**Topic:** Documentation of Queries for CA2

**Class:** DAAA/FT/2A/03

**Group Name:** FYRE

**Team Members:**

Rachel Tan Min Zhi p2011802

Fleming Siow p2011828

Pek Yi Chern p2012184

Ethan Tan p2012085

**Query 1: What is the percentage growth in sales performance for each quarter between 2012 and 2013?**

| USE MusicStoreDWFYRE;  GO  DROP FUNCTION IF EXISTS dbo.Get\_Quarterly\_Performance\_For\_Year;  GO  CREATE FUNCTION dbo.Get\_Quarterly\_Performance\_For\_Year(@year SMALLINT) RETURNS TABLE AS  RETURN  SELECT  T.[Year],  T.[Quarter],  SUM(Quantity \* UnitPrice) AS [Sales]  FROM  MusicFact M  INNER JOIN  TimeDIM T ON M.DateKey = T.DateKey  WHERE  T.[Year] = @year  GROUP BY  T.[Year],  T.[Quarter];  GO  -- parameter --  DECLARE @custom\_year SMALLINT = NULL;  -- parameter --  DECLARE @selected\_year SMALLINT = (SELECT ISNULL(@custom\_year, MAX([Year])) FROM TimeDIM INNER JOIN MusicFact ON TimeDIM.DateKey = MusicFact.DateKey);  SELECT  Later.[Year] AS [Calendar Year],  Later.[Quarter] AS [Calendar Quarter],  Later.Sales AS [Selected Year Sales],  Earlier.Sales AS [Previous Year Sales],  CAST((Later.Sales - Earlier.Sales) / Earlier.Sales \* 100.0 AS DECIMAL(5, 2)) AS [Percentage Growth]  FROM  dbo.Get\_Quarterly\_Performance\_For\_Year(@selected\_year) AS Later  INNER JOIN  dbo.Get\_Quarterly\_Performance\_For\_Year(@selected\_year - 1) AS Earlier ON Later.[Quarter] = Earlier.[Quarter]  ORDER BY  [Calendar Quarter] ASC;  GO | |
| --- | --- |
| **Results:** | |
| **Insights:** | **Recommendations:** |
| In 2013, there was a fall in sales in quarters 1, 2 and 3, compared to 2012. The greatest drop in sales was in quarter 3. | Rolling Music Store should analyse their sales operations during the first 3 quarters of 2013 - especially in quarter 3 - to understand what resulted in the drop in sales, as well as the 4th quarter of 2013 to understand what went well. With their findings, they should take the necessary actions in the non-performing quarters to increase sales. |

**Query 2: What is the total sales made by all sales agents broken down by local and foreign sales and also the proportion of foreign sales?**

| USE MusicStoreDWFYRE;  GO  DROP FUNCTION IF EXISTS dbo.Get\_Sales\_By\_Salesperson\_And\_Nationality;  GO  CREATE FUNCTION dbo.Get\_Sales\_By\_Salesperson\_And\_Nationality() RETURNS TABLE AS  RETURN  SELECT  EmployeeKey,  IIF(Country = 'Canada', 'Local', 'Foreign') AS [Nationality],  SUM(Quantity) AS [Sales]  FROM  MusicFact M  INNER JOIN  CustomerDIM C ON M.CustomerKey = C.CustomerKey  GROUP BY  M.EmployeeKey,  IIF(Country = 'Canada', 'Local', 'Foreign');  GO  SELECT  E.FirstName + ' ' + E.LastName AS [Employee Name],  L.Sales + F.Sales AS [Total Sales],  L.Sales AS [Local Sales],  F.Sales AS [Foreign Sales],  CAST(F.[Sales Float] / (L.[Sales Float] + F.[Sales Float]) \* 100.0 AS DECIMAL(5, 2)) AS [Foreign Sales Proportion]  FROM  (SELECT  \*,  CAST([Sales] AS FLOAT) AS [Sales Float]  FROM  dbo.Get\_Sales\_By\_Salesperson\_And\_Nationality()  WHERE  [Nationality] = 'Local') AS L  INNER JOIN  (SELECT  \*,  CAST([Sales] AS FLOAT) AS [Sales Float]  FROM  dbo.Get\_Sales\_By\_Salesperson\_And\_Nationality()  WHERE [Nationality] = 'Foreign') F ON L.EmployeeKey = F.EmployeeKey  INNER JOIN  EmployeeDIM E ON L.EmployeeKey = E.EmployeeKey  ORDER BY  [Total Sales] DESC;  GO | |
| --- | --- |
| **Results:** | |
| **Insights:** | **Recommendations:** |
| Overall, Jane Peacock is the top sales support agent. She performs the best in selling music tracks to locals. Conversely, Margaret Park performs the best in selling music tracks to foreigners. | Jane Peacock should be given a bonus to encourage her to keep up her good sales performance.  If there are local customers, they should be assigned to Jane Peacock while Margaret Park should handle foreign customers. |

**Query 3: What are the top 2 positive and negative differences in the number of tracks sold between 2009 and 2013?**

| USE MusicStoreDWFYRE;  GO  DROP FUNCTION IF EXISTS dbo.Get\_Country\_Sales\_By\_Year;  GO  CREATE FUNCTION dbo.Get\_Country\_Sales\_By\_Year() RETURNS TABLE AS  RETURN  SELECT  [Year],  Country,  SUM(Quantity) AS [Sales]  FROM  MusicFact M  INNER JOIN  CustomerDIM C ON M.CustomerKey = C.CustomerKey  INNER JOIN  TimeDIM T ON M.DateKey = T.DateKey  GROUP BY  [Year],  Country;  GO  -- parameters --  DECLARE @custom\_min\_year SMALLINT = NULL;  DECLARE @custom\_max\_year SMALLINT = NULL;  -- parameters --  DECLARE @min\_year CHAR(4) = CONVERT(CHAR(4), (SELECT ISNULL(@custom\_min\_year, MIN([Year])) FROM TimeDIM INNER JOIN MusicFact ON TimeDIM.DateKey = MusicFact.DateKey));  DECLARE @max\_year CHAR(4) = CONVERT(CHAR(4), (SELECT ISNULL(IIF(@custom\_max\_year > @custom\_min\_year, @custom\_max\_year, NULL), MAX([Year])) FROM TimeDIM));  EXEC ('  SELECT  [Country],  [Earlier Sales] AS [Sales (' + @min\_year + ')],  [Later Sales] AS [Sales (' + @max\_year + ')],  [Growth]  FROM  (SELECT  \*,  [Later Sales] - [Earlier Sales] AS [Growth],  RANK() OVER (ORDER BY [Later Sales] - [Earlier Sales] ASC) AS [Asc Rank],  RANK() OVER (ORDER BY [Later Sales] - [Earlier Sales] DESC) AS [Desc Rank]  FROM  (SELECT  ISNULL(Latest\_Sales.[Country], Earliest\_Sales.[Country]) AS [Country],  ISNULL(Earliest\_Sales.[Sales], 0) AS [Earlier Sales],  ISNULL(Latest\_Sales.[Sales], 0) AS [Later Sales]  FROM  (SELECT  \*  FROM  dbo.Get\_Country\_Sales\_By\_Year()  WHERE  [Year] = ' + @min\_year + '  ) Earliest\_Sales  FULL OUTER JOIN  (SELECT  \*  FROM  dbo.Get\_Country\_Sales\_By\_Year()  WHERE  [Year] = ' + @max\_year + '  ) Latest\_Sales ON Earliest\_Sales.Country = Latest\_Sales.Country  ) tmp  ) temp  WHERE  [Asc Rank] <= 2 OR [Desc Rank] <= 2  ORDER BY  [Growth] DESC  '); | |
| --- | --- |
| **Results:** | |
| **Insights:** | **Recommendations:** |
| There is an increase in tracks sold to customers in Argentina and Canada. While there is a decrease in tracks sold to customers in the United States and Germany. Out of the 4 countries, the majority of Rolling Music Store’s business is done with customers from the United States. | Rolling Music Store should collect feedback from their customers in these 4 countries to understand their flaws and strengths.  To improve customer loyalty in Argentina and Canada, they should offer a small gift, such as vouchers, to customers and express their gratitude.  For the United States and Germany, they should research on the more popular artists in those countries, and investigate possible causes for the drop in sales, such as overly-priced tracks. |

**Query 4: Which are the top 3 and bottom 3 genres in the variation in sales over the 4 quarters and what is their average annual quantity?**

| USE MusicStoreDWFYRE;  GO  DROP FUNCTION IF EXISTS dbo.STD\_ROW;  GO  CREATE FUNCTION dbo.STD\_ROW(@n1 FLOAT, @n2 FLOAT, @n3 FLOAT, @n4 FLOAT, @df BIT = 1) RETURNS FLOAT AS  BEGIN  DECLARE @n INT = IIF(@df = 1, 3, 4);  DECLARE @mean FLOAT = (@n1 + @n2 + @n3 + @n4) / 4;  RETURN SQRT((SQUARE(@n1 - @mean) + SQUARE(@n2 - @mean) + SQUARE(@n3 - @mean) + SQUARE(@n4 - @mean)) / @n);  END;  GO  DROP FUNCTION IF EXISTS dbo.Get\_Genre\_By\_Quarter;  GO  CREATE FUNCTION dbo.Get\_Genre\_By\_Quarter(@quarter INT = 0) RETURNS VARCHAR(MAX) AS  BEGIN  DECLARE @filter\_condition VARCHAR(50) = IIF(@quarter = 0, '', 'WHERE t.[Quarter] = ' + CAST(@quarter AS CHAR(1)));  DECLARE @label VARCHAR(20) = IIF(@quarter = 0, 'Total', 'Q' + CAST(@quarter AS CHAR(1)));  RETURN  'SELECT  tr.Genre,  SUM(Quantity) AS ' + @label + '  FROM  MusicFact m  INNER JOIN  TimeDIM t ON m.DateKey = t.DateKey  INNER JOIN  TrackDIM tr ON m.TrackKey = tr.TrackKey  ' + @filter\_condition + '  GROUP BY  tr.Genre'  END;  GO  DROP FUNCTION IF EXISTS dbo.Get\_Total\_Quarters;  GO  CREATE FUNCTION dbo.Get\_Total\_Quarters(@quarter INT = 0) RETURNS VARCHAR(MAX) AS  BEGIN  DECLARE @label VARCHAR(20) = 'Number of Q' + CAST(@quarter AS CHAR(1));  RETURN  'SELECT  COUNT(\*) AS ''' + @label + '''  FROM  (SELECT  [Year], [Quarter]  FROM  TimeDIM t  WHERE  t.[Quarter] = 1  GROUP BY  [Year],  [Quarter]) tmp';  END;  GO  -- parameters --  DECLARE @prefix VARCHAR(30) = 'Proportion (Q';  DECLARE @postfix VARCHAR(10) = ')';  DECLARE @top\_count TINYINT = 3;  DECLARE @bottom\_count TINYINT = 3;  -- parameters --  DECLARE @q1 VARCHAR(MAX) = dbo.Get\_Genre\_By\_Quarter(1);  DECLARE @q2 VARCHAR(MAX) = dbo.Get\_Genre\_By\_Quarter(2);  DECLARE @q3 VARCHAR(MAX) = dbo.Get\_Genre\_By\_Quarter(3);  DECLARE @q4 VARCHAR(MAX) = dbo.Get\_Genre\_By\_Quarter(4);  DECLARE @num\_of\_q1 VARCHAR(MAX) = dbo.Get\_Total\_Quarters(1);  DECLARE @num\_of\_q2 VARCHAR(MAX) = dbo.Get\_Total\_Quarters(2);  DECLARE @num\_of\_q3 VARCHAR(MAX) = dbo.Get\_Total\_Quarters(3);  DECLARE @num\_of\_q4 VARCHAR(MAX) = dbo.Get\_Total\_Quarters(4);  EXEC ('  SELECT  Genre,  [q1] AS [' + @prefix + '1' + @postfix + '],  [q2] AS [' + @prefix + '2' + @postfix + '],  [q3] AS [' + @prefix + '3' + @postfix + '],  [q4] AS [' + @prefix + '4' + @postfix + '],  [Average Annual Qty],  [Standard Deviation]  FROM  (SELECT  Genre,  [q1],  [q2],  [q3],  [q4],  [Average Annual Qty],  [Standard Deviation],  RANK() OVER (ORDER BY [Standard Deviation] ASC) AS ''std asc'',  RANK() OVER (ORDER BY [Standard Deviation] DESC) AS ''std desc''  FROM  (SELECT  \*,  CAST(dbo.STD\_ROW([q1], [q2], [q3], [q4], 0) AS DECIMAL(4, 2)) AS [Standard Deviation]  FROM  (SELECT  Genre,  CAST([q1] / [Average Annual Qty] \* 100.0 AS DECIMAL(5, 2)) AS [q1],  CAST([q2] / [Average Annual Qty] \* 100.0 AS DECIMAL(5, 2)) AS [q2],  CAST([q3] / [Average Annual Qty] \* 100.0 AS DECIMAL(5, 2)) AS [q3],  CAST([q4] / [Average Annual Qty] \* 100.0 AS DECIMAL(5, 2)) AS [q4],  [Average Annual Qty]  FROM  (SELECT  \*,  [q1] + [q2] + [q3] + [q4] AS ''Average Annual Qty''  FROM  (SELECT  A.Genre AS ''Genre'',  CAST(CAST(A.[Q1] AS FLOAT) / [Number of Q1] AS DECIMAL(5, 2)) AS [q1],  CAST(CAST(B.[Q2] AS FLOAT) / [Number of Q2] AS DECIMAL(5, 2)) AS [q2],  CAST(CAST(C.[Q3] AS FLOAT) / [Number of Q3] AS DECIMAL(5, 2)) AS [q3],  CAST(CAST(D.[Q4] AS FLOAT) / [Number of Q4] AS DECIMAL(5, 2)) AS [q4]  FROM  (' + @q1 + ') AS A  INNER JOIN  (' + @q2 + ') AS B ON A.Genre = B.Genre  INNER JOIN  (' + @q3 + ') AS C ON B.Genre = C.Genre  INNER JOIN  (' + @q4 + ') AS D ON C.Genre = D.Genre,  (' + @num\_of\_q1 + ') AS Q1\_no,  (' + @num\_of\_q2 + ') AS Q2\_no,  (' + @num\_of\_q3 + ') AS Q3\_no,  (' + @num\_of\_q4 + ') AS Q4\_no  ) tmp  ) temp  ) tmp  ) temp  ) tmp  WHERE  [std asc] <= ' + @bottom\_count + ' OR [std desc] <= ' + @top\_count + '  ORDER BY  [Standard Deviation] DESC;'  );  GO | |
| --- | --- |
| **Results:** | |
| **Insights:** | **Recommendations:** |
| “Classical”, “Sci Fi & Fantasy” and “R&B/Soul” genres have the highest variation across the 4 quarters.  “Classical” had the highest sales in the 2nd and 4th quarters.  “Sci Fi & Fantasy” had the highest sales in the 2nd quarter.  “R&B/Soul” had the highest sales in the 3rd quarter.  Latin, Rock and Soundtrack genres have the lowest variation across the 4 quarters, which implies that the sales are more consistent. | Rolling Music Store should stock up on tracks from the   * “Classical” and “Sci Fi & Fantasy” genres in the 2nd quarter * “R&B/Soul” genre in the 3rd quarter * Classical in the 4th quarter   To ensure that there will be sufficient stocks to reduce loss of sales due to out of stock. |

**Query 5: Who is the best-selling artist for each of the top 3 genres with the highest total sales?**

| USE MusicStoreDWFYRE;  GO  -- parameter --  DECLARE @top\_genre\_count TINYINT = 3;  -- parameter --  SELECT  TOP (@top\_genre\_count)  [Genre],  [Best Selling Artist],  [Total Sales],  [Total Sales (this genre)],  CAST([Proportion of Sales] AS DECIMAL(5, 2)) [Proportion of Sales / %]  FROM  (SELECT  Genre,  [Best Selling Artist],  [Total Sales],  100.0 \* [Total Sales] / SUM([Total Sales]) OVER (PARTITION BY Genre) AS [Proportion of Sales],  SUM([Total Sales]) OVER (PARTITION BY Genre) [Total Sales (this genre)],  [Sales Rank]  FROM  (SELECT  tr.Genre,  tr.Artist AS [Best Selling Artist],  SUM(m.Quantity \* m.UnitPrice) AS [Total Sales],  RANK() OVER (PARTITION BY tr.Genre ORDER BY SUM(m.Quantity \* m.UnitPrice) DESC) AS [Sales Rank]  FROM  MusicFact m  INNER JOIN  TrackDIM tr ON m.TrackKey = tr.TrackKey  GROUP BY  tr.Genre,  tr.Artist  ) AS tmp  ) temp  WHERE  [Sales Rank] = 1  ORDER BY  [Total Sales (this genre)] DESC;  GO | |
| --- | --- |
| **Results:** | |
| **Insights:** | **Recommendations:** |
| Rock, Latin and Metal are the genres with the highest total sales.  Compared to Rock and Latin, the best selling artist for Metal, Metallica, has the most significant contribution (34.47%) to the total sales for the Metal genre. | Rolling Music Store should organise a promotional event, such as offering a 10-20% discount for songs or bundle sales (Buy 2 get 1 free) that belong to these artists. This event would likely attract many customers into the store as these are the most popular artists and genres. |