2024

Year Project

ITRI 611

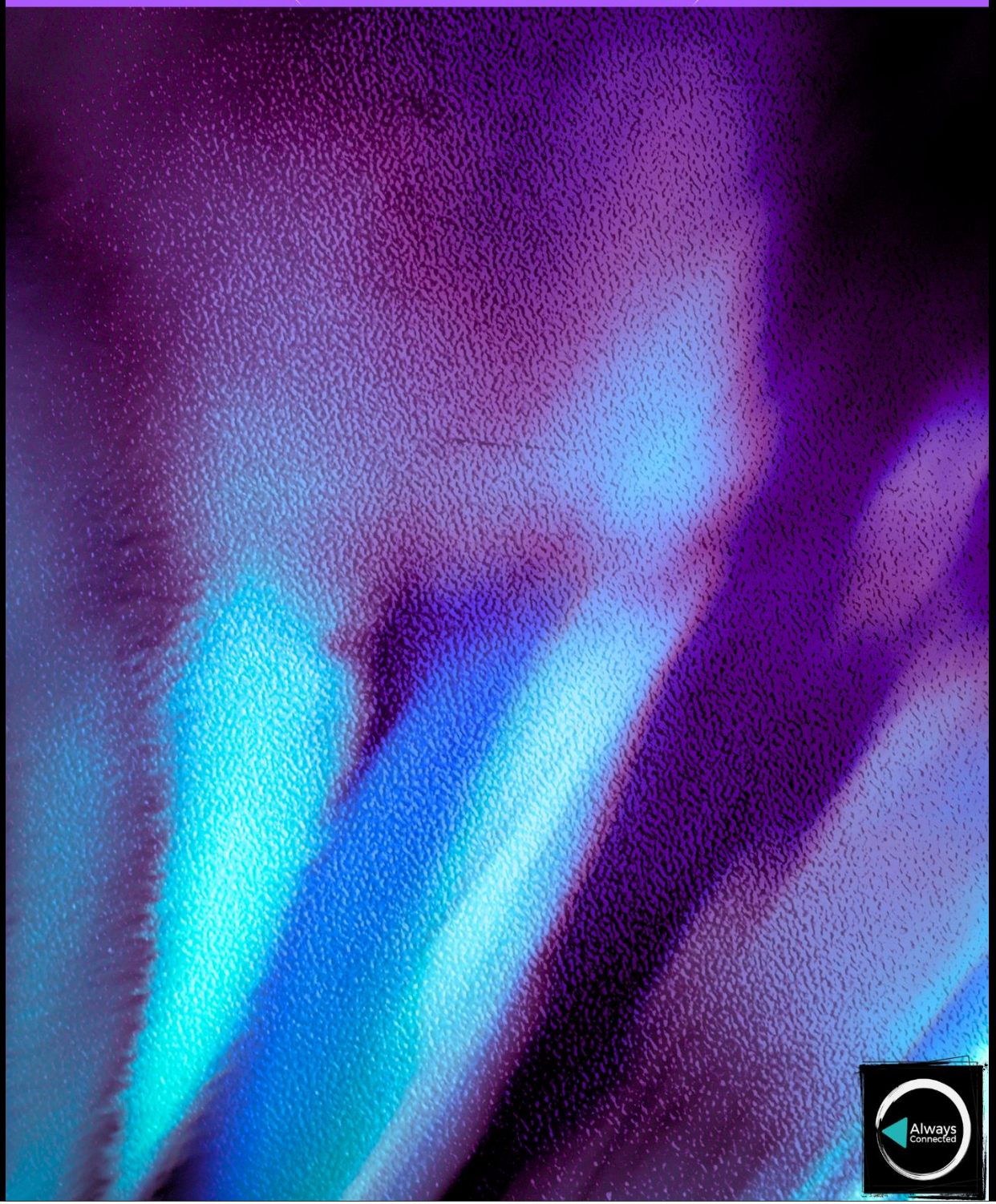

Table of Contents

Table of Figures

Data Warehouse Developers

Brain Du Plessis: 37776983

Dewald Oosthuizen: 38336529

ETec Data Warehouse Development

This document outlines the development process of a Data Warehouse specifically designed to address the requirements of ETec's management. Furthermore, the development process is described by delving into all aspects of the development process, including the requirements analysis, conceptual design, logical design, and physical design. Additionally, the selected database management system will be described as well as a tailored indexing strategy.

Moreover, this document includes all information related to the Data Warehouse, including all development notes, the ETL specification documentation, and information presentation strategies. Nonetheless, this document ensures the maintainability of the developed system by thoroughly describing aspects such as source-to-target mapping and all transformation logic.

Problem Statement

The goal of this project is to create a new Data Mart for ETec, with the purpose of fulfilling their needs for sales reporting and querying. ETec currently primarily emphasises sales performance statistics, however, there is an expected strategic transition towards metrics that are centred around suppliers.

The system must be capable of managing various sales channels, such as wholesalers, retailers, and direct customers, and be able to handle products kept in multiple national warehouses. Furthermore, it must be capable of accommodating a hierarchical structure for product categories. The task at hand is to create and execute a solution that effectively combines these objectives in order to improve decision-making processes and adjust to future strategic directions.

Key Stakeholders

The key stakeholders involved in this project include the ETec Management Team, the ETec IT Department, the Sales Department, the Finance Department, and the Supply Chain Department.

Business Requirements

This section will list all business requirements identified after a thorough analysis of the problem statement. These requirements are separated into functional and non-functional requirements.

Functional Requirements

Functional requirements specify the precise behaviours or functions that the system must be capable of executing:

- Diverse Report Production
  - The system should facilitate the generation of diverse reports that provide detailed sales data. This include reports that are generated on a daily, weekly, monthly, quarterly, and annual basis.
- Versatile Query Capabilities
  - The system should provide versatile querying functionalities that may adjust to the evolving information requirements of management.
- Integration of Supplier-Focused Data
  - The system should include the ability to incorporate data that specifically emphasises suppliers, in line with the planned transition towards supplier-oriented analytics.
- Integration of Atomic Conformed Dimensions
  - Introduce atomic conformed dimensions in the data warehouse to guarantee the smooth integration of data from different sources, enabling uniform and consistent reporting.
- Aggregated View of Sales Data
  - Offer features that enable the generation of unified perspectives on combined sales data to assist in the examination of sales trends based on various factors such as product categories, geographic regions, and sales channels.

Non-Functional Requirements

Non-functional requirements pertain to the operation or qualities of the system that are necessary to support the functional requirements:

- Optimization of Data Mart Query Performance

Optimise query performance for the data mart to provide efficient data retrieval procedures, which is essential for facilitating extensive data analysis and reporting.

Data Management Requirements

The source data for the proposed data mart solution, comes from the existing operational system. The source data consists of a singular relational database which contains the entities specified in Appendix A.

ETL Processes

Upon analysing the data in the source system, it is evident that extraction, transformation, and loading techniques will be necessary to ensure that the data is standardised to a suitable level for its intended use. The source data will have to be subjected to several procedures, such as correcting

inaccurate data points, removing rows with null values, breaking down data points with multiple values, and loading the data into specific dimensions.

Query Performance

As the firm transitions to a supplier-centric model, it is poised to handle a significantly larger volume of data. Managing this anticipated increase effectively requires prioritising scalability in our data mart architecture. Improving scalability will help the system maintain efficiency and responsiveness as data volumes grow, supporting the future strategic shift without performance drawbacks. This focus on scalability will assist in handling future growth and ensuring optimal query response times.

Query Capabilities

The proposed data mart should be designed with advanced capabilities to support ad hoc, complex, and aggregation queries, ensuring that the management of ETec can extract comprehensive insights and maximum value from the system. It allows managers to analyse sales data trends, performance metrics, and other important indicators across different dimensions of the business. Aggregation queries are particularly important as sales data is typically aggregated to provide daily, weekly, monthly, yearly, and quarterly sales performance insights.

Reporting

The reports that need to be supported include aggregated views of sales performance, which should allow for filtering based on numerous aspects, such as suppliers, customers, products, and regions, just to name a few. Furthermore, to address the requirement of ETec's management to support future supplier-oriented data views, the systems should allow various sales insights to be derived as they pertain to different suppliers.

To ensure the longevity of the data mart, users should be allowed to initiate reports. This is achieved by modelling business processes with a business-oriented focus instead of modelling the data mart to accommodate specific reports in a manner that is data-focused.

Proposed Solution

Once all the essential requirements for the new system were gathered, a conceptual solution was devised to fulfil these requirements.

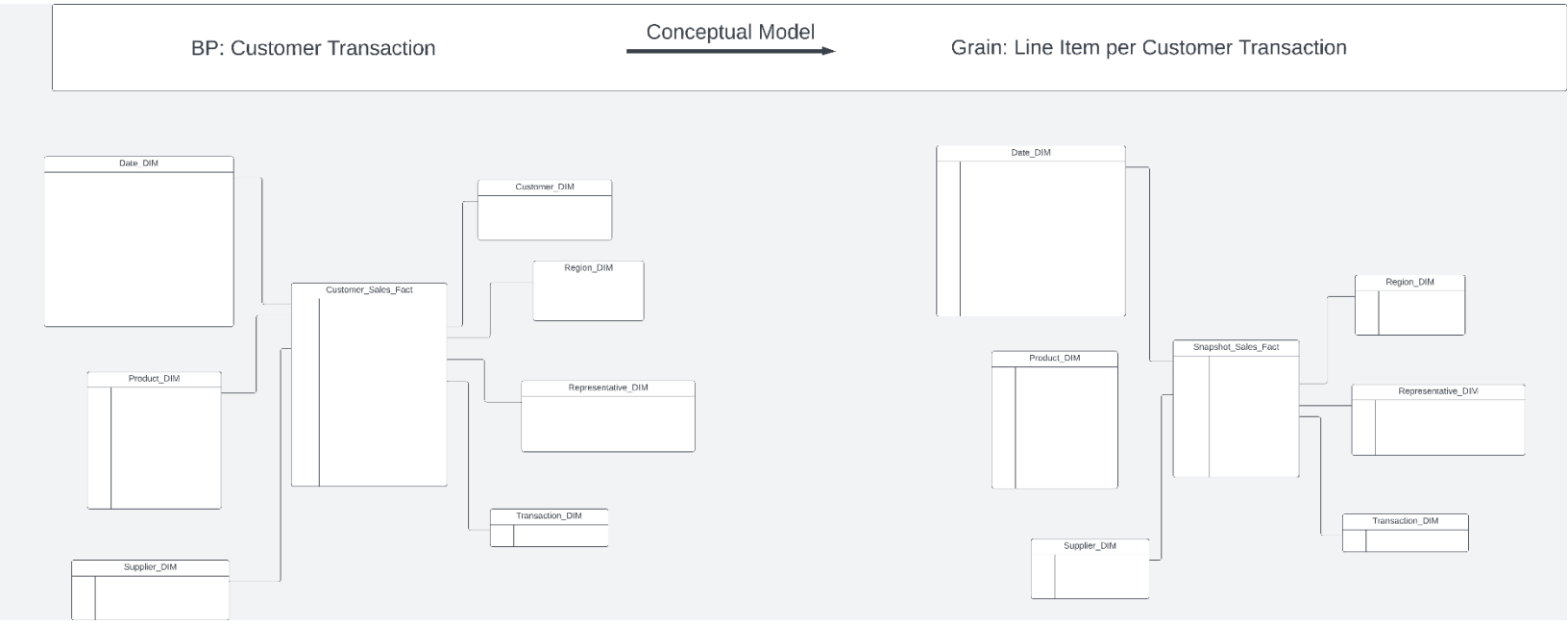

Figure 1. Conceptual Model of Proposed Data Mart

The conceptual model of the proposed solution incorporates two fact tables to address the major requirements of ETec. The first fact table "Customer_Sales_Fact" is a transactional table with the purpose of capturing detailed sales data per transaction item of a customer. Furthermore, the second fact table "Snapshot_Sale_Fact" is a snapshot style fact table that increases the performance of queries that require a holistic view of sales performance over a period of time, by providing pre-aggregated data. A detailed layout of each fact table and their attributes are listed in the subsequent section.

The dimensions incorporated in this model allow for filtering performance figures and labelling results sets according to various aspects relevant to ETec. The dimensions and fact tables are constructed to be as atomic as possible to facilitate detailed data analysis and ensure longevity of the system.

Data Mart Design

This section investigates the dimensions and attributes of the proposed model as mentioned in the Proposed Solution section. Additionally, an information package is created to aid the decision-making process regarding data presentation, data access, data granularity, the size of the data warehouse, and the required frequency of data refreshes.

The knowledge gained from the information package is then used to construct a logical design of the proposed data mart. Following this, an example of the joining processes will be provided, along with a proposed index type to optimise these processes.

Logical Model Design Process

	Information Subject: Customer Sales						
	Dimentions						
	Date	Product	Customer	Region	Representative	Transactoin	Supplier
Heirarchies / Categories	Calendar_Year { Int }	Product_Fault_Indicator { Varchar(10) }	Customer_Identification { Varchar(10) }	Region_Country { Varchar(15) }	Representative_Type { Varchar(40) }	Transaction_Type { Varchar(25) }	Supplier_Exclusive { Varchar(13) }
	Calendar_Quarter { Int }	Product_Code { Varchar(20) }	Customer_Category { Varchar(20) }	Region_Provice { Varchar(20) }	Representative_Commision_Type { Varchar(12) }		Supplier_Name { Varchar(30) }
	Calendar_Month { Varchar(10) }	Product_Category { Varchar(10) }	Customer_Is_Current { Bit }	Region_City { Varchar(30) }	Representative_Commision { Decimal(3, 2) }		Supplier_Credit_Limit { Decimal(19, 2) }
	Calendar_YYYY_MM { Varchar(6) }	Product_Brand { Varchar(1) }			Representative_Identification { Varchar(3) }		
	Calendar_Week_Number_In_Year { Int }	Product_Gender { Varchar(6) }					
	Day_Of_Week { Varchar(10) }	Product_Style { Varchar(30) }					
	Holiday_Indicator { Varchar(10) }	Product_Material { Varchar(10) }					
	Weekday_Indicator { Varchar(11) }	Product_Colour { Varchar(20) }					
	Calendar_Month_Number_In_Year { Int }	Product_Branding { Varchar(20) }					
	Day_Number_In_Calendar_Year { Int }	Product_Is_Current { Bit }					
	Day_Number_In_Calendar_Month { Int }						
	Fiscal_Month { Varchar(10) }						
	Fiscal_Month_Number_In_Year { Int }						
	Facts: Quantity, Sales_Amount, Discount, Discount_Percentage, Sales_Amount_With_Discount, Product_Sell_Price, Product_Bought_Price						

Figure 2. Information Package Diagram

This image represents the identified information subject, Customer Sales, for this solution, along with the dimensions and their associated attributes, arranged from least detailed to most detailed. This design choice enables drilling up and down through the data, providing both summary and detailed levels of information. Additionally, the diagram shows the identified measured facts for this data mart: Quantity, Sales Amount, Discount, Discount Percentage, and Sales Amount with Discount, Product_Sell_Price, Product_Bought_Price.

The information package was utilised to answer important design questions such as whether the dimensional attributes are defined as granularly as possible. By investigating the dimensions and their attributes a uniform fact table gain was identified, and the logical design of the proposed data mart was compiled.

Propose Logical Model

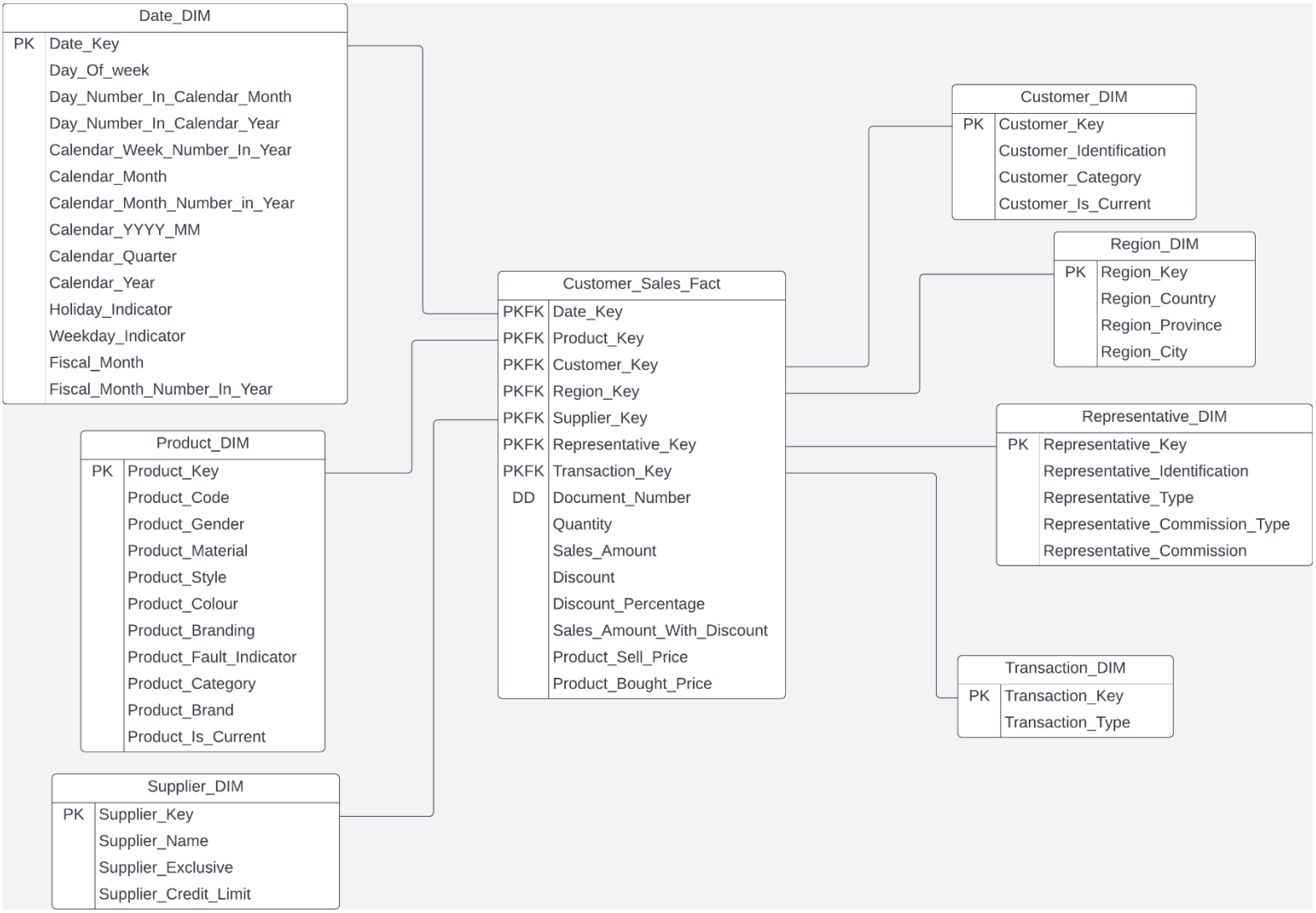

Figure 3. Fact Table Logical Model

With the construction of the Logical model the grain which was identified for the transactional fact table is “Line Item per customer transaction” which is the motivating principal behind the design of the “Customer_Sales_Fact” table.

The “Customer_Sales_Fact” table captures the quantity of a specific product sold, the net value, discount percentage and value, the net value of the sale after the discount deduction, the sold unit price, and finally the bought unit price. Dimensions described to provide additional context regarding transaction line items which includes the Customer involved in the transaction, Region where the purchase occurred, Representative responsible for the sale, Account History of the customer, the Date of the transaction, the Product sold, and the Supplier responsible for the product. This provides a nuanced view of each transactional line item.

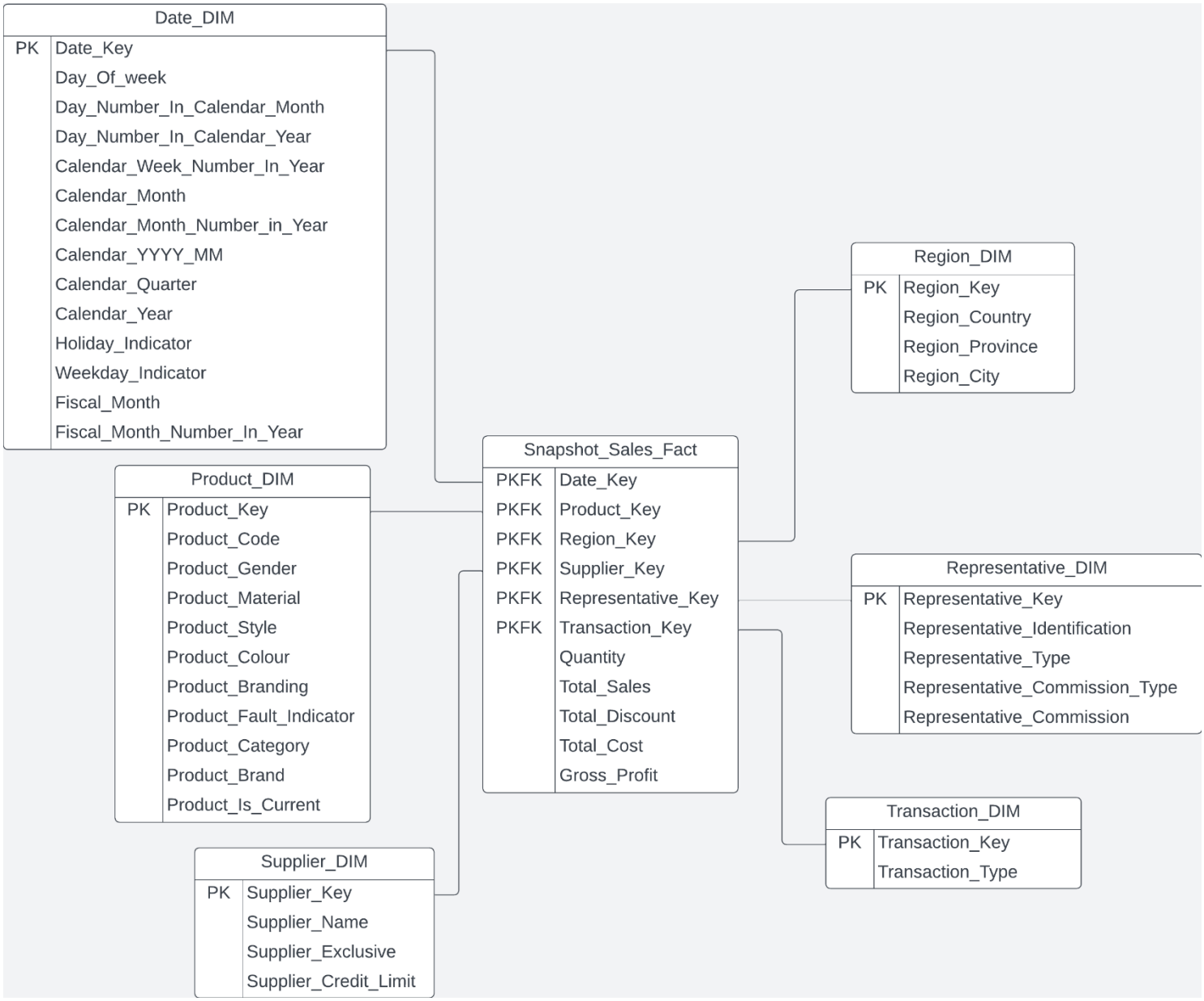

Figure 4. Snapshot Fact Table Logical Model

The Snapshot fact table named “Snapshot_Sales_Fact” is a pre-aggregated view of sales data per product per region per supplier per representative per transaction type per month. Summarised metrics include the quantity of the product sold, total sales amount, discount, cost, and calculated gross profit.

More detailed information about the fact tables and their associated dimensions can found in Appendix B and Appendix C.

Proposed Physical Model

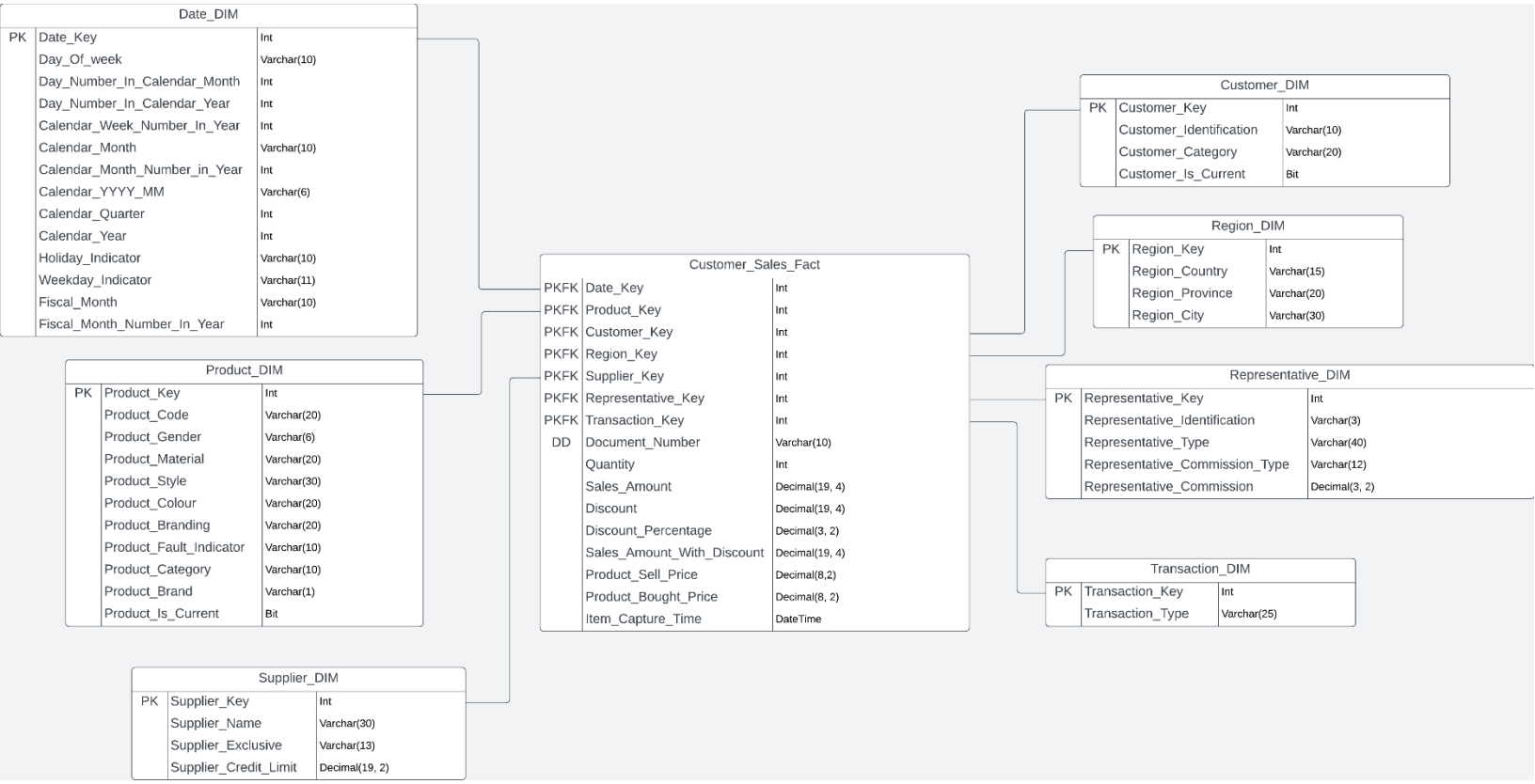

Figure 5. Fact Table Physical Model

After the creation of the logical model, Microsoft SQL Server was identified to be the chosen database management system for this project (Microsoft, 2024a). Therefore, the physical model was created with the indication of data types corresponding to those used in SQL Server (Gupta, 2019). Additionally, the following image represents the physical snapshot fact table, which includes each attribute's corresponding SQL Server data type.

Figure 6. Snapshot Fact Table Physical Model

Model Join Operation and Optimization

This section will provide an example of how JOIN operations can be performed using the proposed model (Figure 5. Fact Table Physical Model). Furthermore, an Index is proposed to accelerate JOIN performance.

SQL Code Example for a JOIN operation:

```
SELECT
    SUM(F.Discount) AS "Discount",
    R.Region_Province AS "Province",
    P.Product_Code AS "Product"
FROM
    Customer_Sales_Fact AS F
JOIN
    Date_DIM AS D ON F.Date_Key = D.Date_Key
JOIN
    Product_DIM AS P ON F.Product_Key = P.Product_Key
JOIN
    Region_DIM AS R ON F.Region_Key = R.Region_Key
WHERE
    D.Calendar_Month_Number_In_Year = 2
GROUP BY
    P.Product_Code, R.Region_Province;
```

This code example gathers data of the total discount per product per region and constrains the output to products sold in the month of February.

Indexing

Various approaches can be deployed to improve joining performance in the context of a data warehouse of which a notable solution is to use Indexes on the attributes used for the join operation. However, since the proposed model utilises primary keys for join operations, Indexing is implemented by default on primary key columns through the use of unique clustered indexes by SQL server.

To further improve the overall joining performance non-clustered indexes will be implemented on frequently joined non-key attributes to indirectly improve the joining performance. An example of these attributes includes Transaction_Type, Region_Province, Product_Code, Product_Gender, and Supplier_Name. The reason for the proposed indexing strategy is that non-clustered indexing allows the creation of multiple independent indexes within the same physical dimension table (Microsoft, 2023). Subsequently, resulting in improved record retrieval performance and reduced I/O operations (Microsoft, 2023).

ETL Specifications

This section provides a detailed description of the ETL process for each dimension and fact table, focussing on technical specifications such as source data profiling, source systems, and slow changing dimension handling techniques. Nonetheless, the goal of this section is to ensure detailed and thorough documentation of ETL processes performed throughout this project.

General Specifications

This section covers design decisions relevant to the Data Warehouse as a hole, describing extraction strategies, data quality tracking strategies, default slow changing dimension handling strategies, location, and structure of the staging area.

Extraction Strategy

As described in previous sections, the data utilized in the Data Warehouse will be retrieved from ETec's transaction processing system, which is their employed operational system. For this project, this data source is the only consideration when constructing the Data Warehouse.

Furthermore, the source system data is extracted using Microsoft Excel and loaded into staging tables corresponding to the tables utilized in the source system (Microsoft, 2024b, 2024c).

Data Quality Tracking Strategy

The purpose of the data quality tracking strategy is to ensure the correctness, quality, and accuracy of data in the Data Warehouse (Kimball *et al.*, 2008:382). This is achieved through data comparison techniques involving the comparison of source system values and data in the core staging area; relationship validation (Structural Screens) , which ensures referential integrity; format checking (Column Screens), which validates the format and ranges of data values; and logical checks (Business-Rule Screens), which involve the validation of data adherence according to the business rules (Kimball *et al.*, 2008:382).

Slow Changing Dimension Handling Strategies

For this project, the default strategy to handle changes in dimension attributes is to apply type one slow changing dimension handling. This approach ensures simplistic updates without significant impacts to Data Warehouse performance and reduces the required storage space (Kimball *et al.*, 2008:389).

Cleanup Strategy

For this project, after all appropriate values from the staging tables are loaded into the final star schema, the staging area tables are purged. This reduces the amount of storage space waisted and effectively addresses the requirements of this project, as no archiving or linage methods are required (Kimball *et al.*, 2008:436).

Location and Structure of the Staging Area

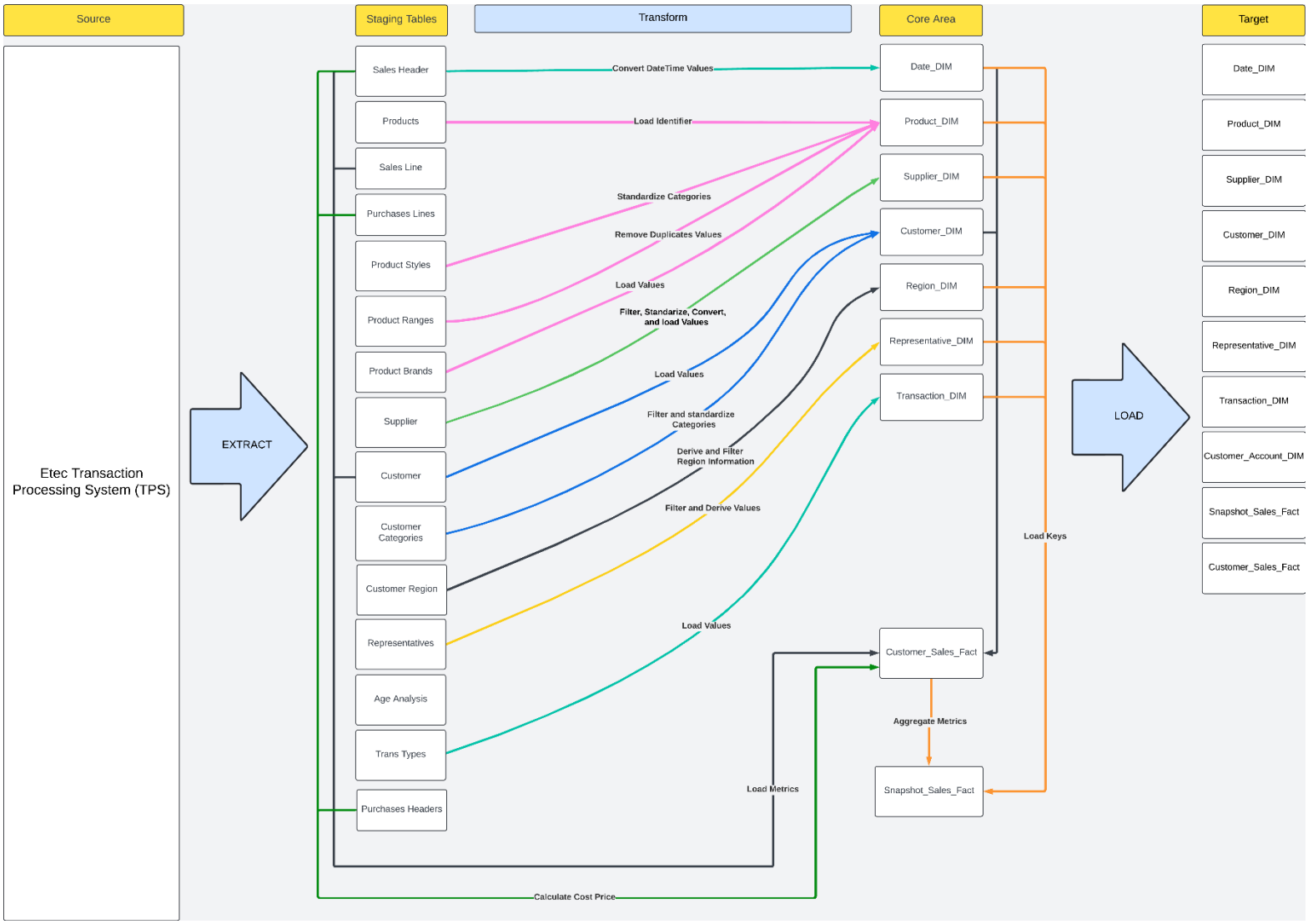

Figure 7. High-Level Staging Plan

The Data Warehouse will be employed on a local SQL Server instance (Microsoft, 2024a). The high-level staging plan, represented in Figure 7, illustrates the source systems, the staging area for the extracted source data, the transformations performed on the staging tables, and the corresponding core area, which contains a set of tables Identical to the final star schema in the Data Warehouse.

Additionally, the staging area is responsible for initial filtering, cleaning, standardizing, and removing unnecessary columns/rows. Furthermore, the core area provides functionality to address data quality, such as relationship checking, and handling of late arriving data. This ultimately ensures all core data characteristics are present before the data is loaded into the final schema.

Table-Specific Specifications

This section will present the ETL specifications related to each table in the final star schema of the Data Warehouse. Furthermore, this encompasses addressing aspects such as table design, handling of slow changing dimensions, detailed source-to-target mapping, source data profiling, and data transformation logic.

Date Dimension

Table 1. Physical Date Dimension Design

Colum Name	Date Type	Permit Nulls?	Primary Key	Comment
Date_Key	Int	No	Yes	Suggorate primary key
Day_Of_Week	Varchar(10)	Yes	No	Weekday name
Day_Number_In_Calander_Month	Int	Yes	No	Day number in Calander month
Day_Number_In_Calander_Year	Int	Yes	No	Day number in Calander year
Calendar_Week_Number_In_Year	Int	Yes	No	Number of week in year
Calendar_Month	Varchar(10)	Yes	No	Name of calendar month
Calendar_Month_Number_In_Year	Int	Yes	No	Number of month in year
Calendar_YYYY_MM	Varchar(6)	Yes	No	Shortened year and month string
Calendar_Quarter	Int	Yes	No	Quarter in calendar year
Calendar_Year	Int	Yes	No	Year
Holiday_Indicator	Varchar(10)	Yes	No	Indicates if day is a holiday
Weekday_Indicator	Varchar(11)	Yes	No	Indicates if day is a weekday
Fiscal_Month	Varchar(10)	Yes	No	Name of fiscal month
Fiscal_Month_Number_In_Year	Int	Yes	No	Number of month in fiscal year

This table represents the physical design of the Date Dimension in the star schema and describes where nulls are permitted, as well as identifying key attributes. Nonetheless, a surrogate primary key is created for each date dimension row and a unique primary key constraint is enforced on each.

Historical Load

The data for the Date Dimension is extracted/derived from the “Sales Header” staging table and consists of 1255 unique date rows and dates ranging from 2015 to 2020. For this dimension the dates are pre-populated and range from 2015 to 2030 (5475 total rows).

Slow Changing Dimension Handling

With the pre-populated nature of the Date Dimension there are no slow changing dimension handling techniques implemented.

Source Data Profiling

The date dimension, as mentioned previously, is pre-populated therefore it does not have a direct source table, however, the range of required dates are retrieved from the “Sales Header” table, more specifically the “TRANS_DATE” attribute. This attribute stores transactions dates in the datetime format, with a minimum date value of “2015/03/01” and a maximum date value of “2020/01/31”. Take note, there are no null values present in the source data attributes and the formats are consistent.

Date Dimension Source-To-Target Mapping

Table Properties	Description
Table Name	Date_DIM
Table Type	Dimension
Description	Describes Temporal Context
Used in Schemas	Customer Transaction Schema
Row Size	101 bytes
Total Size	0.53 MB (5475 rows)

	Traget					Source				
Column Name	Description	Datatype	Size (Bytes)	Example Values	SCD	Source System	Source Table	Source Field Name	Source Datatype	Data Transform Notes
Date_Key	Suggorate primary key	Int	4	1,2,3,4, ...	N/A	N/A	N/A	N/A	N/A	N/A
Day_Of_Week	Weekday name	Varchar(10)	12	"Monday", "Tuesday", ...	N/A	Etec TPS	Sales Header	TRANS_DATE	DateTime	Convert DateTime to Weekday Name
Day_Number_In_Calendar_Month	Day number in calendar month	Int	4	1,2,3,4, ...	N/A	Etec TPS	Sales Header	TRANS_DATE	DateTime	Convert DateTime to Day Number in Month
Day_Number_In_Calendar_Year	Day number in calendar year	Int	4	1,2,3,4, ...	N/A	Etec TPS	Sales Header	TRANS_DATE	DateTime	Convert DateTime to Day Number in Year
Calendar_Week_Number_In_Year	Number of week in year	Int	4	1,2,3,4, ...	N/A	Etec TPS	Sales Header	TRANS_DATE	DateTime	Convert DateTime to Week Number in Year
Calendar_Month	Name of calendar month	Varchar(10)	12	"January", "February", ...	N/A	Etec TPS	Sales Header	TRANS_DATE	DateTime	Convert DateTime to Name of Month
Calendar_Month_Number_In_Year	Number of month in year	Int	4	1,2,3,4, ...	N/A	Etec TPS	Sales Header	TRANS_DATE	DateTime	Convert DateTime to Month Number in Year
Calendar_YYYY_MM	Shortend year and month string	Varchar(6)	8	"200208", "200209", ...	N/A	Etec TPS	Sales Header	TRANS_DATE	DateTime	Convert DateTime to Year-Month String
Calendar_Quarter	Quarter in calendar year	Int	4	1, 2, 3, 4	N/A	Etec TPS	Sales Header	TRANS_DATE	DateTime	Convert DateTime to Quarter in Year
Calendar_Year	Year	Int	4	1,2,3,4, ...	N/A	Etec TPS	Sales Header	TRANS_DATE	DateTime	Convert DateTime to Year
Holiday_Indicator	Indicates if day is a holiday	Varchar(10)	12	"Is Holidy", "Not Holidy"	N/A	Etec TPS	Sales Header	TRANS_DATE	DateTime	Compare DateTime to Official Holidays
Weekday_Indicator	Indicates if day is a weekday	Varchar(11)	13	"Is Weekday", "Not Weekday"	N/A	Etec TPS	Sales Header	TRANS_DATE	DateTime	Compare DateTime to Official Weekdays
Fiscal_Month	Name of fiscal month	Varchar(10)	12	"January", "February", ...	N/A	Etec TPS	Sales Header	TRANS_DATE	DateTime	Convert DateTime to Name of Month
Fiscal_Month_Number_In_Year	Number of month in fiscal year	Int	4	1,2,3,4, ...	N/A	Etec TPS	Sales Header	TRANS_DATE	DateTime	Convert DateTime to Month Number in Year

Figure 8. Date Dimension Source-To-Target Map

This figure illustrates detailed information regarding the source-to-target mapping process. This includes information on applied transformations, source attributes, target attributes, their byte sizes, examples, and approximate storage size of the dimension as a whole.

Transformation logic

The dates form the “Sales Headers” are utilized alongside SQL functions to obtain the data required for each target attribute. Furthermore, fiscal periods are manually defined though the use of ETec metadata.

Difficulty

Overall, the complexity of the ETL effort for the Date Dimension is considered "easy", as no complex calculations/transformation logic is applied, and no complex loading techniques are needed as the data originates from one source.

Product Dimension

Table 2. Physical Product Dimension Design

Colum Name	Date Type	Permit Nulls?	Primary Key	Comment
Product_Key	Int	No	Yes	Suggorate primary key
Product_Code	Varchar(20)	No	No	Operational system identifier
Product_Gender	Varchar(6)	Yes	No	Customer gender focus
Product_Material	Varchar(20)	Yes	No	Material compisition of product
Product_Style	Varchar(30)	Yes	No	Product design style
Product_Colour	Varchar(20)	Yes	No	Colour theme of the product
Product_Branding	Varchar(20)	Yes	No	Size of product branding
Product_Fault_Indicator	Varchar(10)	No	No	Describes product quality
Product_Category	Varchar(10)	Yes	No	Indicates product range
Product_Brand	Varchar(1)	Yes	No	Brand of product
Product_Is_Current	Bit	No	No	Inducates if record is current

This table represents the physical design of the Product Dimension in the star schema and describes where nulls are permitted, as well as identifying key attributes. Nonetheless, a surrogate primary key is created for each product dimension row and a unique primary key constraint is enforced on each.

Furthermore, there is a natural key for each product in the source system, however, surrogate keys are employed to prevent data disparities, such as operational key reuse, and provides several key advantages, such as improved query performance, mapping of disparate sources, and enables the handling of unknown/not applicable conditions (Kimball *et al.*, 2008:243-244). Additionally, surrogate keys allow the application of slow changing dimension handling techniques (Kimball *et al.*, 2008:243-244).

Historical Load

The data for the Product Dimension is extracted from several staging tables, namely "Products", "Product Styles", "Product Ranges", and "Product Brands". The "Products" table consists of 7 992 unique product rows, "Product Styles" consists of 2 798 rows and 1 728 unique style combinations, "Product Ranges" consists of 2 unique product type rows, and "Product Brands" consists of 9 unique brand code rows.

Slow Changing Dimension Handling

As the Product Dimension contains information with varying levels of historical importance, type 1 and 2 slow changing dimension handling techniques are employed (Kimball *et al.*, 2008:257-258). Type 1 is used to override values with no significant historical significance, such as product code, which is the operational identifier of a product; and type 2 is utilized on attributes with historical importance, such as the product's intended gender audience and the fault indicator of the product (Kimball *et al.*, 2008:257-258).

Source Data Profiling

The Product Dimension, as mentioned previously, utilizes several staging tables, namely “Products”, “Product Styles”, “Product Ranges”, and “Product Brands”. The “Products” table consists of 7 992 unique product rows with no null values, however, the attribute utilized for the Product Dimension, “INVENTORY_CODE”, in the “Product” table consists of varying length strings, which are accounted for when conducting the transformation logic.

Furthermore, the “Product Styles” table consists of 2 798 rows and 1 728 unique style combinations, however, there are various null/not applicable values in the source data, especially in the attributes utilized for the Product Dimension, such as “GENDER”, “MATERIAL”, “STYLE”, “COLOUR”, “BRANDING”, and “QUAL_PROBS”.

Penultimately, the “Product Ranges” table consists of 2 unique product type rows, however, it is important to note the presence of duplicates in the types of products.

Ultimately, the “Product Brands” table consists of 9 unique brand code rows, with no significant faults.

Product Dimension Source-To-Target Mapping

Table Properties	Description
Table Name	Product_Dim
Table Type	Dimension
Description	Product Description
Used in Schemas	Customer Transaction Schema
Row Size	148 bytes
Total Size	1,13 MB (7992 rows)

	Traget					Source				
Column Name	Description	Datatype	Size (Bytes)	Example Values	SCD	Source System	Source Table	Source Field Name	Source Datatype	Data Transform Notes
Product_Key	Suggorate primary key	Int	4	1,2,3,4, ...	N/A	N/A	N/A	N/A	N/A	N/A
Product_Code	Operational system identifier	Varchar(20)	22	"1235COL740/47", "1240COL710/51", ...	1	Etec TPS	Products	INVENTORY_CODE	Short Text	N/A
Product_Gender	Customer gender focus	Varchar(6)	8	"Male", "Female", ...	2	Etec TPS	Product Styles	GENDER	Short Text	Standardize Categories
Product_Material	Material compisition of product	Varchar(20)	22	"Plastic", "Metal", "Combination", ...	2	Etec TPS	Product Styles	MATERIAL	Short Text	Standardize Categories
Product_Style	Product design style	Varchar(30)	32	"Sporty", "Classic", ...	2	Etec TPS	Product Styles	STYLE	Short Text	Standardize Categories
Product_Colour	Colour theme of the product	Varchar(20)	22	"Fashion", "Classic", ...	2	Etec TPS	Product Styles	COLOUR	Short Text	Standardize Categories
Product_Branding	Size of product branding	Varchar(20)	22	"Discreet", "Prominent", ...	2	Etec TPS	Product Styles	BRANDING	Short Text	Standardize Categories
Product_Fault_Indicator	Describes product quality	Varchar(10)	12	"Faulty", "Not Faulty"	2	Etec TPS	Product Styles	QUAL_PROBS	Short Text	Standardize Categories
Product_Category	Indicates product range	Varchar(10)	12	"Product", "Part", ...	1	Etec TPS	Product Ranges	PRAN_DESC	Short Text	Remove Duplicate Values
Product_Brand	Brand of product	Varchar(1)	3	"A", "B", "C", ...	2	Etec TPS	Product Brands	PRODBRA_DESC	Short Text	N/A
Product_Is_Current	Inducates if record is current	Bit	1	0, 1	1	N/A	N/A	N/A	N/A	N/A

Figure 9. Product Dimension Source-To-Target Map

This figure illustrates detailed information regarding the source-to-target mapping process. This includes information on applied transformations, source attributes, target attributes, their byte sizes, examples, and approximate storage size of the dimension as a whole.

Transformation logic

The attributes mentioned in the source data profiling section, required standardization, in terms of removing not applicable/null conditions, and removal of duplicates.

Preconditions

As the retrieval of attributes relevant to a specific product is dependent on an operational identifier, the existence of this identifier needs to be ensured before the record can be loaded into the relevant core area dimension table.

Difficulty

Overall, the complexity of the ETL effort for the Product Dimension is considered "Medium", as complex data cleaning/transformation techniques are utilized to ensure data correctness, this includes standardization of values and using the correct operational identifier to retrieve the attributes relevant to a specific product.

Supplier Dimension

Table 3. Physical Supplier Dimension Design

Colum Name	Date Type	Permit Nulls?	Primary Key	Comment
Supplier_Key	Int	No	Yes	Suggorate primary key
Supplier_Name	Varchar(30)	No	No	Supplier company name
Supplier_Exclusive	Varchar(13)	Yes	No	Indicates exclusivity with Etec
Supplier_Credit_Limit	Decimal(19, 2)	No	No	Credit limit with supplier

This table represents the physical design of the Supplier Dimension in the star schema and describes where nulls are permitted, as well as identifying key attributes. Nonetheless, a surrogate primary key is created for each Supplier Dimension row and a unique primary key constraint is enforced on each.

Historical Load

The data for the Supplier Dimension is extracted from the "Supplier" staging table, which consists of 22 unique rows.

Slow Changing Dimension Handling

As the Supplier Dimension contains no information with historical importance, the default strategy of utilizing the type 1 slow changing dimension handling technique is employed (Kimball *et al.*, 2008:257-258).

Source Data Profiling

The Supplier Dimension, as mentioned previously, utilizes the attributes of the staging table "Supplier", namely "SUPPLIER_DESC", "EXCLSV", and "CREDIT_LIMIT". The data from the "Supplier" staging table consists of 22 unique rows with no null values, and the numeric attribute,

“CREDIT_LIMIT”, utilized the minimum value of zero throughout as a default value. The only unusual characteristic is the utilization of a placeholder record with no significant importance.

Supplier Dimension Source-To-Target Mapping

Table Properties	Description
Table Name	Supplier_DIM
Table Type	Dimension
Description	Product Retailer/Manufacturer
Used in Schemas	Customer Transaction Schema
Row Size	60 bytes
Total Size	0,0013 MB (22 rows)

	Traget					Source				
Column Name	Description	Datatype	Size (Bytes)	Example Values	SCD	Source System	Source Table	Source Field Name	Source Datatype	Data Transform Notes
Supplier_Key	Suggorate primary key	Int	4	1,2,3,4, ...	N/A	Etec TPS	N/A	N/A	N/A	N/A
Supplier_Name	Supplier company name	Varchar(30)	32	"MDPLASTICS (Pty) Ltd", "ROYAL CO.", ...	1	Etec TPS	Suppliers	SUPPLIER_DESC	Short Text	Filter and Standarize Values
Supplier_Exclusive	Indicates exclusivity with Etec	Varchar(13)	15	"Exclusive", "Not Exclusive"	1	Etec TPS	Suppliers	EXCLSV	Short Text	Convert Values to Categories
Supplier_Credit_Limit	Credit limit with supplier	Decimal(19, 2)	9	0.00 - 999,999,999.99	1	Etec TPS	Suppliers	CREDIT_LIMIT	Number	N/A

Figure 10. Supplier Dimension Source-To-Target Map

This figure illustrates detailed information regarding the source-to-target mapping process. This includes information on applied transformations, source attributes, target attributes, their byte sizes, examples, and approximate storage size of the dimension as a whole.

Transformation logic

For the “SUPPLIER_DESC” attribute in the “Suppliers” staging table, the values are extracted, then filtered to remove irrelevant rows that do not represent company names, and finally standardized, such as converting the character values in the “EXCLSV” attribute to a category string to improve the interpretability and readability of the results set.

Difficulty

Overall, the complexity of the ETL effort for the Supplier Dimension is considered “easy”, as no complex calculations/transformation logic is applied, and no complex loading techniques are needed as the data originates from one source.

Customer Dimension

Table 4. Physical Customer Dimension Design

Colum Name	Date Type	Permit Nulls?	Primary Key	Comment
Customer_Key	Int	No	Yes	Suggorate primary key
Customer_Identification	Varchar(10)	No	No	Operational system identifier
Customer_Category	Varchar(20)	No	No	Description of customer
Customer_Is_Current	Bit	No	No	Inducates if record is current

This table represents the physical design of the Customer Dimension in the star schema and describes where nulls are permitted, as well as identifying key attributes. Nonetheless, a surrogate primary key is created for each Customer Dimension row and a unique primary key constraint is enforced on each.

Historical Load

The data for the Customer Dimension is extracted from the "Customer" and "Customer Categories" staging tables, which consists of 2 920 and 51 rows respectively.

Slow Changing Dimension Handling

As the Customer Dimension contains information with varying levels of historical importance, type 1 and 2 slow changing dimension handling techniques are employed (Kimball *et al.*, 2008:257-258). Type 1 is used to override values with no significant historical significance, such as Customer Identification, which is the operational identifier of a customer; and type 2 is utilized on attributes with historical importance, such as the category a customer is classified as (Kimball *et al.*, 2008:257-258).

Source Data Profiling

The Customer Dimension, as mentioned previously, utilizes attributes from two staging tables, namely "Customer" and "Customer Categories". The attribute utilized from "Customer" is "CUSTOMER_NUMBER" (2 920 unique values with no nulls present), which is the operational identifier for a customer from the source system, and "CCAT_DESC" from "Customer Categories" (7 unique values with no nulls present), which classifies a customer. However, it is important to note the "CCAT_DESC" attribute boasts a variety of irrelevant records. Furthermore, the "Customer" and "Customer Categories" staging tables have the unusual characteristic of utilising a placeholder record, which has no significant purpose.

Customer Dimension Source-To-Target Mapping

Table Properties	Description
Table Name	Customer_DIM
Table Type	Dimension
Description	Customer Details
Used in Schemas	Customer Transaction Schema
Row Size	39 bytes
Total Size	0,11 MB (2920 rows)

	Traget					Source				
Column Name	Description	Datatype	Size (Bytes)	Example Values	SCD	Source System	Source Table	Source Field Name	Source Datatype	Data Transform Notes
Customer_Key	Suggorate primary key	Int	4	1,2,3,4, ...	N/A	N/A	N/A	N/A	N/A	N/A
Customer_Identification	Operational system identifier	Varchar(10)	12	"AACJ01", "ABLE01", ...	1	Etec TPS	Customer	CUSTOMER_NUMBER	Short Text	N/A
Customer_Category	Description of customer	Varchar(20)	22	"Consignment", "House Account", ...	2	Etec TPS	Customer Categories	CCAT_DESC	Short Text	Filter and Standarize Categories
Customer_Is_Current	Inducates if record is current	Bit	1	0, 1	1	N/A	N/A	N/A	N/A	N/A

Figure 11.Customer Dimension Source-To-Target Map

This figure illustrates detailed information regarding the source-to-target mapping process. This includes information on applied transformations, source attributes, target attributes, their byte sizes, examples, and approximate storage size of the dimension as a whole.

Transformation logic

The transformation process of the Customer Dimension data is primarily centred around standardising the customer category descriptions. This standardisation is achieved through removing records that are not related to the subject matter and combining multiple versions into a single conformed version.

Difficulty

Overall, the complexity of the ETL effort for the Customer Dimension is considered “medium”, as no complex calculations/transformation logic is applied, however there is a significant fluctuation of data veracity in the “CCAT_DESC” attribute, which will require a moderate amount of ETL focus.

Region Dimension

Table 5. Physical Region Dimension Design

Colum Name	Date Type	Permit Nulls?	Primary Key	Comment
Region_Key	Int	No	Yes	Suggorate primary key
Region_Country	Varchar(15)	No	No	Name of Country
Region_Province	Varchar(20)	No	No	Province in Country
Region_City	Varchar(30)	No	No	City in Province

This table represents the physical design of the Region Dimension in the star schema and describes where nulls are permitted, as well as identifying key attributes. Nonetheless, a surrogate primary key is created for each Region Dimension row and a unique primary key constraint is enforced on each.

Historical Load

The data for the Region Dimension is extracted from the "Customer Region" staging table, which consists of 34 rows.

Slow Changing Dimension Handling

As the Region Dimension contains no information with historical importance, the default strategy of utilizing the type 1 slow changing dimension handling technique is employed (Kimball *et al.*, 2008:257-258).

Source Data Profiling

The Region Dimension, as mentioned previously, utilises data from the "Customer Region" staging table, more specifically the "REGION_DESC" (34 unique values with no nulls present) attribute. However, it is important to note the "REGION_DESC" attribute boasts a variety of irrelevant, multi-valued, and inconsistently formatted records. Furthermore, "Customer Categories" have an unusual characteristic of utilising a placeholder record, which has no significant purpose. The only unusual characteristic is the utilization of a placeholder record with no significant importance

Region Dimension Source-To-Target Mapping

Table Properties	Description
Table Name	Region_DIM
Table Type	Dimension
Description	Region Details
Used in Schemas	Customer Transaction Schema
Row Size	82 bytes
Total Size	0,0027 MB (34 rows)

	Traget					Source				
Column Name	Description	Datatype	Size (Bytes)	Example Values	SCD	Source System	Source Table	Source Field Name	Source Datatype	Data Transform Notes
Region_Key	Suggorate primary key	Int	4	1,2,3,4, ...	N/A	N/A	N/A	N/A	N/A	N/A
Region_Country	Name of Country	Varchar(15)	17	"South-Africa", "Botswana", ...	1	Etec TPS	Customer Region	REGION_DESC	Short Text	Derive From Location Information
Region_Province	Province in Country	Varchar(20)	22	"Gauteng", "Limpopo", ...	1	Etec TPS	Customer Region	REGION_DESC	Short Text	Filter From Region Information
Region_City	City in Province	Varchar(30)	32	"Vanderbijlpark", "Sandton", ...	1	Etec TPS	Customer Region	REGION_DESC	Short Text	Filter From Region Information

Figure 12. Region Dimension Source-To-Target Map

This figure illustrates detailed information regarding the source-to-target mapping process. This includes information on applied transformations, source attributes, target attributes, their byte sizes, examples, and approximate storage size of the dimension as a whole.

Transformation logic

The transformation process of the Region Dimension data is primarily centred around standardising, filtering, and deriving location information from the data in the "REGION_DESC" attribute. This is achieved by removing records that are not related to the subject matter, separating multi-valued

records, combining multiple versions of the same value into a single conformed version, and deriving other important location details using the corrected data.

Difficulty

Overall, the complexity of the ETL effort for the Customer Dimension is considered "Hard" due to the significant amount of ETL effort required to correct the data values, this entails standardising different versions of location information to the same grain and deriving other important location information.

Representative Dimension

Table 6. Physical Representative Dimension Design

Colum Name	Date Type	Permit Nulls?	Primary Key	Comment
Representative_Key	Int	No	Yes	Suggorate primary key
Representative_Identification	Varchar(3)	No	No	Represents various representatives
Representative_Type	Varchar(40)	No	No	Types of representatives
Representative_Commission_Type	Varchar(12)	No	No	Indicates commission calculation method
Representative_Commission	Decimal(3, 2)	No	No	Commission percentage

This table represents the physical design of the Representative Dimension in the star schema and describes where nulls are permitted, as well as identifying key attributes. Nonetheless, a surrogate primary key is created for each Representative Dimension row and a unique primary key constraint is enforced on each.

It is important to note that each of the representative records have a natural key identifier in the source system; however, surrogate keys are employed to prevent data disparities, such as operational key reuse, and provide several key advantages, such as improved query performance, mapping of disparate sources, and enabling the handling of unknown/not applicable conditions (Kimball *et al.*, 2008:243-244). Additionally, surrogate keys allow the application of slow changing dimension handling techniques (Kimball *et al.*, 2008:243-244).

Historical Load

The data utilized for the Representative Dimension is extracted from the "Representatives" staging table, which consists of 128 rows.

Slow Changing Dimension Handling

The Representative Dimension contains no information of historical importance; therefore, the default strategy of utilising the type 1 slow changing dimension handling technique is employed (Kimball *et al.*, 2008:257-258).

Source Data Profiling

The Representative Dimension, as mentioned previously, utilises data from the "Representatives" staging table. More specifically, data from the "REP_CODE", "REP_DEC", "COMM_METHOD", and "COMMISION" attributes are utilised. The "REP_CODE" attribute consists of 128 unique values and contains no null records. The values in the "REP_CODE" attribute are operational identifiers for

various representatives in the source system. The “REP_DEC” attribute consists of 120 unique values and contains no null records. The data in the “REP_DEC” describe the various type of representatives in the source system; however, it is important to note that this attribute and the “REP_CODE” attribute contain various irrelevant records with no operational importance. Furthermore, the “REP_CODE” is a semantic key, which will be addressed in the transformation process.

The “COMM_METHOD” attribute consists of 2 unique values; however, this attribute also contains blank and irrelevant values with no operational importance. Finally, the last attribute utilised is “COMMISION” which contains 2 distinct values which range from 0.0 to 0.5. It is important to note that the only unusual characteristic in the “Representatives” staging table is the utilisation of a placeholder record of no significant importance.

Representative Dimension Source-To-Target Mapping

Table Properties	Description
Table Name	Representative_DIM
Table Type	Dimension
Description	Sales Representative Information
Used in Schemas	Customer Transaction Schema
Row Size	70 bytes
Total Size	0,0085 MB (128 rows)

	Traget					Source				
Column Name	Description	Datatype	Size (Bytes)	Example Values	SCD	Source System	Source Table	Source Field Name	Source Datatype	Data Transform Notes
Representative_Key	Suggorate primary key	Int	4	1,2,3,4, ...	N/A	N/A	N/A	N/A	N/A	N/A
Representative_Identification	Represents various representatives	Varchar(3)	5	"01", "02", "03", ...	1	Etec TPS	Representatives	REP_CODE	Short Text	Filter and Derive From Representative Information
Representative_Type	Types of representative	Varchar(40)	42	"House Account and Department Store", "Telesales", "Sales Representative"	1	Etec TPS	Representatives	REP_DEC	Short Text	Filter and Derive From Representative Information
Representative_Commission_Type	Indicates commission calculation method	Varchar(12)	14	"Sales", "Gross Profit"	1	Etec TPS	Representatives	COMM_METHOD	Short Text	Filter From Representative Information
Representative_Commission	Commission percentage	Decimal(3, 2)	5	0.0 to 1.0	1	Etec TPS	Representatives	COMMISION	Number	N/A

Figure 13. Representative Dimension Source-To-Target Map

This figure illustrates detailed information regarding the source-to-target mapping process. This includes information on applied transformations, source attributes, target attributes, their byte sizes, examples, and approximate storage size of the dimension as a whole.

Transformation logic

The transformation process of the Representative Dimension is primarily centred around the “REP_CODE” and “REP_DEC” attributes. For the “REP_CODE” attribute the transformation process mostly entails the standardization of the values into distinct representative categories. The “REP_DEC” attribute’s data, which are descriptions to the identifiers in the “REP_CODE” attribute, will be standardized and filtered to match the transformed representative identifier information. The

values in the “COMM_MET” will be standardized to create distinct categories and the “COMMISSION” value will be directly loaded.

Difficulty

Overall, the complexity of the ETL effort for the Representative Dimension is considered “Hard” due to the significant amount of ETL effort required to correct the data values, separate multi-valued data, and standardize representative information.

Transaction Dimension

Table 8. Physical Transaction Dimension Design

Colum Name	Date Type	Permit Nulls?	Primary Key	Comment
Transaction_Key	Int	No	Yes	Suggorate primary key
Transaction_Type	Varchar(25)	No	No	Indicates type of transaction

This table represents the physical design of the Transaction Dimension in the star schema and describes where nulls are permitted, as well as identifying key attributes. Nonetheless, a surrogate primary key is created for each Transaction Dimension row and a unique primary key constraint is enforced on each.

Historical Load

The data utilized for the Transaction Dimension is extracted from the “Trans Types” staging table, which consists of 5 rows.

Slow Changing Dimension Handling

The Transaction Dimension contains no information of historical importance; however, the contained data should not be overwritten; therefore, no slow changing dimension handling techniques are utilised.

Source Data Profiling

The Transaction Dimension, as mentioned previously, utilises data from only the “Trans Types” staging table. More specifically, the “TRANSTYPE_DESC” attribute is utilised, which consists of 5 unique values and contains no null values.

Transaction Dimension Source-To-Target Mapping

Table Properties	Description
Table Name	Transaction_DIM
Table Type	Dimension
Description	Describe Type Of Transaction
Used in Schemas	Customer Transaction Schema
Row Size	31 bytes
Total Size	0,00015 MB (5 rows)

	Traget					Source				
Column Name	Description	Datatype	Size (Bytes)	Example Values	SCD	Source System	Source Table	Source Field Name	Source Datatype	Data Transform Notes
Transaction_Key	Suggorate primary key	Int	4	1,2,3,4, ...	N/A	N/A	N/A	N/A	N/A	N/A
Transaction_Type	Indicates type of transaction	Varchar(25)	27	"TAX INVOICE", "CREDIT NOTE", ...	N/A	Etec TPS	Trans Types	TRANSTYPE_DESC	Short Text	N/A

Figure 14.Transaction Dimension Source-To-Target Map

This figure illustrates detailed information regarding the source-to-target mapping process. This includes information on applied transformations, source attributes, target attributes, their byte sizes, examples, and approximate storage size of the dimension as a whole.

Transformation logic

There is no specific transformation process for the Transaction Dimension as the data will be directly loaded from the staging tables.

Difficulty

Overall, the complexity of the ETL effort for the Transaction Dimension is considered “easy”, as no complex calculations/transformation logic is applied, and no complex loading techniques are needed as the data is loaded without alteration from the “Trans Types” staging table.

Customer Sales Fact Table

Table 9. Physical Customer Sales Fact Design

Colum Name	Date Type	Permit Nulls?	Primary Key	Comment
Date_Key	Int	No	Yes	Suggorate primary key
Product_Key	Int	No	Yes	Suggorate primary key
Customer_Key	Int	No	Yes	Suggorate primary key
Region_Key	Int	No	Yes	Suggorate primary key
Supplier_Key	Int	No	Yes	Suggorate primary key
Representative_Key	Int	No	Yes	Suggorate primary key
Transaction_Key	Int	No	Yes	Suggorate primary key
Document_Number	Varchar(10)	No	Yes	Link to operational transactional documentation
Quantitiy	Int	No	No	Number of line item purchased
Sales_Amount	Decimal(19, 4)	No	No	Line-item total
Discount	Decimal(19, 4)	No	No	Discount applied to line-item total
Discount_Percentage	Decimal(3, 2)	No	No	Discount percentage to be applied

Sales_Amount_With_Discount	Decimal(19, 4)	No	No	Line-item total after discount
Product_Sell_Price	Decimal(8, 2)	No	No	Sell price of the product
Product_Bought_Price	Decimal(8, 2)	No	No	Bought price of the product -- (Get Correct Unit Price by Comparing Transaction Date with Supplier Purchase Date)

This table represents the physical design of the Customer Sales Fact in the star schema and describes where nulls are permitted, as well as identifying key attributes. For the Customer Sales Fact, a composite foreign-primary key is used, which reference corresponding primary keys in the dimension tables. A unique primary key constraint is enforced on the composite foreign-primary key to ensure each row is uniquely identified. It is important to note that due to duplicate composite keys the degenerate dimension is also included as part of the fact table primary key.

Historical Load

The data utilized for the Customer Sales Fact is extracted from various staging table, namely "Sales Header", "Sales Line", "Customer", "Purchases Lines", "Purchases Headers". The "Sales Header" staging table consists of 67 988 rows. The "Sales Line" staging table consists of 322 823 rows. The "Customer" staging table consists of 2 920 rows. The "Purchases Lines" staging table consists of 3 077 rows. Finaly, the "Purchases Headers" staging table consists of 71 rows.

Slow Changing Dimension Handling

The Customer Sales Fact contains the measurements needed to analyse the business process; furthermore, each record is a snapshot of a transaction at a specific point in time, therefore, no slow changing dimension handling techniques are employed (Najm *et al.*, 2022:11).

Source Data Profiling

The Customer Sales Fact, as mentioned previously, utilises data from various tables such as "Sales Header", "Sales Line", "Customer", "Purchases Lines", "Purchases Headers".

In the "Sales Header" staging table, the "DOC_NUMBER" and "TRANS_DATE" attributes are utilised, which consist of 67 988 unique values and 1 256 unique dates, respectively; there are also no nulls present in these attributes. For "TRANS_DATE" the oldest date is "3/1/2015" and the latest date is "1/31/2020".

In the "Sales Line" staging table the "QUANTITY", "TOTAL_LINE_PRICE", and "UNIT_SELL_PRICE" is utilised all of which do not contain any null values. The "QUANTITY" attribute consists of values ranging from -293 to 1 708; "TOTAL_LINE_PRICE" consists of values ranging from -280 170 to 541 312; and "UNIT_SELL_PRICE" consists of values ranging from -280 170 to 280 170. All of these attributes have the unusual characteristic of containing sales and return data, however, for this project only the sales data is considered. Furthermore, In the "Customer" staging table the "DISCOUNT" attribute is utilised, which consists of numeric values ranging from 0 to 58 and contains no null values.

For the “Purchases Lines” staging table the “UNIT_COST_PRICE” is utilised, which consists of numeric values ranging from 0 to 3 400 and no nulls, however, to determine the unit cost price at various points in time for each line item in a customer transaction, for historical context, the “PURCH_DATE” attribute in “Purchases Headers” and “TRANS_DATE” attribute in Sales Header are used. “PURCH_DATE” attribute consists of 71 values and 17 unique dates, furthermore, the oldest data is “1/3/2018” and latest date is “11/1/2019”. The “TRANS_DATE” attribute consists of 67 988 values and 1257 unique dates. Both attributes do not contain null values.

Customer Sales Fact Source-To-Target Mapping

Table Properties	Description
Table Name	Customer_Sales_Fact
Table Type	Fact Table
Description	Captures Transaction Line Item Information
Used in Schemas	Customer Transaction Schema
Row Size	98 bytes
Total Size	30 MB (322 823 rows)

	Traget					Source				
Column Name	Description	Datatype	Size (Bytes)	Example Values	SCD	Source System	Source Table	Source Field Name	Source Datatype	Data Transform Notes
Date_Key	Suggorate primary key	Int	4	1,2,3,4, ...	N/A	Customer Transaction Schema	Date_DIM	Date_Key	Int	N/A
Product_Key	Suggorate primary key	Int	4	1,2,3,4, ...	N/A	Customer Transaction Schema	Product_DIM	Product_Key	Int	N/A
Customer_Key	Suggorate primary key	Int	4	1,2,3,4, ...	N/A	Customer Transaction Schema	Customer_DIM	Customer_Key	Int	N/A
Region_Key	Suggorate primary key	Int	4	1,2,3,4, ...	N/A	Customer Transaction Schema	Region_DIM	Region_Key	Int	N/A
Supplier_Key	Suggorate primary key	Int	4	1,2,3,4, ...	N/A	Customer Transaction Schema	Supplier_DIM	Supplier_Key	Int	N/A
Representative_Key	Suggorate primary key	Int	4	1,2,3,4, ...	N/A	Customer Transaction Schema	Representative_DIM	Representative_Key	Int	N/A
Transaction_Key	Suggorate primary key	Int	4	1,2,3,4, ...	N/A	Customer Transaction Schema	Transaction_DIM	Transaction_Key	Int	N/A
Document_Number	Link to operational transactional documentation	Varchar(10)	12	"ID500405", "ID500392", ...	N/A	Etec TPS	Sales Header	DOC_NUMBER	Short Text	N/A
Quantitiy	Number of line item purchased	Int	4	1,2,3,4, ...	N/A	Etect TPS	Sales Line	QUANTITY	Number	N/A
Sales_Amount	Line item total	Decimal(19, 4)	9	0.00 - 999,999,999,999,999.9999	N/A	Etect TPS	Sales Line	TOTAL_LINE_PRICE	Number	N/A
Discount	Discount applied to line item total	Decimal(19, 4)	9	0.00 - 999,999,999,999,999.9999	N/A	Etect TPS	N/A	N/A	N/A	Calculate
Discount_Percentage	Discount percentage to be applied	Decimal(3, 2)	5	0.0 to 1.0	N/A	Etect TPS	Customer	DISCOUNT	Number	Convert To Value Bewteen 0.0 And 1.0
Sales_Amount_With_Discount	Line item total after discount	Decimal(19, 4)	9	0.00 - 999,999,999,999,999.9999	N/A	Etect TPS	N/A	N/A	N/A	Calculate
Product_Sell_Price	Sell price of the product	Decimal(8, 2)	5	0.00 - 999,999.99	N/A	Etec TPS	Sales Line	UNIT_SELL_PRICE	Number	Filter Positive Values
Product_Bought_Price	Bought price of product	Decimal(8, 2)	5	0.00 - 999,999.99	N/A	Etec TPS	Purchases Lines, Purchases Headers, Sales Header	UNIT_COST_PRICE	Number	Get Correct Unit Price by Comparing Transaction Date with Supplier Purchase Date

Figure 15. Customer Sales Fact Source-To-Target Map

This figure illustrates detailed information regarding the source-to-target mapping process. This includes information on applied transformations, source attributes, target attributes, their byte sizes, examples, and approximate storage size of the fact table as a whole.

Dependencies

For the Customer Sales Fact, all the dimension tables must be populated before the loading can occur. This includes the "DATE_DIM", "PRODUCT_DIM", "SUPPLIER_DIM", "CUSTOMER_DIM", "REGION_DIM", "REPRESENTATIVE_DIM", "CUSTOMER_ACCOUNT_DIM", and "TRANSACTION_DIM" dimension tables being loaded and checked before the fact table is populated.

Transformation Logic

For the Customer Sales Fact table, various transformations are applied. This mainly includes the conversion of "DISCOUNT" attribute values in the "Customer" staging table to numeric values between 0.0 and 1.0; the filtering of the "UNIT_SELL_PRICE" attribute values to extract only positive values; and the loading/derivation of unit cost values from the "UNIT_COST_PRICE" attribute and using the "PURCH_DATE" and "TRANS_DATE" attributes.

Preconditions

As previously mentioned, the population of the customer sales fact depends on various foreign keys from the star schema dimensions, therefore, the keys which a specific transaction line-item are dependent on need to be confirmed before the line-item information is inserted into the fact table. Additionally, each line-item is grouped by a document number that represents the parent transaction; this document number needs to be ensured before the record can be loaded.

Difficulty

Overall, the complexity of the ETL effort for the Customer Sales Fact Table is considered "Hard", as complex data cleaning/transformation/derivation techniques are utilized to ensure data correctness.

Snapshot Fact Table

Table 10. Physical Snapshot Sales Fact Table Design

Colum Name	Date Type	Permit Nulls?	Primary Key	Comment
Date_Key	Int	No	Yes	Suggorate primary key
Product_Key	Int	No	Yes	Suggorate primary key
Region_Key	Int	No	Yes	Suggorate primary key
Supplier_Key	Int	No	Yes	Suggorate primary key
Representative_Key	Int	No	Yes	Suggorate primary key
Transaction_Key	Int	No	Yes	Suggorate primary key
Quantity	Int	No	No	Total number of line item purchased
Total_Sales	Decimal(19, 4)	No	No	Aggregated sales value without discount
Total_Discount	Decimal(19, 4)	No	No	Aggregated discount value
Total_Cost	Decimal(19, 4)	No	No	Aggregated product cost
Gross_Profit	Decimal(19, 4)	No	No	Calculated Gross Profit

This table represents the physical design of the Snapshot Sales Fact in the star schema and describes where nulls are permitted, as well as identifying key attributes. For the Snapshot Sales

Fact, a composite foreign-primary key is used, which reference corresponding primary keys in the dimension tables. A unique primary key constraint is enforced on the composite foreign-primary key to ensure each row is uniquely identified.

Historical Load

For the Snapshot Sales Fact table, no historical data is directly loaded from the staging tables, however, the data for this table is loaded and aggregated, to a monthly grain, from the main "Customer Sales Fact" table, which consists of approximately 322 823 rows.

Slow Changing Dimension Handling

The Snapshot Sales Fact contains the aggregated measurements needed to analyse the business process; furthermore, each record is an aggregated snapshot of the total sales per product per region per supplier per representative per transaction type per month; therefore, no slow changing dimension handling techniques are employed (Najm *et al.*, 2022:11).

Source Data Profiling

The Snapshot Sales Fact table, as mentioned previously, utilises data from the Customer Sales Fact Table. More specifically, the "Quantity", "Sale Amount", "Discount", and "Product Bought Price". The "Quantity" attribute consists of values ranging from 0 to 2 147 483 648, with no nulls; "Sale Amount" consisting of values ranging from 0.00 to 999 999 999 999 999.9999, with no nulls; "Discount" consisting of values ranging from 0.00 to 999 999 999 999 999.9999, with no nulls; and "Product Bought Price" consisting of values ranging from 0.00 to 999 999.9999, with no nulls.

Additionally, the Snapshot Sales Fact table utilises some of its own attributes to calculate the gross profit of the total sales per product per month; these attributes are "Total Sales", "Total Cost", and "Total Discount", each of which consist of values ranging from 0.00 to 999 999 999 999 999.9999, with no null values.

Snapshot Sales Fact Source-To-Target Mapping

Table Properties	Description
Table Name	Snapshort_Sales_Fact
Table Type	Snapshot Fact Table
Description	Aggregated Values From Fact Table Per Month
Used in Schemas	Customer Transaction Schema
Row Size	64 bytes
Total Size	N/A

	Traget					Source				
Column Name	Description	Datatype	Size (Bytes)	Example Values	SCD	Source System	Source Table	Source Field Name	Source Datatype	Data Transform Notes
Date_Key	Suggorate primary key	Int	4	1,2,3,4, ...	N/A	Customer Transaction Schema	Date_DIM	Date_Key	Int	N/A
Product_Key	Suggorate primary key	Int	4	1,2,3,4, ...	N/A	Customer Transaction Schema	Product_DIM	Product_Key	Int	N/A
Region_Key	Suggorate primary key	Int	4	1,2,3,4, ...	N/A	Customer Transaction Schema	Region_DIM	Region_Key	Int	N/A
Supplier_Key	Suggorate primary key	Int	4	1,2,3,4, ...	N/A	Customer Transaction Schema	Supplier_DIM	Supplier_Key	Int	N/A
Representative_Key	Suggorate primary key	Int	4	1,2,3,4, ...	N/A	Customer Transaction Schema	Representative_DIM	Representative_Key	Int	N/A
Transaction_Key	Suggorate primary key	Int	4	1,2,3,4, ...	N/A	Customer Transaction Schema	Customer_Account_DIM	Transaction_Key	Int	N/A
Quantity	Total number of a line item purchased	Int	4	1,2,3,4, ...	N/A	Customer Transaction Schema	Fact_Table	Quantitiy	Int	Aggregate
Total_Sales	Aggregated sales value without discount	Decimal(19, 4)	9	0.00 - 999,999,999,999,999.9999	N/A	Customer Transaction Schema	Fact_Table	Sales_Amount	Decimal(19, 4)	Aggregate
Total_Discount	Aggregated discount value	Decimal(19, 4)	9	0.00 - 999,999,999,999,999.9999	N/A	Customer Transaction Schema	Fact_Table	Discount	Decimal(19, 4)	Aggregate
Total_Cost	Aggregated product cost	Decimal(19, 4)	9	0.00 - 999,999,999,999,999.9999	N/A	Customer Transaction Schema	Fact_Table	Product_Bought_Price, Quantity	Decimal	Aggregate
Gross_Profit	Calculated Gross Profit	Decimal(19, 4)	9	0.00 - 999,999,999,999,999.9999	N/A	Customer Transaction Schema	Snapshort_Sales_Fact	Total_Sales, Total_Cost, Total_Discount	Decimal	Calculate

Figure 16. Snapshot Sales Fact Source-To-Target Map

This figure illustrates detailed information regarding the source-to-target mapping process. This includes information on applied transformations, source attributes, target attributes, their byte sizes, and examples.

Dependencies

For the Snapshot Sales Fact, some of the dimension tables and the customer sales fact table must be populated before the loading can occur. This includes the “DATE_DIM”, “PRODUCT_DIM”, “SUPPLIER_DIM”, “REGION_DIM”, “REPRESENTATIVE_DIM”, “CUSTOMER_ACCOUNT_DIM”, and “TRANSACTION_DIM” dimension tables, as well as the “CUSTOMER_SALES_FACT” fact table being loaded and checked before the snapshot fact table is populated.

Transformation Logic

For the Snapshot Sales Fact table, the main transformation is the extraction and aggregation of transaction information from the Customer Sales Fact table to a per product per region per supplier per representative per transaction type per month granularity.

Preconditions

As previously mentioned, the population of the snapshot sales fact depends on various foreign keys from the star schema dimensions and values from the Customer Sales Fact table; therefore, all keys

in the star schema dimensions and values in the Customer Sales Fact table need to be confirmed before the aggregated monthly sales information is calculated and inserted into the snapshot fact table.

Difficulty

Overall, the complexity of the ETL effort for the Snapshot Sales Fact Table is considered “easy”, as no complex calculations or transformation techniques are utilized.

Business Intelligence: Documentation

Business intelligence applications are commonly the only window into the Data warehousing system and thus the vehicle for delivering value from such a project; therefore, these applications are complementary to the Data warehouse (Kimball *et al.*, 2008:10). This section provides detail on the implementation of the Data warehouse plan and the business intelligence reporting dashboards/ad hoc reports needed to address ETec's reporting needs.

This section's structure is as follows: It starts with an overview of the build process, followed by the specifics of the DW development process. Finaly, this section concludes with specifications on the design process of the business intelligence dashboards/ad hoc reports along with technical details regarding the innerworkings of these reports. Additionally, examples of how the BI solution can address ETec's reporting needs are provided.

Build Process

This build process was initiated after data from ETec's transactional processing system was extracted and transformed to meet the Data warehouse requirements. The data warehouse development process started with the creation of SQL scripts in SQL Server, which was chosen due to its simplicity and comprehensiveness, to provide the initial framework of the star schema (Microsoft, 2024a). Following the creation of the star schema, the transformed data was transferred from the staging tables to the target dimensional models. It is important to note that each of the dimensional models were fully loaded before the fact table was populated. Afterward, the data warehouse was evaluated to ensure correct loading.

With the Data warehouse complete, focus was shifted to the development of the BI dashboards and ad hoc reports. The chosen BI tool, which was deemed the most suitable for addressing ETec's requirements, is Power BI by Microsoft (Microsoft, 2024d). Through this application, it was possible to create comprehensive reports with adequate capabilities for effectively meeting ETec's reporting requirements.

Data Warehouse Development

As mentioned, the Data Warehouse development was initiated alongside the data extraction and transformation processes. The first phase of the development process was the creation of the target star schema for the transformed data. This entails the creation of the physical database and tables.

More specifically, the physical database was created using SQL Server, which provides ample capabilities for scalability and performance, in turn ensuing that the solution can be as optimal as possible (Microsoft, 2024e). The following image represents the SQL script executed on SQL server to create the physical database.

```
IF NOT EXISTS (SELECT * FROM sys.databases WHERE name = 'ETec Sales Data Mart')
BEGIN
    CREATE DATABASE [ETec Sales Data Mart]
END
```

Figure 17. Creation Script of Physical Database

This script, as shown in the above figure, creates a database with the name "ETec Sales Data Mart" which will contain the overall star schema solution, consisting of all the dimensional models and fact tables described in the Data Mart Design Section, excluding the Snapshot fact table.

Table Creation

With the base physical database created, all the tables needed were also designed and created. This includes the Date, Product, Supplier, Customer, Region, Representative, and Transaction dimensional tables as well as the Customer Sales Fact Table. The following subsections present the SQL code for each and a quick description.

Date Dimension

The following figure represents the SQL script written to create the proposed Date dimension for the Data warehouse solution.

```
IF NOT EXISTS (SELECT * FROM sysobjects WHERE [NAME]= 'DATE_DIM' AND xtype = 'U')
BEGIN

CREATE TABLE [dbo].[DATE_DIM](
    [date_key] int IDENTITY(1,1) NOT NULL,      -- Suggorate primary key
    [day_of_week] varchar(10) NULL,             -- Weekday name
    [day_number_in_calendar_month] int NULL,    -- Day number in calander month
    [day_number_in_calendar_year] int NULL,     -- Day number in calander year
    [calendar_week_number_in_year] int NULL,    -- Number of week in year
    [calendar_month] varchar(10) NULL,          -- Name of calendar month
    [calendar_month_number_in_year] int NULL,   -- Number of month in year
    [calendar_yyyy_mm] varchar(6) NULL,         -- Shortend year and month string
    [calendar_quarter] int NULL,                -- Quarter in calendar year
    [calendar_year] int NULL,                   -- Year
    [holiday_indicator] varchar(11) NULL,       -- Indicates if day is a holiday
    [weekday_indicator] varchar(11) NULL,       -- Indicates if day is a weekday
    [fiscal_month] varchar(10) NULL,            -- Name of fiscal month
    [fiscal_month_number_in_year] int NULL,     -- Number of month in fiscal year

    CONSTRAINT [PK_DATE_DIM] PRIMARY KEY CLUSTERED
    (

        [date_key] ASC

    ) WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

END
```

Figure 18. SQL Script for Creating Date Dimension

This script, as shown in the above figure, creates the Date dimension table of the proposed star schema. The table design follows the exact specifications as mentioned in the ETL Specifications Section for the Date Dimension. This entails using the correct data types, indicating where nulls are permissible, and primary key constraints.

Product Dimension

The following figure represents the SQL script written to create the proposed Product dimension for the Data warehouse solution.

```
IF NOT EXISTS (SELECT * FROM sysobjects WHERE [NAME]= 'PRODUCT_DIM' AND xtype = 'U')
BEGIN

CREATE TABLE [dbo].[PRODUCT_DIM](
    [product_key] int IDENTITY(1,1) NOT NULL,       -- Suggorate primary key
    [product_code] varchar(20) NOT NULL,            -- Operational system identifier
    [product_gender] varchar(6) NULL,               -- Customer gender focus
    [product_material] varchar(20) NULL,            -- Material compisition of product
    [product_style] varchar(30) NULL,               -- Product design style
    [product_colour] varchar(20) NULL,              -- Colour theme of the product
    [product_branding] varchar(20) NULL,            -- Size of product branding
    [product_fault_indicator] varchar(10) NOT NULL, -- Describes product quality
    [product_category] varchar(10) NULL,            -- Indicates product range
    [product_brand] varchar(1) NULL,                -- Brand of product
    [product_is_current] bit NOT NULL,              -- Inducates if record is current

    CONSTRAINT [PK_PRODUCT_DIM] PRIMARY KEY CLUSTERED
    (

        [product_key] ASC

    ) WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

END
```

Figure 19. Script for Creating Product Dimension

This script, as shown in the above figure, creates the Product dimension table of the proposed star schema. The table design follows the exact specifications as mentioned in the ETL Specifications Section for the Product Dimension. This entails using the correct data types, indicating where nulls are permissible, and primary key constraints. Additionally, the type two slow changing dimension maintenance columns are also indicated and created.

Supplier Dimension

The following figure represents the SQL script written to create the proposed Supplier dimension for the Data warehouse solution.

```
IF NOT EXISTS (SELECT * FROM sysobjects WHERE [NAME]= 'SUPPLIER_DIM' AND xtype = 'U')
BEGIN

CREATE TABLE [dbo].[SUPPLIER_DIM](
    [supplier_key] int IDENTITY(1,1) NOT NULL,      -- Suggorate primary key
    [supplier_name] varchar(30) NOT NULL,           -- Supplier company name
    [supplier_exclusive] varchar(13) NULL,          -- Indicates exclusivity with Etec
    [supplier_credit_limit] decimal(19, 2) NOT NULL,-- Credit limit with supplier

    CONSTRAINT [PK_SUPPLIER_DIM] PRIMARY KEY CLUSTERED
    (

        [supplier_key] ASC

    ) WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

END
```

Figure 20. Script for Creating Supplier Dimension

This script, as shown in the above figure, creates the Supplier dimension table of the proposed star schema. The table design follows the exact specifications as mentioned in the ETL Specifications Section for the Supplier Dimension. This entails using the correct data types, indicating where nulls are permissible, and primary key constraints.

Customer Dimension

The following figure represents the SQL script written to create the proposed Customer dimension for the Data warehouse solution.

```
IF NOT EXISTS (SELECT * FROM sysobjects WHERE [NAME]= 'CUSTOMER_DIM' AND xtype = 'U')
BEGIN

CREATE TABLE [dbo].[CUSTOMER_DIM](
    [customer_key] int IDENTITY(1,1) NOT NULL,      -- Suggorate primary key
    [customer_identification] varchar(10) NOT NULL, -- Operational system identifier
    [customer_category] varchar(20) NOT NULL,       -- Description of customer
    [customer_is_current] bit NOT NULL,             -- Inducates if record is current

    CONSTRAINT [PK_CUSTOMER_DIM] PRIMARY KEY CLUSTERED
    (

        [customer_key] ASC

    ) WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

END
```

Figure 21. Script for Creating Customer Dimension

This script, as shown in the above figure, creates the Customer dimension table of the proposed star schema. The table design follows the exact specifications as mentioned in the ETL Specifications Section for the Supplier Dimension. This entails using the correct data types, indicating where nulls are permissible, and primary key constraints. Additionally, the type two slow changing dimension maintenance columns are also indicated and created.

Region Dimension

The following figure represents the SQL script written to create the proposed Region dimension for the Data warehouse solution.

```
IF NOT EXISTS (SELECT * FROM sysobjects WHERE [NAME]= 'REGION_DIM' AND xtype = 'U')
BEGIN

CREATE TABLE [dbo].[REGION_DIM](
    [region_key] int IDENTITY(1,1) NOT NULL,     -- Surrogate primary key
    [region_country] varchar(15) NOT NULL,       -- Name of Country
    [region_province] varchar(30) NOT NULL,      -- Province in Country
    [region_city] varchar(30) NOT NULL,          -- City in Province

    CONSTRAINT [PK_REGION_DIM] PRIMARY KEY CLUSTERED
    (

        [region_key] ASC

    ) WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

END
```

Figure 22. Script for Creating Region Dimension

This script, as shown in the above figure, creates the Region dimension table of the proposed star schema. The table design follows the exact specifications as mentioned in the ETL Specifications Section for the Region Dimension. This entails using the correct data types, indicating where nulls are permissible, and primary key constraints.

Representative Dimension

The following figure represents the SQL script written to create the proposed Representative dimension for the Data warehouse solution.

```
IF NOT EXISTS (SELECT * FROM sysobjects WHERE [NAME]= 'REPRESENTATIVE_DIM' AND xtype = 'U')
BEGIN

CREATE TABLE [dbo].[REPRESENTATIVE_DIM](
    [representative_key] int IDENTITY(1,1) NOT NULL,        -- Surrogate primary key
    [representative_identification] varchar(3) NOT NULL,    -- Represents various representatives
    [representative_type] varchar(40) NOT NULL,             -- Types of representative
    [representative_commission_type] varchar(12) NOT NULL,  -- Indicates commission calculation method
    [representative_commission] decimal(3,2) NOT NULL,      -- Commission percentage

    CONSTRAINT [PK_REPRESENTATIVE_DIM] PRIMARY KEY CLUSTERED
    (

        [representative_key] ASC

    ) WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

END
```

Figure 23. Script for Creating Representative Dimension

This script, as shown in the above figure, creates the Representative dimension table of the proposed star schema. The table design follows the exact specifications as mentioned in the ETL Specifications Section for the Representative Dimension. This entails using the correct data types, indicating where nulls are permissible, and primary key constraints.

Transaction Dimension

The following figure represents the SQL script written to create the proposed Transaction dimension for the Data warehouse solution.

```
IF NOT EXISTS (SELECT * FROM sysobjects WHERE [NAME]= 'TRANSACTION_DIM' AND xtype = 'U')
BEGIN

CREATE TABLE [dbo].[TRANSACTION_DIM](
    [transaction_key] int IDENTITY(1,1) NOT NULL,    -- Surrogate primary key
    [transaction_type] varchar(25) NOT NULL,         -- Indicates type of transaction

    CONSTRAINT [PK_TRANSACTION_DIM] PRIMARY KEY CLUSTERED
    (

        [transaction_key] ASC

    ) WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

END
```

Figure 24. Script for Creating Transaction Dimension

This script, as shown in the above figure, creates the Transaction dimension table of the proposed star schema. The table design follows the exact specifications as mentioned in the ETL Specifications Section for the Transaction Dimension. This entails using the correct data types, indicating where nulls are permissible, and primary key constraints.

Customer Sales Fact

With all the dimensions created, the last step of the development process was to create the fact table of the proposed star schema. The following image illustrates the SQL script for creating the Customer Sales Fact Table.

```
IF NOT EXISTS (SELECT * FROM sysobjects WHERE [NAME]= 'CUSTOMER_SALES_FACT' AND xtype = 'U')
BEGIN

CREATE TABLE [dbo].[CUSTOMER_SALES_FACT](
    [date_key] int NOT NULL,                            -- Part of composite primary key and foreign key to DATE_DIM
    [product_key] int NOT NULL,                         -- Part of composite primary key and foreign key to PRODUCT_DIM
    [customer_key] int NOT NULL,                        -- Part of composite primary key and foreign key to CUSTOMER_DIM
    [region_key] int NOT NULL,                          -- Part of composite primary key and foreign key to REGION_DIM
    [supplier_key] int NOT NULL,                        -- Part of composite primary key and foreign key to SUPPLIER_DIM
    [representative_key] int NOT NULL,                  -- Part of composite primary key and foreign key to REPRESENTATIVE_DIM
    [transaction_key] int NOT NULL,                     -- Part of composite primary key and foreign key to TRANSACTION_DIM
    [document_number] varchar(10) NOT NULL,             -- Link to operational transactional documentation
    [quantity] int NOT NULL,                            -- Number of line item purchased
    [sales_amount] decimal(19,4) NOT NULL,              -- Line item total
    [discount] decimal(19,4) NOT NULL,                  -- Discount applied to line item total
    [discount_percentage] decimal(3,2) NOT NULL,        -- Discount percentage to be applied
    [sales_amount_with_discount] decimal(19,4) NOT NULL,-- Line item total after discount
    [product_sell_price] decimal(8, 2) NOT NULL,        -- Sell price of the product
    [product_bought_price] decimal(8,2) NOT NULL,       -- Bought price of the product

    CONSTRAINT [PK_CUSTOMER_SALES_FACT] PRIMARY KEY CLUSTERED
    (

        [date_key] ASC,
        [product_key] ASC,
        [customer_key] ASC,
        [region_key] ASC,
        [supplier_key] ASC,
        [representative_key] ASC,
        [transaction_key] ASC

    ) WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]
```

Figure 25. Script for Creating Customer Sales Fact: Part 1

As the Fact table is more complex to design, the above image illustrates the first part of the SQL script. This part of the script illustrates the design of the fact table according to the exact specifications mentioned in the ETL Specifications Section for the Customer Sales Fact Table. This entails using the correct data types, indicating where nulls are permissible, and primary key constraints. Additionally, the fact table has a composite primary-foreign key, therefore, foreign key constraints were necessary to ensure proper referential integrity. These constraints are indicated in the following image, which illustrates the second part of the SQL script.

```
-- Foreign Key Constraints

-- DATE DIMENSION
ALTER TABLE [dbo].[CUSTOMER_SALES_FACT] WITH CHECK ADD CONSTRAINT [FK_CUSTOMER_SALES_FACT_DATE] FOREIGN KEY([date_key])
REFERENCES [dbo].[DATE_DIM] ([date_key])
ALTER TABLE [dbo].[CUSTOMER_SALES_FACT] CHECK CONSTRAINT [FK_CUSTOMER_SALES_FACT_DATE]

-- PRODUCT DIMENSION
ALTER TABLE [dbo].[CUSTOMER_SALES_FACT] WITH CHECK ADD CONSTRAINT [FK_CUSTOMER_SALES_FACT_PRODUCT] FOREIGN KEY([product_key])
REFERENCES [dbo].[PRODUCT_DIM] ([product_key])
ALTER TABLE [dbo].[CUSTOMER_SALES_FACT] CHECK CONSTRAINT [FK_CUSTOMER_SALES_FACT_PRODUCT]

-- CUSTOMER DIMENSION
ALTER TABLE [dbo].[CUSTOMER_SALES_FACT] WITH CHECK ADD CONSTRAINT [FK_CUSTOMER_SALES_FACT_CUSTOMER] FOREIGN KEY([customer_key])
REFERENCES [dbo].[CUSTOMER_DIM] ([customer_key])
ALTER TABLE [dbo].[CUSTOMER_SALES_FACT] CHECK CONSTRAINT [FK_CUSTOMER_SALES_FACT_CUSTOMER]

-- REGION DIMENSION
ALTER TABLE [dbo].[CUSTOMER_SALES_FACT] WITH CHECK ADD CONSTRAINT [FK_CUSTOMER_SALES_FACT_REGION] FOREIGN KEY([region_key])
REFERENCES [dbo].[REGION_DIM] ([region_key])
ALTER TABLE [dbo].[CUSTOMER_SALES_FACT] CHECK CONSTRAINT [FK_CUSTOMER_SALES_FACT_REGION]

-- SUPPLIER DIMENSION
ALTER TABLE [dbo].[CUSTOMER_SALES_FACT] WITH CHECK ADD CONSTRAINT [FK_CUSTOMER_SALES_FACT_SUPPLIER] FOREIGN KEY([supplier_key])
REFERENCES [dbo].[SUPPLIER_DIM] ([supplier_key])
ALTER TABLE [dbo].[CUSTOMER_SALES_FACT] CHECK CONSTRAINT [FK_CUSTOMER_SALES_FACT_SUPPLIER]

-- REPRESENTATIVE DIMENSION
ALTER TABLE [dbo].[CUSTOMER_SALES_FACT] WITH CHECK ADD CONSTRAINT [FK_CUSTOMER_SALES_FACT_REPRESENTATIVE] FOREIGN KEY ([representative_key])
REFERENCES [dbo].[REPRESENTATIVE_DIM] ([representative_key])
ALTER TABLE [dbo].[CUSTOMER_SALES_FACT] CHECK CONSTRAINT [FK_CUSTOMER_SALES_FACT_REPRESENTATIVE]

-- TRANSACTION DIMENSION
ALTER TABLE [dbo].[CUSTOMER_SALES_FACT] WITH CHECK ADD CONSTRAINT [FK_CUSTOMER_SALES_FACT_TRANSACTION] FOREIGN KEY ([transaction_key])
REFERENCES [dbo].[TRANSACTION_DIM] ([transaction_key])
ALTER TABLE [dbo].[CUSTOMER_SALES_FACT] CHECK CONSTRAINT [FK_CUSTOMER_SALES_FACT_TRANSACTION]

END
```

Figure 26. Script for Creating Customer Sales Fact: Part 2

This image illustrates the foreign key constraints created on the composite primary key for the fact table to ensure proper referential integrity.

Data Warehouse Start Schema

After the execution of the table creation scripts, the proposed star schema was fully created in SQL Server. The following illustration represents the star schema data model of the Data warehouse, as viewed by the database management system.

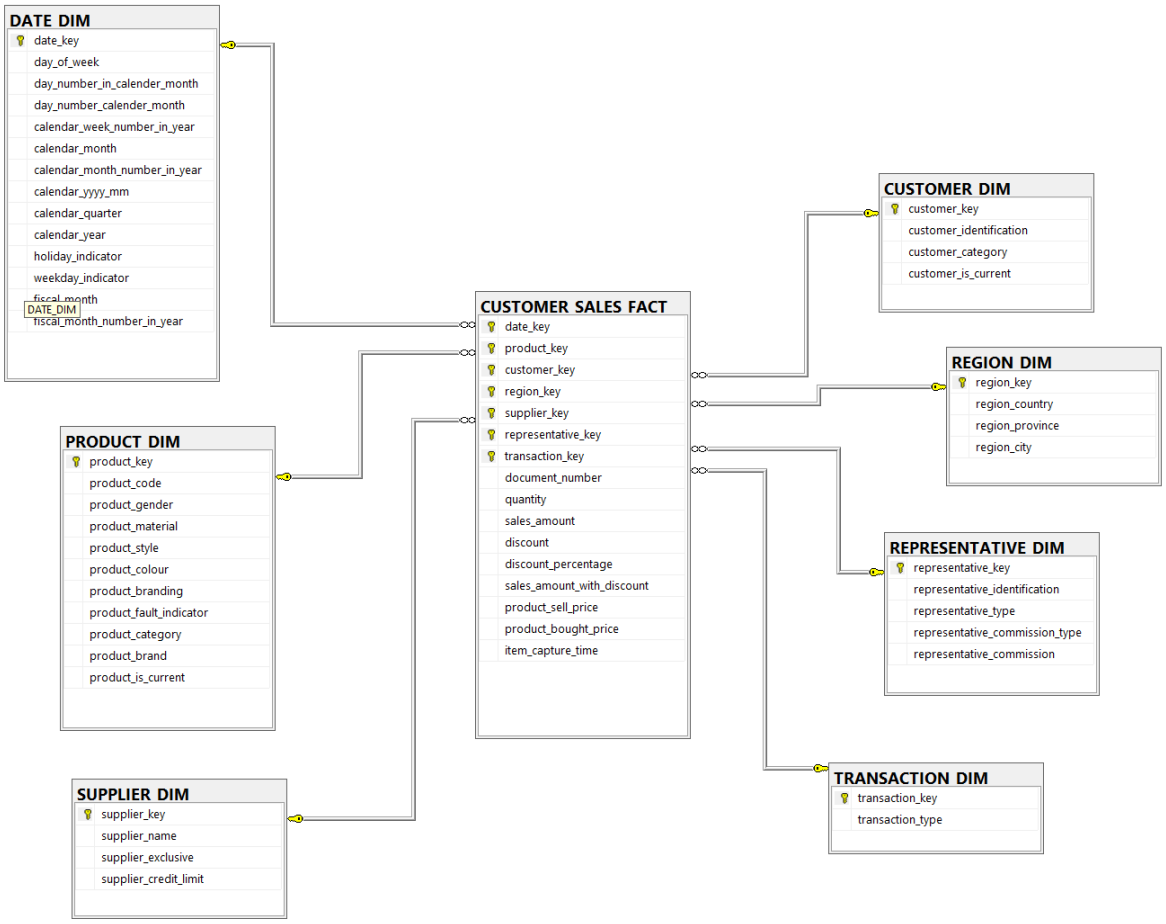

Figure 27. SSMS Data Model

This figure illustrates the created dimensions and fact table in the database management system, SQL Server as well as all foreign key relationships.

Indexing

During the creation of the star schema tables, clustered indexes were created on all table primary key attributes. This indexing is the base indexing created on each table to enable the effective retrieval of data, due to data being organised, allowing queries to find desired rows effectively (Cioloca & Georgescu, 2011:14-15). Additionally, indexing the primary keys allows improved joins across tables, as indexing assists in finding matching records more effectively, and having clustered indexing reduces disc I/O due to the table data being physically structured in order (Cioloca & Georgescu, 2011:14-15). To further improve the query performance in the data warehouse we

employed non-clustering on attributes of interest based on the reporting requirements of ETec (Cioloca & Georgescu, 2011:16).

```
CONSTRAINT [PK_CUSTOMER_SALES_FACT] PRIMARY KEY CLUSTERED
(

    [date_key] ASC,
    [product_key] ASC,
    [customer_key] ASC,
    [region_key] ASC,
    [supplier_key] ASC,
    [representative_key] ASC,
    [transaction_key] ASC

) WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
```

Figure 28. Example of Clustered Index

This figure illustrates a SQL statement creating a clustered index on a key attribute.

```
-- ============================================
-- CREATE INDEXES
-- ============================================
-- DATE_DIM
CREATE NONCLUSTERED INDEX idx_fiscal_month ON [dbo].[DATE_DIM] ([fiscal_month]);
CREATE NONCLUSTERED INDEX idx_calander_year ON [dbo].[DATE_DIM] ([calendar_year]);
CREATE NONCLUSTERED INDEX idx_calendar_quarter ON [dbo].[DATE_DIM] ([calendar_quarter]);
CREATE NONCLUSTERED INDEX idx_day_in_month ON [dbo].[DATE_DIM] ([day_number_in_calender_month]);

-- SUPPLIER_DIM
CREATE NONCLUSTERED INDEX idx_suppier_name ON [dbo].[SUPPLIER_DIM] ([supplier_name]);

-- REGION_DIM
CREATE NONCLUSTERED INDEX idx_region_provice ON [dbo].[REGION_DIM] ([region_province]);

-- PRODUCT_DIM
CREATE NONCLUSTERED INDEX idx_product_branding ON [dbo].[PRODUCT_DIM] ([product_branding]);
CREATE NONCLUSTERED INDEX idx_product_material ON [dbo].[PRODUCT_DIM] ([product_material]);
CREATE NONCLUSTERED INDEX idx_product_style ON [dbo].[PRODUCT_DIM] ([product_style]);
CREATE NONCLUSTERED INDEX idx_product_gender ON [dbo].[PRODUCT_DIM] ([product_gender]);

-- REPRESENTATIVE_DIM
CREATE NONCLUSTERED INDEX idx_representative_identification ON [dbo].[REPRESENTATIVE_DIM] ([representative_identification]);

-- FACT TABLE CUSTOMER_SALES_FACT
CREATE NONCLUSTERED INDEX idx_customer_sales_fact_document_number ON [dbo].[CUSTOMER_SALES_FACT] ([document_number]);
```

Figure 29. Nonclustered Index SQL Creation Script

Historical Load

After the star schema's development, the focus shifted to loading the data from the staging tables into the target dimensional model. This process entailed exporting the transformed data from Microsoft Excel into a flat file format, CSV, which is then loaded table for table into the data warehouse (Michael & Ahirao, 2020:3). It is important to note that, as mentioned earlier, all of the dimensional tables were populated before the fact table (Michael & Ahirao, 2020:3).

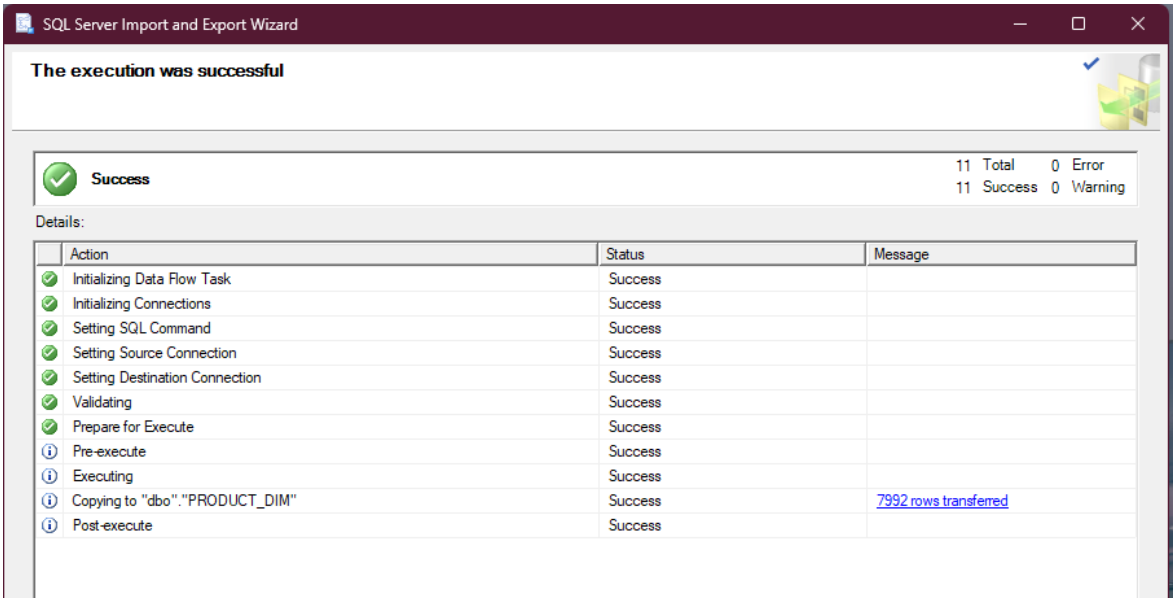

Figure 30. Example of Successful Loading

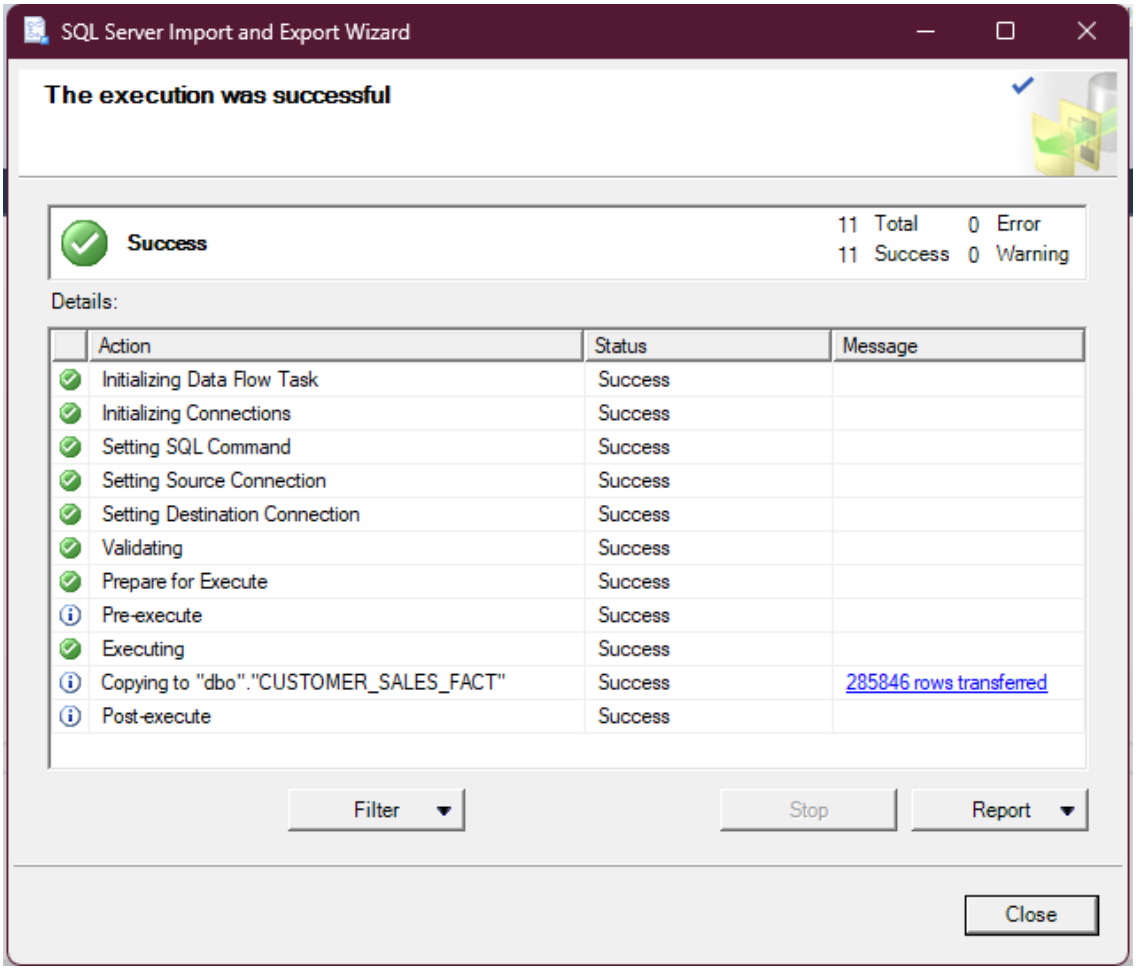

Figure 31. Fact Table Successful Loading

With the loading of all data into the Data Warehouse, the backend component of the ETL pipeline is complete. This completion entails the end of data extraction, transformation, and loading phases, which encompass cleaning, conforming, and structuring raw data for optimal use. However, as the backend is complete, the focus shifts to the front room. This entails adding value to the data in the Data Warehouse through the use of analysis tools, reports, and data visualisations to provide ETec with deeper insights into the data that enable informed decisions that can drive strategic business outcomes.

Report Building

This section will provide insights into the creation of dashboards, reports, and more, that were developed to address the querying requirements of ETec. As mentioned previously, all BI operations will be done through Power BI (Michael & Ahirao, 2020:3-4; Microsoft, 2024d).

Reporting Context

The following reports were all based on the reporting requirements discussed with the ETec team, specifically their requirements for reports on sales data filtered on daily, weekly, monthly, quarterly, and annual periods. Additionally, other reports were also created based on their additional requirement of shifting their focus from sales performance to supplier orientated reporting.

Reports presented in this section extract data from formulated database views instead of querying data directly from the presentation server. These views differ from their physical database equivalents in that operational identifiers were excluded with the exception of transaction document numbers. Furthermore, views also encompass pre-calculated aggregates such as total cost and total profit per transaction line item.

Sales Report

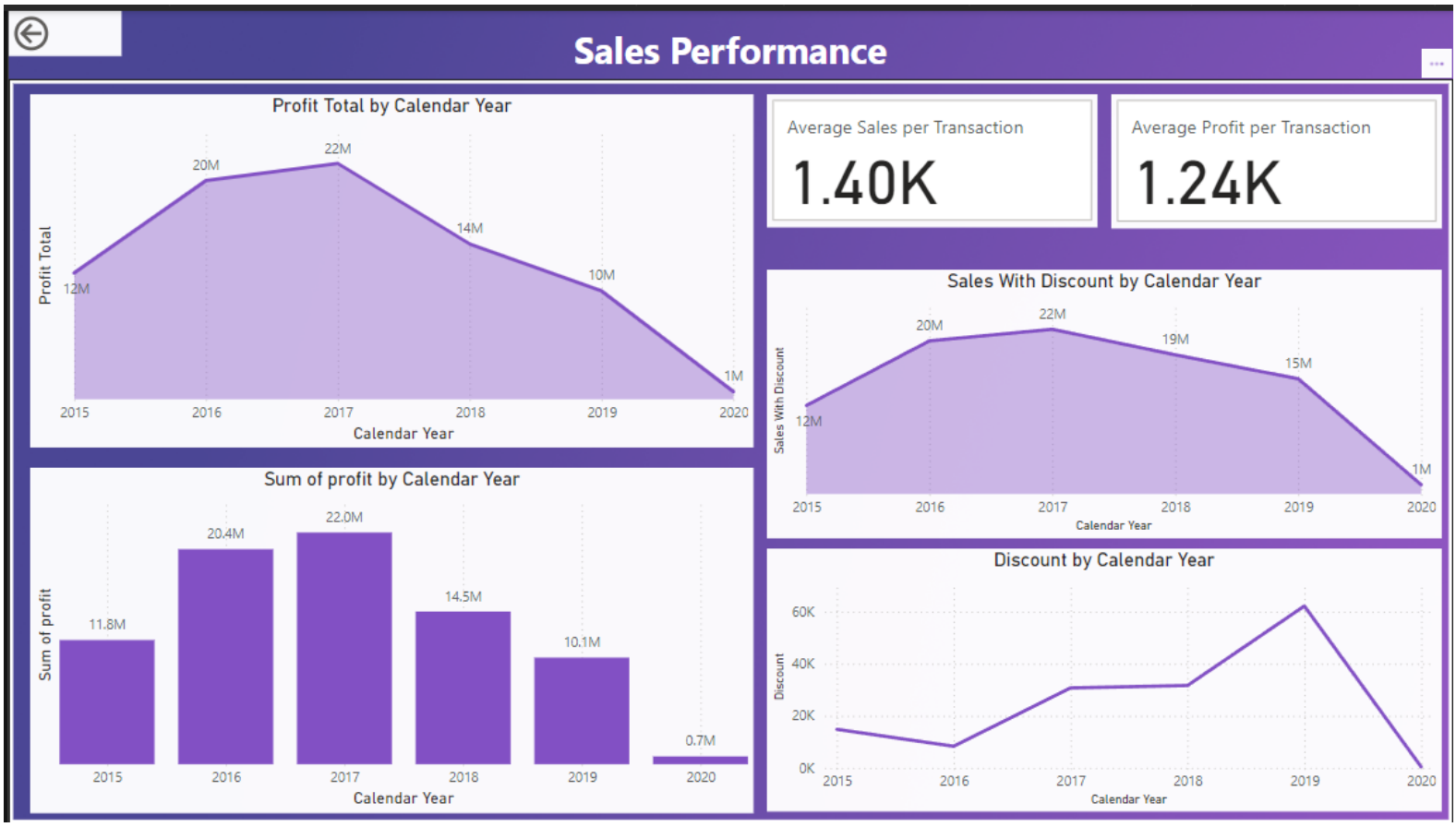

Figure 32. Sales Performance Report

This report represents the sales performance across the company's history. More specifically, this report represents the sales performance by year through describing aspects such as profit by calendar year, discount given by calendar year, and the sales with discount by calendar year. Furthermore, key performance indicators included in this report encompass average sales per transaction and average profit per transaction. Nonetheless, the dashboard this report is based on enables drill down capabilities to address the requirements of ETec's management and allows unencumbered data analysis.

It is important to note that this report is not based on specific queries and was generated using the functionality provided by Power BI, which abstracts query formulation.

Operational Insights

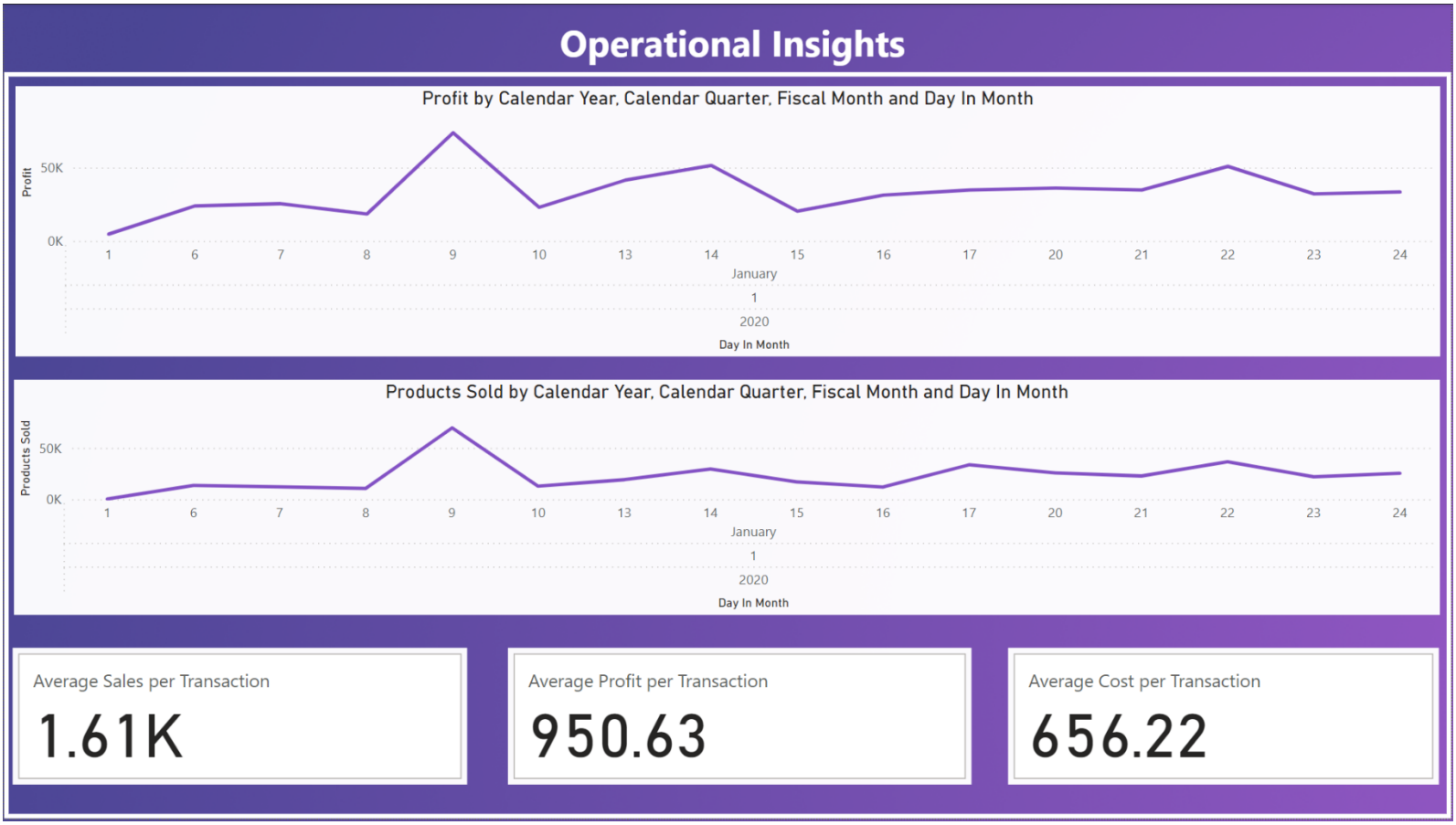

Figure 33. Operational Insights Report

To satisfy the requirement of detailed operational reports, this report is generated at a per day granularity over the month of the current fiscal year. This requirement is satisfied through representing products sold by day and profit by day on line graphs , which support simplified time-series analysis. Additionally, added contextual details are provided through describing the average sales per transaction, average profit per transaction, and average cost per transaction. More specifically, these indicators were calculated through creating transaction averages over document numbers and thus considering multiple line items per transaction. It is important to note that in the dashboard, the indicators are filtered and calculated based on the granularity of the line graphs, each of which allows drill down capabilities.

Another important note is that this report is not based on specific queries and was generated using the functionality provided by Power BI, which abstracts query formulation.

Employee Performance

Figure 34. Employee Performance Report

To provide more contextual detail surrounding the sales performance metrics described by previous reports, this report illustrates the total profits earned, total products sold, and total discounts given by each representative in the organisation. This supports employee performance analysis and connects sales performance to specific representatives to facilitate ETec's sales performance cantered management initiative. Notwithstanding, the dashboard that this report is based on can be filtered by time period, region, and transition types supporting a myriad of analysis perspectives.

Take note that this report is not based on specific queries and was generated using the functionality provided by Power BI, which abstracts query formulation.

Customer Analysis

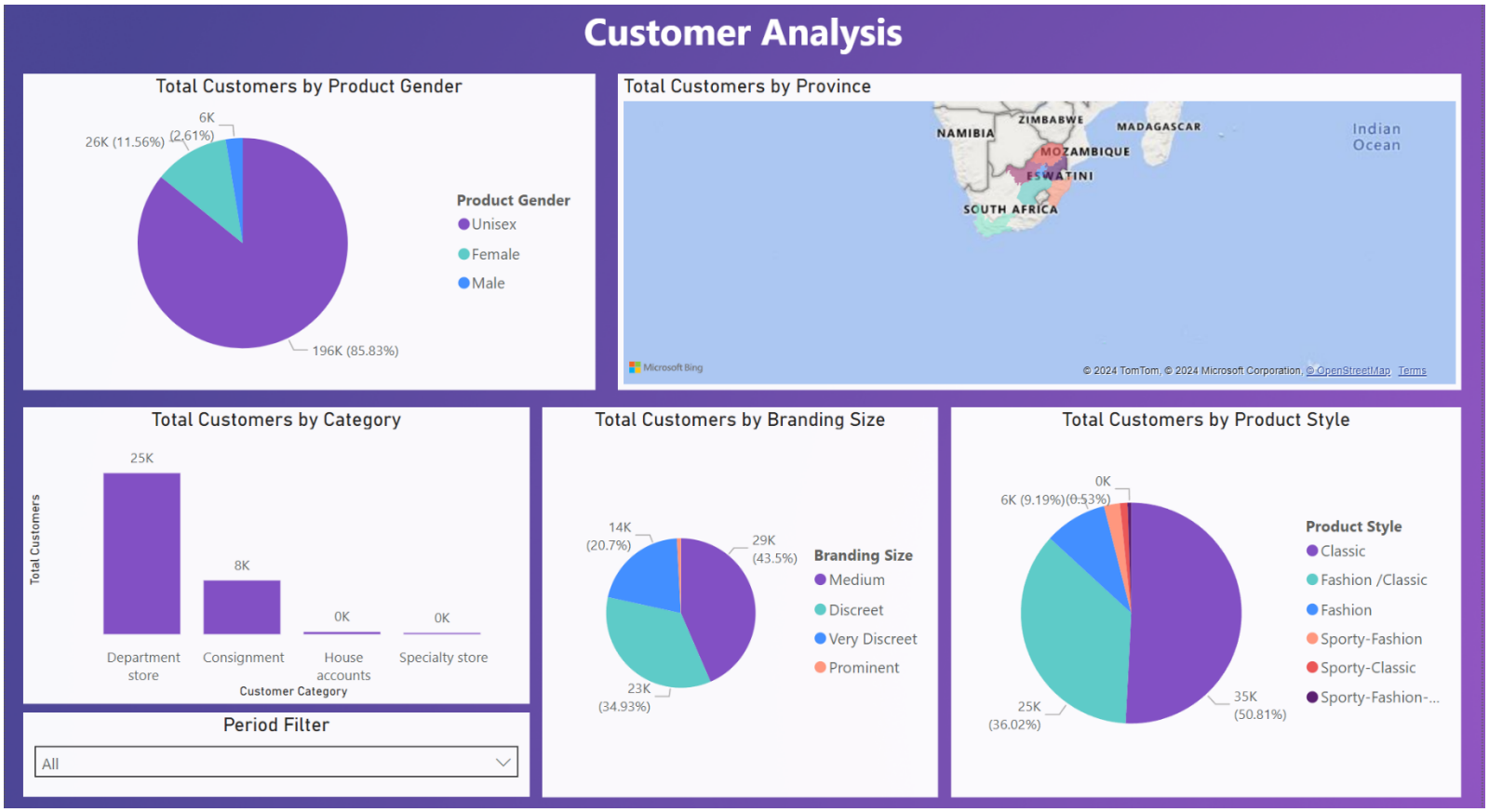

Figure 35. Customer Analysis Report

Customers are the cornerstone of any organisation which highlights the importance of understanding one's customer base. Figure 35 illustrates the core characteristics of ETec's customer base and encompasses product gender analysis, customer category analysis, preferred branding size analysis, and preferred product style analysis. These characteristics thoroughly describes the purchasing habits of ETec's customers and thus facilitates the required understanding. Furthermore, the dashboard on which the report is based includes a region map analysis that describes the locations of ETec's customers. The dashboard also enables filtering capabilities based on a chosen time-period.

Take note that this report is not based on specific queries and was generated using the functionality provided by Power BI, which abstracts query formulation.

Product Sales Analysis

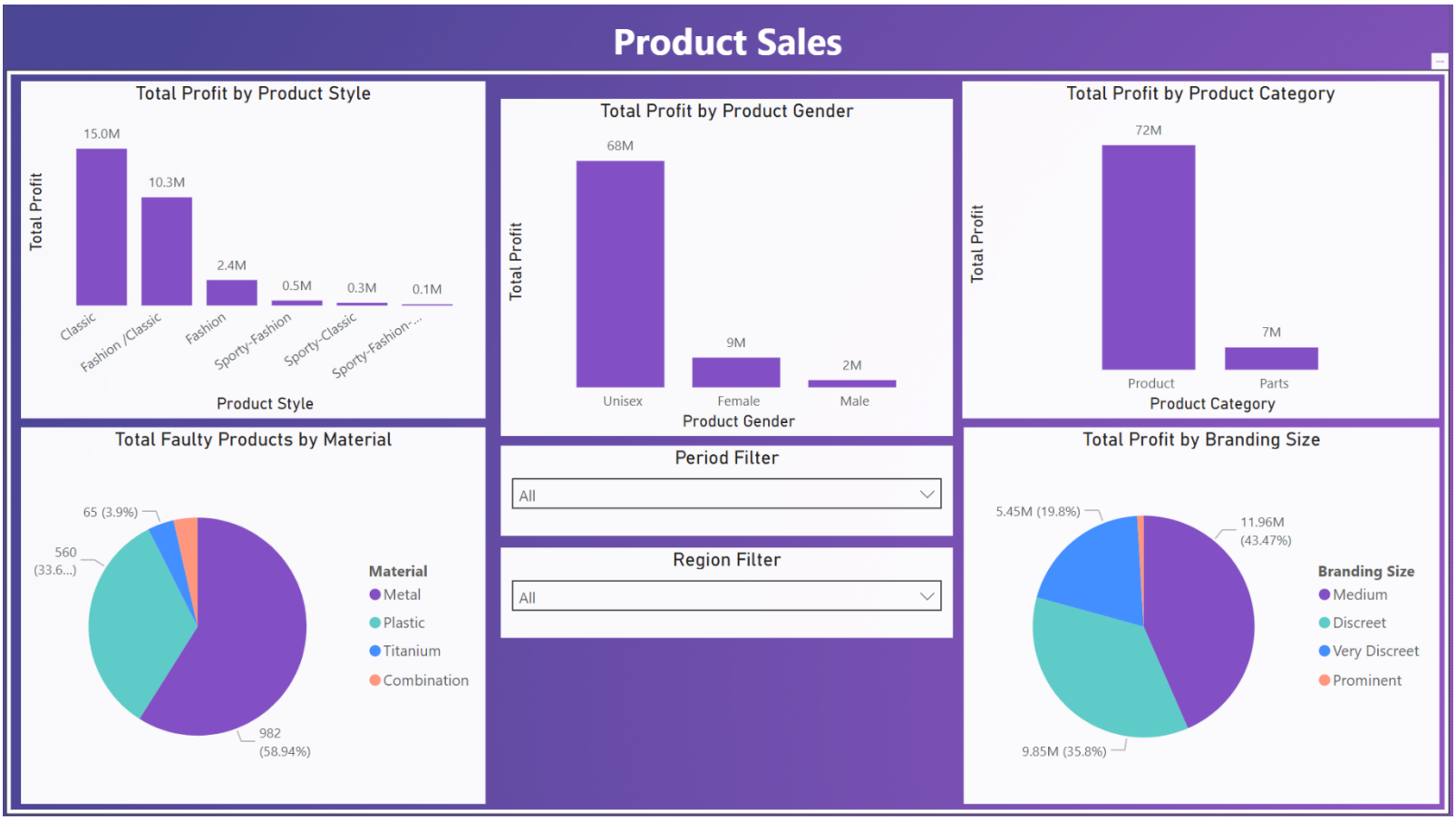

Figure 36. Product Sales Analysis

ETec's business function is primarily centred around providing products to customers; to ensure the quality of this function and ETec's relevancy, the popularity of specific products needs to be analysed along with the quality of these products. The report presented, in Figure 36, aims to address this important function by thoroughly describing and visualising faults per product material, profit by branding size, profit by product category, and profit by product style. Additionally, the popularity of products in terms of target gender is also investigated by examining total profits by product gender. The dashboard this report is based on enables filtering capabilities, namely, by time period and region.

Take note that this report is not based on specific queries and was generated using the functionality provided by Power BI, which abstracts query formulation.

Supplier Analysis

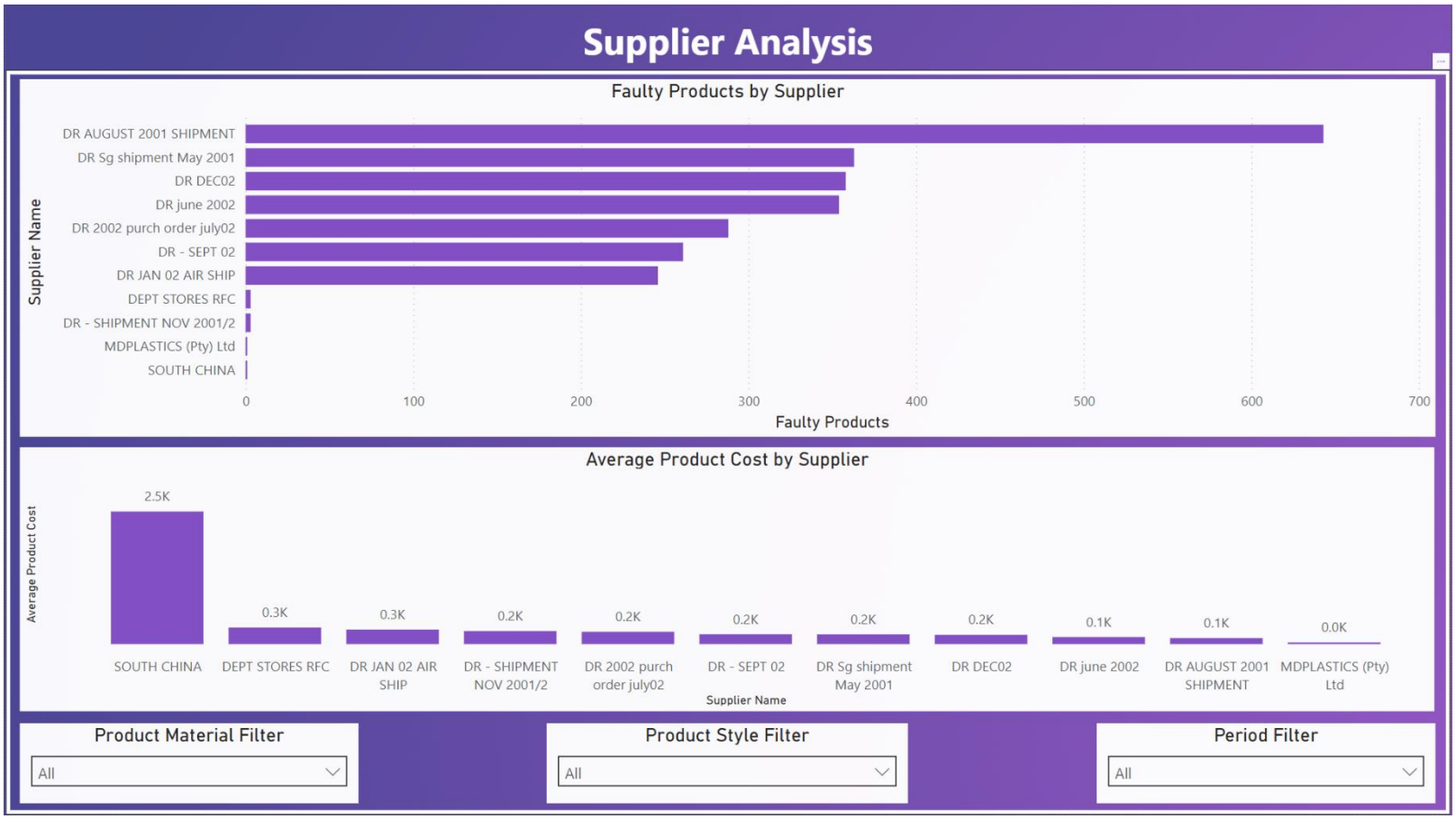

Figure 37. Supplier Analysis Report

As ETec aims to shift their analysis efforts from sales performance to supplier-based insights, the report, in Figure 37, was developed to facilitate their transition initiative. More specifically, this report enables the analysis of ETec's suppliers through illustrating the product faults per supplier and product cost per supplier. Note that the figures considered as part of this report are limited in nature due to the scarcity of supplier information in ETec's operational system; it is therefore recommended to initiate refactoring the operational system. Nonetheless, the presented figures in the corresponding dashboard can be filtered by product material, product style, and period of purchase.

Take note that this report is not based on specific queries and was generated using the functionality provided by Power BI, which abstracts query formulation.

Data Warehouse Performance Demonstration

This section aims to demonstrate the performance capabilities of the developed system through examining key performance indicators for queries expected to be used by ETec's management team, such as query execution speeds.

```
-- ============================================
-- TEST QUERY PERFORMANCE
-- ============================================
-- Statisitcs for performace analysis on
SET STATISTICS TIME ON;  -- Shows query execution time
SET STATISTICS IO ON;    -- Shows I/O statistics

-- Query
SELECT
    REP.representative_key AS 'Representative Identity',
    REGION.region_province AS 'Province',
    TEMP.calendar_year AS 'Date',
    SUM(FACT.sales_amount_with_discount) AS 'Sales Total'
FROM CUSTOMER_SALES_FACT AS FACT
    JOIN REPRESENTATIVE_DIM AS REP ON FACT.representative_key = REP.representative_key
    JOIN REGION_DIM AS REGION ON FACT.region_key = REGION.region_key
    JOIN DATE_DIM AS TEMP ON FACT.date_key = TEMP.date_key
WHERE REGION.region_key = 2 AND TEMP.calendar_year = 2019
GROUP BY REP.representative_key, REGION.region_province, TEMP.calendar_year
ORDER BY REP.representative_key


-- Statisitcs for performace analysis off
SET STATISTICS TIME OFF;
SET STATISTICS IO OFF;
```

Figure 38. Representative Performance Analysis Query

This query supports employee performance analysis and connects sales performance to specific representatives to facilitate ETec's sales performance cantered management initiative.

Results | Messages

	Representative Identity	Province	Date	Sales Total
1	14	KwaZulu-Natal	2019	297.0000
2	24	KwaZulu-Natal	2019	3150.0000
3	37	KwaZulu-Natal	2019	46402.0000
4	39	KwaZulu-Natal	2019	7448.2500
5	42	KwaZulu-Natal	2019	6315.0000
6	43	KwaZulu-Natal	2019	5654.0000
7	44	KwaZulu-Natal	2019	7859.0000
8	46	KwaZulu-Natal	2019	5930.0000
9	47	KwaZulu-Natal	2019	2090.0000
10	73	KwaZulu-Natal	2019	10906.0000
11	79	KwaZulu-Natal	2019	34612.0000
12	80	KwaZulu-Natal	2019	1295.0000
13	111	KwaZulu-Natal	2019	868.8000

Figure 39. Representative Performance Analysis Query Result

Figure 39 indicates the result set obtained after executing the query presented in Figure 38. From the result set, it is clear that the performance of specific representatives can be analysed with this data, thus presenting a realistic analysis query ideal for performance demonstration purposes.

```
 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

(13 rows affected)
Table 'Worktable'. Scan count 0, logical reads 0, ph
Table 'Workfile'. Scan count 0, logical reads 0, phy
Table 'CUSTOMER_SALES_FACT'. Scan count 365, logical
Table 'DATE_DIM'. Scan count 1, logical reads 3, phy
Table 'REPRESENTATIVE_DIM'. Scan count 1, logical re
Table 'REGION_DIM'. Scan count 0, logical reads 2, p

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 4 ms.

Completion time: 2024-10-07T13:42:25.3868195+02:00
```

Figure 40. Representative Query Speed

As indicated by the performance metrics this query was executed almost instantaneously, which demonstrates the well optimised nature of the developed data warehouse as well as the effectiveness of the implemented indexing strategy.

```sql
-- ============================================
-- TEST QUERY PERFORMANCE
-- ============================================
-- Statisitcs for performace analysis on
SET STATISTICS TIME ON;  -- Shows query execution time
SET STATISTICS IO ON;    -- Shows I/O statistics

-- Query
SELECT
    PROD.product_code AS 'Product',
    TEMP.calendar_year AS 'Year',
    REGION.region_province AS 'Province',
    SUM(FACT.sales_amount_with_discount) AS 'Total Sales'
FROM CUSTOMER_SALES_FACT AS FACT
    JOIN REGION_DIM AS REGION ON FACT.region_key = REGION.region_key
    JOIN DATE_DIM AS TEMP ON FACT.date_key = TEMP.date_key
    JOIN PRODUCT_DIM AS PROD ON FACT.product_key = PROD.product_key
GROUP BY PROD.product_code, REGION.region_province, TEMP.calendar_year
ORDER BY TEMP.calendar_year, REGION.region_province


-- Statisitcs for performace analysis off
SET STATISTICS TIME OFF;
SET STATISTICS IO OFF;
```

Figure 41. Product Sales Analysis Query

Another query is presented in Figure 41, which provides detail regarding all products sold through the history of ETec. More specifically, each product's sales figures are presented per calendar year and per province.

Results Messages

	Product	Year	Province	Total Sales
1	786COL588	2015	East Cape	644.0000
2	RSIG664	2015	East Cape	298.0000
3	2331 COL 220	2015	East Cape	2129.0000
4	794COL642	2015	East Cape	395.0000
5	2428COL581	2015	East Cape	429.0000
6	2320COL220/50	2015	East Cape	921.0000
7	670 COL 103	2015	East Cape	3003.0000
8	1233COL700	2015	East Cape	3370.0000
9	2460COL531	2015	East Cape	3728.0000
10	RMIN724/48	2015	East Cape	215.0000
11	7576COLQ93	2015	East Cape	429.0000
12	BLANKS-B/MIRROR	2015	East Cape	1040.0000
13	BLANKS-GREEN	2015	East Cape	100.0000
14	8503COL982	2015	East Cape	94.0000
15	2463COL200	2015	East Cape	466.0000

Figure 42. Product_Analysis_Query_Results_Top15

This figure illustrates the first 15 rows of the 35 430 records returned through the execution of the query presented in Figure 41. Nonetheless, the appropriateness of this results set stems from the fact that it presents a commonly analysed sales perspective.

```
 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

(35430 rows affected)
Table 'CUSTOMER_SALES_FACT'. Scan count 8, logical
Table 'REGION_DIM'. Scan count 2, logical reads 4,
Table 'PRODUCT_DIM'. Scan count 2, logical reads 23
Table 'DATE_DIM'. Scan count 2, logical reads 34, p
Table 'Worktable'. Scan count 0, logical reads 0, p

 SQL Server Execution Times:
   CPU time = 591 ms,  elapsed time = 540 ms.

Completion time: 2024-10-07T14:04:17.7703289+02:00
```

Figure 43. Product Analysis Query Speed

As indicated by this figure, a significant amount of data was processed and returned; however, through the performance metrics, it is evident that the query was executed optimally, resulting in a considerably short response time considering the number of records retrieved and the complexity of the query.

Conclusion

In conclusion, this document thoroughly described the development of a data mart to address the requirements of ETec's management team. All aspects related to the identified topic were thoroughly documented to ensure the maintainability of the developed solution. Nonetheless, the performance of the developed system was also demonstrated through the execution of queries expected to be utilised by the company, which serves the purpose of supporting the claims of the overall solution's performance and suitability.

Additionally, data dashboards are presented and described to demonstrate the analytical capabilities of the solution for addressing the reporting requirements of ETec's management team. A logical flow of report generation is also described to ensure usability and documentation of functionality.

References

Cioloca, C. & Georgescu, M. 2011. Increasing database performance using indexes. *Database Systems Journal*, 2(2):13-22.

Gupta, R. 2019. *An overview of SQL Server data types*. https://www.sqlshack.com/an-overview-of-sql-server-data-types/ Date of access: 15 August.

Kimball, R., Ross, M., Thornthwaite, W., Mundy, J. & Becker, B. 2008. *The data warehouse lifecycle toolkit*. John Wiley & Sons.

Michael, A.V. & Ahirao, P. 2020. Improved use of ETL tool for updation and creation of data warehouse from different RDBMS. In. Proceedings of the 3rd International Conference on Advances in Science & Technology (ICAST).

Microsoft. 2023. *SQL Server and Azure SQL index architecture and design guide*. https://learn.microsoft.com/en-us/sql/relational-databases/sql-server-index-design-guide?view=sql-server-ver16 Date of access: 18 May.

Microsoft. 2024a. *Try SQL Server on-premises or in the cloud*. https://www.microsoft.com/en-za/sql-server/sql-server-downloads Date of access: 8 August.

Microsoft. 2024b. *Excel*. https://www.microsoft.com/en-id/microsoft-365/p/excel/cfq7ttc0hr4r Date of access:

Microsoft. 2024c. *Microsoft Excel*. https://www.microsoft.com/en-za/microsoft-365/excel Date of access: 7 May.

Microsoft. 2024d. *Power BI*. https://www.microsoft.com/en-us/power-platform/products/power-bi Date of access: 24 July.

Microsoft. 2024e. *Editions and supported features of SQL Server 2022*. https://learn.microsoft.com/en-us/sql/sql-server/editions-and-components-of-sql-server-2022?view=sql-server-ver16 Date of access: 1 October.

Najm, I.A., Dahr, J.M., Hamoud, A.K., Alasady, A.S., Awadh, W.A., Kamel, M.B. & Humadi, A.M. 2022. OLAP Mining with Educational Data Mart to Predict Students' Performance. *Informatica*, 46(5),

Appendix A

Entities in the relational source data
Age Analysis
Payment Lines
Customer Account Parameters
Payment Header
Representatives
Customer Regions
Customer Categories
Sales Header
Purchases Headers
Suppliers
Sales Line
Purchases Lines
Product Brands
Trans Types
Products Styles
Products
Product Categories
Product Ranges

Appendix B

Dimension	Attributes
Date	Date_key (Surrogate Key)
	Calendar_Year
	Calendar_Quarter
	Calendar_Month
	Calendar_YYYY_MM
	Calendar_Week_Number_In_Year
	Day_Of_Week
	Holiday_Indicator
	Weekday_Indicator
	Calendar_Month_Number_In_Year
	Day_Number_In_Calendar_Year
	Day_Number_In_Calendar_Month
	Fiscal_Month
	Fiscal_Month_Number_In_Year
Product	Product_Key (Surrogate Key)
	Product_Fault_Indicator
	Product_Code
	Product_Category
	Product_Brand
	Product_Gender
	Product_Style

	Product_Material
	Product_Colour
	Product_Branding
	Product_Is_Current
Customer	Customer_Key (Surrogate Key)
	Customer_Identification
	Customer_Category
	Customer_Is_Current
Region	Region_Key (Surrogate Key)
	Region_Country
	Region_Provice
	Region_City
Representative	Representative_Key (Surrogate Key)
	Representative_Type
	Representative_Commision_Type
	Representative_Commision
	Representative_Identification
Transaction	Transaction_Key (Surrogate Key)
	Transaction_Type
Supplier	Supplier_Key (Surrogate Key)
	Supplier_Exclusive
	Supplier_Name
	Supplier_Credit_Limit

Appendix C

Fact Table	Attributes
Customer_Sales_Fact	Date_Key (Primary-Foreign key)
	Product_Key (Primary-Foreign key)
	Customer_Key (Primary-Foreign key)
	Region_Key (Primary-Foreign key)
	Supplier_Key (Primary-Foreign key)
	Representative_Key (Primary-Foreign key)
	Transaction_Key (Primary-Foreign key)
	Document_Number (Degenerate Dimension)
	Quantity
	Sales_Amount

	Discount
	Discount_Percentage
	Sales_Amount_With_Discount
	Product_Sell_Price
	Product_Bought_Price
	Item_Capture_Time
Snapshot_Sales_Fact	Date_Key
	Product_Key
	Region_Key
	Supplier_Key
	Representative_Key
	Transaction_Key
	Quantity
	Total_Sales
	Total_Discount
	Total_Cost
	Gross_Profit