

--Query 1—

Find the all the **Customers** Details from the Customers table for Customers who have the letter 'j' anywhere within the CompanyName

```
SELECT *  
FROM Customers  
WHERE CompanyName LIKE '%j%'
```

3 rows returned

CustomerID	CompanyName	ContactName	City
	Ana Trujillo Emparedados y helados		
	Vaffeljernet		
	Wolski Zajazd		

+ the rest of the details

--Query 2—

Using the Products table find a list of all the products, which are currently out of stock, in your answer show the following details, ProductName, QuantityPerUnit and the UnitPrice of each Product. Order the results of your query in descending order based on the ProductName.

(5 row(s) affected)

```
SELECT ProductName, QuantityPerUnit, UnitPrice, UnitsInStock  
FROM Products  
WHERE UnitsInStock = 0  
ORDER BY ProductName DESC
```

ProductName	QuantityPerUnit	UnitPrice
Thüringer Rostbratwurst	50 bags x 30 sausgs.	123.79
Perth Pasties	48 pieces	32.80
Gorgonzola Telino	12 - 100 g pkgs	12.50
Chef Anton's Gumbo Mix	36 boxes	21.35
Alice Mutton	20 - 1 kg tins	39.00

(5 row(s) affected)

--QUERY 3--

Using the **Products** Table, show the ProductID and ProductName and the unitsInStock only for products which have at least 120 units in Stock but no more than 150 units in Stock.

(3 row(s) affected)

```
SELECT productID, ProductName, UnitsInstock  
FROM Products  
WHERE UnitsInstock BETWEEN 120 and 150
```

<u>ProductID</u>	<u>ProductName</u>	<u>unitsInStock</u>
6	Grandma's Boysenberry Spread	120
40	Boston Crab Meat	123
75	Rhönbräu Klosterbier	125

--QUERY 4--

You have been asked to find the CategoryID, CategoryName, Description and ProductName for products, which have a value of 'Dairy Products' for their CategoryName, you also need to rename the attribute CategoryName to that of 'Category of Product'

1. **SELECT with JOIN on common field btw Categories, Products**
2. **LIMIT to CategoryName = 'Dairy Products' using WHERE**
3. **ALIAS CategoryName AS 'Category of Product'**

```
SELECT c.categoryID, c.categoryName AS 'Category of Product',  
c.description, p.ProductName  
FROM Categories AS c, products AS p  
WHERE c.CategoryID =p.CategoryID AND c.CategoryName = 'Dairy  
Products'
```

(10 row(s) affected)

--Query 5--

Find the Maximum UnitsInStock of Products for **each** Supplier, keep in mind that we only want to see Suppliers who have a SupplierID greater than 22 in your result set, also include the SupplierID, Company Name and ProductName.
(15 row(s) affected)

- 1. SELECT with JOIN on common field btw Suppliers, Products**
- 2. LIMIT to SupplierID>22 using WHERE**
- 3. Use GROUP BY to Group by SupplierID & other columns selected**
- 4. Use MAX to return max value of UnitsInStock**

**SELECT Suppliers.SupplierID, Suppliers.CompanyName, ProductName,
MAX(Products.UnitsInStock) AS 'Max UnitsInStock'**

FROM Suppliers, Products

**WHERE Suppliers.SupplierID = Products.SupplierID AND
Suppliers.SupplierID >22**

GROUP BY Suppliers.SupplierID, Suppliers.CompanyName, ProductName

--QUERY 6--

Find all the Suppliers who have Products on order check to see if the level of the UnitsOnOrder when taken away from the UnitsInStock, causes the level of stock to drop below that of the ReorderLevel. Within your answer show the Company name, ProductName, SupplierID the stock level after the UnitsOnOrder have been taken away and the ReorderLevel of the Stock. (any product which has a ReorderLevel of 0) can be ignored. **(18 row(s) affected)**

1. **SELECT with JOIN on common field btw Suppliers, Products**
2. **LIMIT to Condition given (stock level=UnitsInStock-UnitsOnOrder < ReOrderLevel) using WHERE**
3. **LIMIT to Reorderlevel>0 using WHERE**
4. **ALIAS UnitsInStock-UnitsOnOrder AS 'LEVEL OF STOCK'**

```
SELECT ProductName, Suppliers.SupplierID, CompanyName, UnitsInStock-  
UnitsOnOrder AS 'LEVEL OF STOCK',ReorderLevel
```

```
FROM Suppliers, Products
```

```
WHERE Suppliers.SupplierID = Products.SupplierID AND UnitsInStock -  
UnitsOnOrder < ReorderLevel AND ReorderLevel > 0
```

Results:

ProductName	SupplierID	CompanyName	LEVEL OF STOCK	ReorderLevel
Chang	1	Exotic Liquids	-23	25
Aniseed Syrup	1	Exotic Liquids	-57	25
Queso Cabrales	5	Cooperativa de +	-8	30
Sir Rodney's Scones	8	Specialty Biscuits, Ltd	-37	5

(18 row(s) affected)

--QUERY 7--

Find the **Monetary Value of all discontinued stock**, which is presently still in stock. In your result set show the ProductID, the ProductName, the value of discontinued stock and the CompanyName, **compute** the total value of the Discontinued Stock

(hint discontinued stock has a value of 1)

Note: In this question, you will have to deduce that the Monetary Value of stuff still in stock = UnitsInStock*UnitPrice

1. **SELECT with JOIN on common field btw Suppliers, Products**
2. **LIMIT to Conditions given (Discontinued=1 & UnitsInStock>0) using WHERE**
3. **ALIAS UnitsInStock*UnitPrice AS 'Value of Stock'**
4. **To compute the total value of discontinued stock use COMPUTE SUM(UnitPrice * UnitsInStock)**

SELECT Products.ProductID, Products.ProductName, UnitPrice * UnitsInStock AS 'Value of Stock', CompanyName

FROM Products, Suppliers

WHERE Products.SupplierID = Suppliers.SupplierID AND Discontinued = 1 AND UnitsInStock > 0

COMPUTE SUM(UnitPrice * UnitsInStock)

ProductID	ProductName	Value of Stock	CompanyName
9	Mishi Kobe Niku	2813.0000	Tokyo Traders
24	Guaraná Fantástica	90.0000	Refrescos
28	Rössle Sauerkraut	1185.6000	Plutzer
42	Singaporean Hokkien Fried Mee	364.0000	Leka Trading
		sum	
		=====	
		4452.6000	

NORTHWIND DATABASE

