

UNIVERSITY OF ECONOMICS AND LAW

FACULTY OF INFORMATION SYSTEMS



FINAL PROJECT REPORT

DATA WAREHOUSE AND INTEGRATION

**TOPIC: Build data warehouse for managing risk in
manufacturing process**

Lecturer: Le Ba Thien, MSc

Group: Wabisabi

Ho Chi Minh City, December, 2023

Team Members

<i>No</i>	Full name	Student ID	Point / 10 (Individual Contribution)	Signature
<i>1</i>	Nguyễn Hoàng Duy Thông	K214162156	10	
<i>2</i>	Lục Minh Phú	K214161342	10	
<i>3</i>	Nguyễn Thị Thu Thảo	K214162154	10	
<i>4</i>	Hoàng Trần Thế Phúc	K214162150	10	

Commitment

We affirm that the discoveries presented in the thesis are the result of our original work and not copied from any other source. The information provided throughout the entire thesis is either based on personal knowledge or gathered from various sources. Each reference has been appropriately acknowledged and cited.

We are fully accountable for the statements made and are prepared to undertake any required disciplinary actions.

Ho Chi Minh City, 12/2023

Group Wabisabi

Acknowledgement

Foremost, we extend our deep appreciation to the faculty teachers at the Faculty of Management Information Systems for their invaluable assistance in completing this project, providing essential knowledge, expertise, and information.

Furthermore, we want to convey our genuine thanks to MSc. Le Ba Thien, our supervisor, for affording us the opportunity to conduct research and for the valuable guidance provided throughout this research endeavor.

Finally, despite our diligent efforts in carrying out the implementation process accurately, inevitable errors may occur. Consequently, we anticipate that suggestions from teachers and readers will be acknowledged and embraced.

Ho Chi Minh City, 12/2023

Group WabiSabi

Table of Contents

Team Members	i
Commitment	ii
Acknowledgement	iii
Table of Contents.....	iv
List of Figures.....	vii
List of Tables	ix
List of Algorithms	x
Table of Acronyms	xi
Abstract.....	xii
Chapter 1. Project Overview	1
1.1. Reasons.....	1
1.2. Objective.....	1
1.3. Objects and Scopes	2
1.4. Approaching	2
1.5. Project Structure	2
Chapter 2. Literature Review	3
2.1. Risk Management.....	3
2.2. Data Warehouse	4
2.3. Data Warehouse in Risk Management	5
Chapter 3. Theoretical Background	7
3.1. Data Warehouse	7
3.1.1. Inmon Approach: Top-Down Design	7
3.1.2. Kimball Approach: Bottom-Up Design	8
3.1.3. Data Warehouse Characteristics	8
3.1.4. Data Warehouse Schema	9

3.2.	ETL (Extract, Transform, Load)	10
3.3.	Slowly Changing Dimension - (SCD).....	12
3.3.1.	Definition	12
3.3.2.	Type of SCD	12
3.3.3.	Benefits and Drawbacks	13
Chapter 4.	Use Case Analysis	15
4.1.	Process and Problems Analysis	15
4.2.	Data Description and Meanings	18
4.3.	Metrics	23
4.3.1.	Purchasing	23
4.3.2.	Production	24
4.4.	Extract – Transform - Load	26
4.5.	Fact tables and measures	28
4.5.1.	Purchasing	28
4.5.2.	Production	28
4.6.	Schema Development.....	29
4.6.1.	Create Date Dimensional Table	29
4.6.2.	Table Relationship Defined	32
4.6.3.	Analyzing the correlation between the problem and the data warehouse model	33
4.7.	SCD Use Cases.....	33
4.7.1.	Describe the Context.....	33
4.7.2.	Implementation Process	33
Chapter 5.	Analysis with MDX	37
5.1.	Cube Building.....	37
5.2.	OLAP Analysis.....	40

5.2.1.	Identify Potential Vendors	40
5.2.2.	Demand Trends	44
5.2.3.	Manufacturing Analysis	47
Chapter 6.	Discussion and Future Works	54
6.1.	Discussion	54
6.2.	Solving Usecase	56
6.3.	Future Works	57
6.3.1.	Limitations of Current Research:.....	57
6.3.2.	Future Research Directions:.....	57
References	58

List of Figures

Figure 4.1. Supply chain management process	15
Figure 4.2. Source type in project.....	26
Figure 4.3. Illustrate processing 'Load'.....	27
Figure 4.4. Illustrate using the ADO.NET Destination	28
Figure 4.5. Dim_Date Table	29
Figure 4.6. Galaxy Schema.....	32
Figure 4.7. Dim_Product table	34
Figure 4.8. Data description of New_Products_Price table	34
Figure 4.9. SCD process	35
Figure 4.10. Data before using SCD type 2.....	35
Figure 4.11. Data after using SCD type 2	36
Figure 5.1. Data source - Data Source View - Cube - Dimensions of project	37
Figure 5.2. Data Source View of Production schema	38
Figure 5.3. Data Source View of Production schema	39
Figure 5.4. Rejection Rate Measure	39
Figure 5.5. Stocked Rate Measure.....	40
Figure 5.6. Redundant Cost Rate Measure	40
Figure 5.7. The results of Credit Rating of 1	41
Figure 5.8. The results of Credit Rating 2	41
Figure 5.9. Top 10 highest Rejection Rate suppliers.....	42
Figure 5.10. Results of evaluate the performance of vendors	43
Figure 5.11. Results of Rejection Rate	43
Figure 5.12. Result of Production time.....	44
Figure 5.13. Result of Products demands	45
Figure 5.14. Analyzing line total by quarter and product category	45
Figure 5.15. Results of analyzing line total by ordered months and product category	46
Figure 5.16. Results of line total rolling down from 2011 to 2014 by product categories	47
Figure 5.17. Results of analyzing reasons with the highest differences in lead time due to scrap.....	48

Figure 5.18. Analyzing actual cost by product category and year	48
Figure 5.19. Analyzing actual cost by product category and quarter	49
Figure 5.20. Analyzing actual cost by product category and ordered	50
Figure 5.21. Analyzing actual cost by product category and week day name	51
Figure 5.22. Top 10 products has highest redundant cost rate.	53
Figure 6.1. Production dashboard for reporting	55
Figure 6.2. Purchasing dashboard for reporting	56

List of Tables

Table 4.1. Table of Purchasing.PurchaseOrderDetail	18
Table 4.2 Table Purchasing.PurchaseOrderHeader	19
Table 4.3 Table Production.Product	21
Table 4.4. Production.WorkOrder Table	22
Table 4.5. Production.WorkOrderRouting	23

List of Algorithms

Algorithm 4.1. SQL query for updating prices of new products.....	35
Algorithm 5.1. MDX query for counting total vendors with specific criteria	41
Algorithm 5.2. MDX query for retrieving top vendors by rejection rate.....	42
Algorithm 5.3. MDX query for analyzing stocked rate by year and credit rating	42
Algorithm 5.4. MDX query for analyzing rejection rates by year and credit rating	43
Algorithm 5.5. MDX query for analyzing line total by year and product category	44
Algorithm 5.6. MDX query for analyzing line total by quarter and product category	45
Algorithm 5.7. MDX query for analyzing line total by ordered months and product category	45
Algorithm 5.8. MDX query for Yearly Product Category Sales Analysis.....	46
Algorithm 5.9. MDX query for analyzing reasons with the highest differences in lead time due to scrap	48
Algorithm 5.10. MDX query for analyzing actual cost by product category and year	48
Algorithm 5.11. MDX query for analyzing actual cost by product category and quarter	49
Algorithm 5.12. MDX query for analyzing actual cost by product category and ordered months	50
Algorithm 5.13. MDX query for analyzing actual cost by product category and week day name	51
Algorithm 5.14. MDX query for Analyzing Top Products by Redundant Cost Rate ..	52

Table of Acronyms

Acronyms	Meaning
SCM	Supply Chain Management
DWH	Data Warehouse
MDX	Multidimensional Expressions
SCD	Slowly Changing Dimensions
ADO.NET	ActiveX Data Objects for .NET
SQL	Structured Query Language
ETL	Extract, Transform, Load
OLAP	Online Analytical Processing
KPI	Key Performance Indicator
DIM	Dimension
SSIS	SQL Server Integration Services

Abstract

The document details a strategic project aimed at transforming business performance in the manufacturing industry through a dual approach: optimizing supply chain management and implementing an advanced data warehouse system. The project's core objective is to enhance procurement efficiency by carefully managing the supply of raw materials in alignment with market demands and customer trends. This involves identifying reliable vendors, forecasting market dynamics, and ensuring a seamless supply chain operation. By doing so, the project seeks to reduce procurement costs, bolster customer satisfaction, and navigate the complexities of a rapidly evolving business environment.

Additionally, the project underscores the importance of a data warehouse system, designed to improve data accessibility and consistency within the organization. This system is set to provide real-time feedback, aiding swift and effective decision-making processes. The expected outcome is a more agile, responsive, and data-driven approach to managing business operations, leading to increased efficiency and cost savings.

Covering a period from July 2011 to June 2015 and utilizing AdventureWorks sales transaction history, the project adopts a methodical approach. It involves defining data table content, selecting necessary data for system analysis, and developing scales for key variables affecting manufacturing efficiency and effectiveness. With a structure comprising a project overview, literature review, theoretical background, use case analysis, and analysis with MDX, the document presents a well-rounded strategy aimed at optimizing processes and enhancing performance in the competitive landscape of today's business environment.

Chapter 1. Project Overview

In Chapter 1, from reasons of enhancing business performance and reducing cost optimally, the project proposed some objectives and identify scope and object, the authors discusses about general information of this project.

1.1. Reasons

The team's decision to investigate the optimization of supply chain management processes stems from the desire to achieve cost reductions, enhance customer satisfaction, and effectively address the complexities of a dynamic business environment. The primary focus is on optimizing risk in the procurement of raw materials by identifying potential vendors and forecasting market demand, ensuring a seamless supply chain. Streamlining these operations not only lowers procurement costs but also improves the speed and accuracy of feedback in product delivery, fostering stronger customer relationships. The ability to swiftly adapt and remain flexible becomes instrumental in navigating market fluctuations, while robust risk management practices facilitate the anticipation and minimization of disruptions in the supply chain. Ultimately, this research not only unlocks opportunities for enhancing overall business performance but also serves as a cornerstone for competitiveness and sustainability in the contemporary business landscape.

1.2. Objective

Optimize the process of purchasing input materials: To enhance the process of procuring raw materials, we propose a comprehensive strategy that involves identifying potential suppliers and concurrently relying on customer demand trends to determine the required quantity of materials. This approach not only aids in selecting the most quality-conscious and cost-effective supply partners but also adjusts input quantities based on actual customer demand. In this manner, the raw material procurement process becomes not only more efficient but also contributes to cost savings and optimal inventory management. Identifying suppliers with the best performance and adjusting material quantities according to market demand trends helps maintain flexibility and readiness to respond swiftly to market changes.

Build a warehouse data system to optimize processes: In doing so, our aim is to broaden the accessibility and utilization of data within the organization while concurrently minimizing issues associated with inconsistency and delays in information updates. The Data Warehouse system will establish a reliable source of information, facilitating various departments to operate based on a unified database. Additionally, it provides real-time feedback capabilities to support swift and effective decision-making. This promises to deliver efficiency and flexibility in data management, enhancing the organization's responsiveness to dynamic market demands.

1.3. Objects and Scopes

- Objects: AdventureWorks 2019
- Scopes:
 - Time scope: Time of business transactions: July 2017 - June 2020.
 - Space scope: Use the business's sales transaction history of AdventureWork.

1.4. Approaching

To complete the research goal of optimizing and streamlining the supply chain management process, the team will first take a general approach. Clearly define the content of data tables. Filter and select necessary data for system analysis. Next, the goal is to develop a scale for variables that are important to the efficiency and effectiveness of the manufacturing process. In this way, the team wishes to offer specific strategies and solutions to optimize processes and improve performance in today's competitive business environment.

1.5. Project Structure

Chapter 1: Project Overview

Chapter 2: Literature Review

Chapter 3: Theoretical Background

Chapter 4: Use case analyst

Chapter 5: Analysis with MDX

Chapter 2. Literature Review

Chapter 2 focuses on summary information of scientific research articles in Risk Management, Data Warehouse for readers to consider the roles of Risk Management in business performance as well as the importance of IT solutions like Data Warehouse in digital world. From that, the combination between Data Warehouse and Risk Management can be made use for the process to be performed fluently through evidence from the researches.

2.1. Risk Management

The single flow of production from raw material suppliers to manufacturers and then to markets. Nowadays, due to shorter product lifecycle and increasing demand among all businesses, they want to perform the production process effectively, smoothly as well as optimizing manufacturing costs by limiting risks which can be happened in operation. Risk management plays an important role in production in order for product quality to be better and for enterprises to handle the arise problems. According to (Tupa, J., Simota, J., & Steiner, F., 2017), risk management is one of the nine knowledge areas propagated by the Project Management Institute (PMI) and is probably the most difficult aspect of project management. Generally, risk management is a “method of managing that concentrates on identifying and controlling the areas or events that have a potential of causing unwanted change” (Pritchard, C. L., & PMP, P. R., 2014). With the discussion of (Banaitene, N. and Banaitis, A., 2012), risk management in the project management context is a comprehensive and systematic way of identifying, analyzing and responding to risks to achieve the project objectives, and this helps organizations understand what the risk is, who is at risk, what the current controls are for those risks and the judgements that need to be made about whether or not such controls are adequate. To implement the risk management process successfully, businesses need to understand what they undertake in each step. (Berg, H. P., 2010) supposed that the risk management includes 7 steps: (i) Establishing goals and context, (ii) identifying risks, (iii) analyzing the identified risks, (iv) assessing or evaluating the risks, (v) treating or managing the risks, (vi) monitoring and reviewing the risks and the risk environment regularly, and (vii) continuously communicating, consulting with stakeholders and reporting. Moreover, the risk management process is performed in many companies

which are in different fields. The article of (Houghton, J. R. and et al., 2017) presents a detailed examination of how food risks are managed in Western Europe, the evolution of FRM practices, and the various perspectives on FRM quality. It highlights the importance of considering multiple viewpoints and the challenges involved in implementing effective FRM policies. In Dutch,(Meuwissen, M. P. M., Huirne, R. B. M. & Hardaker, J. B, 2001) emphasizes the importance of understanding the highly individualistic and farm-specific nature of risk perceptions among farmers. This understanding is crucial for developing effective risk management strategies and policies tailored to the diverse needs of the agricultural sector by the methodology, which involves a questionnaire survey conducted among a large sample of livestock farmers in the Netherlands.

2.2. Data Warehouse

In the era 4.0, digital world is developed more and more, therefore, data is indispensable thing nowadays, especially as most of enterprises in various fields. In order to manage big data as well as support subject – oriented decision – making, Data Warehouse is implemented in businesses which can operate more optimally and perform effectively. Enterprise data warehouses have the potential to provide substantial benefits to organizations by enhancing their strategic advantage when implemented effectively. Nevertheless, initiatives in this field continue to pose considerable challenges and come with significant risks. With the global economy currently facing upheaval, organizations must possess pertinent information to navigate the worldwide economic downturn successfully. Leveraging technology, such as business intelligence and enterprise data warehousing, can aid organizations in making the difficult decisions required for survival. It's worth noting that data warehousing projects typically account for over 10% of the IT budgets of many corporate entities, with an estimated failure rate of approximately 50%. (Legodi, I., Barry, M. L., 2010) . In the pharmaceutical R&D supply chain, (Alshawhi, S., Saez-Pujol, I., & Irani, Z., 2003) discussed that IT departments applied data warehouse to integrate 13 external data sources and 19 internal data sources to the organisation in order to store, relate, and analyse the information, so that it can support decision making at the different organizational levels: strategic, management control, knowledge-level and operational control. Importantly, data warehouse

implementation reduces time of accessing information, improve data quality and organizational productivity to increase the time of the drug to be on market with patient protection as well as enhance competitive advantage against their competitors. In learning institutions, the article of (Yu, X., 2021) highlights the role of data warehousing in supporting decision-making and data analytics within organizations. It emphasizes the potential benefits of data warehousing in the academic processes of educational institutions, particularly those with multiple branches. The article explains the need for such systems in education institutions and how they can be implemented to aid in management. It also reviews various methods for data mining and outlines stages for promoting and actualizing these systems.

2.3. Data Warehouse in Risk Management

With the significance of Data Warehouse, it can be deployed in risk management project. Although this application in the world is quite new, Data Warehouse techniques have created the great value for the firms. (Nielsen, A. C., 2011) discusses the establishment of a veterinary data warehouse by the Danish Veterinary and Food Administration (DVFA). This warehouse integrates various existing databases in Denmark to facilitate access to comprehensive data regarding animal husbandry, health, and welfare in production animals. Therefore, this comprehensive approach to integrating and utilizing data reflects a significant advancement in the field of veterinary science and animal welfare, particularly in the context of risk management and communication. Next, the authors of (Chunying, Z., Weiqing, G., 2009) propose a platform based on data warehouses to enhance risk management capabilities for these companies. Key aspects of the platform include real-time warning systems, post-clearance audit, risk decision support, and risk mining functions. The platform is designed to support various functions essential for risk management in securities companies. These include real-time monitoring of business and computer/network operations, auditing of transactions and financial activities, and support for risk-based decision-making. The platform also employs data mining techniques to identify potential risks and establish correlation and classification rules for managing these risks. In Lagos, Nigeria, facing to the challenges posed by the existing system of cargo scanning and data handling at the ports, where security risks are a major concern due to

the importation of dangerous goods, (Wilson, N., Olufunke, A. F., 2016) proposed data warehouse model to overcome these challenges. This model aims to provide a centralized repository for data generated from cargo scanning operations. It is designed to facilitate efficient data retrieval, analysis, and intelligence sharing among various stakeholders. This solution is expected to significantly enhance the efficiency and effectiveness of security and risk management at Nigerian ports, aiding in the prevention of illegal imports and improving overall national security. In conclusion, data warehouse implementation is essential for risk management in digital world so as for businesses, including startups to corporation to optimize the cost of operating processes.

Chapter 3. Theoretical Background

In this chapter, before implementation stage, theoretical basis is mentioned to Data Warehouse techniques which are applied in this project. Each technique or method is presented in detail for clearly understanding as well as deploying the project effectively.

3.1. Data Warehouse

In the field of data warehousing, two primary methodologies have emerged, developed by Bill Inmon and Ralph Kimball. These approaches, while aiming at the same goal of creating efficient data warehouses, differ significantly in their philosophy, implementation, and management.

3.1.1. Inmon Approach: Top-Down Design

Kimball defines the data warehouse as "A copy of transactional data specifically structured for query and analysis" (R. Kimball and M. Ross, 1996). In addition, according to Kimball, the purpose of a data warehouse is "to provide information to support decision-making in a company" (R. Kimball and M. Ross, 2013). Therefore, the data warehouse is a dedicated database used in the context of decision-making and analysis.

- Centralized Data Repository: Inmon advocates for creating a centralized data warehouse that contains all the organizational data in a normalized form. This concept focuses on building a comprehensive database where every piece of data is non-redundant and consistent.
- Enterprise-Wide Integration: This approach emphasizes integrating data across the entire enterprise, covering all subject areas and providing a complete view of the organization.
- Normalization: The data in an Inmon-style warehouse is typically normalized to 3rd Normal Form (3NF), which reduces redundancy and improves data integrity.
- Data Marts: Inmon sees data marts as subsets of the data warehouse. These are created for specific lines of business or departments, and are designed after the main data warehouse is built.
- Iterative Development: This methodology often requires a longer initial implementation time due to its comprehensive and enterprise-wide scale. It is

usually more suitable for large organizations that need a detailed analysis across different departments.

3.1.2. Kimball Approach: Bottom-Up Design

- **Data Mart Focus:** Kimball's methodology starts with building data marts for specific business processes. These data marts are designed to provide quick access to frequently needed data.
- **Dimensional Modeling:** This approach uses a star schema (or snowflake schemas in some cases) for organizing data in the data warehouse. This involves a fact table linked to various dimension tables, making it more understandable and accessible for end-users.
- **Rapid Development:** The bottom-up approach allows for the faster implementation of data marts, providing quick wins and visible results to business users. This can be particularly advantageous for small to medium-sized businesses or individual departments within larger organizations.
- **Integration of Data Marts:** Over time, these individual data marts can be integrated to form a comprehensive data warehouse, though this integration can be complex if not planned properly from the beginning.
- **Denormalization:** In contrast to the Inmon approach, Kimball often involves some level of denormalization for performance optimization and easier user access.

3.1.3. Data Warehouse Characteristics

In the following, we explain the characteristics mentioned in the precedent definition (Yessad, L. and Labiod, A.):

- **Subject-oriented:** the data is linked to the company's business and organized by function.
- **Integrated:** signifies that the data obtained from several operational and external systems must be met, which involves solving problems because of data definition and content differences, such as different formats and coding of data.
- **Time-variant:** data are identified per specific periods; it means we keep the history of all transactions.

- Non-volatile: data are used for queries and cannot be changed. So, the update operations are not allowed, only reading is possible. In summary, a data warehouse is a centralized repository that stores the operational data in a specific way and makes them available and usable for analysis.

3.1.4. Data Warehouse Schema

A data warehouse schema serves as a logical representation of the entire database, delineating the structure and organization of data within the data warehouse. Three primary types of data warehouse schemas include the Star Schema, Snowflake Schema, and Galaxy Schema (alternatively termed Fact Constellation Schema), as identified by (Hecht, J., 2019).

The Star Schema is not only the most widely used but also the simplest schema. It comprises a central fact table surrounded by multiple related dimension tables. The fact table holds primary data within the data warehouse and is encircled by smaller dimension lookup tables providing details for various fact tables. In each dimension, the primary key is linked to a foreign key in the fact table. This implies that the fact table incorporates two column types: foreign keys connecting to dimension tables and measures containing numerical facts. At the core of the star schema lies the fact table, while the surrounding points of the star represent the dimension tables. Fact tables adhere to the Third Normal Form (3NF), while dimension tables are structured in denormalized form. Importantly, each dimension in the star schema should be exclusively represented by a one-dimensional table (StreamSets., 2021).

The Snowflake Schema exhibits a higher level of complexity and normalization compared to the star schema. In this schema, dimension tables undergo normalization, resulting in their decomposition into smaller tables. This normalization process is aimed at reducing data redundancy and enhancing data integrity. The term "snowflake" is used to describe the schema due to the visual resemblance of its diagram to a snowflake. Despite its increased complexity, the snowflake schema maintains the fact table at the core, with dimension tables connected through a series of one-to-many relationships. While more intricate than the star schema, the snowflake schema offers greater flexibility and is capable of handling more complex queries (Software Testing Help., 2021).

The Galaxy Schema is a combination of the star schema and the snowflake schema. It consists of multiple fact tables and is used for data mining queries. The galaxy schema is also known as the fact constellation schema because it resembles a constellation of stars. The galaxy schema is more complex than the star schema but less complex than the snowflake schema. It is used when there are multiple fact tables that are not related to each other. The galaxy schema is useful when there are multiple business processes that need to be analyzed separately (EDUCBA). The Galaxy Schema is a combination of the Star Schema and the Snowflake Schema. Each schema has its own advantage and disadvantage but we will develop galaxy schema to fulfill our specific requirements.

3.2. ETL (Extract, Transform, Load)

The Extract, Transform, Load (ETL) process plays a vital role in data warehousing, data mining, and business intelligence. This process encompasses the extraction of data from diverse and varied sources, its transformation, and subsequent loading into the data warehouse. Serving as a cornerstone for any data warehouse, the ETL process is fundamental in guaranteeing the accuracy, consistency, and currency of the data (Nwokeji, J. C. et al, 2018).

The ETL process initiates by extracting data from diverse sources, including databases, flat files, and web services. Subsequently, the extracted data undergoes transformation to adopt a format conducive to analysis and reporting. This transformative phase encompasses activities such as data cleaning, integration, enrichment, and validation. Ultimately, the refined data is loaded into the data warehouse, rendering it accessible to business analysts, data scientists, and other stakeholders for utilization (Nwokeji, J. C. et al, 2021).

- Extraction: This initial phase of the ETL process constitutes data extraction, encompassing the collection and retrieval of information from diverse and heterogeneous sources. These sources may encompass structured data, such as SQL databases, semi-structured data like XML files, or unstructured data, including text documents. The extraction process is meticulously designed to efficiently gather data while minimizing disruption to the source systems.

- Transformation: Once data is extracted, it undergoes transformation, a crucial step to ensure that it fits the business needs and the schema of the target system (like a data warehouse). Transformation can include a wide range of processes, such as cleaning (removing errors or inconsistencies), standardizing (ensuring uniform formats), de-duplicating, verifying, sorting, and joining data from different sources. Advanced transformations may also involve data enrichment, where data is enhanced using additional sources, or aggregation, where data is summarized or detailed according to specific requirements.
- Loading: The final stage involves transferring the processed data into the target system for further use in business analysis, reporting, or decision-making. Loading can be done in different ways: a full load, where the entire data set is loaded, or incremental load, where only new or changed data is added. This step must ensure that the transfer is secure, efficient, and does not disrupt the operations of the target system.

The ETL process which is a complicated and time-consuming task is requires careful planning and execution. It involves several stages, including data profiling, data mapping, data transformation, and data loading. The success of the ETL process depends on the quality of the data, the efficiency of the transformation process, and the accuracy of the loaded data.

In summary, the ETL process is a crucial and integral aspect of data warehousing, data mining, and business intelligence. Its core activities involve extracting data from diverse sources, transforming it, and subsequently loading it into the data warehouse. Although the ETL process is intricate and time-consuming, its significance lies in ensuring the accuracy, consistency, and currency of data. Key technologies associated with ETL include data extraction, transformation, incremental loading, and break-points transmission. The success of the ETL process is contingent upon data quality, the efficiency of the transformation process, and the accuracy of loaded data. A detailed exploration of the ETL process, encompassing its steps and considerations for selecting ETL tools, is provided. Various ETL tools, such as SSIS, Xplenty, AWS Glue, Talend, Stitch, Skyvia, and Informatica PowerCenter, are currently in use. For this project, SSIS

will be employed for implementing data integration with ETL (Sreemathy, J. et al, 2021).

3.3. Slowly Changing Dimension - (SCD)

3.3.1. Definition

The term "Slowly Changing Dimension" (SCD) pertains to a concept within the realms of data management and data warehousing (DWH), which elucidates the transformation of data over a period. SCD encompasses the approach employed in data warehousing systems to update, monitor, and uphold information as details about entities undergo evolution over time.(Windisch, M et al, 2007). The integration of Slowly Changing Dimension (SCD) is typically situated within the design and implementation phase of a Data Warehouse (DWH) project, particularly in the Extract, Transform, Load (ETL) process. This involves extracting data from the source, transforming it to conform to the DWH model, and subsequently loading it into the data warehouse.

3.3.2. Type of SCD

Within the domain of Data Warehousing, various types of Slowly Changing Dimensions (SCDs) are employed to oversee and monitor alterations in data over time. In this field, practitioners commonly leverage three principal SCD types, each equipped with its unique methodology for managing changes. Let's explore these three primary SCD types:

Type 1 - Overwrite/No history: In this SCD type, when changes occur in the source data, the existing record in the dimension table is overwritten with the updated information. No historical data is preserved, with the emphasis on reflecting only the most recent state of the dimension. This type is suitable when historical tracking is not crucial or is of lesser importance.

Type 2 - Add new row/history-preserving: When changes occur, a new tuple is added to the dimension table to capture the modified data while retaining the existing records. Historical versions of the entity are preserved, usually with effective dates or timestamps, offering a comprehensive historical perspective. Although it provides detailed historical information, it can result in larger dimension tables. SCD2 is typically used when there's a need to track the history of changes and analyze data over time.

Type 3 - Add columns/partial history: Specific columns within the dimension table are added to store both the current and previous versions of certain attributes. Partial historical information is maintained, focusing on specific attributes rather than the entire entity. This type is chosen when a balance between historical context and table size is sought.

3.3.3. Benefits and Drawbacks

- Benefits
 - Historical tracking: SCD allows for the preservation of historical data, providing a comprehensive record of how entities have changed over time. This historical tracking is valuable for trend analysis, reporting, and understanding the evolution of data.
 - Accurate reporting: By maintaining historical versions of data, SCD ensures that reporting reflects the state of entities at specific points in time. This accuracy is essential for business intelligence and decision-making processes.
 - Auditability: SCD facilitates auditability by keeping a detailed record of changes. This is beneficial for compliance purposes, allowing organizations to trace and understand modifications to their data.
 - Flexibility: SCD provides flexibility in handling different types of changes. Depending on the specific requirements, organizations can choose the appropriate SCD type to balance historical accuracy and storage considerations.
- Drawbacks
 - Increased storage requirements: Particularly with Type 2 SCD, where new rows are added for each change, storage requirements can increase significantly over time. This may lead to larger database sizes and increased resource utilization.
 - Complexity in maintenance: Managing and maintaining SCD structures, especially in scenarios involving frequent changes, can become complex. It requires careful planning and execution to ensure the integrity of historical data.

- Query performance: In situations where dimension tables become large due to historical preservation, query performance may be impacted. Retrieving and processing extensive historical data sets could result in slower query response times.
- Decision overhead: The decision on which SCD type to use involves trade-offs between historical accuracy and storage considerations. Organizations must carefully weigh these factors to make informed decisions, which can add a layer of complexity to the design process.

Chapter 4. Use Case Analysis

This chapter identifies the business problem, business requirements in order to interpreting the dataset and undertake implementation process. Besides, it focuses on how early stages in the Data Warehouse solution is performed , creating the premise for solving objectives in enterprise in Chapter 5.

4.1. Process and Problems Analysis

As we mentioned in Chapter 1, the objectives of this project is to help business gain competitive advantages and increase customer satisfaction though cutting order lead and manufacture time. First, we have some concepts about the components appeared in supply chain management process:

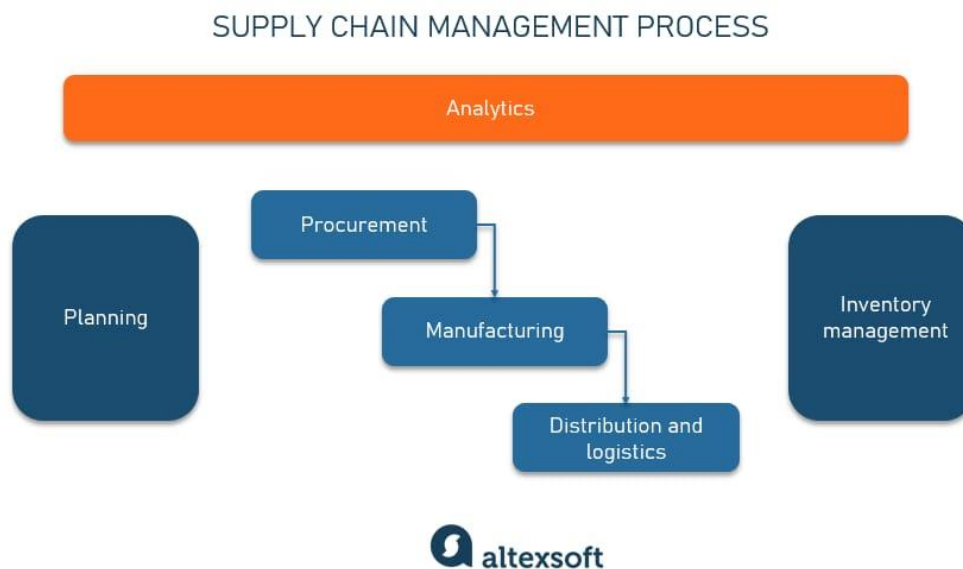


Figure 4.1.Supply chain management process

(Source:Altexsoft)

- Planning: Involves forecasting future demand and formulating corresponding strategies for each department, such as sourcing materials, scheduling staff, coordinating transportation, and more. This critical stage relies on comprehensive market information and analytics to ensure accurate predictions.
- Manufacturing: The process of converting raw materials into finished products using human labor, equipment, and, in some cases, other external factors. Certain businesses, particularly those in sectors like eCommerce or retail that deal with pre-made goods or services, may not include this stage in their supply chains.

- Procurement: Encompasses acquiring the appropriate quantity of materials at the best possible price, along with selecting suppliers and nurturing business relationships.
- Distribution and logistics: Involves locating customers, negotiating deals, organizing storage and transportation, and delivering goods. This stage also encompasses managing returns.
- Inventory management: Focuses on controlling the assortment and level of stock throughout the supply chain.
- Analytics: A common thread that runs through all stages, aiding in the monitoring of supply chain performance, identifying areas for enhancement, and facilitating decision-making.

To begin, the planning component will identify the consumer demand (make a prediction), the materials needed to manufacture, possible vendors,... This is followed by procurement component, they will choose the vendors with optimal price and time delivered that have been listed in planning step. Next, they will start the purchase order process, the procurement department issues a purchase order (PO) to approve the final purchase with the vendor. Upon the arrival of goods from the vendor, the packing slip is validated against the purchase order (PO) invoice. As a preparation for the following stage, the materials from vendor is used to craft product in manufacturing component before distributing to retailer or customer, products have to pass quality control, quality assurance process. In the manufacturing part, inventory management play an vital roles in the process of take out product for customers and take in materials from vendors. The last step is distribution and logistics component which finding the best logistic route and delivering goods to retailer and customer though third party delivery business. This one also interacting with inventory management to help sorting product in shelves.

In the context of AdventureWorks, there will be some limited in dataset and we are willing to strongly focus on manufacturing process so there are only two departments using for analysis: Purchasing, Production. To optimize the performance in these department to help increasing manufacturing efficiency and identifying potential vendor, there are some aspects and functions that we focus on each department. To identifying this, the business need to address questions about manufacturing problems:

“How to predict product quantity for effectively manufacture planning and reducing risk?”

To predict how much of a product to make and reduce risks in manufacturing, it's crucial to look at both buying and making processes. On the buying side, we meticulously examine Purchase Order details, including rejection and stock quantities in comparison to the ordered quantity. We also scrutinize the Average Lead Time to better plan and identify alternative suppliers if necessary. Delving into suborders within the Purchase Order Header provides additional insights into market demand. Vendor evaluation involves assessing credit ratings and the frequency of item rejections. The ActiveFlag status and credit ratio assist in gauging vendor risk. On the making side, we conduct a thorough analysis of work orders, comparing planned start times with actual start times. We assess deviations from expected timelines and their impact on costs. Tracking different product categories over time enables us to identify trends in our manufacturing processes. Additionally, we incorporate insights from analyzing products with high redundant cost rates, developing strategies to address them. Furthermore, our planning process includes the integration of defective products analysis, allowing us to compare actual results with set targets. This comprehensive approach enables us to make informed predictions, enhance planning strategies, and mitigate potential issues such as delays or increased costs. It's akin to having a robust plan that keeps us prepared for various scenarios that may arise in both the making and buying of products.

Then we split it into small questions to declare a clear view about steps to achieve this objective of the project:

- What is the percentage of rejection and stocking ratios to inform vendors quality?
- How many vendors currently active has high credit rating?
- How can we identify delays and cost implications in manufacturing work order routing?
- What is the product demand for each category in purchasing order?
- How the vendor perform based on their credit rating and average lead time ?

“How to ensure data transparency and consistency between departments in manufacturing process?”

Manufacturing process systems frequently operate in silos, with each department keeping its own databases and systems. This fragmented approach can result in various data transparency and consistency difficulties, such as data silos, inconsistent data formats, delayed data updates, and so on. If the data quality is poor, the business process cannot be optimized. So we decide to develop a data warehouse to enable data transparency across all departments, with real-time updates to assure no delays and consistent data structure.

- How the data warehouse help the business identifying the stages in the process that slow down the flow is crucial for improvement?
- How the data warehouse help enhancing data quality?

4.2. Data Description and Meanings

- Purchasing
 - Purchasing.PurchaseOrderDetail: This table is typically used to store detailed information about the items or products included in a purchase order and provide detailed information about the items, quantities, prices, and other relevant details associated with a purchase order. The relationships between this table and other tables in the database, such as PurchaseOrderHeader and Product, allow for comprehensive tracking and management of purchasing information within the AdventureWorks.

Table 4.1. Table of Purchasing.PurchaseOrderDetail

Name	Data type	Description
PurchaseOrderID	int	Primary key. Foreign key to PurchaseOrderHeader.PurchaseOrderID.
PurchaseOrderDetailID	int	Primary key. One line number per purchased product.
DueDate	datetime	Date the product is expected to be received.
OrderQty	smallint	Quantity ordered.
ProductID	int	Product identification number. Foreign key to Product.ProductID.
UnitPrice	money	Vendor's selling price of a single product.
LineTotal	money	Per product subtotal. Computed as OrderQty * UnitPrice.
ReceivedQty	decimal(8, 2)	Quantity actually received from the vendor.

RejectedQty	decimal(8, 2)	Quantity rejected during inspection.
StockedQty	decimal(9, 2)	Quantity accepted into inventory. Computed as ReceivedQty - RejectedQty.
DateKey	datetime	PrimaryKey of Dim_Date
ModifiedDate	datetime	Date and time the record was last updated. Default: getdate()

- Purchasing.PurchaseOrderHeader:** It is used to store information about purchase orders, representing the main header details for each purchase order. This table is connected to the Purchasing.PurchaseOrderDetail table, which contains detailed information about the items included in each purchase order. The Purchasing.PurchaseOrderHeader table is crucial for tracking and managing information related to purchase orders. It helps businesses keep records of orders, monitor their status, and analyze purchasing trends and expenses.

Table 4.2 Table Purchasing.PurchaseOrderHeader

Name	Data type	Description
PurchaseOrderID	int	Primary key.
RevisionNumber	tinyint	Incremental number to track changes to the purchase order over time. Default: 0
Status	tinyint	Order current status. 1 = Pending; 2 = Approved; 3 = Rejected; 4 = Complete Default: 1
VendorID	int	Vendor with whom the purchase order is placed. Foreign key to Vendor.BusinessEntityID.
OrderDate	datetime	Purchase order creation date. Default: getdate()
ShipDate	datetime	Estimated shipment date from the vendor.
SubTotal	money	Purchase order subtotal. Computed as SUM(PurchaseOrderDetail.LineTotal)for

		the appropriate PurchaseOrderID. Default: 0.00
TotalDue	money	Total due to vendor. Computed as Subtotal + TaxAmt + Freight. Computed: isnull(([SubTotal]+[TaxAmt])+[Freight],(0))
ModifiedDate	datetime	Date and time the record was last updated. Default: getdate()
StandardPrice	money	The vendor's usual selling price.
AverageLeadTime	int	The average span of time (in days) between placing an order with the vendor and receiving the purchased product.
OnOrderQty	int	The quantity currently on order.
VendorName	nvarchar(50)	Company name.
CreditRating	tinyint	1 = Superior, 2 = Excellent, 3 = Above average, 4 = Average, 5 = Below average
PreferredVendorStatus	bit	0 = Do not use if another vendor is available. 1 = Preferred over other vendors supplying the same product. Default: 1
ActiveFlag	bit	0 = Vendor no longer used. 1 = Vendor is actively used. Default: 1

- Production:

The Production schema in AdventureWorks serves as a blueprint for managing the complex operations of the manufacturing process. This schema consists of several interrelated tables, each capturing different facets of the manufacturing lifecycle, providing a wealth of data crucial for various research objectives.

- **Production.Product:**

This table is the core of the product catalog. It contains rich details on every item manufactured by AdventureWorks, including names, descriptions, and the hierarchical relationship between finished goods and their components. For research, this table is indispensable for understanding product composition, managing product lines, and analyzing the impact of product variations on manufacturing efficiency and sales performance.

Table 4.3 Table Production.Product

Name	Data type	Description
ProductID	int	Primary key for Product records
Name	Nvarchar(50)	Name of product
ProductNumber	Nvarchar(25)	Unique product identification number
MakeFlag	bit	0 = Product is purchased, 1 = Product is manufactured in house, Default: 1
StandardCost	money	Standard cost of the product
ListPrice	money	Selling price
DaysToManufacture	int	Number of days required to manufacture the product.
ProductSubcategoryID	int	Product is a member of this product model. Foreign key to ProductModel.ProductModelID. References Production.ProductModel
DiscontinuedDate	datetime	Date the product was discontinued.
ModifiedDate	datetime	Date and time the record was last updated. Default: getdate()
ProductCategoryName	nvarchar(50)	Category description. References: Production.ProductModelIllustration

- **Production.WorkOrder:**

The WorkOrder table is pivotal in tracking the actual manufacturing processes. It logs each work order issued for the production of goods, detailing quantities, due

dates, and the current status of production tasks. Vital for production planning and scheduling, this table helps in identifying production bottlenecks, ensuring that manufacturing capacity aligns with demand forecasts and customer orders, thus improving the overall supply chain flow.

Table 4.4. Production.WorkOrder Table

Name	Data type	Description
WorkOrderID	int	Primary key for WorkOrder records. Identity
ProductID	int	Product identification number. Foreign key to Product.ProductID. References: Production.Product
OrderQty	int	Product quantity to build.
StockedQty	int	Quantity built and put in inventory.
ScrappedQty	int	Quantity that failed inspection.
StartDate	smallint	Work order start date.
EndDate	datetime	Work order end date.
DueDate	datetime	Work order due date.
ScrapReasonName	smallint	Reason for inspection failure. References: Production.ScrapReason
ModifiedDate	datetime	Date and time the record was last updated. Default: getdate()

- Production.WorkOrderRouting Table:

This table is an essential record of the details of work orders in the production process.. For the purpose of the project, this table is crucial for tracking the manufacturing process of each product. It allows for a comprehensive analysis of production efficiency, resource utilization, and cost management. This detailed oversight is vital for optimizing production schedules, improving resource allocation, and reducing production costs. Additionally, insights from this table aid in fine-tuning

the manufacturing process, leading to more effective inventory management by aligning production closely with demand and reducing wastage or overproduction.

Table 4.5. Production.WorkOrderRouting

Name	Data type	Description
WorkOrderID	int	Primary key. Foreign key to WorkOrder.WorkOrderID. References: Production.WorkOrder
ProductID	int	Primary key. Foreign key to Product.ProductID.
OperationSequence	smallint	Primary key. Indicates the manufacturing process sequence.
ScheduledStartDate	datetime	Planned manufacturing start date.
ScheduledEndDate	datetime	Planned manufacturing end date.
ActualStartDate	datetime	Actual start date.
ActualEndDate	datetime	Actual end date
PlannedCost	money	Estimated manufacturing cost.
ActualCost	money	Actual manufacturing cost.
ModifiedDate	datetime	Date and time the record was last updated. Default: getdate()
DateKey	datetime	PrimaryKey of Dim_Date

By thoroughly analyzing these tables, researchers can gain deep insights into how the manufacturing process can be optimized. They can explore ways to enhance productivity, reduce costs, and ensure that the manufacturing operations align closely with the company's sales strategies and market demands. Each table in the Production schema not only supports operational needs but also acts as a rich data source for strategic analysis and decision-making in the pursuit of competitive advantage and customer satisfaction.

4.3. Metrics

4.3.1. Purchasing

Stocked rate:

- Measure: Stocked Rate is a metric that measures the percentage of items or products which was ordered divide by total items stocked.
- Formula: $\text{Stocked rate} = (\text{Order Quantity} / \text{Stocked Quantity}) * 100$.
- Description:
 - Stocked rate: StockedRate (Measure) in Purchasing.PurchaseOrderDetail.
 - Order Quantity: OrderQty in Purchasing.PurchaseOrderDetail.
 - Stocked Quantity: StockedQty in Purchasing.PurchaseOrderHeader.

Rejection rate:

- Measure: The Rejection Rate is a metric that measures the percentage of items or products received from a supplier that are rejected due to quality issues or other reasons. This metric is important for assessing the performance of suppliers and the quality of the materials or products they provide.
- Formula: $\text{Rejection Rate} = (\text{Rejected Quantity} / \text{Received Quantity}) * 100$.
- Description:
 - Rejection Rate: RejectionRate (Measure) in Purchasing.PurchaseOrderDetail.
 - Rejected Quantity: RejectedQty in Purchasing.PurchaseOrderHeader.
 - Received Quantity: ReceivedQty in Purchasing.PurchaseOrderHeader.

4.3.2. Production

Manufacture lead time

- Measure: Manufacturing Lead Time (MLT) refers to the total time required to produce a product from the initiation of the manufacturing process to its completion. It encompasses all the stages involved in manufacturing, from the receipt of raw materials to the finished product leaving the production line. MLT is a crucial metric in production planning, supply chain management, and overall operational efficiency.
- Formula: $\text{Manufacturing Lead Time} = \text{Actual End Date} - \text{Actual Start Date}$.
- Description:
 - Manufacturing Lead Time: ManufactureLeadTime (Measure) in Production.WorkOrderRouting table.
 - Actual Start Date: ActualStartDate in Production.WorkOrderRouting table.

- Actual End Date: ActualEndDate in Production.WorkOrderRouting table.

Scheduled Manufacturing Lead Time:

- Measure: Scheduled Manufacturing Lead Time (MLT) is the aggregate duration that has been prearranged and designated as necessary to manufacture a product, spanning from the initiation of the manufacturing process to its final completion.
- Formula: Manufacturing Lead Time = Scheduled End Date – Scheduled Start Date.
- Description:
 - Scheduled Manufacturing Lead Time: ScheduledManufactureLeadTime (Measure) in Production.WorkOrderRouting table.
 - Scheduled Start Date: ScheduledStartDate in Production.WorkOrderRouting table.
 - Scheduled End Date: ScheduledEndDate in Production.WorkOrderRouting table.

DifferentLeadTime:

- Measure: To determine whether production time is faster or slower than expected and to find out the root cause of the problem.
- Formula: Difference Lead Time = Scheduled Manufacturing Lead Time – Manufacturing Lead Time.
- Description:
 - Scheduled Manufacturing Lead Time: ScheduledManufactureLeadTime (Measure) in Production.WorkOrderRouting table.
 - Manufacturing Lead Time: ManufactureLeadTime (Measure) in Production.WorkOrderRouting table.

Different Scheduled Start Date:

- Measure: Date different between scheduled start date and actual start date.
- Formula: Different Scheduled Start Date = Scheduled Start Date – Schedule End Date.
- Description:
 - Different Scheduled Start Date: DiffScheduledStartDate (Measure) in Production.WorkOrderRouting.

- Scheduled Start Date: ScheduledStartDate in Production.WorkOrderRouting.
- Scheduled End Date: ScheduledEndDate in Production.WorkOrderRouting.

Cost per day

- Measure: Consider operating costs if production is not on schedule
- Formula: $\text{ActualCost} / (\text{Actual End Date} - \text{Actual Start Date})$
- Description:
 - ActualCost: Actual manufacturing cost.
 - Actual Start Date: ActualStartDate in Production.WorkOrderRouting table.
 - Actual End Date: ActualEndDate in Production.WorkOrderRouting table.

4.4. Extract – Transform - Load

First, our group's data comes from many different sources: XML, BAK, Excel.






Name	Date modified	Type	Size
 Production	11/20/2023 1:10 AM	BAK File	23,647 KB
 Purchasing.ProductVendor.xml	11/22/2023 7:59 PM	XML Document	207 KB
 Purchasing.PurchaseOrderDetail.xml	11/22/2023 8:12 PM	XML Document	3,803 KB
 Purchasing.PurchaseOrderHeader.xml	11/22/2023 8:17 PM	XML Document	1,947 KB
 Purchasing.Vendor.xml	11/22/2023 8:41 PM	XML Document	33 KB

Figure 4.2. Source type in project

We use SSIS (SQL Server Integration Services) to load them into a common database named 'Manufacturing.' In this scenario, our team will demonstrate the process of loading data from a single XML source into the database.

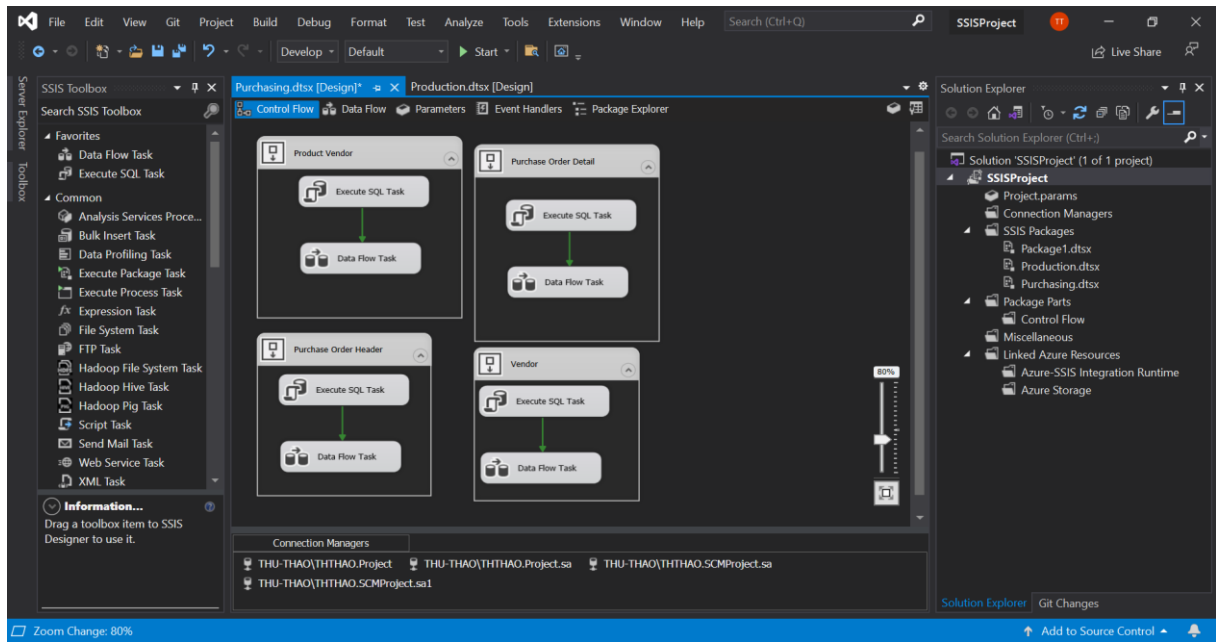


Figure 4.3. Illustrate processing 'Load'

Subsequently, we plan to employ four sequence containers to load four tables of Purchasing schema from the XML source. Each container will contain an Execute SQL Task connected to a Data Flow Task, meaning that during execution, SSIS will run the Execute SQL Task first to truncate the table, and then proceed to execute the Data Flow Task.

Since we don't frequently insert new data, we have opted for the Full Load approach as a method for ETL (Extract, Transform, Load) operations. In conclusion, our comprehensive ETL strategy, encompassing the Full Load approach, staging tables, error handling, parallel processing, and ongoing monitoring and maintenance, empowers us to efficiently and reliably integrate data from multiple sources into our database using the ADO.NET Destination.

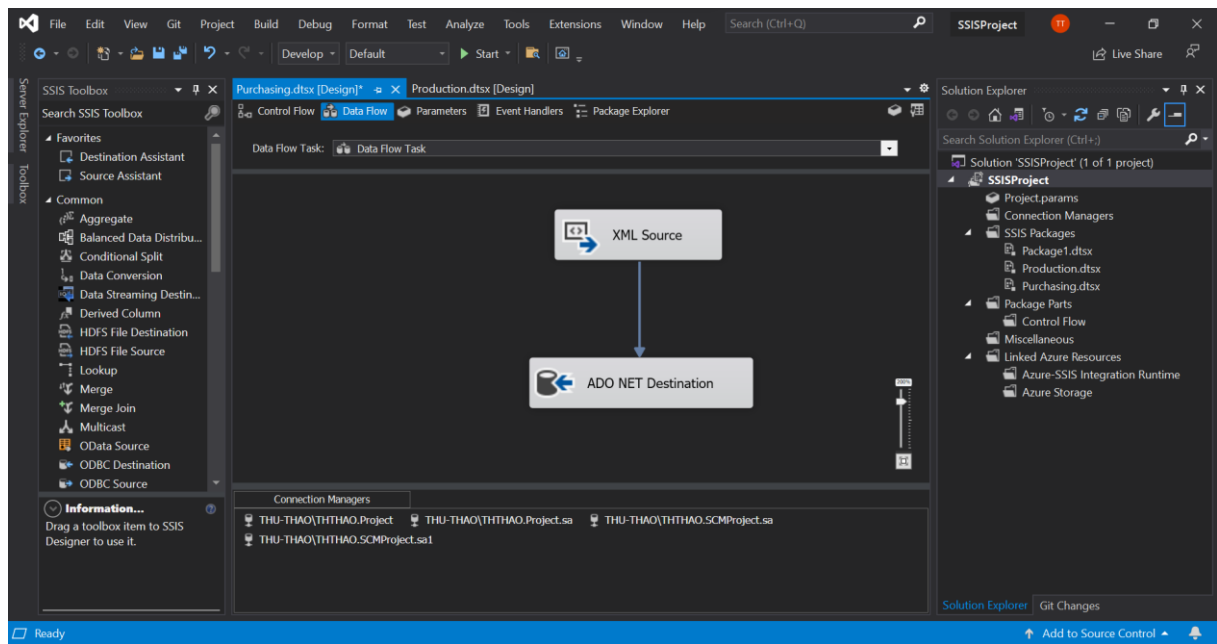


Figure 4.4. Illustrate using the ADO.NET Destination

4.5. Fact tables and measures

In this part, we will identify fact tables and measure of it for each department. Furthermore, we will illustrates the source codes of inserting measures added to fact table from different departments. All of the measures has already defined in 3.3.

4.5.1. Purchasing

- Fact table: Purchasing.PurchaseOrderDetail
- Measures: RejectionRate, StockRate

4.5.2. Production

- Fact table: Production.WorkOrderRouting
- Measures: ManufactureLeadTime, ScheduledManufactureLeadTime, DifferentLeadTime, DifferentScheduledStartDate

4.6. Schema Development

4.6.1. Create Date Dimensional Table


	Column Name	Data Type	Allow Nulls
	DateKey	int	<input type="checkbox"/>
	Date	date	<input type="checkbox"/>
	Weekday	tinyint	<input type="checkbox"/>
	WeekDayName	varchar(10)	<input type="checkbox"/>
	Month	tinyint	<input type="checkbox"/>
	MonthName	varchar(10)	<input type="checkbox"/>
	Quarter	tinyint	<input type="checkbox"/>
	QuarterName	varchar(6)	<input type="checkbox"/>
	Year	int	<input type="checkbox"/>
	FirstDateofYear	date	<input checked="" type="checkbox"/>
	LastDateofYear	date	<input checked="" type="checkbox"/>
	FirstDateofQuater	date	<input checked="" type="checkbox"/>
	LastDateofQuater	date	<input checked="" type="checkbox"/>
	FirstDateofMonth	date	<input checked="" type="checkbox"/>
	LastDateofMonth	date	<input checked="" type="checkbox"/>
	FirstDateofWeek	date	<input checked="" type="checkbox"/>
	LastDateofWeek	date	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figure 4.5. Dim_Date Table

The team creates 14 data columns in the Dim_Date table, of which the Datekey column will be assigned as the primary key to identify unit columns between columns, and this primary key is used to refer to other tables.

Specifically here, the Dim_Date table will be used to refer to the Fact_WorkOrderRouting table and the Fact_PurchaseOrderDetail table by using primary key DateKey.

References to these tables are intended to support further analysis of changes in order quantities and suppliers in the data set.

```
CREATE TABLE [dbo].[Dim_Date] (  
    [DateKey] [int] NOT NULL,  
    [Date] [date] NOT NULL,  
    [Weekday] [tinyint] NOT NULL,  
    [WeekDayName] [varchar](10) NOT NULL,  
    [Month] [tinyint] NOT NULL,  
    [MonthName] [varchar](10) NOT NULL,  
    [Quarter] [tinyint] NOT NULL,  
    [QuarterName] [varchar](6) NOT NULL,
```

```

[Year] [int] NOT NULL,
[FirstDateofYear] DATE NULL,
[LastDateofYear] DATE NULL,
[FirstDateofQuater] DATE NULL,
[LastDateofQuater] DATE NULL,
[FirstDateofMonth] DATE NULL,
[LastDateofMonth] DATE NULL,
[FirstDateofWeek] DATE NULL,
[LastDateofWeek] DATE NULL
PRIMARY KEY CLUSTERED ([DateKey] ASC)
)

SET NOCOUNT ON

TRUNCATE TABLE DIM_Date

DECLARE @CurrentDate DATE = '2011-01-01'
DECLARE @EndDate DATE = '2015-12-31'

WHILE @CurrentDate < @EndDate
BEGIN
    INSERT INTO [dbo].[Dim_Date] (
        [DateKey],
        [Date],
        [Weekday],
        [WeekDayName],
        [Month],
        [MonthName],
        [Quarter],
        [QuarterName],
        [Year],
        [FirstDateofYear],
        [LastDateofYear],
        [FirstDateofQuater],
        [LastDateofQuater],
        [FirstDateofMonth],
        [LastDateofMonth],
        [FirstDateofWeek],
        [LastDateofWeek]
    )
    SELECT DateKey = YEAR(@CurrentDate) * 10000 +
MONTH(@CurrentDate) * 100 + DAY(@CurrentDate),
        DATE = @CurrentDate,
        WEEKDAY = DATEPART(dw, @CurrentDate),
        WeekDayName = DATENAME(dw, @CurrentDate),
        [Month] = MONTH(@CurrentDate),
        [MonthName] = DATENAME(mm, @CurrentDate),
        [Quarter] = DATEPART(q, @CurrentDate),
        [QuarterName] = CASE
            WHEN DATENAME(qq, @CurrentDate) = 1
                THEN 'First'
            WHEN DATENAME(qq, @CurrentDate) = 2

```

```

        THEN 'second'
    WHEN DATENAME(qq, @CurrentDate) = 3
        THEN 'third'
    WHEN DATENAME(qq, @CurrentDate) = 4
        THEN 'fourth'
    END,
    [Year] = YEAR(@CurrentDate),
    [FirstDateofYear] = CAST(CAST(YEAR(@CurrentDate) AS
    VARCHAR(4)) + '-01-01' AS DATE),
    [LastDateofYear] = CAST(CAST(YEAR(@CurrentDate) AS
    VARCHAR(4)) + '-12-31' AS DATE),
    [FirstDateofQuater] = DATEADD(qq, DATEDIFF(qq, 0,
    GETDATE()), 0),
    [LastDateofQuater] = DATEADD(dd, - 1, DATEADD(qq,
    DATEDIFF(qq, 0, GETDATE()) + 1, 0)),
    [FirstDateofMonth] = CAST(CAST(YEAR(@CurrentDate) AS
    VARCHAR(4)) + '-' + CAST(MONTH(@CurrentDate) AS VARCHAR(2)) +
    '-01' AS DATE),
    [LastDateofMonth] = EOMONTH(@CurrentDate),
    [FirstDateofWeek] = DATEADD(dd, - (DATEPART(dw,
    @CurrentDate) - 1), @CurrentDate),
    [LastDateofWeek] = DATEADD(dd, 7 - (DATEPART(dw,
    @CurrentDate)), @CurrentDate)

    SET @CurrentDate = DATEADD(DD, 1, @CurrentDate)
END

```

```

update Purchase.Fact_PurchaseOrderDetail
set ModifiedDate = CONVERT(DATE, ModifiedDate)

```

```

update Purchase.Fact_PurchaseOrderDetail
set ModifiedDate = CONVERT(DATE, ModifiedDate)

```

```

update Purchase.Fact_PurchaseOrderDetail
set ModifiedDate = CONVERT(DATE, ModifiedDate)

```

```

alter table Production.Fact_WorkOrderRouting
add DateKey int;

```

```

alter table Purchase.Fact_PurchaseOrderDetail
add DateKey int;

```

```

alter table Sales.Fact_SalesOrderDetail
add DateKey int;

```

```

UPDATE Production.Fact_WorkOrderRouting
SET DateKey = (SELECT DateKey FROM Dim_Date WHERE
Dim_Date.Date = Fact_WorkOrderRouting.ModifiedDate)

```

```

UPDATE Purchase.Fact_PurchaseOrderDetail

```

```
SET DateKey = (SELECT DateKey FROM Dim_Date WHERE
Dim_Date.Date = Fact_PurchaseOrderDetail.ModifiedDate)
```

```
UPDATE Sales.Fact_SalesOrderDetail
SET DateKey = (SELECT DateKey FROM Dim_Date WHERE
Dim_Date.Date = Fact_SalesOrderDetail.ModifiedDate)
```

4.6.2. Table Relationship Defined

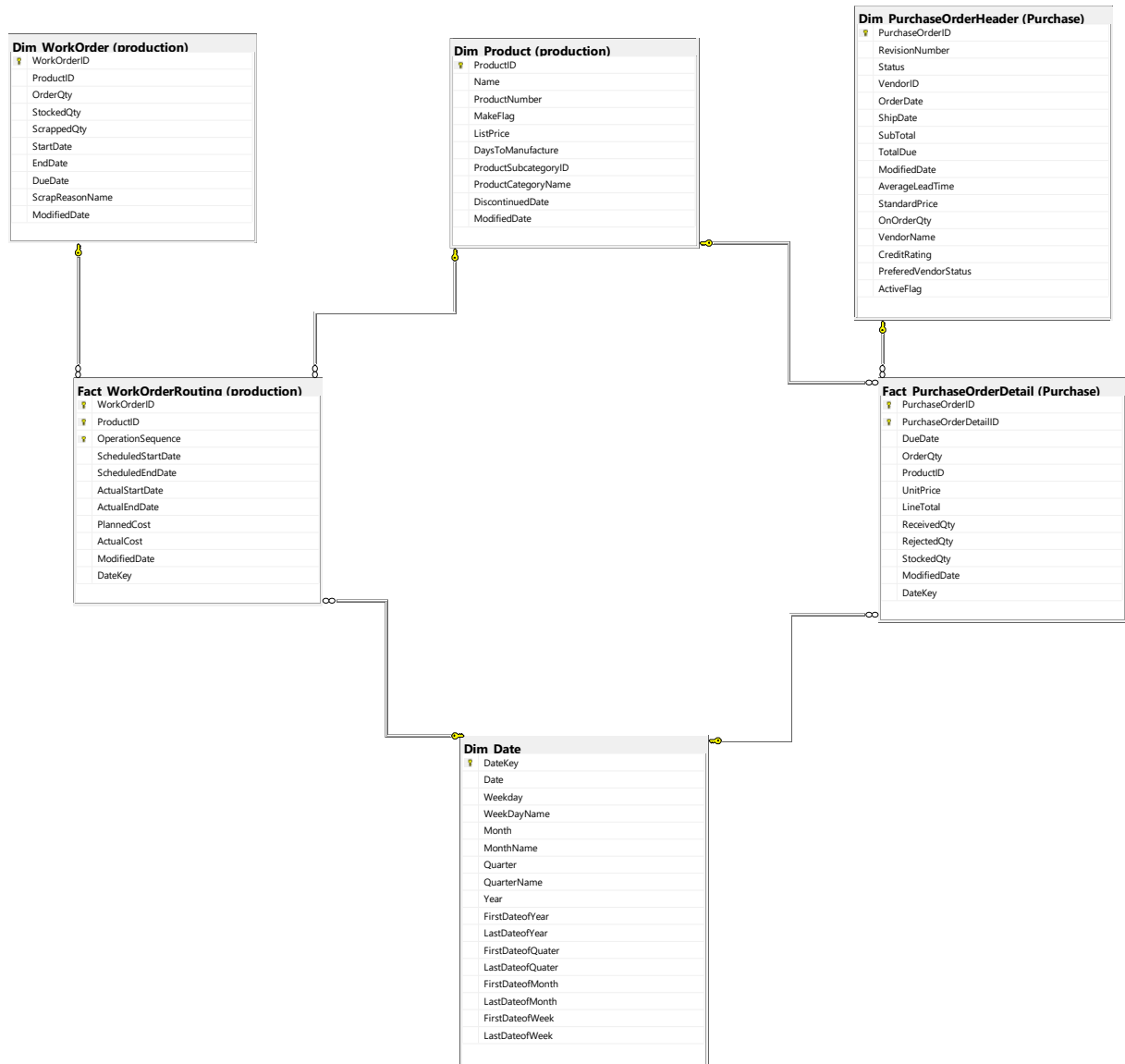


Figure 4.6. Galaxy Schema

We built a data warehouse according to Galaxy Schema. The Galaxy Schema model includes many fact tables that share many dimension tables. Here, we have 2 fact tables, WorkOrderRouting and PurchaseOrderDetail, each table has its own dim table and there are 2 shared dim tables: Product and Date. Galaxy Schema will allow data to be more tightly organized and avoid data redundancy. Additionally, Galaxy reduces

heterogeneity by helping database developers identify, reuse, customize, and advertise related schema components.

4.6.3. Analyzing the correlation between the problem and the data warehouse model

With Galaxy Schema, we can optimize data redundancy by having the Purchase and Production departments share two dimension tables, Time and Product. This way, if there are changes in the shared tables, both departments' data will be updated, avoiding data silos and inconsistencies. To ensure data transparency, our Galaxy Schema builds a centralized data warehouse. This allows timely updates for both the Production and Purchase departments when there are changes in the Product data. This contributes to effective data management processes and ensures data quality, thus maintaining data integrity.

4.7. SCD Use Cases

4.7.1. Describe the Context

Using Slowly Changing Dimensions (SCD) in price management poses an important step towards understanding and responding to frequent market fluctuations and macro factors. The meaning of applying SCD in this case is not only simply updating price information but also bringing many benefits in helping businesses make decisions. First of all, SCD type 2 helps preserve detailed price change history, does not lose old information and creates a data system capable of tracking the development of prices over time. This not only aids in understanding the root causes of change, but also helps predict trends and respond flexibly to fluctuations in the business environment. Additionally, SCD provides easy and flexible data querying capabilities. When it comes to looking at prices at a specific point in time or analyzing price history over a time series, this becomes simple and effective, supporting strategic decision making and effective price management proactive way.

4.7.2. Implementation Process

First, once the team has completed the ETL process from various data sources into the data warehouse system, the next step needs to identify the table and column content that needs to be changed. Figure 4.6 Here the team defines the Dim_Product table in the Production schema with the column that needs to be changed as ListPrice.

ProductID	Name	ProductNumber	MakeFlag	FinishedGoodsFlag	SafetyStockLevel	ReorderPoint	StandardCost	ListPrice	SizeUnitMeasureCode	WeightUnitMeasureCode	DaysToManufacture	ProductLine
1	Adjustable Race	AR-5381	0	0	1000	750	0.00	0.00	NULL	NULL	0	NULL
2	Bearing Ball	BA-8327	0	0	1000	750	0.00	0.00	NULL	NULL	0	NULL
3	BB Ball Bearing	BE-2349	1	0	800	600	0.00	0.00	NULL	NULL	1	NULL
4	Headset Ball Bearings	BE-2908	0	0	800	600	0.00	0.00	NULL	NULL	0	NULL
318	Blade	BL-2036	1	0	800	600	0.00	0.00	NULL	NULL	1	NULL
317	LL Crankarm	CA-5965	0	0	500	375	0.00	0.00	NULL	NULL	0	NULL
318	ML Crankarm	CA-6738	0	0	500	375	0.00	0.00	NULL	NULL	0	NULL
319	HL Crankarm	CA-7457	0	0	500	375	0.00	0.00	NULL	NULL	0	NULL
320	Chaining Bolts	CB-2903	0	0	1000	750	0.00	0.00	NULL	NULL	0	NULL
321	Chaining Nut	CN-6137	0	0	1000	750	0.00	0.00	NULL	NULL	0	NULL
322	Chaining	CR-7833	0	0	1000	750	0.00	0.00	NULL	NULL	0	NULL
323	Crown Race	CR-9981	0	0	1000	750	0.00	0.00	NULL	NULL	0	NULL
324	Chain Stays	CS-2812	1	0	1000	750	0.00	0.00	NULL	NULL	1	NULL
325	Decal 1	DC-8732	0	0	1000	750	0.00	0.00	NULL	NULL	0	NULL
326	Decal 2	DC-8824	0	0	1000	750	0.00	0.00	NULL	NULL	0	NULL
327	Down Tube	DT-2377	1	0	800	600	0.00	0.00	NULL	NULL	1	NULL
328	Mountain End Caps	EC-M092	1	0	1000	750	0.00	0.00	NULL	NULL	1	NULL
329	Road End Caps	EC-R098	1	0	1000	750	0.00	0.00	NULL	NULL	1	NULL
330	Touring End Caps	EC-T209	1	0	1000	750	0.00	0.00	NULL	NULL	1	NULL
331	Fork End	FE-3760	1	0	800	600	0.00	0.00	NULL	NULL	1	NULL
332	Freewheel	FW-2961	0	0	500	375	0.00	0.00	NULL	NULL	0	NULL
341	Flat Washer 1	FW-1000	0	0	1000	750	0.00	0.00	NULL	NULL	0	NULL
342	Flat Washer 6	FW-1200	0	0	1000	750	0.00	0.00	NULL	NULL	0	NULL
343	Flat Washer 2	FW-1400	0	0	1000	750	0.00	0.00	NULL	NULL	0	NULL
344	Flat Washer 9	FW-3400	0	0	1000	750	0.00	0.00	NULL	NULL	0	NULL
345	Flat Washer 4	FW-3800	0	0	1000	750	0.00	0.00	NULL	NULL	0	NULL
346	Flat Washer 3	FW-5160	0	0	1000	750	0.00	0.00	NULL	NULL	0	NULL

Figure 4.7. Dim_Product table

Then, the team will create a new data table named New_Products_Price with the same number of columns and attributes as the Product table in the data warehouse.

Column Name	Data Type	Allow Nulls
ProductID	int	<input type="checkbox"/>
Name	nvarchar(50)	<input type="checkbox"/>
ProductNumber	nvarchar(25)	<input type="checkbox"/>
MakeFlag	bit	<input type="checkbox"/>
FinishedGoodsFlag	bit	<input type="checkbox"/>
SafetyStockLevel	smallint	<input type="checkbox"/>
ReorderPoint	smallint	<input type="checkbox"/>
StandardCost	money	<input type="checkbox"/>
ListPrice	money	<input type="checkbox"/>
SizeUnitMeasureCode	nchar(3)	<input checked="" type="checkbox"/>
WeightUnitMeasureCode	nchar(3)	<input checked="" type="checkbox"/>
DaysToManufacture	int	<input type="checkbox"/>
ProductLine	nchar(2)	<input checked="" type="checkbox"/>
Class	nchar(2)	<input checked="" type="checkbox"/>
ProductSubcategoryID	int	<input type="checkbox"/>
ProductModelID	int	<input type="checkbox"/>
SellStartDate	datetime	<input type="checkbox"/>
SellEndDate	datetime	<input type="checkbox"/>
DiscontinuedDate	datetime	<input checked="" type="checkbox"/>
rowguid	uniqueidentifier	<input type="checkbox"/>
ModifiedDate	datetime	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Figure 4.8. Data description of New_Products_Price table

After creating the data table and changing the price, next the team will use SCD type 2 to update data from the New_Products_Price table to the Product table in the data warehouse. The process is performed as shown below:

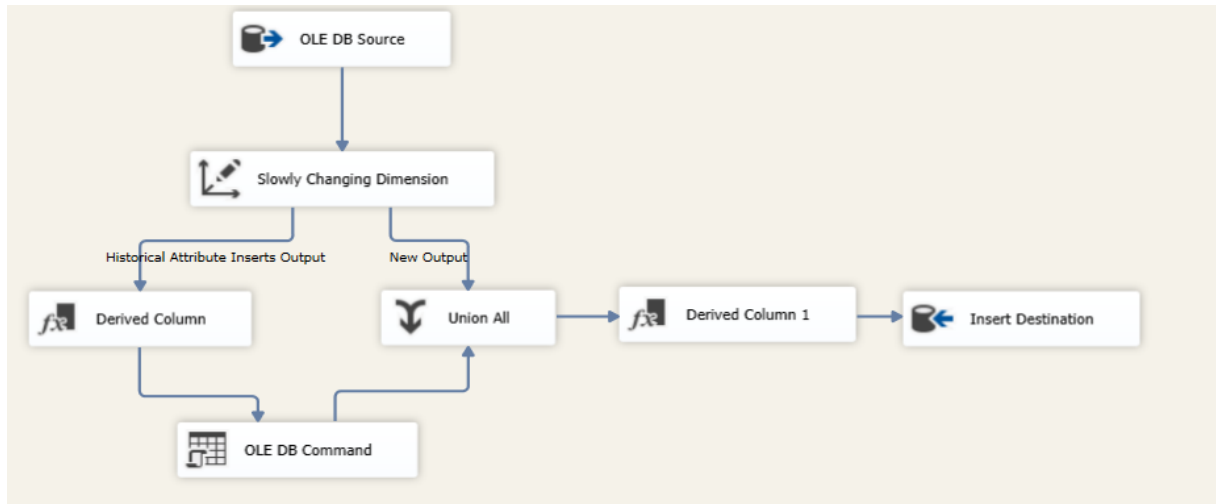


Figure 4.9. SCD process

Next, to change the price, the team executes the update command on SQL:

```

UPDATE [New_Products_Price]
SET [ListPrice] = CASE
    WHEN [ProductID] = 404 THEN 156.23
    WHEN [ProductID] = 402 THEN 233.52
    WHEN [ProductID] = 403 THEN 95.36
WHERE [ProductID] IN (402, 403, 404);
  
```

Algorithm 4.1. SQL query for updating prices of new products

- Table before using SCD type 2 to update:

	ProductID	Name	ProductNumber	ListPrice	Current_Status
1	402	Keyed Washer	KW-4091	0.00	1
2	403	External Lock Washer 3	LE-1000	0.00	1
3	404	External Lock Washer 4	LE-1200	0.00	1

Figure 4.10. Data before using SCD type 2

It can be seen that the current data lines all have Current_Status equal to 1. This means that these data lines are all being used and are unchanged data.

- Table after using SCD type 2 to update:

	ProductID	ProductKey	Name	ProductNumber	ListPrice	Current_Status
1	402	81	Keyed Washer	KW-4091	0.00	0
2	403	82	External Lock Washer 3	LE-1000	0.00	0
3	404	83	External Lock Washer 4	LE-1200	0.00	0
4	402	510	Keyed Washer	KW-4091	233.52	1
5	403	511	External Lock Washer 3	LE-1000	95.36	1
6	404	512	External Lock Washer 4	LE-1200	156.23	1

Figure 4.11. Data after using SCD type 2

After the data set is updated, new data lines with new ListPrice values of ProductID 402, 403, 404 are changed. Then the old data about these tuples will be stored with Current_Status equal to 0. The ListPrice data will be updated with Current_Status equal to 1.

Chapter 5. Analysis with MDX

After bulding Data Warehouse, the project resolve some requirements of use case by using MDX (Multidimensional Expressions) in OLAP Cube to analyze and collect some insights for development strategies of companies.

5.1. Cube Building

In this section, the team will present the schema of the constructed cubes, specifying the type, fact tables, and their relationships. Based on the initial schema in section 4.6.2, the team has created two schemas: Production and Purchasing, both utilizing a star schema design.

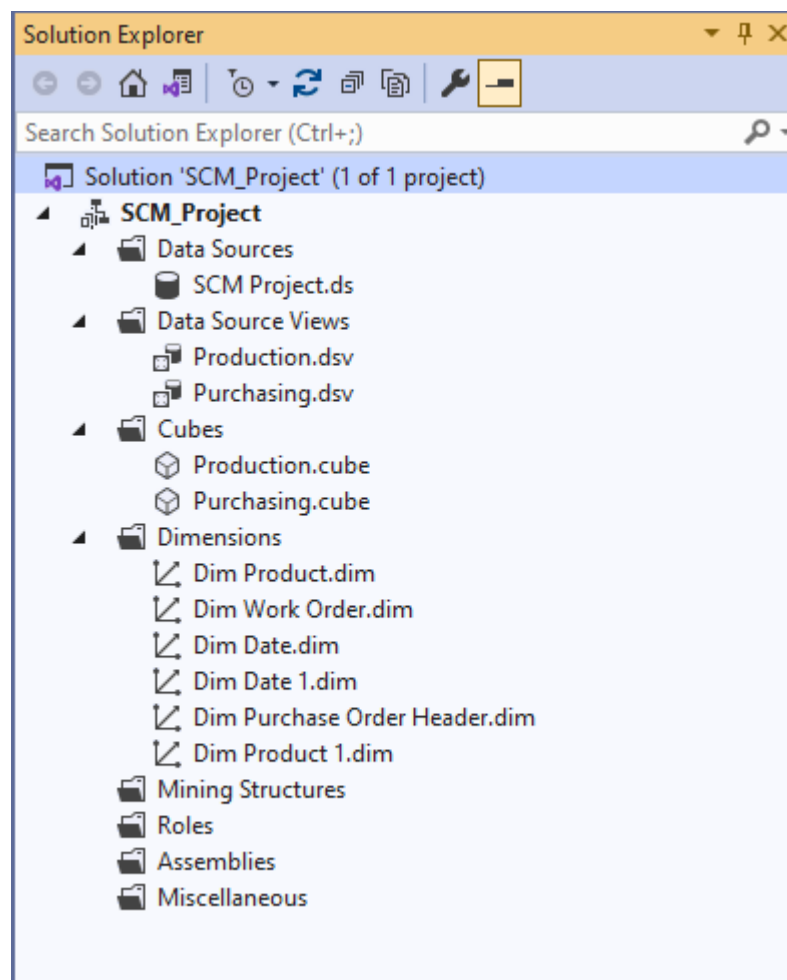


Figure 5.1. Data source - Data Source View - Cube - Dimensions of project

For the Production cube, it follows a star schema structure with the "Fact_WorkOrderRouting" as the fact table containing measures such as "PlannedCost," "ActualCost," "DifferentLeadTime," "DifferentScheduledStartDate," and

"FactWorkOrderRoutingCount". There are three dimensional tables: "Dim_Product," "Dim_WorkOrder," and "Dim_Date".

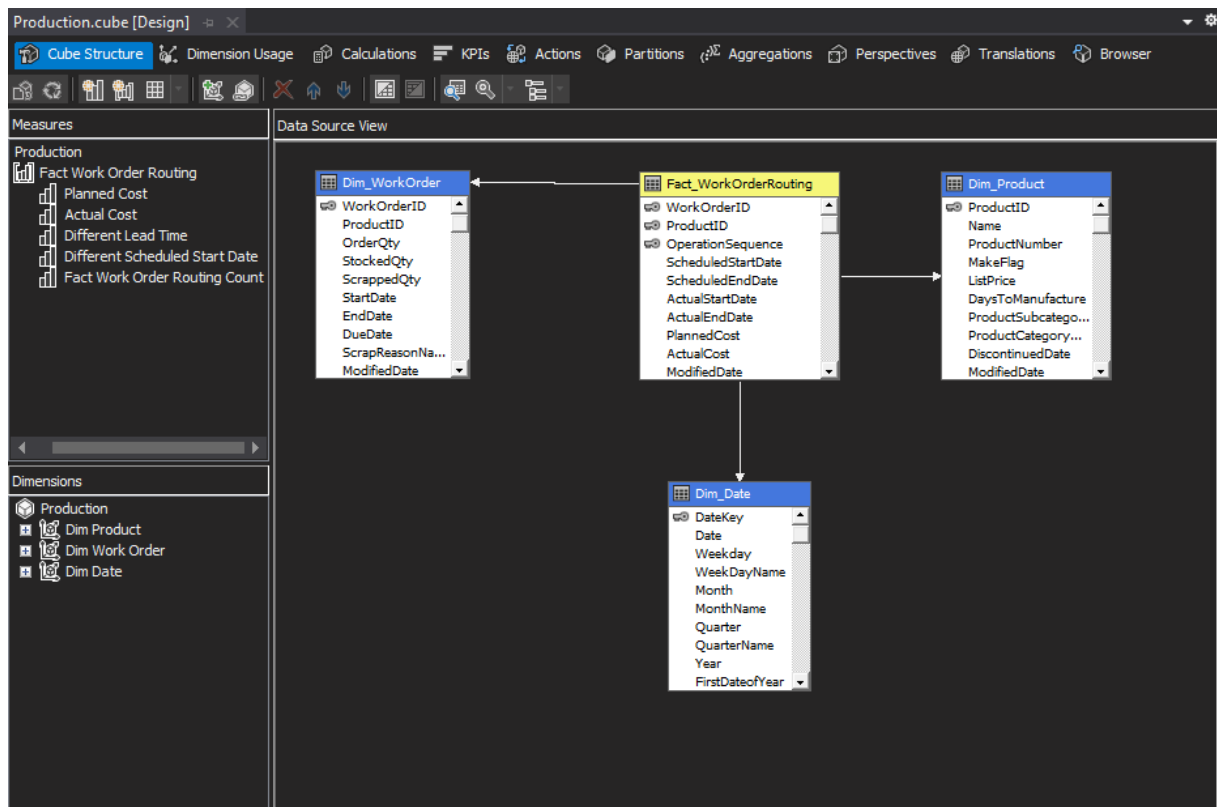


Figure 5.2. Data Source View of Production schema

In the Purchasing cube, it also adopts a star schema, where the fact table is "Fact_PurchaseOrderDetail" with key measures including "OrderQty," "UnitPrice," "LineTotal," "ReceivedQty," "RejectedQty," "StockedQty," and "FactPurchaseOrderDetail." The dimensional tables consist of "Dim_Product," "Dim_Date," and "PurchaseOrderHeader." The repetition of the "Dim_Date" and "Dim_Product" in both cubes is due to the fact that both departments make use of these dimensions.

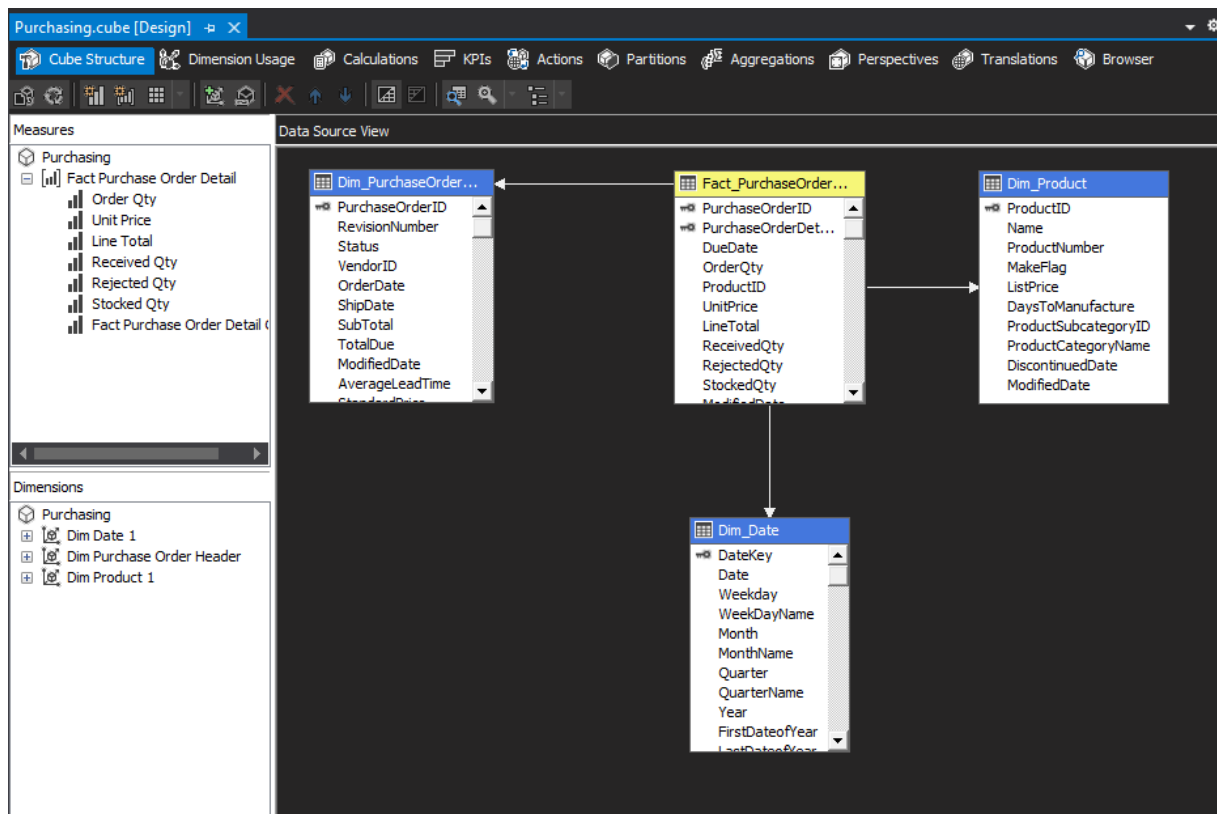


Figure 5.3. Data Source View of Production schema

Additionally, in the Purchasing cube, two calculated measures, namely RejectionRate and StockedRate, have been added, and these are computed in the Calculations tab.

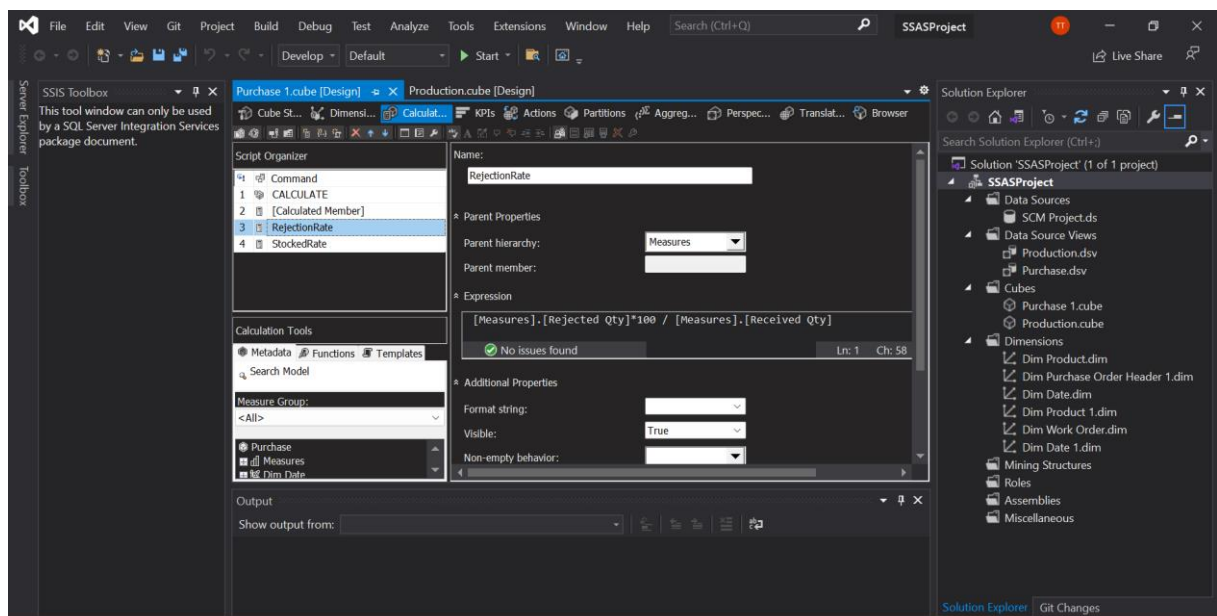


Figure 5.4. Rejection Rate Measure

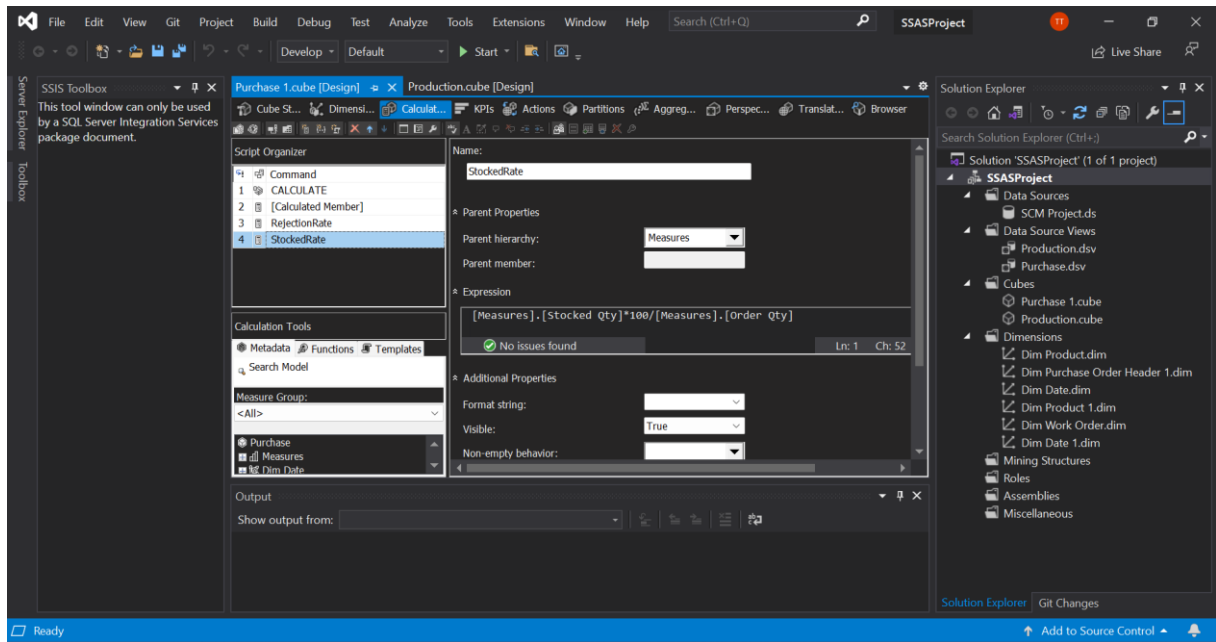


Figure 5.5. Stocked Rate Measure

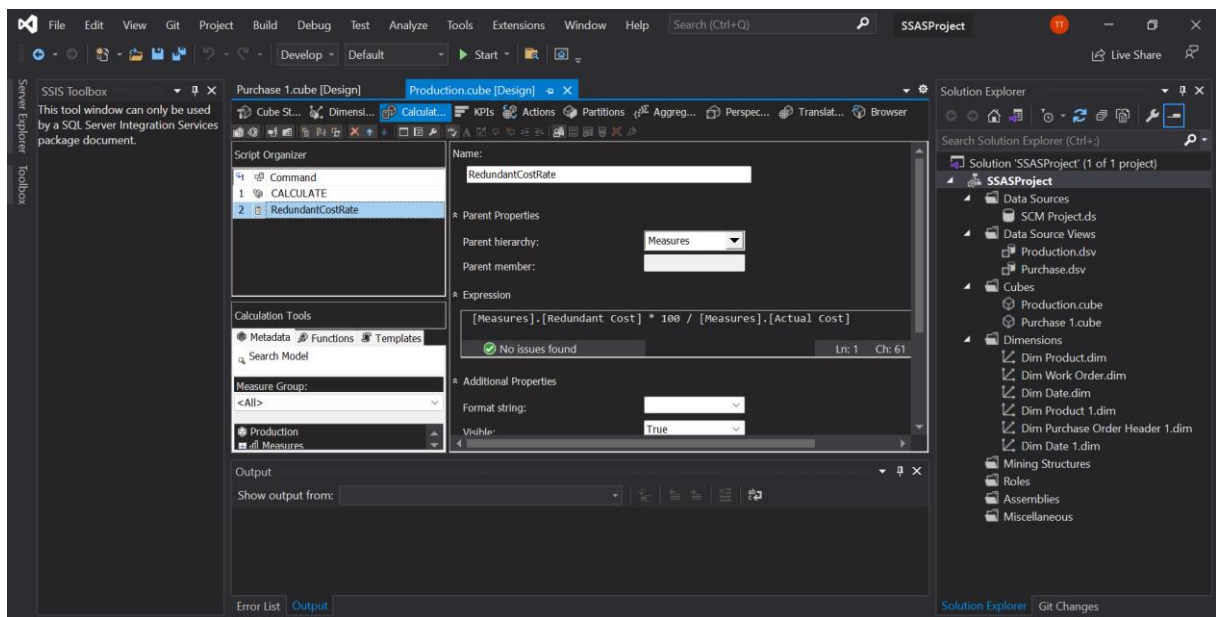


Figure 5.6. Redundant Cost Rate Measure

5.2. OLAP Analysis

5.2.1. Identify Potential Vendors

The team is conducting an analysis of the current status of vendors with the aim of mitigating risks in the supply chain. Specifically, they are focusing on determining the number of active vendors. Additionally, within the total number of linked vendors, they are evaluating whether their credit rating falls within the high range (level 1 or 2).

This determination is crucial for the team to formulate and assess strategies for comparing and evaluating the sustainability of the supply of raw materials for product manufacturing. The team is taking the following steps to make these calculations:

- Active Vendor Count: Identify the total number of vendors that are currently active and operational.
- Credit Rating Analysis: Assess the credit ratings of vendors to determine if they fall within the higher levels, specifically levels 1 or 2.

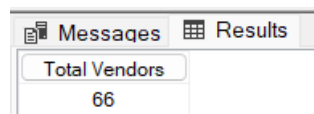
And the code is following these:

```
WITH MEMBER [Measures].[Total Vendors] AS
    DISTINCTCOUNT ([Dim Purchase Order Header].[Vendor
Name].children)
SELECT
    [Measures].[Total Vendors] ON COLUMNS
FROM [Purchasing].[Dim Purchase Order Header].[Vendor
Name].children ON ROWS
WHERE
    ([Dim Purchase Order Header].[Credit Rating].[1]
    , [Dim Purchase Order Header].[Credit Rating].[2]]
    , [Dim Purchase Order Header].[Active Flag].&[TRUE])
```

Algorithm 5.1. MDX query for counting total vendors with specific criteria

The result:

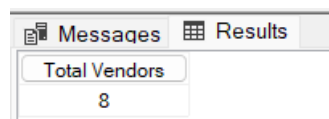
- The results shown for currently active Vendors have a Credit Rating of 1 is 66/82



Total Vendors
66

Figure 5.7. The results of Credit Rating of 1

- The results shown for currently active Vendors have a Credit Rating of 2 is 8/82



Total Vendors
8

Figure 5.8. The results of Credit Rating 2

Based on the analysis results indicating the quantity of active vendors with a high credit rating (74/82) falling within the range of 1-2 , the team has observed that the business currently has suppliers of raw materials that meet both quality and order quantity requirements.

However, to better identify raw material suppliers to be able to evaluate and consider reasonable work or maintain operations, the team continues to dig deeper into analyzing Top 10 highest Rejection Rate suppliers. And the code is following these:

```
WITH SET HIGHESTVENDOR
AS TOPCOUNT ([Dim Purchase Order Header].[Vendor
Name].children, 10,
[Measures].[RejectionRate])
SELECT nonempty(HIGHESTVENDOR) ON ROWS,
nonempty([Measures].[RejectionRate]) ON COLUMNS
FROM [Purchasing]
```

Algorithm 5.2. MDX query for retrieving top vendors by rejection rate

	RejectionRate
International	5.88235294117647
Anderson's Custom Bikes	5.47935263807243
Signature Cycles	5.39895326416307
Crowley Sport	5.33278214138074
Capital Road Cycles	5.33019545846508
Mitchell Sports	5.28998661459193
Premier Sport, Inc.	5.27131782945736
Superior Bicycles	5.24837021393811
Lakewood Bicycle	5.16642547033285
SUPERSALES INC.	5.15227215768071

Figure 5.9. Top 10 highest Rejection Rate suppliers

Thus, it can be seen from the results that there are 10 businesses with the highest rate of raw material transportation that does not meet the business's criteria. Based on these results, businesses can implement monitoring measures, thereby devising plans to find other vendors to replace or quantify and resolve problems with relevant parties to find a direction go optimally.

To evaluate the performance of vendors, we analyze the stocked rate and rejection rate across different Credit Rating levels

```
SELECT
NONEMPTY([Dim Date 1].[Year].Children) on columns,
NONEMPTY([Dim Purchase Order Header].[Credit
Rating].Children) ON rows
FROM [Purchasing]
where [Measures].[StockedRate]
```

Algorithm 5.3. MDX query for analyzing stocked rate by year and credit rating

	2011	2012	2013	2014	2015
1	89.5019786455611	89.6601262024695	94.935777471449	96.7552974829865	99.6987169497688
2	100	91.5394705461483	96.5634301151102	97.4654001729236	(null)
3	100	71.1155997513984	97.2573839662447	97.4950628832762	(null)
4	100	72.5227272727273	97.951871657754	95.5442424242424	(null)
5	(null)	64.1818181818182	98.7090909090909	98.4194214876033	(null)

Figure 5.10. Results of evaluate the performance of vendors

Based on the result table, it is evident that vendors with a Credit Rating of 1 have a lower stocked rate compared to those at Credit Rating 2. The business should pay more attention to these vendors and investigate the reasons behind the lower inventory levels for Credit Rating 1. Similarly, we examine them under the Rejection Rate index.

	2011	2012	2013	2014	2015
1	9.75003764493299	9.39911797133407	4.12614825222275	2.29465491432725	0
2	0	8.26362332695985	2.55012183647374	1.70852616752337	(null)
3	0	27.5561886672998	2.16600417705316	1.48093687637853	(null)
4	0	25.6350501048707	2.04812834224599	3.69230017349657	(null)
5	(null)	35.8181818181818	0.898105217042094	0.855430212712817	(null)

Figure 5.11. Results of Rejection Rate

From this table, we observe that while the Rejection Rate may start slightly higher, the business seems to be attentive and gradually improving its vendors. However, at most stages, Credit Rating 1 is still lower than Credit Rating 2. This indicates that the business needs to revisit its criteria for evaluating its vendors.

Finally, to demonstrate that the business lacks an effective vendor evaluation criterion, we analyze the Rejection Rate across different Credit Ratings based on their average lead times.

```
SELECT
    NONEMPTY([Dim Date 1].[Year].Children) on columns,
    NONEMPTY([Dim Purchase Order Header].[Credit
Rating].Children) ON rows
FROM [Purchasing]
where [Measures].[RejectionRate]
```

Algorithm 5.4. MDX query for analyzing rejection rates by year and credit rating

	120	15	16	17	18	19	25	30	45
1	0	2.62677169656058	4.04996357607576	3.41147034453483	3.39594661393969	3.90500017915745	0	0	0
2	(null)	1.76087637462645	(null)	1.79212436911284	(null)	3.25031543145537	(null)	(null)	(null)
3	(null)	3.44231040813444	3.3704390847248	(null)	1.06951871657754	3.52246902719742	(null)	(null)	(null)
4	(null)	(null)	(null)	4.56653041303154	(null)	(null)	(null)	(null)	(null)
5	(null)	(null)	(null)	(null)	(null)	2.33562365850871	(null)	(null)	(null)

Figure 5.12. Result of Production time

Regarding production time, businesses at Credit Rating 2 have a average lead time equivalent to Credit Rating 1 but exhibit a better Rejection Rate. When comparing all three criteria—average lead time, Rejection Rate, and Stocked Rate—it becomes clear that businesses at Credit Rating 2 outperform those at Credit Rating 1. This is the good point for the business to focus on to optimize the purchasing process.

Next, the team will analyze the total Line Total Purchasing values for different Product Categories to identify the procurement trends in various stages throughout the years, quarters, and months.

Starting at the yearly level, we can observe an overall increasing trend in various product categories, with the highest figure belonging to the "Components" category and the second highest to the "Uncategorized" category. The "Clothing" sector only began ordering in 2015. The presence of uncategorized products may pose challenges in sorting and storing inventory as their nature and trends are unclear.

5.2.2. Demand Trends

Next we analyze the line total and order purchase quantity to understand more about the products demands. First, let's look at the line total in different time frame.

```
SELECT
    NONEMPTY([Dim Date 1].[Year].Children) on COLUMNS,
    NONEMPTY([Dim Product 1].[Product Category Name].Children)
ON ROWS
FROM [Purchasing]
where [Measures].[Line Total]
```

Algorithm 5.5. MDX query for analyzing line total by year and product category

	2011	2012	2013	2014	2015
	210764.779	1387414.287	7017479.448	13693906.8839999	(null)
Accessories	(null)	748232.1	3893285.55	7934549.69999998	95868.315
Clothing	(null)	(null)	(null)	(null)	2068470.6
Components	192371.025	1791246.66	8243728.39499995	16514677.0949998	(null)

Figure 5.13. Result of Products demands

To gain a clearer insight into these trends throughout the year, we analyze the figures by quarter and month.

```
SELECT
    NONEMPTY([Dim Date 1].[Quarter].Children) ON ROWS,
    NONEMPTY([Dim Product 1].[Product Category Name].Children)
ON COLUMNS
FROM [Purchasing]
WHERE [Measures].[Line Total]
```

Algorithm 5.6. MDX query for analyzing line total by quarter and product category

		Accessories	Clothing	Components
1	5430380.10900001	3025233.75	(null)	6619693.81499997
2	6777419.33500001	3910379.55	(null)	7887324.26999995
3	6647326.6545	3803522.265	2068470.6	7918304.09999995
4	3454439.2995	1932800.1	(null)	4316700.99000001

Figure 5.14. Analyzing line total by quarter and product category

First, at the quarterly level, apart from the "Clothing" category, which began ordering in Q3 of 2015, other categories generally show an increasing trend in Q1 and Q2, followed by a decline in Q3 and Q4. Thus, within each quarter, different product categories exhibit either an increasing or decreasing Line Total trend, indicating varying demand for different products but still following a common trend.

```
WITH
    SET [OrderedMonths] AS
        ORDER (
            NONEMPTY([Dim Date 1].[Month].Children),
            Val([Dim Date 1].[Month].CurrentMember.Name)
        )
SELECT
    [OrderedMonths] ON ROWS,
    NonEmpty([Dim Product 1].[Product Category Name].Children)
ON COLUMNS
FROM [Purchasing]
WHERE [Measures].[Line Total]
```

Algorithm 5.7. MDX query for analyzing line total by ordered months and product category

		Accessories	Clothing	Components
3	2045111.25	1094044.875	(null)	2615729.13000001
4	1880345.9335	1193461.5	(null)	2310979.65
5	2475491.403	1528434.6	(null)	2809613.52000001
6	2421581.9985	1188483.45	(null)	2766731.10000001
7	2337394.941	1381362.675	(null)	2921524.20000001
8	2616013.995	1615986.915	2068470.6	2920675.59000001
9	1693917.7185	806172.675	(null)	2076104.31
10	798470.169	568294.65	(null)	1108748.34
11	855902.8695	496239.975	(null)	1025663.415
12	1800066.261	868265.475	(null)	2182289.235

Figure 5.15. Results of analyzing line total by ordered months and product category

From a monthly perspective, all product categories begin a decreasing trend from September and steadily decline until the end of November. December witnesses a recovery before transitioning into a new year. This suggests that procurement of products is concentrated mostly in Q1, Q2, and Q3 of the business year. Users tend to have lower demand in Q4, allowing for strategic production planning to optimize costs.

This analysis provides valuable insights into the procurement patterns, helping the business plan and adjust its strategies based on seasonal demand fluctuations.

Next, we will analyze the demand trends for each quarter in individual years from 2011 to 2014. The code snippet below represents the Line Total data for each quarter in a year by different Product Categories.

```
SELECT {[Dim Date 1].[Hierarchy].[Year].&[2011].&[1]:[Dim Date 1].[Hierarchy].[Year].&[2011].&[4],
[Dim Date 1].[Hierarchy].[Year].&[2012].&[1]:[Dim Date 1].[Hierarchy].[Year].&[2012].&[4],
[Dim Date 1].[Hierarchy].[Year].&[2013].&[1]:[Dim Date 1].[Hierarchy].[Year].&[2013].&[4],
[Dim Date 1].[Hierarchy].[Year].&[2014].&[1]:[Dim Date 1].[Hierarchy].[Year].&[2014].&[4]} on ROWS,
[Dim Product 1].[Product Category Name].Children on COLUMNS
FROM [Purchasing]
WHERE [Measures].[Line Total]
```

Algorithm 5.8. MDX query for analyzing quarterly product category line total for each year

Line Total	Column Labels			
Row Labels	Accessories	Clothing	Components	Grand Total
2011				
2	103895.821			103895.821
4	106868.958		192371.025	299239.983
2012				
1	604547.622	252506.1	770728.77	1627782.492
2	377939.499	219028.425	344527.995	941495.919
3	268997.4105	276697.575	423050.67	968745.6555
4	135929.7555		252939.225	388868.9805
2013				
1	43728.888		87756.9	131485.788
2	505819.4295	434014.35	651478.065	1591311.845
3	3256290.545	1526471.1	3633102.69	8415864.334
4	3211640.586	1932800.1	3871390.74	9015831.426
2014				
1	4782103.599	2772727.65	5761208.145	13316039.39
2	5789764.586	3257336.775	6891318.21	15938419.57
3	3122038.7	1904485.275	3862150.74	8888674.714

Figure 5.16. Results of line total rolling down from 2011 to 2014 by product categories

From the line total data of product categories in each quarter from 2011 to 2014, there is an overall significant growth trend. In specific observations, Accessories and Bikes consistently show positive increases, especially in the second and fourth quarters of 2014. On the contrary, the Components category experiences a notable and consistent decline, particularly in the first and fourth quarters of 2014. This may suggest strategic opportunities for enhancing marketing efforts for Accessories and Bikes, while also warranting a review of the production process and sales strategy for Components.

5.2.3. Manufacturing Analysis

First, we will examine the overall Different Lead Time of Work Orders with defective products. Then, we will identify the reasons contributing to the variance between the planned production time and the actual production time.

```
WITH SET ScrapDiffTime
AS TOPCOUNT ([Dim Work Order].[Scrap Reason Name].children,
6,
[Measures].[Different Lead Time])
SELECT
NONEMPTY([Measures].[Different Lead Time]) ON COLUMNS,
NONEMPTY(ScrapDiffTime) ON ROWS
```

FROM [Production]

Algorithm 5.95.9. MDX query for analyzing reasons with the highest differences in lead time due to scrap

	Different Lead Time
	91364
Trim length too long	200
Drill pattern incorrect	159
Color incorrect	155
Trim length too short	135
Primer process failed	130

Figure 5.17. Results of analyzing reasons with the highest differences in lead time due to scrap

The above results indicate the top 5 reasons causing disruptions in the production process, namely "Trim length too long," "Drill pattern incorrect," "Color incorrect," "Trim length too short," and "Primer process failed".

Following by the reason for product scrapped the most, we want more details about cost in manufacturing department. Multiple source code below illustrates cost in different time frame by Product Category.

```
SELECT
    NONEMPTY([Dim Product].[Product Category Name].Children) ON
COLUMNS,
    NONEMPTY([Dim Date].[Year].Children) ON ROWS
FROM [Production]
WHERE [Measures].[Actual Cost]
```

Algorithm 5.10. MDX query for analyzing actual cost by product category and year

		Bikes	Components
2011	33356.75	46452	232536
2012	86178.75	122059	596579.5
2013	163868.25	247303	1028710.25
2014	106415.75	197568	626942.25

Figure 5.18. Analyzing actual cost by product category and year

- For Bikes:
 - From 2011 to 2012, there was an increase of approximately 158.35%.
 - From 2012 to 2013, the increase was around 90.15%.
 - However, from 2013 to 2014, there was a decrease of about 35.06%.
 - Actual cost in 2013 is highest: 247303, Actual cost in 2011 is lowest: 46452
- For Components:

- From 2011 to 2012, there was an increase of roughly 162.76%.
 - From 2012 to 2013, the increase was about 102.61%.
 - From 2013 to 2014, there was a decrease of 20.11%.
 - Actual cost in 2013 is highest: 1028710.25, Actual cost in 2011 is lowest: 232536
- Compound Annual Growth Rate (CAGR):
- For Bikes, the CAGR over the period from 2011 to 2014 is approximately 47.21%.
 - For Components, the CAGR is higher at approximately 62.02%.

These insights suggest that both product categories experienced significant growth in the initial two years, with a notable reduction in the last year. The CAGR indicates that despite the decrease in 2014, both categories grew at a robust rate over the four-year period, with Components growing at a higher rate than Bikes.

```
SELECT
    NONEMPTY([Dim Product].[Product Category Name].Children) ON
COLUMNS,
    NONEMPTY([Dim Date].[Quarter].Children) ON ROWS
FROM [Production]
WHERE [Measures].[Actual Cost]
```

Algorithm 5.11. MDX query for analyzing actual cost by product category and quarter

		Bikes	Components
1	99127	158368	627540.75
2	102214	177037	665058.75
3	90209	127351	576084
4	98269.5	150626	616084.5

Figure 5.19. Analyzing actual cost by product category and quarter

- For Bikes:
 - There was a 3.11% increase from Q1 to Q2.
 - A decrease of 11.74% from Q2 to Q3.
 - An increase of 8.94% from Q3 to Q4.
 - Actual cost in Q2 is highest: 177037, actual cost in Q3 is lowest: 127351
- For Components:
 - An 11.79% increase from Q1 to Q2.
 - A substantial decrease of 28.07% from Q2 to Q3.

- An increase of 18.28% from Q3 to Q4.
- Actual cost in Q2 is highest: 665058.75, actual cost in Q3 is lowest: 576084

From these figures, it's evident that there are fluctuations in the quantities of Bikes and Components, as well as in Costs, with increases in Q2 and Q4 and decreases in Q3. The Components category seems to have more significant fluctuations compared to Bikes, which could indicate varying demand or inventory changes.

```
WITH
    SET [OrderedMonths] AS
        ORDER (
            NONEMPTY([Dim Date].[Month].Children),
            Val([Dim Date].[Month].CurrentMember.Name)
        )

SELECT
    NONEMPTY([Dim Product].[Product Category Name].Children) ON
COLUMNS,
    NONEMPTY([OrderedMonths]) ON ROWS
FROM [Production]
WHERE [Measures].[Actual Cost]
```

Algorithm 5.12. MDX query for analyzing actual cost by product category and ordered months

		Bikes	Components
1	35414.75	54831	223011.75
2	29755.25	48167	191809.75
3	33957	55370	212719.25
4	33785.5	60613	220597.75
5	35500.5	61985	234001.5
6	32928	54439	210459.5
7	28812	39886	185791.5
8	32242	44884	203833.75
9	29155	42581	186458.75
10	33013.75	46256	205147
11	32156.25	49784	201820

Figure 5.20. Analyzing actual cost by product category and ordered

- For Bikes:
 - The average monthly production cost for Bikes is approximately \$32,429.
 - There is a standard deviation of around \$2,328, indicating a relatively moderate fluctuation in costs from month to month.
 - The minimum monthly cost observed for Bikes is \$28,812, while the maximum is \$35,500.50.

- For Components:
 - The average monthly production cost for Components is significantly higher at approximately \$50,800.
 - There is a larger standard deviation of around \$7,225 for Components, suggesting more variability in the monthly costs compared to Bikes.
 - The minimum monthly cost for Components is \$39,886, and the maximum is \$61,985.
- Month-over-Month Percentage Change:
 - On average, the month-over-month percentage change in costs for Bikes is a decrease of about 0.41%, indicating a slight downward trend across the months.
 - The Components' costs show an average month-over-month percentage change of a decrease of about 0.07%, which is quite stable and indicates less volatility in comparison to Bikes.

Understanding the reasons behind these trends would require further context and analysis, such as market demand, production efficiency, and cost of materials

```
SELECT
  NONEMPTY([Dim Product].[Product Category Name].Children) ON
COLUMNS,
  NONEMPTY([Dim Date].[Week Day Name].Children) ON ROWS
FROM [Production]
WHERE [Measures].[Actual Cost]
```

Algorithm 5.13. MDX query for analyzing actual cost by product category and week day name

		Bikes	Components
Friday	57538.25	87759	364276
Monday	54194	86338	346994.25
Saturday	52822	84721	336430.5
Sunday	57795.5	91777	367586.75
Thursday	56938	91287	365214.5
Tuesday	57366.75	91042	368169
Wednesday	53165	80458	336097

Figure 5.21. Analyzing actual cost by product category and week day name

- For Bikes:
 - The average daily production cost for Bikes is about \$55,688.

- There's a relatively small standard deviation of around \$2,200, suggesting that daily costs do not vary widely.
- The maximum daily cost for Bikes is on Sunday at \$57,795.50, and the minimum is on Saturday at \$52,822.
- For Components:
 - The average daily production cost for Components is significantly higher at approximately \$87,626.
 - The standard deviation is about \$4,159, indicating somewhat greater variability in daily costs compared to Bikes.
 - The highest cost for Components is also on Sunday at \$91,777, while the lowest is on Wednesday at \$80,458.
- Percentage of Total Weekly Costs:
 - Sunday accounts for the highest percentage of weekly costs for both Bikes and Components, with approximately 14.83% and 14.96% respectively.
 - Wednesday has the lowest percentage of weekly costs for Components at 13.12%, while Saturday has the lowest for Bikes at 13.55%.

These insights suggest that production costs are higher towards the end of the week, particularly on Sunday. This could be due to a variety of factors, such as overtime pay for workers on weekends, rush orders that need to be completed before the next week, or perhaps Sunday is the end of the production cycle when more products are finished. Further investigation into the operational aspects of the production would be needed to understand the reasons behind these cost patterns.

The code and image below represents top five products with highest Redundant Cost Rate. The business should take care of these products to optimize the manufacturing cost.

```
WITH SET Top5ProductHighRedundateCostRate as
topcount([Dim Product].[Name].Children, 10,
[Measures].[RedundantCostRate])

SELECT Top5ProductHighRedundateCostRate ON ROWS,
       [Measures].[RedundantCostRate] ON COLUMNS
FROM [Production]
```

Algorithm 5.14. MDX query for analyzing top products by redundant cost rate

	RedundantCostRate
Mountain-300 Black, 38	31.25
Mountain-300 Black, 40	31.25
Mountain-300 Black, 44	31.25
Mountain-300 Black, 48	31.25
Road-450 Red, 44	31.25
Road-450 Red, 48	31.25
Road-450 Red, 52	31.25
Road-450 Red, 58	31.25
Road-450 Red, 60	31.25
ML Mountain Handlebars	31.2496535796767

Figure 5.22. Top 10 products has highest redundant cost rate.

Here are the top 10 products with the highest redundant cost rates, indicating that producing these products would lead to a significant waste of the company's resources. Based on the calculated results, it is evident that the majority of these top 10 products have a redundancy rate of approximately 31.25%. This figure signifies that 31.25% of the total production cost comes from discrepancies between planned and actual production. The company should focus on improving production planning for these specific products to optimize profitability.

Chapter 6. Discussion and Future Works

In this chapter, the authors have a comprehensive view about project and make assessments through visualized results in reports. Therefore, the project have some future direction for more in-depth research to develop strengths and improve limitations.

6.1. Discussion

In this section, we will examine the correlation between stated objectives and the outcomes of the analysis. Firstly, in the production section, the team has analyzed four key metrics to evaluate the efficiency of production operations. There are two main aspects here: firstly, regarding defective products, the team investigated the hours of deviation in production compared to the planned schedule and how the production unfolded. Secondly, the team examined the ratio of products that incur excessive redundant costs due to inefficient production planning. The third aspect involved observing cost trends across different product categories over various time frames. In the Production department, the team successfully achieved the initial objectives related to identifying costs and delays in planning and actual production. Furthermore, the team analyzed factors that impede the production speed of the enterprise, such as reasons for product defects. They also evaluated how these defective products impact production speed through Different Lead Time. Therefore, concerning the question of factors

influencing costs and production planning for products, this report has adequately addressed the issues.

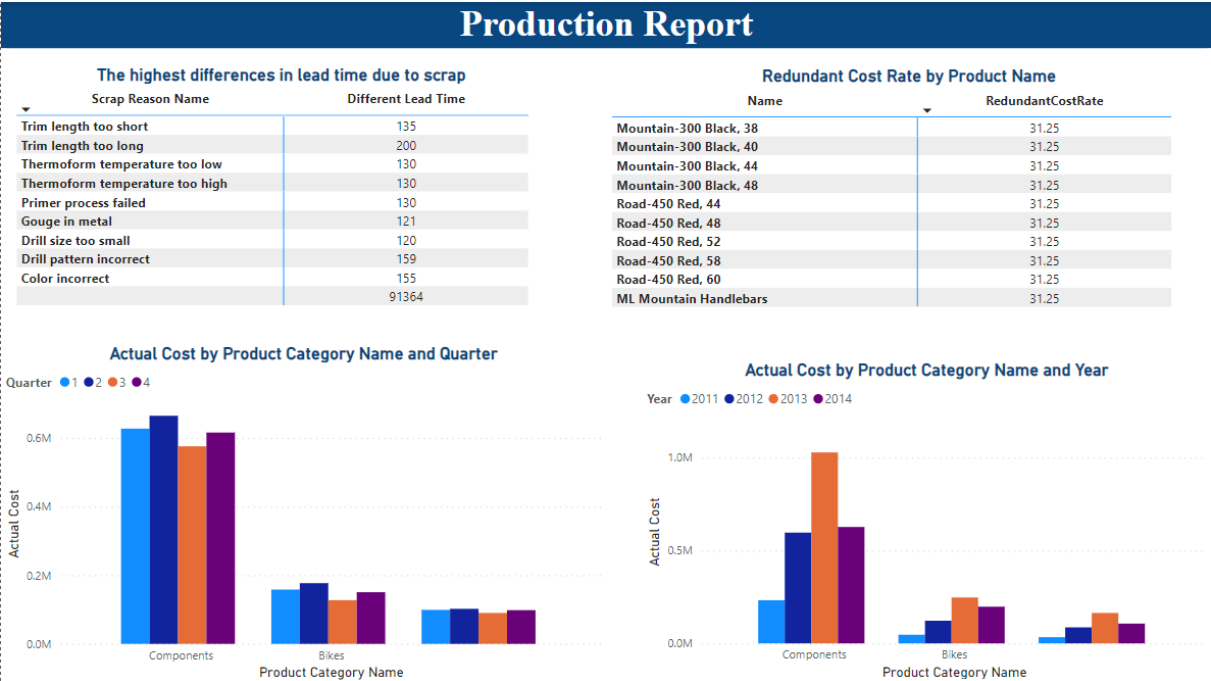


Figure 6.1. Production dashboard for reporting

Next is Purchasing, in this section, our team has analyzed values related to Rejection Rate, Stocked Rate of vendors across different credit rating levels to assess the operational performance of vendors and determine if the business is accurately evaluating the credibility levels of vendors. Additionally, to ensure a comprehensive assessment of vendor quality, our team also evaluates the Rejection Rate of vendors based on their Average Lead Time. We track the five vendors with the lowest quality based on Rejection Rate and Stocked Rate to formulate appropriate strategies. Lastly, a crucial factor is the quantity of orders placed by the business, indicating the demand trends over the years, quarters, and months, as analyzed in section (5.2.2).

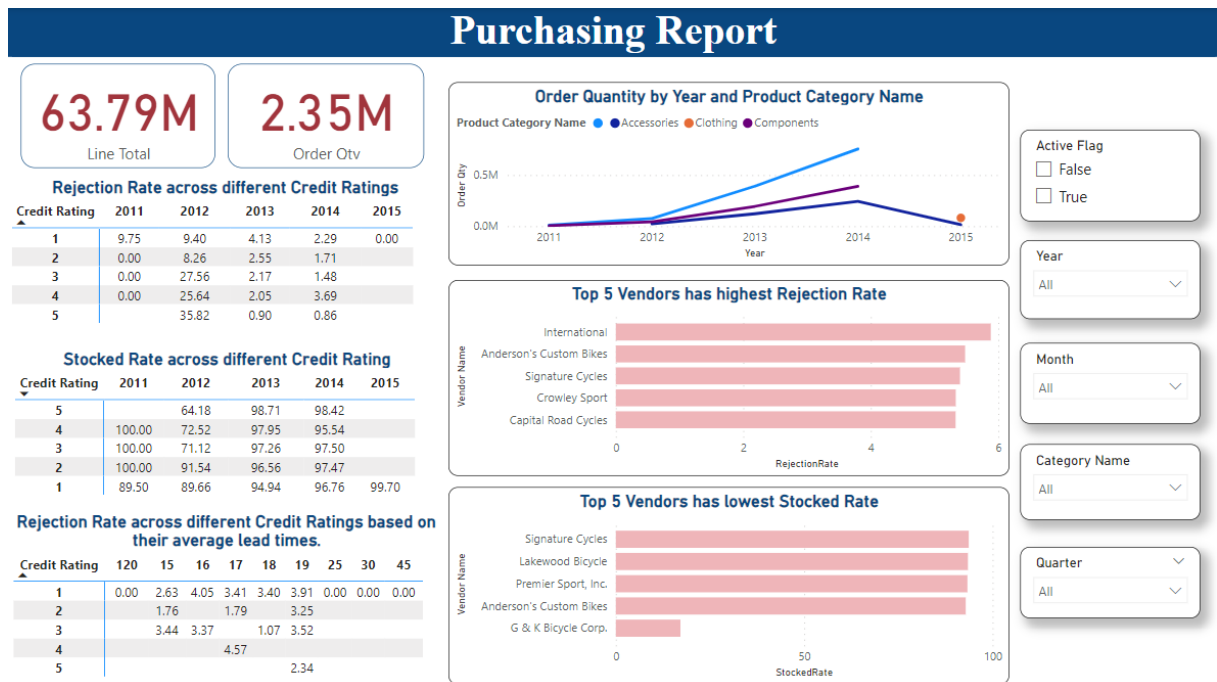


Figure 6.2. Purchasing dashboard for reporting

6.2. Solving Usecase

After having traversed the fundamental content analyses outlined in the problem statement, the next step is to examine how these analyses have contributed to the initial posed problem.”What is the percentage of rejection and stocking ratios to inform vendors quality?”

With the above analyses, the team has identified high and low-quality vendor groups by combining the Rejection Rate and Stocked Rate indices, considering factors such as vendor reliability, Average Lead Time, and time-related aspects. This makes the selection of high-quality vendors easier, allowing for the retention of reliable vendors and the elimination of low-quality ones. Secondly, the team analyzed trends in increasing and decreasing orders to vendors. Analyzing these cycles enables businesses to be more proactive in inventory planning, optimizing production plans to minimize costs. Finally, the analysis identified products that incur significant costs in the production process, as well as factors contributing to production delays from scrapped products. These results help businesses pinpoint issues in their production processes, reduce production time, minimize unnecessary costs from inaccurate planning, and improve the delivery time of orders to customers.

6.3. Future Works

6.3.1. Limitations of Current Research:

The present study, centered on the manufacturing process at AdventureWorks, encounters limitations primarily in its scope and data usage. Our analysis is confined to two key departments: Purchasing and Production. This focused approach, while insightful, may not encompass the full spectrum of variables influencing the manufacturing environment. Additionally, the reliance on specific datasets from AdventureWorks poses a challenge in generalizing our findings to other manufacturing contexts or industries, where different dynamics might prevail.

6.3.2. Future Research Directions:

Expanding Departmental Focus: To obtain a more comprehensive understanding of manufacturing operations, future research should extend beyond the realms of Purchasing and Production. Including additional departments such as Logistics and Quality Control could unveil more intricate interactions and dependencies within the manufacturing process.

Diversifying Data Sources: Broadening the data sources to include external market dynamics, customer feedback, and supplier data could significantly enhance the depth and breadth of our analysis. This diversification would not only provide a richer dataset but also improve the applicability and robustness of our conclusions across different manufacturing scenarios.

Enhancing Data Integration: A critical area for future exploration is the effective integration of data from disparate sources. This includes overcoming challenges like data silos and inconsistent data formats, which can impede the seamless flow and analysis of information across different manufacturing departments.

Real-time Data Processing and Analytics: Investigating the potential of real-time data processing and analytics could revolutionize decision-making processes in manufacturing. By enabling immediate insights into production dynamics, such approaches could facilitate more proactive and adaptive management strategies.

References

- (1) Banaitiene, N., & Banaitis, A. (2012). Risk management in construction projects. *Risk management-current issues and challenges*, 429-448.
- (2) Tupa, J., Simota, J., & Steiner, F. (2017). Aspects of risk management implementation for Industry 4.0. *Procedia manufacturing*, 11, 1223-1230.
- (3) Pritchard, C. L., & PMP, P. R. (2014). Risk management: concepts and guidance. CRC Press.
- (4) Berg, H. P. (2010). Risk management: procedures, methods and experiences. *Reliability: Theory & Applications*, 5(2 (17)), 79-95.
- (5) Houghton, J. R., Rowe, G., Frewer, L. J., Van Kleef, E., Chryssochoidis, G., Kehagia, O., ... & Strada, A. (2008). The quality of food risk management in Europe: Perspectives and priorities. *Food Policy*, 33(1), 13-26.
- (6) R. Kimball and M. Ross, "The data warehousing toolkit", New York :John Wiley&Sons, 1996.
- (7) R. Kimball and M. Ross, "The data warehouse toolkit :The definitive guide to dimensional modeling", John Wiley & Sons, 2013.
- (8) Yessad, L., & Labiod, A. (2016, November). Comparative study of data warehouses modeling approaches: Inmon, Kimball and Data Vault. In *2016 International Conference on System Reliability and Science (ICSRS)* (pp. 95-99). IEEE.
- (9) Hecht, J. (2019, October 22). What Are the Three Types of Schema in a Data Warehouse? The Hecht Group. <https://www.hechtgroup.com/what-are-the-three-types-of-schema-in-a-data-warehouse/>
- (10) StreamSets. (2021, March 10). Schemas in Data Warehouses: Star, Galaxy, Snowflake. StreamSets. <https://streamsets.com/blog/schemas-data-warehouses-star-galaxy-snowflake/>
- (11) Software Testing Help. (2021, August 27). Data Warehouse Modeling: Star Schema, Snowflake Schema, and Fact Constellation. Software Testing Help. <https://www.softwaretestinghelp.com/data-warehouse-modeling-star-schema-snowflake-schema/>

- (12) EDUCBA. (n.d.). Data Warehouse Schema. EDUCBA.
<https://www.educba.com/data-warehouse-schema/>
- (13) Nwokeji, J. C., Aqlan, F., Anugu, A., & Olagunju, A. (2018, July). Big Data ETL Implementation Approaches: A Systematic Literature Review (P). In *SEKE* (pp. 714-713).
- (14) Nwokeji, J. C., & Matovu, R. (2021). A systematic literature review on big data extraction, transformation and loading (etl). In *Intelligent Computing: Proceedings of the 2021 Computing Conference, Volume 2* (pp. 308-324). Springer International Publishing.
- (15) Sreemathy, J., Durai, K. N., Priya, E. L., Deebika, R., Suganthi, K., & Aisshwarya, P. T. (2021, March). Data integration and ETL: a theoretical perspective. In *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)* (Vol. 1, pp. 1655-1660). IEEE.
- (16) Nguyen, T. M., Tjoa, A. M., Nemec, J., & Windisch, M. (2007). An approach towards an event-fed solution for slowly changing dimensions in data warehouses with a detailed case study. *Data & Knowledge Engineering*, 63(1), 26-43.
- (17) Loyola, R. C., Sepulveda, A. U., & Hernandez, M. W. (2015, November). Optimization slowly changing dimensions of a data warehouse using object-relational. In *2015 34th International Conference of the Chilean Computer Science Society (SCCC)* (pp. 1-6). IEEE.
- (18) Sepulveda, A. U., Loyola, R. C., & Hernandez, M. W. (2015, November). Performance comparison slowly changing dimensions using model relational and object-relational. In *2015 34th International Conference of the Chilean Computer Science Society (SCCC)* (pp. 1-6). IEEE.
- (19) Nguyen, T. M., Tjoa, A. M., Nemec, J., & Windisch, M. (2007). An approach towards an event-fed solution for slowly changing dimensions in data warehouses with a detailed case study. *Data & Knowledge Engineering*, 63(1), 26-43.
- (20) Loyola, R. C., Sepulveda, A. U., & Hernandez, M. W. (2015, November). Optimization slowly changing dimensions of a data warehouse using object-relational. In *2015 34th International Conference of the Chilean Computer Science Society (SCCC)* (pp. 1-6). IEEE.

- (21) Sepulveda, A. U., Loyola, R. C., & Hernandez, M. W. (2015, November). Performance comparison slowly changing dimensions using model relational and object-relational. In *2015 34th International Conference of the Chilean Computer Science Society (SCCC)* (pp. 1-6). IEEE.
- (22) Legodi, I., & Barry, M. L. (2010). The current challenges and status of risk management in enterprise data warehouse projects in South Africa. *PICMET 2010 Technology Management for Global Economic Growth*, 1-5
- (23) Meuwissen, M. P., Huirne, R. B. M., & Hardaker, J. B. (2001). Risk and risk management: an empirical analysis of Dutch livestock farmers. *Livestock production science*, 69(1), 43-53
- (24) Alshawhi, S., Saez-Pujol, I., & Irani, Z. (2003). Data warehousing in decision support for pharmaceutical R&D supply chain. *International Journal of Information Management*, 23(3), 259-268.
- (25) Yu, X. (2021). The Application of Data Warehouse in Teaching Management in Colleges and Universities. In *Journal of Physics: Conference Series* (Vol. 1738, No. 1, p. 012090). IOP Publishing.
- (26) Nielsen, A. C. (2011, December). Data warehouse for assessing animal health, welfare, risk management and–communication. In *Acta Veterinaria Scandinavica* (Vol. 53, No. 1, pp. 1-4). BioMed Central.
- (27) Chunying, Z., & Weiqing, G. (2009, July). Risk management based on data warehouse of securities companies. In *2009 4th International Conference on Computer Science & Education* (pp. 819-822). IEEE.
- (28) Nwankwo, W., & Famuyide, O. (2016). A model for implementing security and risk management data warehouse for scanning operations in Nigeria. *International Journal of Engineering Research and Technology*, India.
- (29) Chaudhuri, S., & Dayal, U. (1997). An overview of data warehousing and OLAP technology. *ACM Sigmod record*, 26(1), 65-74.
- (30) McGrath, H., Stefanakis, E., & Nastev, M. (2014). Development of a data warehouse for riverine and coastal flood risk management. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40, 41-48.