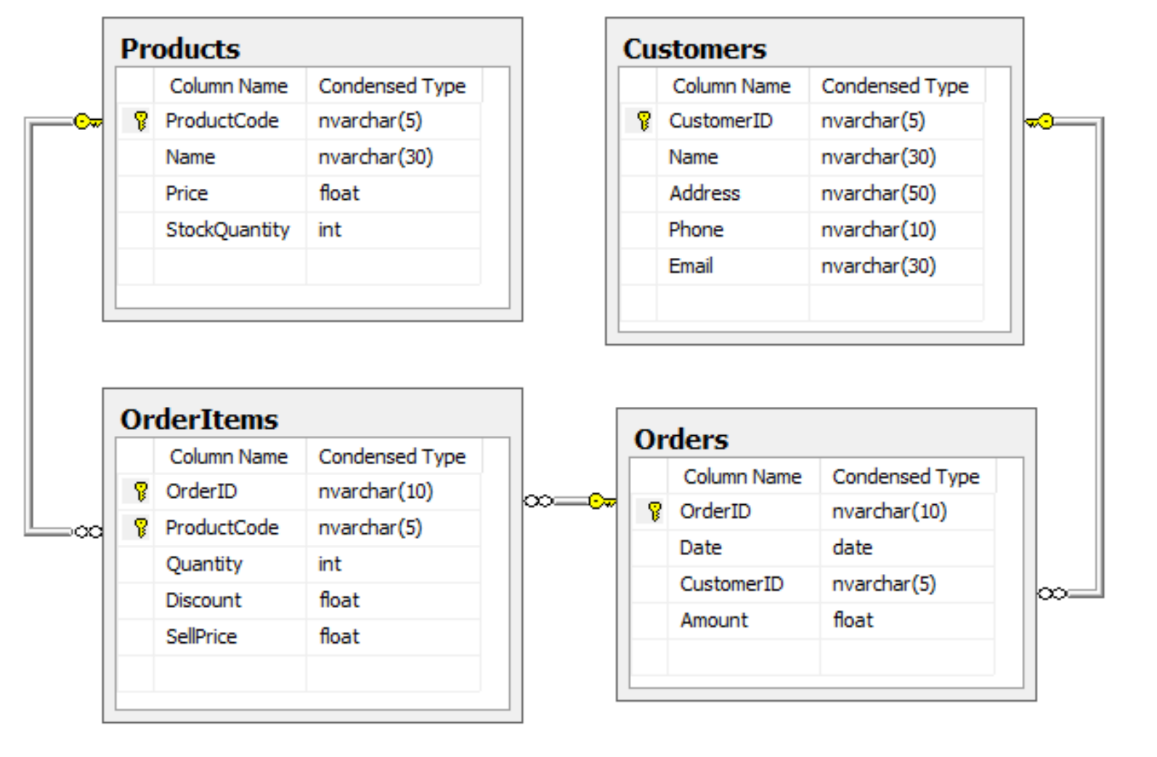
**LAB4-SQL**

**Using database “ProductOrders” as following:**



**I. Simple SQL queries:**

In the database “ProductOrders” created in the previous exercise, write the following queries:

1. Find all Products in the database
2. Find the name and the phone number of all customers who live in ‘BINH CHANH’
3. Show the Name, Price, StockQuantity and Amount (which is equal to StockQuantity \* Price) of all Products having Amount < 20000.
4. Find all customers who lives in ‘BINH CHANH’ or ‘TAN BINH’
5. Find all Products having 10<= Price <= 45
6. Find all customers who have a phone number.
7. Find months and years in which customer ‘C01’ made at least an order.
8. Find all customers whose name begin with ‘TRAN ‘.
9. Find all LCD monitor products.
10. Find the CustomerID of all customer who made orders in May 2000.
11. List all Products by descending order of Price.
12. List all customer by descending order of Address, with customers having same address we order by ascending order of Name.
13. Find the product with the highest Price.

**II. Queries involving more than one relations**

In the database “ProductOrders” created in the previous exercise, write the following queries:

1. Find OrderID, Date, CustomerID, CustomerName, Address, Phone, Email of all orders.
2. List all orders of all customers including CustomerID, CustomerName, OrderID, Date. Order the result by descending order of CustomerName (Note that we need to know also the customer who have no order). Try to use left outer join for this question.
3. Same as Question 2 but using right outer join for this exercise.
4. List CustomerID of all customers who have never bought any product.
5. Do the intersect between the list of CutomerID from Customers and the list of CustomerID from Orders? How many rows in the output?
6. Do the union between the list of CutomerID from Customers and the list of CustomerID from Orders? How many rows in the output?
7. Do the union all between the list of CutomerID from Customers and the list of CustomerID from Orders? How many rows in the output?
8. List all orders of ‘NGUYEN THI BE’ and ‘LE VAN SANG’, including CustomerID, CustomerName, OrderID, Date; ordered by CustomerName.
9. List OrderID, ProductCode, ProductName, Price, SellPrice, SoldQuantity, Profit ((SellPrice - Price)\*SoldQuantity) by ascending order of OrderID.
10. List CustomerID, CustomerName, OrderID, ProductCode, ProductName, Price, SellPrice, SoldQuantity, Profit ((SellPrice - Price)\*SoldQuantity) by ascending order of CustomerName.
11. List 5 OrderItems having the highest profit (profit = (SellPrice - Price)\*SoldQuantity).
12. List all products (ProductCode, ProductName) bought by both ‘NGUYEN THI BE’ and ‘MAI THI QUE ANH’.
13. List all products with highest price (Note that there could have many products with the highest price). You should not use the aggregate function max().

**III. SubQueries in SQL:**

In the database “ProductOrders” created in the previous exercise, write the following queries using subqueries:

1. List all products having Price smaller than the price of product ‘P05’
2. List all customers who have made an order by using EXISTS in WHERE clause.
3. List all customers who have never made an order by using NOT EXISTS in WHERE clause
4. List all customers who have made an order by using IN in WHERE clause.
5. List all customers who have never made an order by using NOT IN in WHERE clause
6. List all products bought in order ‘Or002’
7. List all products having Price greater than all the prices of ‘Keyboard’ and ‘Mouse’
8. List all products having price greater than the prices ‘Keyboard’ or ‘Mouse’
9. List all products which have been bought by customer ‘NGUYEN THI BE’ (Nguyễn Thị Bé).
10. List all products having price greater than the average price of all products in Products table.