**Create a query with the following columns:**

**1. PurchaseOrderID, from Purchasing.PurchaseOrderDetail**

**2. PurchaseOrderDetailID, from Purchasing.PurchaseOrderDetail**

**3. OrderQty, from Purchasing.PurchaseOrderDetail**

**4. UnitPrice, from Purchasing.PurchaseOrderDetail**

**5. LineTotal, from Purchasing.PurchaseOrderDetail**

**6. OrderDate, from Purchasing.PurchaseOrderHeader**

**7. A derived column, aliased as “OrderSizeCategory”, calculated via CASE logic as follows:**

**o When OrderQty > 500, the logic should return “Large”**

**o When OrderQty > 50 but <= 500, the logic should return “Medium”**

**o Otherwise, the logic should return “Small”**

**8. The “Name” field from Production.Product, aliased as “ProductName”**

**9. The “Name” field from Production.ProductSubcategory, aliased as “Subcategory”; if this value is**

**NULL, return the string “None” instead**

**10. The “Name” field from Production.ProductCategory, aliased as “Category”; if this value is NULL,**

**return the string “None” instead**

**Only return rows where the order date occurred in December of ANY year. The**

**MONTH function should provide a helpful shortcut here**

The Sales data in our AdventureWorks database is structured almost identically to our Purchasing data.

It is so similar, in fact, that we can actually align columns from several of the Sales and Purchasing tables

to create a unified dataset in which some rows pertain to Sales, and some to Purchasing. Note that we

are talking about combining data by columns rather than by rows here – think UNION.

So with that said, your second challenge is to enhance your query from Challenge 1 by “stacking” it with

the corresponding Sales data. That may seem daunting, but it is actually WAY easier than it sounds! It

turns out that our two Purchasing tables from the Exercise 1 query map to an equivalent Sales table:

• Purchasing.PurchaseOrderDetail maps to Sales.SalesOrderDetail

• Purchasing.PurchaseOrderHeader maps to Sales.SalesOrderHeader

Create a query with the following columns:

11. BusinessEntityID, from Person.Person

12. PersonType, from Person.Person

13. A derived column, aliased as “FullName”, that combines the first, last, and middle names from

Person.Person.

* There should be exactly one space between each of the names.
* If “MiddleName” is NULL and you try to “add” it to the other two names, the result will

be NULL, which isn’t what you want.

* You could use ISNULL to return an empty string if the middle name is NULL, but then

you’d end up with an extra space between first and last name – a space we would have

needed if we had a middle name to work with.

* So what we really need is to apply conditional, IF/THEN type logic; if middle name is

NULL, we just need a space between first name and last name. If not, then we need a

space, the middle name, and then another space. See if you can accomplish this with a

CASE statement.

14. The “AddressLine1” field from Person.Address; alias this as “Address”.

15. The “City” field from Person.Address

16. The “PostalCode” field from Person.Address

17. The “Name” field from Person.StateProvince; alias this as “State”.

18. The “Name” field from Person.CountryRegion; alias this as “Country”.

Only return rows where person type (from Person.Person) is “SP”, OR the postal code begins with a

“9” AND the postal code is exactly 5 characters long AND the country (i.e., “Name” from

Person.CountryRegion) is “United States”

Enhance your query from Exercise 3 as follows:

1. Join in the HumanResources.Employee table to Person.Person on BusinessEntityID. Note that

many people in the Person.Person table are not employees, and we don’t want to limit our

output to just employees, so choose your join type accordingly.

2. Add the “JobTitle” field from HumanResources.Employee to our output. If it is NULL (as it will be

for people in our Person.Person table who are not employees, return “None”.

3. Add a derived column, aliased as “JobCategory”, that returns different categories based on the

value in the “JobTitle” column as follows:

* If the job title contains the words “Manager”, “President”, or “Executive”, return

“Management”. Applying wildcards with LIKE could be a helpful approach here.

* If the job title contains the word “Engineer”, return “Engineering”.
* If the job title contains the word “Production”, return “Production”.
* If the job title contains the word “Marketing”, return “Marketing”.
* If the job title is NULL, return “NA”.
* If the job title is one of the following exact strings (NOT patterns), return “Human

Resources”: “Recruiter”, “Benefits Specialist”, OR “Human Resources Administrative

Assistant”. You could use a series of ORs here, but the IN keyword could be a nice

shortcut.

* As a default case when none of the other conditions are true, return “Other”.

Select the number of days remaining until the end of the current month; that is, the difference in days

between the current date and the last day of the current month.

Your solution should be dynamic: it should work no matter what day, month, or year you run it, which

means it needs to calculate the end of the current month based on the current date.

**(To practice)DW OLAP**

**Q1 . What is the total sales?**

SELECT SUM(SalesAmount) AS TotalSales

FROM dbo.FactInternetSales;

**Q2. What is the total profit?**

select sum(b.TotalProductCost) as tpcost, sum(b.SalesAmount) as tsales,

sum(b.SalesAmount-b.TotalProductCost) as tprofit

from dbo.FactInternetSales b;

or can be done in other way:

SELECT

SUM(SalesAmount) AS TotalSales,

SUM(TotalProductCost) AS TotalCost,

SUM(SalesAmount - TotalProductCost) AS TotalProfit

FROM

dbo.FactInternetSales;

**Q3. What is the total cost amount?**

select sum(a.TotalProductCost) as Tcost

from dbo.FactInternetSales a;

**Q4. What is the sales per year?**

select year(OrderDate) as odate ,sum(SalesAmount)

from dbo.FactInternetSales

group by year(OrderDate)

**q5.What is the average sales per customers?**

**Q6. What is the number of products in each category?**

**Q7. Top 10 Customers with the highest purchase**

**Q8. Top 10 Customers with the highest order**

**Q9. Top 10 Employees with the highest sale**

**Q10. Top 10 most sale products**

**Q11. What is the total customer?**

**Q12. What is the total transaction?**

**Q13. Distribution of order**

**Distribution of order is simply to see how customers are making orders.**

**Q14. Ranking customers by sales**

**SELECT TABLE\_NAME**

**FROM AdventureWorksDW2022.INFORMATION\_SCHEMA.TABLES**

**WHERE table\_type = 'BASE TABLE'**

**Answer**

**SELECT A.[PurchaseOrderID]**

**,A.[PurchaseOrderDetailID]**

**,A.[OrderQty]**

**,A.[UnitPrice]**

**,A.[LineTotal]**

**,B.[OrderDate]**

**,[OrderSizeCategory] =**

**CASE**

**WHEN A.[OrderQty] > 500 THEN 'Large'**

**WHEN A.[OrderQty] > 50 THEN 'Medium'**

**ELSE 'Small'**

**END**

**,[ProductName] = C.[Name]**

**,[Subcategory] = ISNULL(D.[Name], 'None')**

**,[Category] = ISNULL(E.[Name],'None')**

**FROM [AdventureWorks2019].[Purchasing].[PurchaseOrderDetail] A**

**JOIN [AdventureWorks2019].[Purchasing].[PurchaseOrderHeader] B**

**ON A.[PurchaseOrderID] = B.[PurchaseOrderID]**

**JOIN [AdventureWorks2019].[Production].[Product] C**

**ON A.[ProductID] = C.[ProductID]**

**LEFT JOIN [AdventureWorks2019].[Production].[ProductSubcategory] D**

**ON C.[ProductSubcategoryID] = d.[ProductSubcategoryID]**

**LEFT JOIN [AdventureWorks2019].[Production].[ProductCategory] E**

**ON D.[ProductCategoryID] = E.[ProductCategoryID]**

**WHERE MONTH(B.[OrderDate]) = 12**

**Q2.ANSWER**

**SELECT**

**[OrderType] = 'Sale'**

**,[OrderID] = A.[SalesOrderID]**

**,[OrderDetailID] = A.[SalesOrderDetailID]**

**,A.[OrderQty]**

**,A.[UnitPrice]**

**,A.[LineTotal]**

**,B.[OrderDate]**

**,[OrderSizeCategory] =**

**CASE**

**WHEN A.[OrderQty] > 500 THEN 'Large'**

**WHEN A.[OrderQty] > 50 THEN 'Medium'**

**ELSE 'Small'**

**END**

**,[ProductName] = C.[Name]**

**,[Subcategory] = ISNULL(D.[Name], 'None')**

**,[Category] = ISNULL(E.[Name],'None')**

**FROM [AdventureWorks2019].[Sales].[SalesOrderDetail] A**

**JOIN [AdventureWorks2019].[Sales].[SalesOrderHeader] B**

**ON A.[SalesOrderID] = B.[SalesOrderID]**

**JOIN [AdventureWorks2019].[Production].[Product] C**

**ON A.[ProductID] = C.[ProductID]**

**LEFT JOIN [AdventureWorks2019].[Production].[ProductSubcategory] D**

**ON C.[ProductSubcategoryID] = d.[ProductSubcategoryID]**

**LEFT JOIN [AdventureWorks2019].[Production].[ProductCategory] E**

**ON D.[ProductCategoryID] = E.[ProductCategoryID]**

**WHERE MONTH(B.[OrderDate]) = 12**

**UNION ALL**

**SELECT**

**[OrderType] = 'Purchase'**

**,A.[PurchaseOrderID]**

**,A.[PurchaseOrderDetailID]**

**,A.[OrderQty]**

**,A.[UnitPrice]**

**,A.[LineTotal]**

**,B.[OrderDate]**

**,[OrderSizeCategory] =**

**CASE**

**WHEN A.[OrderQty] > 500 THEN 'Large'**

**WHEN A.[OrderQty] > 50 THEN 'Medium'**

**ELSE 'Small'**

**END**

**,[ProductName] = C.[Name]**

**,[Subcategory] = ISNULL(D.[Name], 'None')**

**,[Category] = ISNULL(E.[Name],'None')**

**FROM [AdventureWorks2019].[Purchasing].[PurchaseOrderDetail] A**

**JOIN [AdventureWorks2019].[Purchasing].[PurchaseOrderHeader] B**

**ON A.[PurchaseOrderID] = B.[PurchaseOrderID]**

**JOIN [AdventureWorks2019].[Production].[Product] C**

**ON A.[ProductID] = C.[ProductID]**

**LEFT JOIN [AdventureWorks2019].[Production].[ProductSubcategory] D**

**ON C.[ProductSubcategoryID] = d.[ProductSubcategoryID]**

**LEFT JOIN [AdventureWorks2019].[Production].[ProductCategory] E**

**ON D.[ProductCategoryID] = E.[ProductCategoryID]**

**WHERE MONTH(B.[OrderDate]) = 12**

**ORDER BY [OrderDate] DESC**

**Q3 ANSWER**

**SELECT A.[BusinessEntityID]**

**,A.[PersonType]**

**,[FullName] =**

**A.[FirstName] + ' ' +**

**CASE WHEN A.[MiddleName] IS NULL THEN '' ELSE A.[MiddleName] + ' ' END +**

**A.[LastName]**

**,[Address] = C.[AddressLine1]**

**,C.[City]**

**,C.[PostalCode]**

**,[State] = D.[Name]**

**,[Country] = E.[Name]**

**FROM [AdventureWorks2019].[Person].[Person] A**

**JOIN [AdventureWorks2019].[Person].[BusinessEntityAddress] B**

**ON A.[BusinessEntityID] = B.[BusinessEntityID]**

**JOIN [AdventureWorks2019].[Person].[Address] C**

**ON B.[AddressID] = C.[AddressID]**

**JOIN [AdventureWorks2019].[Person].[StateProvince] D**

**ON C.[StateProvinceID] = D.[StateProvinceID]**

**JOIN [AdventureWorks2019].[Person].[CountryRegion] E**

**ON D.[CountryRegionCode] = E.[CountryRegionCode]**

**WHERE (LEFT(C.[PostalCode], 1) = '9' AND LEN(C.[PostalCode]) = 5 AND E.[Name] = 'United States')**

**OR A.[PersonType] = 'SP'**

**Q4.ANSWER**

**SELECT A.[BusinessEntityID]**

**,A.[PersonType]**

**,[FullName] =**

**A.[FirstName] + ' ' +**

**CASE WHEN A.[MiddleName] IS NULL THEN '' ELSE A.[MiddleName] + ' ' END +**

**A.[LastName]**

**,[Address] = C.[AddressLine1]**

**,C.[City]**

**,C.[PostalCode]**

**,[State] = D.[Name]**

**,[Country] = E.[Name]**

**,[JobTitle] = ISNULL(F.[JobTitle],'NA')**

**,[JobCategory] =**

**CASE**

**WHEN F.[JobTitle] LIKE '%Manager%' OR F.[JobTitle] LIKE '%President%' OR F.[JobTitle] LIKE '%Executive%' THEN 'Management'**

**WHEN F.[JobTitle] LIKE '%Engineer%' THEN 'Engineering'**

**WHEN F.[JobTitle] LIKE '%Production%' THEN 'Production'**

**WHEN F.[JobTitle] LIKE '%Marketing%' THEN 'Marketing'**

**WHEN F.[JobTitle] IS NULL THEN 'NA'**

**WHEN F.[JobTitle] IN('Recruiter', 'Benefits Specialist', 'Human Resources Administrative Assistant') THEN 'Human Resources'**

**ELSE 'Other'**

**END**

**FROM [AdventureWorks2019].[Person].[Person] A**

**JOIN [AdventureWorks2019].[Person].[BusinessEntityAddress] B**

**ON A.[BusinessEntityID] = B.[BusinessEntityID]**

**JOIN [AdventureWorks2019].[Person].[Address] C**

**ON B.[AddressID] = C.[AddressID]**

**JOIN [AdventureWorks2019].[Person].[StateProvince] D**

**ON C.[StateProvinceID] = D.[StateProvinceID]**

**JOIN [AdventureWorks2019].[Person].[CountryRegion] E**

**ON D.[CountryRegionCode] = E.[CountryRegionCode]**

**LEFT JOIN [AdventureWorks2019].[HumanResources].[Employee] F**

**ON A.[BusinessEntityID] = F.[BusinessEntityID]**

**WHERE (LEFT(C.[PostalCode], 1) = '9' AND LEN(C.[PostalCode]) = 5 AND E.[Name] = 'United States')**

**OR A.[PersonType] = 'SP'**

**Q5. ANSWER**

**SELECT [Days Until EOM] =**

**DATEDIFF(DAY, GETDATE(),**

**DATEADD(DAY, -1,**

**DATEADD(MONTH,1,**

**DATEFROMPARTS(YEAR(GETDATE()), MONTH(GETDATE()), 1))))**