# Administrative Details

Name: Ethan Tan

Admin: p2012085

Class: DAAA/FT/2A/03

# Exporting Data

I used the SSMS Import and Export Wizard to export the CustomerDIM, EmployeeDIM, TrackDIM and MusicFact data to flat files of the corresponding labels, specifying the pipe (|) delimiter and Unicode encoding.

In Databricks, I created the directory /FileStore/tables/MusicStoreDW2012085 to contain the data files exported above.

I then uploaded the data files to the directory /FileStore/tables/MusicStoreDW2012085.

# Data Warehouse Setup

Using the data files mentioned above, I created the CustomerDIM, EmployeeDIM, TrackDIM and MusicFact tables (with data) mirroring my group’s Data Warehouse on SQL Server.

The MusicFact table required some adjustments as there is no TimeDIM here.

# 3 Insightful Queries

## Query 1) Year-on-Year comparison of Sales Growth by Quarter (group)

|  |
| --- |
| USE MusicStoreDW2012085;  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  (SELECT  YEAR(DateValue) AS `Year`,  QUARTER(DateValue) AS `Quarter`,  SUM(Quantity \* UnitPrice) AS `Sales`  FROM  MusicFact M  WHERE  YEAR(DateValue) = 2013  GROUP BY  `Year`,  `Quarter`) AS Later  INNER JOIN  (SELECT  YEAR(DateValue) AS `Year`,  QUARTER(DateValue) AS `Quarter`,  SUM(Quantity \* UnitPrice) AS `Sales`  FROM  MusicFact M  WHERE  YEAR(DateValue) = 2012  GROUP BY  `Year`,  `Quarter`) AS Earlier ON Later.`Quarter` = Earlier.`Quarter`  ORDER BY  `Calendar Quarter` ASC; |
|  |
|  |
| **Insights:**  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.  Nonetheless, sales improved from 2012 Q4 to 2013 Q4. |
| **Recommendations:**  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.  With their findings, they should take the necessary measures to increase sales in the under-performing quarters.  They ought to take note of what went well in the 4th quarter of 2013 too. If any practices enhanced their sales during that time period, they should implement them in the future.  For instance, if the sales growth in 2013 Q4 (from 2012 Q4) was due to a holiday sale, they should organise such events more often. |

## Query 2) Sales Growth by Country from 2009 to 2013 (group)

|  |
| --- |
| USE MusicStoreDW2012085;  WITH tmp (min\_year, max\_year) AS (SELECT 2009, 2013)  SELECT  `Country`,  `Earlier Sales` AS `Sales (2009)`,  `Later Sales` AS `Sales (2013)`,  `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  IFNULL(Latest\_Sales.`Country`, Earliest\_Sales.`Country`) AS `Country`,  IFNULL(Earliest\_Sales.`Sales`, 0) AS `Earlier Sales`,  IFNULL(Latest\_Sales.`Sales`, 0) AS `Later Sales`  FROM  (SELECT  \*  FROM  (SELECT  YEAR(DateValue) AS `Year`,  Country,  SUM(Quantity) AS `Sales`  FROM  MusicFact M  INNER JOIN  CustomerDIM C ON M.CustomerKey = C.CustomerKey  GROUP BY  `Year`,  Country), tmp  WHERE  `Year` = min\_year  ) Earliest\_Sales  FULL OUTER JOIN  (SELECT  \*  FROM  (SELECT  YEAR(DateValue) AS `Year`,  Country,  SUM(Quantity) AS `Sales`  FROM  MusicFact M  INNER JOIN  CustomerDIM C ON M.CustomerKey = C.CustomerKey  GROUP BY  `Year`,  Country), tmp  WHERE  `Year` = max\_year  ) Latest\_Sales ON Earliest\_Sales.Country = Latest\_Sales.Country  ) temp  ) temp  WHERE  `Asc Rank` <= 2 OR `Desc Rank` <= 2  ORDER BY  `Growth` DESC; |
|  |
|  |
| **Insights:**  There is an increase in tracks sold to customers in Argentina and Canada, while there is a decrease in tracks sold to customers from the United States and Germany.  Out of the above 4 countries, the majority of Rolling Music Store’s business is done with customers from the United States. |
| **Recommendations:**  Rolling Music Store should collect feedback from their customers in these 4 countries to understand their flaws and strengths.  If sales growth in Argentina (for example) is consistent, Jackson Sam should consider setting up a local retail store in Argentina.  He should perhaps give exclusive benefits to their United States customers, such as free shipping for deliveries. This rewards United States customers collectively for their high sales revenue and encourages sales growth, all while avoiding any potential bias among their customers.  For the United States and Germany, they should research on the more popular artists/genres in those countries, and investigate possible causes for the drop in sales, such as unreasonable track pricing or lack of promotional reach. |

## Query 3) Determine Awards for Commendable Employee Performance (individual)

|  |
| --- |
| USE MusicStoreDW2012085;  WITH employee\_performance AS (  SELECT  `Employee`,  `Year`,  `Mean Sales`,  `Std of Sales`,  `Sales Rank`,  `Consistency Rank`  FROM  (SELECT  `Employee`,  `Year`,  ROUND(AVG(`Sales`), 2) AS `Mean Sales`,  ROUND(STD(`Sales`), 2) AS `Std of Sales`,  RANK() OVER (PARTITION BY `Year` ORDER BY AVG(`Sales`) DESC) `Sales Rank`,  RANK() OVER (PARTITION BY `Year` ORDER BY STD(`Sales`) ASC) `Consistency Rank`  FROM  (SELECT  Complete\_Time.`Year`,  Complete\_Time.`Month`,  CONCAT(FirstName, ' ', LastName) AS `Employee`,  IFNULL(Sales\_Info.`Sales`, 0.0) AS `Sales`  FROM  (SELECT  YEAR(DateValue) AS `Year`,  MONTH(DateValue) AS `Month`,  EmployeeKey,  SUM(Quantity \* UnitPrice) AS `Sales`  FROM  MusicFact  GROUP BY  `Year`,  `Month`,  EmployeeKey  ) Sales\_Info  RIGHT OUTER JOIN  (SELECT  \*  FROM  (SELECT DISTINCT YEAR(DateValue) AS `Year` FROM MusicFact) Y  FULL OUTER JOIN  (SELECT DISTINCT MONTH(DateValue) AS `Month` FROM MusicFact) M  FULL OUTER JOIN  (SELECT DISTINCT EmployeeKey FROM MusicFact) E  ) Complete\_Time ON  Sales\_Info.`Year` = Complete\_Time.`Year`  AND  Sales\_Info.`Month` = Complete\_Time.`Month`  AND  Sales\_Info.`EmployeeKey` = Complete\_Time.`EmployeeKey`  INNER JOIN  EmployeeDIM ON Complete\_Time.EmployeeKey = EmployeeDIM.EmployeeKey)  GROUP BY  `Employee`,  `Year`)  WHERE  `Sales Rank` = 1  OR  `Consistency Rank` = 1  )  SELECT  IFNULL(Best\_Sales.Employee, Most\_Consistent.Employee) AS `Employee`,  `Best Performing Salesperson of`,  `Most Consistent Salesperson of`  FROM  ((SELECT  `Employee`,  COLLECT\_SET(CONCAT(`Year`, ' with mean monthly sales of $', `Mean Sales`)) AS `Best Performing Salesperson of`  FROM  employee\_performance  WHERE  `Sales Rank` = 1  GROUP BY  `Employee`) Best\_Sales  FULL OUTER JOIN  (SELECT  `Employee`,  COLLECT\_SET(`Year`) AS `Most Consistent Salesperson of`  FROM  employee\_performance  WHERE  `Consistency Rank` = 1  GROUP BY  `Employee`) Most\_Consistent ON Best\_Sales.`Employee` = Most\_Consistent.`Employee`  ); |
|  |
| Insights:  … |
| Recommendations:  … |