A picture containing text

Description automatically generated

Data Warehousing - DMM 531

Coursework - Part 1

SURYA L RAMESH (0206793)

Module Co-ordinator: DR. HATEM AHRIZ

OCTOBER 31 ,2020

Table of Contents

[Task 1: Data Modelling and ETL 2](#_Toc55040481)

[1.1 Table Identification 2](#_Toc55040482)

[1.2 List of Relationships 3](#_Toc55040483)

[1.3 List of Hierarchies 3](#_Toc55040484)

[1.4 Future Proofing the design 4](#_Toc55040485)

[Task 2: Data Analysis 4](#_Toc55040486)

[2.1 DAX formulae 4](#_Toc55040487)

[2.2 Answers to Questions 6](#_Toc55040488)

[2.3 Pivot Tables 6](#_Toc55040489)

[2.4 Discussion & Conclusion 8](#_Toc55040490)

# Task 1: Data Modelling and ETL

## 1.1 Table Identification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table Name | No. of rows | Type | Columns | Justification |
| Calendar Lookup | 729 | Dimension | Date | PK |
| Day | Calendar table has been created as its essential for time-based analysis. The Day, Month Name, Year and Day are all related to the date, which will be linked to the transaction date. |
| Month Name |
| Year |
| Quarter |
| Customer | 8,842 | Dimension | customer\_id | PK |
| first\_name | These columns contain information related to the customer and can be identified using customer\_id which is the Primary Key. |
| last\_name |
| customer\_acct\_num |
| customer\_address |
| birthdate |
| marital\_status |
| yearly\_income |
| gender |
| total\_children |
| num\_children\_at\_home |
| education |
| acct\_open\_date |
| member\_card |
| occupation |
| homeowner |
| customer\_postal\_code | FK |
| Product | 1,559 | Dimension | product\_id | PK |
| product\_name | These columns contain information related to the customer and can be identified using product\_id which is the Primary Key. |
| product\_brand |
| product\_sku |
| product\_retail\_price |
| product\_cost |
| product\_weight |
| Recyclable |
| low\_fat |
| Store | 24 | Dimension | store\_id | PK |
| store\_type | These columns contain information related to the store and can be identified using store\_id which is the Primary Key. |
| store\_name |
| store\_street\_address |
| store\_city |
| store\_state |
| store\_country |
| store\_phone |
| first\_opened\_date |
| last\_remodel\_date |
| total\_sqft |
| grocery\_sqft |
| region\_id | FK |
| Sales Region | 23 | Dimension | region\_id | PK |
| sales\_region | These columns contain information related to the Sales region and can be identified using region\_id which is the Primary Key. |
| sales\_district |
| Postcode Lookup | 8,512 | Dimension | customer\_postal\_code | Primary Key |
| Customer\_city | All related to the postcode and has been added to this separate table to avoid redundancy and can be identified using store\_id which is the Primary Key. |
| Customer\_state\_province |
| Customer\_country |
| Transaction | 269,720 | Fact | customer\_id | PK |
| product\_id | PK |
| store\_id | PK |
| transaction\_date | PK, FK |
| quantity | Not related to any other table’s primary key and needed for calculations. |
| stock\_date | Not relates to any other table’s primary key and only relevant to the stock purchased i.e. to this table. |

## 

## 1.2 List of Relationships

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Primary Key | Table | Foreign Key |
| Date | Date | Transactions | transaction\_date |
| Customer | customer\_id | Transactions | customer\_id |
| Product | product\_id | Transactions | product\_id |
| Store | store\_id | Transactions | store\_id |
| Sales Region | region\_id | Store | region\_id |

## 1.3 List of Hierarchies

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Hierarchy Name | Columns included in hierarchy | Justification |
| Date | **Calendar** | Year  Quarter  Month  Date | All related to the calendar and can be used drill down while using power pivot |
| Product | **Brand** | product\_brand,  product\_name | All related to the calendar and can be used to slice the information related to brand |
| Store | **Store\_Geography** | store\_city  store\_state  store\_country | All related to the calendar and can be used to slice the information related to the geography of the store |
| Sales Region | **Sales** | sales\_region  sales\_district | All related to the calendar and can be used to slice the information |
| Postcode Lookup | **Geography** | customer\_country  customer\_state\_province  customer\_city  customer\_postal\_code | All related to the calendar and can be used to slice the information |

## 1.4 Future Proofing the design

To allow for future data to be added, the Power Queries are loaded from a Folder instead of a single file. Multiple files associated with each year can be added to this folder.

The query has a filter added to it, to pick up only files with extensions starting with ‘.xls’. This avoids picking up any document (.doc) or any other type of file (.jpg,.zip etc) in the same folder. However, it will look at ‘.xls’ ,‘.xlsx’ and any files with macros (‘.xlsm’)files.

For an accurate date range, which look at future possible dates as well, A ‘Calendar Lookup’ table has been created which contains all the dates from a min and max range. The min and max ranges where chosen by the following code:

fx = List.Min(Transactions[transaction\_date] - renamed as Min Date

fx = List.Max(Transactions[transaction\_date]) - renamed as Max Date

And the date range was compiled using the following code:

List.Dates (#”Min Date”, Duration. Days(#”Max Date” - #”Min Date”) + 1 , #duration(1,0,0,0) - renamed as Date Range

This range of dates was converted into the type ‘Date’. This table was added to the current Data model as a connection.

The effectiveness of this , has been tested with spurious data (10 rows) added to a new excel file called ‘Coursework\_newdata’ which was copied to the source folder. It was seen that the data from this file was added to the current file and the range expanded to include the maximum date in the new file as well.

# Task 2: Data Analysis

## DAX formulae

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table Name | CC/M # | Name of CC or M | CC or M | Formula |
| Product | 1 | Max Retail Price | M | Max Retail Price:=MAX([product\_retail\_price]) |
| Customer | 2 | Average Age | M | Average Age:=AVERAGE([Customer\_Age]) |
| Store | 3 | days\_since\_opening | CC | = DATEDIFF(Store[first\_opened\_date],TODAY(),DAY) |
| Store | 4 | supermarket\_size | CC | =SWITCH(TRUE(),Store[total\_sqft]>35000,"Large",Store[total\_sqft]>25000&&Store[total\_sqft]<=35000,"Medium","Small") |
| Customer | 5 | Total Customers | M | Total Customers:=COUNTA([customer\_id]) |
| Store | 6 | store\_street\_number | CC | = LEFT(Store[store\_street\_address],SEARCH(" ",Store[store\_street\_address])-1) |
| Customer | 7 | age | CC | =DATEDIFF(Customer[birthdate],TODAY(),YEAR) |
| Customer | 8 | customer\_priority | CC | = IF ((Customer[total\_children]>=3) && (Customer[member\_card] = "Golden") && (Customer[homeowner] = "Y"),"High-Priority","Normal") |
| Customer | 9 | house\_number | CC | = LEFT(Customer[customer\_address],SEARCH (" ",Customer[customer\_address] )- 1) |
| Transactions | 10 | Weekend | CC | = IF((WEEKDAY(Transactions[transaction\_date])= 1)|| (WEEKDAY(Transactions[transaction\_date])= 7),"Y","N") |
| Transactions | 11 | Low-Fat Quantity | M | Total Low Fat quantity sold:=SUM([LowF\_quantityperitem]) |
| Transactions | 12 | Total Cost | M | Total\_CostperItemSold = RELATED('Product'[product\_cost]) \* [quantity]  Total Cost:=SUM([Total\_CostperItemSold]) |
| Transactions | 13 | Total Revenue | M | Total\_RevenueperItemSold =RELATED ('Product'[product\_retail\_price]) \* [quantity]  Total Revenue:=SUM([Total\_RevenueperItemSold]) |
| Transactions | 14 | Profit | M | Profit:=[Total Revenue] -[Total Cost] |
| Transactions | 15 | Product Brand Rank | M | =IF(HASONEVALUE('Product'[product\_brand]),RANKX(ALL('Product'[product\_brand]),Transactions[Sum of Profit\_perItemsold]))  Output seen in Excel sheet: Rank |
| Transactions | 16 | MTD Profit | M | =CALCULATE(Transactions[Sum of Profit\_perItemsold],DATESMTD('Calendar Lookup'[Date])) |
| Transactions | 17 | Last Month Profit | M | Last Month Profit:=CALCULATE((Transactions[Profit]),DATEADD('Calendar Lookup'[Date],-1,MONTH)) |
| Transactions | 18 | MoM Profit % Change | M | SalesDiff:=[MTD Profit]-[Last Month Profit]  MoM Profit % Change:=DIVIDE([SalesDiff],[Last Month Profit])  Format the result to % |

## Answers to Questions

|  |  |  |
| --- | --- | --- |
| Question # | Question | Answer |
| 1 | What is the maximum retail price for the "Green Ribbon" product brand? | $ 3.11 |
| 2 | Which store opened first (store number)? | 22 |
| 3 | How many customers are female? | 4386 |
| 4 | What was the total Low-Fat quantity sold for "High Top" product brand? | 10635 |
| 5 | What is the total cost of Tri-State products sold? | $20,283 |
| 6 | Which district saw the highest profit? | Los Angeles |
| 7 | Which brand is ranked #25? | Bravo |

## Pivot Tables

Table

Description automatically generated

Figure 1: Pivot Table 1: Month View

If the Data bars are applied only for the MTD profit column, in association with the Dates, it will show a gradually increasing bar. This may not be of much use. Hence, it has been chosen to add the Month in this chart as well. Now, it is possible to have two views:

* The daily profits (by expanding the fields) and the MTD profits associated with it.
* Month view by collapsing the fields and been able to see a more relevant Data bars with it.

Figure 1 shows the view with all fields collapsed and Figure 2 shows the partial view (For Jan) with all Dates. Figure 3 shows the second Pivot table with the Monthly profit, LMP and Mom Profit change %.

Table

Description automatically generated

Figure 2:Pivot Table 1- Partial Daily View

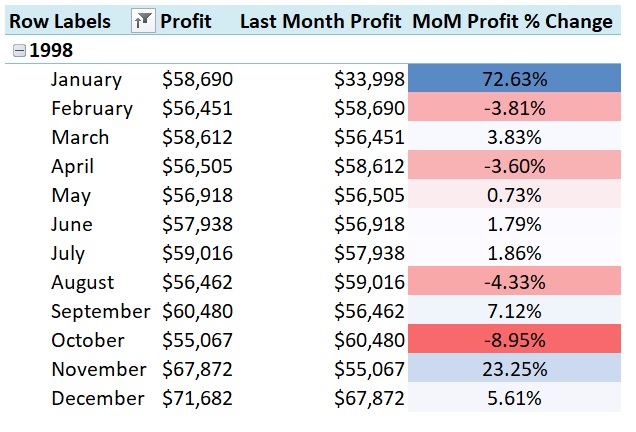


Figure 3: Pivot Table 2

## Discussion & Conclusion

**Cost:** This is a very cost-effective solution for businesses who do not want to invest in additional licenses for a dedicated database solution. Everything is done through Excel, which is a well-known software with additional functionalities built in.

**Amount of data:** The use of PowerPivot allows to circumvent the maximum limit of rows that is an issue in ‘.xls’ and ‘.xlsx’ files. As it is possible to load data from multiple files into our data model

**Scalability:** As everything is done in-memory, it may not be suited for large amounts of data. It is a bit slow during the stage where data is been refreshed. It does function well enough for this superstore which has 24 branches. But will need to be reconsidered if they are expecting to expand a lot more or dramatically increase their sales.

**Rigid Format of input files:** This design will work on the premises that a folder has been created for the source files to be saved into. Any accidental files in this folder (not of type Excel) will be ignored. It also must be noted that the users will need to add additional data (for future) in the similar format.

**Data Clean-up**: This solution also assumes that the data that is contained in the Excel sheet is pretty accurate, if you required to clean up the data in any way, a different solution may need to be looked at.

**Postcode Lookup:** In this solution, the postcode has been separated into a different table, which eliminates data redundancies in the customer table, as multiple customers can have the same postcode.

**Removal of Irrelevant Columns:** Ideally, it should be confirmed with the client if the stock\_date (in transaction table) is of any future use in analysis. For Data models in Excel, the degree of compression that can be achieved, depends primarily on the number of unique values in each column [REF 2]. Hence if a column is not needed, it needs to be removed to minimise memory usage. If it is needed and additional info regarding stock date will be provided in future, then a separate table for stock info can be created.

**Calendar Lookup:** A calendar has been created from the min and max range of dates in the transaction table and will be modified according to the data, when it is refreshed. Having a separate calendar table allows the users to analyse data based on various calendar periods (day, month, quarter, year) and identify sales anomalies and take right course of action to remediate it. However, the calendar can be made more efficient by only pulling in dates that have transactions in it .

The solution also hides certain columns from the Client tools (Day in Calendar, Primary keys in Fact Table) . The primary keys in the Fact table are hidden to avoid multiple options been given to the user. The day is hidden as it tabulates the results by all the days in the year starting with the same number and not by date as needed by the requirements. The day column has be left in the calendar table, in case future analysis by particular day in all the months (for example: beginning of the month or after pay day ) is needed. But if it is confirmed by the client as not needed, it needs to be removed.

Overall, the solution has met all the requirements given by the client. Further modifications as discussed above can be made, based on more information from the client. If the scale of Business is hoped to increase dramatically in the next few years, then a different database solution will need to be looked at. Till then, this solution is recommended to be used their business Intelligence solution.

References

Bibliography

Pg5-6 (DatewarehouseSystems)

<https://docs.microsoft.com/en-gb/dax/>

REF 2 - <https://support.microsoft.com/en-us/office/create-a-memory-efficient-data-model-using-excel-and-the-power-pivot-add-in-951c73a9-21c4-46ab-9f5e-14a2833b6a70#__toc373850953>

Ahriz,H., 2020. *Lecture 03- Normalisation*, CMM524. [PowerPoint presentation]. Advanced Database Management. Robert Gordon University, School of Computing. Available from <http://campusmoodle.rgu.ac.uk/course/view.php?id=97372> [Accessed 12-10-2020].

<https://radacad.com/basics-of-modeling-in-power-bi-what-is-a-dimension-table-and-why-say-no-to-a-single-big-table>

# <https://support.microsoft.com/en-us/office/excel-specifications-and-limits-1672b34d-7043-467e-8e27-269d656771c3#ID0EBABAAA=Office_2007>