**Students**

**Classes**

**Teach**

**Teachers**

**Attend**

1. **1.5 point**

Create one database and then write SQL statements to create all tables derived from the given ERD. Those tables have to locate under the database that you have just created with appropriate attributes, primary keys and foreign keys.

**NOTICE that when creating the SQL commands as request, you MUST keep the name of tables, relationship and attributes and data type of attributes as SAME as given in the above ERD.**

**Attributes have written with underline are Primary Key of each entity.**

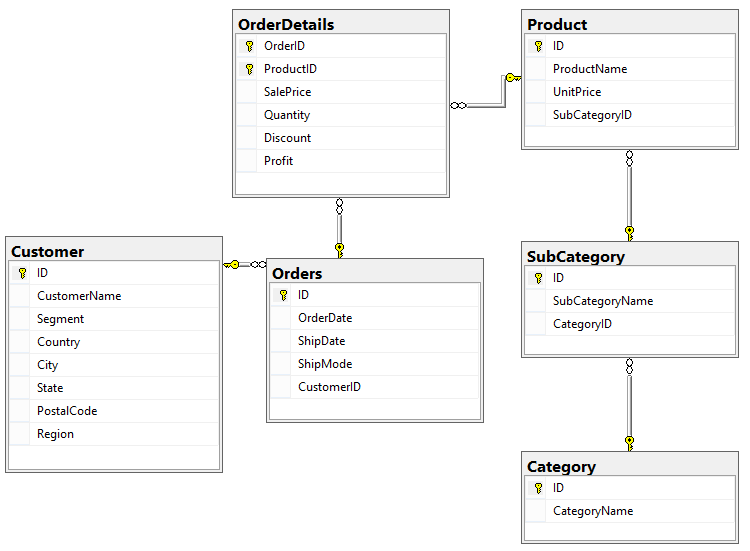
**When submitting the responses for this question, submit only SQL statements for creating tables with corresponding keys and foreign keys. Do not use “create database” or “use database\_name” statements in your submission.**

1. **1 point:**

Write SQL statements to insert information about:

* Teacher that his (TeacherID, Name, Address, Gender) are (1, Bui Chien, Cau Giay – Ha Noi, M).
* Student that his (StudentID, Name, Address, Gender) are (1, Nguyen Hang, Cau Giay – Ha Noi, F).
* Class whose (ClassID, GroupID, CourseID, year, Semester, NoCredits) are (1,SE1316,DBI202,2019 , S, 3) and is taught by teacher having TeacherId 1.
* Student having StudentId 1 has attended the class 1 in slot 6, 15/03/2019. Note that attribute Attend should be ‘1’ when the student is attended.

**From question 3, you use the given database has name PEDemo, that base on the ERD as bellow:**



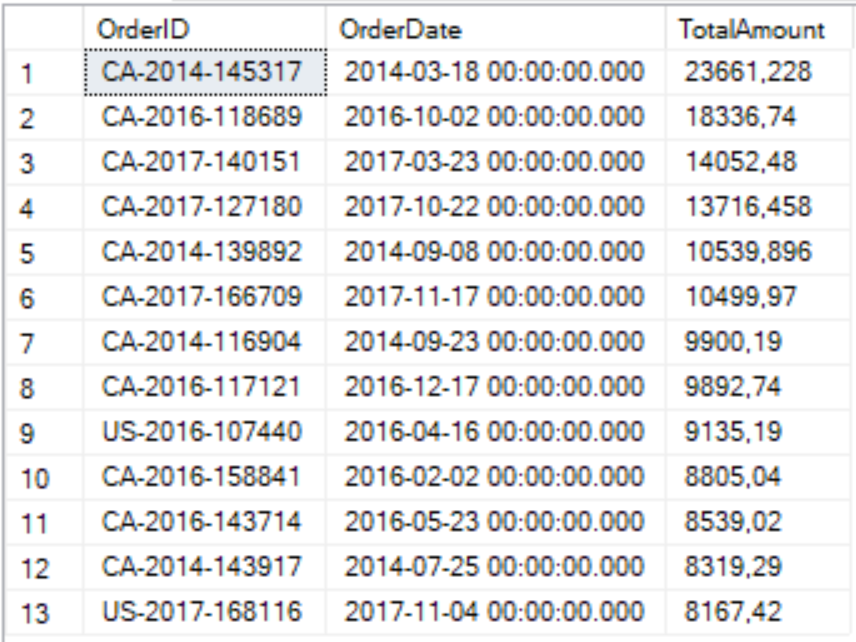
1. **1 point.** Write a query to display all SubCategory in category 3 as follows



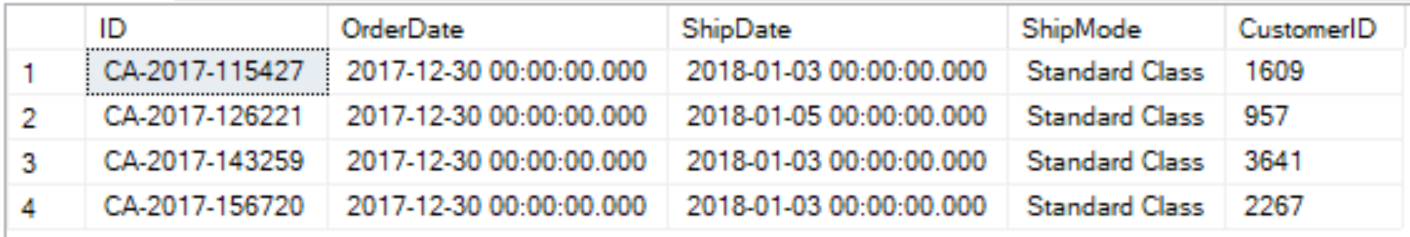
1. Write a query to display the ID, CustomerName, City, State of all customers who placed orders from 05 December 2017 to 10 December 2017 and their orders are shipped less than 3 days. Display the result by ascending order of State and then by descending order of City.



1. **1 point.** The Amount of each product in an order is calculated by Quantity\*SalePrice\*(1-Discount). The TotalAmount of each order is the sum of the Amount of all product in the order. Write a query to display OrderID, OrderDate, TotalAmount of all orders having TotalAmount greater than 8000, by descending order of TotalAmount as follows:

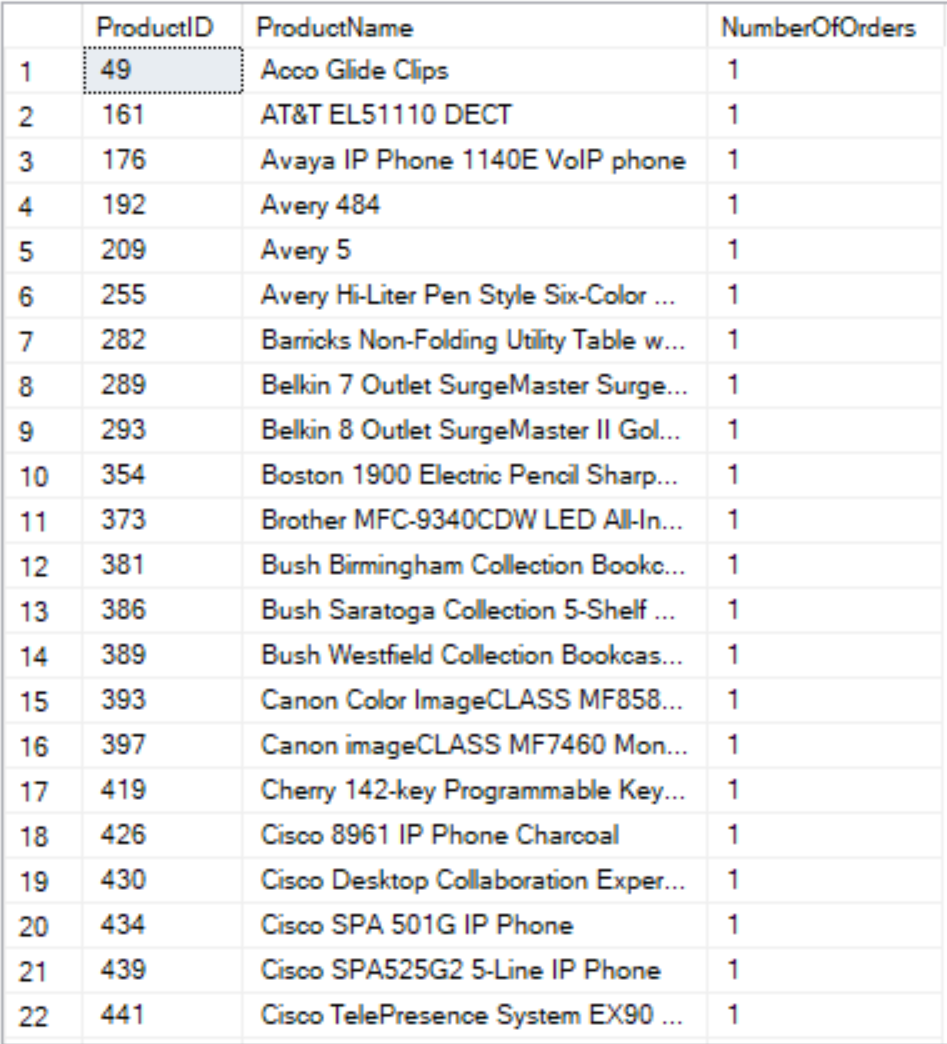


1. Find all orders that were ordered on the same day as the latest order as follows:

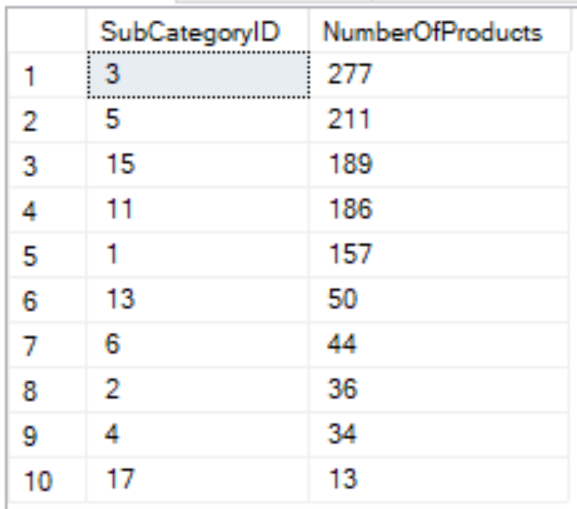


1. Find all the products which appeared in the smallest number of orders, order by ProductID. The results should be displayed in the form of ProductID, ProductName, NumberOfOrders.

Note that the following figure show only 22 first rows of the results. In fact, the query should return 90 rows.



1. Write a query to display 5 sub-categories having the highest numbers of different products and the 5 sub-categories having the smallest numbers of different products, by descending order of number of different products as follows:



1. **0.5 point.** Write a stored procedure named TotalAmount to calculate the total amount of an order with OrderID (nvarchar(255)) is its input parameter and the TotalAmount (float) is the output parameter. Note that the Amount of each product in an order is calculate as SalePrice\*Quantity\*(1-Discount) and the TotalAmount of each order is the sum of all the Amounts of all products in the order.
2. **0.5 point.** Create a trigger InsertSubCategory which will be activate by an insert statement into the SubCategory table. The trigger will display the SubCategoryName and the CategoryName of the sub-categories which have just been inserted by the insert statement.

The result should be:

