1. Show the subject names of books supplied by *supplier2*

```
mysql> select Subjects.CategoryName
    -> from ((Subjects
    -> INNER JOIN Books ON Subjects.SubjectID = Books.SubjectID)
    -> INNER JOIN Suppliers on Books.SupplierID = Suppliers.SupplierID) WHERE Suppliers.CompanyName = 'supplier2';
+--------------+
| CategoryName |
+--------------+
| category1    |
| category2    |
+--------------+
2 rows in set (0.04 sec)
```

2. Show the name and price of the most expensive book supplied by *supplier3*.

```
mysql> select Books.Title, Books.UnitPrice
    -> from (Books
    -> INNER JOIN Suppliers ON Books.SupplierID = Suppliers.SupplierID)
    -> WHERE Suppliers.CompanyName = 'supplier3'
    -> GROUP BY Books.Title, Books.UnitPrice
    -> ORDER BY Books.UnitPrice DESC
    -> LIMIT 1;
+-------+-----------+
| Title | UnitPrice |
+-------+-----------+
| book7 |     56.90 |
+-------+-----------+
1 row in set (0.03 sec)
```

3. Show the unique names of all books ordered by *lastname1 firstname1*

```
mysql> SELECT DISTINCT Books.Title
    -> from (((Books
    -> INNER JOIN OrderDetails ON Books.BookID = OrderDetails.BookID)
    -> INNER JOIN Orders ON OrderDetails.OrderID = Orders.OrderID)
    -> INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID)
    -> WHERE
    -> Customers.LastName = 'lastname1' AND Customers.FirstName = 'firstname1';
+-------+
| Title |
+-------+
| book1 |
| book4 |
| book6 |
| book7 |
+-------+
4 rows in set (0.03 sec)
```

4. Show the title of books which have more than 10 units in stock.

```
mysql> select Title from Books
    -> where Quantity > 10;
+-------+
| Title |
+-------+
| book4 |
| book6 |
| book7 |
+-------+
3 rows in set (0.04 sec)
```

5. Show the total price *lastname1 firstname1* has paid for the books.

```
mysql> SELECT Customers.FirstName, Customers.LastName, SUM(Books.UnitPrice) AS TotalPaid
    -> from (((Books
    -> INNER JOIN OrderDetails ON Books.BookID = OrderDetails.BookID)
    -> INNER JOIN Orders ON OrderDetails.OrderID = Orders.OrderID)
    -> INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID)
    -> WHERE
    -> Customers.LastName = 'lastname1' AND Customers.FirstName = 'firstname1'
    -> GROUP BY Customers.FirstName, Customers.LastName;
+------------+-----------+-----------+
| FirstName  | LastName  | TotalPaid |
+------------+-----------+-----------+
| firstname1 | lastname1 |    128.48 |
+------------+-----------+-----------+
```

6. Show the names of the customers who have paid less than $80 in totals

```
mysql> SELECT
    -> Customers.FirstName, Customers.LastName, SUM(Books.UnitPrice * OrderDetails.Quantity) AS TotalOrder
    -> from (((Customers
    -> INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID)
    -> INNER JOIN OrderDetails ON OrderDetails.OrderID = Orders.OrderID)
    -> INNER JOIN Books ON OrderDetails.BookID = Books.BookID)
    -> GROUP BY Customers.FirstName, Customers.LastName
    -> HAVING TotalOrder < '80';
+------------+-----------+------------+
| FirstName  | LastName  | TotalOrder |
+------------+-----------+------------+
| firstname2 | lastname2 |      78.90 |
| firstname3 | lastname3 |      12.34 |
+------------+-----------+------------+
2 rows in set (0.04 sec)
```

7. Show the total price each customer paid and their names. List the result in descending price

```
mysql> SELECT
    -> Customers.FirstName, Customers.LastName, SUM(Books.UnitPrice * OrderDetails.Quantity) AS TotalOrder
    -> from (((Customers
    -> INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID)
    -> INNER JOIN OrderDetails ON OrderDetails.OrderID = Orders.OrderID)
    -> INNER JOIN Books ON OrderDetails.BookID = Books.BookID)
    -> GROUP BY Customers.FirstName, Customers.LastName
    -> ORDER BY TotalOrder DESC;
+------------+-----------+------------+
| FirstName  | LastName  | TotalOrder |
+------------+-----------+------------+
| firstname4 | lastname4 |     296.38 |
| firstname1 | lastname1 |     266.96 |
| firstname2 | lastname2 |      78.90 |
| firstname3 | lastname3 |      12.34 |
+------------+-----------+------------+
4 rows in set (0.04 sec)
```

8. Show the names of all the books shipped on 08/04/2016 and their shippers' names

```
mysql> Select Books.Title, Shippers.ShipperName
    -> from (((Books
    -> INNER JOIN OrderDetails ON Books.BookID = OrderDetails.BookID)
    -> INNER JOIN Orders ON OrderDetails.OrderID = Orders.OrderID)
    -> INNER JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID)
    -> WHERE
    -> Orders.ShippedDate = '2016-08-04 00:00:00' AND '2016-08-04 23:59:59';
+-------+-------------+
| Title | ShipperName |
+-------+-------------+
| book5 | shipper2    |
| book3 | shipper1    |
| book4 | shipper1    |
| book7 | shipper1    |
| book1 | shipper1    |
+-------+-------------+
5 rows in set, 1 warning (0.04 sec)
```

9. Show the names of all the ordered books and their total quantities. List the result in ascending quantity

```
mysql> SELECT Books.Title, SUM(OrderDetails.Quantity) AS TotalOrdered
    -> FROM (Books
    -> INNER JOIN OrderDetails ON Books.BookID = OrderDetails.BookID)
    -> GROUP BY Books.Title
    -> ORDER BY TotalOrdered ASC;
+-------+--------------+
| Title | TotalOrdered |
+-------+--------------+
| book5 |            1 |
| book6 |            2 |
| book3 |            2 |
| book4 |            2 |
| book7 |            4 |
| book1 |            6 |
+-------+--------------+
6 rows in set (0.03 sec)
```

10. Show the names of the customers who ordered at least 2 books

```
mysql> SELECT
    -> DISTINCT Customers.FirstName, Customers.LastName, SUM(OrderDetails.Quantity) AS TotalOrdered
    -> from (((Customers
    -> INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID)
    -> INNER JOIN OrderDetails ON OrderDetails.OrderID = Orders.OrderID)
    -> INNER JOIN Books ON OrderDetails.BookID = Books.BookID)
    -> GROUP BY Customers.FirstName, Customers.LastName
    -> HAVING TotalOrdered >= '2';
+------------+-----------+--------------+
| FirstName  | LastName  | TotalOrdered |
+------------+-----------+--------------+
| firstname1 | lastname1 |            9 |
| firstname4 | lastname4 |            6 |
+------------+-----------+--------------+
2 rows in set (0.04 sec)
```

11. Show the name of the customers who have ordered at least a book in *category3* or *category4* and the book names.

```
mysql> SELECT
    -> DISTINCT Customers.FirstName, Customers.LastName
    -> from ((((Customers
    -> INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID)
    -> INNER JOIN OrderDetails ON OrderDetails.OrderID = Orders.OrderID)
    -> INNER JOIN Books ON OrderDetails.BookID = Books.BookID)
    -> INNER JOIN Subjects ON Books.SubjectID = Subjects.SubjectID)
    -> WHERE Subjects.CategoryName = 'category3' OR Subjects.CategoryName = 'category4';
+------------+-----------+
| FirstName  | LastName  |
+------------+-----------+
| firstname1 | lastname1 |
| firstname4 | lastname4 |
+------------+-----------+
2 rows in set (0.04 sec)
```

12. Show the name of the customer who has ordered at least one book written by *author1*

```
mysql> SELECT
    -> DISTINCT Customers.FirstName, Customers.LastName
    -> from (((Customers
    -> INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID)
    -> INNER JOIN OrderDetails ON OrderDetails.OrderID = Orders.OrderID)
    -> INNER JOIN Books ON OrderDetails.BookID = Books.BookID)
    -> WHERE Books.Author = 'author1';
+------------+-----------+
| FirstName  | LastName  |
+------------+-----------+
| firstname1 | lastname1 |
| firstname4 | lastname4 |
| firstname3 | lastname3 |
+------------+-----------+
3 rows in set (0.04 sec)
```

13.Show the name and total sale (price of orders) of each employee

```
mysql> SELECT
    -> Employees.FirstName, Employees.LastName, SUM(Books.UnitPrice * OrderDetails.Quantity) AS TotalSales
    -> from (((Employees
    -> INNER JOIN Orders ON Employees.EmployeeID = Orders.EmployeeID)
    -> INNER JOIN OrderDetails ON OrderDetails.OrderID = Orders.OrderID)
    -> INNER JOIN Books ON OrderDetails.BookID = Books.BookID)
    -> GROUP BY Employees.FirstName, Employees.LastName
    -> Order BY  TotalSales DESC;
+------------+-----------+------------+
| FirstName  | LastName  | TotalSales |
+------------+-----------+------------+
| firstname6 | lastname6 |     491.76 |
| firstname5 | lastname5 |     162.82 |
+------------+-----------+------------+
2 rows in set (0.03 sec)
```

14. Show the names of customers who have ordered more than 1 book and the corresponding quantities. List the result in the descending quantity

```
mysql> SELECT
    -> Customers.FirstName, Customers.LastName, OrderDetails.Quantity
    -> from (((Customers
    -> INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID)
    -> INNER JOIN OrderDetails ON OrderDetails.OrderID = Orders.OrderID)
    -> INNER JOIN Books ON OrderDetails.BookID = Books.BookID)
    -> WHERE OrderDetails.Quantity > '1'
    -> GROUP BY Customers.FirstName, Customers.LastName, OrderDetails.Quantity
    -> Order BY  OrderDetails.Quantity DESC;
+------------+-----------+----------+
| FirstName  | LastName  | Quantity |
+------------+-----------+----------+
| firstname1 | lastname1 |        3 |
| firstname1 | lastname1 |        2 |
| firstname4 | lastname4 |        2 |
+------------+-----------+----------+
3 rows in set (0.03 sec)
```

15. Show the names of customers who have ordered more than 3 books and their respective telephone numbers

```
mysql> SELECT
    -> DISTINCT Customers.FirstName, Customers.LastName, Customers.Phone, SUM(OrderDetails.Quantity) AS Quantities
    -> from (((Customers
    -> INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID)
    -> INNER JOIN OrderDetails ON OrderDetails.OrderID = Orders.OrderID)
    -> INNER JOIN Books ON OrderDetails.BookID = Books.BookID)
    -> GROUP BY Customers.FirstName, Customers.LastName, Customers.Phone
    -> HAVING Quantities > '3'
    -> Order BY  Quantities DESC;
+------------+-----------+-------------+------------+
| FirstName  | LastName  | Phone       | Quantities |
+------------+-----------+-------------+------------+
| firstname1 | lastname1 | 334-001-001 |          9 |
| firstname4 | lastname4 | 334-004-004 |          6 |
+------------+-----------+-------------+------------+
2 rows in set (0.04 sec)
```