Design of Data Warehouse and Business Intelligence System

A case study of an Online Retail Industry

Team 4 – Team members:

Leah Nguyen – 144377485

Gerardo Bohórquez Restrepo – 11579873

Javier Chapto- 14081525

Claire Ongley- 98074596

Table of Contents

The Context

Hooli is an Australian company that operates an online supermarket chain in Australia. They provide groceries such as vegetables, fruits, meat, packaged goods, etc., as well as various household items such as DVDs, magazines, pet food, baby products, stationery, etc. We can state categorically that Business Intelligence would ultimately increase Hooli's revenue by enhancing the company's ability to leverage customer data to understand buying patterns, gain insight into the success of new products, features, and services, provide visibility into the most profitable customers, and target promotions and marketing towards failing products.

The Challenges

Increasing demand for target marketing

As retail sector competition becomes more intense, acquiring new customers is exponentially more expensive than retaining existing ones. In fact, acquiring new customers might cost five times as much as retaining existing ones (Massey, Montoya- Weiss & Holcom, 2001). According to Winer (2001), repeat clients might create more than twice the amount of gross revenue as new customers.

As the amount spent on online marketing continues to rise, Hooli is concerned about its inability to assess and control its successful usage in marketing. They have found that it is likely more effective to invest in customers who are valuable or have the potential to be valuable while restricting their efforts to customers who are not worth it (i.e., not all relationships are profitable or desirable). The deployment of promotional management systems and customer segmentation will enable the company to regain control of these promotional activities and maximise the return on marketing investment by employing more personalised marketing strategies for each consumer group. Due to these types of discoveries and the fact that customers want to be handled according to their individual and specific demands, Hooli must create and manage its relationships with its customers so that they are long-lasting and profitable.

In particular, the Marketing team is eager to investigate the following business questions:

- How to segment customers and find the most valuable?
- How to optimise digital marketing channels?
- What promotions are more likely to generate effective sales?

To answer these business problems, it is necessary to create a Data warehouse and business intelligence system to assist the company's decision-makers and business strategists in making better judgments based on historical data.

Increasing data usage demand from business users in the organisation

Retail was one of the first industries to invest significantly in gathering and integrating customer data in data warehouses. The objective of the project is to aid Hooli’s management in making better decisions using the available historical data. The business users (decision makers) are unable to efficiently obtain data when needed. Several departments within the retail organisation locate their own resources, utilise different data sources, and hire consultants to meet their specific short-term data needs to remedy this deficiency.

In many instances, the same data was retrieved from the same source systems so that other departments could access it without a comprehensive information-delivery plan. As a result of a lack of integration, the disparate data sources had a detrimental impact on the reports presented by managers, which was recognised by management.

Given the significance of the information to the retail organisation, the management was encouraged to address the issue of data inconsistency by implementing a central data warehouse and guaranteeing that all users, regardless of department, have access to the data. The harmonisation of data and the necessity for consistent and high-quality reports gave rise to the company's data warehouse initiative.

Scope of the Project

This project's scope will be focused on the design and development of a data warehouse and business intelligence, which will involve data analysis and information presentation using analysis and reporting services capabilities. A business intelligence system is extremely difficult to create without a data warehouse. Due to time restrictions and information availability, the project will not cover all aspects of the company; instead, the focus will be on one data mart that represents the organization's key department, the sales and marketing divisions.

Project Objectives

- To investigate the significance of data warehouses and business intelligence systems in the retail industry.
- Create a data warehouse and business intelligence solution for the retail industry.
- To assess how the decision tools would help the decision maker make better business judgments.
- Using the case study, validate the design of the data warehouse and business intelligence.

Data Warehouse Framework

Definition

- *Business intelligence* is the timely distribution of reliable, usable information to suitable decision makers to assist successful decision-making.
- *A Data Warehouse* is a system that gathers and consolidates data from source systems on a regular basis and stores it in a dimensional or normalised data store. It typically stores years of data and is searched for corporate intelligence or other analytical purposes. It is usually updated in batches rather than every time a transaction occurs in the source system.
- *Data Mart* is a subset of a data warehouse and is described as a corpus of historical data in an electronic repository that is not involved in the organization's everyday activities. This data is instead utilised to generate business insight. The data in the data mart is often relevant to a certain region of the company.
- *Fact Table* is the principal table in a dimensional model that stores the business's numerical performance data. We attempt to consolidate the measurement data generated by a business process into a single data mart.
- *Dimension Table* is an essential component of a fact table. The textual descriptions of the business are contained in the dimension tables. Dimension tables in a well-designed dimensional model include several columns or features. These properties describe the dimension table rows. Dimension tables are often shallow in terms of row count (sometimes considerably less than 1 million rows), yet vast with several huge columns. The entry points into the fact table are dimension tables. The dimensions serve as the data warehouse's user interface.
- *Online analytic processing (OLAP)* is a system for storing, organising, and querying data that is especially developed to enable business intelligence applications.
- *Extract, Transform, and Load (ETL)* system is a collection of procedures that clean, transform, combine, de-duplicate, archive, conform, and arrange data for usage in data warehouses.

Data Model - Kimball's Model

The design of the database is dependent on the techniques of the data warehouse's founding father. Bill Inmon's process and Ralph Kimball's method are the two design processes. For this project, our group adopted Kimball, which is described in further detail below.

The key tenet of the Kimball approach is to establish the data warehouse gradually over time by combining independently built data marts. ETL for one or more data marts initiates the process. No shared staging location for data is required. There is often a distinct location for each data mart. Each department's data is treated as a data mart, and we may construct a comprehensive data warehouse and business intelligence for selected departments from the ground up.

Due to time constraints and project limitations, it is simpler to finish a process for a portion of a firm based on the data mart and then connect it as the business expands. The investigation, Analysis of the present environment, identification needs, identification architecture, data warehouse design, installation, and continuous data administration are given as steps for the process. Kimball et al (2002).

Data Warehouse Architecture: Kimballs approach

One of the primary objectives of the data warehouse is to extract data from many OLTP or flat file sources and combine it into a single repository for quick access and optimal data utilisation. Data warehouse procedures include data load and access. To accomplish the objective, the system's architecture was extremely resilient. The ETL (Extract, Transformation, and Load) procedure was utilised to populate the data warehouse.

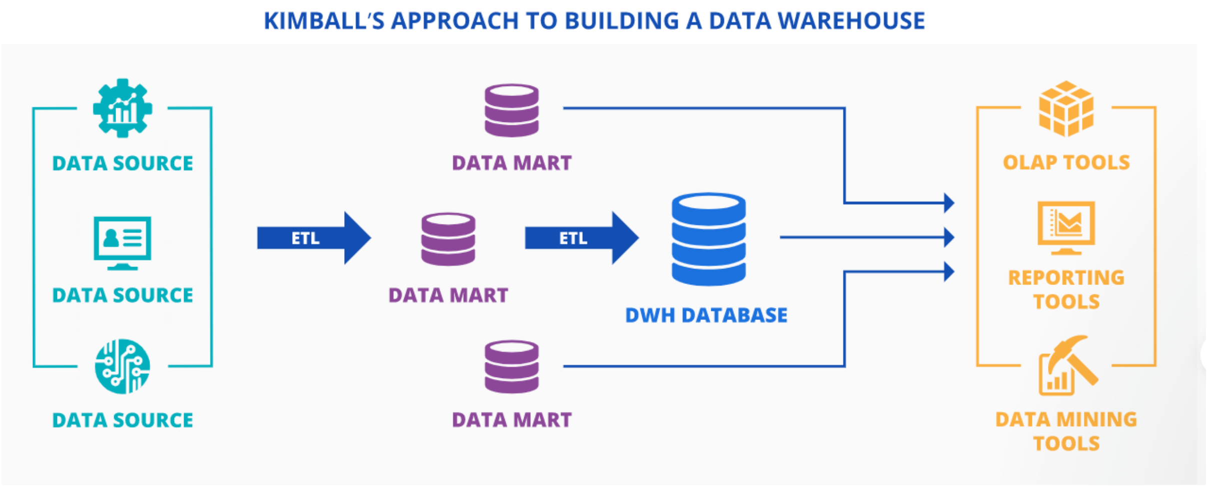


Figure 1: Kimball's Datawarehouse architecture (ScienceSoft 2022)

This project's chosen data warehouse and business intelligence architecture is displayed above. As mentioned, the Kimball approach was used to design Hooli's Data warehouse architecture. A data mart for the marketing and sales department will be created first, and users will be able to extract insights and reports from it, in the future when more data marts are added into the mix, a data warehouse for the whole organisation will be created from the different data marts.

System Design – Star Schema

We begin our data mart architecture by establishing the measures, which are the foundational and feedback data required by decision makers. We compare these criteria to the source system's capabilities (OLTP). For the purposes of this project, the data warehouse design utilised the star schema. The star schema is a relational database structure used in data marts to store measurements and dimensions. Measures are kept in a fact table and dimensions in dimension tables. The star schema is so named because for each data mart there is just one measure surrounded by dimension tables.

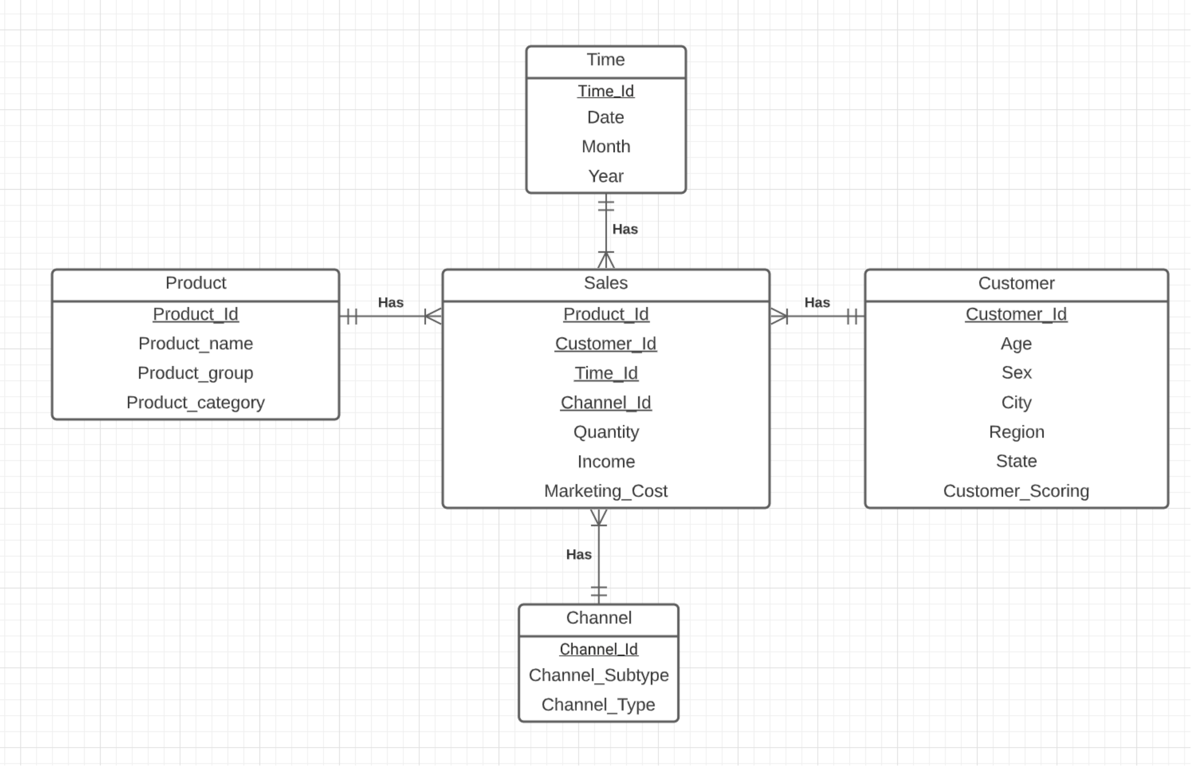


Figure 2 : Simplified star schema for Hooli

The fact table constitutes the star's centre. A column for the measure and a column for each dimension carrying the foreign key for a member of that dimension are included in the fact table. This table's primary key is created by concatenating all of the foreign key fields. Typically, the primary key of the fact table is referred to as the composite key. It includes the measurements, therefore the term "Fact."

Dimension tables include the dimensions. The dimension table has a column for the member's unique identification, which is often an integer or a short character value. There is an additional column for the description. To adhere to the naming policy for this project, we will name the dimension tables based on the information they contain and prefix them with "Dim."

Problems faced by Hooli

Problem 1: How to segment customers?

Not all customers are the same, having a unique marketing strategy for every customer will result in lost opportunities and lack of engagement with customers, identifying common characteristics between customers, in both demographic and in terms of behaviour. (El Falah et al. 2021) This allows companies to define different strategies for different types of customers that are likely going to produce better results, as they were thought based on the characteristics mentioned before.

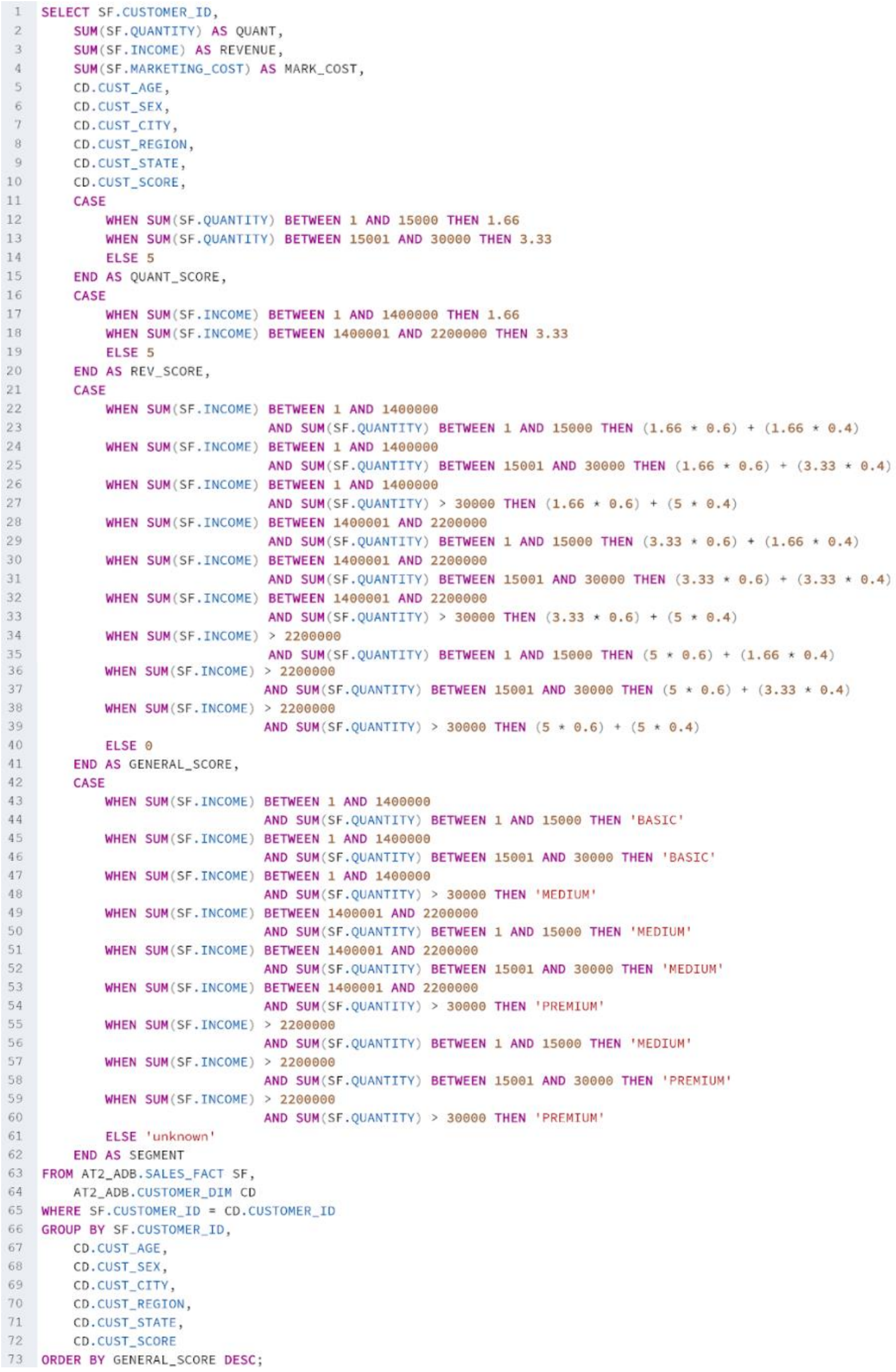
According to Wong and Yan (2018), understanding customers and segmenting them are essential activities to not only retaining high-value customers but also acquiring new ones, analysing online shopping behaviour in useful insights to generate customised packages for valuable customers at the optimal time.

Segmentation queries will be implemented to:

- Categorise customers according to their value for Hooli via a scoring method, taking different parameters into consideration, allowing Hooli to create a differentiated value proposition.
- Use customer demographic factors such as gender, age and location as categories to provide summary tables to help determine the segments where strategies might result in better outcomes (ie: promotions, further discussed in problem #3).

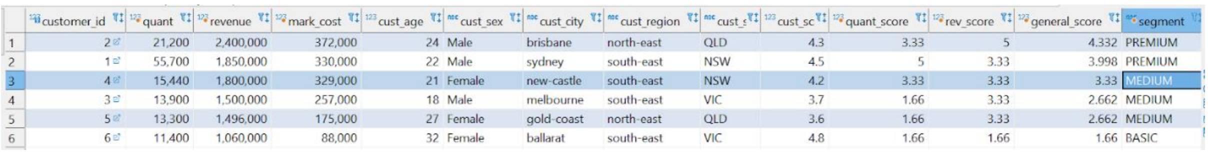
Categorise customers according to their value for Hooli via a scoring method:

This method will allow the marketing managers for Hooli to separate customers into 3 different segments, determined by the number of sales (40% weight) as well as the revenue generated (60% weight). Each customer receives a score (either 1.66, 3.33 or 5) depending on the relevant number of sales and revenue and these scores are multiplied by the corresponding weight and then aggregated to obtain the general score and segment. The scoring method can be seen in Query 1 and results in Figure 4 below.



```sql
SELECT SF.CUSTOMER_ID,
    SUM(SF.QUANTITY) AS QUANT,
    SUM(SF.INCOME) AS REVENUE,
    SUM(SF.MARKETING_COST) AS MARK_COST,
    CD.CUST_AGE,
    CD.CUST_SEX,
    CD.CUST_CITY,
    CD.CUST_REGION,
    CD.CUST_STATE,
    CD.CUST_SCORE,
    CASE
        WHEN SUM(SF.QUANTITY) BETWEEN 1 AND 15000 THEN 1.66
        WHEN SUM(SF.QUANTITY) BETWEEN 15001 AND 30000 THEN 3.33
        ELSE 5
    END AS QUANT_SCORE,
    CASE
        WHEN SUM(SF.INCOME) BETWEEN 1 AND 1400000 THEN 1.66
        WHEN SUM(SF.INCOME) BETWEEN 1400001 AND 2200000 THEN 3.33
        ELSE 5
    END AS REV_SCORE,
    CASE
        WHEN SUM(SF.INCOME) BETWEEN 1 AND 1400000
                            AND SUM(SF.QUANTITY) BETWEEN 1 AND 15000 THEN (1.66 * 0.6) + (1.66 * 0.4)
        WHEN SUM(SF.INCOME) BETWEEN 1 AND 1400000
                            AND SUM(SF.QUANTITY) BETWEEN 15001 AND 30000 THEN (1.66 * 0.6) + (3.33 * 0.4)
        WHEN SUM(SF.INCOME) BETWEEN 1 AND 1400000
                            AND SUM(SF.QUANTITY) > 30000 THEN (1.66 * 0.6) + (5 * 0.4)
        WHEN SUM(SF.INCOME) BETWEEN 1400001 AND 2200000
                            AND SUM(SF.QUANTITY) BETWEEN 1 AND 15000 THEN (3.33 * 0.6) + (1.66 * 0.4)
        WHEN SUM(SF.INCOME) BETWEEN 1400001 AND 2200000
                            AND SUM(SF.QUANTITY) BETWEEN 15001 AND 30000 THEN (3.33 * 0.6) + (3.33 * 0.4)
        WHEN SUM(SF.INCOME) BETWEEN 1400001 AND 2200000
                            AND SUM(SF.QUANTITY) > 30000 THEN (3.33 * 0.6) + (5 * 0.4)
        WHEN SUM(SF.INCOME) > 2200000
                            AND SUM(SF.QUANTITY) BETWEEN 1 AND 15000 THEN (5 * 0.6) + (1.66 * 0.4)
        WHEN SUM(SF.INCOME) > 2200000
                           AND SUM(SF.QUANTITY) BETWEEN 15001 AND 30000 THEN (5 * 0.6) + (3.33 * 0.4)
        WHEN SUM(SF.INCOME) > 2200000
                           AND SUM(SF.QUANTITY) > 30000 THEN (5 * 0.6) + (5 * 0.4)
        ELSE 0
    END AS GENERAL_SCORE,
    CASE
        WHEN SUM(SF.INCOME) BETWEEN 1 AND 1400000
                           AND SUM(SF.QUANTITY) BETWEEN 1 AND 15000 THEN 'BASIC'
        WHEN SUM(SF.INCOME) BETWEEN 1 AND 1400000
                           AND SUM(SF.QUANTITY) BETWEEN 15001 AND 30000 THEN 'BASIC'
        WHEN SUM(SF.INCOME) BETWEEN 1 AND 1400000
                           AND SUM(SF.QUANTITY) > 30000 THEN 'MEDIUM'
        WHEN SUM(SF.INCOME) BETWEEN 1400001 AND 2200000
                           AND SUM(SF.QUANTITY) BETWEEN 1 AND 15000 THEN 'MEDIUM'
        WHEN SUM(SF.INCOME) BETWEEN 1400001 AND 2200000
                           AND SUM(SF.QUANTITY) BETWEEN 15001 AND 30000 THEN 'MEDIUM'
        WHEN SUM(SF.INCOME) BETWEEN 1400001 AND 2200000
                           AND SUM(SF.QUANTITY) > 30000 THEN 'PREMIUM'
        WHEN SUM(SF.INCOME) > 2200000
                           AND SUM(SF.QUANTITY) BETWEEN 1 AND 15000 THEN 'MEDIUM'
        WHEN SUM(SF.INCOME) > 2200000
                           AND SUM(SF.QUANTITY) BETWEEN 15001 AND 30000 THEN 'PREMIUM'
        WHEN SUM(SF.INCOME) > 2200000
                           AND SUM(SF.QUANTITY) > 30000 THEN 'PREMIUM'
        ELSE 'unknown'
    END AS SEGMENT
FROM AT2_ADB.SALES_FACT SF,
    AT2_ADB.CUSTOMER_DIM CD
WHERE SF.CUSTOMER_ID = CD.CUSTOMER_ID
GROUP BY SF.CUSTOMER_ID,
    CD.CUST_AGE,
    CD.CUST_SEX,
    CD.CUST_CITY,
    CD.CUST_REGION,
    CD.CUST_STATE,
    CD.CUST_SCORE
ORDER BY GENERAL_SCORE DESC;
```

Query 1: Sample query to determine customer score and other relevant values

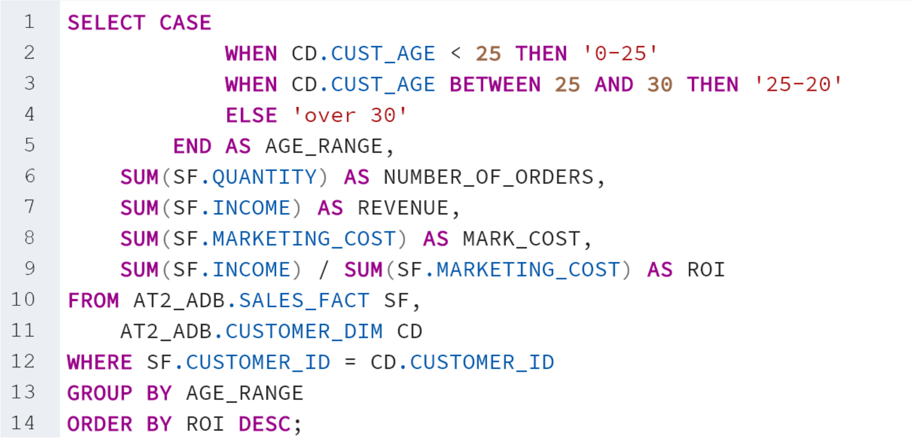


	customer_id	quant	revenue	mark_cost	cust_age	cust_sex	cust_city	cust_region	cust_s	cust_sc	quant_score	rev_score	general_score	segment
1	2	21,200	2,400,000	372,000	24	Male	brisbane	north-east	QLD	4.3	3.33	5	4.332	PREMIUM
2	1	55,700	1,850,000	330,000	22	Male	sydney	south-east	NSW	4.5	5	3.33	3.998	PREMIUM
3	4	15,440	1,800,000	329,000	21	Female	new-castle	south-east	NSW	4.2	3.33	3.33	3.33	MEDIUM
4	3	13,900	1,500,000	257,000	18	Male	melbourne	south-east	VIC	3.7	1.66	3.33	2.662	MEDIUM
5	5	13,300	1,496,000	175,000	27	Female	gold-coast	north-east	QLD	3.6	1.66	3.33	2.662	MEDIUM
6	6	11,400	1,060,000	88,000	32	Female	ballarat	south-east	VIC	4.8	1.66	1.66	1.66	BASIC

Figure 3: Results of Query 1 to determine the most valuable customers

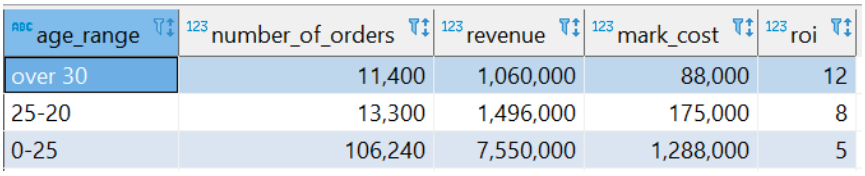
Use customer demographic factors as categories to provide summary information:

Summary table by age group:



```
SELECT CASE
            WHEN CD.CUST_AGE < 25 THEN '0-25'
            WHEN CD.CUST_AGE BETWEEN 25 AND 30 THEN '25-20'
            ELSE 'over 30'
        END AS AGE_RANGE,
    SUM(SF.QUANTITY) AS NUMBER_OF_ORDERS,
    SUM(SF.INCOME) AS REVENUE,
    SUM(SF.MARKETING_COST) AS MARK_COST,
    SUM(SF.INCOME) / SUM(SF.MARKETING_COST) AS ROI
FROM AT2_ADB.SALES_FACT SF,
    AT2_ADB.CUSTOMER_DIM CD
WHERE SF.CUSTOMER_ID = CD.CUSTOMER_ID
GROUP BY AGE_RANGE
ORDER BY ROI DESC;
```

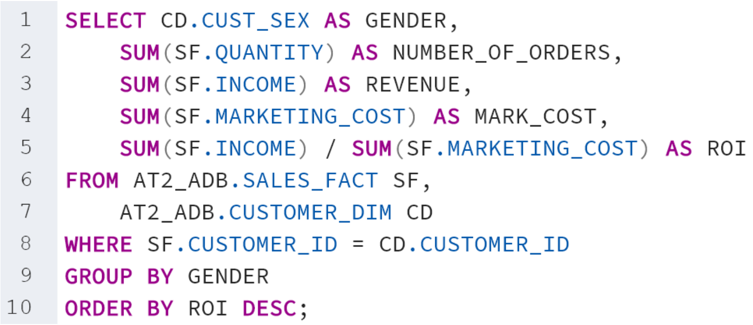
Query 2: Sample query for ROI by age range



age_range	number_of_orders	revenue	mark_cost	roi
over 30	11,400	1,060,000	88,000	12
25-20	13,300	1,496,000	175,000	8
0-25	106,240	7,550,000	1,288,000	5

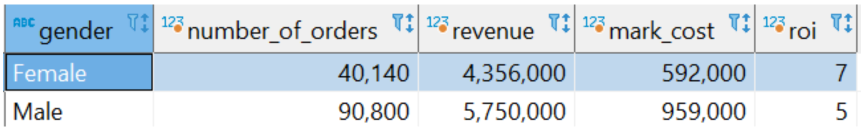
Figure 4: Results of Query 2 – ROI by age range

Summary table by Gender:



```
SELECT CD.CUST_SEX AS GENDER,
    SUM(SF.QUANTITY) AS NUMBER_OF_ORDERS,
    SUM(SF.INCOME) AS REVENUE,
    SUM(SF.MARKETING_COST) AS MARK_COST,
    SUM(SF.INCOME) / SUM(SF.MARKETING_COST) AS ROI
FROM AT2_ADB.SALES_FACT SF,
    AT2_ADB.CUSTOMER_DIM CD
WHERE SF.CUSTOMER_ID = CD.CUSTOMER_ID
GROUP BY GENDER
ORDER BY ROI DESC;
```

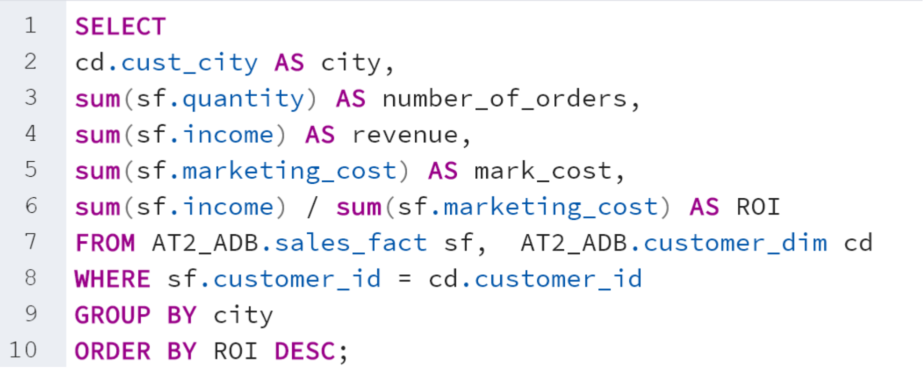
Query 3: Sample query for ROI by gender



gender	number_of_orders	revenue	mark_cost	roi
Female	40,140	4,356,000	592,000	7
Male	90,800	5,750,000	959,000	5

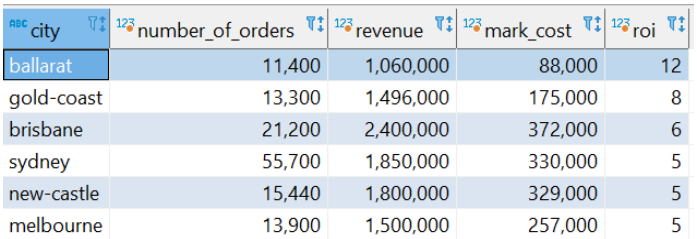
Figure 5: Results of Query 3 - ROI by gender

Summary table by location (city):



```sql
SELECT
cd.cust_city AS city,
sum(sf.quantity) AS number_of_orders,
sum(sf.income) AS revenue,
sum(sf.marketing_cost) AS mark_cost,
sum(sf.income) / sum(sf.marketing_cost) AS ROI
FROM AT2_ADB.sales_fact sf,  AT2_ADB.customer_dim cd
WHERE sf.customer_id = cd.customer_id
GROUP BY city
ORDER BY ROI DESC;
```

Query 4: Sample query for ROI by location



city	number_of_orders	revenue	mark_cost	roi
ballarat	11,400	1,060,000	88,000	12
gold-coast	13,300	1,496,000	175,000	8
brisbane	21,200	2,400,000	372,000	6
sydney	55,700	1,850,000	330,000	5
new-castle	15,440	1,800,000	329,000	5
melbourne	13,900	1,500,000	257,000	5

Figure 6: Results of Query 4 - ROI by location

The three summary tables created above, and the marketing segmentation based on scoring only provide information for decision makers, real value from performing this segmentation will rely on the ability of managers to further explore the different segments and come up with valuable actions like the ones discussed on problem 3.

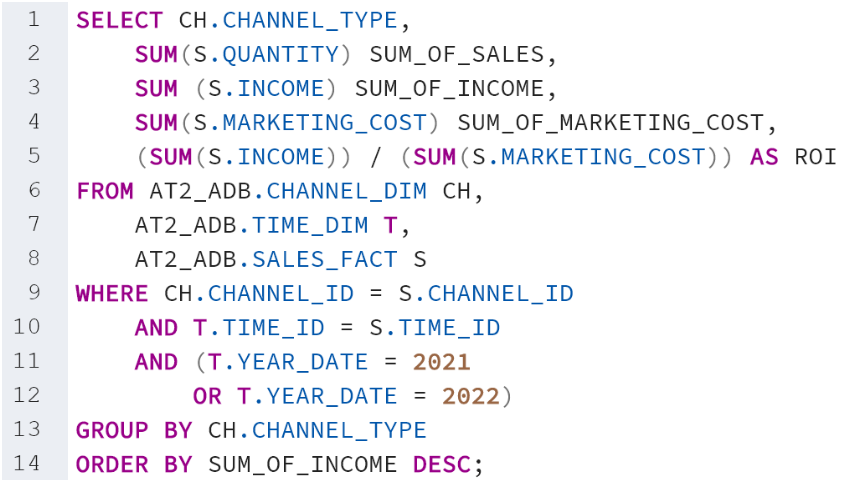
Problem 2: How to optimise digital marketing channels?

The company we are studying has different ways to drive traffic to its website and get online sales. It uses organic digital marketing channels such as social media marketing, SEO and email marketing, as well as paid digital marketing channels such as Facebook Ads, Google Ads Search, Google Ads Display and Google Shopping. Also, marketing managers have to decide how much budget should be allocated to each digital marketing channel in order to maximise income, number of sales and return on investment. Using a data warehouse that joins data from different sources such as Google Ads, Google Analytics and Facebook Ads, data analysts can make online analytical processing (OLAP) queries that support decision making by managers. Also, the main KPIs have to be presented to managers in dashboards that show information visually in a manner that is easy to understand and allows them to make good business decisions fast. This process in which we change data to information, information to knowledge and knowledge into plans that drive profitable business action can be defined as business intelligence. It’s important to emphasise that metrics need to be actionable rather than vanity metrics. Marketing managers don’t need metrics that lack substance, context and make the business appear impressive. Rather than that, they need actionable metrics. For example, metrics like income, number of sales and return on investment of each digital marketing channel, that will help them to decide if they will increase or decrease the budget in each digital marketing channel.

If marketing managers analysed the metrics provided by each digital marketing channel’s platform separately, the process would be highly inefficient. For example, if marketing managers had to enter Facebook Ads, Google Ads, Google Analytics, Mailchimp, google search console and Instagram insights, it would be difficult for them to compare the different channels, and to decide which is performing better. Also, they would see too many metrics, and it would be difficult for them to decide which KPIs should they pay more attention to. Using a dashboard that gets the information from a custom-made data warehouse solves all these problems.

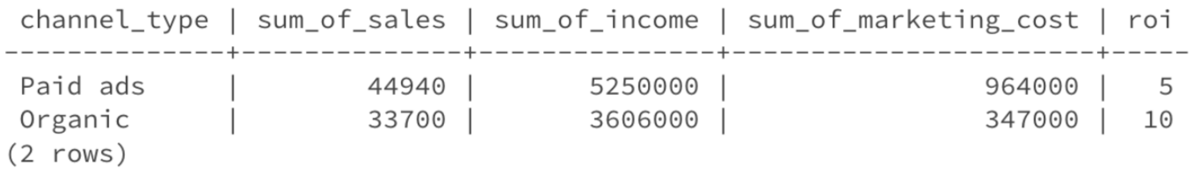
Some may argue that using Google Analytics, that is a tool that shows information from many different sources and assigns sales and income to whatever source generated the last click is a good enough solution. However, it is important to consider that Google Analytics doesn’t include the amount spent on advertisement channels that aren’t owned by Google, like Facebook Ads for example. Also, it doesn’t include data like the cost of work hours, which is especially important when considering strategies like SEO and social media marketing, which are highly time consuming. The marketing costs related to each digital marketing channel are highly relevant because they are necessary to calculate the return on investment (ROI) of each channel, which is one of the main KPIs used to determine how well a determined channel is performing (ROI of each channel = income of each channel / marketing costs of each channel). As all marketing costs can be loaded to a data warehouse, using a data warehouse and dashboards would allow marketing managers to make a more informed decision, than just using Google Analytics.

Using our proposed schema marketing managers could analyse how each channel is performing for each customer, each day and each product, for example. But knowing all this information is not really necessary to make decisions. It is better to summarise or aggregate data using a roll up (also called drill up) OLAP operation. For example, the following query shows the sum of sales, sum of income, sum of marketing cost and ROI of each type of channel ordered by income:



```sql
SELECT CH.CHANNEL_TYPE,
    SUM(S.QUANTITY) SUM_OF_SALES,
    SUM (S.INCOME) SUM_OF_INCOME,
    SUM(S.MARKETING_COST) SUM_OF_MARKETING_COST,
    (SUM(S.INCOME)) / (SUM(S.MARKETING_COST)) AS ROI
FROM AT2_ADB.CHANNEL_DIM CH,
    AT2_ADB.TIME_DIM T,
    AT2_ADB.SALES_FACT S
WHERE CH.CHANNEL_ID = S.CHANNEL_ID
    AND T.TIME_ID = S.TIME_ID
    AND (T.YEAR_DATE = 2021
        OR T.YEAR_DATE = 2022)
GROUP BY CH.CHANNEL_TYPE
ORDER BY SUM_OF_INCOME DESC;
```

Query 5: Sample query for determining ROI based on marketing

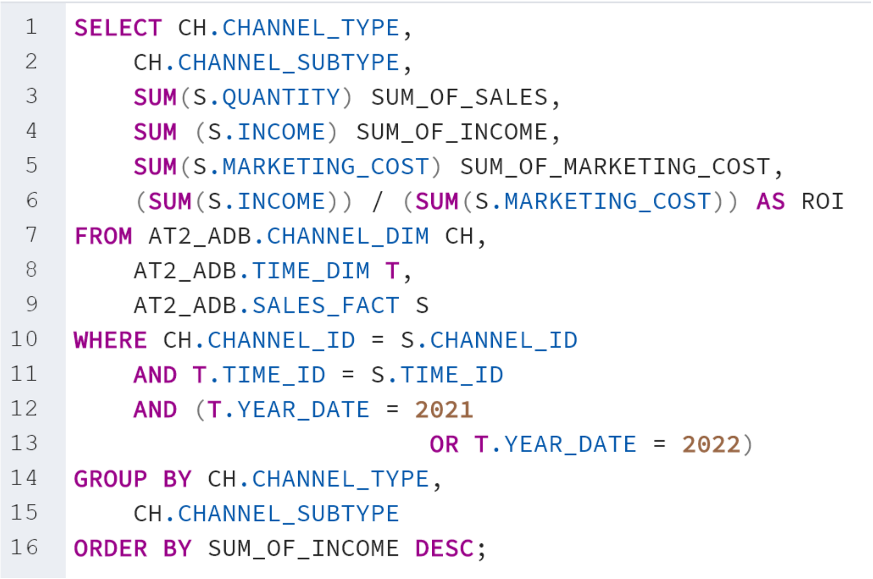


```
 channel_type | sum_of_sales | sum_of_income | sum_of_marketing_cost | roi
--------------+--------------+---------------+-----------------------+-----
 Paid ads     |        44940 |       5250000 |                964000 |   5
 Organic      |        33700 |       3606000 |                347000 |  10
(2 rows)
```

Figure 7: Sample output of marketing channels based on above query 5

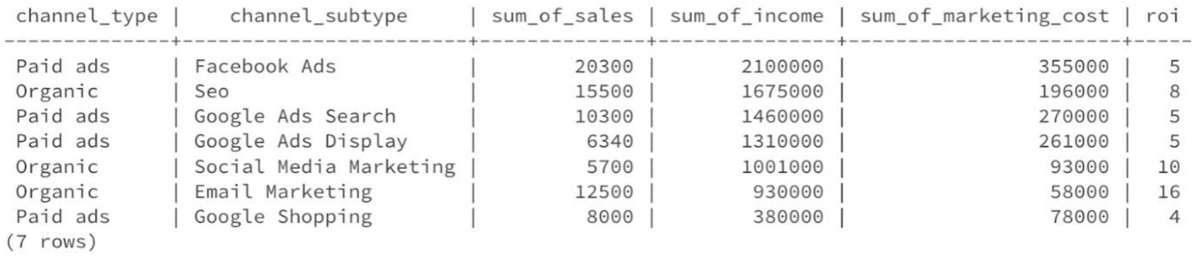
This aggregation is achieved by climbing up hierarchy (hierarchical roll-up), ignoring sub-types of channels, and by dimensional reduction, ignoring dimension tables such as the customer dimension table and the product dimension table. Also, since everything changed since the COVID 19 pandemic, it may be convenient to see only data of years 2021 and 2022 (after the pandemic). This can be done with a WHERE clause in the SQL query, which is a DICE OLAP operation because there is a range select condition on one dimension, which is the year number.

However, marketing directors may also want to analyse data with a higher level of granularity and may want to be able to see the performance of each channel subtype as well. To show more specific data, the roll-up operation can be reversed with a roll down or drill down OLAP operation. This is shown in Query 6, which shows the sum of sales, sum of income, sum of marketing cost and ROI of each type and subtype of channel ordered by income in a descending order:



```sql
SELECT CH.CHANNEL_TYPE,
    CH.CHANNEL_SUBTYPE,
    SUM(S.QUANTITY) SUM_OF_SALES,
    SUM (S.INCOME) SUM_OF_INCOME,
    SUM(S.MARKETING_COST) SUM_OF_MARKETING_COST,
    (SUM(S.INCOME)) / (SUM(S.MARKETING_COST)) AS ROI
FROM AT2_ADB.CHANNEL_DIM CH,
    AT2_ADB.TIME_DIM T,
    AT2_ADB.SALES_FACT S
WHERE CH.CHANNEL_ID = S.CHANNEL_ID
    AND T.TIME_ID = S.TIME_ID
    AND (T.YEAR_DATE = 2021
                        OR T.YEAR_DATE = 2022)
GROUP BY CH.CHANNEL_TYPE,
    CH.CHANNEL_SUBTYPE
ORDER BY SUM_OF_INCOME DESC;
```

Query 6: Sample query to determine ROI for specific age ranges



```
 channel_type |    channel_subtype     | sum_of_sales | sum_of_income | sum_of_marketing_cost | roi
--------------+------------------------+--------------+---------------+-----------------------+-----
 Paid ads     | Facebook Ads           |        20300 |       2100000 |                355000 |   5
 Organic      | Seo                    |        15500 |       1675000 |                196000 |   8
 Paid ads     | Google Ads Search      |        10300 |       1460000 |                270000 |   5
 Paid ads     | Google Ads Display     |         6340 |       1310000 |                261000 |   5
 Organic      | Social Media Marketing |         5700 |       1001000 |                 93000 |  10
 Organic      | Email Marketing        |        12500 |        930000 |                 58000 |  16
 Paid ads     | Google Shopping        |         8000 |        380000 |                 78000 |   4
(7 rows)
```

Figure 8: Results of query 6 - drilled down for ROI from marketing

Problem 3: What promotions are more likely to generate effective sales?

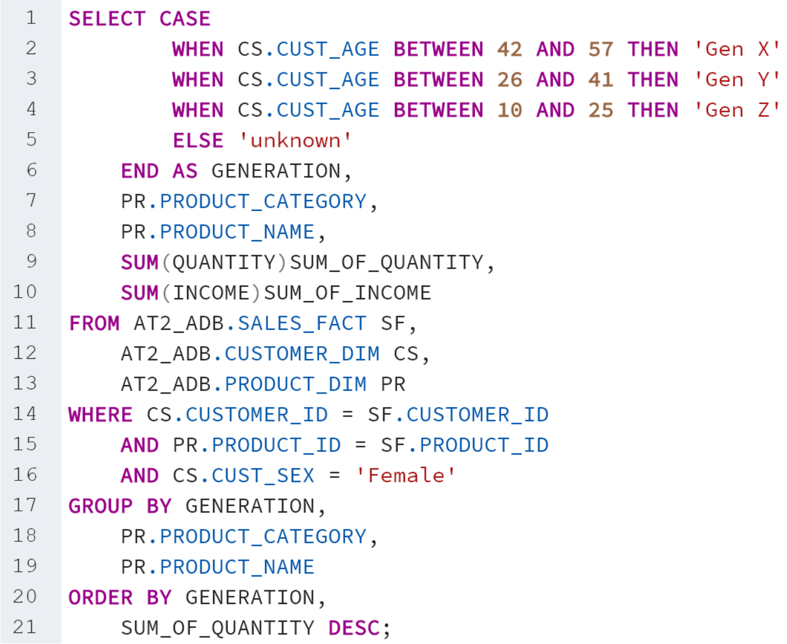
To enhance sales made by Hooli, different promotions targeting different demographics are required. This is because it is illogical to target over 65yr olds with marketing for a product that is purchased by mainly 18-20yr olds. In the star schema for the data warehouse, the customer dimension has several fields that can help with segmenting the sales to determine valid promotions. These promotions could also be time based using the time dimension such as looking at Christmas sales or Halloween for example. (El Falah et al 2021)

The analysis of sales by regional segregation will allow managers or marketing teams to determine which areas are over performing and underperforming. (Oketjuni and Omodara, 2011). Regional promotions can be determined by changing the query to group by CUST_REGION or CUST_STATE. This change would allow for marketing departments to enhance sales by region as each region would not have identical sales to every other region. It is also possible to combine both regional and demographic queries to target an even more specific customer demographic attempt to increase sales there. (Oketjuni and Omodara, 2011)

By using customer segregation to determine the associations rules that are present within the data, the marketing managers will be able to more efficiently target customers in particular demographics to increase sales in general and generate more revenue by selling more products to the same customer base. It should also be noted that data warehousing could be used to track which demographic makes the least purchases and thus allow the production and marketing that target these demographics.

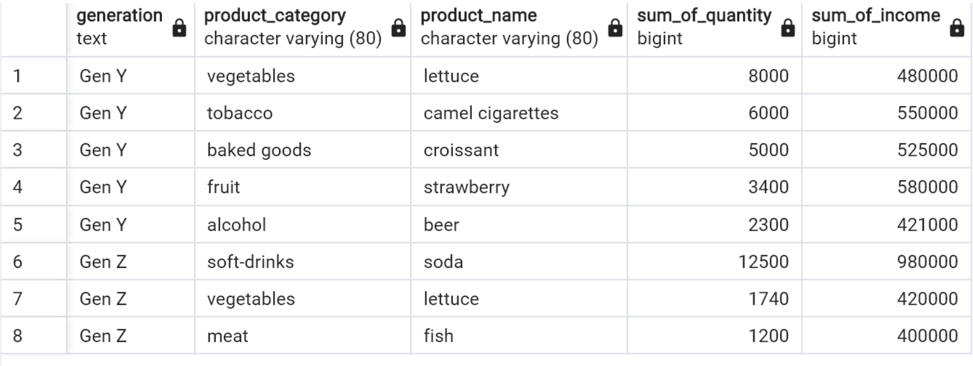
While the query above mainly focuses on the customer segregation to allow for marketing to enhance sales, other forms of segmentation could be used including examining click through rates on the company website and use of Google Analytics, Facebook Ads or Google Ads to determine which products were being clicked on and then applying this to marketing strategies and promotions. Other types of sale segregation could be determined such as people who use their phones to make purchases or a personal computer as well as payment method or other such characteristics (Insani and Soemitro, 2016). These separation methods could be added in the future to help with marketing decisions if further detail was required.

It is possible that when using the data warehouse to determine the most sold item per demographic, it may not be as expected when taking other factors into account. For the sake of testing the data, the following query was created:



```sql
SELECT CASE
        WHEN CS.CUST_AGE BETWEEN 42 AND 57 THEN 'Gen X'
        WHEN CS.CUST_AGE BETWEEN 26 AND 41 THEN 'Gen Y'
        WHEN CS.CUST_AGE BETWEEN 10 AND 25 THEN 'Gen Z'
        ELSE 'unknown'
    END AS GENERATION,
    PR.PRODUCT_CATEGORY,
    PR.PRODUCT_NAME,
    SUM(QUANTITY)SUM_OF_QUANTITY,
    SUM(INCOME)SUM_OF_INCOME
FROM AT2_ADB.SALES_FACT SF,
    AT2_ADB.CUSTOMER_DIM CS,
    AT2_ADB.PRODUCT_DIM PR
WHERE CS.CUSTOMER_ID = SF.CUSTOMER_ID
    AND PR.PRODUCT_ID = SF.PRODUCT_ID
    AND CS.CUST_SEX = 'Female'
GROUP BY GENERATION,
    PR.PRODUCT_CATEGORY,
    PR.PRODUCT_NAME
ORDER BY GENERATION,
    SUM_OF_QUANTITY DESC;
```

Query 7: Sample query to determine demographic by generation



	generation text	product_category character varying (80)	product_name character varying (80)	sum_of_quantity bigint	sum_of_income bigint
1	Gen Y	vegetables	lettuce	8000	480000
2	Gen Y	tobacco	camel cigarettes	6000	550000
3	Gen Y	baked goods	croissant	5000	525000
4	Gen Y	fruit	strawberry	3400	580000
5	Gen Y	alcohol	beer	2300	421000
6	Gen Z	soft-drinks	soda	12500	980000
7	Gen Z	vegetables	lettuce	1740	420000
8	Gen Z	meat	fish	1200	400000

Figure 9: Sample output using query in Query 7

This query takes all the sales made by the company and determines the customers' generation based on their age. The results are returned first ordered by generation and then SUM_OF_QUANTITY so that it is possible to see the most sold item in the company. It is possible that marketing managers instead would prefer to use SUM_OF_INCOME to determine promotions but that is not shown in this query. Customer segmentation helps provide a “deeper understanding” of customers and can be used to identify “profitable customers” (Insani and Soemitro, 2016). The sales made can have association rules applied to them as well. Because this query can separate customers by age range, it is possible to determine if females in the 26-41 age group purchase more of a particular product than those in the 42-57 age group. Association rules could be made by further looking at what sales were made in conjunction with these and then applying promotions to those items to further increase sales. In other words, females in the 26-41 age range (A) purchase lettuce (B) thus implying that A -> B. Customer segregation as a method of analysis can be seen in Oketjuni and Omodara (2011).

Conclusion

As a FMCG organisation, Hooli’s concerns about the increasing demand of targeted marketing and data source inconsistency are relevant but can be addressed by the implementation of the proposed data warehouse solutions. Through analysis of three areas, Hooli can effectively increase their sales and therefore revenue by customer segmentation, identifying the most effective promotion opportunities and optimisation of digital marketing channels. Data marts catered to each department will allow effective use of the historical data to determine relevant trends and associations as necessary.

Although the proposed solutions and implementations represent considerable progress in providing Hooli with data and tools to generate valuable insights and as support for decision making, future work can be done to improve those solutions and implementations. Like for example including more facts and dimension tables that would allow Hooli to take more factors into consideration when performing segmentation, analysing channels and measuring promotions success. This could include but not be limited to adding data about customer behaviour within the platform and the different channels (number of sessions, sessions per purchase, average ticket, etc) or adding data about historical promotions performance.

Further work recommended includes the creation of data marts for the other departments to consolidate the organisational data warehouse and implementing dashboards and visualisations tools.

References

El Falah, Z., Rafalia, N., & Abouchabaka, J. (2021). An Intelligent Approach for Data Analysis and Decision Making in Big Data: A Case Study on

E-commerce Industry. *International Journal of Advanced Computer Science and Applications*, *12*(7), 723-736. https://pdfs.semanticscholar.org/bfb5/71b60e6c04f4681a1cc3697f6ef044b4785f.pdf

Insani, R., & Soemitro, H. L. (2016, 27th February). *Business Intelligence for Profiling Telecommunication Customers* [Proceedings Paper]. Second Asia Pacific Conference on Advanced Research, Melbourne. https://apiar.org.au/wp-content/uploads/2016/05/APCAR_BRR7120_ICT-289-298.pdf

Kimball R. and Ross M., (2002) the Data Warehouse Toolkit: Second Edition, the Complete Guide to Dimensional Modeling.

Massey, A. P., Montoya-Weiss, M. M., & Holcom, K. (2001). Re-engineering the customer relationship: Leveraging knowledge assets at IBM. Decision Support Systems, 32(2), 155-170. https://doi.org/10.1016/S0167-9236(01)00108-7

Oketunji, T. A., & Omodara, R. O. (2011). *Design of Data Warehouse and Business Intelligence System* [Blekinge Institute of Technology]. Karlskrona, Sweden. https://www.diva-portal.org/smash/get/diva2:831050/FULLTEXT01.pdf

ScienceSoft USA Corporation, *Science Soft*, accessed 9th of October 2022, https://www.scnsoft.com/analytics/data-warehouse/building

Winer, R. S. (2001). A Framework for Customer Relationship Management. California Management Review, 43(4), 89–105. https://doi.org/10.2307/41166102

Wong, E, & Yan, W. (2018). *Customer online shopping experience data analytics: Integrated customer segmentation and customised services prediction model*. International Journal of Retail & Distribution Management; Bradford, Vol 46, Iss 4, 406-420. https://www-proquest-com.ezproxy.lib.uts.edu.au/docview/2034194183?accountid=17095&parentSessionId=5YXmtpvVaCdEjZG9%2BMg%2BA1Sy3kZjogFZweDLviFAhkU%3D&pq-origsite=primo

Appendix – Queries in text form

Query 1: Sample query to determine customer score and other relevant values

```
SELECT sf.customer_id, sum(sf.quantity) AS quant, sum(sf.income) AS
revenue, sum(sf.marketing_cost) AS mark_cost,

 cd.cust_age,cd.cust_sex,cd.cust_city,cd.cust_region, cd.cust_state,
cd.cust_score,

 CASE WHEN sum(sf.quantity) BETWEEN 1 AND 15000 THEN 1.66

 WHEN sum(sf.quantity) BETWEEN 15001 AND 30000 THEN 3.33

 ELSE 5

 END AS QUANT_SCORE,

  CASE WHEN sum(sf.income) BETWEEN 1 AND 1400000 THEN 1.66

 WHEN sum(sf.income) BETWEEN 1400001 AND 2200000 THEN 3.33

 ELSE 5

END AS rev_score,

  CASE WHEN sum(sf.income) BETWEEN 1 AND 1400000 AND
sum(sf.quantity) BETWEEN 1 AND 15000 THEN (1.66*0.6)+(1.66*0.4)

  WHEN sum(sf.income) BETWEEN 1 AND 1400000 AND sum(sf.quantity)
BETWEEN 15001 AND 30000 THEN (1.66*0.6)+(3.33*0.4)

  WHEN sum(sf.income) BETWEEN 1 AND 1400000 AND sum(sf.quantity) >
30000 THEN (1.66*0.6)+(5*0.4)

  WHEN sum(sf.income) BETWEEN 1400001 AND 2200000 AND
sum(sf.quantity) BETWEEN 1 AND 15000 THEN (3.33*0.6)+(1.66*0.4)

  WHEN sum(sf.income) BETWEEN 1400001 AND 2200000 AND
sum(sf.quantity) BETWEEN 15001 AND 30000 THEN (3.33*0.6)+(3.33*0.4)

  WHEN sum(sf.income) BETWEEN 1400001 AND 2200000 AND
sum(sf.quantity) > 30000 THEN (3.33*0.6)+(5*0.4)

  WHEN sum(sf.income) >2200000 AND sum(sf.quantity) BETWEEN 1 AND
15000 THEN (5*0.6)+(1.66*0.4)

  WHEN sum(sf.income) >2200000 AND sum(sf.quantity) BETWEEN 15001
AND 30000 THEN (5*0.6)+(3.33*0.4)

  WHEN sum(sf.income) >2200000 AND sum(sf.quantity) >30000 THEN
(5*0.6)+(5*0.4)
```

```
  ELSE 0

  END AS general_score,

  CASE WHEN sum(sf.income) BETWEEN 1 AND 1400000 AND
sum(sf.quantity) BETWEEN 1 AND 15000 THEN 'BASIC'

  WHEN sum(sf.income) BETWEEN 1 AND 1400000 AND sum(sf.quantity)
BETWEEN 15001 AND 30000 THEN 'BASIC'

  WHEN sum(sf.income) BETWEEN 1 AND 1400000 AND sum(sf.quantity) >
30000 THEN 'MEDIUM'

  WHEN sum(sf.income) BETWEEN 1400001 AND 2200000 AND
sum(sf.quantity) BETWEEN 1 AND 15000 THEN 'MEDIUM'

  WHEN sum(sf.income) BETWEEN 1400001 AND 2200000 AND
sum(sf.quantity) BETWEEN 15001 AND 30000 THEN 'MEDIUM'

  WHEN sum(sf.income) BETWEEN 1400001 AND 2200000 AND
sum(sf.quantity) > 30000 THEN 'PREMIUM'

  WHEN sum(sf.income) >2200000 AND sum(sf.quantity) BETWEEN 1 AND
15000 THEN 'MEDIUM'

  WHEN sum(sf.income) >2200000 AND sum(sf.quantity) BETWEEN 15001
AND 30000 THEN 'PREMIUM'

  WHEN sum(sf.income) >2200000 AND sum(sf.quantity) >30000 THEN
'PREMIUM'

  ELSE 'unk'

  END AS segment

 FROM AT2_ADB.sales_fact sf,  AT2_ADB.customer_dim cd

 WHERE sf.customer_id = cd.customer_id

 GROUP BY sf.customer_id,
cd.cust_age,cd.cust_sex,cd.cust_city,cd.cust_region, cd.cust_state,
cd.cust_score

 ORDER BY GENERAL_SCORE DESC;
```

Query 2: Sample query for ROI by age range

```
SELECT

CASE WHEN cd.cust_age<25 THEN '0-25'

WHEN cd.cust_age BETWEEN 25 AND 30 THEN '25-20'

ELSE 'over 30'
```

```
END AS age_range,
sum(sf.quantity) AS number_of_orders,
sum(sf.income) AS revenue,
sum(sf.marketing_cost) AS mark_cost,
sum(sf.income) / sum(sf.marketing_cost) AS ROI
FROM AT2_ADB.sales_fact sf,  AT2_ADB.customer_dim cd
WHERE sf.customer_id = cd.customer_id
GROUP BY age_range
ORDER BY ROI DESC;
```

Query 3: Sample query for ROI by gender

```
SELECT
cd.cust_sex AS Gender,
sum(sf.quantity) AS number_of_orders,
sum(sf.income) AS revenue,
sum(sf.marketing_cost) AS mark_cost,
sum(sf.income) / sum(sf.marketing_cost) AS ROI
FROM AT2_ADB.sales_fact sf,  AT2_ADB.customer_dim cd
WHERE sf.customer_id = cd.customer_id
GROUP BY Gender
ORDER BY ROI DESC;
```

Query 4: Sample query for ROI by location

```
SELECT
cd.cust_city AS city,
sum(sf.quantity) AS number_of_orders,
sum(sf.income) AS revenue,
sum(sf.marketing_cost) AS mark_cost,
sum(sf.income) / sum(sf.marketing_cost) AS ROI
FROM AT2_ADB.sales_fact sf,  AT2_ADB.customer_dim cd
WHERE sf.customer_id = cd.customer_id
```

```
GROUP BY city

ORDER BY ROI DESC;
```

Query 5: Sample query for determining ROI based on marketing

```
SELECT ch.channel_type,

SUM(s.quantity) sum_of_sales,

SUM (s.income) sum_of_income,

SUM(s.marketing_cost) sum_of_marketing_cost,
(SUM(s.income))/(SUM(s.marketing_cost)) as roi

FROM AT2_ADB.channel_dim ch , AT2_ADB.time_dim t, AT2_ADB.sales_fact
s

WHERE ch.channel_id = s.channel_id AND t.time_id = s.time_id

AND (t.year_date = 2021 OR t.year_date = 2022)

GROUP BY ch.channel_type

ORDER BY sum_of_income DESC;
```

Query 6: Sample query to determine ROI for specific age ranges

```
SELECT

ch.channel_type,

ch.channel_subtype,

SUM(s.quantity) sum_of_sales,

SUM (s.income) sum_of_income,

SUM(s.marketing_cost) sum_of_marketing_cost,
(SUM(s.income))/(SUM(s.marketing_cost)) as roi

FROM AT2_ADB.channel_dim ch , AT2_ADB.time_dim t, AT2_ADB.sales_fact
s

WHERE ch.channel_id = s.channel_id AND t.time_id = s.time_id AND
(t.year_date = 2021 OR t.year_date = 2022)

GROUP BY ch.channel_type, ch.channel_subtype

ORDER BY sum_of_income DESC;
```

Query 7: Sample query to determine demographic by generation

```
select case

      when cs.cust_age between 42 and 57 then 'Gen X'

      when cs.cust_age between 26 and 41 then 'Gen Y'

      when cs.cust_age between 10 and 25 then 'Gen Z'

end as generation, pr.product_category, pr.product_name,
sum(quantity)sum_of_quantity, sum(income)sum_of_income,

from at2_adb.sales_face sf, at2_adb.customer_dim cs,
at2_adb.product_dim pr

where cs.customer_id = sf.customer_id and pr.product_id =
sf.product_id and cs.cust_sex = 'Female'

group by generation, pr.product_category, pr.product_name

order by generation, sum_of_quantity desc;
```

Sample data for testing queries

```
CREATE SCHEMA AT2_ADB; -- CREATE a NEW schema for this project.

--DROP table AT2_ADB.time_dim CASCADE;

--DROP table AT2_ADB.channel_dim CASCADE;

--DROP table AT2_ADB.product_dim CASCADE;

--DROP table AT2_ADB.customer_dim CASCADE;

--DROP table AT2_ADB.sales_fact CASCADE;


CREATE TABLE AT2_ADB.time_dim (

  time_id INT NOT NULL,

  date_date DATE NOT NULL,

  month_date INT NOT NULL,

  year_date INT NOT NULL,

  PRIMARY KEY (time_id)

);


INSERT INTO AT2_ADB.time_dim

    (time_id, date_date, month_date, year_date)

VALUES
```

```
    (1,'2017-01-01', 1,2017),
    (2,'2018/01/01', 1,2018),
    (3,'2019-01-01', 1,2019),
    (4,'2020-01-01', 1,2020),
    (5,'2021-01-01', 1,2021),
    (6,'2022-01-01', 1,2022)
    ;

    CREATE TABLE AT2_ADB.product_dim (
  product_id INT NOT NULL,
  product_category VARCHAR(80) NOT NULL,
  product_group VARCHAR(80) NOT NULL,
  product_name VARCHAR(80) NOT NULL,
  PRIMARY KEY (product_id)
);

INSERT INTO AT2_ADB.product_dim
    (product_id, product_category, product_group, product_name)
VALUES
    (1,'fruit','food', 'strawberry'),
    (2,'vegetables','food', 'lettuce'),
    (3,'meat','food', 'fish'),
    (4,'soft-drinks', 'beverages', 'soda'),
    (5,'tobacco', 'other', 'camel cigarettes'),
    (6,'alcohol','beverages', 'beer'),
    (7,'baked goods', 'other', 'croissant')
    ;


 CREATE TABLE AT2_ADB.customer_dim (
  customer_id INT NOT NULL,

  cust_age INT NOT NULL,

  cust_sex VARCHAR(80) NOT NULL,

  cust_city VARCHAR(80) NOT NULL,

  cust_region VARCHAR(80) NOT NULL,

  cust_state VARCHAR(80) NOT NULL,

  cust_score DECIMAL NOT NULL,

  PRIMARY KEY (customer_id)

);


INSERT INTO AT2_ADB.customer_dim

    (customer_id,cust_age,cust_sex,cust_city,cust_region,
cust_state, cust_score)

VALUES

    (1, 22, 'Male', 'sydney','south-east','NSW', 4.5 ),

    (2, 24, 'Male', 'brisbane','north-east','QLD', 4.3),

    (3, 18, 'Male', 'melbourne','south-east','VIC', 3.7),

    (4, 21, 'Female', 'new-castle','south-east','NSW', 4.2),

    (5, 27, 'Female', 'gold-coast','north-east','QLD', 3.6),

    (6, 32, 'Female', 'ballarat','south-east','VIC', 4.8)

   ;


CREATE TABLE AT2_ADB.channel_dim (

  channel_id INT NOT NULL,

  channel_subtype VARCHAR(80) NOT NULL,

  channel_type VARCHAR(80) NOT NULL,

  PRIMARY KEY (channel_id)

);


INSERT INTO AT2_ADB.channel_dim

    (channel_id, channel_subtype, channel_type)
```

```
VALUES
    (1,'Facebook Ads', 'Paid ads'),
    (2,'Google Ads Search', 'Paid ads'),
    (3,'Google Ads Display', 'Paid ads'),
    (4,'Google Shopping', 'Paid ads'),
    (5,'Seo', 'Organic'),
    (6,'Social Media Marketing', 'Organic'),
    (7,'Email Marketing', 'Organic')
    ;

CREATE TABLE AT2_ADB.sales_fact (
  product_id INT NOT NULL,
  customer_id INT NOT NULL,
  time_id INT NOT NULL,
  channel_id INT NOT NULL,
  quantity INT NOT NULL,
  income INT NOT NULL,
  marketing_cost INT NOT NULL,
  PRIMARY KEY (product_id, customer_id, time_id, channel_id),
  FOREIGN KEY (product_id) REFERENCES
AT2_ADB.product_dim(product_id),
  FOREIGN KEY (customer_id) REFERENCES
AT2_ADB.customer_dim(customer_id),
  FOREIGN KEY (channel_id) REFERENCES
AT2_ADB.channel_dim(channel_id),
  FOREIGN KEY (time_id) REFERENCES AT2_ADB.time_dim(time_id)
);

INSERT INTO AT2_ADB.sales_fact
    (product_id, customer_id, time_id, channel_id, quantity, income,
marketing_cost)
VALUES
```

```
(1,1,1,1,10000,300000,80000),
(2,1,2,1,20000,400000,70000),
(3,1,3,2,15000,200000,50000),
(4,1,4,1,7300,350000,40000),
(5,1,5,1,3400,600000,90000),
(6,3,6,1,5900,550000,75000),
(7,3,6,2,6000,500000,72000),
(1,3,6,2,2000,450000,110000),
(2,4,6,3,1740,420000,95000),
(3,4,5,3,1200,400000,83000),
(4,4,5,4,8000,380000,78000),
(4,4,5,5,4500,600000,73000),
(5,5,6,5,6000,550000,64000),
(7,5,5,5,5000,525000,59000),
(6,5,5,6,2300,421000,52000),
(1,6,5,6,3400,580000,41000),
(2,6,6,7,8000,480000,47000),
(3,2,6,7,4500,450000,11000),
(3,2,5,1,6000,420000,97000),
(4,2,6,1,5000,530000,93000),
(7,2,5,2,2300,510000,88000),
(6,2,6,3,3400,490000,83000)
;
```