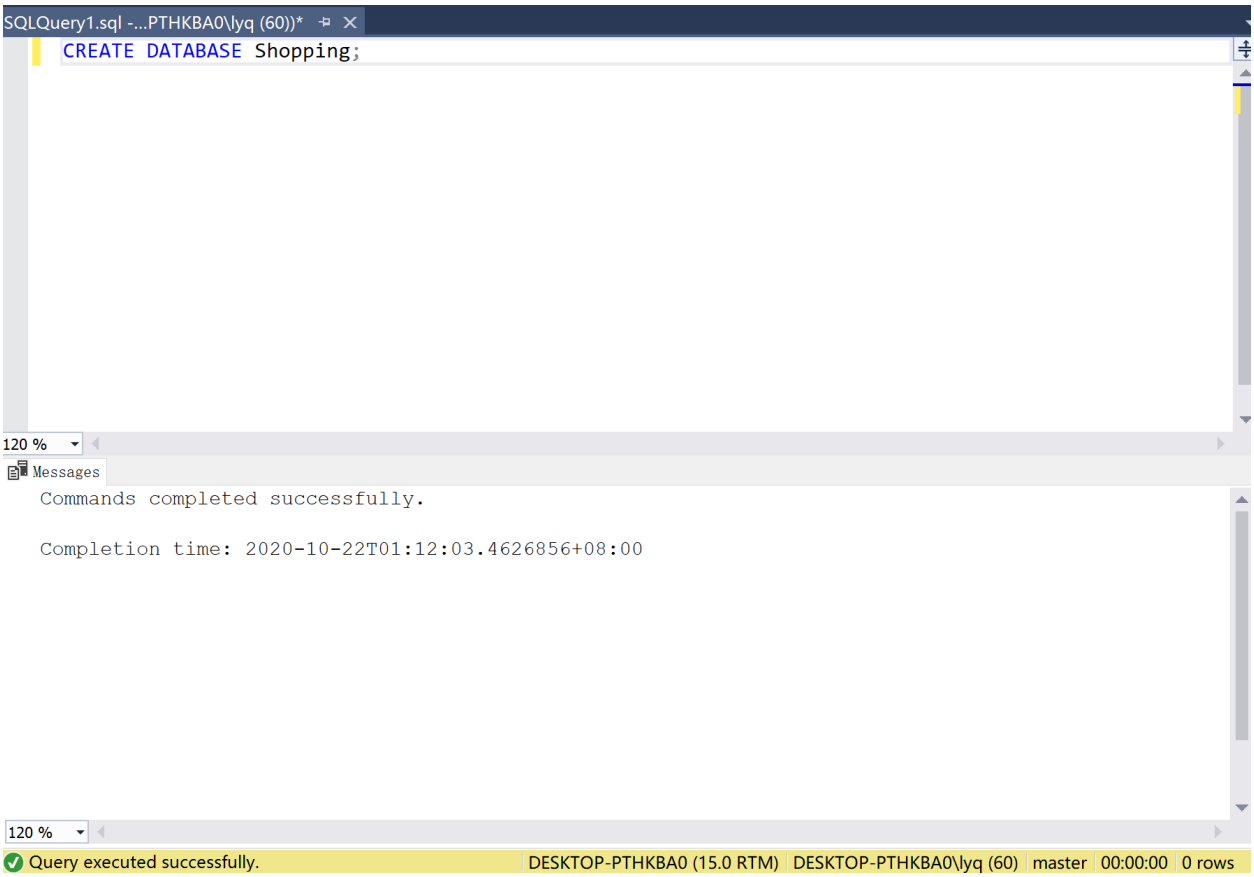
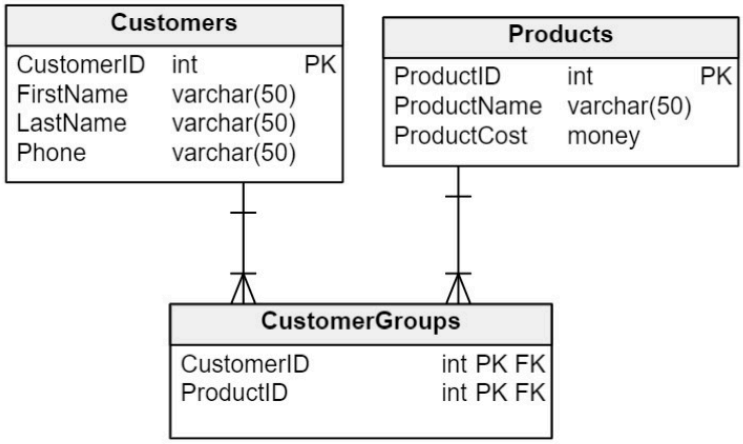
Lab 7: Database Implementation

1. Create a new database named Shopping.

Here is the screenshot:



2.(1) Describe the relationship type shown in figure (one-to-one, one-to-many or many-to-many).



The relationship between Customers and CustomerGroups is one-to-many, and the relationship between CustomerGroups and Products is many-to-one.

(2) Write the CREATE TABLE statements needed to implement the following design in the Shopping database. Include foreign key constraints. Define CustomerID and ProductID as identity columns. Decide which columns should allow null values, if any, and explain your decision. Define the ProductCost column with a default of zero and a check constraint to allow only positive values.

```
CREATE TABLE Customers
(CustomerID INT PRIMARY KEY  IDENTITY,
FirstName VARCHAR(50) NOT NULL,
LastName VARCHAR(50) NOT NULL,
Phone VARCHAR(50) NOT NULL)

CREATE TABLE Products
(ProductID INT PRIMARY KEY IDENTITY,
ProductsName VARCHAR(50) NOT NULL,
ProductCost MONEY NOT NULL DEFAULT 0 CHECK (ProductCost>=0))

CREATE TABLE CustomerGroups
(CustomerID INT REFERENCES Customers(CustomerID),
ProductID INT REFERENCES Products(ProductID));
```

120 %

Messages

```
Commands completed successfully.

Completion time: 2020-10-22T01:34:55.1082228+08:00
```

120 %

Query executed successfully. DESKTOP-PTHKBA0 (15.0 RTM) DESKTOP-PTHKBA0\lyq (51) Shopping 00:00:00 0 rows

3. Write the CREATE INDEX statements to create a clustered index on the ProductID column and a nonclustered index on the CustomerID column of the CustomerGroups table.

SQLQuery3.sql -...PTHKBA0\lyq (61))* | SQLQuery2.sql -...PTHKBA0\lyq (51))* | SQLQuery1.sql -...PTHKBA0\lyq (60))*

```sql
USE Shopping
CREATE CLUSTERED INDEX IndexofProductID
ON CustomerGroups(ProductID)
CREATE INDEX IndexofCustomerID
ON CustomerGroups(CustomerID);
```

120 %

Messages

```
Commands completed successfully.

Completion time: 2020-10-22T01:42:55.7978352+08:00
```

120 %

Query executed successfully. | DESKTOP-PTHKBA0 (15.0 RTM) | DESKTOP-PTHKBA0\lyq (61) | Shopping | 00:00:00 | 0 rows

4. Write an ATER TABLE statement that adds a new column, BillsPaid, to the Customers table. Use the bit data type, disallow null values, and assign a default Boolean value of False.

SQLQuery4.sql -...PTHKBA0\lyq (64)) | SQLQuery3.sql -...PTHKBA0\lyq (61)) | SQLQuery2.sql -...PTHKBA0\lyq (51))

```sql
USE Shopping
ALTER TABLE Customers
ADD BillsPaid bit NOT NULL DEFAULT 0;
```

120 %

Messages

```
Commands completed successfully.

Completion time: 2020-10-22T01:49:19.2678952+08:00
```

120 %

Query executed successfully. | DESKTOP-PTHKBA0 (15.0 RTM) | DESKTOP-PTHKBA0\lyq (64) | Shopping | 00:00:00 | 0 rows

5. Write an ALTER TABLE statement that adds two new check constraints to the Invoices table (in AP database) of the AP database. The first should allow (1) PaymentDate to be null only if PaymentTotal is zero and (2) PaymentDate to be not null only if PaymentTotal is greater than zero. The second constraint should prevent the sum of PaymentTotal and CreditTotal from being greater than InvoiceTotal.

SQLQuery5.sql -...PTHKBA0\lyq (62)) | SQLQuery4.sql -...PTHKBA0\lyq (64)) | SQLQuery3.sql -...PTHKBA0\lyq (61))

```sql
USE AP
ALTER TABLE Invoices
ADD CHECK ((PaymentTotal=0 AND PaymentDate IS NULL) OR
           (PaymentTotal>0 AND PaymentDate IS NOT NULL)),
    CHECK ((PaymentTotal + CreditTotal) <= InvoiceTotal);
```

120 %

Messages

```
Commands completed successfully.

Completion time: 2020-10-22T01:57:26.9272012+08:00
```

120 %

Query executed successfully. | DESKTOP-PTHKBA0 (15.0 RTM) | DESKTOP-PTHKBA0\lyq (62) | AP | 00:00:00 | 0 rows

6. Delete the CustomerGroups table from the Shopping database. Then, write a CREATE TABLE statement that recreates the table, this time with a unique constraint that prevents a customer from being a customer-group member buying same product twice.

SQLQuery6.sql -...PTHKBA0\lyq (63))* | SQLQuery5.sql -...PTHKBA0\lyq (62)) | SQLQuery4.sql -...PTHKBA0\lyq (64))

```sql
USE  Shopping
DROP TABLE CustomerGroups;
```

120 %

Messages

```
Commands completed successfully.

Completion time: 2020-10-22T01:59:01.3069755+08:00
```

120 %

Query executed successfully. | DESKTOP-PTHKBA0 (15.0 RTM) | DESKTOP-PTHKBA0\lyq (63) | Shopping | 00:00:00 | 0 rows

SQLQuery6.sql -...PTHKBA0\lyq (63)) | SQLQuery5.sql -...PTHKBA0\lyq (62)) | SQLQuery4.sql -...PTHKBA0\lyq (64))

```sql
USE  Shopping
CREATE TABLE CustomerGroups
 (CustomerID INT REFERENCES Customers(CustomerID),
 ProductID INT REFERENCES Products(ProductID),
 UNIQUE(CustomerID,ProductID))
```

120 %

Messages

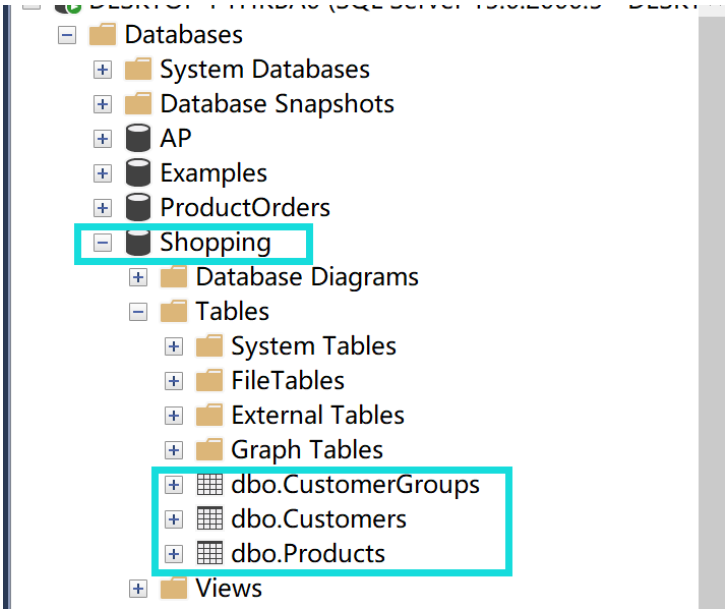
```
Commands completed successfully.

Completion time: 2020-10-22T02:05:34.0823515+08:00
```

120 %

Query executed successfully. | DESKTOP-PTHKBA0 (15.0 RTM) | DESKTOP-PTHKBA0\lyq (63) | Shopping | 00:00:00 | 0 rows

Here is the screenshot of my final database with tables:



7. Use the Management Studio to create a new database called Shopping2 using the default settings. (Do not use SQL query to do this).

The screenshots are as below: