1. Select first 5 rows of the "CustomerName" and "City" columns from the "Customers" table.

```
mysql> SELECT CustomerName, City FROM Customers LIMIT 5;
+------------------------------------+-------------+
| CustomerName                       | City        |
+------------------------------------+-------------+
| Alfreds Futterkiste                | Berlin      |
| Ana Trujillo Emparedados y helados | Mexico D.F. |
| Antonio Moreno Taquerna            | Mexico D.F. |
| Around the Horn                    | London      |
| Berglunds snabbkup                 | Lulee       |
+------------------------------------+-------------+
5 rows in set (0.01 sec)
```

2. Select all the customers (their name and country name) from the country "Mexico".

```
mysql> SELECT CustomerName, Country FROM Customers WHERE Country="Mexico";
+------------------------------------+---------+
| CustomerName                       | Country |
+------------------------------------+---------+
| Ana Trujillo Emparedados y helados | Mexico  |
| Antonio Moreno Taquerna            | Mexico  |
| Centro comercial Moctezuma         | Mexico  |
| Pericles Comidas clasicas          | Mexico  |
| Tortuga Restaurante                | Mexico  |
+------------------------------------+---------+
5 rows in set (0.02 sec)
```

3. Select all customers' records from the country "Germany" and the city "Berlin" or "Mannheim", sort by the Address from A to Z.

```
mysql> SELECT * FROM Customers WHERE Country="Germany" AND City="Berlin" OR City="Mannheim" ORDER BY Customername ASC;
+------------+-------------------------+-------------+---------------+----------+------------+---------+
| CustomerID | CustomerName            | ContactName | Address       | City     | PostalCode | Country |
+------------+-------------------------+-------------+---------------+----------+------------+---------+
|          1 | Alfreds Futterkiste     | Maria Anders | Obere Str. 57 | Berlin   | 12209      | Germany |
|          6 | Blauer See Delikatessen | Hanna Moos   | Forsterstr. 57 | Mannheim | 68306      | Germany |
+------------+-------------------------+-------------+---------------+----------+------------+---------+
2 rows in set (0.02 sec)
```

4. Select all customers' records, whose name ends with "s", from country, which name does not contain the pattern "land" and starts with "b", sort by the Country from A to Z.

```
mysql> SELECT * FROM Customers WHERE CustomerName LIKE "%s"
    -> AND Country NOT LIKE "%land%"
    -> AND Country LIKE "b%"
    -> ORDER BY Country ASC;
+------------+-----------------------+-----------------+-------------------------+---------------+------------+---------+
| CustomerID | CustomerName          | ContactName     | Address                 | City          | PostalCode | Country |
+------------+-----------------------+-----------------+-------------------------+---------------+------------+---------+
|         76 | Suprumes dulices      | Pascale Cartrain | Boulevard Tirou, 255    | Charleroi     | B-6000     | Belgium |
|         31 | Gourmet Lanchonetes   | Andru Fonseca   | Av. Brasil, 442         | Campinas      | 04876-786  | Brazil  |
|         34 | Hanari Carnes         | Mario Pontes    | Rua do Pauo, 67         | Rio de Janeiro | 05454-876  | Brazil  |
|         67 | Ricardo Adocicados    | Janete Limeira  | Av. Copacabana, 267     | Rio de Janeiro | 02389-890  | Brazil  |
|         81 | Tradibro Hipermercados | Anabela Domingues | Av. Inks de Castro, 414 | Suo Paulo     | 05634-030  | Brazil  |
+------------+-----------------------+-----------------+-------------------------+---------------+------------+---------+
5 rows in set (0.02 sec)
```

5. Select all customers' records from a City of "Paris", "Berlin", "Mannheim" or "London" and their IDs greater than 10, but less than 20.

```
mysql> SELECT * FROM Customers WHERE City IN ("Paris", "Berlin", "Mannheim","London")
    -> AND CustomerID>10
    -> AND CustomerID<20;
+------------+---------------------+-----------------+---------------------------+--------+------------+---------+
| CustomerID | CustomerName        | ContactName     | Address                   | City   | PostalCode | Country |
+------------+---------------------+-----------------+---------------------------+--------+------------+---------+
|         11 | B's Beverages       | Victoria Ashworth | Fauntleroy Circus       | London | EC2 5NT    | UK      |
|         16 | Consolidated Holdings | Elizabeth Brown | Berkeley Gardens 12 Brewery | London | WX1 6LT | UK      |
|         19 | Eastern Connection  | Ann Devon       | 35 King George            | London | WX3 6FW    | UK      |
+------------+---------------------+-----------------+---------------------------+--------+------------+---------+
3 rows in set (0.02 sec)
```

6. Selects all orders with an ordered between 04-July-1996 and 09-July-1996.

```
mysql> SELECT * FROM Orders WHERE OrderDate BETWEEN "1996-07-04" AND "1996-07-09";
+---------+------------+------------+------------+------------+
| OrderID | CustomerID | EmployeeID | OrderDate  | ShipperID  |
+---------+------------+------------+------------+------------+
|   10248 |         90 |          5 | 1996-07-04 |          3 |
|   10249 |         81 |          6 | 1996-07-05 |          1 |
|   10250 |         34 |          4 | 1996-07-08 |          2 |
|   10251 |         84 |          3 | 1996-07-08 |          1 |
|   10252 |         76 |          4 | 1996-07-09 |          2 |
+---------+------------+------------+------------+------------+
5 rows in set (0.02 sec)
```

7. Select the list of orders (its ID, customer's Name and the date of an order) which were made in the second decade of February, 1997

```
mysql> SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate
    -> FROM Orders JOIN Customers ON Orders.CustomerID = Customers.CustomerID
    -> WHERE Orders.OrderDate BETWEEN "1997-02-11" AND "1997-02-20";
+---------+--------------------+------------+
| OrderID | CustomerName       | OrderDate  |
+---------+--------------------+------------+
|   10442 | Ernst Handel       | 1997-02-11 |
|   10443 | Reggiani Caseifici | 1997-02-12 |
+---------+--------------------+------------+
2 rows in set (0.02 sec)
```

8. Find all employees (their names) and "Amount of orders" they made (if any), sort them by name from A to Z, ensure all result columns have appropriate names.

```
mysql> SELECT Employees.FirstName, COUNT(Orders.EmployeeID) AS AmountOfOrders
    -> FROM Employees LEFT JOIN Orders ON Employees.EmployeeID = Orders.EmployeeID
    -> GROUP BY Employees.FirstName ASC;
+-----------+----------------+
| FirstName | AmountOfOrders |
+-----------+----------------+
| Adam      |              0 |
| Andrew    |             20 |
| Anne      |              6 |
| Janet     |             31 |
| Laura     |             27 |
| Margaret  |             40 |
| Michael   |             18 |
| Nancy     |             29 |
| Robert    |             14 |
| Steven    |             11 |
+-----------+----------------+
10 rows in set, 1 warning (0.02 sec)
```

9. Select all unique UK cities, where customers and suppliers live, sort from A to Z.

```
mysql> SELECT DISTINCT Customers.City, Customers.Country FROM Customers
    -> WHERE Customers.Country = "UK"
    -> UNION
    -> SELECT DISTINCT Suppliers.City, Suppliers.Country FROM Suppliers
    -> WHERE Suppliers.Country = "UK"
    -> ORDER BY City ASC;
+------------+---------+
| City       | Country |
+------------+---------+
| Cowes      | UK      |
| London     | UK      |
| Londona    | UK      |
| Manchester | UK      |
+------------+---------+
4 rows in set (0.02 sec)
```

10. Select products (their names) and prices records that have an above average price, but cheaper than 33, sort ascending.

```
mysql> SELECT ProductName, Price
    -> FROM Products
    -> WHERE Price > (SELECT AVG(Price) FROM Products)
    -> AND Price < 33
    -> ORDER BY Price ASC;
+----------------------------------+-------+
| ProductName                      | Price |
+----------------------------------+-------+
| Uncle Bob's Organic Dried Pears  | 30.00 |
| Ikura                            | 31.00 |
| Gumbur Gummiburchen              | 31.23 |
| Mascarpone Fabioli               | 32.00 |
| Perth Pasties                    | 32.80 |
+----------------------------------+-------+
5 rows in set (0.02 sec)
```

***

11. Select shippers that shipped an above average (to column "Sender"), amount of orders they shipped (to column "Items Sent") and percentage of shipping from overall amount (to column "Quota", values have to be with '%' sign)

```
SELECT DISTINCT Shippers.ShipperName AS Sender, SUM(OrderDetails.Quantity) AS ItemsSent,
CONCAT(FORMAT((SUM(OrderDetails.Quantity) / (SELECT SUM(Quantity) FROM OrderDetails))*100, 2), " %") AS Quota
FROM Shippers JOIN Orders ON Shippers.ShipperID = Orders.ShipperID
                    JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID
GROUP BY Shippers.ShipperName
HAVING SUM(OrderDetails.Quantity) > (SELECT SUM(Quantity) FROM OrderDetails) / (SELECT Count(ShipperName)
FROM Shippers);
```

```
mysql> SELECT DISTINCT Shippers.ShipperName AS Sender, SUM(OrderDetails.Quantity) AS ItemsSent,
    -> CONCAT(FORMAT((SUM(OrderDetails.Quantity) / (SELECT SUM(Quantity) FROM OrderDetails))*100, 2), " %") AS Quota
    -> FROM Shippers JOIN Orders ON Shippers.ShipperID = Orders.ShipperID
    ->                          JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID
    -> GROUP BY Shippers.ShipperName
    -> HAVING SUM(OrderDetails.Quantity) > (SELECT SUM(Quantity) FROM OrderDetails) / (SELECT Count(ShipperName) FROM Shippers);
+-----------------+-----------+---------+
| Sender          | ItemsSent | Quota   |
+-----------------+-----------+---------+
| Federal Shipping |      4366 | 34.26 % |
| United Package   |      4802 | 37.68 % |
+-----------------+-----------+---------+
2 rows in set (0.02 sec)
```