# A. Introduction

Your task is to write SQLite queries based on the given Northwind database, nws.db. The nsw.db database consists of 11 tables: Category, Customer, Employee,

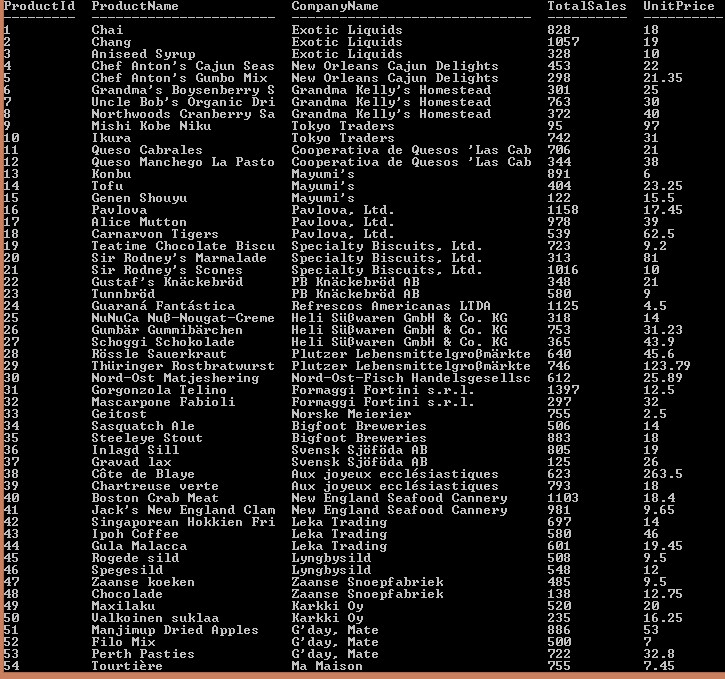
EmployeeTerritory, Order, OrderDetail, Product, Region, Shipper, Supplier, and Territory. For each of the tables, you can view the schema of the tables with the .schema command (e.g, .schema Category).

# B. Preparation

1. Download the modified Northwind SQLite database (nws.db), the Northwind database entity relationship diagram (Northwind\_ERD.png), and the template SQL file (cwk2.sql) to write your SQL codes. Put these files in a folder/directory.
2. Open a terminal window in the directory and open the database with the command sqlite3 nws.db. Make sure you use the more recent version of SQLite3 (download from https://www.sqlite.org/index.html).

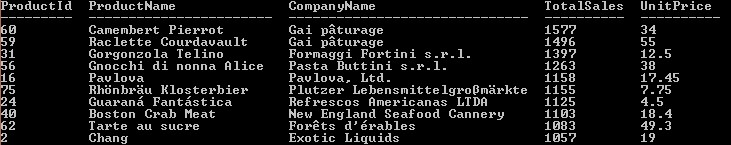
Question 3(a) Write an SQLite query to create a view called vSales for total products sold as well as the product and supplier information. Use JOIN construct(s) to retrieve the total products sold based on quantity (named TotalSales), product id, product name, company selling the product, and unit prices (from product table) as shown by sample output.

Example output of the view after the query:



Question 3(b) Using the view vSales, write an SQLite query to create a view called vTop10Products for top 10 products based on TotalSales in descending order.

Example output of the view after the query:



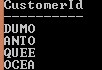
Question 4(a) Write an SQLite query to create a view called vLoyalCustomer for the details of total number of orders for all customer IN the Customer table. The details to be retrieved are TotalOrder, CustomerId, CompanyName and Region. Note that there are orders with customer NOT IN the Customer table.

Example output of the view after the query:



Question 4(b) Write an SQLite query to create a view called vCustomerMissingDetails with customers (based on CustomerId) in the Order table but not in the Customer table.

Example output of the view after the query:



Question 4(c) (Advanced) Using the view vLoyalCustomer, write an SQLite query to create a view called vTopRegionCustomer for the customer with the highest total order in each region. Hint: you can use SQLite RANK() function with PARTITION and ORDER BY clause.

This question is a more challenging in which you can use more advanced SQL functions such as RANK and PARTITION. You can refer to the SQLite documentation of these functions at https://sqlite.org/windowfunctions.html#the\_partition\_by\_clause.

Example output of the view after the query:

