

AdventureWorks Database – 2019

Get the entity relationship diagram: [Click here](#)

1. Write a query that returns data from the Person.Address table in this format AddressLine1 (City PostalCode) from the Person.Address table.

```
SELECT CONCAT(AddressLine1, ' (' + City + ' ', PostalCode + ')') AS AddressLine1
FROM Person.Address
```

2. Write a query using the Production.Product table displaying the Product ID, color and name columns. If the color column contains a NULL, replace the color with No color.

```
SELECT CAST(ProductID AS nvarchar(10)) + ', ' + ISNULL(Color, 'No color') + ', ' +
Name AS 'ID & Color & Name'
FROM Production.Product
```

3. Modify the query written in Q2 so that the description of the product is returned formatted as Name: Color. Make sure that all rows displays a value even if the Color Value is missing.

```
SELECT Name + ': ' + ISNULL(Color, 'No color') AS 'Description of the Products'
FROM Production.Product
```

4. Write a query using the Production.Product table displaying a description with the ProductID : Name format

```
SELECT CONCAT(ProductID, ': ', Name) AS 'Description of the Products'
FROM Production.Product
```

5. Write a query that displays the first ten characters of the AddressLine1 Column in the Person.Address table.

```
SELECT LEFT(AddressLine1, 10) AS Firt_10_characters
FROM Person.Address
```

6. Write a query that displays character 10 to 15 of the AddressLine1 column in the Person.Address table.

```
SELECT SUBSTRING(AddressLine1, 10, 5) AS 'Characters 10 to 15'
FROM Person.Address
```

7. Write a query displaying the first and last names from the Person.Person table all in Uppercase.

```
SELECT UPPER(FirstName) AS 'Upper Firstname', UPPER(LastName) AS 'Upper Lastname'
FROM Person.Person
```

8. The ProductNumber in the Production.Product table contains a hyphen (-). Write a query that uses the SUBSTRING function and CHARINDEX function to display the characters in the product number following the hyphen. Note: There is also a second hyphen in many of the rows, ignore the second hyphen for this question.

```
SELECT ProductNumber,
       CASE
         WHEN ProductNumber LIKE '%-%-%'
           THEN SUBSTRING(ProductNumber, CHARINDEX('-', ProductNumber) + 1,
                           CHARINDEX('-', ProductNumber, CHARINDEX('-', ProductNumber) + 1) - CHARINDEX('-', ProductNumber) - 1)
         WHEN ProductNumber LIKE '%-%'
           THEN SUBSTRING(ProductNumber, CHARINDEX('-', ProductNumber) + 1, 5)
       END AS CODE_number
FROM Production.Product
```

9. Write a query using the Sales.SalesOrderHeader table that displays the Subtotal rounded to two decimal places. Include the SalesOrderID column in the results.

```
SELECT SalesOrderID, ROUND(SubTotal, 2)
FROM Sales.SalesOrderHeader
```

10. Modify the query from Q9 so that the Subtotal is rounded to the nearest dollar but still displays two zeros to the right of the decimal place.

```
SELECT SalesOrderID, ROUND(SubTotal, 0)
FROM Sales.SalesOrderHeader
```

11. Write a query that calculates the square root of the SalesOrderID value from the Sales.SalesOrderHeader table.

```
SELECT SalesOrderID, SQRT(SalesOrderID) AS SquareRoot
FROM Sales.SalesOrderHeader;
```

12. Write a query using Sales.SpecialOffer table. Display the difference between the MinQty and MaxQty columns along with the SpecialOfferID and Description columns.

```
SELECT SpecialOfferID, Description, (MaxQty - MinQty) AS 'The difference between
MaxQty & MinQty'
FROM Sales.SpecialOffer
```

13. Write a query using the Sales.SpecialOffer table. Multiply the MinQty column by the DiscountPct column. Include the SpecialOfferID and Description columns in the results.

```
SELECT SpecialOfferID, Description, (MinQty * DiscountPct) AS MinDiscount
FROM Sales.SpecialOffer
```

14. Write a query using the Sales.SpecialOffer table that multiplies the MaxQty column by the DiscountPct column. If the MaxQty value is NULL, replace it with the value 10.

Include the SpecialOfferID and Description columns in the results.

```
SELECT SpecialOfferID, [Description],
       CASE
         WHEN MaxQty IS NULL THEN 10 * DiscountPct
         ELSE MaxQty * DiscountPct END AS MaxDiscount
FROM Sales.SpecialOffer;
```

```
SELECT SpecialOfferID, [Description], ISNULL(MaxQty, 10) * DiscountPct AS MaxDiscount
FROM Sales.SpecialOffer
```

15. Write a query that calculates the number of days between the date an order was placed and the date that it was shipped using Sales.SalesOrderHeader table.

Include the SalesOrderID, Orderdate, ShipDate columns.

```
SELECT SalesOrderID, OrderDate, ShipDate, DATEDIFF(day, OrderDate, ShipDate ) AS
periodtime
FROM Sales.SalesOrderHeader
```

16. Write a query that displays only the date, not the time, for the order date and ship date in Sales.SalesOrderHeader table.

```
SELECT SalesOrderID,
       FORMAT(OrderDate, 'yyyy-M-d') AS OrderDate,
       FORMAT(ShipDate, 'yyyy-M-d') AS ShipDate
FROM Sales.SalesOrderHeader
```

17. Write a query that adds six months to each order date in the Sales.SalesOrderHeader table. Include the SalesOrderID and OrderDate columns.

```
SELECT SalesOrderID,
       OrderDate,
       DATEADD(month, 6, OrderDate) AS SixmoreMonths
FROM Sales.SalesOrderHeader
```

18. Write a query that displays the year of each OrderDate and the numeric month of each OrderDate in separate columns in the results. Include the SalesOrderID and OrderDate columns.

```
SELECT SalesOrderID,
       OrderDate,
       DATENAME(year, OrderDate) AS YearOrder,
       DATEPART(month, OrderDate) AS MonthOrder
FROM Sales.SalesOrderHeader;
```

19. Change the query written in Q18 to display the month name instead.

```
SELECT SalesOrderID,
       OrderDate,
       DATENAME(year, OrderDate) AS YearOrder,
       DATENAME(month, OrderDate) AS MonthOrder
FROM Sales.SalesOrderHeader
```

20. Write a SELECT statement that return the date five quarters in the past from today's date.

```
SELECT DATEADD(QUARTER, -5, GETDATE()) AS 'The past 5 quarters from today's date'
```

21. Write a query using the HumanResources.Employee table to display the BusinessEntityID column. Also include a CASE expression that displays Even When the BusinessEntityID value is an even number or Odd when it is odd.

```
SELECT BusinessEntityID,
       CASE
           WHEN BusinessEntityID % 2 = 0 THEN 'Even'
           Else 'Odd' END AS IDtype
FROM HumanResources.Employee
```

22. Write a query using the Sales.SalesOrderDetail table to display a value (Under 10 or 10-19 or 30-39 or 40 and over) Based on the OrderQty value by using the CASE expression. Include the SalesOrderID and OrderQty columns in the results.

```
SELECT SalesOrderID, OrderQty,
       CASE
           WHEN OrderQty < 10 THEN 'Under 10'
           WHEN OrderQty <= 19 AND OrderQty >= 10 THEN '10-19'
           WHEN OrderQty <= 39 AND OrderQty >= 30 THEN '30-39'
           WHEN OrderQty >= 40 THEN '40 and over'
           END AS QuantityRange
FROM Sales.SalesOrderDetail
ORDER BY OrderQty DESC
```

23. Using the Person.Person table, build the full names using the Title, FirstName, MiddleName, LastName and Suffix columns. Check the table definition to see which columns allow NULL values and use the COALESCE function on Appropriate columns.

```
SELECT COALESCE(Title, '') + FirstName + ' ' + COALESCE(MiddleName + ' ', '') + LastName
+ COALESCE(' ' + Suffix, '') AS FULLNAME
FROM Person.Person
```

24. Look up the SERVERPROPERTY function in SQL Server's online documentation. Write a statement that displays the edition, instance name and machine name using this function.

```
SELECT SERVERPROPERTY('Edition') AS 'Edition' ,  
        SERVERPROPERTY('InstanceName') AS 'Instance Name',  
        SERVERPROPERTY('MachineName') AS 'Machine Name'
```

25. Write a query using the Sales.SalesOrderHeader to display the orders placed during 2021 by using a function. Include SalesOrderID and OrderDate columns in the results.

```
SELECT SalesOrderID, OrderDate  
FROM Sales.SalesOrderHeader  
WHERE YEAR(OrderDate) = 2021
```

26. Write a query using Sales.SalesOrderHeader table listing the sales in order of month the Order was placed and then the year the Order was placed. Include SalesOrderID, OrderDate columns in the results.

```
SELECT SalesOrderID, OrderDate  
FROM Sales.SalesOrderHeader  
ORDER BY MONTH(OrderDate), YEAR(OrderDate)
```

27. Write a query that displays the PersonType and the name columns from the Person.Person table. Sort the results so that rows with a PersonType of IN, SP or SC sort by LastName. The other rows should sort by FirstName.

```
SELECT FirstName, MiddleName, LastName, PersonType  
FROM Person.Person  
ORDER BY  
    CASE  
        WHEN PersonType IN ('IN', 'SP', 'SC') THEN LastName  
        ELSE FirstName END;
```