

Questions A (1 marks x 10) = 10 Marks

1. From the Suppliers table, retrieve all supplier information (Id, CompanyName, ContactName, City, Country)

SELECT Id, CompanyName, ContactName, City, Country FROM Suppliers;

```
Command Prompt - mysql -u < > + - x
| 4 | Tokyo Traders | Yoshi Nagase | Tokyo | Japan | | | |
| 5 | Cooperativa de Quesos 'Las Cabras' | Antonio del Valle Saavedra | Oviedo | Spain |
| 6 | Mayumi's | Mayumi Ohno | Osaka | Japan |
| 7 | Pavlova, Ltd. | Ian Devling | Melbourne | Australia |
| 8 | Specialty Biscuits, Ltd. | Peter Wilson | Manchester | UK |
| 9 | PB Kn|fckebr|Ad AB | Lars Peterson | G|Ateborg | Sweden |
| 10 | Refrescos Americanas LTDA | Carlos Diaz | Sao Paulo | Brazil |
| 11 | Heli S|d|fwaren GmbH & Co. KG | Petra Winkler | Berlin | Germany |
| 12 | Plutzer Lebensmittelgro|fm|rkte AG | Martin Bein | Frankfurt | Germany |
| 13 | Nord-Ost-Fisch Handelsgesellschaft mbH | Sven Petersen | Cuxhaven | Germany |
| 14 | Formaggi Fortini s.r.l. | Elio Rossi | Ravenna | Italy |
| 15 | Norske Meierier | Beate Vileid | Sandvika | Norway |
| 16 | Bigfoot Breweries | Cheryl Saylor | Bend | USA |
| 17 | Svensk Sjt|Af|Ada AB | Michael Bj|Arn | Stockholm | Sweden |
| 18 | Aux joyeux eccl|siastiques | Guyl|ne Nodier | Paris | France |
| 19 | New England Seafood Cannery | Robb Merchant | Boston | USA |
| 20 | Leka Trading | Chandra Leka | Singapore | Singapore |
| 21 | Lyngbysild | Niels Petersen | Lyngby | Denmark |
| 22 | Zaanse Snoepfabriek | Dirk Luchte | Zaandam | Netherlands |
| 23 | Karkki Oy | Anne Heikkonen | Lappeenranta | Finland |
| 24 | G'day, Mate | Wendy Mackenzie | Sydney | Australia |
| 25 | Ma Maison | Jean-Guy Lauzon | Montr|al | Canada |
| 26 | Pasta Buttini s.r.l. | Giovanni Giudici | Salerno | Italy |
| 27 | Escargots Nouveaux | Marie Delamare | Montceau | France |
| 28 | Gai p|oturage | Eliane Noz | Annecy | France |
| 29 | For|ts d'|ables | Chantal Goulet | Ste-Hyacinthe | Canada |
+-----+
29 rows in set (0.00 sec)

mysql> SELECT Id, CompanyName, ContactName, City, Country FROM Suppliers;
```

2. Display the names of suppliers located in the United States of America.

SELECT CompanyName FROM SUPPLIERS WHERE Country = 'USA';

```
Command Prompt - mysql -u < > + - x
| New Orleans Cajun Delights |
| Grandma Kelly's Homestead |
| Bigfoot Breweries |
| New England Seafood Cannery |
+-----+
4 rows in set (0.00 sec)

mysql> SELECT CompanyName FROM SUPPLIERS WHERE Country = 'USA';
+-----+
| CompanyName |
+-----+
| New Orleans Cajun Delights |
| Grandma Kelly's Homestead |
| Bigfoot Breweries |
| New England Seafood Cannery |
+-----+
4 rows in set (0.00 sec)

mysql> SELECT CompanyName FROM SUPPLIERS WHERE Country = 'USA';
+-----+
| CompanyName |
+-----+
| New Orleans Cajun Delights |
| Grandma Kelly's Homestead |
| Bigfoot Breweries |
| New England Seafood Cannery |
+-----+
4 rows in set (0.00 sec)

mysql>
```

3. Retrieve a supplier's contact name and phone number with the ID 11.

SELECT ContactName, Phone FROM SUPPLIERS WHERE Id = 11;

```
Command Prompt - mysql -u x + v
+-----+
| Petra Winkler | (010) 9984510 |
+-----+
1 row in set (0.01 sec)

mysql> SELECT ContactName, Phone FROM SUPPLIERS WHERE Id = 11;
+-----+
| ContactName | Phone |
+-----+
| Petra Winkler | (010) 9984510 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT ContactName, Phone FROM SUPPLIERS WHERE Id = 11;
+-----+
| ContactName | Phone |
+-----+
| Petra Winkler | (010) 9984510 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT ContactName, Phone FROM SUPPLIERS WHERE Id = 11;
+-----+
| ContactName | Phone |
+-----+
| Petra Winkler | (010) 9984510 |
+-----+
1 row in set (0.00 sec)

mysql>
```

4. Get a particular order's total amount and date by its ID 783.

SELECT TotalAmount FROM ORDERS WHERE Id = 783;

```
Command Prompt - mysql -u x + v
+-----+
| 16321.90 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT TotalAmount FROM ORDERS WHERE Id = 783;
+-----+
| TotalAmount |
+-----+
| 16321.90 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT TotalAmount FROM ORDERS WHERE Id = 783;
+-----+
| TotalAmount |
+-----+
| 16321.90 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT TotalAmount FROM ORDERS WHERE Id = 783;
+-----+
| TotalAmount |
+-----+
| 16321.90 |
+-----+
1 row in set (0.00 sec)

mysql>
```

5. Count the orders placed in December month of any year.

SELECT COUNT(*) FROM ORDERS WHERE MONTH(OrderDate) = 12;

```
Command Prompt - mysql -u x + v
+-----+
|      79 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT COUNT(*) FROM ORDERS WHERE MONTH(OrderDate) = 12;
+-----+
| COUNT(*) |
+-----+
|      79 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT COUNT(*) FROM ORDERS WHERE MONTH(OrderDate) = 12;
+-----+
| COUNT(*) |
+-----+
|      79 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT COUNT(*) FROM ORDERS WHERE MONTH(OrderDate) = 12;
+-----+
| COUNT(*) |
+-----+
|      79 |
+-----+
1 row in set (0.00 sec)

mysql>
```

6. Count the number of orders placed by a customer with ID 10.

SELECT COUNT(*) FROM ORDERS WHERE CustomerId = 10;

```
Command Prompt - mysql -u x + v
+-----+
|      14 |
+-----+
1 row in set (0.02 sec)

mysql> SELECT COUNT(*) FROM ORDERS WHERE CustomerId = 10;
+-----+
| COUNT(*) |
+-----+
|      14 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT COUNT(*) FROM ORDERS WHERE CustomerId = 10;
+-----+
| COUNT(*) |
+-----+
|      14 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT COUNT(*) FROM ORDERS WHERE CustomerId = 10;
+-----+
| COUNT(*) |
+-----+
|      14 |
+-----+
1 row in set (0.00 sec)

mysql>
```

7. Find the average unit price of products supplied by a specific supplier.

SELECT AVG(UnitPrice) FROM PRODUCTS WHERE SupplierId = 11;

```
Command Prompt - mysql -u x + v
+-----+
| 29.710000 |
+-----+
1 row in set (0.02 sec)

mysql> SELECT AVG(UnitPrice) FROM PRODUCTS WHERE SupplierId = 11;
+-----+
| AVG(UnitPrice) |
+-----+
| 29.710000 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT AVG(UnitPrice) FROM PRODUCTS WHERE SupplierId = 11;
+-----+
| AVG(UnitPrice) |
+-----+
| 29.710000 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT AVG(UnitPrice) FROM PRODUCTS WHERE SupplierId = 11;
+-----+
| AVG(UnitPrice) |
+-----+
| 29.710000 |
+-----+
1 row in set (0.00 sec)

mysql>
```

8. Display the names of customers who have placed at least one order.

SELECT DISTINCT FirstName, LastName FROM CUSTOMERS INNER JOIN ORDERS ON CUSTOMERS.Id = ORDERS.CustomerId;

```
Command Prompt - mysql -u x + v
| Maurizio | Moroni |
| Janete | Limeira |
| Michael | Holz |
| Alejandra | Camino |
| Jonas | Bergulfsen |
| Jose | Pavarotti |
| Hari | Kumar |
| Jytte | Petersen |
| Dominique | Perrier |
| Art | Braunschweiger |
| Pascale | Cartrain |
| Liz | Nixon |
| Liu | Wong |
| Karin | Josep |
| Miguel | Angel Paolino |
| Anabela | Domingues |
| Helvetius | Nagy |
| Palle | Ibsen |
| Mary | Saveley |
| Paul | Henriot |
| Rita | Muller |
| Pirkko | Koskitalo |
| Paula | Parente |
| Karl | Jablonski |
| Matti | Karttunen |
| Zbyszek | Piestrzeniewicz |
+-----+
89 rows in set (0.00 sec)

mysql> SELECT DISTINCT FirstName, LastName FROM CUSTOMERS INNER JOIN ORDERS ON CUSTOMERS.Id = ORDERS.CustomerId;
```

9. Get the total number of orders placed by each customer.

SELECT CustomerId, COUNT(*) AS TotalOrders FROM Orders GROUP BY CustomerId;

```
Command Prompt - mysql -u x + v
+-----+
| CustomerId | TotalOrders |
+-----+
| 68 | 10 |
| 69 | 5 |
| 70 | 6 |
| 71 | 31 |
| 72 | 9 |
| 73 | 7 |
| 74 | 4 |
| 75 | 9 |
| 76 | 12 |
| 77 | 4 |
| 78 | 3 |
| 79 | 6 |
| 80 | 10 |
| 81 | 6 |
| 82 | 3 |
| 83 | 11 |
| 84 | 10 |
| 85 | 5 |
| 86 | 10 |
| 87 | 15 |
| 88 | 9 |
| 89 | 14 |
| 90 | 7 |
| 91 | 7 |
+-----+
89 rows in set (0.00 sec)

mysql> SELECT CustomerId, COUNT(*) AS TotalOrders
-> FROM Orders
-> GROUP BY CustomerId;
```

10. Display the product name and unit price for products with a unit price greater than \$50.

SELECT ProductName, UnitPrice FROM Products WHERE UnitPrice > 50;

```
Command Prompt - mysql -u x + v

mysql> SELECT ProductName, UnitPrice FROM Products WHERE UnitPrice > 50;
+-----+-----+
| ProductName | UnitPrice |
+-----+-----+
| Mishu Kobe Niku | 97.00 |
| Carnarvon Tigers | 62.50 |
| Sir Rodney's Marmalade | 81.00 |
| Thring Rostbratwurst | 123.79 |
| Chate de Blaye | 263.50 |
| Manjimup Dried Apples | 53.00 |
| Raclette Courdavault | 55.00 |
+-----+-----+
7 rows in set (0.00 sec)

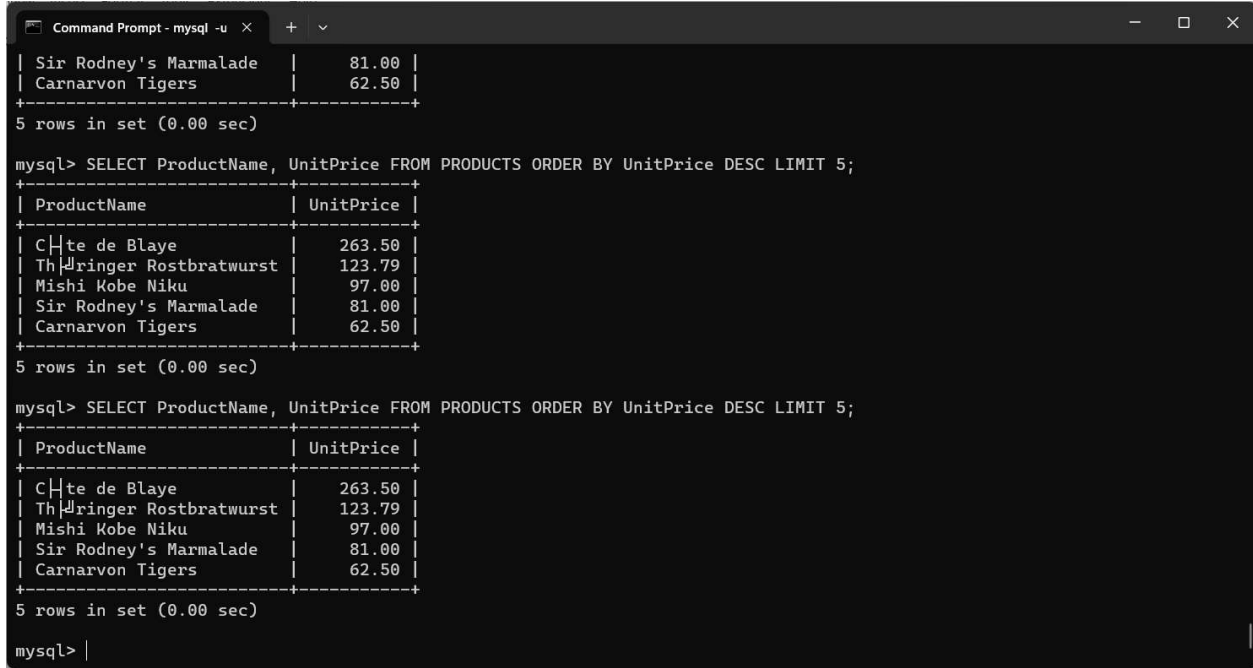
mysql> SELECT ProductName, UnitPrice FROM Products WHERE UnitPrice > 50;
+-----+-----+
| ProductName | UnitPrice |
+-----+-----+
| Mishu Kobe Niku | 97.00 |
| Carnarvon Tigers | 62.50 |
| Sir Rodney's Marmalade | 81.00 |
| Thring Rostbratwurst | 123.79 |
| Chate de Blaye | 263.50 |
| Manjimup Dried Apples | 53.00 |
| Raclette Courdavault | 55.00 |
+-----+-----+
7 rows in set (0.00 sec)

mysql> |
```

Questions (3 marks x 10) = 30 Marks

1. List the top 5 most expensive products.

SELECT ProductName, UnitPrice FROM PRODUCTS ORDER BY UnitPrice
DESC LIMIT 5;



```
Command Prompt - mysql -u x + v
| Sir Rodney's Marmalade | 81.00 |
| Carnarvon Tigers | 62.50 |
+-----+
5 rows in set (0.00 sec)

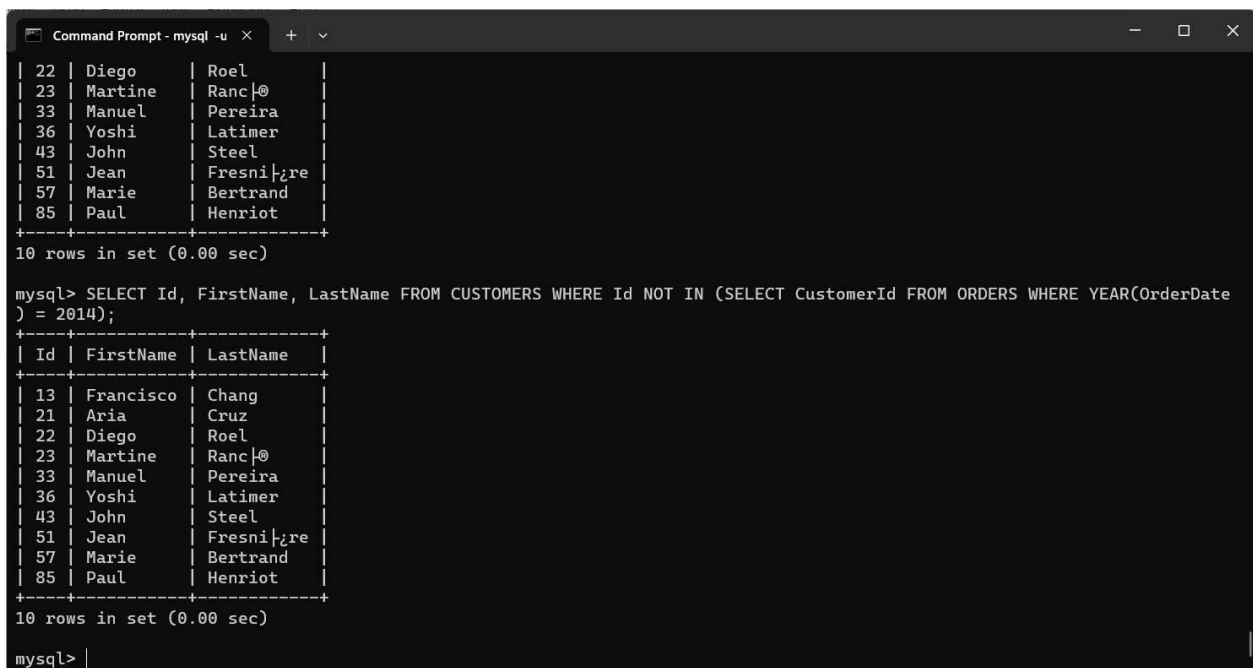
mysql> SELECT ProductName, UnitPrice FROM PRODUCTS ORDER BY UnitPrice DESC LIMIT 5;
+-----+
| ProductName | UnitPrice |
+-----+
| Chateau de Blaye | 263.50 |
| Thüringer Rostbratwurst | 123.79 |
| Mishi Kobe Niku | 97.00 |
| Sir Rodney's Marmalade | 81.00 |
| Carnarvon Tigers | 62.50 |
+-----+
5 rows in set (0.00 sec)

mysql> SELECT ProductName, UnitPrice FROM PRODUCTS ORDER BY UnitPrice DESC LIMIT 5;
+-----+
| ProductName | UnitPrice |
+-----+
| Chateau de Blaye | 263.50 |
| Thüringer Rostbratwurst | 123.79 |
| Mishi Kobe Niku | 97.00 |
| Sir Rodney's Marmalade | 81.00 |
| Carnarvon Tigers | 62.50 |
+-----+
5 rows in set (0.00 sec)

mysql> |
```

2. Identify customers who have not placed any orders in the year 2014.

SELECT Id, FirstName, LastName FROM CUSTOMERS WHERE Id NOT IN
(SELECT CustomerId FROM ORDERS WHERE YEAR(OrderDate) = 2014);



```
Command Prompt - mysql -u x + v
| 22 | Diego | Roel |
| 23 | Martine | Ranc |
| 33 | Manuel | Pereira |
| 36 | Yoshi | Latimer |
| 43 | John | Steel |
| 51 | Jean | Fresniere |
| 57 | Marie | Bertrand |
| 85 | Paul | Henriot |
+-----+
10 rows in set (0.00 sec)

mysql> SELECT Id, FirstName, LastName FROM CUSTOMERS WHERE Id NOT IN (SELECT CustomerId FROM ORDERS WHERE YEAR(OrderDate) = 2014);
+-----+
| Id | FirstName | LastName |
+-----+
| 13 | Francisco | Chang |
| 21 | Aria | Cruz |
| 22 | Diego | Roel |
| 23 | Martine | Ranc |
| 33 | Manuel | Pereira |
| 36 | Yoshi | Latimer |
| 43 | John | Steel |
| 51 | Jean | Fresniere |
| 57 | Marie | Bertrand |
| 85 | Paul | Henriot |
+-----+
10 rows in set (0.00 sec)

mysql> |
```

3. Find the total revenue generated in the October month.

SELECT SUM(TotalAmount) FROM ORDERS WHERE MONTH(OrderDate) = 10;

```
Command Prompt - mysql -u x + v
+-----+
| 111532.10 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT SUM(TotalAmount) FROM ORDERS WHERE MONTH(OrderDate) = 10;
+-----+
| SUM(TotalAmount) |
+-----+
| 111532.10 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT SUM(TotalAmount) FROM ORDERS WHERE MONTH(OrderDate) = 10;
+-----+
| SUM(TotalAmount) |
+-----+
| 111532.10 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT SUM(TotalAmount) FROM ORDERS WHERE MONTH(OrderDate) = 10;
+-----+
| SUM(TotalAmount) |
+-----+
| 111532.10 |
+-----+
1 row in set (0.00 sec)

mysql>
```

4. List the products that have never been ordered.

SELECT ProductName FROM PRODUCTS WHERE Id NOT IN (SELECT ProductId FROM ORDERITEM);

```
Command Prompt - mysql -u x + v
+-----+
| Stroopwafels |
+-----+
1 row in set (0.05 sec)

mysql> SELECT ProductName FROM PRODUCTS WHERE Id NOT IN (SELECT ProductId FROM ORDERITEM);
+-----+
| ProductName |
+-----+
| Stroopwafels |
+-----+
1 row in set (0.00 sec)

mysql> SELECT ProductName FROM PRODUCTS WHERE Id NOT IN (SELECT ProductId FROM ORDERITEM);
+-----+
| ProductName |
+-----+
| Stroopwafels |
+-----+
1 row in set (0.00 sec)

mysql> SELECT ProductName FROM PRODUCTS WHERE Id NOT IN (SELECT ProductId FROM ORDERITEM);
+-----+
| ProductName |
+-----+
| Stroopwafels |
+-----+
1 row in set (0.00 sec)

mysql>
```

5. Retrieve the names of suppliers who provide more than 3 different products.

```
SELECT CompanyName FROM SUPPLIERS WHERE (SELECT COUNT(*)  
FROM PRODUCTS WHERE SupplierId = SUPPLIERS.Id) > 3;
```

```
Command Prompt - mysql -u x + v  
| New Orleans Cajun Delights |  
| Pavlova, Ltd. |  
| Specialty Biscuits, Ltd. |  
| Plutzer Lebensmittelgroßmärkte AG |  
+-----+  
4 rows in set (0.00 sec)  
  
mysql> SELECT CompanyName FROM SUPPLIERS WHERE (SELECT COUNT(*) FROM PRODUCTS WHERE SupplierId = SUPPLIERS.Id) > 3;  
+-----+  
| CompanyName |  
+-----+  
| New Orleans Cajun Delights |  
| Pavlova, Ltd. |  
| Specialty Biscuits, Ltd. |  
| Plutzer Lebensmittelgroßmärkte AG |  
+-----+  
4 rows in set (0.00 sec)  
  
mysql> SELECT CompanyName FROM SUPPLIERS WHERE (SELECT COUNT(*) FROM PRODUCTS WHERE SupplierId = SUPPLIERS.Id) > 3;  
+-----+  
| CompanyName |  
+-----+  
| New Orleans Cajun Delights |  
| Pavlova, Ltd. |  
| Specialty Biscuits, Ltd. |  
| Plutzer Lebensmittelgroßmärkte AG |  
+-----+  
4 rows in set (0.00 sec)  
  
mysql> |
```

6. Calculate the average order value for each customer.

```
SELECT AVG(TotalAmount) AS AvgOrderValue FROM ORDERS GROUP BY  
CustomerId;
```

```
Command Prompt - mysql -u x + v  
| 629.633333 |  
| 1174.945455 |  
| 2003.320000 |  
| 293.458000 |  
| 955.858333 |  
| 3731.399677 |  
| 1908.005556 |  
| 2591.207143 |  
| 605.837500 |  
| 1387.744444 |  
| 2058.700000 |  
| 840.250000 |  
| 649.080000 |  
| 825.666667 |  
| 1081.215000 |  
| 1218.436667 |  
| 523.733333 |  
| 1513.072727 |  
| 993.710000 |  
| 296.000000 |  
| 1065.385000 |  
| 1107.806667 |  
| 720.077778 |  
| 2076.675000 |  
| 451.621429 |  
| 504.564286 |  
+-----+  
89 rows in set (0.00 sec)  
  
mysql> SELECT AVG(TotalAmount) AS AvgOrderValue FROM ORDERS GROUP BY CustomerId;|
```


7. Find the most popular product (the one with the highest total quantity ordered).

SELECT ProductName, SUM(Quantity) as TotalQuantity FROM PRODUCTS, ORDERITEM WHERE PRODUCTS.Id = ORDERITEM.ProductId GROUP BY ProductId ORDER BY TotalQuantity DESC LIMIT 1;

```
Command Prompt - mysql -u x + v
1 row in set (0.01 sec)

mysql> SELECT ProductName, SUM(Quantity) as TotalQuantity FROM PRODUCTS, ORDERITEM WHERE PRODUCTS.Id = ORDERITEM.Product
Id GROUP BY ProductId ORDER BY TotalQuantity DESC LIMIT 1;
+-----+-----+
| ProductName | TotalQuantity |
+-----+-----+
| Camembert Pierrot | 1577 |
+-----+-----+
1 row in set (0.01 sec)

mysql> SELECT ProductName, SUM(Quantity) as TotalQuantity FROM PRODUCTS, ORDERITEM WHERE PRODUCTS.Id = ORDERITEM.Product
Id GROUP BY ProductId ORDER BY TotalQuantity DESC LIMIT 1;
+-----+-----+
| ProductName | TotalQuantity |
+-----+-----+
| Camembert Pierrot | 1577 |
+-----+-----+
1 row in set (0.01 sec)

mysql> SELECT ProductName, SUM(Quantity) as TotalQuantity FROM PRODUCTS, ORDERITEM WHERE PRODUCTS.Id = ORDERITEM.Product
Id GROUP BY ProductId ORDER BY TotalQuantity DESC LIMIT 1;
+-----+-----+
| ProductName | TotalQuantity |
+-----+-----+
| Camembert Pierrot | 1577 |
+-----+-----+
1 row in set (0.01 sec)

mysql> |
```

8. List the customers who have spent more than a certain amount in total.

SELECT FirstName, LastName FROM CUSTOMERS WHERE (SELECT SUM(TotalAmount) FROM ORDERS WHERE CustomerId = CUSTOMERS.Id) > 100000;

```
Command Prompt - mysql -u x + v
+-----+
| Roland | Mendel |
| Horst  | Kloss  |
| Jose   | Pavarotti |
+-----+
3 rows in set (0.00 sec)

mysql> SELECT FirstName, LastName FROM CUSTOMERS WHERE (SELECT SUM(TotalAmount) FROM ORDERS WHERE CustomerId = CUSTOMERS
.Id) > 100000;
+-----+
| FirstName | LastName |
+-----+
| Roland    | Mendel   |
| Horst     | Kloss    |
| Jose      | Pavarotti |
+-----+
3 rows in set (0.00 sec)

mysql> SELECT FirstName, LastName FROM CUSTOMERS WHERE (SELECT SUM(TotalAmount) FROM ORDERS WHERE CustomerId = CUSTOMERS
.Id) > 100000;
+-----+
| FirstName | LastName |
+-----+
| Roland    | Mendel   |
| Horst     | Kloss    |
| Jose      | Pavarotti |
+-----+
3 rows in set (0.00 sec)

mysql> |
```

9. Identify products that have been supplied by multiple suppliers.

SELECT DISTINCT P1.ProductName FROM Products P1, Products P2 WHERE
P1.SupplierId <> P2.SupplierId and P1.Id=P2.Id;

Empty set

10. Find the total number of orders placed in each country.

SELECT c.Country, COUNT(o.Id) AS TotalOrders
FROM Customers c
JOIN Orders o ON c.Id = o.CustomerId
GROUP BY c.Country;

```
-> JOIN Orders o ON c.Id = o.CustomerId
-> GROUP BY c.Country;
```

Country	TotalOrders
Germany	122
Mexico	28
UK	56
Sweden	37
France	77
Spain	23
Canada	30
Argentina	16
Switzerland	18
Brazil	83
Austria	40
Italy	28
Portugal	13
USA	122
Venezuela	46
Ireland	19
Belgium	19
Norway	6
Denmark	18
Finland	22
Poland	7

21 rows in set (0.03 sec)

mysql>