

Business Data Analysis Project using SQL and Tableau

SQL

Analysing data using the Northwind Database in Microsoft SQL Server:

Products and Inventory Data Analysis

1. Find top 10 purchased products by name based on units sold (quantity) and arrange them in descending order.

```
select top 10 p.ProductName, sum(od.Quantity) as [No. of units sold]
from
[Order Details] od
inner join
Products p
on od.ProductID = p.ProductID
group by p.ProductName
order by [No. of units sold] desc
```

2. Find the product that has the second highest cost in the company. (N-1)

```
SELECT ProductName, UnitPrice FROM Products p1
WHERE 1 = (SELECT COUNT(DISTINCT UnitPrice)
FROM Products p2
WHERE p2.UnitPrice > p1.UnitPrice)
```

3. Find the rank of sold products in each city in the USA based on quantity. (DENSE_RANK())

```
SELECT p.ProductName, c.City, od.quantity,
DENSE_RANK() OVER (PARTITION BY c.Country ORDER BY od.quantity DESC) AS Rank
FROM
Customers c
INNER JOIN Orders o
ON c.CustomerID = o.CustomerID
INNER JOIN
[Order Details] od
ON o.OrderID = od.OrderID
INNER JOIN
Products p
ON od.ProductID = p.ProductID
WHERE Country = 'USA'
ORDER BY od.Quantity DESC
```

4. Find all orders in the Order table that took more than 2 days to ship after the order date, and the number of days after the order date with the customer ID and Country, where the total sale value is greater than 10000.

```
SELECT o.OrderID, o.CustomerID, o.OrderDate, o.ShippedDate, o.ShipCountry,
DATEDIFF(DAY, OrderDate, ShippedDate) AS Duration_to_Ship,
SUM(od.Quantity * od.UnitPrice) AS [Total Sale Amount]
FROM
Orders o
INNER JOIN
[Order Details] od
ON o.OrderID = od.OrderID
WHERE
DATEDIFF(DAY, OrderDate, ShippedDate) > 2
GROUP BY o.OrderID, CustomerID, OrderDate, ShippedDate, ShipCountry
HAVING SUM(od.Quantity * od.UnitPrice) > 10000
```

ORDER BY DATEDIFF(DAY, OrderDate, ShippedDate) DESC

5. Create product stock status using CASE Statement.

```
SELECT productid, ProductName,
CASE
WHEN (UnitsInStock < UnitsOnOrder and
Discontinued = 0)
THEN 'Negative Inventory - Order Now!'
WHEN ((UnitsInStock - UnitsOnOrder) <
ReorderLevel and Discontinued = 0)
THEN 'Reorder Level Reached - Place Order'
WHEN (Discontinued = 1)
THEN '***Discontinued***'
ELSE 'In Stock'
END AS [Stock Status]
FROM products
```

Sales and Customers Data Analysis

1. Find the top ten customers who contributed to the highest sales with their cities and countries for the year 2018.

```
SELECT TOP 10 c.CompanyName, c.City, c.Country,
SUM(od.Quantity * od.UnitPrice) AS Total
FROM Customers c
INNER JOIN Orders o
ON c.CustomerID = o.CustomerID
INNER JOIN
[Order Details] od
ON o.OrderID = od.OrderID
WHERE YEAR(o.OrderDate) = '2018'
GROUP BY c.CompanyName, c.City, c.Country
ORDER BY Total DESC
```

2. Prepare a list of products that generate total sale greater than or equal to 30000 and number of units sold for each product for the year 2018.

```
SELECT p.ProductName, SUM(od.Quantity) as [Number of Unites],
SUM(od.Quantity * od.UnitPrice) as [Total Sale Amount]
FROM Orders o
INNER JOIN
[Order Details] od
ON o.OrderID = od.OrderID
INNER JOIN
Products p
ON od.ProductID = p.ProductID
WHERE YEAR(o.orderdate) = '2018'
GROUP BY p.ProductName
HAVING SUM(od.Quantity * od.UnitPrice) >= 30000
ORDER BY [Total Sale Amount] DESC
```

3. Classify customers from A to C based on their sales volumes as follows:
When total sale is greater than or equal to 30000, then grade is A; less than 30000 and greater or equal than 20000, then grade is B; and less than 20000, grade is C.

```
SELECT c.CompanyName,
SUM(od.Quantity * od.UnitPrice) AS Total,
CASE
WHEN SUM(od.Quantity * od.UnitPrice) >= 30000 THEN 'A'
```

```

WHEN SUM(od.Quantity * od.UnitPrice) < 30000 and sum(od.Quantity * od.UnitPrice) >=
20000
THEN 'B'
ELSE 'C'
END AS Customer_Grade
FROM Customers c
INNER JOIN Orders o
ON c.CustomerID = o.CustomerID
inner join
[Order Details] od
ON o.OrderID = od.OrderID
GROUP BY c.CompanyName
ORDER BY Total DESC

```

4. Find customers that generated total sale amount more than the average sale volume in the company for the year 2018.

```

SELECT c.CompanyName, c.City, c.Country,
SUM(od.Quantity * od.UnitPrice) AS Total
FROM Customers c
INNER JOIN Orders o
ON c.CustomerID = o.CustomerID
INNER JOIN
[Order Details] od
ON o.OrderID = od.OrderID
WHERE YEAR (o.OrderDate) = '2018'
GROUP BY c.CompanyName, c.City, c.Country
HAVING
SUM(od.Quantity * od.UnitPrice) >=
(SELECT AVG(Quantity * UnitPrice) FROM [Order Details])
ORDER BY Total DESC

```

5. Create the customer list and find the sale volume per customer for each year.

```

IF OBJECT_ID('dbo.Sale_Year') IS NOT NULL
DROP VIEW Sale_Year
GO
CREATE VIEW Sale_Year AS
(
SELECT c.CompanyName AS [Customer Name], YEAR(o.orderdate) AS Year, (od.unitprice *
od.Quantity) AS Sale
FROM
Customers c
INNER JOIN Orders o
ON c.CustomerID = o.CustomerID
INNER JOIN
[Order Details] od
ON o.OrderID = od.OrderID
)
SELECT * FROM Sale_year
PIVOT (SUM(sale) for Year in ([2016],[2017],[2018])) AS SumSalesPerYear
ORDER BY [Customer Name]

```

6. Find customers who did not buy from the company for more than 20 months.

```

SELECT c.CompanyName, MAX(o.OrderDate) AS [Last Order Date],
DATEDIFF(MONTH, MAX(o.OrderDate), GETDATE()) AS [Months Since Last Order]
FROM Customers AS c
INNER JOIN Orders AS o ON c.CustomerID = o.CustomerID
GROUP BY c.CompanyName
HAVING DATEDIFF(MONTH, MAX(o.OrderDate), GETDATE()) > 20
ORDER BY [Months Since Last Order] DESC

```

7. Find the customers list and the number of orders per customer.

```
SELECT c.CompanyName, c.City ,
(SELECT COUNT(OrderID) FROM Orders o
WHERE c.CustomerID = o.CustomerID ) AS [Number Of Orders] FROM
Customers c
ORDER BY [Number Of Orders] DESC
```

8. Find the customer with the 3rd highest sale volume.

```
IF OBJECT_ID('[dbo.Customer Sale]') IS NOT NULL
DROP VIEW [dbo.Customer Sale]
GO
CREATE VIEW [Customer Sale] AS
(
SELECT c.CompanyName AS [Customer Name], c.Country, SUM(od.unitprice * od.Quantity)
AS Sale
FROM
Customers c
INNER JOIN Orders o
ON c.CustomerID = o.CustomerID
INNER JOIN
[Order Details] od
ON o.OrderID = od.OrderID
GROUP BY c.CompanyName, c.Country
)
SELECT [Customer Name], Country, Sale FROM [Customer Sale] cs1
WHERE 2 = (SELECT COUNT(DISTINCT Sale)
FROM [Customer Sale] cs2
WHERE cs2.sale > cs1.sale)
```

9. Find the duration in days between two orders.

```
SELECT a.CustomerID, a.OrderDate,
DATEDIFF(DAY, a.OrderDate, b.OrderDate) AS [Days between two orders]
FROM Orders a
INNER JOIN
Orders b
ON a.OrderID = b.OrderID - 1
ORDER BY a.OrderDate
```

10. Prepare a sales data analysis over a particular time period.

```
IF OBJECT_ID('[Customer Sale]') IS NOT NULL
DROP VIEW [Customer Sale]
GO
CREATE VIEW [Customer Sale] AS
(
SELECT c.CompanyName AS [Customer Name], c.Country, YEAR(o.OrderDate) AS Year, MONTH
(o.OrderDate) AS Month,
SUM(od.unitprice * od.Quantity)
AS Sale
FROM
Customers c
INNER JOIN Orders o
ON c.CustomerID = o.CustomerID
INNER JOIN
[Order Details] od
ON o.OrderID = od.OrderID
GROUP BY c.CompanyName, c.Country, YEAR(o.OrderDate), MONTH(o.OrderDate)
```

```

)
GO
SELECT SUM
(SALE) AS SUM_SALE, MONTH, YEAR
FROM [Customer Sale]
WHERE YEAR = 2016
GROUP BY MONTH, YEAR
ORDER BY MONTH, SUM_SALE DESC
SELECT SUM
(SALE) AS SUM_SALE, MONTH, YEAR
FROM [Customer Sale]
WHERE YEAR = 2017 AND MONTH IN (1,2,3,4,5)
GROUP BY MONTH, YEAR
ORDER BY MONTH, SUM_SALE DESC
SELECT SUM
(SALE) AS SUM_SALE, MONTH, YEAR
FROM [Customer Sale]
WHERE YEAR = 2018
GROUP BY MONTH, YEAR
ORDER BY MONTH, SUM_SALE DESC

```

Employees and Staff Data Analysis

1. Find employees who achieved the top 3 highest sale volume, their sale volumes, cities, and bonus (0.02 of their total sale) for Jan (2018). Concatenate the employees' first and last names.

```

SELECT TOP 3 e.FirstName + ' ' + e.LastName AS [Full Name], e.City,
SUM(od.Quantity * od.UnitPrice) as [Total Sale],
ROUND(SUM(od.Quantity * od.UnitPrice)* .02, 0) AS [Bonus]
FROM
Employees e
INNER JOIN
Orders o
ON e.EmployeeID = o.EmployeeID
INNER JOIN
[Order Details] od
ON o.OrderID = od.OrderID
WHERE YEAR(o.OrderDate) = '2018' AND MONTH(o.OrderDate) = '1'
GROUP BY e.FirstName + ' ' + e.LastName, e.City
ORDER BY [Total Sale] DESC

```

2. Find the number of employees in each position or title for every city.

```

SELECT Title, City, COUNT(Title) AS [Number of Employees] FROM Employees
GROUP BY Title, City

```

3. Find the names of employees and their employment duration in the company in every city.

```

SELECT Lastname, Firstname, Title, DATEDIFF(YEAR, HireDate, GETDATE()) AS [Work Years
in the Company]
FROM Employees
WHERE City = 'London'

```

4. Find the employees and their ages for those older than 70 years in every city.

```

SELECT Lastname, Firstname, Title, DATEDIFF(YEAR, BirthDate, GETDATE()) AS [Age]
FROM Employees
WHERE DATEDIFF(YEAR, BirthDate, GETDATE()) >= 70

```

Tableau

Visualising data using the Sales and Customer Database in Tableau Public:

1. Map showing sales volume in each country.
2. Bar chart showing sales volume in each country.
3. Bar chart showing sales volume in each country per year, adding a filter for years.
4. Line charts showing number of orders per date and sales volume per date.
5. Bar chart showing the quantity sold of each product from different categories.
6. Dashboard with above charts.
7. Story with above charts.