Optimizing Employee Data Management Using SQL

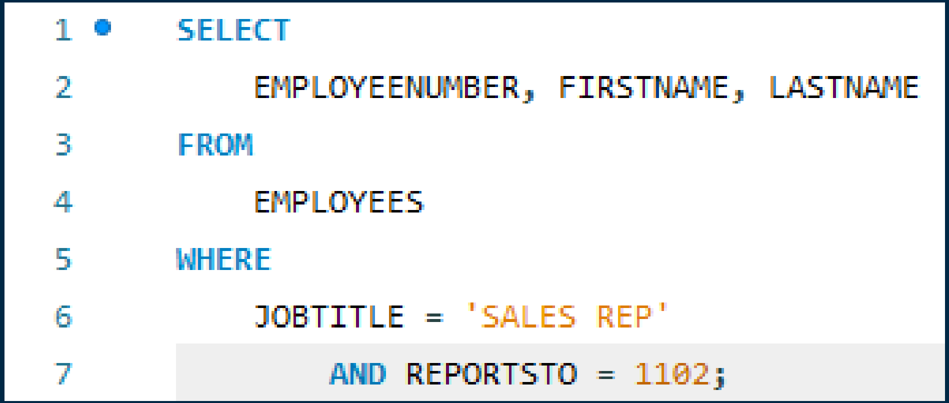
Introduction

In today's data-driven trade environment, viable administration of representative data is vital for achieving the organizational goals. Optimizing employee data management not only enhances operational efficiency but also ensures compliance with legal and regulatory requirements. SQL, or Structured Query Language, is a powerful tool for managing and querying relational databases, making it an ideal solution for handling complex employee data. This topic explores the various ways SQL can be leveraged to streamline employee data management processes, improve data accuracy, and facilitate insightful analytics. By utilizing SQL's robust capabilities, organizations can achieve better data organization, enhance reporting, and support strategic decision-making, ultimately leading to improved workforce management and productivity.

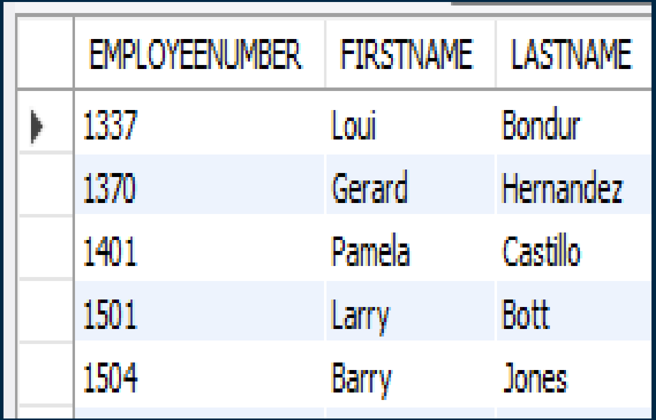
Given Below are some practical examples using SQL queries which let us understand how SQL helps in managing the data of employees and helps performing various operations on it.

1.

Fetch the employee number, first name and last name of those employees who are working as Sales Rep reporting to employee with employee number 1102



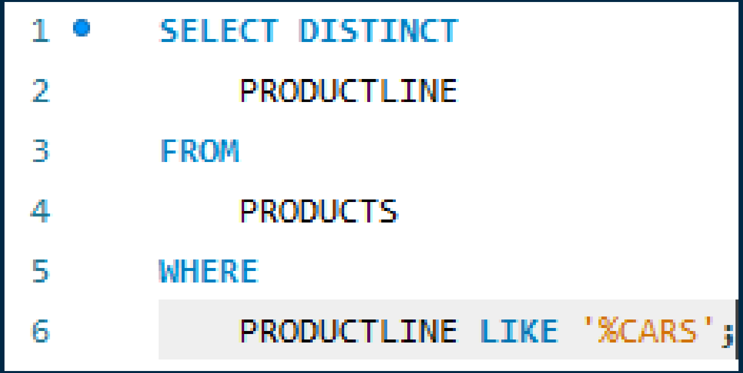
```
SELECT
    EMPLOYEENUMBER, FIRSTNAME, LASTNAME
FROM
    EMPLOYEES
WHERE
    JOBTITLE = 'SALES REP'
        AND REPORTSTO = 1102;
```



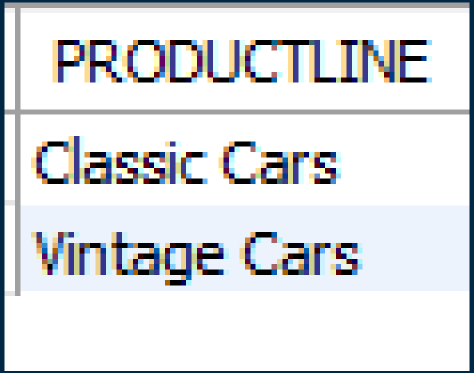
EMPLOYEENUMBER	FIRSTNAME	LASTNAME
1337	Loui	Bondur
1370	Gerard	Hernandez
1401	Pamela	Castillo
1501	Larry	Bott
1504	Barry	Jones

2.

Show the unique product line values containing the word cars at the end from the products table.



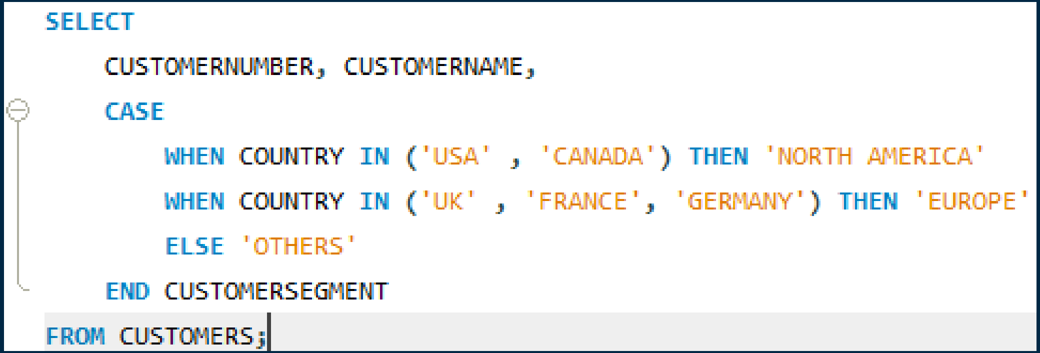
```sql
SELECT DISTINCT
    PRODUCTLINE
FROM
    PRODUCTS
WHERE
    PRODUCTLINE LIKE '%CARS';
```



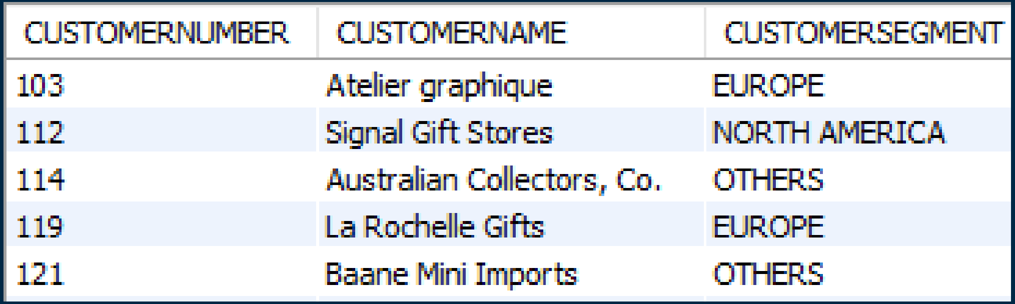
PRODUCTLINE
Classic Cars
Vintage Cars

3.

Using a CASE statement, segment customers into three categories based on their country:



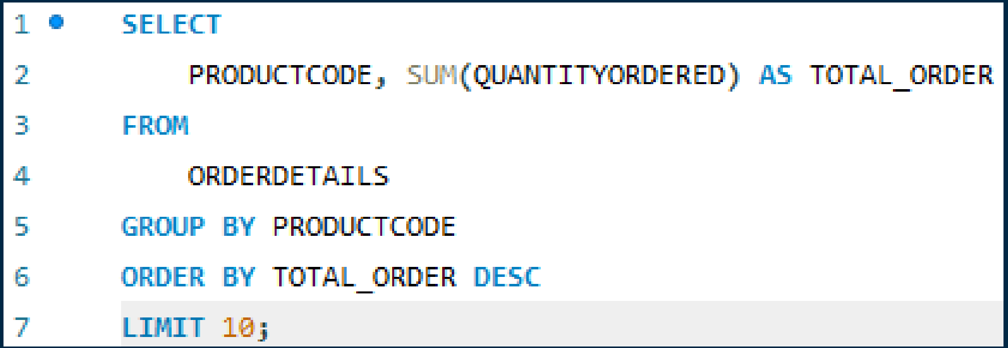
```
SELECT
    CUSTOMERNUMBER, CUSTOMERNAME,
    CASE
        WHEN COUNTRY IN ('USA' , 'CANADA') THEN 'NORTH AMERICA'
        WHEN COUNTRY IN ('UK' , 'FRANCE', 'GERMANY') THEN 'EUROPE'
        ELSE 'OTHERS'
    END CUSTOMERSEGMENT
FROM CUSTOMERS;
```



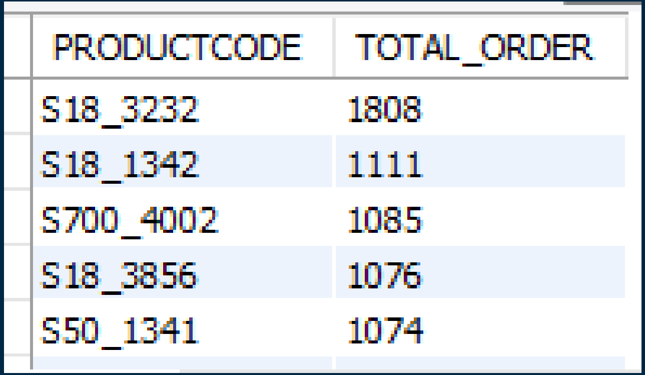
CUSTOMERNUMBER	CUSTOMERNAME	CUSTOMERSEGMENT
103	Atelier graphique	EUROPE
112	Signal Gift Stores	NORTH AMERICA
114	Australian Collectors, Co.	OTHERS
119	La Rochelle Gifts	EUROPE
121	Baane Mini Imports	OTHERS

4.

Using the OrderDetails table, identify the top 10 products (by productCode) with the highest total order quantity across all orders.



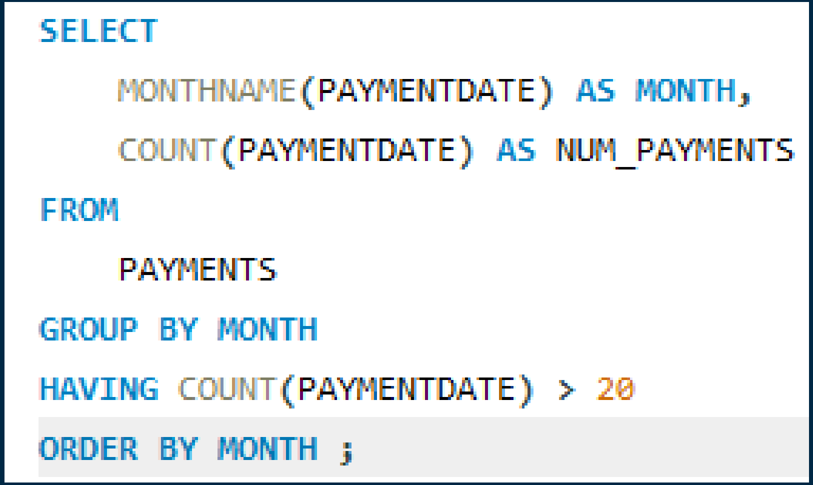
```
SELECT
    PRODUCTCODE, SUM(QUANTITYORDERED) AS TOTAL_ORDER
FROM
    ORDERDETAILS
GROUP BY PRODUCTCODE
ORDER BY TOTAL_ORDER DESC
LIMIT 10;
```



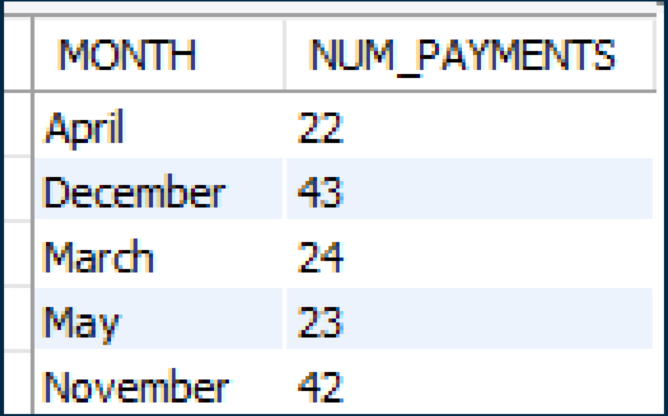
PRODUCTCODE	TOTAL_ORDER
S18_3232	1808
S18_1342	1111
S700_4002	1085
S18_3856	1076
S50_1341	1074

5.

Extract the month name from the payment date to count the total number of payments for each month and include only those months with a payment count exceeding 20



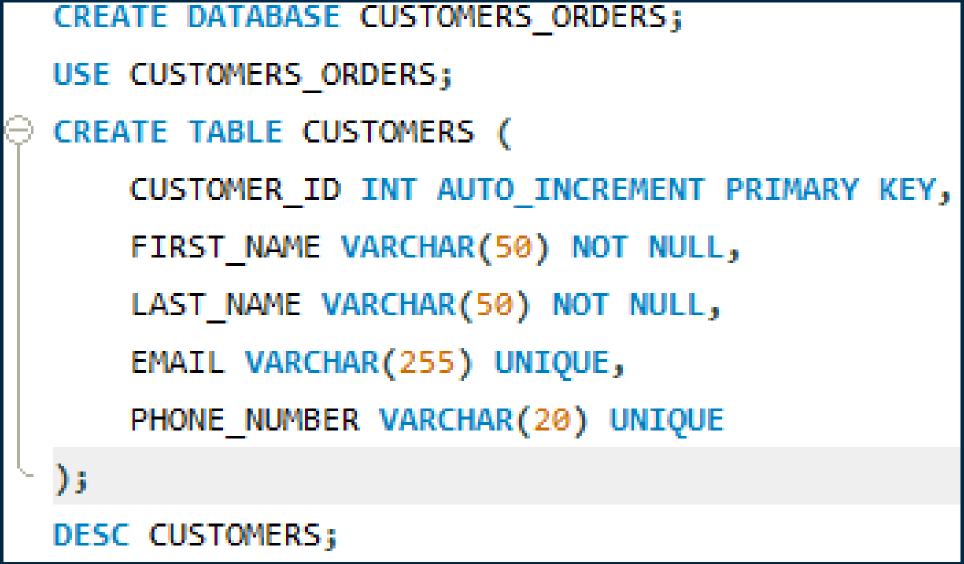
```
SELECT
    MONTHNAME(PAYMENTDATE) AS MONTH,
    COUNT(PAYMENTDATE) AS NUM_PAYMENTS
FROM
    PAYMENTS
GROUP BY MONTH
HAVING COUNT(PAYMENTDATE) > 20
ORDER BY MONTH ;
```



MONTH	NUM_PAYMENTS
April	22
December	43
March	24
May	23
November	42

6.

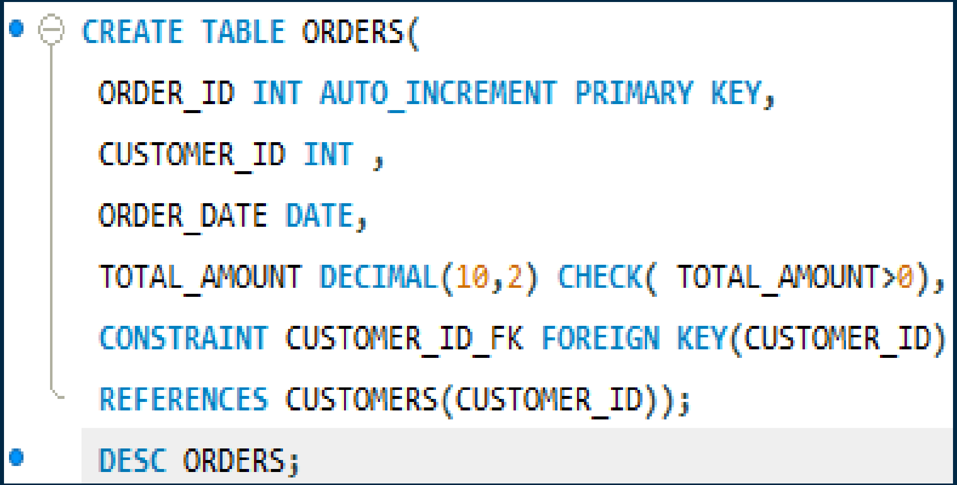
Create a new database named and Customers_Orders and add the following tables as per the description-
Create a table named Customers to store customer information. Include the following columns:



```
CREATE DATABASE CUSTOMERS_ORDERS;
USE CUSTOMERS_ORDERS;
CREATE TABLE CUSTOMERS (
    CUSTOMER_ID INT AUTO_INCREMENT PRIMARY KEY,
    FIRST_NAME VARCHAR(50) NOT NULL,
    LAST_NAME VARCHAR(50) NOT NULL,
    EMAIL VARCHAR(255) UNIQUE,
    PHONE_NUMBER VARCHAR(20) UNIQUE
);
DESC CUSTOMERS;
```

7.

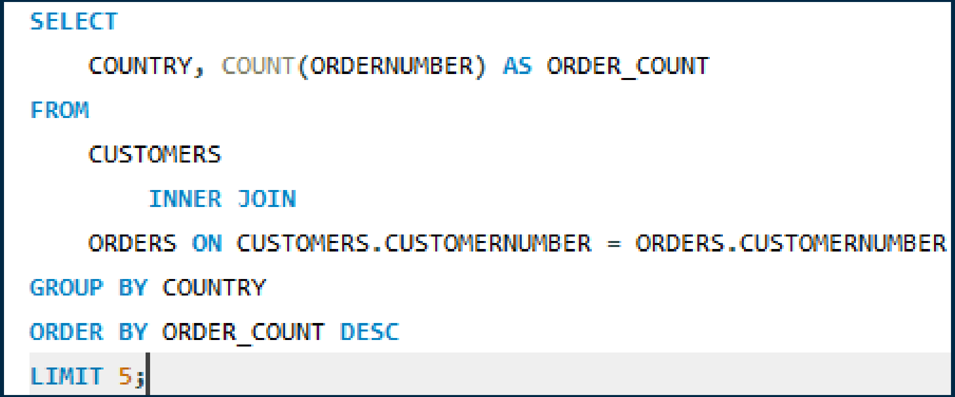
Create a table named Orders to store information about customer orders. Include the following columns:



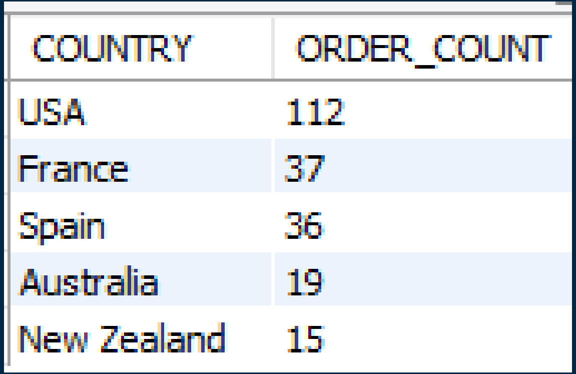
```
CREATE TABLE ORDERS(
 ORDER_ID INT AUTO_INCREMENT PRIMARY KEY,
 CUSTOMER_ID INT ,
 ORDER_DATE DATE,
 TOTAL_AMOUNT DECIMAL(10,2) CHECK( TOTAL_AMOUNT>0),
 CONSTRAINT CUSTOMER_ID_FK FOREIGN KEY(CUSTOMER_ID)
 REFERENCES CUSTOMERS(CUSTOMER_ID));
 DESC ORDERS;
```

8.

List the top 5 countries (by order count) that Classic Models ships to.



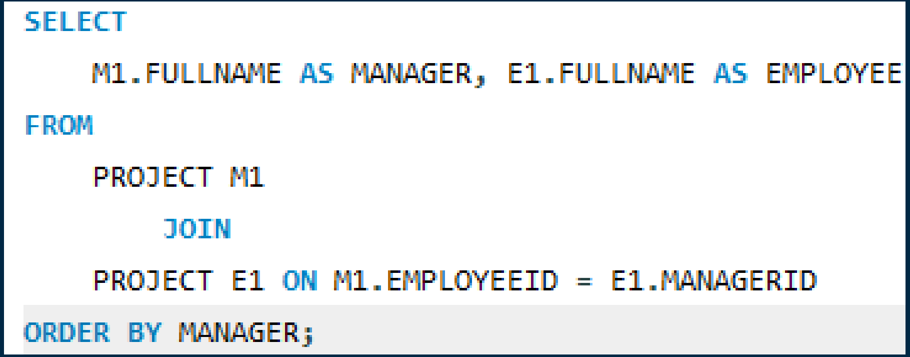
```
SELECT
    COUNTRY, COUNT(ORDERNUMBER) AS ORDER_COUNT
FROM
    CUSTOMERS
        INNER JOIN
    ORDERS ON CUSTOMERS.CUSTOMERNUMBER = ORDERS.CUSTOMERNUMBER
GROUP BY COUNTRY
ORDER BY ORDER_COUNT DESC
LIMIT 5;
```



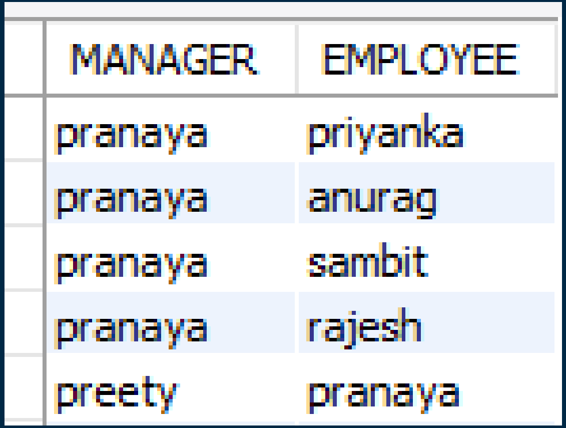
COUNTRY	ORDER_COUNT
USA	112
France	37
Spain	36
Australia	19
New Zealand	15

9.

Find out the names of employees and their related managers.



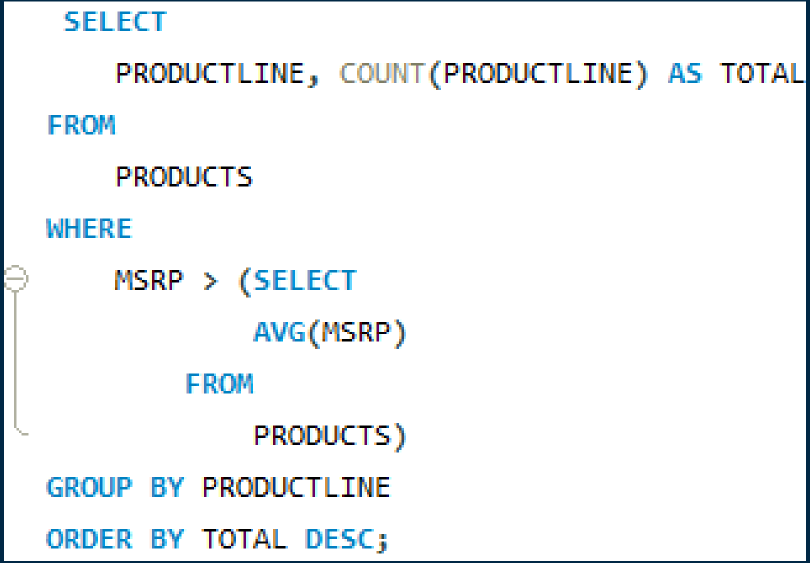
```
SELECT
    M1.FULLNAME AS MANAGER, E1.FULLNAME AS EMPLOYEE
FROM
    PROJECT M1
        JOIN
    PROJECT E1 ON M1.EMPLOYEEID = E1.MANAGERID
ORDER BY MANAGER;
```



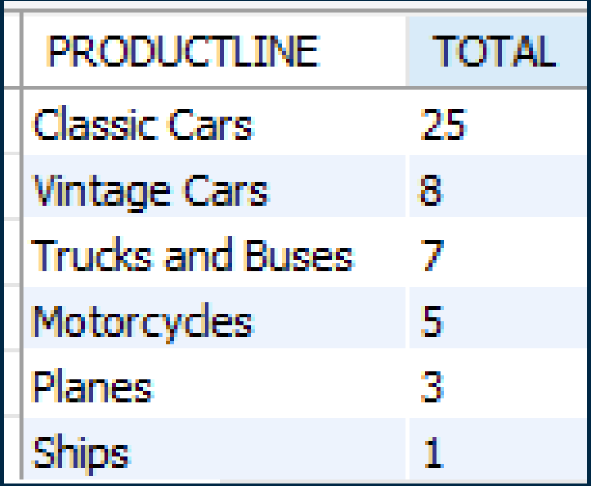
MANAGER	EMPLOYEE
pranaya	priyanka
pranaya	anurag
pranaya	sambit
pranaya	rajesh
preety	pranaya

10.

Find out how many product lines are there for which the buy price value is greater than the average of buy price value. Show the output as product line and its count.



```
 SELECT
     PRODUCTLINE, COUNT(PRODUCTLINE) AS TOTAL
FROM
     PRODUCTS
WHERE
     MSRP > (SELECT
             AVG(MSRP)
         FROM
             PRODUCTS)
GROUP BY PRODUCTLINE
ORDER BY TOTAL DESC;
```



PRODUCTLINE	TOTAL
Classic Cars	25
Vintage Cars	8
Trucks and Buses	7
Motorcycles	5
Planes	3
Ships	1

THANK YOU!