

BI Assignment 3 – Task 3

3.1

The screenshot shows the SQL Server Enterprise Manager interface with an MDX query window and a results pane.

MDX Query Window:

```
SELECT
  [Measures].[Profit] ON COLUMNS,
  NON EMPTY ORDER(
    [Product].[TopCategory].Members,
    [Measures].[Profit],
    BDESC
  ) ON ROWS
FROM [bike_sales]
WHERE [OrderDate].[Year].[2021]
```

Axis #0: {[OrderDate].[Year].[2021]}

Axis #1: {[Measures].[Profit]}

Axis #2: {[Product.ProductCategory].[Bikes]}, {[Product.ProductCategory].[Accessories]}, {[Product.ProductCategory].[Clothing]}

Row #0: \$3,183,670.33
Row #1: \$217,753.85
Row #2: \$69,062.19

Execute

Results Pane:

3_1 3_2 3_3 3_4 3_5 3_6 3_7

1 • SELECT

```
2 dp.ProductTopCategoryName AS `Product Top Category`,
3 SUM(fi.OrderLineProfit) AS `Profit`
4 FROM Fact_InternetSales fi
5 JOIN Dim_Product dp ON fi.ProductKey = dp.ProductKey
6 JOIN Dim_Date dd ON fi.OrderDateKey = dd.DateKey
7 WHERE dd.CalendarYear = 2021
8 GROUP BY dp.ProductTopCategoryName
9 ORDER BY `Profit` DESC;
```

Result Grid:

Product Top Category	Profit
Bikes	3183670.3256
Accessories	217753.8488
Clothing	69062.1949

Form Editor

3.2

The screenshot shows the SQL Server Enterprise Manager interface with an MDX query window and a results pane.

MDX Query Window:

```
SELECT
  [Measures].[Revenue] ON COLUMNS,
  ORDER(
    [ShippedTo].[Country].Members,
    [Measures].[Revenue],
    BDESC
  ) ON ROWS
FROM [bike_sales]
WHERE [OrderDate].[Year].[2019]
```

Axis #0: {[OrderDate].[Year].[2019]}

Axis #1: {[Measures].[Revenue]}

Axis #2: {[ShippedTo.Territory].[Pacific].[Australia]}, {[ShippedTo.Territory].[North America].[USA]}, {[ShippedTo.Territory].[North America].[Canada]}, {[ShippedTo.Territory].[Europe].[United Kingdom]}, {[ShippedTo.Territory].[Europe].[Germany]}, {[ShippedTo.Territory].[Europe].[France]}

Row #0: \$2,124,783.18
Row #1: \$2,023,814.61
Row #2: \$587,476.48
Row #3: \$582,032.74
Row #4: \$550,070.74
Row #5: \$522,422.20

Execute

Results Pane:

3_1 3_2 3_3 3_4 3_5 3_6 3_7

1 • SELECT

```
2 d1.Country,
3 SUM(fi.OrderLineTotal) AS `Revenue`
4 FROM Fact_InternetSales fi
5 JOIN Dim_Location d1 ON fi.ShipToLocationKey = d1.LocationKey
6 JOIN Dim_Date dd ON fi.OrderDateKey = dd.DateKey
7 WHERE dd.CalendarYear = 2019
8 GROUP BY d1.Country
9 ORDER BY `Revenue` DESC;
```

Result Grid:

Country	Revenue
Australia	2124783.1831
USA	2023814.6111
Canada	587476.4783
United Kingdom	582032.7359
Germany	550070.7367
France	522422.2022

Form Editor

3.3

The screenshot shows a SQL query window with the following SQL code:

```
1 • SELECT
2     RANK() OVER (ORDER BY SUM(fi.OrderLineProfit) DESC) AS `Customer Rank`,
3     dc.FullName AS `Customer Name`,
4     SUM(fi.OrderLineProfit) AS `Profit`
5 FROM Fact_InternetSales fi
6 JOIN Dim_Customer dc ON fi.CustomerKey = dc.CustomerKey
7 JOIN Dim_Date dd ON fi.OrderDateKey = dd.DateKey
8 WHERE dd.CalendarYear = 2021
9     AND dd.MonthNumberOfYear BETWEEN 1 AND 6
10 GROUP BY dc.FullName
11 ORDER BY `Customer Rank`
12 LIMIT 10;
```

Below the query window, the 'Result Grid' shows the following data:

Customer Rank	Customer Name	Profit
1	Jordan C Turner	2379.3114
2	Marco Lopez	2357.3527
3	Lacey C Zheng	2137.7282
4	Larry Munoz	2131.1552
5	Ruben Suarez	2088.7512
6	Martin Suri	2079.3738
7	Ariana D Gray	2056.1764
8	Ricky M Navarro	2047.9379
9	Kelvin A Carson	2047.2475
10	Lawrence M Sanz	2042.1645

3.4

The screenshot shows a SQL query window with the following SQL code:

```
1 • SELECT
2     d1.Region AS `Region`,
3     dc.FullName AS `Customer Name`,
4     SUM(fi.OrderQty) AS `Quantity Sold`,
5     RANK() OVER (ORDER BY SUM(fi.OrderQty) DESC) AS `Rank`
6 FROM Fact_InternetSales fi
7 JOIN Dim_Customer dc ON fi.CustomerKey = dc.CustomerKey
8 JOIN Dim_Location d1 ON fi.ShipToLocationKey = d1.LocationKey
9 WHERE d1.Region = 'Europe'
10 GROUP BY dc.FullName
11 ORDER BY `Quantity Sold` DESC
12 LIMIT 5;
```

Below the query window, the 'Result Grid' shows the following data:

Region	Customer Name	Quantity Sold	Rank
Europe	April L Shan	58	1
Europe	Lisa Cai	25	2
Europe	Jordan C Turner	17	3
Europe	Lacey C Zheng	17	3
Europe	Marco Lopez	17	3

3.5

3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8

Limit to 1000 rows

```

1 • SELECT
2     dl.Country AS `Country`,
3     dd.CalendarYear AS `Year`,
4     dd.EnglishMonthName AS `Month`,
5     SUM(fi.OrderLineShippingCost) AS `Shipping Costs`
6 FROM Fact_InternetSales fi
7 JOIN Dim_Location dl ON fi.ShipToLocationKey = dl.LocationKey
8 JOIN Dim_Date dd ON fi.OrderDateKey = dd.DateKey
9 WHERE dl.Country = 'United Kingdom'
10    AND dd.CalendarYear = 2020
11    AND fi.ShipMethod = 'Cargo International'
12    AND dd.MonthNumberOfYear BETWEEN 1 AND 6
13 GROUP BY dl.Country, dd.CalendarYear, dd.MonthNumberOfYear, dd.EnglishMonthName
14 ORDER BY dd.MonthNumberOfYear;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Country	Year	Month	Shipping Costs
United Kingdom	2020	January	195.0000
United Kingdom	2020	February	390.0000
United Kingdom	2020	March	312.0000
United Kingdom	2020	April	351.0000
United Kingdom	2020	May	332.0000
United Kingdom	2020	June	1473.0000

Result Grid
Form Editor

3.6

SQL File 13* 3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8

Limit to 1000 rows

```

1 • WITH RankedProducts AS (
2     SELECT
3         dp.ProductTopCategoryName AS `Product Top Category`,
4         dp.ProductSubCategoryName AS `Product Sub Category`,
5         dp.ProductName AS `Product Model`,
6         SUM(fi.OrderQty) AS `Quantity Sold`,
7         RANK() OVER (
8             PARTITION BY dp.ProductTopCategoryName
9             ORDER BY SUM(fi.OrderQty) DESC
10        ) AS `Rank`
11 FROM Fact_InternetSales fi
12 JOIN Dim_Product dp ON fi.ProductKey = dp.ProductKey
13 GROUP BY dp.ProductTopCategoryName, dp.ProductSubCategoryName, dp.ProductName
14 )
15 SELECT
16     `Product Top Category`,
17     `Product Sub Category`,
18     `Product Model`,
19     `Quantity Sold`
20 FROM RankedProducts
21 WHERE `Rank` <= 3
22 ORDER BY `Product Top Category`, `Quantity Sold` DESC;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Product Top Category	Product Sub Category	Product Model	Quantity Sold
Accessories	Bottles and Cages	Water Bottle - 30 oz.	4244
Accessories	Tires and Tubes	Patch Kit/8 Patches	3191
Accessories	Tires and Tubes	Mountain Tire Tube	3095
Bikes	Mountain Bikes	Mountain-200 Black, 46	620
Bikes	Mountain Bikes	Mountain-200 Black, 42	614
Bikes	Mountain Bikes	Mountain-200 Silver, 38	596
Clothing	Caps	AWC Logo Cap	2190
Clothing	Gloves	Half-Finger Gloves, M	499
Clothing	Gloves	Half-Finger Gloves, S	488

Result Grid
Form Editor
Field Types

3.7

File 13* 3_1 3_2 3_3 3_4 3_5 3_6 3_7

Limit to 1000 rows

```

1 • SELECT
2     dl.Country,
3     SUM(fi.OrderLineProfit) AS `Profit`
4 FROM Fact_InternetSales fi
5 JOIN Dim_Product dp ON fi.ProductKey = dp.ProductKey
6 JOIN Dim_Location dl ON fi.ShipToLocationKey = dl.LocationKey
7 WHERE fi.UnitPrice BETWEEN 1000 AND 2000
8 GROUP BY dl.Country
9 ORDER BY `Profit` DESC
10 LIMIT 3;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

	Country	Profit
▶	USA	415614.1974
	Australia	343058.6932
	United Kingdom	163509.0315

3.8

MDX Query - connected to bike_sales.xml

Schema 4 bike_sales.xml Connect

```

SELECT
  [Measures].[TaxAmount] ON COLUMNS,
  ([OrderDate.Days].[2021].[1],[ShippedTo].[Country].&[France]),
  ([OrderDate.Days].[2021].[1],[ShippedTo].[Country].&[Germany]),
  ([OrderDate.Days].[2021].[2],[ShippedTo].[Country].&[France]),
  ([OrderDate.Days].[2021].[2],[ShippedTo].[Country].&[Germany]),
  ([OrderDate.Days].[2021].[3],[ShippedTo].[Country].&[France]),
  ([OrderDate.Days].[2021].[3],[ShippedTo].[Country].&[Germany]),
  ([OrderDate.Days].[2021].[4],[ShippedTo].[Country].&[France]),
  ([OrderDate.Days].[2021].[4],[ShippedTo].[Country].&[Germany]),
  ([OrderDate.Days].[2021].[5],[ShippedTo].[Country].&[France]),
  ([OrderDate.Days].[2021].[5],[ShippedTo].[Country].&[Germany]),
  ([OrderDate.Days].[2021].[6],[ShippedTo].[Country].&[France]),
  ([OrderDate.Days].[2021].[6],[ShippedTo].[Country].&[Germany]) ON ROWS
FROM [bike_sales]

```

```

Axis #0:
{}
Axis #1:
{[Measures].[TaxAmount]}
Axis #2:
{([OrderDate.Days].[2021].[1],[ShippedTo.Territory].[Europe].[France])
([OrderDate.Days].[2021].[1],[ShippedTo.Territory].[Europe].[Germany])
([OrderDate.Days].[2021].[2],[ShippedTo.Territory].[Europe].[France])
([OrderDate.Days].[2021].[2],[ShippedTo.Territory].[Europe].[Germany])
([OrderDate.Days].[2021].[3],[ShippedTo.Territory].[Europe].[France])
([OrderDate.Days].[2021].[3],[ShippedTo.Territory].[Europe].[Germany])
([OrderDate.Days].[2021].[4],[ShippedTo.Territory].[Europe].[France])
([OrderDate.Days].[2021].[4],[ShippedTo.Territory].[Europe].[Germany])
([OrderDate.Days].[2021].[5],[ShippedTo.Territory].[Europe].[France])
([OrderDate.Days].[2021].[5],[ShippedTo.Territory].[Europe].[Germany])
([OrderDate.Days].[2021].[6],[ShippedTo.Territory].[Europe].[France])
([OrderDate.Days].[2021].[6],[ShippedTo.Territory].[Europe].[Germany])}
Row #0: $28,179.94
Row #1: $16,455.83
Row #2: $19,852.65
Row #3: $17,006.64
Row #4: $29,396.27
Row #5: $21,615.93
Row #6: $25,516.98
Row #7: $21,735.85
Row #8: $33,606.18
Row #9: $23,530.64
Row #10: $563.21
Row #11: $380.12

```

Execute

ile 13*

3_13_23_33_43_53_63_73_8

Limit to 1000 rows

1 • SELECT

2 dd.CalendarYear AS `Calendar Year`,

3 dd.EnglishMonthName AS `Month`,

4 dl.Country,

5 SUM(fi.OrderLineTaxAmt) AS `Tax Amount`

6 FROM Fact_InternetSales fi

7 JOIN Dim_Location dl ON fi.ShipToLocationKey = dl.LocationKey

8 JOIN Dim_Date dd ON fi.OrderDateKey = dd.DateKey

9 WHERE dl.Country IN ('France', 'Germany')

10 AND dd.CalendarYear = 2021

11 AND dd.MonthNumberOfYear BETWEEN 1 AND 6

12 GROUP BY dd.CalendarYear, dd.EnglishMonthName, dl.Country, dd.MonthNumberOfYear

13 ORDER BY dd.CalendarYear, dd.MonthNumberOfYear, dl.Country;

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Result Grid

Form Editor

Field Types

Query Stats

	Calendar Year	Month	Country	Tax Amount
▶	2021	January	France	28179.9395
	2021	January	Germany	16455.8284
	2021	February	France	19852.6459
	2021	February	Germany	17006.6435
	2021	March	France	29396.2691
	2021	March	Germany	21615.9339
	2021	April	France	25516.9813
	2021	April	Germany	21735.8471
	2021	May	France	33606.1848
	2021	May	Germany	23530.6368
	2021	June	France	563.2117
	2021	June	Germany	380.1160