Ad-hoc Analysis

You are a data analyst at an NYC-based wholesale food supplier called Osiris-Foods. As part of your companies mission to make more data-driven decisions, you are tasked with generating an ad-hoc report on your companies database. You will report on descriptive statistics using SQL and generate respective visualizations using pandas to provide your company an overview of its' past performance.
For this project, you will analyze an ERD diagram to formulate DML SQL queries within a Jupyter notebook, and provide a write-up of your findings.
Keep in mind that your write-up should include pertinent numerical details to back your findings. Any claim that you make must be supported by evidence!

Instructions

Market.db

Before beginning your project, we recommend checking out the ERD diagram below:
ERD Diagram of Market Database
Use this ERD diagram to formulate how you should form your joins across tables and to find out more about your table's columns.
If you download the `sqlite3` VSCode extension, you will also be able to open your database within your code editor to view the content and columns of your table.
Note that you can also test out your SQL queries in `sqlite3` by opening your database in your command line or terminal. Check out the guide for more information on how to interact with your database in the shell.

adhoc_report.ipynb

Within this Jupyer notebook, you will write Python code and SQL queries to answer 8 ad-hoc questions. Each question will involve you creating an appropriate DML SQL query and observing this output.
After you've verified that your query's output sufficiently answers the question, you will then create a pandas dataframe using this output and finally generate a visualization from this dataframe.
Just as in the weather-analysis project, we will not directly indicate which visualizations you should make. Instead, we ask that you use your notes and study material to discern appropriate visualizations.
Test your queries in the shell, and be sure to *iteratively* build your queries. Especially when we are first learning SQL, we rarely come up with an immediate answer. Start with your basic structure (SELECT ... FROM ...) and build up to the correct answer.

Questions:

1. How many products in our Products table cost less than 10 EUR?
2. What is the most common country of origin in our Suppliers table? Sort your output in descending order **Hint**: `GROUP BY` can be used to calculate grouped aggregates.
3. What is the most common country of origin in our Customers table? Sort your output in descending order. **Hint**: `GROUP BY` can be used to calculate grouped aggregates.
4. What are the least popular products by order quantity? Limit your output to the bottom 20 products. Sort this table in ascending order and be sure to include the product name in your output and visualizations. **Hint**: You will have to perform a join to calculate this answer.
5. What are the least popular products by **total revenue** (order quantity * price)? Sort this table in ascending order and be sure to include the product name in your output and visualizations. **Hint**: You will have to perform a join to calculate this answer.
6. Which country's have placed the most orders? For each country, list its' name in your output and visualizations. **Hint**: You will have to perform a join to calculate this answer.
7. Which countries have at least one customer who has placed **no** orders? Count up the total number of customers who have placed no orders for each respective country. For each country, list its' name in your output and visualizations.**Hint**: A join is necessary. We can also check if a column is `NULL` using the `IS` keyword.
8. What are the most popular suppliers according to order count? List the supplier names and their number of orders. Sort your output in descending order.**Hint**: You might need to use a subquery to join more than 2 tables together.