***WORST - SIMPLE***

-------------------------------------------------------------------------------------------------------------------------------

--The number of productIds and the locationId being less than 50 with

--their listed price being in descending order greatest to least

--(354 rows affected)(Worse 1 Candidate)

SELECT c.ProductID

,C.LocationID

,d.ListPrice

FROM [Production].[ProductInventory] AS c

INNER JOIN [Production].[ProductListPriceHistory] AS d ON c.ProductID = d.ProductID

WHERE c.LocationID < 50

ORDER BY d.ListPrice DESC

FOR JSON PATH, root('ProductsInformation'), include\_null\_values;

***FIXED - Simple***

--(Fixed)

--1 The number of productIds with

--their listed price being in descending order greatest to least

--(583 rows affected)

--Note: Changed the locationid relevance

--it brought nothing new to the query as well as there was no specific

--reason behind having that constraint be there in the first place

SELECT c.ProductID

,C.LocationID

,d.ListPrice

FROM [Production].[ProductInventory] AS c

INNER JOIN [Production].[ProductListPriceHistory] AS d ON c.ProductID = d.ProductID

ORDER BY d.ListPrice DESC

FOR JSON PATH, root('ProductsInformation'), include\_null\_values;

-------------------------------------------------------------------------------------------------------------------------------

***WORST - Medium***

--12 What is the contact information of the customer in a country where there are no employees, --in alphabetical order of country?(71 rows affected)(Worse 2 Candidate)

SELECT c.CustomerCountry

,c.CustomerPhoneNumber

,c.CustomerFaxNumber

FROM Sales.Customer AS c

OUTER APPLY HumanResources.Employee

WHERE c.CustomerCountry NOT IN (

SELECT E.[EmployeeCountry]

FROM [HumanResources].[Employee] AS E

)

GROUP BY c.CustomerCountry

,c.CustomerPhoneNumber

,c.CustomerFaxNumber

ORDER BY c.CustomerCountry

FOR JSON PATH, root('CustomerinNOTEmployeeCountry'), include\_null\_values;

***FIXED - Medium***

--(Fixed)

--12 What is the contact information of the customer and the id and last name of the employee in a country in alphabetical order of country?(819 rows affected)(Worse 2 Candidate)

--Note: Was just a reversed version of the previous query so to change it up and make it make more sense,

--this time I made it be the contact information as well as the country in which there are orders,

--this query also includes the employeeid and employee last name for more needed information

USE Northwinds2020TSQLV6

SELECT c.CustomerCountry

,c.CustomerPhoneNumber

,c.CustomerFaxNumber

,e.EmployeeId

,e.EmployeeLastName

FROM Sales.Customer AS c

OUTER APPLY HumanResources.Employee AS e

GROUP BY c.CustomerCountry

,c.CustomerPhoneNumber

,c.CustomerFaxNumber

,e.EmployeeId

,e.EmployeeLastName

ORDER BY c.CustomerCountry

FOR JSON PATH, root('CustomerinNOTEmployeeCountry'), include\_null\_values;

-------------------------------------------------------------------------------------------------------------------------------

***WORST - Complex***

--15 2 There is a sugar shortage in your town and you must find out how to get sugar,

--identify which products are confections and which aren't

--and then which suppliers deliver those products and which do not ship confections,

--it is life or death(77 rows affected)(Worse 3 Candidate)

USE Northwinds2020TSQLV6

GO

DROP FUNCTION

IF EXISTS Production.Confection

GO

CREATE FUNCTION Production.Confection (

@SupplierName NVARCHAR(15)

,@Categoryid NVARCHAR(15)

)

RETURNS NVARCHAR(60)

AS

BEGIN

IF (

@Categoryid = 3

AND @SupplierName = 'Supplier BWGYE'

)

RETURN 'This is a confection product and confection supplier #1'

ELSE IF (

@Categoryid = 3

AND @SupplierName = 'Supplier ELCRN'

)

RETURN 'This is a confection product and confection supplier #3'

ELSE IF (

@Categoryid = 3

AND @SupplierName = 'Supplier FNUXM'

)

RETURN 'This is a confection product and confection supplier #4'

ELSE IF (

@Categoryid = 3

AND @SupplierName = 'Supplier GQRCV'

)

RETURN 'This is a confection product and confection supplier #5'

ELSE IF (

@Categoryid = 3

AND @SupplierName = 'Supplier OGLRK'

)

RETURN 'This is a confection product and confection supplier #6'

ELSE IF (

@Categoryid = 3

AND @SupplierName = 'Supplier ZPYVS'

)

RETURN 'This is a confection product and confection supplier #2'

ELSE IF (

@Categoryid <> 3

AND @SupplierName = 'Supplier BWGYE'

)

RETURN 'This is NOT a confection product but is confection supplier #1'

ELSE IF (

@Categoryid <> 3

AND @SupplierName = 'Supplier ELCRN'

)

RETURN 'This is NOT a confection product but is confection supplier #3'

ELSE IF (

@Categoryid <> 3

AND @SupplierName = 'Supplier FNUXM'

)

RETURN 'This is NOT confection product but is confection supplier #4'

ELSE IF (

@Categoryid <> 3

AND @SupplierName = 'Supplier GQRCV'

)

RETURN 'This is NOT confection product but is confection supplier #5'

ELSE IF (

@Categoryid <> 3

AND @SupplierName = 'Supplier OGLRK'

)

RETURN 'This is NOT confection product but is confection supplier #6'

ELSE IF (

@Categoryid <> 3

AND @SupplierName = 'Supplier ZPYVS'

)

RETURN 'This is NOT confection product but is confection supplier #2'

ELSE IF (

@Categoryid <> 3

AND @SupplierName <> 'Supplier BWGYE'

)

RETURN 'This is NOT a confection product nor a confection supplier'

ELSE IF (

@Categoryid <> 3

AND @SupplierName <> 'Supplier ELCRN'

)

RETURN 'This is NOT a confection product nor a confection supplier'

ELSE IF (

@Categoryid <> 3

AND @SupplierName <> 'Supplier FNUXM'

)

RETURN 'This is NOT a confection product nor a confection supplier'

ELSE IF (

@Categoryid <> 3

AND @SupplierName <> 'Supplier GQRCV'

)

RETURN 'This is NOT a confection product nor a confection supplier'

ELSE IF (

@Categoryid <> 3

AND @SupplierName <> 'Supplier OGLRK'

)

RETURN 'This is NOT a confection product nor a confection supplier'

ELSE IF (

@Categoryid <> 3

AND @SupplierName = 'Supplier ZPYVS'

)

RETURN 'This is NOT a confection product nor a confection supplier'

RETURN 'Done'

END;

GO

SELECT pc.CategoryId

,pc.CategoryName

,pp.ProductId

,ps.SupplierCompanyName

,Production.Confection(ps.SupplierCompanyName, pc.CategoryId) AS ConfectionSupplierInformation

FROM Production.Category AS pc

LEFT OUTER JOIN Production.Product AS pp ON pp.CategoryId = pc.CategoryId

LEFT OUTER JOIN Production.Supplier AS ps ON ps.SupplierId = pp.SupplierId

GROUP BY pc.CategoryId

,pc.CategoryName

,pp.ProductId

,ps.SupplierCompanyName

,Production.Confection(ps.SupplierCompanyName, pc.CategoryId)

ORDER BY ps.SupplierCompanyName

--FOR JSON PATH, root('ConfectionCrazedTown - ProductAndSupplier'), include\_null\_values;

***FIXED- Complex***

--(Fixed)

--15 2 There is a sugar shortage in your town and you must find out how to get sugar,

--identify which products are confections and which aren't

--and then which suppliers deliver those products and which do not ship confections,

--it is life or death(77 rows affected)

--Note: I had added another level of verification that was not necessary, it had to

--do with the understanding of the statements I had already suggested, pretty much

--I was unncessarily doubly checking if a supplier or category id was 3 or a supplier

--who supplies convections and then checking to make sure they were not the same categoryid

--nor a supplier who never supplied convections, by removing my last set of cases that had

--to do with <> categoryid 3 and <> a certain supplier, and replace the 'done' final return as

--'This is NOT a confection product nor a confection supplier', it properly filled all the same

--cases and allowed the answers to be exactly the same despite less lines of code

USE Northwinds2020TSQLV6

GO

DROP FUNCTION

IF EXISTS Production.Confection2

GO

CREATE FUNCTION Production.Confection2 (

@SupplierName NVARCHAR(15)

,@Categoryid NVARCHAR(15)

)

RETURNS NVARCHAR(60)

AS

BEGIN

IF (

@Categoryid = 3

AND @SupplierName = 'Supplier BWGYE'

)

RETURN 'This is a confection product and confection supplier #1'

ELSE IF (

@Categoryid = 3

AND @SupplierName = 'Supplier ELCRN'

)

RETURN 'This is a confection product and confection supplier #3'

ELSE IF (

@Categoryid = 3

AND @SupplierName = 'Supplier FNUXM'

)

RETURN 'This is a confection product and confection supplier #4'

ELSE IF (

@Categoryid = 3

AND @SupplierName = 'Supplier GQRCV'

)

RETURN 'This is a confection product and confection supplier #5'

ELSE IF (

@Categoryid = 3

AND @SupplierName = 'Supplier OGLRK'

)

RETURN 'This is a confection product and confection supplier #6'

ELSE IF (

@Categoryid = 3

AND @SupplierName = 'Supplier ZPYVS'

)

RETURN 'This is a confection product and confection supplier #2'

ELSE IF (

@Categoryid <> 3

AND @SupplierName = 'Supplier BWGYE'

)

RETURN 'This is NOT a confection product but is confection supplier #1'

ELSE IF (

@Categoryid <> 3

AND @SupplierName = 'Supplier ELCRN'

)

RETURN 'This is NOT a confection product but is confection supplier #3'

ELSE IF (

@Categoryid <> 3

AND @SupplierName = 'Supplier FNUXM'

)

RETURN 'This is NOT confection product but is confection supplier #4'

ELSE IF (

@Categoryid <> 3

AND @SupplierName = 'Supplier GQRCV'

)

RETURN 'This is NOT confection product but is confection supplier #5'

ELSE IF (

@Categoryid <> 3

AND @SupplierName = 'Supplier OGLRK'

)

RETURN 'This is NOT confection product but is confection supplier #6'

ELSE IF (

@Categoryid <> 3

AND @SupplierName = 'Supplier ZPYVS'

)

RETURN 'This is NOT confection product but is confection supplier #2'

RETURN 'This is NOT a confection product nor a confection supplier'

RETURN 'Done'

END;

GO

SELECT pc.CategoryId

,pc.CategoryName

,pp.ProductId

,ps.SupplierCompanyName

,Production.Confection2(ps.SupplierCompanyName, pc.CategoryId) AS ConfectionSupplierInformation

FROM Production.Category AS pc

LEFT OUTER JOIN Production.Product AS pp ON pp.CategoryId = pc.CategoryId

LEFT OUTER JOIN Production.Supplier AS ps ON ps.SupplierId = pp.SupplierId

GROUP BY pc.CategoryId

,pc.CategoryName

,pp.ProductId

,ps.SupplierCompanyName

,Production.Confection(ps.SupplierCompanyName, pc.CategoryId)

ORDER BY ps.SupplierCompanyName

FOR JSON PATH, root('ConfectionCrazedTown - ProductAndSupplier'), include\_null\_values;

-------------------------------------------------------------------------------------------------------------------------------

***BEST - Complex***

--18 5 Find the number of orders made in Decembers by the HighestLevelImportance in order by the orderdate(15 rows affected)(1 Best)

USE Northwinds2020TSQLV6

DROP FUNCTION

IF EXISTS dbo.CustomerTitleImportance

GO

CREATE FUNCTION dbo.CustomerTitleImportance (

@CustomerContactTitle NVARCHAR(30))

RETURNS NVARCHAR(40)

AS

BEGIN

IF (

@CustomerContactTitle = 'Owner'

OR @CustomerContactTitle = 'Order Administrator'

)

RETURN 'Highest Level Importance'

IF (

@CustomerContactTitle = 'Accounting Manager'

OR @CustomerContactTitle = 'Marketing Manager'

OR @CustomerContactTitle = 'Sales Manager'

)

RETURN 'Intermediate Level Importance'

IF (

@CustomerContactTitle = 'Sales Associate'

OR @CustomerContactTitle = 'Assistant Sales Agent'

OR @CustomerContactTitle = 'Marketing Assistant'

OR @CustomerContactTitle = 'Assistant Sales Representative'

OR @CustomerContactTitle = 'Sales Representative'

OR @CustomerContactTitle = 'Sales Agent'

)

RETURN 'Lower Level Importance'

RETURN 'Mixed Importance'

END

GO

SELECT so.OrderId

,so.OrderDate

,so.EmployeeId

,sc.CustomerContactTitle

,dbo.CustomerTitleImportance(sc.CustomerContactTitle) AS 'Relative Importance Level'

FROM Sales.[Order] AS so

INNER JOIN [HumanResources].[Employee] AS hr ON hr.EmployeeId = so.EmployeeId

INNER JOIN Sales.Customer AS sc ON sc.CustomerId = so.CustomerId

WHERE dbo.CustomerTitleImportance(sc.CustomerContactTitle) = 'Highest Level Importance'

AND MONTH(so.OrderDate) = 12

GROUP BY so.OrderId

,so.OrderDate

,so.EmployeeId

,sc.CustomerContactTitle

,dbo.CustomerTitleImportance(sc.CustomerContactTitle)

ORDER BY so.OrderDate

--FOR JSON PATH, root('BuyerImportanceGauger'), include\_null\_values;

-------------------------------------------------------------------------------------------------------------------------------

***BEST - Simple***

--+ 4 What is the amount of people in the Database of people who do not have a middle name

--AND the date it was modified not being null (8498 rows affected)(1 Best)

SELECT DISTINCT p.FirstName

,p.MiddleName

,p.LastName

,p.ModifiedDate

FROM [Person].[Person] AS p

FULL OUTER JOIN person.CountryRegion AS t ON t.ModifiedDate = p.ModifiedDate

WHERE p.MiddleName IS NULL

AND p.ModifiedDate IS NOT NULL

--FOR JSON PATH, root('NoMiddleNames'), include\_null\_values;

-------------------------------------------------------------------------------------------------------------------------------

***BEST - Medium***

USE Northwinds2020TSQLV6

--9 Which customers and which employees live in the same general area,

--within the same country,in the same region,in the same city?(1 Best)

SELECT c.CustomerId

,e.EmployeeId

,c.CustomerRegion

,e.EmployeeRegion

,e.EmployeeCountry

,c.CustomerCountry

,c.CustomerCity

,e.EmployeeCity

FROM [HumanResources].[Employee] AS e

INNER JOIN Sales.Customer AS c ON e.EmployeeCity = c.CustomerCity

INNER JOIN Sales.[Order] AS o ON o.ShipToCity = c.CustomerCity

WHERE c.CustomerRegion IS NOT NULL

AND e.EmployeeRegion IS NOT NULL

GROUP BY c.CustomerId

,e.EmployeeId

,c.CustomerRegion

,e.EmployeeRegion

,e.EmployeeCountry

,c.CustomerCountry

,c.CustomerCity

,e.EmployeeCity

FOR JSON PATH, root('CustomerEmployeeSameArea'), include\_null\_values;