Data Warehouse Project Report

Group 15: Elaheh Bastani and Natalya Smith

COMP 810 - October 2023

Contents

1 Executive Summary **3**

2 Introduction **3**

3 Source Data Overview **3**
3.1 Datastream File 3
3.2 Masterdata File 4

4 Part A: DW Schema Design **4**
4.1 Tables' Structure 5
4.2 ETL Process 7

5 Part B: Multidimensional Data Analysis **7**
5.1 Question 1 7
5.2 Question 2 7
5.3 Question 3 8
5.4 Question 4 8
5.5 Question 5: Creating a Materialised View for Sales Analysis 8
5.6 Question 6: Comprehensive Sales Analysis Using ROLLUP and CUBE 9
5.6.1 Analysis and Insights 10
5.6.2 Overall Sales Performance and Promotion Effectiveness 11
5.6.3 Product-Specific Insights and Promotion Impact 11
5.6.4 Recommendations for Management 12
5.7 Question 7 12
5.7.1 Analysis of Monthly Sales Data 12
5.7.2 Analysis of Sales Channels Performance 13

6 Conclusion **13**

A APPENDIX **13**
A.1 SQL Codes 13
A.1.1 Building Data Warehouse 13
A.1.2 SQL Code for Question 1 15
A.1.3 SQL Code for Question 2 15
A.1.4 SQL Code for Question 3 16
A.1.5 SQL Code for Question 4 16
A.1.6 SQL Code for Question 5 16
A.1.7 SQL Code for ROLLUP Operation 17
A.1.8 SQL Code for CUBE Operation 17
A.1.9 SQL Code for Question 7 17
A.2 Figures 19
A.2.1 Result of Running the DW SQL Code 19
A.2.2 Data of Question 7 - First Analysis 19

1 Executive Summary

This project focused on designing a data warehouse for 'OfficeProducts', an Auckland CBD-based enterprise. Leveraging the Snowflake Schema, we integrated data, primarily from the Transactional Fact Table, which encompass vital sales information interconnected with products, clientele, sales avenues, and timestamps. The chosen VARCHAR for ID columns ensured an optimal mix of adaptability and efficiency. Distinct tables were established for various parameters (e.g., products and clients) to streamline data analysis. An ETL method was formulated to efficiently manage data transfers into the warehouse. The final structure was implemented in Oracle as per client stipulations. Key insights revealed that certain products, like the "Envoy Ambassador", consistently performed well in sales, indicating their market demand. Online promotions significantly improved sales, but a concerning dip in 2002 necessitates further investigation. While Direct Sales led as the primary sales channel, its dip in 2002 suggests possible market shifts that warrant attention.

2 Introduction

The project's objective was to build a Data Warehouse (DW) tailored to the 'OfficeProducts' unique needs. Using the snowflake schema, the design aimed to efficiently organise and centralise sales and product information, ensuring minimal redundancy and faster data access. This approach offers detailed insights into sales data, paving the way for informed business strategies. With this new DW design, 'OfficeProducts' can better understand its sales dynamics, strategise promotions effectively, and elevate the overall data management, setting the groundwork for leveraging data-driven insights in their operations.

'OfficeProducts', a leading supplier of office items with a vast array of products and outlets nationwide, serves diverse clients, from small businesses to large corporation. Their reputation as a trusted office supply source is built on their dedication to customer satisfaction. This commitment drives 'OfficeProducts' to consistently analyse their clients' shopping patterns, shaping their business decisions. Recently, the company observed a significant shift towards online sales due to evolving market dynamics. Therefore, the motivation for designing a DW for 'OfficeProducts' stems from the need to optimise the selling methods so the business strategies can be adjusted effectively. With the majority of sales occurring online, there is a large transactional data available that, if analysed correctly, can provide valuable insights into customer behaviour and preferences.

The DW design aims to consolidate data from various sources (i.e., both sales and product-related data). By employing a snowflake schema, the DW ensures a normalised structure that minimises data redundancy and enhances query performance. This structured approach to data management will enable 'OfficeProducts' to conduct comprehensive analyses of shopping behaviour, leading to more informed decision-making. Through the DW, 'OfficeProducts' will seek, for example, to optimise courier costs, plan promotions more effectively, and ultimately enhance the customer shopping experience. The DW design is a strategic investment in the company's future, laying the groundwork for data-driven insights and operational excellence.

3 Source Data Overview

3.1 Datastream File

This file provides transactional data with the following attributes:

Attribute	Data Type and Size
TRANSACTION_ID	VARCHAR2(8)
PRODUCT_ID	VARCHAR2(6)
CLIENT_ID	VARCHAR2(4)
CLIENT_NAME	VARCHAR2(30)
CHANNEL_ID	VARCHAR2(3)
CHANNEL_DESC	VARCHAR2(20)
DATE	DATE
QUANTITY	NUMBER(3,0)

3.2 Masterdata File

This file offers product-related master data:

Attribute	Data Type and Size
PRODUCT_ID	VARCHAR2(6)
PRODUCT_NAME	VARCHAR2(30)
SUPPLIER_ID	VARCHAR2(5)
SUPPLIER_NAME	VARCHAR2(30)
PRICE	NUMBER(5,2) DEFAULT 0.0

4 Part A: DW Schema Design

The *Snowflake Schema's Design* revolves around the transactional fact table, which holds the transactional data from the Datastream file. This fact table is linked to dimensions derived from both the Datastream and Masterdata files. The *Transactional Fact Table* is critical in this design. By capturing each sale transaction, it is intrinsically linked to product details, client data, sales channels, and sale dates. The *Entity Relationship Diagram* (ERD) in Figure 1 visually encapsulates the snowflake schema design. This diagram acts as a guide for the warehouse's structure, ensuring seamless data analysis and reporting. The fact table is at the heart of the schema, surrounded by dimension tables. The relationships between tables are forged using primary and foreign keys, such as the PRODUCT_ID link between the fact table and the DIM_PRODUCT dimension.

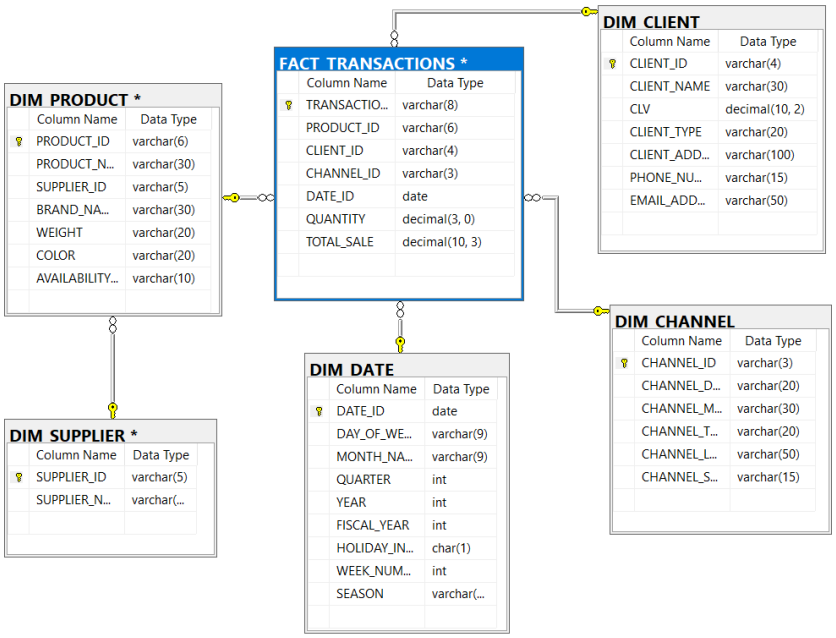


Figure 1: Entity Relationship Diagram of the Snowflake Schema

Using VARCHAR for ID columns in our database optimises storage space and supports a variety of ID types, facilitating smooth data integration from multiple sources. Modern databases efficiently handle VARCHAR indexing, promoting quick data retrieval and enhancing application user interaction. However, there are challenges: searches may be slower compared to integer columns, potential data errors could arise without proper constraints, additional storage overhead can occur, and consistent formats require meticulous maintenance. Despite these challenges, we chosen VARCHAR for its balance between flexibility, compatibility, and performance.

4.1 Tables' Structure

The Transactional Fact Table 1 encompasses attributes pertinent to each transaction. Dimension tables capture specific aspects of data:

- DIM_PRODUCT Dimension Table 2 captures product and supplier data.
- DIM_CLIENT Dimension Table 3 centralises client-related data.
- DIM_CHANNEL Dimension Table 4 describes sales channels.
- DIM_SUPPLIER Dimension Table 5 records supplier-specific information.
- DIM_DATE Dimension Table 6 assists in time-based analysis.

Table 1: Transactional Fact Table

Attribute	Data Type and Size	Description
TRANSACTION_ID	VARCHAR2(8)	Unique identifier for each transaction
PRODUCT_ID	VARCHAR2(6)	Identifier for the product involved in the transaction
CLIENT_ID	VARCHAR2(4)	Identifier for the client making the purchase
CHANNEL_ID	VARCHAR2(3)	Identifier for the sales channel used
DATE	DATE	Date of the transaction
QUANTITY	NUMBER(3,0)	Quantity of the product sold
TOTAL_SALE	NUMBER(10,2)	Total sale amount (QUANTITY * PRICE)

Table 2: DIM_PRODUCT Dimension Table

Attribute	Data Type and Size	Description
PRODUCT_ID	NUMBER	Unique identifier for the product
PRODUCT_NAME	VARCHAR2(100)	Name of the product
BRAND_NAME	VARCHAR2(50)	Brand name of the product
WEIGHT	NUMBER(5,2)	Weight of the product
COLOR	VARCHAR2(20)	Color of the product
SUPPLIER_ID	VARCHAR2(5)	Unique identifier for the supplier
AVAILABILITY_STATUS	VARCHAR2(20)	Availability status of the product

Table 3: DIM_CLIENT Dimension Table

Attribute	Data Type and Size	Description
CLIENT_ID	NUMBER	Unique identifier for the client
CLIENT_NAME	VARCHAR2(100)	Name of the client
CLV	NUMBER(10,2)	Customer Lifetime Value
CLIENT_TYPE	VARCHAR2(20)	Type of the client
CLIENT_ADDRESS	VARCHAR2(100)	Address of the client
PHONE_NUMBER	VARCHAR2(15)	Phone number of the client
EMAIL_ADDRESS	VARCHAR2(50)	Email address of the client

Table 4: DIM_CHANNEL Dimension Table

Attribute	Data Type and Size	Description
CHANNEL_ID	NUMBER	Unique identifier for the sales channel
CHANNEL_NAME	VARCHAR2(100)	Name of the sales channel
CHANNEL_MANAGER	VARCHAR2(30)	Manager of the sales channel
CHANNEL_TYPE	VARCHAR2(20)	Type of the sales channel
CHANNEL_LOCATION	VARCHAR2(50)	Location of the sales channel
CHANNEL_STATUS	VARCHAR2(20)	Status of the sales channel

Table 5: DIM_SUPPLIER Dimension Table

Attribute	Data Type and Size	Description
SUPPLIER_ID	VARCHAR2(5)	Unique identifier for each supplier
SUPPLIER_NAME	VARCHAR2(30)	Name of the supplier

Table 6: DIM_DATE Dimension Table

Attribute	Data Type and Size	Description
DATE_ID	NUMBER	Unique identifier for the date
DAY_OF_WEEK	VARCHAR2(9)	Day of the week
MONTH_NAME	VARCHAR2(9)	Name of the month
QUARTER	NUMBER(1)	Quarter of the year
YEAR	NUMBER(4)	Year
FISCAL_YEAR	NUMBER(4)	Fiscal year
HOLIDAY_INDICATOR	VARCHAR2(1)	Indicator for holidays
WEEK_NUMBER	NUMBER(2)	Week number of the year
SEASON	VARCHAR2(10)	Season of the year

4.2 ETL Process

The ETL process will:

- Extract data from the Datastream and Masterdata files.
- Join the data using the PRODUCT_ID column.
- Load these columns into the warehouse: PRODUCT_ID, CLIENT_ID, CLIENT_NAME, CHANNEL_ID, CHANNEL_DESC, DATE, QUANTITY, PRODUCT_NAME, SUPPLIER_ID, SUPPLIER_NAME, and TOTAL_SALE (QUANTITY $\times$ PRICE).

5 Part B: Multidimensional Data Analysis

5.1 Question 1

Top three countries with the highest 'total sales' are shown in Figure 2. For the SQL code used in this analysis, refer to Appendix A, Section A.1.2.

	COUNTRY_ISO_CODE	COUNTRY_NAME	TOTAL_SALES
1	US	United States of America	52910773.15
2	DE	Germany	9210129.22
3	JP	Japan	7207880.09

Figure 2

5.2 Question 2

The most sold products in the US for each year in 1998-2001 are illustrated in Figure 3. The SQL code is in Appendix A, Section A.1.3.

	YEAR	PROD_NAME	TOTAL_QUANTITY
1	1998	O/S Documentation Set - English	4729
2	1999	Mouse Pad	4603
3	2000	1.44MB External 3.5" Diskette	4353
4	2001	Keyboard Wrist Rest	4587

Figure 3

5.3 Question 3

The number of sales transactions recorded for the product with the highest sales revenue in 2001, accompanied by the channel name and number of items sold, can be observed in Figure 4. The SQL code is in Appendix A, Section A.1.4.

	PROD_ID	CALENDAR_YEAR	CHANNEL_ID	CHANNEL_DESC	NUM_TRANS	TOTAL_QUANTITY
1	18	2001	3	Direct Sales	1198	1198

Figure 4

5.4 Question 4

The three countries with the lowest sales performance in 1998 are illustrated in Figure 5. The SQL code is in Appendix A, Section A.1.5.

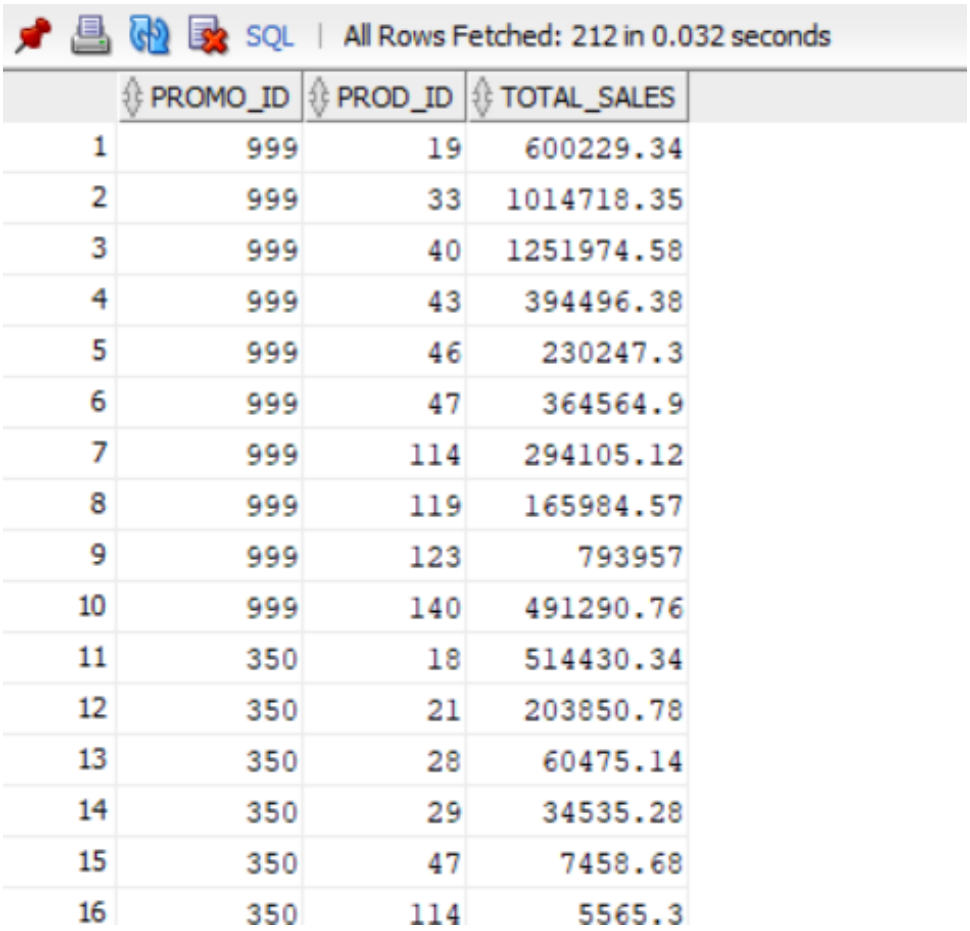
	COUNTRY_ISO_CODE	COUNTRY_NAME	SALES
1	PL	Poland	80.97
2	NZ	New Zealand	270.93
3	CN	China	1267.91

Figure 5

5.5 Question 5: Creating a Materialised View for Sales Analysis

This section discusses the creation of a materialised view[1] to facilitate efficient and faster sales analysis. In our case, the materialised view consists of a total of 212 rows (refer to to Appendix A.1.6 for the SQL code used to create the materialised view).

[1]That store data based on remote tables and are used to increase query performance on aggregated data, especially for large datasets.



	PROMO_ID	PROD_ID	TOTAL_SALES
1	999	19	600229.34
2	999	33	1014718.35
3	999	40	1251974.58
4	999	43	394496.38
5	999	46	230247.3
6	999	47	364564.9
7	999	114	294105.12
8	999	119	165984.57
9	999	123	793957
10	999	140	491290.76
11	350	18	514430.34
12	350	21	203850.78
13	350	28	60475.14
14	350	29	34535.28
15	350	47	7458.68
16	350	114	5565.3

Figure 6: Partial View of the Materialised View Data

5.6 Question 6: Comprehensive Sales Analysis Using ROLLUP and CUBE

We next provide an in-depth analysis of sales data derived from the "Promotion_Analysis_mv" materialised view, utilising both ROLLUP and CUBE operations to uncover trends and patterns in product sales across various promotions. The objective is to identify top-selling products, assess the impact of promotions, and offer strategic insights for future sales initiatives.

The *ROLLUP Operation* provides a way to perform aggregate data analysis, moving from the most detailed level to a grand total, and it produced a total of 215 records for our analysis. The SQL code for this operation can be found in Appendix A.1.7. Figure 7 displays the first few rows of the result set obtained from the ROLLUP operation, showing the aggregated sales data at various levels.

SQL | All Rows Fetched: 215 in 0.112 seconds

	PROMO_NAME	PROD_NAME	TOTAL_SALES
1	NO PROMOTION #	1.44MB External 3.5" Diskette	212584.08
2	NO PROMOTION #	128MB Memory Card	562744.8
3	NO PROMOTION #	17" LCD w/built-in HDTV Tuner	6832364.4
4	NO PROMOTION #	18" Flat Panel Graphics Monitor	5041656.11
5	NO PROMOTION #	256MB Memory Card	578184.52
6	NO PROMOTION #	3 1/2" Bulk diskettes, Box of 100	349138.25
7	NO PROMOTION #	3 1/2" Bulk diskettes, Box of 50	240620.66
8	NO PROMOTION #	5MP Telephoto Digital Camera	6068447.58
9	NO PROMOTION #	64MB Memory Card	27933.33
10	NO PROMOTION #	8.3 Minitower Speaker	3767390.16
11	NO PROMOTION #	Adventures with Numbers	170085.95
12	NO PROMOTION #	Bounce	240871.68
13	NO PROMOTION #	CD-R Mini Discs	374796.25

Figure 7: Partial Results of ROLLUP Operation

The CUBE Operation allows data analysis across multiple dimensions, providing a comprehensive view of the data. This operation is more granular than ROLLUP, producing a total of 286 records for our analysis. The SQL code for this operation can be found in Appendix A.1.8. Figure 8 displays the first few rows of the result set obtained from the CUBE operation, showcasing the multidimensional analysis of sales data.

SQL | All Rows Fetched: 286 in 0.96 seconds

	PROMO_NAME	PROD_NAME	TOTAL_SALES
1	NO PROMOTION #	1.44MB External 3.5" Diskette	212584.08
2	NO PROMOTION #	128MB Memory Card	562744.8
3	NO PROMOTION #	17" LCD w/built-in HDTV Tuner	6832364.4
4	NO PROMOTION #	18" Flat Panel Graphics Monitor	5041656.11
5	NO PROMOTION #	256MB Memory Card	578184.52
6	NO PROMOTION #	3 1/2" Bulk diskettes, Box of 100	349138.25
7	NO PROMOTION #	3 1/2" Bulk diskettes, Box of 50	240620.66
8	NO PROMOTION #	5MP Telephoto Digital Camera	6068447.58
9	NO PROMOTION #	64MB Memory Card	27933.33
10	NO PROMOTION #	8.3 Minitower Speaker	3767390.16
11	NO PROMOTION #	Adventures with Numbers	170085.95
12	NO PROMOTION #	Bounce	240871.68
13	NO PROMOTION #	CD-R Mini Discs	374796.25

Figure 8: Partial Results of CUBE Operation

The difference in the number of records produced by ROLLUP (215 records) and CUBE (286 records) is attributed to their distinct functionalities. ROLLUP generates subtotals that roll up from the most detailed level to a grand total, whereas CUBE provides all possible combinations of aggregation, including the cross-tabulations, offering a more detailed and comprehensive view of the data. This additional granularity in the CUBE operation results in more records and provides valuable insights for multidimensional data analysis, crucial for making informed decisions.

5.6.1 Analysis and Insights

To gain deeper insights into the sales performance of different products under various promotions, we next take a closer look at the top 3 products in each promotion. The tables below highlight these top-performing products, providing valuable information for strategic decision-making and future sales initiatives.

Table 7 shows the top three products for *TV Promotion* #13-351 with total sales of $1,224,503.26. Table 8 details the top three products under *Internet Promotion* #29-350 with total sales of $2,199,380.90. Table 9 lists the top three products under *Post Promotion* #20-33 with total sales of $277,426.26. Lastly, the top three products with *No Promotion*, with total sales of $94,504,520.79, are illustrated in Table 10.

Table 7: Top 3 Products under TV Promotion #13-351

Rank	Product Name	Sales
1	Mini DV Camcorder with 3.5" Swivel LCD	$128,837.36
2	Envoy Ambassador	$91,276.10
3	Home Theatre Package with DVD-Audio/Video Play	$79,869.03

Table 8: Top 3 Products under Internet Promotion #29-350

Rank	Product Name	Sales
1	Envoy Ambassador	$514,430.34
2	Mini DV Camcorder with 3.5" Swivel LCD	$153,137.82
3	Unix/Windows 1-user pack	$60,475.14

Table 9: Top 3 Products under Post Promotion #20-33

Rank	Product Name	Sales
1	SIMM- 16MB PCMCIAII card	$32,697.82
2	18" Flat Panel Graphics Monitor	$204,297.73
3	Multimedia speakers- 3" cones	$5,938.68

Table 10: Top 3 Products without Promotion

Rank	Product Name	Sales
1	Envoy Ambassador	$14,405,936.08
2	Mini DV Camcorder with 3.5" Swivel LCD	$8,032,840.22
3	17" LCD w/built-in HDTV Tuner	$6,832,364.4

5.6.2 Overall Sales Performance and Promotion Effectiveness

The CUBE operation has facilitated a holistic view of the sales data, revealing a total of approximately $98.2 million in sales across all products and promotions. The "NO PROMOTION" category, as highlighted by the ROLLUP operation, accounts for a substantial $94.5 million of the total sales, underscoring the effectiveness of selling products without promotions. In contrast, specific promotions such as "TV promotion #13-351", "internet promotion #29-350", and "post promotion #20-33" have yielded sales of $1.2 million, $2.2 million, and $277,000 respectively. This stark disparity suggests that while promotions have a role in boosting sales, a significant portion of revenue is generated from products sold at regular prices.

5.6.3 Product-Specific Insights and Promotion Impact

The CUBE operation has provided valuable insights into the performance of individual products across different promotions. For instance, the "Envoy Ambassador" and "Mini DV Camcorder with 3.5" Swivel LCD" have emerged as top-performing products, generating substantial sales both under promotions and without any promotions. This indicates their inherent popularity and market demand. On the other hand, the ROLLUP operation has highlighted the effectiveness of "internet promotion #29-350" in driving sales for these products, suggesting that online channels are a potent medium for promoting these items.

5.6.4 Recommendations for Management

Based on the insights derived from both ROLLUP and CUBE operations, management should consider the following strategies:

1. Because products such as "Envoy Ambassador" and "Mini DV Camcorder with 3.5" Swivel LCD" have demonstrated strong sales performance, bundling these popular products with less popular items could potentially increase overall sales and product visibility.

2. Since the effectiveness of promotions varies across different products, tailoring promotions to suit the characteristics of specific products, especially those that respond positively to online promotions, could enhance sales effectiveness.

3. The "NO PROMOTION" category has shown to be highly effective, contributing significantly to the total sales. Therefore, analysing the characteristics of products sold without promotions could provide insights into customer preferences and purchasing behaviour, guiding future product development and marketing strategies.

4. Because the "post promotion #20-33" has demonstrated a relatively low total sales amount of $277,426.26, suggesting that this promotion might not be as effective as others, delving deeper into this promotion's strategy to understand the factors contributing to its underperformance and take corrective actions to enhance its effectiveness can be beneficial.

5.7 Question 7

The `sh.fweek_pscat_sales_mv` materialised view in the SH schema aggregates sales data on a weekly basis, including various dimensions such as product subcategory, sales channel, promotional information, alongside the sales amount in dollars. This view is critical for conducting diverse sales analyses, rendering actionable insights for management to drive data-driven decision-making processes.

5.7.1 Analysis of Monthly Sales Data

The sales data from January 1998 to January 2002, detailed in Appendix A.2.2, unveils critical insights into the sales trends over time. The SQL code utilised to derive this monthly sales data is provided in Appendix A.1.9. Our observations from this data can be summarised as follows:

- Sales figures exhibit variability across different months in 1998. For instance, sales in February amounted to approximately $2.17 million, a substantial increase from the $1.85 million in January. Conversely, sales dropped to $1.90 million in April from $2.29 million in March. This could be indicative of seasonal buying behaviours or the impact of marketing initiatives.

- A progressive increase in annual sales from 1998 to 2001 is notable, peaking at around $28 million in 2001. However, a sharp decline to $57,057.99 in January 2002 warrants further investigation to uncover the underlying factors.

- Certain months like March, August, and December consistently show higher sales figures. For instance, sales in March 1998, August 1998, and December 1998 were approximately $2.29 million, $2.23 million, and $1.45 million respectively, suggesting these months could be opportune for launching promotional campaigns or new products.

5.7.2 Analysis of Sales Channels Performance

The SQL code utilised for this analysis is provided in Appendix A.1.9. The analysis of sales channels performance over the years reveals a distinct dominance of the Direct Sales channel over others like Partners, Internet, and Tele Sales, as can be observed in Table 11, reinforcing its vital role in driving the company's revenue. This consistency underscores the efficiency and effectiveness of the Direct Sales channel in reaching and engaging customers.

Table 11: Yearly sales through Direct Sales channel

Year	Channel ID	Channel Description	Channel Class	Total Sales (USD)
1998	3	Direct Sales	Direct	$108,229,056.75
1999	3	Direct Sales	Direct	$102,744,777.50
2000	3	Direct Sales	Direct	$100,236,684.42
2001	3	Direct Sales	Direct	$93,519,934.16
2002	3	Direct Sales	Direct	$199,113.36

While the Direct Sales channel shows robust sales figures in 1998-2001, there is a significant dip in sales in 2002, to approximately $199,113. This drop in sales calls for a detailed analysis to understand and explain the reasons and develop plans to boost sales through this channel. The significant dominance of the Direct Sales channel compared to others suggests a thorough investigation to understand the reasons for its success. Indeed, understanding these factors could offer valuable insights to optimise other sales channels and enhance overall sales performance.

6 Conclusion

The project was centered on creating a data warehouse design for 'OfficeProducts', a company located in Auckland CBD. We utilised the Snowflake Schema, with the Transactional Fact Table at its core, drawing data from Datastream and Masterdata files. This table records every sale, connecting to details about products, clients, sales channels, and dates. We selected VARCHAR for ID columns, offering a good mix of flexibility and efficiency. We set up distinct tables for various dimensions like products, clients, channels, suppliers, and dates. Additionally, we defined an ETL process to pull, combine, and input the required data into the warehouse. Ultimately, our goal was to set up these tables in Oracle, meeting the client's specifications.

Several key findings emerged as a result. In terms of product performance, products like "Envoy Ambassador" and "Mini DV Camcorder with 3.5" Swivel LCD" stood out with steady sales, showing they are in high demand. In terms of promotion and sales dynamics, online-focused promotions hugely boost sales. Sales trends from 1998 to 2002 offered valuable insights. Notably, the drop in 2002 calls for a closer look. Direct Sales led the way among sales channels. However, its noticeable decline in 2002 points to possible market challenges or changes.

A APPENDIX

A.1 SQL Codes

A.1.1 Building Data Warehouse

```
-- Drop the tables if they exist (to prevent errors if re-running the script)
\begin{figure}
    \centering
    \includegraphics[width=0.5\linewidth]{Q1_TaskResult.png}
```

```
    \caption{Enter Caption}
    \label{fig:enter-label}
\end{figure}
DROP TABLE FACT_TRANSACTIONS CASCADE CONSTRAINTS;
DROP TABLE DIM_PRODUCT CASCADE CONSTRAINTS;
DROP TABLE DIM_CLIENT CASCADE CONSTRAINTS;
DROP TABLE DIM_CHANNEL CASCADE CONSTRAINTS;
DROP TABLE DIM_DATE CASCADE CONSTRAINTS;
DROP TABLE DIM_SUPPLIER CASCADE CONSTRAINTS;

-- Create the FACT_TRANSACTIONS fact table


-- Create the DIM_PRODUCT dimension table
CREATE TABLE DIM_PRODUCT (
    PRODUCT_ID VARCHAR2(6) PRIMARY KEY,
    PRODUCT_NAME VARCHAR2(30),
    SUPPLIER_ID VARCHAR2(5),
    PRICE NUMBER(5,2),
    BRAND_NAME VARCHAR2(30),
    WEIGHT VARCHAR2(20),
    COLOR VARCHAR2(20),
    AVAILABILITY_STATUS VARCHAR2(10)
);

-- Create the DIM_CLIENT dimension table with additional columns
CREATE TABLE DIM_CLIENT (
    CLIENT_ID VARCHAR2(4) PRIMARY KEY,
    CLIENT_NAME VARCHAR2(30),
    CLV NUMBER(10, 2),                  -- Client Lifetime Value
    CLIENT_TYPE VARCHAR2(20),           -- Client Type
    CLIENT_ADDRESS VARCHAR2(100),       -- Client Address
    PHONE_NUMBER VARCHAR2(15),          -- Phone Number
    EMAIL_ADDRESS VARCHAR2(50)          -- Email Address
);

-- Create the DIM_CHANNEL dimension table with additional columns
CREATE TABLE DIM_CHANNEL (
    CHANNEL_ID VARCHAR2(3) PRIMARY KEY,
    CHANNEL_DESC VARCHAR2(20),
    CHANNEL_MANAGER VARCHAR2(30),    -- Channel Manager
    CHANNEL_TYPE VARCHAR2(20),       -- Channel Type
    CHANNEL_LOCATION VARCHAR2(50),   -- Channel Location
    CHANNEL_STATUS VARCHAR2(15)      -- Channel Status
);

-- Create the DIM_DATE dimension table with additional columns
CREATE TABLE DIM_DATE (
    DATE_ID DATE PRIMARY KEY,
    DAY_OF_WEEK VARCHAR2(9),         -- Day of the Week
    MONTH_NAME VARCHAR2(9),          -- Month Name
```

```
    QUARTER NUMBER(1),                  -- Quarter
    YEAR NUMBER(4),                     -- Year
    FISCAL_YEAR NUMBER(4),              -- Fiscal Year
    HOLIDAY_INDICATOR VARCHAR2(1),      -- Holiday Indicator (Y/N)
    WEEK_NUMBER NUMBER(2),              -- Week Number
    SEASON VARCHAR2(15)                 -- Season
);

CREATE TABLE FACT_TRANSACTIONS (
    TRANSACTION_ID VARCHAR2(8) PRIMARY KEY,
    PRODUCT_ID VARCHAR2(6),
    CLIENT_ID VARCHAR2(4),
    CHANNEL_ID VARCHAR2(3),
    DATE_ID DATE,
    QUANTITY NUMBER(3,0),
    TOTAL_SALE NUMBER(10,3),
    FOREIGN KEY (PRODUCT_ID) REFERENCES DIM_PRODUCT(PRODUCT_ID),
    FOREIGN KEY (CLIENT_ID) REFERENCES DIM_CLIENT(CLIENT_ID),
    FOREIGN KEY (CHANNEL_ID) REFERENCES DIM_CHANNEL(CHANNEL_ID),
    FOREIGN KEY (DATE_ID) REFERENCES DIM_DATE(DATE_ID)
);

-- Create indexes on foreign keys for optimization
CREATE INDEX IDX_FK_PRODUCT_ID ON FACT_TRANSACTIONS(PRODUCT_ID);
CREATE INDEX IDX_FK_CLIENT_ID ON FACT_TRANSACTIONS(CLIENT_ID);
CREATE INDEX IDX_FK_CHANNEL_ID ON FACT_TRANSACTIONS(CHANNEL_ID);
CREATE INDEX IDX_FK_DATE ON FACT_TRANSACTIONS(DATE_ID);


-- Repeat for other tables and SQL commands
```

A.1.2 SQL Code for Question 1

```
WITH CountrySales AS (
  SELECT c.country_iso_code, c.country_name, SUM(s.amount_sold) AS total_sales
  FROM SH.COUNTRIES c
  JOIN SH.CUSTOMERS cust ON c.country_id = cust.country_id
  JOIN SH.SALES s ON cust.cust_id = s.cust_id
  GROUP BY c.country_iso_code, c.country_name)
SELECT country_iso_code, country_name, total_sales
FROM (
  SELECT country_iso_code, country_name, total_sales,
         RANK() OVER (ORDER BY total_sales DESC) AS ranking
         FROM CountrySales)
WHERE ranking <= 3;
```

A.1.3 SQL Code for Question 2

```
WITH RankedProducts AS (
  SELECT t2.calendar_year AS year,
```

```
        p2.prod_name,
        SUM(s2.quantity_sold) AS total_quantity,
        RANK() OVER (PARTITION BY t2.calendar_year ORDER BY SUM(s2.quantity_sol
  FROM SH.TIMES t2
  JOIN SH.SALES s2 ON t2.time_id = s2.time_id
  JOIN SH.CUSTOMERS c2 ON s2.cust_id = c2.cust_id
  JOIN SH.PRODUCTS p2 ON s2.prod_id = p2.prod_id
  JOIN SH.COUNTRIES co2 ON c2.country_id = co2.country_id
  WHERE co2.country_iso_code = 'US'
    AND t2.calendar_year BETWEEN '1998' AND '2001'
  GROUP BY t2.calendar_year, p2.prod_name)
SELECT year, prod_name, total_quantity
FROM RankedProducts
WHERE prod_rank = 1;
```

A.1.4 SQL Code for Question 3

```
WITH MaxRevenueProduct AS (
  SELECT s1.prod_id, t1.calendar_year, s1.channel_id, c1.channel_desc,
        COUNT(*) AS num_trans, SUM(s1.quantity_sold) AS total_quantity,
        RANK() OVER (PARTITION BY t1.calendar_year ORDER BY SUM(s1.amount_sold)
  FROM SH.SALES s1
  JOIN SH.TIMES t1 ON s1.time_id = t1.time_id
  JOIN SH.CHANNELS c1 ON s1.channel_id = c1.channel_id
  WHERE t1.calendar_year = '2001'
  GROUP BY s1.prod_id, t1.calendar_year, s1.channel_id, c1.channel_desc)
SELECT m.prod_id, m.calendar_year, m.channel_id, m.channel_desc, m.num_trans, m.
FROM MaxRevenueProduct m
WHERE m.prod_rank = 1;
```

A.1.5 SQL Code for Question 4

```
WITH CountrySales AS (
  SELECT c.country_iso_code, c.country_name, SUM(s.amount_sold) AS total_sales
  FROM SH.COUNTRIES c
  JOIN SH.CUSTOMERS cust ON c.country_id = cust.country_id
  JOIN SH.SALES s ON cust.cust_id = s.cust_id
  JOIN SH.TIMES t ON s.time_id = t.time_id
  WHERE t.calendar_year = '1998'
  GROUP BY c.country_iso_code, c.country_name
)
SELECT country_iso_code, country_name, total_sales AS sales
FROM (
  SELECT country_iso_code, country_name, total_sales,
        DENSE_RANK() OVER (ORDER BY total_sales) AS ranking
        FROM CountrySales)
WHERE ranking <= 3;
```

A.1.6 SQL Code for Question 5

```
DROP MATERIALIZED VIEW Promotion_Analysis_mv;

CREATE MATERIALIZED VIEW Promotion_Analysis_mv
BUILD IMMEDIATE
REFRESH COMPLETE
AS
SELECT   s.PROMO_ID, s.PROD_ID, SUM(s.AMOUNT_SOLD) AS TOTAL_SALES
FROM     SH.SALES s
GROUP BY      s.PROMO_ID,      s.PROD_ID;
```

A.1.7 SQL Code for ROLLUP Operation

```
--Rollup
SELECT
     p.PROMO_NAME,
     pr.PROD_NAME,
    SUM(mv.TOTAL_SALES) AS TOTAL_SALES
FROM
     Promotion_Analysis_mv mv
JOIN
    SH.PRODUCTS pr ON mv.PROD_ID = pr.PROD_ID
JOIN
    SH.PROMOTIONS p ON mv.PROMO_ID = p.PROMO_ID
GROUP BY
    ROLLUP (p.PROMO_NAME, pr.PROD_NAME)
ORDER BY
     p.PROMO_NAME, pr.PROD_NAME;
```

A.1.8 SQL Code for CUBE Operation

```
--CUBE
SELECT
     p.PROMO_NAME,
     pr.PROD_NAME,
    SUM(mv.TOTAL_SALES) AS TOTAL_SALES
FROM
     Promotion_Analysis_mv mv
JOIN
    SH.PRODUCTS pr ON mv.PROD_ID = pr.PROD_ID
JOIN
    SH.PROMOTIONS p ON mv.PROMO_ID = p.PROMO_ID
GROUP BY
    CUBE (p.PROMO_NAME, pr.PROD_NAME)
ORDER BY
     p.PROMO_NAME, pr.PROD_NAME;
```

A.1.9 SQL Code for Question 7

```
SELECT EXTRACT(YEAR FROM WEEK_ENDING_DAY) AS sales_year,
       TO_CHAR(WEEK_ENDING_DAY, 'Month') AS sales_month,
```

```
        SUM(DOLLARS) AS total_sales
FROM sh.fweek_pscat_sales_mv
GROUP BY ROLLUP(EXTRACT(YEAR FROM WEEK_ENDING_DAY),
                TO_CHAR(WEEK_ENDING_DAY, 'Month'))
ORDER BY sales_year, TO_DATE(sales_month, 'Month');

SELECT EXTRACT(YEAR FROM WEEK_ENDING_DAY) AS sales_year,
       TO_CHAR(WEEK_ENDING_DAY, 'Month') AS sales_month,
       SUM(DOLLARS) AS total_sales
FROM sh.fweek_pscat_sales_mv
GROUP BY ROLLUP(EXTRACT(YEAR FROM WEEK_ENDING_DAY),
                TO_CHAR(WEEK_ENDING_DAY, 'Month'))
ORDER BY sales_year, TO_DATE(sales_month, 'Month');
```

A.2 Figures

A.2.1 Result of Running the DW SQL Code

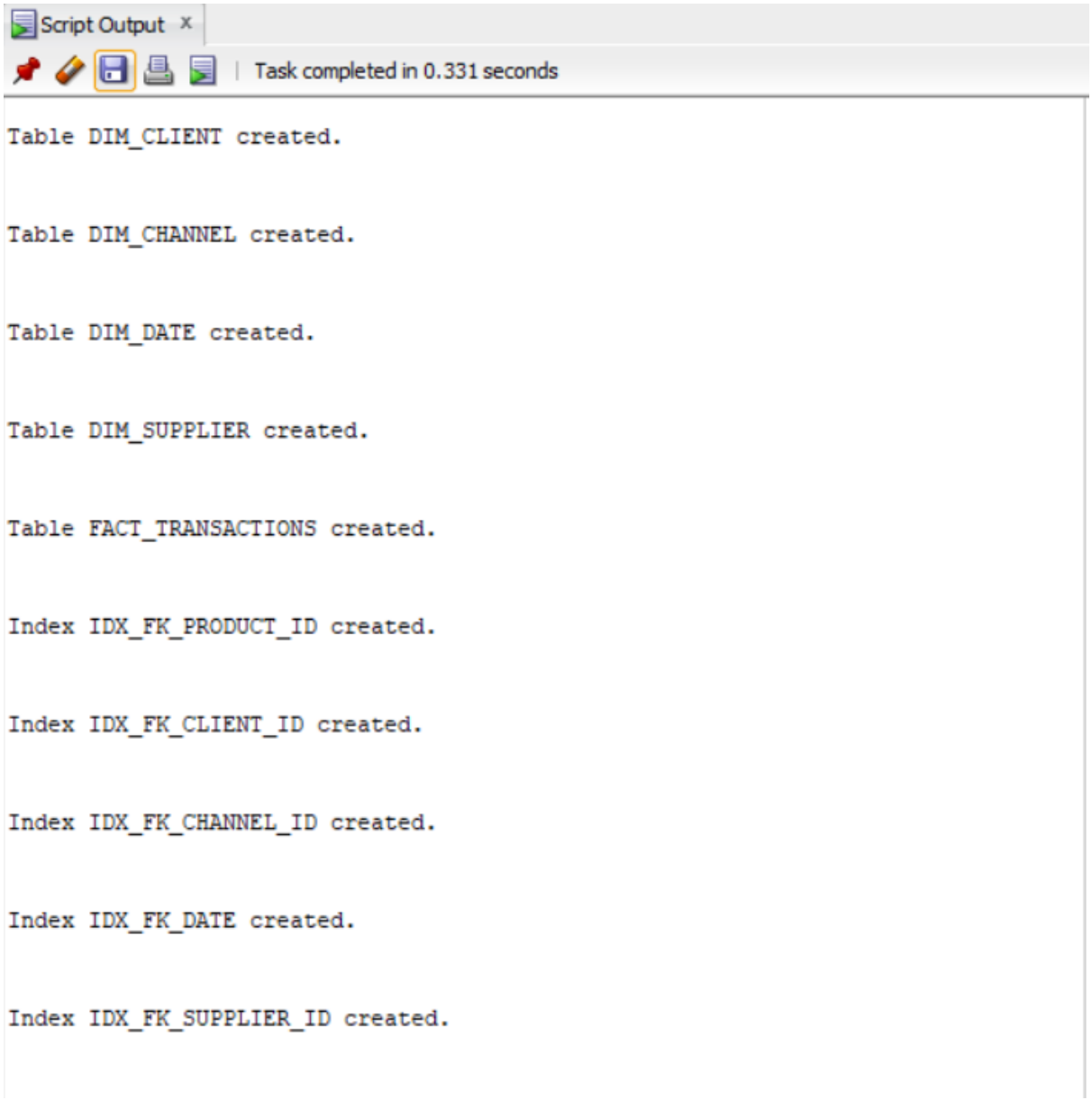


Figure 9: Result of running the SQL code for data warehouse creation

A.2.2 Data of Question 7 - First Analysis

Sales Year	Sales Month	Total Sales (USD)
1998	January	1,857,082.52
1998	February	2,172,231.69
1998	March	2,297,053.47
1998	April	1,901,279.78
1998	May	1,977,302.07
1998	June	1,807,523.24
1998	July	1,693,597.87
1998	August	2,233,923.97

Sales Year	Sales Month	Total Sales (USD)
1998	September	2,117,772.19
1998	October	1,921,230.06
1998	November	2,297,374.63
1998	December	1,455,263.64
1998		23,731,635.13
1999	January	2,429,719.58
1999	February	2,357,629.26
1999	March	1,574,025.17
1999	April	1,430,821.53
1999	May	1,899,666.39
1999	June	1,615,131.62
1999	July	1,732,518.11
1999	August	2,070,300.15
1999	September	1,959,098.21
1999	October	1,852,254.27
1999	November	1,665,733.48
1999	December	1,619,220.2
1999		22,206,117.97
2000	January	2,336,003.02
2000	February	2,102,455.69
2000	March	1,509,297.31
2000	April	2,168,753.1
2000	May	1,522,047.78
2000	June	1,687,211.33
2000	July	2,291,921.55
2000	August	1,693,985.86
2000	September	2,087,783.98
2000	October	2,268,190.23
2000	November	2,172,533.94
2000	December	2,291,432.34
2000		24,131,616.13
2001	January	1,886,848.24
2001	February	2,105,400.36
2001	March	2,060,272.92
2001	April	2,758,319.11
2001	May	2,021,133.15
2001	June	2,024,578.47
2001	July	2,835,022.52
2001	August	2,237,122.84
2001	September	2,736,866.85
2001	October	2,360,350.54
2001	November	2,213,144.32
2001	December	2,840,344.67
2001		28,079,403.99
2002	January	57,057.99
2002		57,057.99
		98,205,831.21