JULY 18, 2021

ASSIGNMENT

DBI202 – ECOMMERCE

STUDENT NAME: DANG MINH ANH | HE153761

STUDENT NAME: PHAM KHOI NGUYEN | HE150978

TEACHER: LAI HIEN PHUONG

TABLE OF CONTENTS

I) INTRODUCE THE SYSTEM

For each user, we have user ID, password, name, phone, email, birthday, a list of address and a cart which include products have been added but not buy.

Each address including adress ID, ID of user who own this address, customerName, ship address and phone number.

Cart of a user include user ID who own this cart, list of products and for each product the quantity.

Products are organized into different categories.

For each product, we have product ID, product name, unit price, description, quantity has sale and quantity in stock.

A product category has category ID, name and description.

Note that each product belongs to a single category and being provided by a single supplier. Information about the suppliers includes the supplier ID, company name, contact name, address, phone, email.

Sales of products are grouped into order including: order ID, ID of user who create this order, the date of the sale, the date of delivery, ship company, ship address, ship cost, payment method, status, list of products sold.

And for each product, the quantity sold, the sell price and note.

Information about shipcompany include ship company ID, name, phone and email.

II) ENTITY – RELATIONSHIP – ER

1) DIFINITION ENTITY – ATTRIBUTE

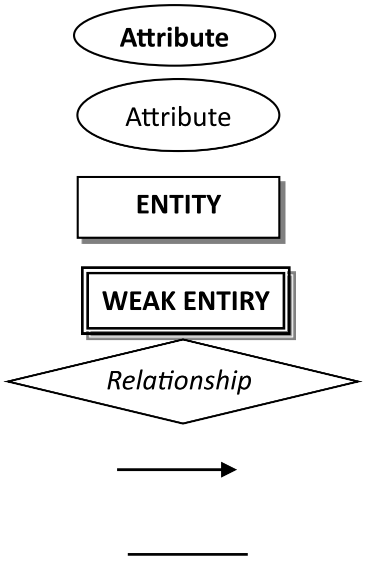
Base on the problem description and management objectives, we can present several entities and attributes of the entity as follow:

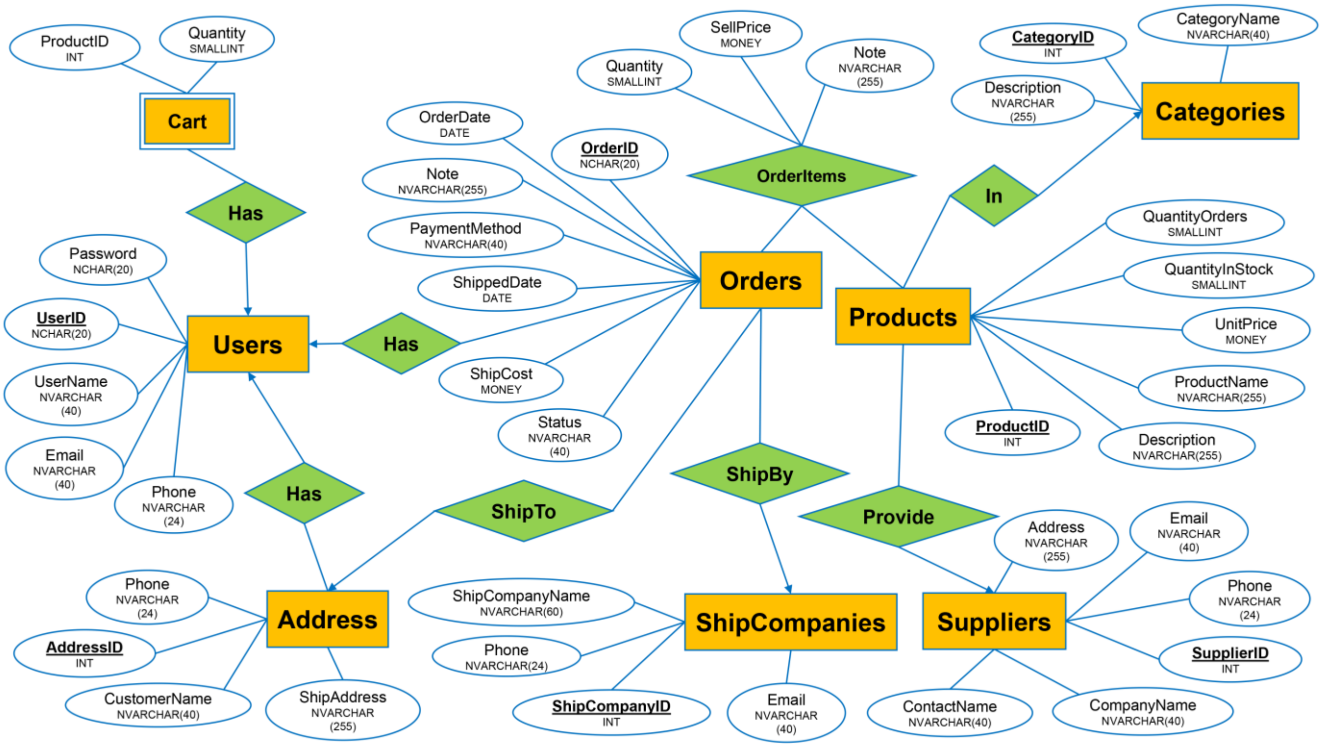
- Users : **UserID**, User Name, Password, Phone, Email.
- Address**: AddressID**, **UserID**, Customer Name, Ship Address, Phone
- Categories: **CategoryID**, Category Name, Description
- Suppliers: **SupplierID**, Company Name, Contact Name, Address, Phone, Email
- Products: **ProductID**, Product Name,Unit Price, Quantity Orders, Quantity In Stock, Description, **CategoryID**, **SupplierID**
- ShipCompanies: **ShipCompanyID**, ShipCompanyName, Phone, Email
- Orders: **OrderID**, Order Date, Ship Date, Ship Cost, Payment Method, Status, **UserID**, **ShipCompanyID**, **AddressID**, Note
- OrderItems: **ProductID**, **OrderID**, Quantity, SellPrice, Note

2) SET-UP ENTITY – RELATIONSHIP

* Some symbols used in the model

- Key / identifier attribute
- Attribute description / description
- Entity
- Weak entity
- Relationship
- Connectivity (force) = 1
- Connectivity = N





III) DATA DICTIONARY

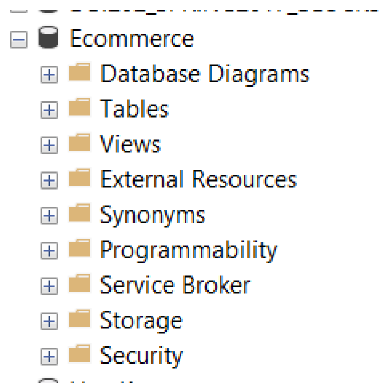
Just for example on some tables (other table are similar, you have to define all the tables in your database).

1) DATABASE AND TABLE

A. CREATE DATABASE **ECOMMERCE**

--create database

```
CREATE DATABASE Ecommerce
```



- Ecommerce
  - Database Diagrams
  - Tables
  - Views
  - External Resources
  - Synonyms
  - Programmability
  - Service Broker
  - Storage
  - Security

B. CREATE TABLE **USERS**

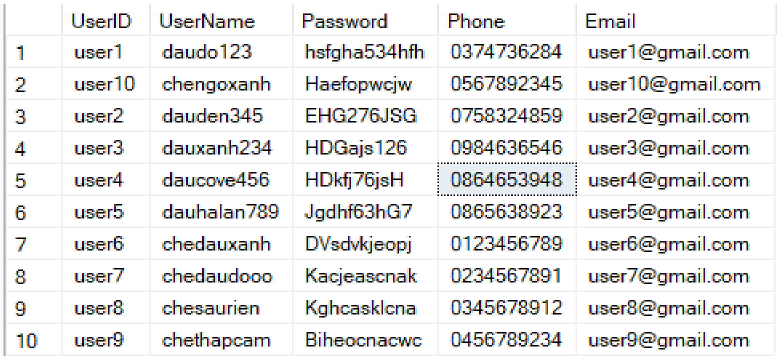
Column Name	Data Type	Default	Check	Key/ Index/ Constraint
UserID	Nchar(20)			Primary key
UserName	Nvarchar(40)			Unique, Not null
Password	Nchar(20)			Not null
Email	Nvarchar(40)			Unique, Not null
Phone	Nvarchar(24)		0 to 9	Unique, Not null

Code:

```
--create table Users

CREATE TABLE Users(
       UserID NCHAR(20) NOT NULL,
       UserName NVARCHAR(40) NOT NULL,
       Password NCHAR(20) NOT NULL,
       Phone NVARCHAR(24) CHECK(Phone LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
       NOT NULL,
       Email NVARCHAR(40) NOT NULL,
       UNIQUE(UserName, Phone, Email),
       PRIMARY KEY(UserID)
)
```

Example:



	UserID	UserName	Password	Phone	Email
1	user1	daudo123	hsfgha534hfh	0374736284	user1@gmail.com
2	user10	chengoxanh	Haefopwcjw	0567892345	user10@gmail.com
3	user2	dauden345	EHG276JSG	0758324859	user2@gmail.com
4	user3	dauxanh234	HDGajs126	0984636546	user3@gmail.com
5	user4	daucove456	HDkfj76jsH	0864653948	user4@gmail.com
6	user5	dauhalan789	Jgdhf63hG7	0865638923	user5@gmail.com
7	user6	chedauxanh	DVsdvkjeopj	0123456789	user6@gmail.com
8	user7	chedaudooo	Kacjeascnak	0234567891	user7@gmail.com
9	user8	chesaurien	Kghcasklcna	0345678912	user8@gmail.com
10	user9	chethapcam	Biheocnacwc	0456789234	user9@gmail.com

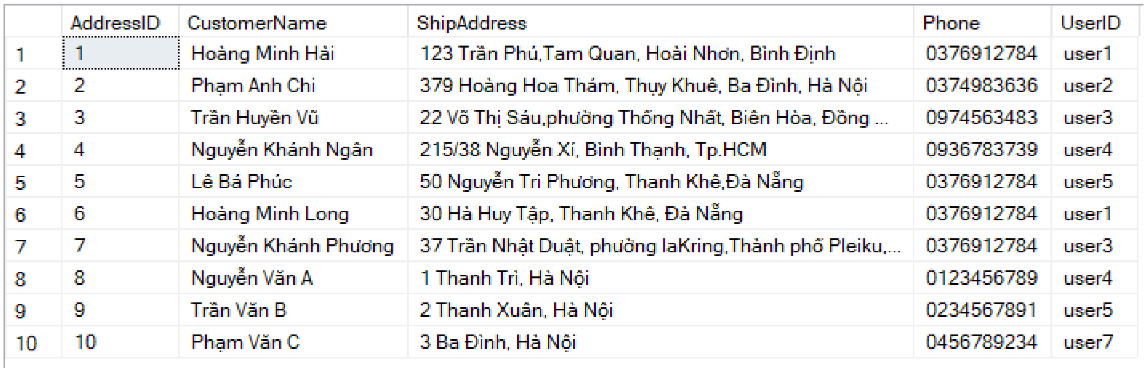
C. CREATE TABLE **ADDRESS**

Column Name	**Data Type**	**Default**	**Check**	**Key/ Index/ Constraint**
AddressID	Int			Primary Key
CustomerName	Nvarchar(40)			Not null
ShipAddress	Nvarchar(255)			Not null
Phone	NVarchar(24)		0 to 9	Not null
UserID	Nchar(20)			Not null, FK references Users(UserID)

Code:

```
CREATE TABLE Address(
	AddressID INT NOT NULL,
	CustomerName NVARCHAR(40) NOT NULL,
	ShipAddress NVARCHAR(255),
	Phone NVARCHAR(24) CHECK(Phone LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
	NOT NULL,
	UserID NCHAR(20) REFERENCES Users(UserID) NOT NULL,
	PRIMARY KEY(AddressID)
)
```

Example:



	AddressID	CustomerName	ShipAddress	Phone	UserID
1	1	Hoàng Minh Hải	123 Trần Phú,Tam Quan, Hoài Nhơn, Bình Định	0376912784	user1
2	2	Phạm Anh Chi	379 Hoàng Hoa Thám, Thụy Khuê, Ba Đình, Hà Nội	0374983636	user2
3	3	Trần Huyền Vũ	22 Võ Thị Sáu,phường Thống Nhất, Biên Hòa, Đồng ...	0974563483	user3
4	4	Nguyễn Khánh Ngân	215/38 Nguyễn Xí, Bình Thạnh, Tp.HCM	0936783739	user4
5	5	Lê Bá Phúc	50 Nguyễn Tri Phương, Thanh Khê,Đà Nẵng	0376912784	user5
6	6	Hoàng Minh Long	30 Hà Huy Tập, Thanh Khê, Đà Nẵng	0376912784	user1
7	7	Nguyễn Khánh Phương	37 Trần Nhật Duật, phường laKring,Thành phố Pleiku,...	0376912784	user3
8	8	Nguyễn Văn A	1 Thanh Trì, Hà Nội	0123456789	user4
9	9	Trần Văn B	2 Thanh Xuân, Hà Nội	0234567891	user5
10	10	Phạm Văn C	3 Ba Đình, Hà Nội	0456789234	user7

A.)CREATE TABLE **SHIPCOMPANIES**

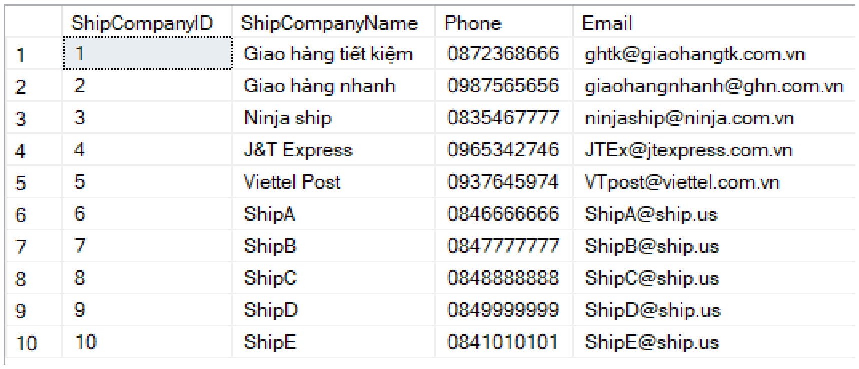
Column Name	**Data Type**	**Default**	**Check**	**Key/ Index/ Constraint**
ShipCompanyID	Int			Primary key
ShipCompanyName	Nvarchar(40)			Not null
Phone	Nvarchar(20)		0 to 9	Not null, Unique
Email	Nvarchar(40)			

Code:

```
--create table ShipCompanies

CREATE TABLE ShipCompanies(
        ShipCompanyID INT NOT NULL,
        ShipCompanyName NVARCHAR(40) NOT NULL,
        Phone NVARCHAR(24) CHECK(Phone LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
        NOT NULL UNIQUE,
        Email NVARCHAR(40) ,
        PRIMARY KEY(ShipCompanyID)
)
```

Example:



	ShipCompanyID	ShipCompanyName	Phone	Email
1	1	Giao hàng tiết kiệm	0872368666	ghtk@giaohangtk.com.vn
2	2	Giao hàng nhanh	0987565656	giaohangnhanh@ghn.com.vn
3	3	Ninja ship	0835467777	ninjaship@ninja.com.vn
4	4	J&T Express	0965342746	JTEx@jtexpress.com.vn
5	5	Viettel Post	0937645974	VTpost@viettel.com.vn
6	6	ShipA	0846666666	ShipA@ship.us
7	7	ShipB	0847777777	ShipB@ship.us
8	8	ShipC	0848888888	ShipC@ship.us
9	9	ShipD	0849999999	ShipD@ship.us
10	10	ShipE	0841010101	ShipE@ship.us

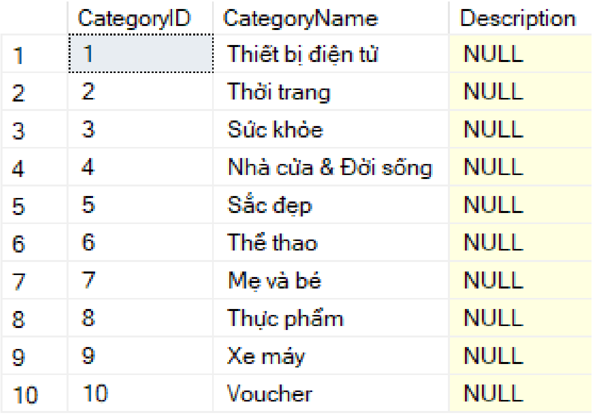
B. CREATE TABLE **CATEGORIES**

Column Name	**Data Type**	**Default**	**Check**	**Key/ Index/ Constraint**
CategoryID	Char(5)			Primary key
CategoryName	Int			Not null
Description	Char(1)			

Code:

```
--create table manager room
CREATE TABLE Categories(
	CategoryID INT NOT NULL,
	CategoryName NVARCHAR(40) NOT NULL,
	Description NVARCHAR(255),
	PRIMARY KEY(CategoryID)
)
```

Example:



	CategoryID	CategoryName	Description
1	1	Thiết bị điện tử	NULL
2	2	Thời trang	NULL
3	3	Sức khỏe	NULL
4	4	Nhà cửa & Đời sống	NULL
5	5	Sắc đẹp	NULL
6	6	Thể thao	NULL
7	7	Mẹ và bé	NULL
8	8	Thực phẩm	NULL
9	9	Xe máy	NULL
10	10	Voucher	NULL

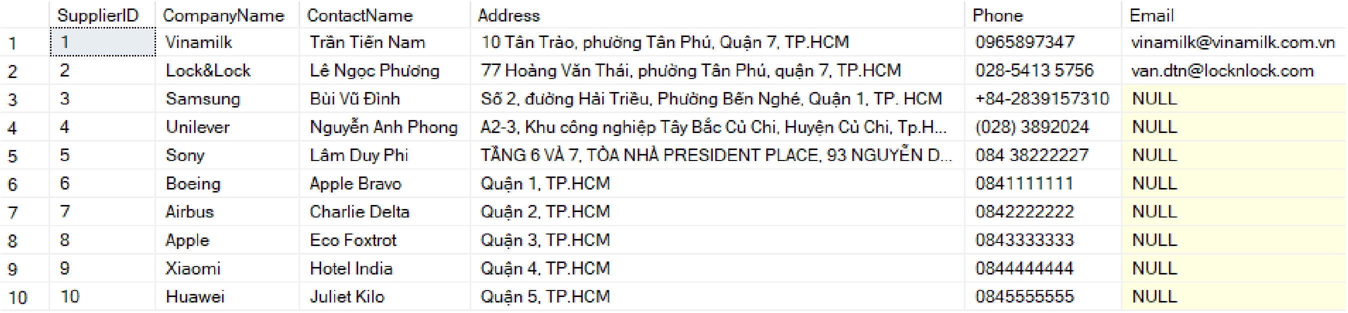
C. CREATE TABLE **SUPPLIERS**

Column Name	Data Type	Default	Check	Key/ Index/ Constraint
SupplierID	Int			Primary key
CompanyName	Nvarchar(40)			Not null
ContactName	Nvarchar(40)			Not null
Address	Nvarchar(255)			Not null
Phone	Nvarchar(24)		0 to 9	Not null
Email	Nvarchar(40)			

Code:

```
--create table Suppliers
CREATE TABLE Suppliers(
	SupplierID INT NOT NULL,
	CompanyName NVARCHAR(40) NOT NULL,
	ContactName NVARCHAR(40) NOT NULL,
	Address NVARCHAR(255) NOT NULL,
	Phone NVARCHAR(24) CHECK(Phone LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
	NOT NULL,
	Email NVARCHAR(40),
	PRIMARY KEY(SupplierID)
)
```

Example:



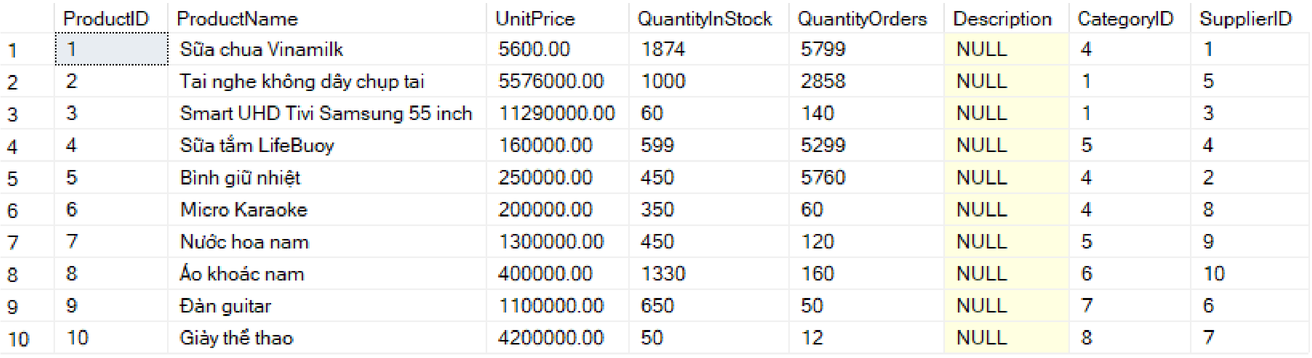
	SupplierID	CompanyName	ContactName	Address	Phone	Email
1	1	Vinamilk	Trần Tiến Nam	10 Tân Trào, phường Tân Phú, Quận 7, TP.HCM	0965897347	vinamilk@vinamilk.com.vn
2	2	Lock&Lock	Lê Ngọc Phương	77 Hoàng Văn Thái, phường Tân Phú, quận 7, TP.HCM	028-5413 5756	van.dtn@locknlock.com
3	3	Samsung	Bùi Vũ Đình	Số 2, đường Hải Triều, Phường Bến Nghé, Quận 1, TP. HCM	+84-2839157310	NULL
4	4	Unilever	Nguyễn Anh Phong	A2-3, Khu công nghiệp Tây Bắc Củ Chi, Huyện Củ Chi, Tp.H...	(028) 3892024	NULL
5	5	Sony	Lâm Duy Phi	TẦNG 6 VÀ 7, TÒA NHÀ PRESIDENT PLACE, 93 NGUYỄN D...	084 38222227	NULL
6	6	Boeing	Apple Bravo	Quận 1, TP.HCM	0841111111	NULL
7	7	Airbus	Charlie Delta	Quận 2, TP.HCM	0842222222	NULL
8	8	Apple	Eco Foxtrot	Quận 3, TP.HCM	0843333333	NULL
9	9	Xiaomi	Hotel India	Quận 4, TP.HCM	0844444444	NULL
10	10	Huawei	Juliet Kilo	Quận 5, TP.HCM	0845555555	NULL

D. CREATE TABLE **PRODUCTS**

Column Name	Data Type	Default	Check	Key/ Index/ Constraint
ProductID	Int			Primary key
ProductName	Nvarchar(255)			Not null
UnitPrice	Money			Not null
QuantityInStock	Smallint			Not null
QuantityOrders	Smallint			Not null
Description	Nvarchar(255)			
CategoryID	Int			Not null, FK references Categories(CategoryID)
SupplierID	Int			Not null,FK references Suppliers(SupplierID)

Code:

```
--create table Products
CREATE TABLE Products(
	ProductID INT NOT NULL,
	ProductName NVARCHAR(255) NOT NULL,
	UnitPrice MONEY NOT NULL,
	QuantityInStock SMALLINT NOT NULL,
	QuantityOrders SMALLINT NOT NULL,
	Description NVARCHAR(255),
	CategoryID INT REFERENCES Categories(CategoryID) NOT NULL,
	SupplierID INT REFERENCES Suppliers(SupplierID) NOT NULL,
	PRIMARY KEY(ProductID)
)
```



	ProductID	ProductName	UnitPrice	QuantityInStock	QuantityOrders	Description	CategoryID	SupplierID
1	1	Sữa chua Vinamilk	5600.00	1874	5799	NULL	4	1
2	2	Tai nghe không dây chụp tai	5576000.00	1000	2858	NULL	1	5
3	3	Smart UHD Tivi Samsung 55 inch	11290000.00	60	140	NULL	1	3
4	4	Sữa tắm LifeBuoy	160000.00	599	5299	NULL	5	4
5	5	Bình giữ nhiệt	250000.00	450	5760	NULL	4	2
6	6	Micro Karaoke	200000.00	350	60	NULL	4	8
7	7	Nước hoa nam	1300000.00	450	120	NULL	5	9
8	8	Áo khoác nam	400000.00	1330	160	NULL	6	10
9	9	Đàn guitar	1100000.00	650	50	NULL	7	6
10	10	Giày thể thao	4200000.00	50	12	NULL	8	7

Example:

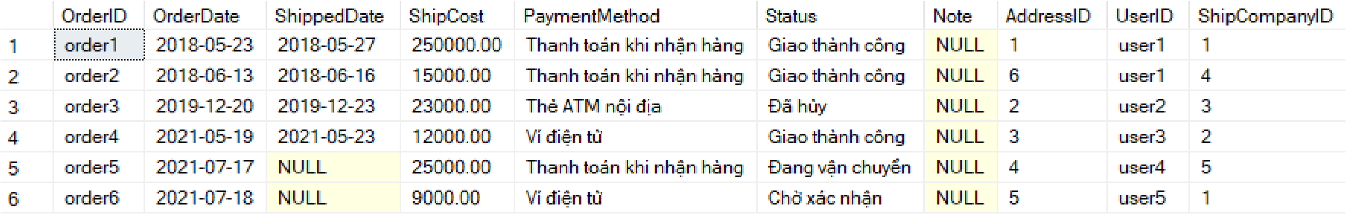
E. CREATE TABLE **ORDERS**

Column Name	Data Type	Default	Check	Key/ Index/ Constraint
OrderID	Nchar(20)			Primary key
OrderDate	Date		<= Getdate()	Not null
ShippedDate	Date			
ShipCost	Money			Not null
PaymentMethod	Nvarchar(40)			Not null
Status	Nvarchar(40)			Not null
Note	Nvarchar(255)			
AddressID	Int			Not null
UserID	Nchar(20)			Not null, FK references Users(UserID)
ShipCompanyID	Int			Not null, FK references ShipCompanies(ShipCompanyID)

Code:

```
--create table Orders
CREATE TABLE Orders(
	OrderID NCHAR(20) NOT NULL,
	OrderDate DATE check(OrderDate <= getDate()) NOT NULL,
	ShippedDate DATE ,
	ShipCost MONEY NOT NULL,
	PaymentMethod NVARCHAR(40) NOT NULL,
	Status NVARCHAR(40) NOT NULL,
	Note NVARCHAR(255),
	AddressID INT REFERENCES Address(AddressID) NOT NULL,
	UserID NCHAR(20) REFERENCES Users(UserID) NOT NULL,
	ShipCompanyID INT REFERENCES Shipcompanies(ShipCompanyID) NOT NULL,
	PRIMARY KEY(OrderID)
 )
```

Example:



	OrderID	OrderDate	ShippedDate	ShipCost	PaymentMethod	Status	Note	AddressID	UserID	ShipCompanyID
1	order1	2018-05-23	2018-05-27	250000.00	Thanh toán khi nhận hàng	Giao thành công	NULL	1	user1	1
2	order2	2018-06-13	2018-06-16	15000.00	Thanh toán khi nhận hàng	Giao thành công	NULL	6	user1	4
3	order3	2019-12-20	2019-12-23	23000.00	Thẻ ATM nội địa	Đã hủy	NULL	2	user2	3
4	order4	2021-05-19	2021-05-23	12000.00	Ví điện tử	Giao thành công	NULL	3	user3	2
5	order5	2021-07-17	NULL	25000.00	Thanh toán khi nhận hàng	Đang vận chuyển	NULL	4	user4	5
6	order6	2021-07-18	NULL	9000.00	Ví điện tử	Chờ xác nhận	NULL	5	user5	1

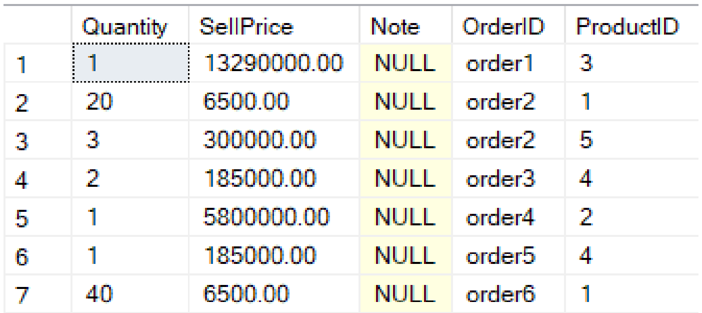
F. CREATE TABLE **ORDERITEMS**

Column Name	Data Type	Default	Check	Key/ Index/ Constraint
Quantity	Smallint			Not null
SellPrice	Money			Not null
Note	Nvarchar(255)			
OrderID	Nchar(20)			Not null, FK references Orders(OrderID)
ProductID	int			Not null, FK references Products(ProductID)

Code:

```
--create table OrderItems
CREATE TABLE OrderItems(
       Quantity SMALLINT NOT NULL,
       SellPrice MONEY NOT NULL,
       Note NVARCHAR(255),
       OrderID NCHAR(20) REFERENCES Orders(OrderID) NOT NULL,
       ProductID INT REFERENCES Products(ProductID) NOT NULL,
       PRIMARY KEY(OrderID, ProductID)
)
```

Example:



	Quantity	SellPrice	Note	OrderID	ProductID
1	1	13290000.00	NULL	order1	3
2	20	6500.00	NULL	order2	1
3	3	300000.00	NULL	order2	5
4	2	185000.00	NULL	order3	4
5	1	5800000.00	NULL	order4	2
6	1	185000.00	NULL	order5	4
7	40	6500.00	NULL	order6	1

G. CREATE TABLE **CART**

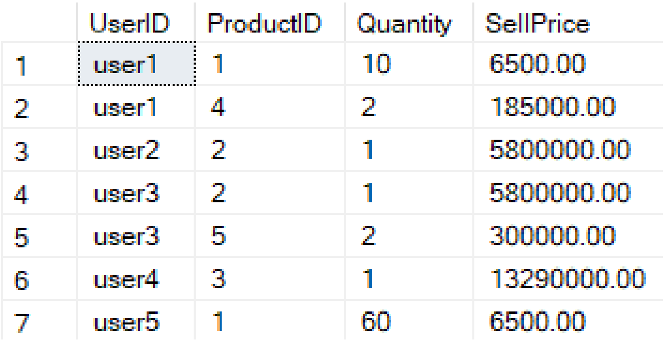
Column Name	Data Type	Default	Check	Key/ Index/ Constraint
UserID	Nchar(20)			FK reference Users(UserID) Primary key
ProductID	Int			FK reference Products(ProductID), Not null
SellPrice	Money			Not null
Quantity	Smallint	1		Not null

Code:

```
--create table student Cart

CREATE TABLE Cart(
	UserID NCHAR(20) NOT NULL,
	ProductID INT REFERENCES Products(ProductID) NOT NULL,
	Quantity SMALLINT DEFAULT 1 NOT NULL,
	SellPrice MONEY NOT NULL,
	PRIMARY KEY(UserID, ProductID)
)
```

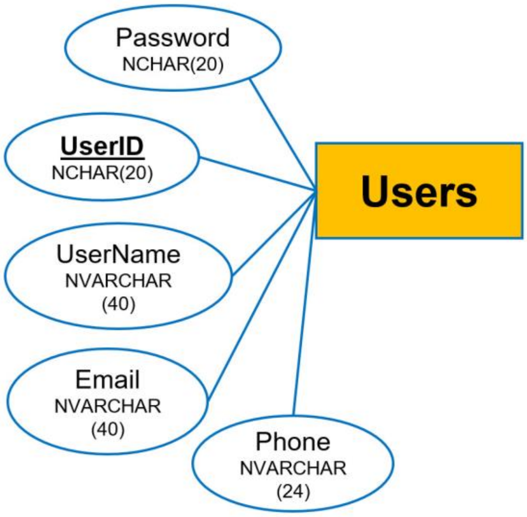
Example:



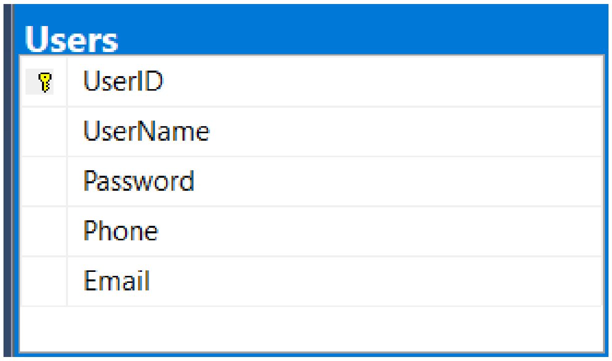
	UserID	ProductID	Quantity	SellPrice
1	user1	1	10	6500.00
2	user1	4	2	185000.00
3	user2	2	1	5800000.00
4	user3	2	1	5800000.00
5	user3	5	2	300000.00
6	user4	3	1	13290000.00
7	user5	1	60	6500.00

IV. ENTITY RELATIONSHIP DIAGRAM (ERD)

A. USERS



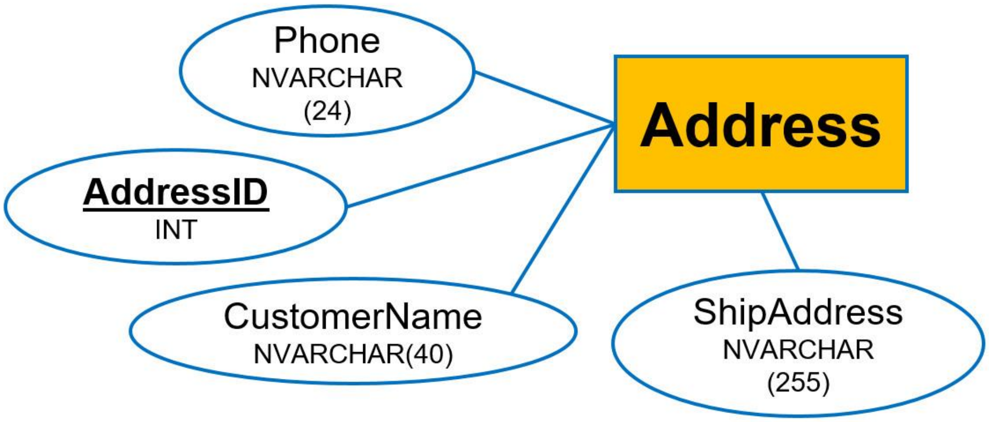
This is the Users entity. Tournament entity has 5 attributes.
The attribute UserID also is the primary key of this entity. Each User has account and contact. Account is UserName and Password. User's contact is Email and Phone.



Users

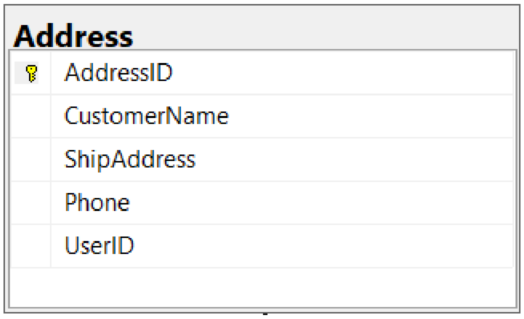
🔑	UserID
	UserName
	Password
	Phone
	Email

B. ADDRESS



This is the Address entity. This has 5 attributes.

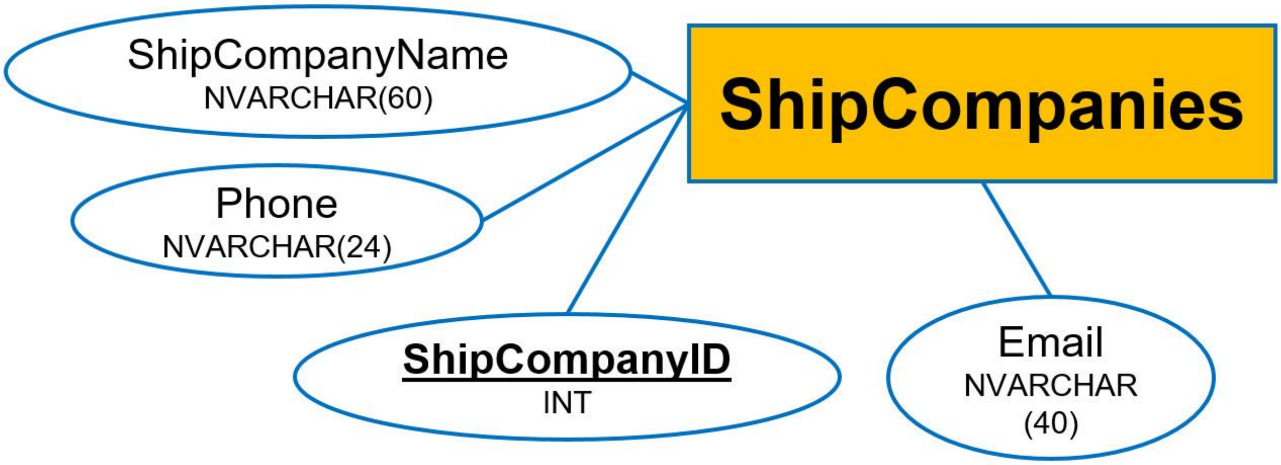
Each address have AddressID is Primary key. Information has CustomerName, ShipAddress and Phone. Each Address is owned by User so this have UserID is foreign key



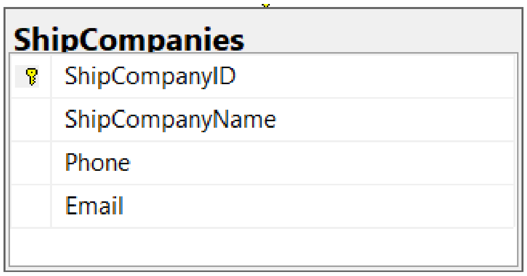
Address

- AddressID
- CustomerName
- ShipAddress
- Phone
- UserID

C. SHIPCOMPANIES



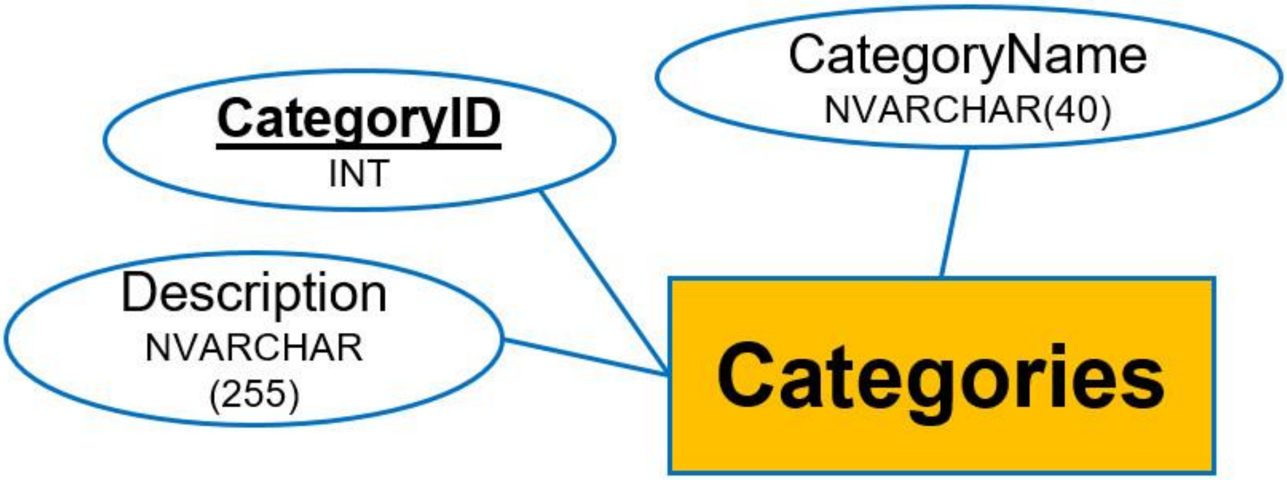
Order need to be delivery to Customer, so we have ShipCompanies to do this. ShipCompanyID is primary key to insert to database, ShipCompanyName is name of delivery company and contact include Phone and Email

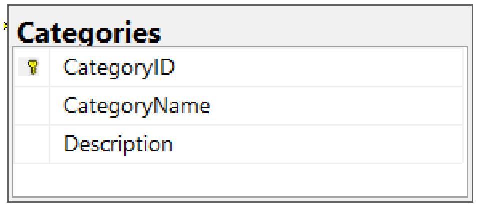


ShipCompanies

- ShipCompanyID
- ShipCompanyName
- Phone
- Email

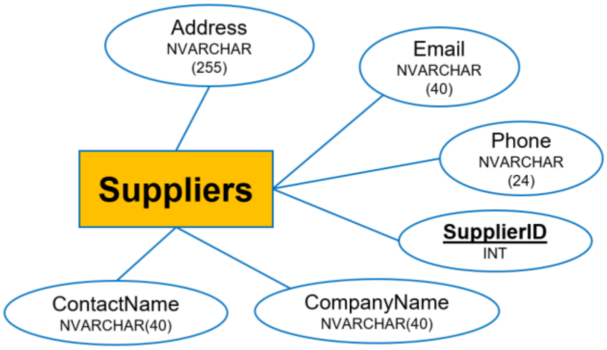
D. CATEGORIES

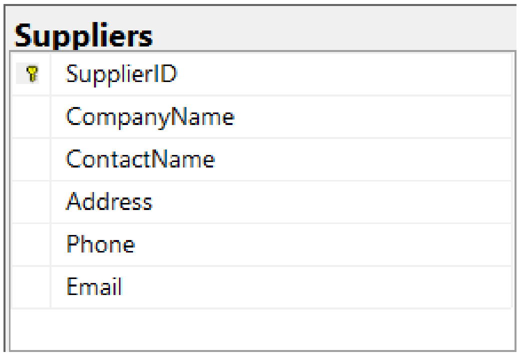



Categories
- CategoryID
- CategoryName
- Description

Category entity have 3 attributes.
Category ID is primary key of the entity.
CategoryName to know the name of category. And description to describe detail of the category.

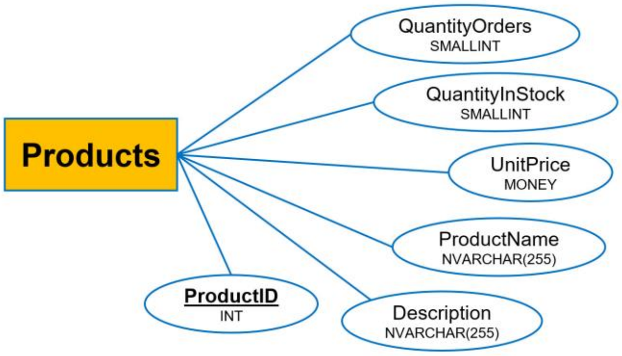
E. SUPPLIERS

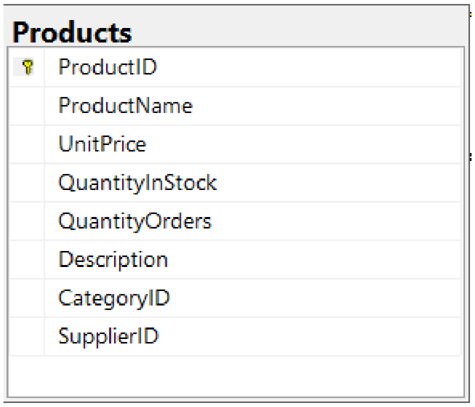



Suppliers
- SupplierID
- CompanyName
- ContactName
- Address
- Phone
- Email

All product have supplier who provide this. Each Supplier have SupplierID. Information is CompanyName and contact include ContactName, Address, Phone and Email.

A. PRODUCTS





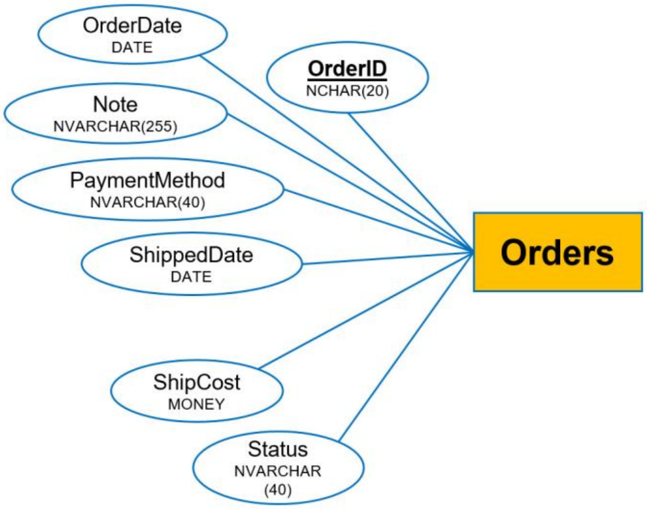
Products

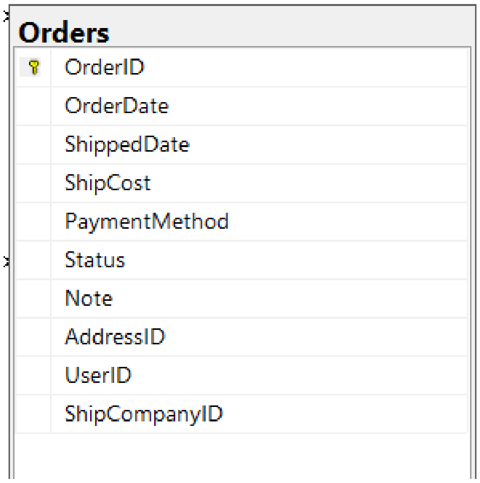
- ProductID
- ProductName
- UnitPrice
- QuantityInStock
- QuantityOrders
- Description
- CategoryID
- SupplierID

This is product entity. ProductID is the primary key. Information about product include UnitPrice – price of a unit, QuantityInStock, QuantityOrders and Description describe detail about product.

Each product belong to a category and be provided by a supplier so that we have foreign key are CategoryID and SupplierID

A. ORDERS





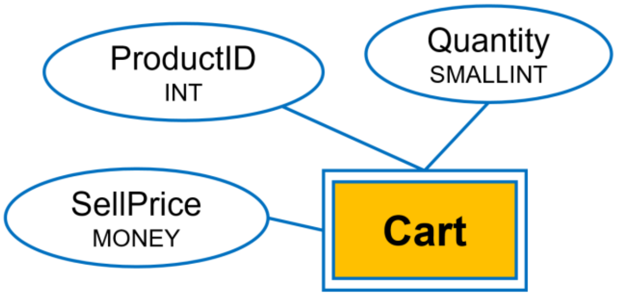
Orders

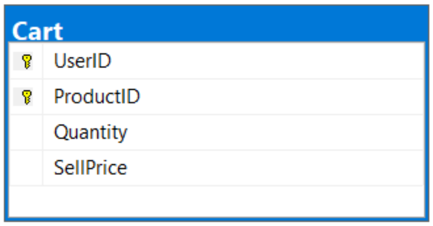
- OrderID
- OrderDate
- ShippedDate
- ShipCost
- PaymentMethod
- Status
- Note
- AddressID
- UserID
- ShipCompanyID

Sale product are group into order.

Order entity has 7 attribute . OrderID is primary key of this entity. Information about order include OrderDate is the date on which this order were created, Shippeddate is date customer have his/her order, ShipCost is delivery fee, PaymentMethod to indicate which method customer use to pay: cash or banking, Status and Note – which customer want to emphasize

A. CART



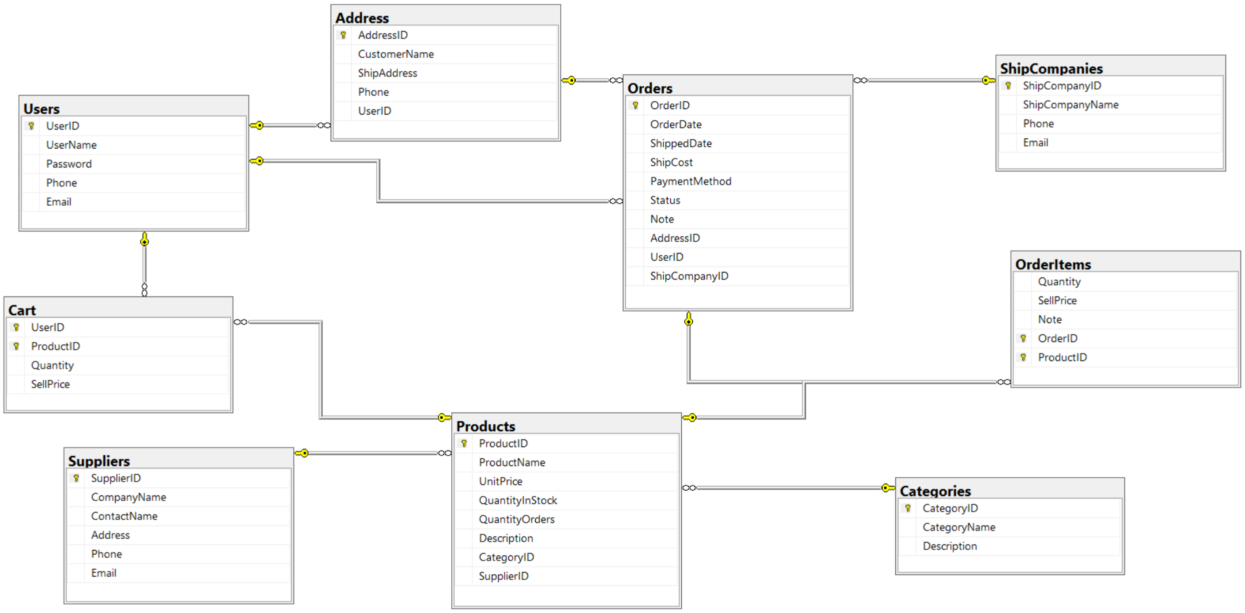


Cart

- UserID
- ProductID
- Quantity
- SellPrice

Each user has a cart contains product they have choose but not buy.
This Cart entity is a weak entity depend on User.
Cart include 2 attributes are ProductID and Quantity.

B. FULL DIAGRAM



Address: AddressID, CustomerName, ShipAddress, Phone, UserID

Users: UserID, UserName, Password, Phone, Email

Orders: OrderID, OrderDate, ShippedDate, ShipCost, PaymentMethod, Status, Note, AddressID, UserID, ShipCompanyID

ShipCompanies: ShipCompanyID, ShipCompanyName, Phone, Email

OrderItems: Quantity, SellPrice, Note, OrderID, ProductID

Cart: UserID, ProductID, Quantity, SellPrice

Products: ProductID, ProductName, UnitPrice, QuantityInStock, QuantityOrders, Description, CategoryID, SupplierID

Suppliers: SupplierID, CompanyName, ContactName, Address, Phone, Email

Categories: CategoryID, CategoryName, Description

V. SQL COMMAND

We’re using Microsoft SQL Server 2018, this server build intelligent, mission-critical applications using a scalable, hybrid database platform that has everything built in—from in-memory performance and advanced security to in-database analytics.

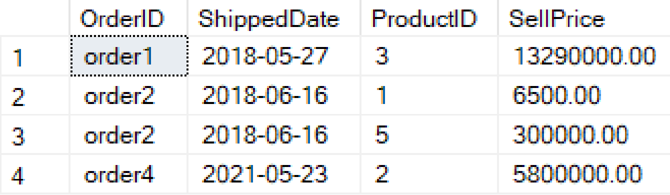
A. QUERY USING INNER JOIN

Code:

```
SELECT Orders.OrderID, Orders.ShippedDate, OrderItems.ProductID, OrderItems.SellPrice
FROM Orders INNER JOIN OrderItems ON Orders.OrderID = OrderItems.OrderID
AND MONTH(ShippedDate) IN (05,06)
```

Result:

Using query INNER JOIN to display OrderID, ShippedDate ProductID, SellPrice of each order in May and June



	OrderID	ShippedDate	ProductID	SellPrice
1	order1	2018-05-27	3	13290000.00
2	order2	2018-06-16	1	6500.00
3	order2	2018-06-16	5	300000.00
4	order4	2021-05-23	2	5800000.00

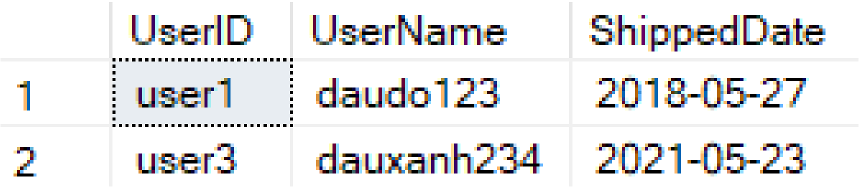
B. QUERY USING OUTER JOIN

Code:

```
SELECT Users.UserID, UserName, ShippedDate
FROM Users LEFT OUTER JOIN Orders ON Users.UserID = Orders.UserID
AND MONTH(ShippedDate)=05
WHERE OrderID IS NOT NULL
```

Result:

Using query OUTER JOIN to display UserID, UserName, ShippedDate in May



	UserID	UserName	ShippedDate
1	user1	daudo123	2018-05-27
2	user3	dauxanh234	2021-05-23

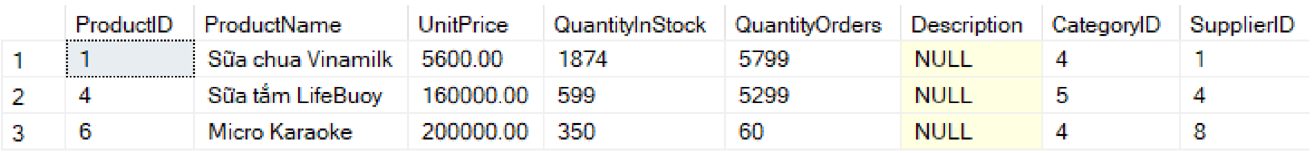
C. QUERY USING SUBQUERY ‘WHERE’

Code:

```
SELECT*FROM Products
WHERE UnitPrice <
(
       SELECT UnitPrice
       FROM Products
       WHERE ProductID=5
 )
```

Result:

Using subquery WHERE to display all products that have UnitPrice cheaper than UnitPrice of ProductID=5



	ProductID	ProductName	UnitPrice	QuantityInStock	QuantityOrders	Description	CategoryID	SupplierID
1	1	Sữa chua Vinamilk	5600.00	1874	5799	NULL	4	1
2	4	Sữa tắm LifeBuoy	160000.00	599	5299	NULL	5	4
3	6	Micro Karaoke	200000.00	350	60	NULL	4	8

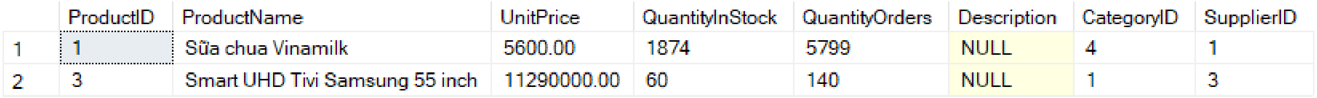
D. QUERY USING SUBQUERY 'FROM'

Code:

```
WITH t AS
(
        SELECT TOP 1*
        FROM Products
        ORDER BY UnitPrice DESC
),
s AS
(
        SELECT TOP 1*
        FROM Products
        ORDER BY UnitPrice ASC
)
SELECT*FROM t
UNION
 SELECT*FROM s
```

Result:

Using subquery FROM to display a product that have highgest UnitPrice and a product that have cheapest UnitPrice



	ProductID	ProductName	UnitPrice	QuantityInStock	QuantityOrders	Description	CategoryID	SupplierID
1	1	Sữa chua Vinamilk	5600.00	1874	5799	NULL	4	1
2	3	Smart UHD Tivi Samsung 55 inch	11290000.00	60	140	NULL	1	3

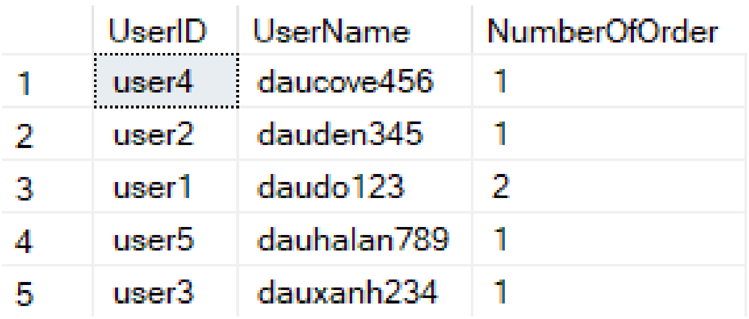
E. QUERY USES GROUP BY AND AGGREGATE FUNCTIONS

Code:

```
SELECT Users.UserID, Users.UserName, COUNT(Orders.OrderID) AS 'NumberOfOrder'
FROM Orders JOIN Users ON Orders.UserID = Users.UserID
 GROUP BY Users.UserID, Users.UserName
```

Result:

Using query GROUP and AGGREGATE FUNTIONS to display UserID, UserName, NumberOfOrder that is the number of order that the user have purchase



	UserID	UserName	NumberOfOrder
1	user4	daucove456	1
2	user2	dauden345	1
3	user1	daudo123	2
4	user5	dauhalan789	1
5	user3	dauxanh234	1

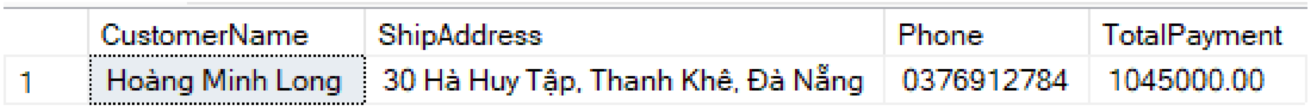
F. STORE PROCEDURED

Code:

```
CREATE PROC CalculateAmount @OrderID NCHAR(20)
AS
BEGIN
	SELECT CustomerName, ShipAddress, Address.Phone, SUM(SellPrice*Quantity) + ShipCost AS 
TotalPayment
	FROM Orders JOIN OrderItems ON OrderItems.OrderID = Orders.OrderID
		JOIN Address ON Address.AddressID  = Orders.AddressID
	WHERE Orders.OrderID = @OrderID
	GROUP BY CustomerName, ShipAddress, Address.Phone, ShipCost
END

EXEC CalculateAmount @OrderID = 'order2'
```

Result:



	CustomerName	ShipAddress	Phone	TotalPayment
1	Hoàng Minh Long	30 Hà Huy Tập, Thanh Khê, Đà Nẵng	0376912784	1045000.00

Enter orderID to calculate the total amount of this order and print out information about customer include CustomerName, ShipAddress and Phone.

G. TRIGGER

Code:

```
CREATE TRIGGER checkPrice ON OrderItems
INSTEAD OF INSERT
AS
BEGIN
	IF NOT EXISTS(
		SELECT * FROM inserted JOIN Products
		ON inserted.ProductID = Products.ProductID AND SellPrice < UnitPrice
	)
	BEGIN
	INSERT INTO OrderItems
	SELECT * FROM inserted
	END
 END
```

This Trigger to ensure that sellprice of a products is always greater than price that we buy into.

THE END