QA June Cohort

Michael Biney

MSQL Query 3 Assignment

Question 1

Retrieve Information about the products with color values except null, red, silver/black, white and list price between £75 and £750. Rename the column StandardCost to Price. Also, sort the results in descending order by list price.

SELECT

ProductID,

Name,

Color,

ListPrice AS Price

FROM

AdventureWorks2022.Production.Product

WHERE

Color NOT IN ('null', 'red', 'silver/black', 'white')

AND ListPrice BETWEEN 75 AND 750

ORDER BY

ListPrice DESC;

Question 2

Find all the male employees born between 1962 to 1970 and with hire date greater than 2001 and female employees born between 1972 and 1975 and hire date between 2001 and 2002.

select\*

from HumanResources.Employee

SELECT \*

FROM HumanResources.Employee

WHERE (Gender = 'M' AND YEAR(BirthDate) BETWEEN 1962 AND 1970 AND YEAR(HireDate) > 2001)

OR (Gender = 'F' AND YEAR(BirthDate) BETWEEN 1972 AND 1975 AND YEAR(HireDate) BETWEEN 2001 AND 2002);

Question 3

Create a list of 10 most expensive products that have a product number beginning with ‘BK’.

SELECT ProductID, Name, Color, ProductNumber

FROM Production.Product

WHERE ProductNumber LIKE 'BK%';

Question 4

Create a list of all contact persons, where the first 4 characters of the last name are the same as the first four characters of the email address.

Also, for all contacts whose first name and the last name begin with the same characters, create a new column called full name combining first name and the last name only.

Also provide the length of the new column full name.

select \*

from Person.EmailAddress

select \*

from Person.Person

SELECT

CONCAT(Person.FirstName, ' ', Person.LastName) AS FullName,

LEN(CONCAT(Person.FirstName, ' ', Person.LastName)) AS FullNameLength,

EmailAddress.EmailAddress,

Person.LastName,

Person.BusinessEntityID

FROM

Person.Person

JOIN

Person.EmailAddress ON SUBSTRING(Person.LastName, 1, 4) = SUBSTRING(EmailAddress.EmailAddress, 1, 4)

WHERE

SUBSTRING(Person.FirstName, 1, 4) = SUBSTRING(Person.LastName, 1, 4);

Question 5

Return all product subcategories that take an average of 3 days or longer to manufacture.

select \*

from Production.ProductSubcategory

SELECT

SC.Name AS SubcategoryName

FROM

Production.ProductSubcategory AS SC

JOIN

Production.Product AS P ON SC.ProductSubcategoryID = P.ProductSubcategoryID

JOIN

Production.ProductCostHistory AS CH ON P.ProductID = CH.ProductID

GROUP BY

SC.Name

HAVING

AVG(DATEDIFF(day, CH.StartDate, CH.EndDate)) >= 3;

Question 6

Create a list of product segmentation by defining criteria that places each item in a predefined segment as follows. If price gets less than £200 then low value. If price is between £201 and £750 then mid value. If between £750 and £1250 then mid to high value else higher value. Filter the results only for black, silver and red color products.

select \*

from Production.Product

SELECT

Name,

color,

CASE

WHEN ListPrice < 200 THEN 'Low Value'

WHEN ListPrice >= 201 AND ListPrice <= 750 THEN 'Mid Value'

WHEN ListPrice > 750 AND ListPrice <= 1250 THEN 'Mid to High Value'

ELSE 'Higher Value'

END AS segmentation

FROM

Production.Product

WHERE

color IN ('black', 'silver', 'red')

ORDER BY

Name;

Question 7

How many Distinct Job title is present in the Employee table?

select\*

from HumanResources.Employee

SELECT COUNT(DISTINCT JobTitle) AS num\_distinct\_job\_titles

FROM HumanResources.Employee;

Question 8

Use employee table and calculate the ages of each employee at the time of hiring.

select\*

from HumanResources.Employee

SELECT BusinessEntityID, NationalIDNumber, JobTitle, HireDate, BirthDate,

DATEDIFF(YEAR, BirthDate, HireDate) AS age\_at\_hiring

FROM HumanResources.Employee;

Question 9

How many employees will be due a long service award in the next 5 years, if long service is 20 years?

select\*

from HumanResources.Employee

SELECT COUNT(\*) AS num\_employees\_due\_award

FROM HumanResources.Employee

WHERE DATEDIFF(YEAR, HireDate, GETDATE()) >= 20

AND DATEDIFF(YEAR, HireDate, GETDATE()) < 25;

Question 10

How many more years does each employee have to work before reaching sentiment, if sentiment age is 65?

select\*

from HumanResources.Employee

SELECT BusinessEntityID, NationalIDNumber, DATEDIFF(YEAR, BirthDate, GETDATE()) AS Age,

(65 - DATEDIFF(YEAR, BirthDate, GETDATE())) AS Years\_To\_Retirement

FROM HumanResources.Employee

order by Age Desc;

Question 11

Implement new price policy on the product table base on the color of the item If white increase price by 8%, If yellow reduce price by 7.5%, If black increase price by 17.2%.

If multi, silver, silver/black or blue take the square root of the price and double the value. Column should be called Newprice. For each item, also calculate commission as 37.5% of newly computed list price.

select \*

from Production.Product

ALTER TABLE Production.Product

ADD NewPrice DECIMAL(18, 2),

Commission DECIMAL(18, 2)

UPDATE Production.Product

SET NewPrice = CASE

WHEN Color = 'White' THEN ListPrice \* 1.08 -- Increase price by 8% for White items

WHEN Color = 'Yellow' THEN ListPrice \* 0.925 -- Reduce price by 7.5% for Yellow items

WHEN Color = 'Black' THEN ListPrice \* 1.172 -- Increase price by 17.2% for Black items

WHEN Color IN ('Multi', 'Silver', 'Silver/Black', 'Blue') THEN SQRT(ListPrice) \* 2 -- Square root of price doubled for specified colors

ELSE ListPrice -- Keep original price for other colors

END,

Commission = CASE

WHEN Color = 'White' THEN (ListPrice \* 1.08) \* 0.375 -- Commission for White items

WHEN Color = 'Yellow' THEN (ListPrice \* 0.925) \* 0.375 -- Commission for Yellow items

WHEN Color = 'Black' THEN (ListPrice \* 1.172) \* 0.375 -- Commission for Black items

WHEN Color IN ('Multi', 'Silver', 'Silver/Black', 'Blue') THEN (SQRT(ListPrice) \* 2) \* 0.375 -- Commission for specified colors

ELSE ListPrice \* 0.375 -- Commission for other colors

END;

Question 12

Print the information about all the Sales.Person and their sales quota. For every Salesperson you should provide their FirstName, LastName, HireDate, SickLeaveHours and Region where they work.

SELECT

p.FirstName,

p.LastName,

p.HireDate,

p.SickLeaveHours,

r.Region

FROM

Sales.SalesPerson p

INNER JOIN HumanResources.Employee e ON p.BusinessEntityID = e.BusinessEntityID

INNER JOIN Person.Address a ON a.AddressID = e.AddressID

INNER JOIN Person.StateProvince s ON s.StateProvinceID = a.StateProvinceID

INNER JOIN Person.CountryRegion cr ON cr.CountryRegionCode = s.CountryRegionCode

INNER JOIN Sales.SalesTerritory st ON st.TerritoryID = e.TerritoryID

INNER JOIN Person.BusinessEntityAddress bea ON bea.BusinessEntityID = e.BusinessEntityID

INNER JOIN Person.AddressType at ON at.AddressTypeID = bea.AddressTypeID

INNER JOIN Person.BusinessEntity be ON be.BusinessEntityID = e.BusinessEntityID

INNER JOIN Sales.SalesPerson sp ON sp.BusinessEntityID = be.BusinessEntityID

INNER JOIN Sales.SalesTerritoryRegion str ON str.TerritoryID = st.TerritoryID

INNER JOIN Person.CountryRegionCurrency crc ON crc.CountryRegionCode = cr.CountryRegionCode

WHERE

at.Name = 'Home'

AND str.Region IS NOT NULL;

Question 13

Using adventure works, write a query to extract the following information.

• Product name

• Product category name

• Product subcategory name

• Sales person

• Revenue

• Month of transaction

• Quarter of transaction

• Region

select \*

from Sales.SalesOrderDetail

-- Add a new column called 'Revenue' to the 'Sales.SalesOrderDetail' table

ALTER TABLE Sales.SalesOrderDetail

ADD Revenue DECIMAL(18, 2);

-- Calculate and update the revenue in the 'Sales.SalesOrderDetail' table

UPDATE Sales.SalesOrderDetail

SET Revenue = UnitPrice \* OrderQty;

Sales.SalesOrderDetail, Person.CountryRegion

select \*

from Sales.SalesOrderDetail

SELECT

p.Name AS 'Product Name',

pc.Name AS 'Product Category',

psc.Name AS 'Product Subcategory',

CONCAT(ps.FirstName, ' ', ps.LastName) AS 'Sales Person',

sod.LineTotal AS 'Revenue',

MONTH(soh.OrderDate) AS 'Month of Transaction',

DATEPART(QUARTER, soh.OrderDate) AS 'Quarter of Transaction',

st.Name AS 'Region'

FROM

Sales.SalesOrderDetail sod

JOIN

Production.Product p ON sod.ProductID = p.ProductID

JOIN

Production.ProductSubcategory psc ON p.ProductSubcategoryID = psc.ProductSubcategoryID

JOIN

Production.ProductCategory pc ON psc.ProductCategoryID = pc.ProductCategoryID

JOIN

Sales.SalesOrderHeader soh ON sod.SalesOrderID = soh.SalesOrderID

JOIN

Sales.SalesTerritory st ON soh.TerritoryID = st.TerritoryID

JOIN

Sales.SalesPerson sp ON st.TerritoryID = sp.TerritoryID

JOIN

Person.Person ps ON sp.BusinessEntityID = ps.BusinessEntityID

JOIN

Person.StateProvince sp1 ON st.TerritoryID = sp1.TerritoryID

JOIN

Person.CountryRegion r ON sp1.CountryRegionCode = r.CountryRegionCode;

Question 14

Display the information about the details of an order i.e. order number, order date, amount of order, which customer gives the order and which salesman works for that customer and how much commission he gets for an order.

SELECT

o.OrderNumber,

o.OrderDate,

o.Amount,

c.CustomerName,

s.SalesmanName,

s.Commission

FROM

Sales.SalesOrderHeader, Production.Product

JOIN Customers c ON o.CustomerID = c.CustomerID

JOIN Salesmen s ON c.SalesmanID = s.SalesmanID;

Question 15

For all the products calculate

Commission as 14.790% of standard cost, Margin, if standard cost is increased or decreased as follows:

Black: +22%,

Red: -12%

Silver: +15%

Multi: +5%

White: Two times original cost divided by the square root of cost For other colours, standard cost remains the same

SELECT

Name,

StandardCost,

CASE

WHEN Name = 'Black' THEN StandardCost \* 1.22

WHEN Name = 'Red' THEN StandardCost \* 0.88

WHEN Name = 'Silver' THEN StandardCost \* 1.15

WHEN Name = 'Multi' THEN StandardCost \* 1.05

WHEN Name = 'White' THEN (2 \* StandardCost) / SQRT(StandardCost)

ELSE StandardCost

END AS AdjustedCost,

AdjustedCost \* 0.1479 AS Commission,

(AdjustedCost - StandardCost) / AdjustedCost AS Margin

FROM Production.Product;

Question 16

Create a view to find out the top 5 most expensive products for each colour.

CREATE VIEW TopExpensiveProductsByColor AS

SELECT

Color,

Name,

StandardCost

FROM

(

SELECT

Color,

Name,

StandardCost,

ROW\_NUMBER() OVER (PARTITION BY Color ORDER BY StandardCost DESC) AS RowNum

FROM

Production.Product

) AS RankedProducts

WHERE

RowNum <= 5;