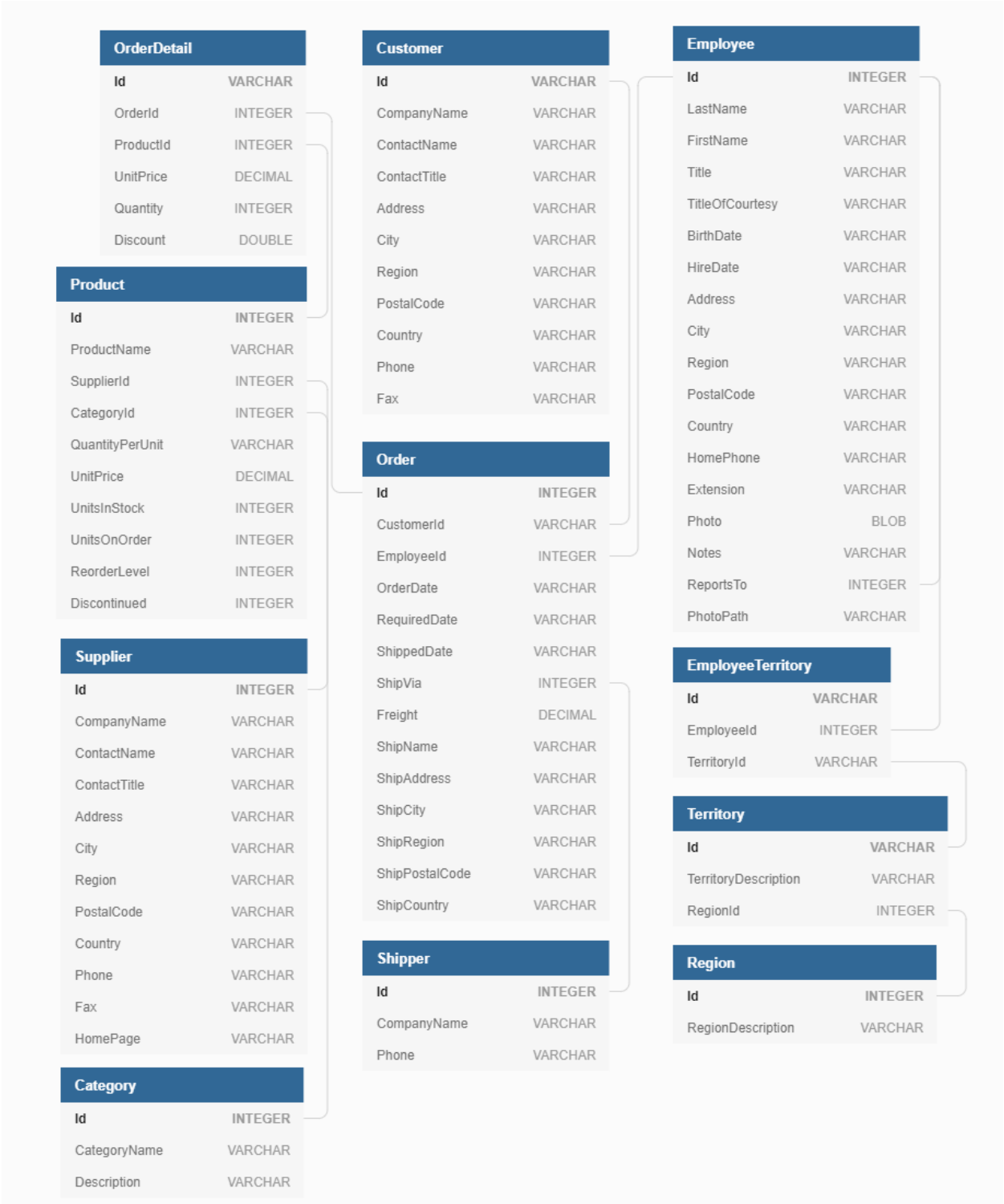

OrderDetail
Id VARCHAR
OrderId INTEGER
ProductId INTEGER
UnitPrice DECIMAL
Quantity INTEGER
Discount DOUBLE
Customer
Id VARCHAR
CompanyName VARCHAR
ContactName VARCHAR
ContactTitle VARCHAR
Address VARCHAR
City VARCHAR
Region VARCHAR
PostalCode VARCHAR
Country VARCHAR
Phone VARCHAR
Fax VARCHAR
Employee
Id INTEGER
LastName VARCHAR
FirstName VARCHAR
Title VARCHAR
TitleOfCourtesy VARCHAR
BirthDate VARCHAR
HireDate VARCHAR
Address VARCHAR
City VARCHAR
Region VARCHAR
PostalCode VARCHAR
Country VARCHAR
HomePhone VARCHAR
Extension VARCHAR
Photo BLOB
Notes VARCHAR
ReportsTo INTEGER
PhotoPath VARCHAR
Product
Id INTEGER
ProductName VARCHAR
SupplierId INTEGER
CategoryId INTEGER
QuantityPerUnit VARCHAR
UnitPrice DECIMAL
UnitsInStock INTEGER
UnitsOnOrder INTEGER
ReorderLevel INTEGER
Discontinued INTEGER
Order
Id INTEGER
CustomerId VARCHAR
EmployeeId INTEGER
OrderDate VARCHAR
RequiredDate VARCHAR
ShippedDate VARCHAR
ShipVia INTEGER
Freight DECIMAL
ShipName VARCHAR
ShipAddress VARCHAR
ShipCity VARCHAR
ShipRegion VARCHAR
ShipPostalCode VARCHAR
ShipCountry VARCHAR
EmployeeTerritory
Id VARCHAR
EmployeeId INTEGER
TerritoryId VARCHAR
Supplier
Id INTEGER
CompanyName VARCHAR
ContactName VARCHAR
ContactTitle VARCHAR
Address VARCHAR
City VARCHAR
Region VARCHAR
PostalCode VARCHAR
Country VARCHAR
Phone VARCHAR
Fax VARCHAR
HomePage VARCHAR
Territory
Id VARCHAR
TerritoryDescription VARCHAR
RegionId INTEGER
Shipper
Id INTEGER
CompanyName VARCHAR
Phone VARCHAR
Region
Id INTEGER
RegionDescription VARCHAR
Category
Id INTEGER
CategoryName VARCHAR
Description VARCHAR

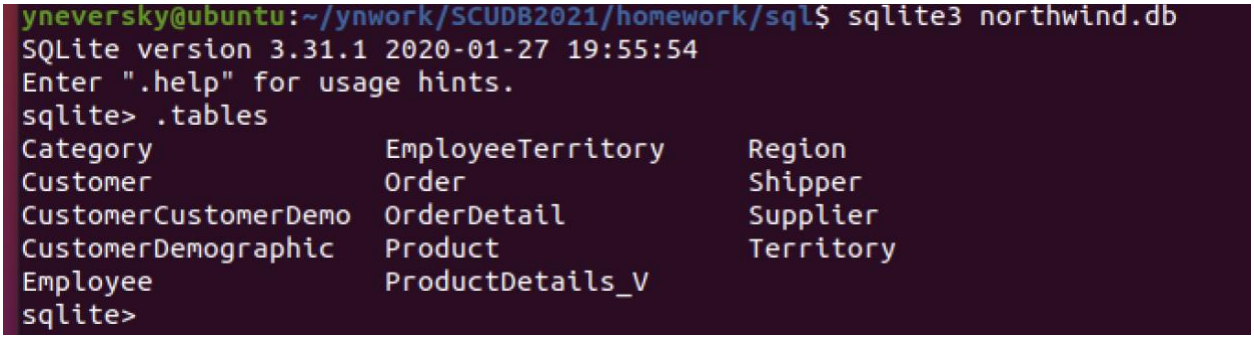
PLACEHOLDER FOLDER

Create the placeholder submission folder with the empty SQL files that you will use for each question:

```
$ mkdir placeholder
$ cd placeholder
$ touch q1_sample.sql\
  q2_string_function.sql\
  q3_northamerican.sql\
  q4_delaypercent.sql\
  q5_aggregates.sql\
  q6_discontinued.sql\
  q7_order_lags.sql\
  q8_total_cost_quartiles.sql\
  q9_youngblood.sql\
  q10_christmas.sql
$ cd ..
```

After filling in the queries, you can compress the folder by running the following command:

```
$ zip -j submission.zip placeholder/*.sql
```



Q1 [5 POINTS] (Q1_STRING_FUNCTION):

Get all unique ShipNames from the Order table that contain a hyphen '-'.
Details: In addition, get all the characters preceding the (first) hyphen. Return ship names alphabetically. Your first row should look like Bottom-Dollar Markets|Bottom

Q2 [5 POINTS] (Q2_NORTHAMERICAN):

Indicate if an order's ShipCountry is in North America. For our purposes, this is 'USA', 'Mexico', 'Canada'

Details: You should print the Order Id, ShipCountry, and another column that is either 'NorthAmerica' or 'OtherPlace' depending on the Ship Country.

Order by the primary key (Id) ascending and return 20 rows starting from Order Id 15445 Your output should look like 15445|France|OtherPlace or 15454|Canada|NorthAmerica

Q3 [5 POINTS] (Q3_DELAYPERCENT):

For each Shipper, find the percentage of orders which are late.

Details: An order is considered late if ShippedDate > RequiredDate. Print the following format, order by descending precentage, rounded to the nearest hundredths, like United Package|23.44

Q4 [10 POINTS] (Q4_AGGREGATES):

Compute some statistics about categories of products
Details: Get the number of products, average unit price (rounded to 2 decimal places), minimum unit price, maximum unit price, and total units on order for categories containing greater than 10 products.
Order by Category Id. Your output should look like Beverages|12|37.98|4.5|263.5|60

Q5 [10 POINTS] (Q5_DISCONTINUED):

For each of the 8 discontinued products in the database, which customer made the first ever order for the product? Output the customer's CompanyName and ContactName

Details: Print the following format, order by ProductName alphabetically: Alice Mutton|Consolidated Holdings|Elizabeth Brown

Q6 [10 POINTS] (Q6_ORDER_LAGS):

For the first 10 orders by CutomerId BLONP: get the Order's Id, OrderDate, previous OrderDate, and difference between the previous and current. Return results ordered by OrderDate (ascending)

Details: The "previous" OrderDate for the first order should default to itself (lag time = 0). Use the julianday() function for date arithmetic (example).
Use lag(expr, offset, default) for grabbing previous dates.
Please round the lag time to the nearest hundredth, formatted like 17361|2012-09-19 12:13:21|2012-09-18 22:37:15|0.57

Note: For more details on window functions, see here.

Q7 [15 POINTS] (Q7_TOTAL_COST_QUARTILES):

For each Customer, get the CompanyName, CustomerId, and "total expenditures". Output the bottom quartile of Customers, as measured by total expenditures.
Details: Calculate expenditure using UnitPrice and Quantity (ignore Discount).
Compute the quartiles for each company's total expenditures using NTILE. The bottom quartile is the 1st quartile, order them by increasing expenditure.
Make sure your output is formatted as follows (round expenditure to nearest hundredths): Bon app|BONAP|4485708.49

Note: There are orders for CustomerIds that don't appear in the Customer table. You should still consider these "Customers" and output them. If the CompanyName is missing, override the NULL to 'MISSING_NAME' using IFNULL.

Q8 [15 POINTS] (Q9_YOUNGBLOOD):

Find the youngest employee serving each Region. If a Region is not served by an employee, ignore it.
Details: Print the Region Description, First Name, Last Name, and Birth Date. Order by Region Id.
Your first row should look like Eastern|Steven|Buchanan|1987-03-04

Q9 [15 POINTS] (Q9_CHRISTMAS):

Concatenate the ProductNames ordered by the Company 'Queen Cozinha' on 2014-12-25.

Details: Order the products by Id (ascending). Print a single string containing all the dup names separated by commas like Mishi Kobe Niku, NuNuCa Nuß-Nougat-Creme...
Hint: You might find Recursive CTEs useful.