system has a function to add, change, or remove customers. After filling the necessary data in the textbox, the table will be updated speedily.



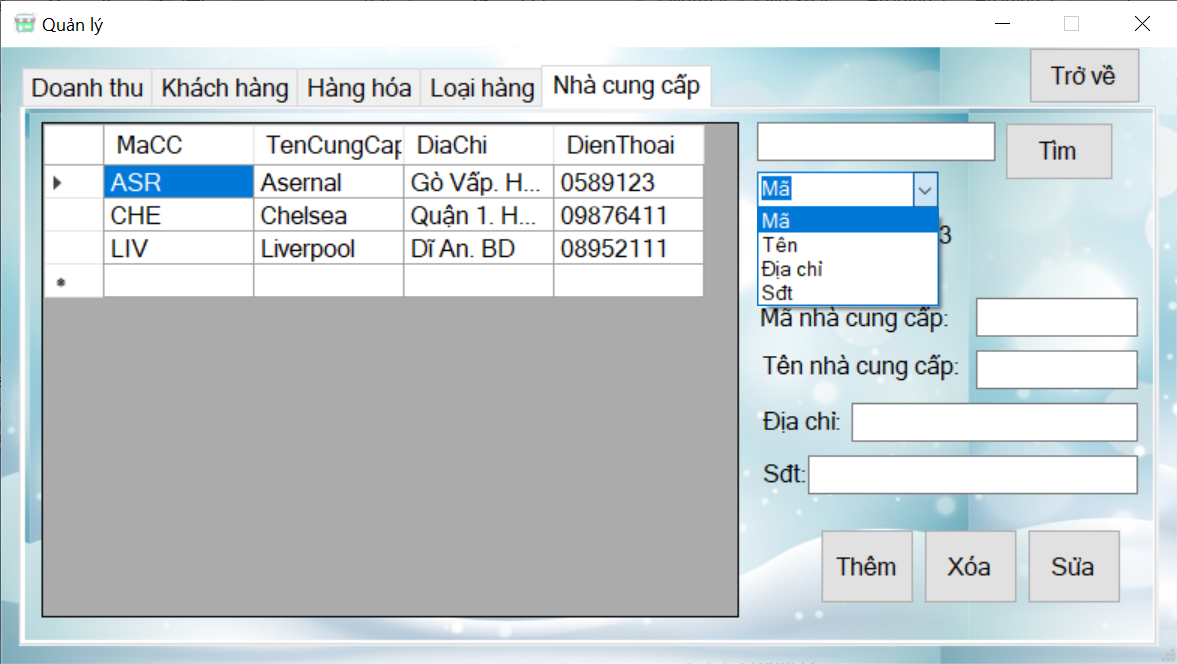
* ***Products***
* This process is used the same as inventory. It means that the owner can see the stock-in-trade of each product. Otherwise, when the owner imports good, he can also update here.



* The search function helps users easier in finding something for specific purposes.
* ***Category***
* To systemize the information of grocery, products are divided into classes. Through the revenue of each class, the owner can arrange products in a suitable situation of grocery to promote sales effectively.

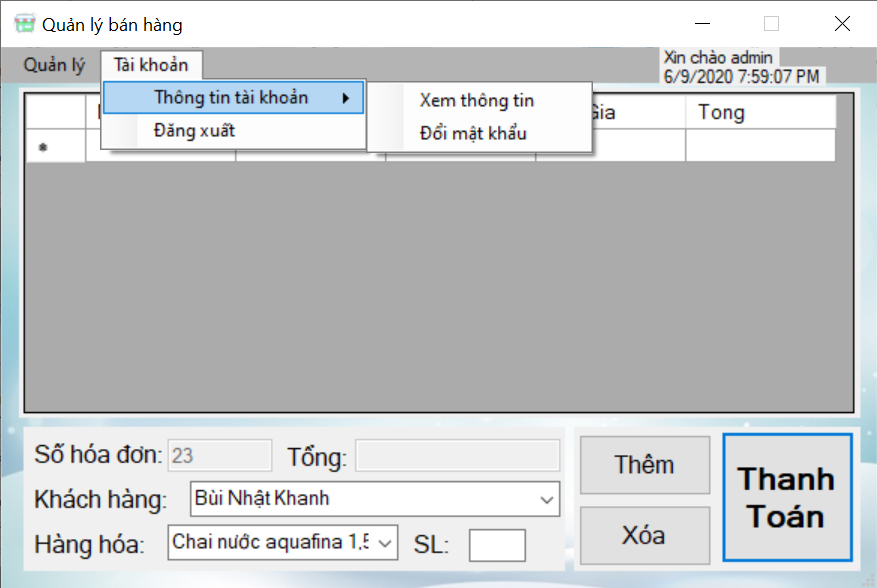


* ***Suppliers***
* The list of suppliers is displayed in this step. The attributes are ID, name, the contact information.

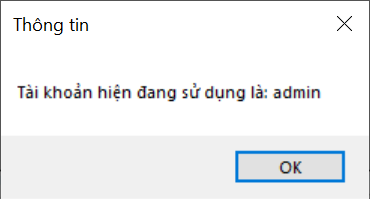


### ***5.2.4 Manage user accounts***

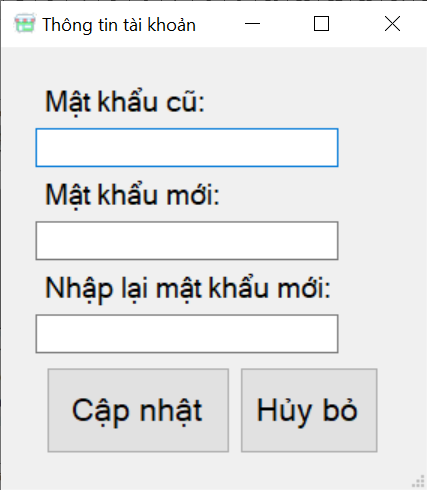
* ***Information***



* To distinguish and recognize who is using this application, the Tab Control “Tài khoản” has a tab that exhibits information about users. The window “Thông tin” shows a message which provides username using at that moment.

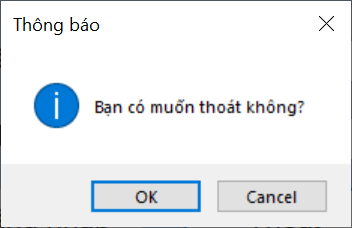


* ***Change password***
* You can change your password whenever you want in “Đổi mật khẩu”. In this step, you need to fill the previous password to confirm this account is yours. After entering a new password, you press the button “Cập nhật” to update the database about users.



### ***5.2.5 Log out***

* After using this application, the owner can end all tasks with an exit button. To make sure you want to close this application, the system shows the message to confirm it.



# **CHAPTER 6: EVALUATION THE PROJECT**

## **6.1 Advantage**

* Building ERD and relational tables on SQL Server for this project improve our knowledge about this subject.
* Our group has identified the basic problem in the management process and recognized the data management object such as customer, invoice, product, etc. And then, we identified the information to be managed and build the ERD model based on the requirement of the data.
* The database structure of our project solved management issues of information for products, customers, suppliers, invoices, etc. Besides, there is some function to access and evaluate information such as calculation and statistics to help users' requirements in managing and deciding.

## **6.2 Disadvantage**

* The user interface is still rudimentary, not really professional, and barely solved the issues related to management activities.
* Data solution and table structure construction help the administrator easily manage and help data structure more closely but bring complexity to users.
* The application is only suitable for a grocery store and needs to upgrade a lot to implementation for the larger business.