



Wide World Importers

A Journey Through Data Integration and Insight

2024

TABLE OF CONTENTS

1	Introduction to WWI	1
1.1	Comprehensive Business Analysis Approach for WWI	1
1.1.1	Business Context and Objectives.....	2
1.1.2	Analysis of Stakeholders	2
1.1.3	The Arrangement of an Organisation's Components and their Relationships.....	2
1.1.4	SWOT Analysis.....	2
1.1.5	Analysis of the Process.....	3
1.1.6	Analysis of the Market.....	3
1.1.7	Evaluation of Financial Data	3
1.1.8	Evaluation Criteria and Key Performance Indicators	4
1.1.9	Evaluation of Hazards and Their Likelihood and Impact.....	4
1.1.10	Strategic Planning	4
2	Business Analysis	5
2.1	Business Context and Objectives	5
2.1.1	Business Model.....	7
2.1.2	Key Business Objectives	8
2.2	Stakeholder Analysis.....	8
2.2.1	Stakeholder Groups and Needs.....	9
2.2.2	Analysis of Stakeholder Relationship.....	10
2.3	Organisational Chart	11
2.3.1	Executive Leadership:	11
2.3.2	Operations Department:.....	12
2.3.3	Finance Department:.....	13
2.3.4	Information Technology (IT) Department:.....	13
2.3.5	Marketing and Sales Department:	14
2.3.6	Human Resources (HR) Department:	14
2.3.7	Procurement Department:	15
2.4	SWOT Analysis	16
2.4.1	Strengths	16
2.4.2	Weaknesses.....	17
2.4.3	Opportunities	17
2.4.4	Threats	18
2.5	Process Flow in WWI	18
2.5.1	Procurement Process:.....	19
2.5.2	Sales Process:.....	20
2.5.3	Order Fulfillment Process:.....	20
2.5.4	Customer Service Process:	21
2.5.5	Identifying Bottlenecks.....	21
2.5.5.1	Procurement Bottlenecks:.....	21
2.5.5.2	Sales Bottlenecks:.....	21
2.5.5.3	Order Fulfillment Bottlenecks:	21
2.5.5.4	Customer Service Bottlenecks:	22
2.5.6	Optimisation Opportunities	22
2.5.6.1	Procurement Optimisation:.....	22

2.5.6.2 Sales Optimisation:.....	22
2.5.6.3 Customer Service Optimisation:.....	23
2.6 Market Analysis for WWI.....	23
2.6.1 Market Segmentation.....	23
2.6.2 Competitive Analysis	23
2.6.3 Customer Analysis.....	24
2.6.4 Financial Analysis	24
2.6.4.1 Revenue Streams	24
2.6.4.2 Cost Structure	25
2.6.4.3 Profitability Analysis.....	25
2.7 User Journey Analysis for WWI.....	27
2.7.1 Retail Customer Journey.....	27
2.7.2 Wholesale Buyer Journey	28
2.7.3 Supplier Journey	30
3 Purchasing Functionality Analysis for WWI	32
3.1 Interaction with Other Organisational Users	33
3.2 Business Process	33
3.3 Critical Information for Purchasing Functionality.....	33
3.4 Purchasing Scope of Organisational Chart	35
3.5 Detailed Interaction from the Purchasing Department	37
3.5.1 Consideration Stage	37
3.5.2 Purchase Stage	37
3.5.3 Delivery Stage	37
3.5.4 Post-Purchase Stage.....	37
3.6 Risk Analysis	38
3.6.1 Operational Risks:	38
3.6.2 Financial Risks:	38
3.6.3 Market Risks:.....	38
3.6.4 Strategic Risks:	38
3.6.5 Mitigation Strategies.....	39
3.6.5.1 Operational Risk Mitigation:.....	39
3.6.5.2 Financial Risk Mitigation:.....	39
3.6.5.3 Market Risk Mitigation:	39
3.6.5.4 Strategic Risk Mitigation:.....	39
3.7 Strategic Planning	40
3.7.1 Market Expansion:.....	40
3.7.2 Product Diversification:.....	40
3.7.3 Operational Efficiency:.....	40
3.7.4 Customer Engagement:.....	41
3.7.5 Sustainability Initiatives:	41
3.8 Implementation Plan	41
3.9 Continuous Monitoring and Improvement.....	42
3.9.1 Regular Reviews.....	42
3.9.2 Feedback Loop.....	43
3.9.3 Adaptability	43
3.10 Conclusion	44

3.10.1	Strategic Context	44
3.10.2	Stakeholder Analysis	45
3.10.3	Organisational Structure	45
3.10.4	SWOT Analysis	45
3.10.5	Process Flow and Optimisation	45
3.10.6	Market and Financial Analysis	45
3.10.7	Risk Management and Strategic Planning	46
3.10.8	Continuous Improvement	46
4	KPIs and Data-Driven Infrastructure for the Purchasing	47
4.1	KPIs and Reporting Requirements for the Purchasing Department	47
4.2	KPI Dashboard.....	48
4.3	Purchasing Dashboard Overview.....	49
4.3.1	KPIs	49
4.3.2	Key Metrics.	50
4.3.3	Operational Insights:.....	50
4.4	Predictive Model	54
4.4.1	Importance of Forecasting for Warehouse Management.....	54
4.4.2	Data Analysis.....	54
4.4.3	Machine Learning for Predicting Purchasing Amount	55
4.5	Data Architecture for Purchasing KPIs.....	58
4.5.1	Entities and Tables to Use:	58
4.5.2	Entity-Relationship Diagram:	58
4.5.3	Understanding the ERD to the OLTP Database:	60
4.5.3.1	Key Tables Identified from the WWI OLTP Database:...60	
4.5.3.2	Relating ERD to Table Identification:.....	61
4.6	Enhancing Business Intelligence	62
4.6.1	OLTP Systems and Their Limitations for Reporting	62
4.6.1.1	Characteristics of OLTP Systems:	62
4.6.1.2	Limitations of OLTP Systems for Reporting	63
4.6.2	Transition to OLAP Systems for Enhanced Analytical Capabilities	63
4.6.2.1	Key Features of OLAP Systems:.....	64
4.6.2.2	The Role of Star Schema in OLAP	64
4.6.3	Advantages of Using a Star Schema.....	65
4.6.3.1	Simplified Queries.....	65
4.6.3.2	Enhanced Performance.....	65
4.6.3.3	Scalability	65
4.6.3.4	Intuitive Design	65
4.6.3.5	Effective Data Segmentation	66
4.6.3.6	Conclusion	66
4.6.4	Star Schema Design for the Purchasing Department.....	66
4.6.4.1	Fact Table: Purchase Order Facts.....	66
4.6.4.2	Dimension Tables.....	66
4.6.4.3	Schema Diagram Layout	67
4.6.4.4	WWI's Purchasing Department	68
4.6.4.5	Conclusion	71

5	Data Load Process for Purchase Data Mart	71
5.1	What is ETL?	71
5.1.1	Limitations of ETL	71
5.1.2	Why ETL Was Historically Preferred	72
5.2	Transition to ELT	73
5.2.1	Advantages of ELT	73
5.2.2	Key Features of ELT	73
5.3	Medallion Architecture for ELT	74
5.3.1	Structure of Medallion Architecture	74
5.3.2	How Medallion Architecture Fits into ELT	75
5.3.3	Benefits of Medallion Architecture in ELT	75
5.4	Medallion Architecture for Purchasing Department	76
5.4.1	Medallion Architecture in the ELT Framework	77
5.4.2	Tools Employed in the Data Pipeline	77
5.4.3	Alternative Tools Comparable to Airbyte and dbt	78
5.4.4	Rationale for Choosing Airbyte and dbt	79
5.4.5	High-Level Steps for Implementing Airbyte on Windows	79
5.4.6	High-Level Steps for Implementing dbt on Windows	81
5.5	Bronze Layer	82
5.5.1	Setup and Operation of Airbyte	82
5.5.2	Data Transfer with Airbyte	86
5.5.3	Normalisation of JSON Raw Data Using dbt Scripts	88
5.6	Silver Layer	92
5.6.1	Validation for Cities	92
5.6.2	Validation for State Provinces	95
5.6.3	Validation for Supplier Transactions	96
5.6.4	Validation for Colors	97
5.6.5	Validation for Countries	98
5.6.6	Validation for Customer Transactions	99
5.6.7	Validation for Customers	101
5.6.8	Validation for Delivery Methods	102
5.6.9	Validation for Invoices	103
5.6.10	Validation for Package Types	104
5.6.11	Validation for Payment Methods	104
5.6.12	Validation for Purchase Order Lines	105
5.6.13	Validation for Purchase Orders	106
5.6.14	Validation for Stock Items	106
5.6.15	Validation for Stock Item Holdings	107
5.6.16	Validation for Stock Item Transactions	108
5.6.17	Validation for Suppliers	109
5.6.18	Validation for Transaction Types	110
5.7	Gold Layer	110
5.7.1	Dimension Customer	111
5.7.2	Fact Purchase	113
6	Appendix	115
6.1	dbt Script Files	115

6.2 SQL Script Files.....	116
---------------------------	-----

TABLE OF FIGURES

Figure 1: WWI Business Model	6
Figure 2: WWI Stakeholder Analysis	8
Figure 3: WWI Organisational Structure	11
Figure 4: WWI SWOT Analysis.....	16
Figure 5: Core Business Processes Mapping	19
Figure 6: Information Flow.....	26
Figure 7: WWI Retail Customer	27
Figure 8: Sequence Diagram - Retail Customer Journey.....	28
Figure 9: Wholesale Buyer	29
Figure 10: Sequence Diagram - Wholesale Buyer Journey	30
Figure 11: Supplier	31
Figure 12: Sequence Diagram - Supplier Journey	32
Figure 13: Information Flow of Purchasing Functionality	34
Figure 14: Scope of Purchasing Functionality.....	35
Figure 15: Purchasing Dashboard.....	51
Figure 16 : Purchasing & Stock Analysis	52
Figure 17: Drill Trough Dashboard - Supplier Performance	52
Figure 18 : Drill Through Dashboard - Geographical Performance	53
Figure 19: Drill Through Dashboard - Stock Group Performance	53
Figure 20: Last Five Years Trend	54
Figure 21: Purchase Volume by City.....	55
Figure 22: LSTM Architecture	56
Figure 23: variation between the predicted and the actual purchase amounts.....	57

Figure 24: ERD for Purchase Functionality.....	59
Figure 25: WWI OLTP Database for Purchasing	61
Figure 26: Star Schema for Purchase Functionality	68
Figure 27: Facts & Dimensions for the Purchasing Department.....	70
Figure 28: Medallion Architecture.....	76
Figure 29: Docker Desktop for Airbyte	79
Figure 30: Airbyte Docker image running on WSL	80
Figure 31: Python Virtual Environment	81
Figure 32: dbt Installation.....	81
Figure 33: Airbyte Sourec Configuration	83
Figure 34: Airbyte Destination Configuration.....	84
Figure 35: Airbyte Connection Setup	85
Figure 36: Airbyte Job Monitoring	85
Figure 37: Airbyte Logs.....	86
Figure 38: Airbyte Normalisation was Not Supported	86
Figure 39: Target DB after Airbyte Run.....	87
Figure 40: Airbyte Raw Table	87
Figure 41: JSON Record.....	88
Figure 42: dbt Script for Supplier Transaction Raw to Normalise Table	88
Figure 43: dbt Solution for Raw to Normalise.....	89
Figure 44: dbt Execution Status for Raw Data	90
Figure 45: Record Comparison - Airbyte & dbt	91
Figure 46: Side-by-Side Comparison of Data - Source & Bronze Layer	91

Figure 47: Data Loss in Cities after Validation	93
Figure 48: Validation Script for Cities.....	94
Figure 49: Data Loss for Supplier Transaction.....	97
Figure 50: Reason for Data Loss with Supplier Transaction	97
Figure 51: Validation for Customer Transactions	100
Figure 52: Dimension Creation for Customer	112
Figure 53: Fact Purchasing	114

1 Introduction to WWI

A sample firm included in Microsoft's SQL Server sample databases is Wide World Importers (WWI). This entity represents a fictitious wholesale corporation that deals in various products, such as toys, novelty items, and other consumer goods. The purpose of the database is to provide a comprehensive data model that is appropriate for contemporary transactional systems, which includes the following:

- The sales and purchases section includes detailed information on sales transactions, purchase orders, and relationships with suppliers.
- Information about items, stock levels, and warehouse operations is included in inventory management terminology.
- Data pertaining to client demographics, purchase patterns, and supplier contracts are included in the category of customer and supplier relationships.
- Recordings of financial transactions, such as invoices, payments, and financial accounts, are referred to as financial transactions.
- Personnel information, job descriptions, and organisational structures are all part of human resources.

The database has many tables and relationships that are designed to simulate real-world business operations. This provides a rich dataset that can be used for testing, learning, and demonstrating the capabilities of SQL Server and for performing data analysis, business intelligence, and application development.

1.1 Comprehensive Business Analysis Approach for WWI

To comprehend the complexities of WWI, one must employ a thorough and systematic business analysis methodology. This technique guarantees a comprehensive perspective on the business, encompassing crucial elements such as the structure of the organisation, the flow of information, and the fundamental business procedures. Through a systematic analysis of these elements, WWI may determine its advantages, disadvantages, potential advantages, and potential disadvantages, which will ultimately inform its strategic decision-making and operational enhancements. This thorough examination aids in maximising present procedures, aligning corporate goals with stakeholder anticipations, improving overall efficiency, and maintaining a competitive edge.

1.1.1 Business Context and Objectives

- Comprehend the Business Model: Examine WWI's activities, unique selling points, sources of income, and expenses.
- Determine the main goals or objectives: Identify the principal objectives of the firm, such as enhancing sales, enhancing customer satisfaction, minimising expenses, or broadening market penetration.

1.1.2 Analysis of Stakeholders

- Determine the individuals or groups who have an interest or are affected by a particular project or decision: Enumerate all relevant stakeholders, encompassing customers, suppliers, employees, and investors.
- Comprehend the requirements of stakeholders: Evaluate the anticipated desires and prerequisites of every stakeholder group.

1.1.3 The Arrangement of an Organisation's Components and their Relationships.

- Generate an Organisational Chart: Create a comprehensive organisational chart that illustrates the hierarchical structure of the corporation. The chart must encompass all departments, functions, and the hierarchical reporting framework. Visualising the hierarchy aids in comprehending the allocation of responsibilities and the flow of command within the organisation.
- Establish clear and specific duties and obligations: Unambiguously establish the particular duties and obligations associated with crucial organisational positions. This requires clearly defining the responsibilities, desired results, and the impact of each function on the company's overarching goals. Comprehending these positions guarantees that each team member knows their responsibilities and how they contribute to the overarching organisational objectives, fostering effectiveness and responsibility.

1.1.4 SWOT Analysis

- Strengths: Identify internal advantages such as distinctive goods, robust supplier ties, or streamlined logistics.
- Weaknesses: Identify internal vulnerabilities such as inventory management challenges, restricted market penetration, or elevated operational expenses.

- Opportunities: Recognise external prospects, such as emerging markets, product diversification, or technology breakthroughs.
- Threats: Identify external factors that pose a risk, such as competition, regulatory changes, or economic downturns.

1.1.5 Analysis of the Process

- Outline Core Processes: Record essential business processes such as procurement, sales, order fulfilment, and customer support.
- Determine areas of congestion or restriction: Conduct a thorough examination of procedures to find any inefficiencies, delays, or areas that can be improved.
- Process Mapping: Record intricate procedures and pivotal decision-making moments
- Determine areas of congestion and potential areas for improvement

1.1.6 Analysis of the Market

- Market Segmentation: Divide the market into several groups based on customer demographics, purchasing behaviour, and preferences.
- Competitive Analysis: Evaluate competitors' strengths, weaknesses, and market positioning.
- Customer Analysis: Gain insights into customer requirements, levels of satisfaction, and loyalty.

1.1.7 Evaluation of Financial Data

- Revenue Streams: Analyse revenue distribution based on product lines, client segments, and geographic locations.
- Cost Structure Analysis: Evaluate the various cost elements, such as fixed and variable expenses, to discover potential cost reduction options.
- Profitability Analysis: Assess the profitability of a business by examining factors such as profit margins, return on investment (ROI), and overall financial well-being.

1.1.8 Evaluation Criteria and Key Performance Indicators

- Define KPIs: Establish specific metrics that will be used to evaluate the level of success achieved in relation to business objectives. These metrics may include sales growth, inventory turnover, and customer satisfaction.
- Benchmarking: Evaluate WWI's performance by comparing it to industry standards and best practices.

1.1.9 Evaluation of Hazards and Their Likelihood and Impact

- Identify hazards: Enumerate potential hazards, encompassing operational, financial, market, and strategic risks.
- Mitigation methods: Formulate methods to minimise identified risks and maintain the uninterrupted operation of the firm.

1.1.10 Strategic Planning

- Formulate strategies: Create plans to accomplish corporate goals, such as expanding into new markets, diversifying products, or optimising costs.
- Implementation Plan: Develop a practical plan for carrying out a project, including specific dates, assigned duties, and allocation of resources.

The structured approach is essential for WWI since it offers a systematic framework to comprehensively comprehend and enhance different aspects of the organisation. By systematically mapping fundamental processes and visually representing the flow of information, it becomes possible to identify and rectify inefficiencies and gaps in communication. Furthermore, conducting a comprehensive examination of the organisational structure aids in clarifying the specific duties and obligations of each team member, thereby guaranteeing that all individuals are following the company's strategic objectives. Financial analysis and performance indicators allow for monitoring progress and assessing economic well-being, while risk analysis helps the organisation prepare for potential obstacles in a proactive manner. By incorporating these components, WWI can attain a well-rounded and knowledgeable viewpoint on its activities, resulting in improved strategic choices and enduring expansion.

2 Business Analysis

2.1 Business Context and Objectives

To understand WWI's operations, we must review its business model comprehensively. This includes analysing key components such as the value proposition, revenue streams, and cost structure. Here is a detailed Business Model Canvas:

- **Key Partners:** Suppliers, logistics providers, financial institutions, and technology partners.
- **Key Activities:** Procurement, inventory management, sales and marketing, customer service, logistics and distribution.
- **Key Resources:** Warehouses, distribution centres, technology infrastructure, skilled workforce, customer database.
- **Value Propositions:** Wide range of products, competitive pricing, reliable supply chain, excellent customer service.
- **Customer Relationships:** Personalised service, loyalty programs, customer support.
- **Channels:** Direct sales, online platform, wholesale distribution, retail partnerships.
- **Customer Segments:** Retailers, individual consumers, corporate clients, international markets.
- **Cost Structure:** Procurement costs, operational expenses (warehousing, logistics), marketing expenses, and technology investments.
- **Revenue Streams:** Product sales, service fees, subscription models, advertising revenue.

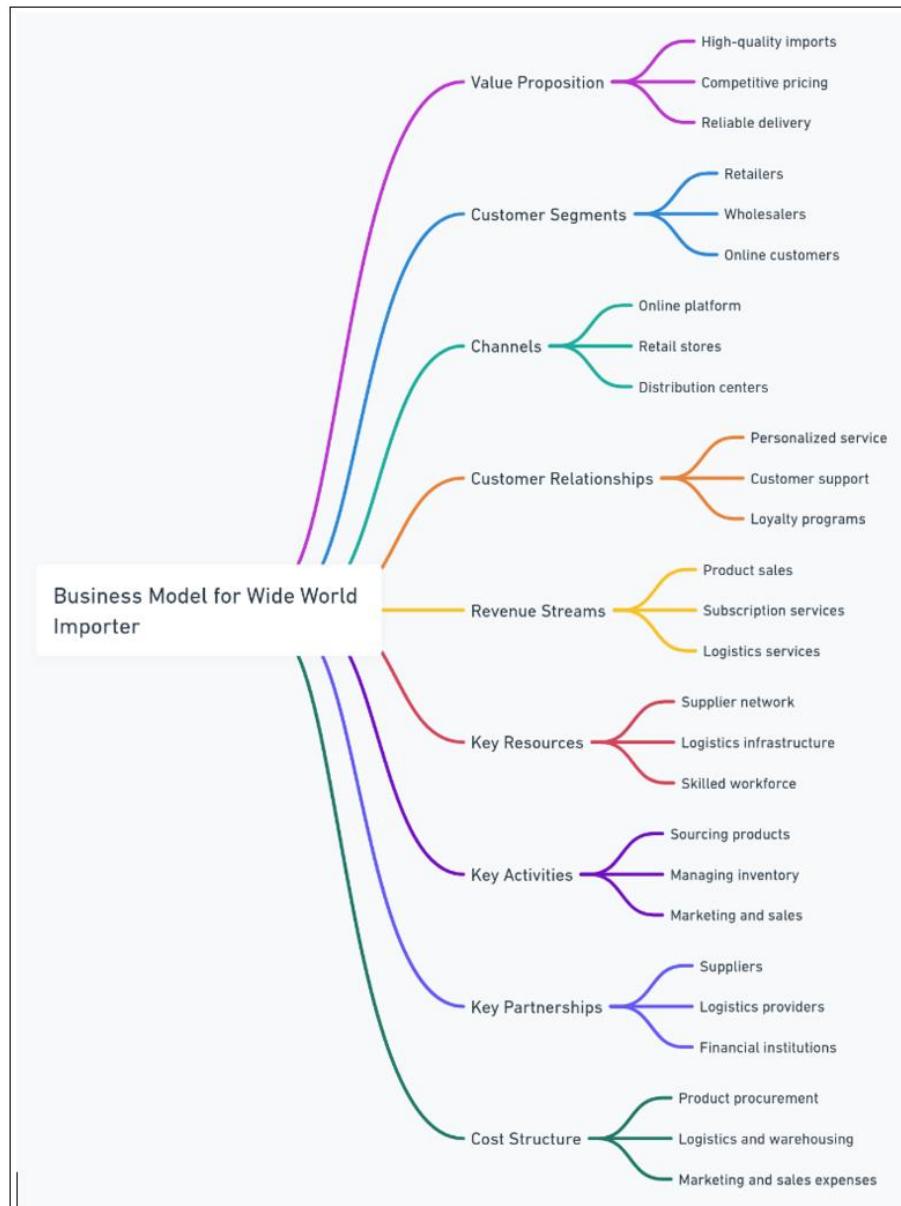
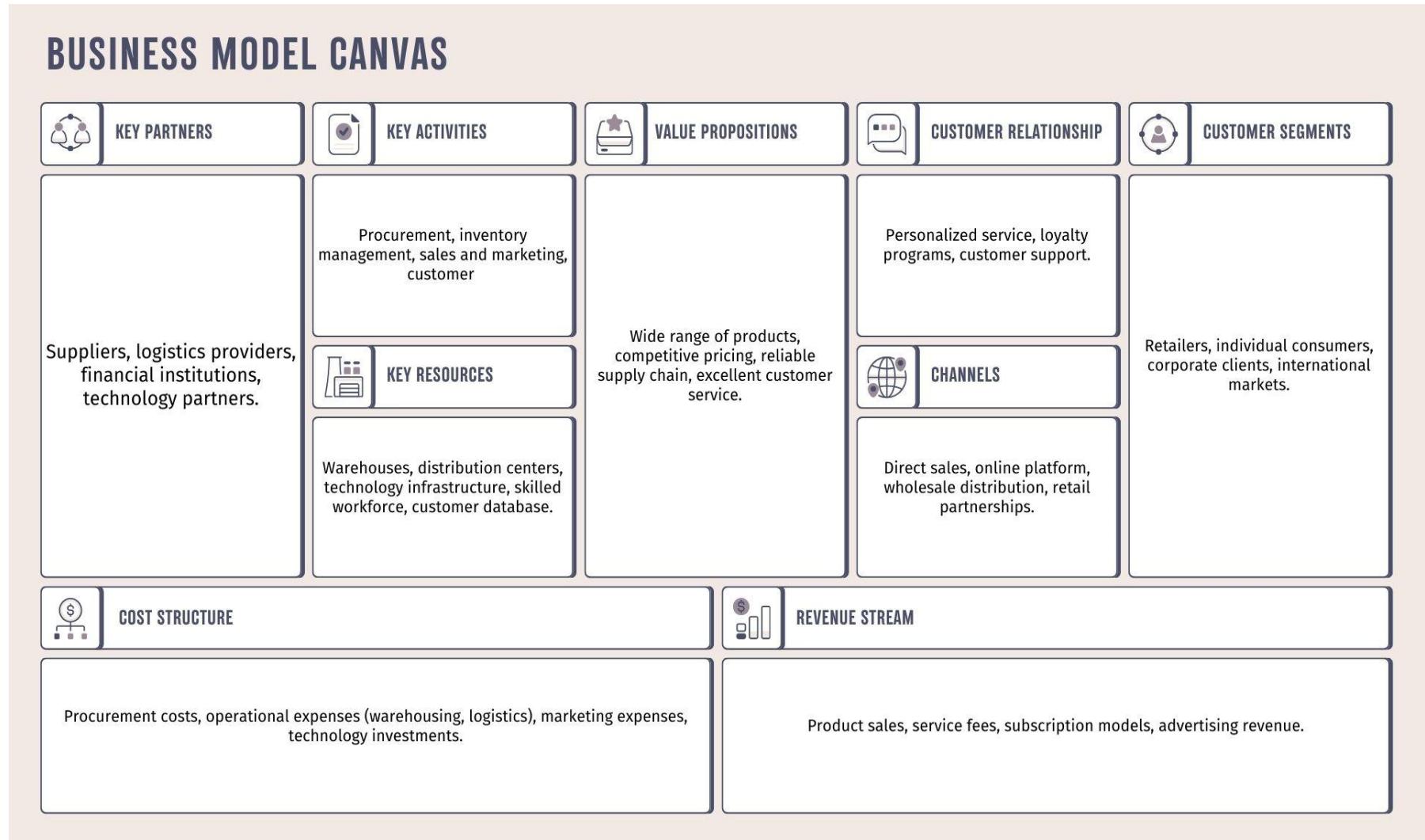


Figure 1: WWI Business Model

2.1.1 Business Model



2.1.2 Key Business Objectives

Identifying the primary goals of WWI is essential for aligning strategies and operations to achieve desired outcomes. The key objectives typically include:

- **Increasing Sales:** Focus on expanding the customer base, enhancing product offerings, and improving sales strategies to drive revenue growth.
- **Improving Customer Satisfaction:** Implement measures to enhance customer service, ensure product quality, and foster strong customer relationships.
- **Reducing Costs:** Identify and eliminate inefficiencies in operations, optimise supply chain management, and adopt cost-saving technologies.
- **Expanding Market Reach:** Explore new markets, diversify product lines, and establish strategic partnerships to increase market presence.

By systematically analysing these elements and understanding the business model and objectives, WWI can develop effective strategies to drive growth, improve efficiency, and achieve long-term success.

2.2 Stakeholder Analysis

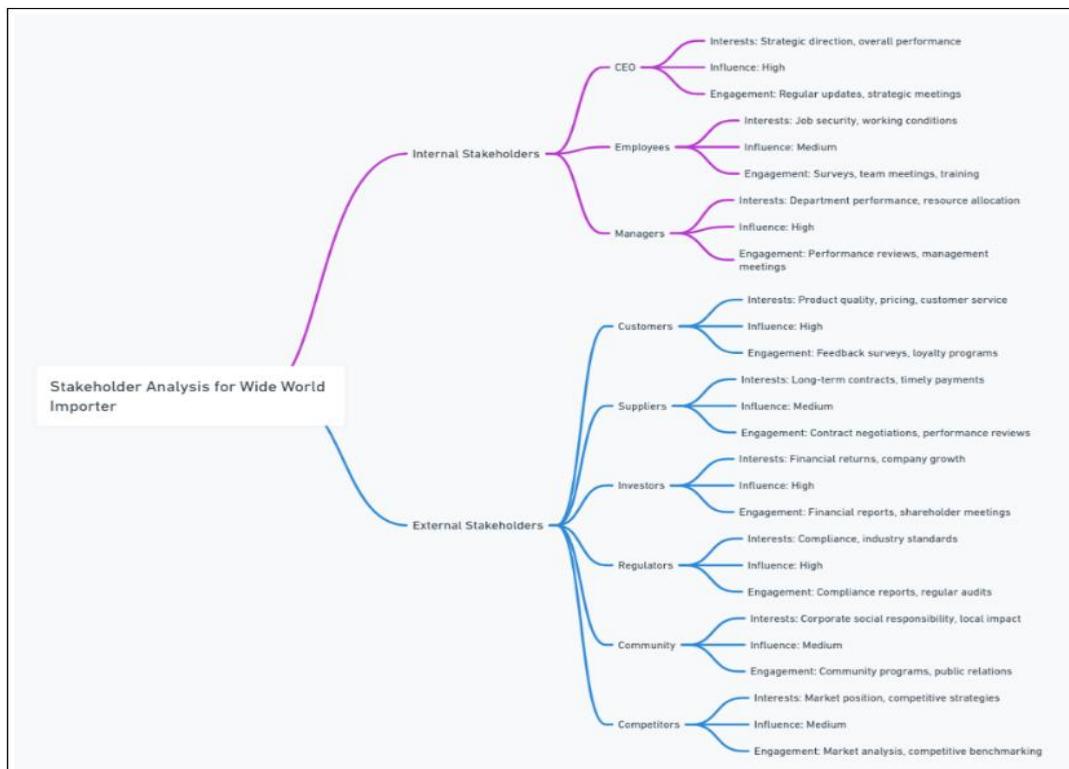


Figure 2: WWI Stakeholder Analysis

Understanding and managing stakeholders is crucial for WWI's success. The diagram (Figure 2) and analysis provide a comprehensive view of WWI's stakeholders and their needs.

2.2.1 Stakeholder Groups and Needs

Customers Expectations and Needs

- High-quality products
- Competitive Pricing
- Excellent customer service
- Timely delivery
- Loyalty programs and personalised offers

Suppliers' Expectations and Needs

- Long-term partnerships
- Reliable and timely payments
- Clear communication
- Fair negotiation terms
- Volume commitments

Employees Expectations and Needs

- Job security
- Competitive compensation and benefits
- Career development and training opportunities
- Positive work environment
- Recognition and rewards

Investors' Expectations and Needs

- Strong financial performance and returns.

- Transparent and timely reporting
- Strategic growth and risk management
- Sustainable business practices
- Regular communication and updates

2.2.2 Analysis of Stakeholder Relationship

By understanding and addressing the needs of these stakeholder groups, WWI can build strong relationships that support its business objectives and long-term success.

Customers:

Customers are the lifeblood of WWI, driving sales and revenue. Meeting their needs ensures customer satisfaction and loyalty, leading to repeat business and positive word-of-mouth.

Suppliers:

Establishing strong, reliable supplier relationships ensures a steady supply of quality products. Clear communication and fair terms foster trust and collaboration.

Employees:

Employees are the backbone of WWI's operations. Satisfied employees are more productive and engaged, contributing to the company's overall success.

Investors:

Providing investors with consistent returns and transparent communication helps secure financial support for growth and expansion initiatives.

2.3 Organisational Chart

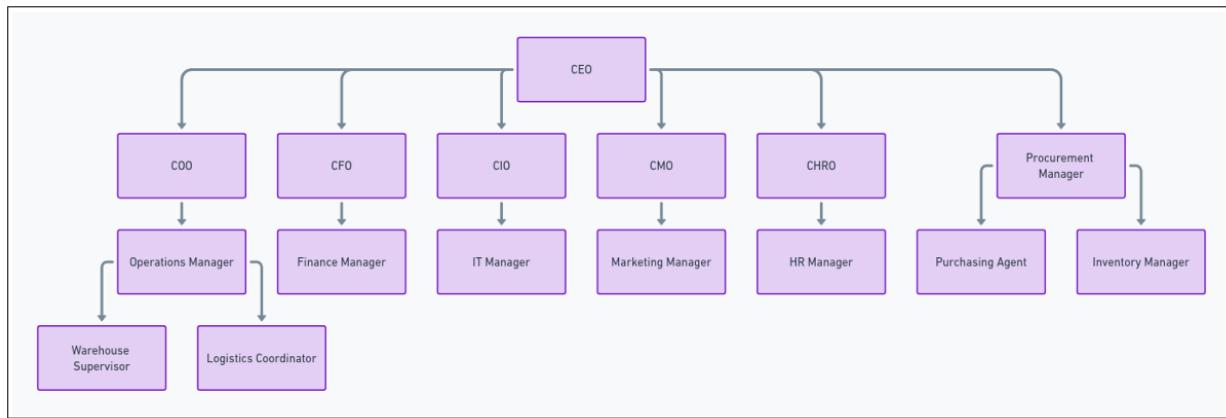


Figure 3: WWI Organisational Structure

The organisational structure of Wide World Importers is designed to ensure efficient operations and clear lines of communication. Below, we will discuss the company's hierarchy, roles, and responsibilities in detail.

2.3.1 Executive Leadership:

Chief Executive Officer (CEO):

- Responsible for the company's overall strategic direction and leadership.
- Manages major corporate decisions and overall operations.
- Acts as the main point of communication between the board of directors and corporate operations.

Chief Operating Officer (COO):

- Oversees the day-to-day operations.
- Ensures operational efficiency and effectiveness.
- Implements business strategies and coordinates activities across various departments.

Chief Financial Officer (CFO):

- The organisation's financial health is managed through comprehensive financial planning, risk management, and financial reporting.
- Oversees budgeting and financial analysis to ensure the company's financial stability.

Chief Information Officer (CIO):

- Responsible for the company's technological direction.
- Ensures that the IT infrastructure supports business goals, manages IT resources, and oversees the implementation of technology solutions.

Chief Marketing Officer (CMO):

- Develops and implements marketing strategies to enhance brand management and customer engagement.
- Oversees marketing campaigns, market research, and public relations efforts to drive the company's market presence.

Chief Human Resources Officer (CHRO):

- Focuses on employee recruitment, development, and welfare.
- Manages HR policies to ensure compliance with labour laws and fosters employee engagement and retention.

2.3.2 Operations Department:

Operations Manager:

- Oversees warehouse and logistics operations.
- Ensures efficient supply chain management and inventory control.

Warehouse Supervisor:

- Manages warehouse staff and daily operations.
- Ensures that inventory is safely stored and managed and maintains warehouse safety.

Logistics Coordinator:

- Manages transportation and delivery schedules.
- Optimises delivery routes selects carriers, and ensures timely and cost-effective delivery of goods.

2.3.3 Finance Department:

Finance Manager:

- Oversees accounting, payroll, and financial analysis.
- Ensures accurate financial reporting and compliance with financial regulations.

Accountants:

- Manage financial records and statements.
- Ensure compliance with financial regulations.

Payroll Specialist:

- Manages employee compensation and manages payroll processing and benefits.

Financial Analysts:

- Perform financial analysis and planning.
- Support budgeting and forecasting processes.

2.3.4 Information Technology (IT) Department:

IT Manager:

- Manages the IT department and oversees IT support, network administration, and system security.

Network Administrators:

- Maintain and support network infrastructure.
- Ensure security and connectivity.

System Analysts:

- Analyse and design IT systems.
- Support their implementation and maintenance.

Database Administrators:

- Manage and maintain databases.
- Ensure data integrity and security.

Support Technicians:

- Provide technical support to staff.
- Troubleshoot IT issues and maintain hardware.

2.3.5 Marketing and Sales Department:

Marketing Manager:

- Develops and executes marketing campaigns.
- Coordinates sales efforts and customer service.

Sales Representatives:

- Engage with customers to drive sales and manage client relationships.

Marketing Specialists:

- Create marketing materials and strategies.
- Support promotional activities and campaigns.

Customer Service Representatives:

- Managed customer inquiries, processed orders, and resolved issues.
- Ensure customer satisfaction.

2.3.6 Human Resources (HR) Department:

HR Manager:

- Oversees recruitment, training, and employee relations.
- Ensures compliance with labor laws and company policies.

Recruitment Specialists:

- Source and recruit new employees.
- Manage the hiring process.

Training and Development Coordinators:

- Organise and conduct training programs.

- Support employee development.

HR Assistants:

- Provide administrative support to HR functions.
- Assist with employee records and HR processes.

2.3.7 Procurement Department:

Procurement Manager:

- Oversees purchasing and supplier management.
- Ensures optimal inventory levels and cost-effective procurement.

Purchasing Agents:

- Manage the procurement of goods and services.
- Negotiate with suppliers and handle contracts.

Inventory Managers:

- Monitor and manage inventory levels.
- Ensure the availability of necessary supplies.

This organisational structure ensures that each department has clearly defined roles and responsibilities, promoting efficiency, accountability, and alignment with the company's overall objectives.

2.4 SWOT Analysis



Figure 4: WWI SWOT Analysis

2.4.1 Strengths

Wide World Importers (WWI) boasts several key strengths that provide a competitive edge in the market:

1. Strong Supplier Relationships:

Established and reliable relationships with suppliers ensure a steady flow of high-quality goods.

2. Efficient Logistics Network:

An efficient logistics network enhances operational capabilities, allowing for timely and cost-effective distribution of products.

3. Diverse Product Range:

A wide range of products caters to various customer needs and preferences, enhancing market appeal.

4. Experienced Management Team:

The experienced management team provides strategic direction and ensures smooth operations across all departments.

2.4.2 Weaknesses

Despite its strengths, WWI faced specific weaknesses that could impact its performance:

1. Dependence on International Suppliers:

Dependence on international suppliers can lead to vulnerabilities in the supply chain, such as delays and increased costs due to geopolitical issues or trade restrictions.

2. High Operational Costs:

High operational costs, including warehousing and logistics expenses, can affect profitability.

3. Limited Online Presence:

A limited online presence restricts reaching a broader customer base and capitalising on e-commerce opportunities.

4. Vulnerability to Supply Chain Disruptions:

Vulnerabilities to supply chain disruptions, such as delays or interruptions, can adversely affect operations and customer satisfaction.

2.4.3 Opportunities

WWI has numerous opportunities to explore for growth and expansion:

1. Market Expansion:

Potential for expansion into new geographic and demographic markets can increase market share and revenue.

2. Adoption of E-commerce Platforms:

Enhancing the online presence through e-commerce platforms can cater to the growing demand for online shopping.

3. Strategic Partnerships:

Strategic partnerships with other businesses can open new avenues for collaboration and mutual growth.

4. Increased Demand for Imported Goods:

Rising demand for imported goods presents an opportunity to capitalise on WWI's strengths in sourcing and distribution.

2.4.4 Threats

WWI also faces several external threats that could impact its business:

1. Economic Fluctuations:

Economic fluctuations, such as changes in exchange rates and economic downturns, can affect financial stability and profitability.

2. Regulatory Changes:

New trade policies, tariffs, and regulatory changes pose significant threats, potentially increasing costs and affecting operations.

3. Intense Competition:

Intense competition in the import and distribution industry can challenge WWI's market position and impact market share.

4. Geopolitical Instability:

Geopolitical instability affecting international trade can disrupt supply chains and impact the availability of goods, posing a threat to operations.

This SWOT analysis allows WWI to develop strategies to leverage its strengths, address weaknesses, capitalise on opportunities, and mitigate threats. This comprehensive understanding aids in strategic planning and decision-making to ensure sustained growth and competitive advantage.

2.5 Process Flow in WWI

Mapping core processes is crucial for identifying inefficiencies and areas for improvement within Wide World Importers. The key business processes include procurement, sales, order fulfilment, and customer service. Each process is documented with detailed steps and decision points. The following figure illustrates core business processors for WWI.

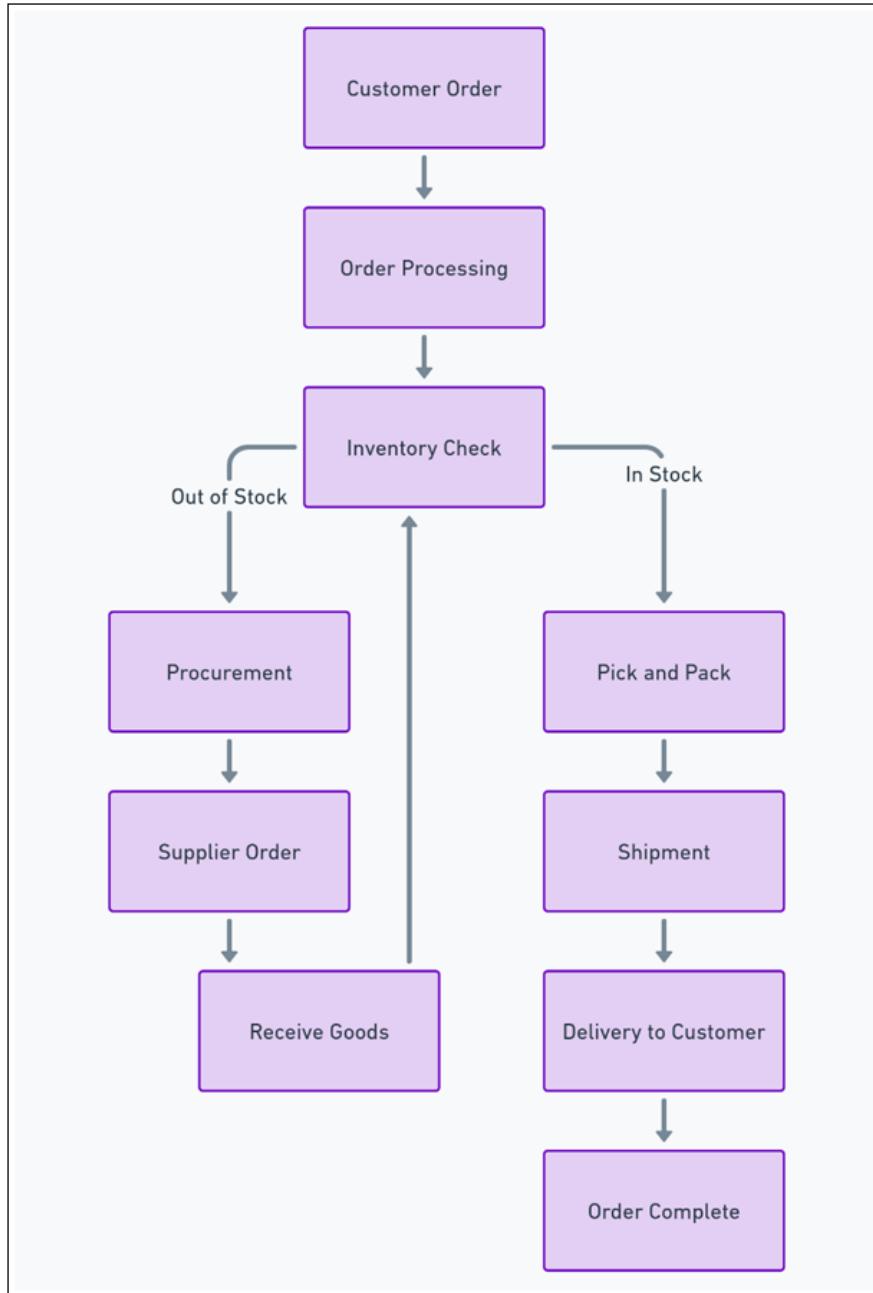


Figure 5: Core Business Processes Mapping

2.5.1 Procurement Process:

- **Demand Forecasting:** Analyse sales data and inventory levels to forecast future demand.
- **Supplier Selection:** Evaluate and select suppliers based on quality, price, and reliability.
- **Order Placement:** Order with selected suppliers specifying quantities, delivery schedules, and terms.

- **Order Tracking:** Monitor the status of orders to ensure timely delivery.
- **Receiving and Inspection:** Inspect received goods for quality and quantity compliance.
- **Payment Processing:** Coordinate with finance to process supplier invoices and payments.
- **Performance Evaluation:** Regularly evaluate supplier performance and adjust procurement strategies.

2.5.2 Sales Process:

- **Lead Generation:** Identify potential customers through marketing efforts and market research.
- **Customer Engagement:** Engage with potential customers through direct sales, online platforms, and retail partnerships.
- **Quotation and Negotiation:** Provide quotes to customers and negotiate terms and conditions.
- **Order Confirmation:** Confirm customer orders and update the Order Management System (OMS).
- **Order Fulfillment:** Coordinate with inventory and logistics to fulfil orders.
- **After-Sales Support:** Provide customer support and address any post-sale issues or inquiries.

2.5.3 Order Fulfillment Process:

- **Order Entry:** Enter customer orders into the OMS.
- **Inventory Check:** Verify inventory availability for the requested items.
- **Pick and Pack:** Gather and pack the products for shipment.
- **Shipping:** Arrange for transportation and delivery of the products to the customer.
- **Delivery Confirmation:** Confirm delivery and update the OMS.
- **Returns Management:** Handle returns and exchanges, ensuring customer satisfaction.

2.5.4 Customer Service Process:

- **Inquiry Handling:** Respond to customer inquiries through various channels (phone, email, chat).
- **Issue Resolution:** Address customer issues like product defects or delivery problems.
- **Feedback Collection:** Gather customer feedback to improve products and services.
- **Customer Relationship Management:** Maintain customer relationships through personalised service and loyalty programs.

2.5.5 Identifying Bottlenecks

Analysing these processes helps identify bottlenecks that can lead to inefficiencies, delays, or increased costs:

2.5.5.1 *Procurement Bottlenecks:*

- **Supplier Delays:** Delays in supplier deliveries can disrupt the supply chain.
- **Quality Issues:** Receiving substandard goods requires additional time and resources to resolve.
- **Complex Approval Processes:** Lengthy approval processes for purchase orders can slow procurement.

2.5.5.2 *Sales Bottlenecks:*

- **Lead Conversion:** Difficulty in converting leads to customers and can hinder sales growth.
- **Quotation Delays:** Delays in providing quotes can lead to lost sales opportunities.
- **Order Processing:** Inefficiencies in order processing can delay order fulfilment.

2.5.5.3 *Order Fulfillment Bottlenecks:*

- **Inventory Shortages:** Stockouts can delay order fulfillment and affect customer satisfaction.
- **Picking Errors:** Errors in picking the correct products can lead to returns and exchanges.
- **Shipping Delays:** Delays in shipping can negatively impact on customer experience.

2.5.5.4 Customer Service Bottlenecks:

- **Response Time:** Slow response times to customer inquiries can lead to dissatisfaction.
- **Issue Resolution:** Prolonged issue resolution can affect customer loyalty.
- **Feedback Implementation:** Delays in implementing customer feedback can hinder service improvements.

2.5.6 Optimisation Opportunities

Identifying optimisation opportunities within these processes can enhance efficiency and performance.

2.5.6.1 Procurement Optimisation:

- **Supplier Collaboration:** Foster closer collaboration with suppliers for better lead time management.
- **Automated Approvals:** Implement automated approval workflows to expedite purchase order approvals.
- **Quality Management:** Establish strict quality control measures to reduce the occurrence of substandard goods.

2.5.6.2 Sales Optimisation:

- **CRM Systems:** Utilise Customer Relationship Management (CRM) systems to streamline lead management and improve conversion rates.
- **Efficient Quotation Systems:** Implement systems to provide quicker and more accurate quotes to potential customers.
- **Order Automation:** Automate order processing to reduce manual errors and speed up fulfilment.
- Order Fulfillment Optimisation:
- **Inventory Management:** Use real-time inventory management systems to ensure optimal stock levels.
- **Automated Picking Systems:** Implement automated picking systems to reduce errors and improve efficiency.
- **Optimised Shipping Routes:** Use route optimisation software to enhance delivery efficiency and reduce shipping times.

2.5.6.3 Customer Service Optimisation:

- **AI Chatbots:** Deploy AI chatbots to provide faster initial responses to customer inquiries.
- **Streamlined Issue Tracking:** Use issue tracking systems to promptly monitor and resolve customer issues.
- **Proactive Feedback Mechanisms:** Implement proactive mechanisms to gather and act on customer feedback more efficiently.

WWI can enhance operational efficiency, reduce costs, and improve customer satisfaction by mapping core processes, identifying bottlenecks, and exploring optimisation opportunities.

2.6 Market Analysis for WWI

The market analysis for Wide World Importers involves a comprehensive examination of market segmentation, competitive analysis, and customer analysis. This analysis provides insights into the market landscape and helps WWI develop effective strategies to enhance its market presence and meet customer needs.

2.6.1 Market Segmentation

Market segmentation is a critical component of understanding WWI's diverse customer base. The company segments its market based on numerous factors, including customer demographics, purchasing behaviour, and preferences. This segmentation allows WWI to tailor its products and services to meet the specific needs of different customer groups.

Retailers constitute a significant segment, seeking a wide range of products at competitive prices to stock their shelves and meet consumer demands. Individual consumers form another important segment, often looking for novelty items and toys that cater to personal or family needs. Corporate clients, including businesses and institutions, represent a segment requiring bulk purchases and reliable supply chains for operational needs. Additionally, WWI serves international markets, focusing on export opportunities to expand its global reach.

2.6.2 Competitive Analysis

Understanding the competitive landscape is crucial for WWI to maintain and enhance its market position. The competitive analysis involves assessing key competitors' strengths, weaknesses, market positioning, and import and distribution industry strategies.

WWI faces competition from other wholesale distributors that offer similar product ranges and target the same customer segments. Key competitors may have established brands, strong supplier networks, and efficient coordination systems, presenting significant challenges. However, WWI's strengths lie in its diverse product range, competitive pricing, and strong supplier relationships. To

stay competitive, WWI must continuously monitor competitor activities, adapt to market trends, and leverage its strengths to differentiate its offerings.

2.6.3 Customer Analysis

Customer analysis provides insights into WWI's customers' needs, satisfaction levels, and loyalty. WWI can develop strategies to improve customer engagement and retention by understanding these factors.

Customers expect high-quality products, competitive pricing, and excellent customer service. Meeting these expectations is essential for ensuring customer satisfaction and loyalty. WWI engages with customers through various channels, including direct sales, online platforms, and wholesale distribution. Personalised service and loyalty programs are vital in maintaining strong customer relationships.

Customer feedback indicates that timely delivery and responsive customer support influence their satisfaction. Addressing these aspects can enhance the overall customer experience. Additionally, analysing purchasing behaviour helps WWI identify trends and preferences, allowing the company to adjust its product offerings and marketing strategies accordingly.

By segmenting the market, assessing competitors, and understanding customer needs, WWI can develop targeted strategies to enhance its market presence and achieve its business objectives. This comprehensive market analysis provides a foundation for informed decision-making and strategic planning.

2.6.4 Financial Analysis

The financial analysis of Wide World Importers (WWI) is crucial for understanding the company's revenue streams, cost structure, and overall financial health. This analysis helps evaluate profitability, identify cost-saving opportunities, and ensure sustainable financial performance.

2.6.4.1 Revenue Streams

WWI generates revenue from several key sources. The primary revenue stream comes from product sales, which include a diverse range of novelty items, toys, and various consumer goods sold to retailers, individual consumers, and corporate clients. The company also derives income from service fees associated with specialised services such as expedited shipping, product customisation, and after-sales support.

Additionally, WWI explores subscription models for recurring revenue, particularly for clients who require regular supplies of specific products. Advertising revenue is another stream earned from partnerships with brands that want to promote their products through WWI's platforms.

Breaking down revenue by product lines, customer segments, and geographic regions provides a detailed understanding of where the company's income originates. This granular view helps identify the most profitable areas and potential growth opportunities.

2.6.4.2 Cost Structure

The cost structure of WWI encompasses several significant components. Procurement costs represent a sizable portion, involving expenses related to purchasing goods from suppliers. Operational expenses, including warehousing, coordination, and distribution costs, also form a considerable part of the cost structure. These costs are essential for maintaining the supply chain and ensuring efficient delivery of products to customers.

Marketing expenses are incurred to promote WWI's products and services, attract new customers, and retain existing ones. Investments in technology infrastructure, such as maintaining and upgrading the company's IT systems, also contribute to the overall costs. Administrative expenses, including salaries, office supplies, and utilities, are also necessary for day-to-day operations.

Analysing these cost components helps WWI identify areas where cost-saving measures can be implemented without compromising service quality or operational efficiency.

2.6.4.3 Profitability Analysis

Evaluating the profitability of WWI involves assessing profit margins, return on investment (ROI), and overall financial health. Profit margins are calculated by comparing the company's revenue against its costs, indicating its operations' financial efficiency and effectiveness.

Return on investment measures the financial return generated from the investments made in various aspects of the business, such as technology, marketing, and human resources. A higher ROI indicates that the investments yield favourable returns, contributing to the company's profitability.

Financial health is assessed through various financial ratios and metrics, including liquidity, solvency, and profitability ratios. These metrics provide insights into the company's ability to meet its short-term and long-term obligations, manage debts, and generate sustainable profits.

By conducting a thorough financial analysis, WWI can clearly understand its financial position, identify areas for improvement, and develop strategies to enhance profitability and ensure long-term financial stability.

Information Flow in WWI

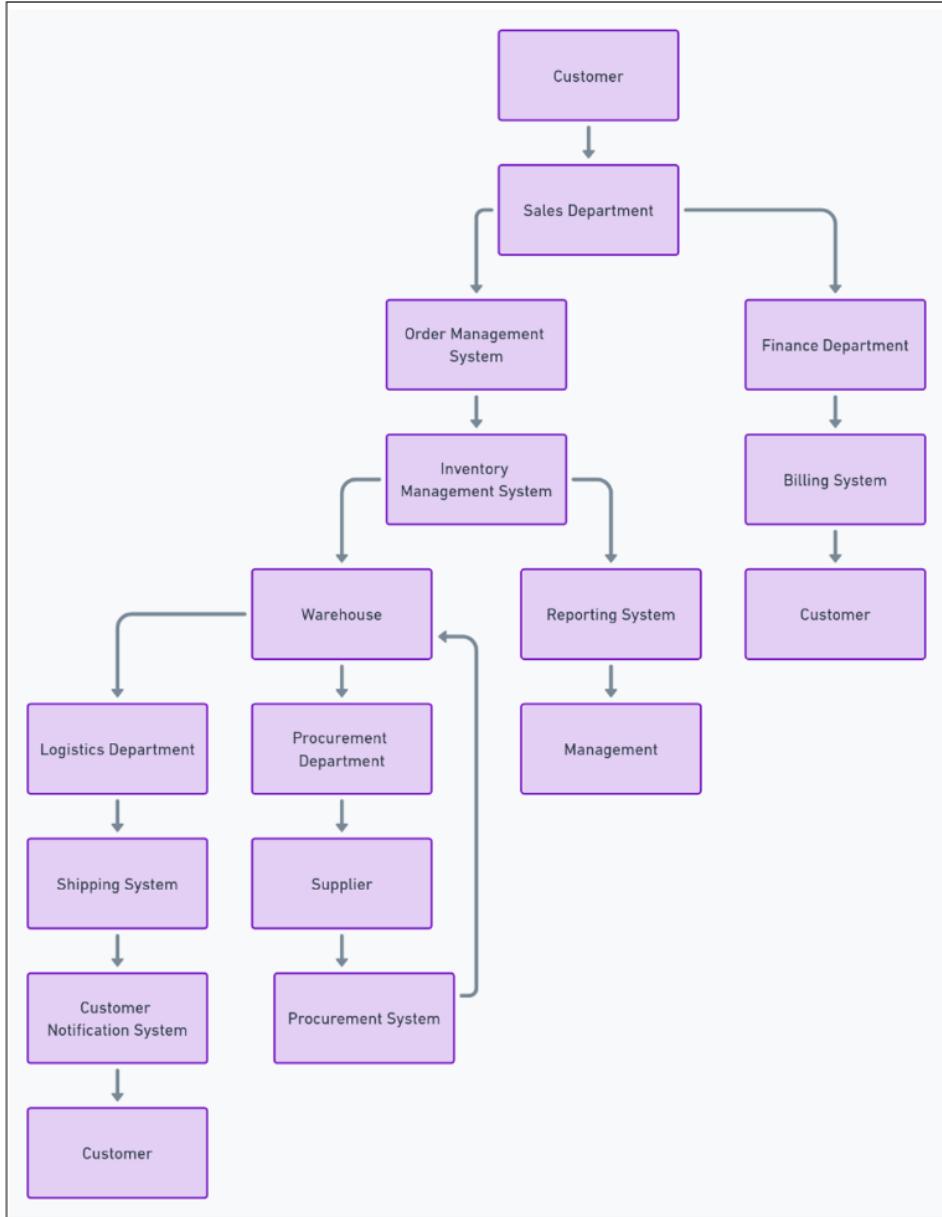


Figure 6: Information Flow

The information flow in Wide World Importer is designed to ensure seamless communication and coordination across various departments and systems. It begins with the customer interacting with the sales department, where orders are initiated and entered into the Order Management System (OMS). The OMS then communicates with the Inventory Management System (IMS) to check stock levels and update inventory records.

The IMS shares information with the warehouse, which handles the physical storage and retrieval of products. Simultaneously, the logistics department is notified to arrange shipment details, which are then processed through the Shipping System. The Shipping System sends updates to the Customer Notification System, keeping customers informed about the status of their orders.

In parallel, the sales department communicates with the finance department to handle billing and payment processes. The finance department uses the Billing System to generate invoices and process payments, providing feedback to the customer.

The warehouse also coordinates with the procurement department when inventory levels are low. The procurement department places orders with suppliers and updates the Procurement System, communicating with the IMS to ensure current inventory records.

The Reporting System collects data from various departments throughout this process, providing valuable insights and reports to the management team. This comprehensive information flow ensures that all stakeholders are well-informed and that operations run smoothly, contributing to the overall efficiency and effectiveness of Wide World Importer.

2.7 User Journey Analysis for WWI

2.7.1 Retail Customer Journey

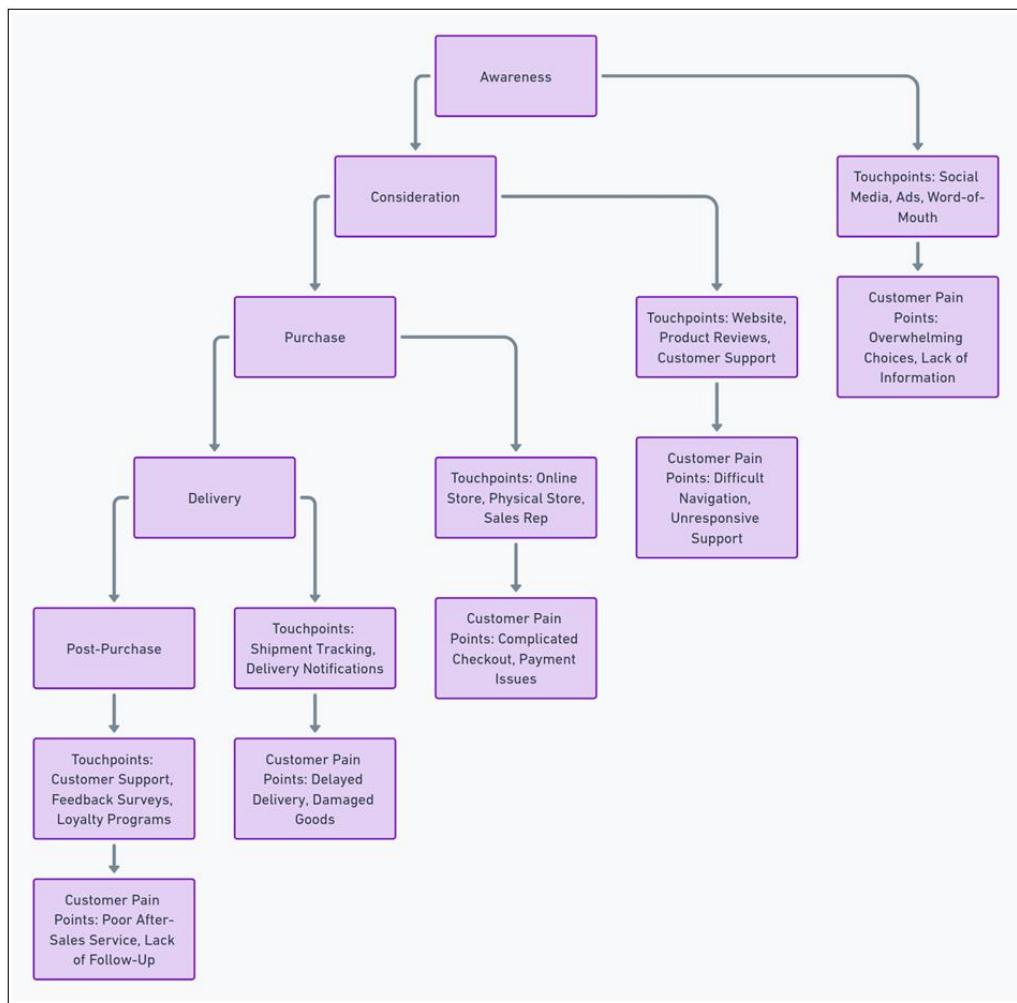


Figure 7: WWI Retail Customer

The retail customer journey begins with awareness through touchpoints like social media, advertisements, and word-of-mouth. Customers interact with the website during consideration, read product reviews, and contact customer support. Pain points here include overwhelming choices and a lack of information. When purchasing, they might face issues with the checkout process or payment. Delivery involves tracking and notifications, with potential delays or damaged goods being common pain points. Post-purchase, customers engage with support, provide feedback, and participate in loyalty programs, where poor after-sales service and lack of follow-up can be problematic.

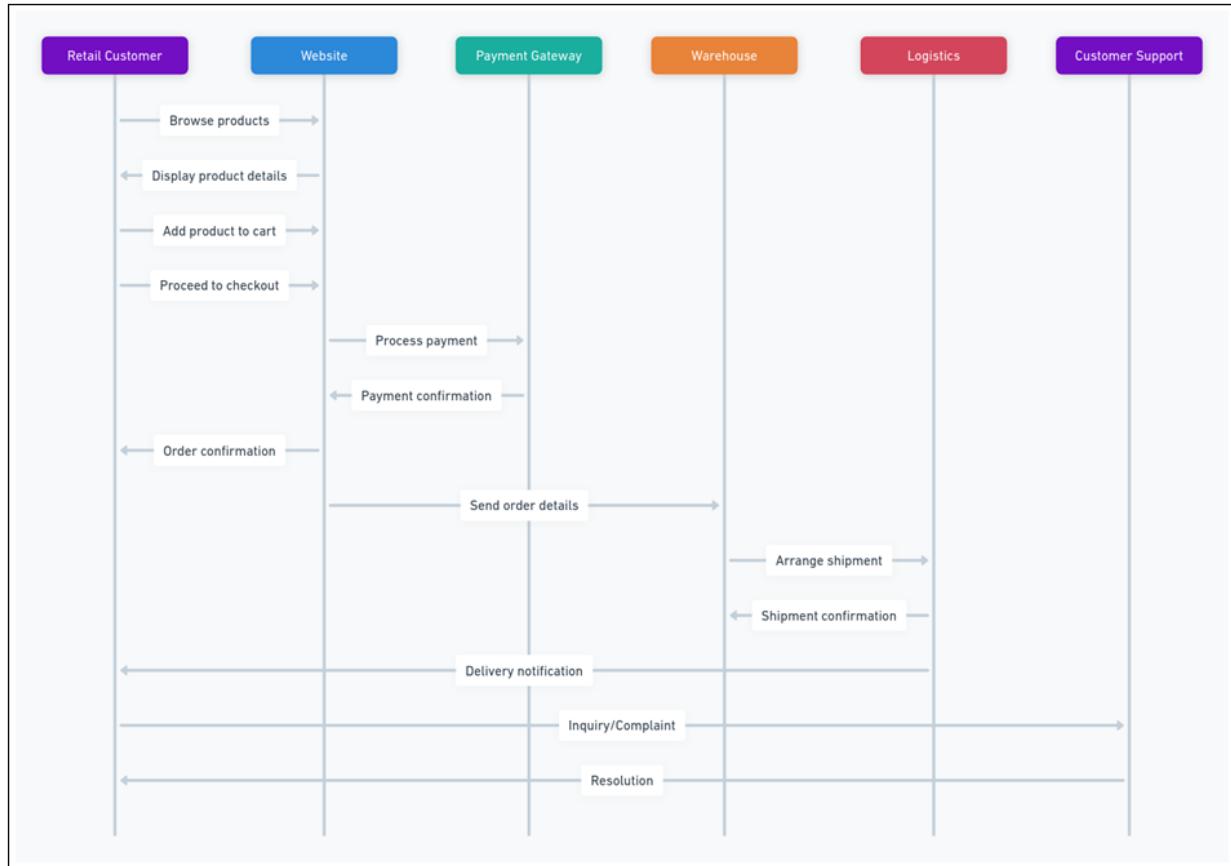


Figure 8: Sequence Diagram - Retail Customer Journey

2.7.2 Wholesale Buyer Journey

Wholesale buyers become aware of Wide World Importer through industry events, trade shows, and B2B marketing. In the consideration stage, they browse the website, examine product catalogues, and interact with sales reps. They might encounter complicated information and unresponsive representatives. The negotiation phase involves meetings, proposals, and contracts, where lengthy negotiations and contract issues can arise. During the purchase phase, they use the online portal or deal directly with sales reps, facing potential order processing and payment issues. Delivery includes tracking and notifications, with common problems like delayed deliveries or

damaged goods. Post-purchase, buyers interact with account management and provide feedback, often encountering poor account management and a lack of follow-up.

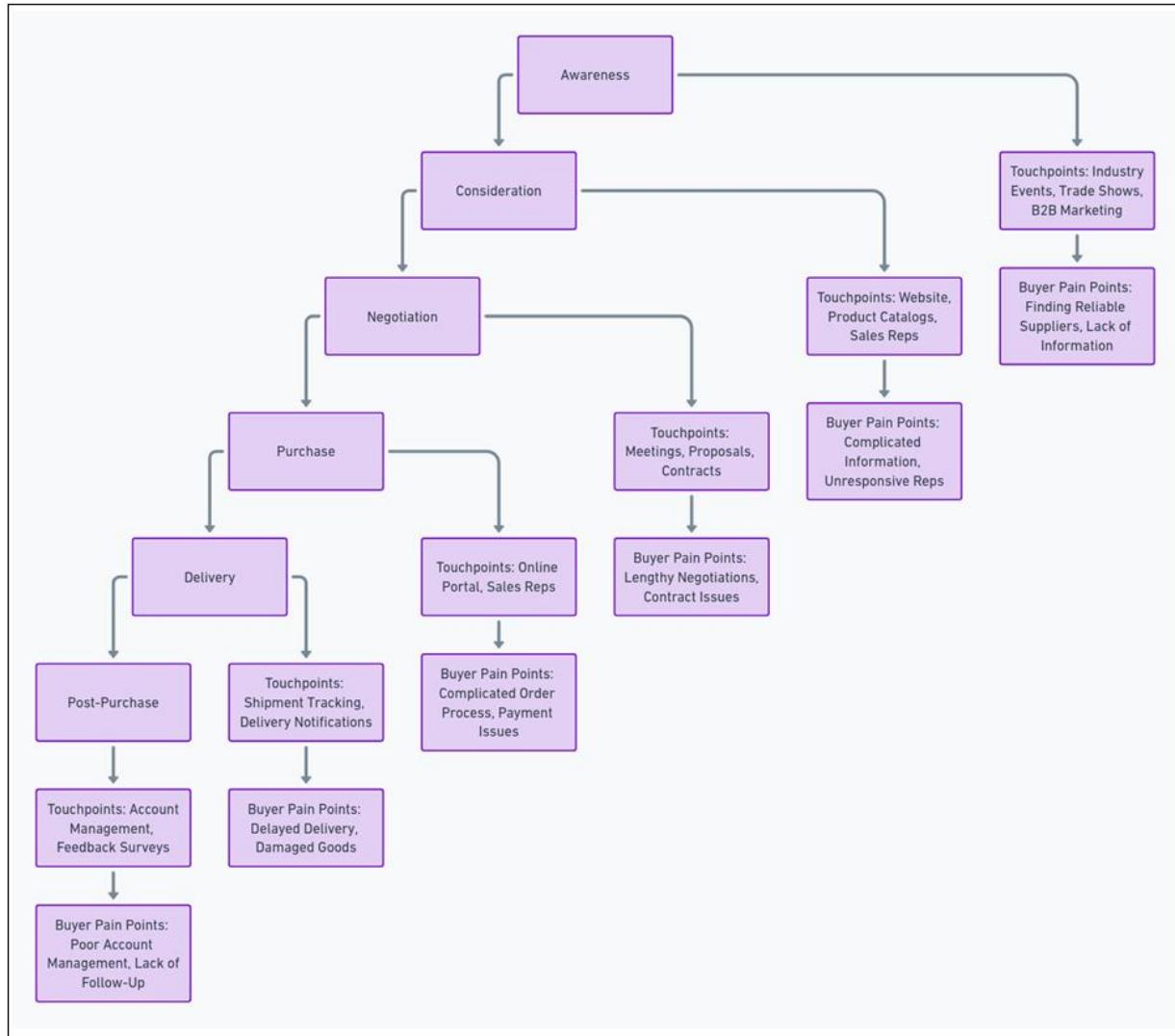


Figure 9: Wholesale Buyer

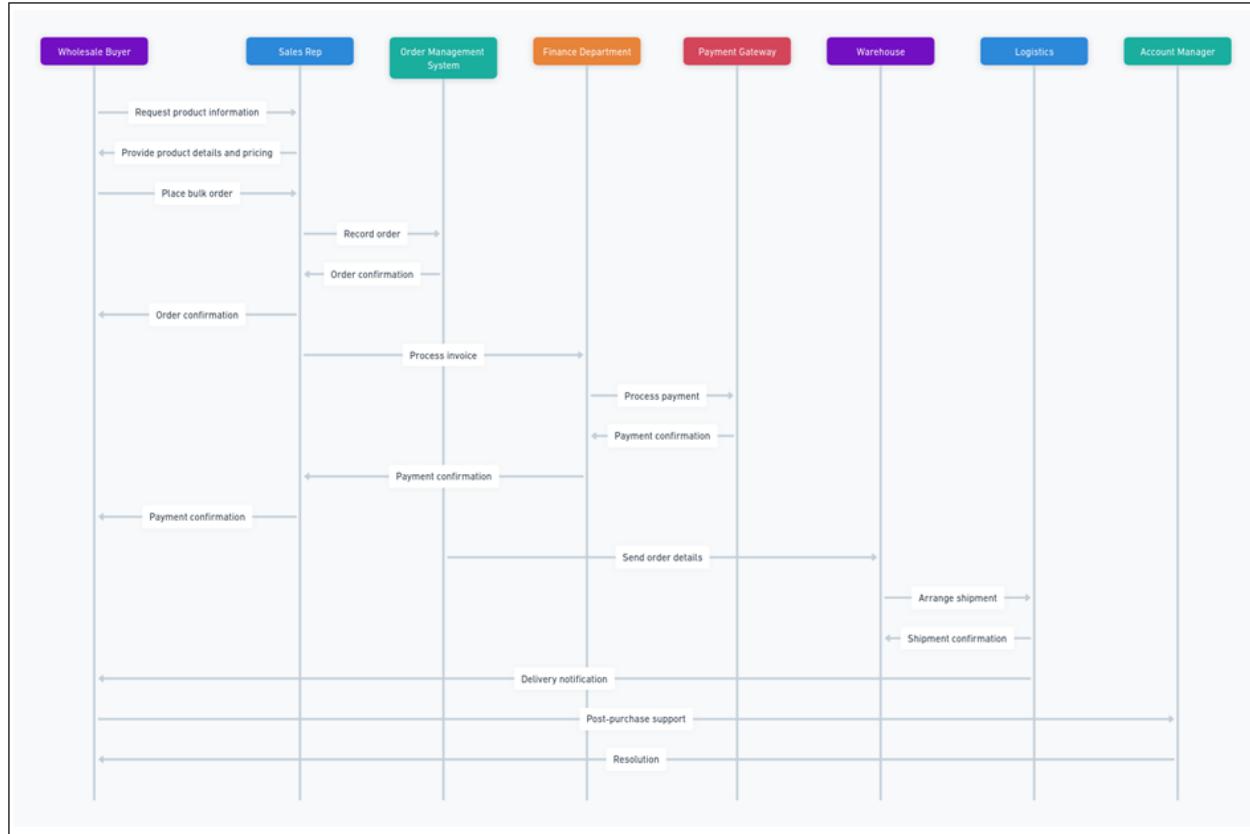


Figure 10: Sequence Diagram - Wholesale Buyer Journey

2.7.3 Supplier Journey

Suppliers become aware of opportunities with Wide World Importer through industry events, trade shows, and B2B marketing. During consideration, they use the website, engage with sales reps, and respond to RFPs, facing challenges like a complicated RFP process and unresponsive reps. Partnership negotiation involves meetings, proposals, and contracts, with potential issues in lengthy negotiations and contract complications. Onboarding includes sessions and the use of the supplier portal, where complex processes and technical issues can be problematic. Order fulfilment involves the order management system and delivery notifications, with joint pain points being complicated management and delivery issues. Post-fulfilment includes performance reviews and feedback sessions, where suppliers may experience poor feedback and a lack of long-term commitment.

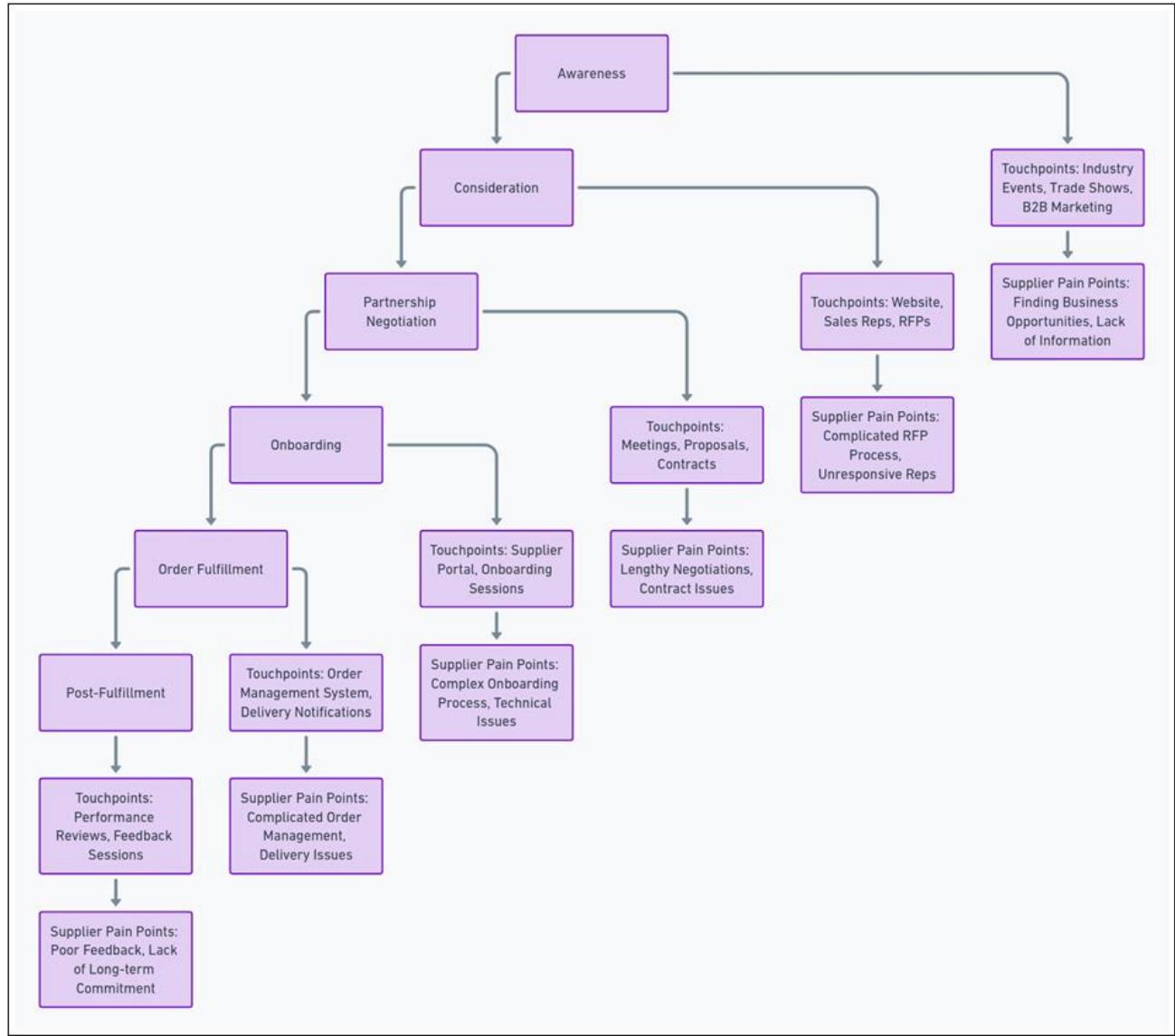


Figure 11: Supplier



Figure 12: Sequence Diagram - Supplier Journey

These user journeys highlight critical touchpoints and pain points, providing valuable insights to improve customer and supplier interactions, enhance satisfaction, and streamline processes at Wide World Importer.

3 Purchasing Functionality Analysis for WWI

The purchasing functionality within Wide World Importer is crucial for maintaining an efficient supply chain and ensuring that the organisation has the necessary products to meet customer demand. The primary role of the purchasing department is to manage relationships with suppliers, negotiate contracts, place orders, and ensure timely delivery of goods.

3.1 Interaction with Other Organisational Users

The purchasing department interacts with several other departments within the organisation to ensure smooth operations:

- **Inventory Management:** The purchasing department relies on inventory data to determine when to reorder products. Accurate and real-time inventory information is critical to avoid stockouts or overstock situations.
- **Finance Department:** Coordination with finance is essential for budgeting, processing payments, and financial reporting.
- **Warehouse and Logistics:** The purchasing team works closely with the warehouse and logistics departments to manage goods' receipt, storage, and distribution.
- **Sales and Customer Service:** These departments provide demand forecasts and customer feedback, which inform purchasing decisions.

3.2 Business Process

The purchasing process typically involves the following steps:

- **Demand Forecasting:** The purchasing department forecasts future demand based on sales data and inventory levels.
- **Supplier Selection:** Evaluate and select suppliers based on quality, price, reliability, and other criteria.
- **Order Placement:** Place orders with selected suppliers, specifying quantities, delivery schedules, and other terms.
- **Order Tracking:** Monitor the status of orders and follow up with suppliers as needed to ensure timely delivery.
- **Receiving and Inspection:** Goods are inspected for quality and quantity compliance upon arrival.
- **Payment Processing:** Coordinate with finance to process supplier invoices and payments.
- **Performance Evaluation:** Regularly evaluate supplier performance and adjust procurement strategies accordingly.

3.3 Critical Information for Purchasing Functionality

- **Inventory Levels:** Real-time data on current stock levels.

- **Demand Forecasts:** Projected sales data to anticipate future needs.
- **Supplier Information:** Details on suppliers, including contact information, pricing, lead times, and performance history.
- **Order Status:** Updates on the status of placed orders, including expected delivery dates.
- **Financial Data:** Budget allocations, cost analyses, and payment statuses.
- **Quality Reports:** Data from quality inspections of received goods.

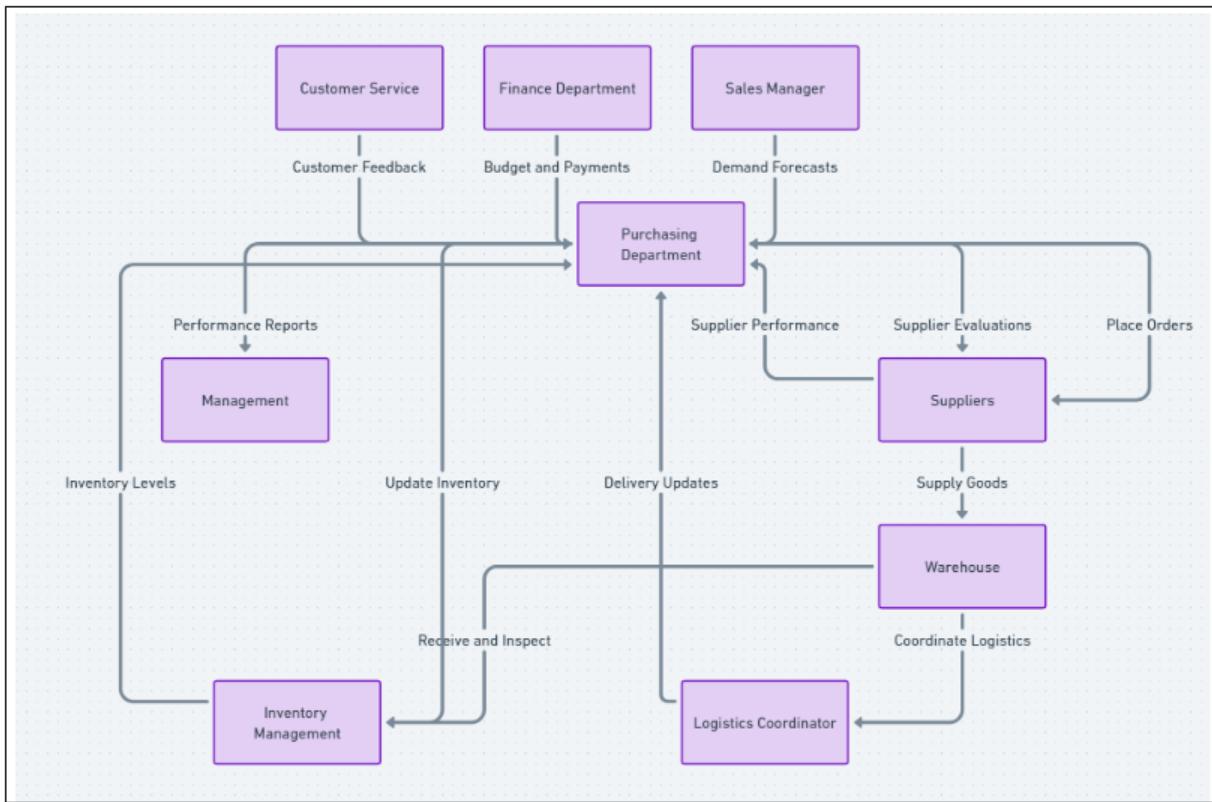


Figure 13: Information Flow of Purchasing Functionality

The purchasing functionality at Wide World Importer is integral to ensuring the company maintains an optimal inventory level to meet customer demand without incurring unnecessary costs. The purchasing department interacts closely with the inventory management team to get real-time updates on stock levels and with the sales team to understand demand forecasts. This collaboration is essential for placing accurate orders with suppliers.

The finance department plays a crucial role in managing budgets and processing payments for purchased goods. Once orders are placed, the purchasing department tracks these orders to ensure timely delivery, coordinating with the warehouse and logistics teams for the receipt, inspection, and distribution of goods. Any quality issues identified during the inspection are communicated to the suppliers for resolution.

Critical information flowing into the purchasing department includes inventory levels, demand forecasts, supplier quotations, quality inspection reports, and order status updates. Outbound information includes purchase orders, payment requests, supplier evaluations, inventory

replenishment data, and performance reports. Key actors involved in this process are the purchasing manager, inventory manager, finance manager, warehouse manager, logistics coordinator, sales manager, and customer service representatives.

This structured approach ensures that Wide World Importer can maintain efficient and cost-effective purchasing operations, ultimately supporting the company's ability to deliver high-quality products to its customers in a timely manner.

3.4 Purchasing Scope of Organisational Chart

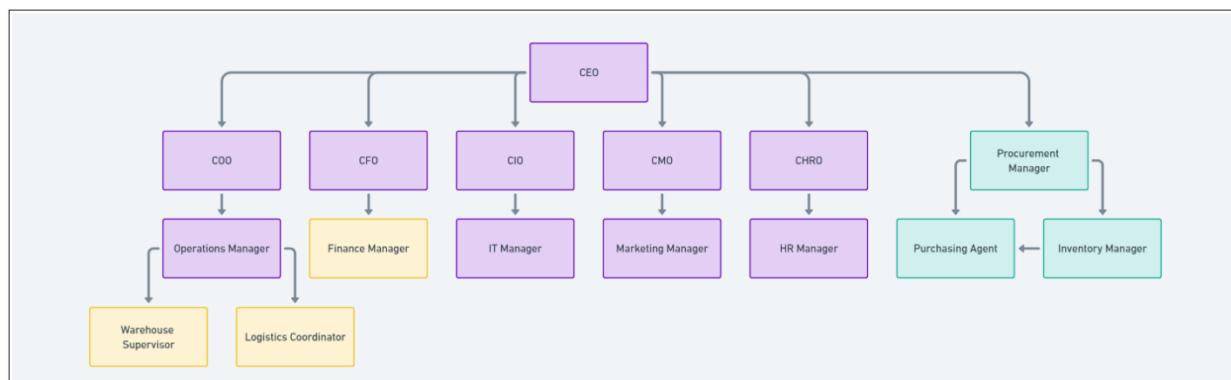


Figure 14: Scope of Purchasing Functionality

Immediate Boundaries (Highlighted in Green)

The immediate boundaries of the purchasing functionality encompass the core roles directly involved in the procurement process. These include the Purchasing Agent, Procurement Manager, and Inventory Manager. These roles are critical for ensuring purchasing operations run smoothly and efficiently.

Purchasing Agent

- **Responsibilities:** The Purchasing Agent is responsible for placing orders with suppliers, managing purchase orders, and ensuring that goods are ordered according to the company's needs.
- **Interactions:** Directly interacts with suppliers to negotiate terms, confirm orders, and handle any issues that arise during the ordering process.

Procurement Manager

- **Responsibilities:** The Procurement Manager oversees the entire procurement process, including supplier selection, contract negotiation, and procurement strategy development.
- **Interactions:** Works closely with the Purchasing Agent to ensure procurement activities align with the company's strategic goals and interacts with the Inventory Manager to align purchasing with inventory needs.

Inventory Manager

- **Responsibilities:** The Inventory Manager monitors stock levels, manages inventory records, and ensures that inventory levels meet demand without resulting in overstock or stockouts.
- **Interactions:** Provides critical inventory data to the Procurement Manager and Purchasing Agent, helping them make informed purchasing decisions.

Next Level (Highlighted in Gold)

The next level of interaction includes departments that provide essential support to the purchasing functionality by supplying necessary information and handling logistics and financial transactions. These departments ensure the smooth flow of goods and information within the organisation.

Finance Manager

- **Responsibilities:** The Finance Manager handles budgeting, payment processing, and financial reporting related to procurement activities.
- **Interactions:** Works with the Procurement Manager and Purchasing Agent to manage budgets, process supplier payments, and ensure financial accuracy in purchasing transactions.

Warehouse Supervisor

- **Responsibilities:** The warehouse supervisor oversees the receipt, inspection, and storage of goods suppliers deliver.
- **Interactions:** Coordinates with the Purchasing Agent and Inventory Manager to verify the quality and quantity of received goods and provides feedback on any discrepancies or issues.

Logistics Coordinator

- **Responsibilities:** Manages the transportation and distribution of goods from suppliers to the warehouse and within the company.
- **Interactions:** Ensures timely delivery of goods ordered by the Purchasing Agent and coordinates with the Warehouse Supervisor for smooth logistics operations.

No Interaction (Highlighted in Purple)

The following roles have no direct involvement in the purchasing process but may have indirect influences or receive reports on purchasing performance.

- Chief Operating Officer (COO)
- Chief Financial Officer (CFO)
- Chief Information Officer (CIO)
- Chief Marketing Officer (CMO)
- Chief Human Resources Officer (CHRO)
- Operations Manager
- IT Manager
- Marketing Manager
- HR Manager

3.5 Detailed Interaction from the Purchasing Department

3.5.1 Consideration Stage

Inventory Management: The purchasing department must ensure that the products displayed on the website and in stores are available. This involves maintaining up-to-date inventory records and forecasting demand to prevent stockouts.

3.5.2 Purchase Stage

Order Fulfilment: The purchasing department plays a critical role in ensuring that there is enough stock to fulfil customer orders. They must work closely with suppliers to manage lead times and ensure timely delivery of goods to the warehouse.

3.5.3 Delivery Stage

Stock Availability: Ensuring that products are available and ready for shipment is a key responsibility of the purchasing department. They coordinate with the logistics team to ensure products are picked, packed, and shipped according to customer orders.

3.5.4 Post-Purchase Stage

Quality Control: The purchasing department must address any feedback related to product quality. This involves working with suppliers to resolve issues and ensuring that corrective actions are taken to improve product quality in future orders.

3.6 Risk Analysis

A comprehensive risk analysis is essential for Wide World Importers (WWI) to identify potential threats and develop strategies to mitigate these risks. By systematically evaluating operational, financial, market, and strategic risks, WWI can ensure business continuity and safeguard against adverse impacts.

3.6.1 Operational Risks:

- **Supply Chain Disruptions:** Dependence on international suppliers can lead to vulnerabilities such as delays, political instability, and natural disasters that disrupt the supply chain.
- **Inventory Management:** Inefficient inventory management can result in stockouts or overstock situations, impacting sales and increasing holding costs.
- **Logistics Challenges:** Transportation issues, including delays and increased shipping costs, can affect the timely delivery of goods to customers.

3.6.2 Financial Risks:

- **Currency Fluctuations:** As WWI engages in international trade, fluctuations in exchange rates can impact profit margins and financial stability.
- **Credit Risk:** The possibility of customers defaulting on payments risks cash flow and financial health.
- **Cost Inflation:** Rising costs of goods, labour, and logistics can erode profit margins if not managed effectively.

3.6.3 Market Risks:

- **Competitive Pressure:** Intense competition from other importers and distributors can affect market share and profitability.
- **Changing Customer Preferences:** Shifts in consumer behaviour and preferences may render certain products obsolete, requiring continuous adaptation and innovation.
- **Economic Downturns:** Economic recessions or slowdowns can reduce consumer spending, impacting sales and revenue.

3.6.4 Strategic Risks:

- **Regulatory Changes:** New trade policies, tariffs, and regulatory requirements can increase operational costs and complicate compliance.

- **Technological Advancements:** Failure to adopt modern technologies can result in inefficiencies and loss of competitive advantage.
- **Reputation Risk:** Negative publicity or failure to meet customer expectations can damage the company's reputation and brand value.

3.6.5 Mitigation Strategies

To address these identified risks, WWI should implement the following mitigation strategies:

3.6.5.1 *Operational Risk Mitigation:*

- **Diversify Suppliers:** Reduce dependence on any single supplier by establishing relationships with multiple suppliers from different regions.
- **Inventory Optimisation:** Implement advanced inventory management systems to maintain optimal stock levels and improve forecasting accuracy.
- **Logistics Planning:** To ensure reliable delivery, develop contingency plans for logistics, including alternative routes and carriers.

3.6.5.2 *Financial Risk Mitigation:*

- **Hedging Strategies:** Use financial instruments to hedge against currency fluctuations and minimise exchange rate risks.
- **Credit Management:** Conduct thorough credit assessments of customers and implement stringent credit control policies to reduce the risk of defaults.
- **Cost Control:** Continuously monitor and manage operational costs through efficiency improvements and cost-saving initiatives.

3.6.5.3 *Market Risk Mitigation:*

- **Competitive Analysis:** Regularly analyse competitor activities and market trends to stay ahead and identify new opportunities.
- **Customer Insights:** Conduct regular market research to understand changing customer preferences and adapt product offerings accordingly.
- **Economic Monitoring:** Keep abreast of economic indicators and adjust business strategies to mitigate the impact of economic downturns.

3.6.5.4 *Strategic Risk Mitigation:*

- **Regulatory Compliance:** Stay informed about regulatory changes and engage with legal experts to ensure compliance with new requirements.

- **Technology Investment:** Invest in modern technologies and innovation to enhance operational efficiency and maintain competitive advantage.
- **Reputation Management:** Develop a robust reputation management strategy to protect and enhance the company's brand image, including effective communication and customer service.

By implementing these mitigation strategies, WWI can proactively manage risks and ensure resilience against potential challenges. This proactive approach helps safeguard the company's operations, financial health, and long-term strategic goals.

3.7 Strategic Planning

Strategic planning is essential for guiding Wide World Importers (WWI) towards achieving its business objectives and sustaining long-term growth. This process involves developing clear strategies, creating an actionable implementation plan, and ensuring continuous monitoring and improvement. To achieve its business objectives, WWI should focus on the following strategic areas:

3.7.1 Market Expansion:

- **Geographic Expansion:** Identify and enter new geographic markets domestically and internationally to increase market share and revenue. This may involve setting up distribution centres strategically to enhance delivery efficiency and reduce shipping costs.
- **Customer Segmentation:** Expand into new customer segments, such as targeting corporate clients with bulk purchasing needs or catering to niche markets with specialised products.

3.7.2 Product Diversification:

- **New Product Development:** Invest in research and development to introduce new products that meet emerging customer needs and preferences. This can help maintain a competitive edge and attract a broader customer base.
- **Product Line Expansion:** Expand existing product lines to include variations and complementary products, providing customers with a broader range of options.

3.7.3 Operational Efficiency:

- **Process Optimisation:** Implement lean management practices and continuous improvement initiatives to streamline operations, reduce waste, and enhance productivity. This includes optimising the procurement process, inventory management, and logistics.

- **Technology Integration:** Invest in advanced technologies like automation and data analytics to improve operational efficiency and decision-making processes. Leveraging technology can also enhance customer experience through better service and faster delivery.

3.7.4 Customer Engagement:

- **Enhanced Customer Service:** Develop a customer-centric approach by improving customer service capabilities, including faster response times, personalised service, and effective issue resolution.
- **Loyalty Programs:** Implement and enhance loyalty programs to increase customer retention and encourage repeat business. Personalised offers and rewards can help build long-term relationships with customers.

3.7.5 Sustainability Initiatives:

- **Green Practices:** Adopt sustainable business practices to reduce environmental impact and appeal to environmentally conscious customers. This includes optimising supply chain processes, reducing waste, and using eco-friendly packaging.
- **Corporate Social Responsibility (CSR):** Engage in CSR activities that benefit the community and enhance the company's reputation as a responsible corporate citizen.

3.8 Implementation Plan

Creating an actionable implementation plan involves outlining the steps to execute the strategies, assigning responsibilities, and setting timelines. The implementation plan for WWI should include:

Action Steps:

Detailed actions are required to achieve each strategic goal, such as market research for expansion, product development milestones, and process optimisation initiatives.

Responsibilities:

Assigning specific tasks to relevant departments and individuals, ensuring accountability and clarity in roles. For instance, the marketing team may be responsible for market research and new product launches, while the operations team focuses on process improvements.

Timelines:

Setting realistic timelines for each action step, with clear milestones and deadlines. This helps track progress and ensure the timely completion of strategic initiatives.

Resource Allocation:

Allocating necessary resources, including budget, personnel, and technology, to support the execution of the strategies. Effective resource management ensures that all initiatives have the support needed for successful implementation.

3.9 Continuous Monitoring and Improvement

Continuous monitoring and improvement are vital for Wide World Importers (WWI) to maintain alignment with its strategic goals, adapt to changing market conditions, and ensure sustained growth and operational excellence. This process involves regular performance reviews, establishing a feedback loop, and maintaining adaptability in strategies and operations.

3.9.1 Regular Reviews

Regular performance reviews are essential for tracking progress against business objectives and key performance indicators (KPIs). These reviews help WWI assess its strategies' effectiveness and make data-driven decisions for ongoing improvements.

Performance Reviews:

- Schedule monthly, quarterly, and annual performance reviews to evaluate the progress of strategic initiatives.
- Measure success using performance metrics and KPIs, such as sales growth, inventory turnover, customer satisfaction, and cost savings.
- Compare actual performance with targets to identify any deviations and analyse the reasons behind them.
- Prepare comprehensive performance reports for management, highlighting key achievements, challenges, and areas needing attention.

Financial Reviews:

- Conduct regular financial reviews to monitor the company's financial health, including profit margins, ROI, and cash flow.
- Analyse financial statements and budget variances to ensure fiscal responsibility and identify cost-saving opportunities.
- Adjust financial strategies based on insights from these reviews to optimise financial performance.

Operational Reviews:

- Perform operational audits to assess the efficiency of core processes, such as procurement, inventory management, and order fulfilment.

- Identify operational bottlenecks and inefficiencies and implement corrective actions to enhance productivity and reduce costs.
- Review supplier performance regularly to ensure reliability and quality in the supply chain.

3.9.2 Feedback Loop

Establishing a robust feedback loop is crucial for incorporating stakeholder input and improving continuously. Customer, employee, and partner feedback provides valuable insights into areas needing enhancement.

Customer Feedback:

- Implement systematic methods for collecting customer feedback, such as surveys, reviews, and direct interactions.
- Analyse feedback to understand customer satisfaction, preferences, and pain points.
- Use insights from feedback to improve products, services, and customer support, ensuring a customer-centric approach.

Employee Feedback:

- Encourage regular employee feedback through surveys, suggestion boxes, and performance reviews.
- Create an open environment where employees feel valued and heard, fostering a culture of continuous improvement.
- Address employee concerns and implement suggestions to improve workplace satisfaction and productivity.

Partner Feedback:

- Engage with suppliers and other business partners to gather feedback on business processes and relationships.
- Use this feedback to enhance collaboration, streamline processes, and ensure mutually beneficial partnerships.

3.9.3 Adaptability

Maintaining adaptability is essential for WWI to respond effectively to dynamic market conditions, emerging trends, and unforeseen challenges. An agile approach allows the company to pivot strategies and operations to stay competitive.

Market Adaptation:

- Monitor market trends, customer behaviour, and competitive actions to identify new opportunities and threats.
- Adjust marketing strategies, product offerings, and sales approaches based on real-time market data.

Operational Flexibility:

- Implement flexible operational practices that can be scaled up or down based on demand fluctuations.
- Leverage technology and automation to quickly adapt to changes in the supply chain and production processes.

Strategic Flexibility:

- Regularly review and update strategic plans to reflect changes in the business environment.
- Ensure that strategic goals align with the company's vision and market realities.

WWI can ensure continuous improvement and sustained growth by conducting regular reviews, establishing a feedback loop, and maintaining adaptability. This proactive approach enables the company to stay ahead of the competition, meet evolving customer needs, and achieve long-term success.

3.10 Conclusion

The comprehensive analysis of Wide World Importers (WWI) underscores the intricate dynamics of a modern wholesale enterprise. A holistic understanding of WWI's strategic and operational landscape has been developed through meticulous examination of the business model, stakeholder interactions, organisational structure, market positioning, financial health, and operational processes.

Wide World Importers operates within a complex ecosystem, balancing relationships with suppliers, customers, and internal departments to ensure efficient and effective business operations. The analysis highlighted key strengths, such as robust supplier relationships and an efficient logistics network, which provide a solid foundation for the company's competitive advantage. However, the examination also revealed critical weaknesses, such as dependence on international suppliers and high operational costs, potentially impeding the company's performance.

3.10.1 Strategic Context

The strategic context of WWI involves a multi-faceted approach to business growth and sustainability. The business model canvas provided insight into the core activities, resources, and value propositions that define WWI. Key objectives were identified, including increasing sales, improving customer satisfaction, reducing costs, and expanding market reach. These objectives form the basis of WWI's strategic initiatives to drive growth and enhance operational efficiency.

3.10.2 Stakeholder Analysis

The stakeholder analysis illuminated WWI's stakeholders' diverse needs and expectations, including customers, suppliers, employees, and investors. Understanding these needs is crucial for maintaining strong relationships and ensuring the alignment of business strategies with stakeholder expectations. Customers demand high-quality products and excellent service, while suppliers seek reliable partnerships and clear communication. Employees value job security and career development opportunities, and investors focus on financial performance and strategic growth.

3.10.3 Organisational Structure

The organisational structure of WWI is designed to support its strategic objectives through clear lines of communication and defined roles and responsibilities. The detailed organisational chart and role descriptions provided insight into how various departments and functions interact to achieve the company's goals. Key roles such as the CEO, COO, CFO, CIO, CMO, and CHRO are critical in steering the company's strategic direction, operational efficiency, and market presence.

3.10.4 SWOT Analysis

The SWOT analysis thoroughly evaluated WWI's internal strengths and weaknesses, as well as external opportunities and threats. Strengths such as a diverse product range and an experienced management team provide a competitive edge. In contrast, weaknesses like limited online presence and vulnerability to supply chain disruptions present challenges that must be addressed. Opportunities for market expansion, adoption of e-commerce platforms, and strategic partnerships can drive growth, while threats from economic fluctuations, regulatory changes, and intense competition must be mitigated through initiative-taking strategies.

3.10.5 Process Flow and Optimisation

Mapping core processes such as procurement, sales, order fulfilment, and customer service helped identify bottlenecks and areas for improvement. Streamlining these processes through technological integration and optimisation can enhance efficiency and reduce costs. The procurement process is critical for maintaining inventory levels and ensuring timely delivery of products. WWI can improve operational performance and customer satisfaction by addressing bottlenecks and leveraging optimisation opportunities.

3.10.6 Market and Financial Analysis

The market analysis underscored the importance of understanding customer segments, the competitive landscape, and market trends. Tailoring products and services to meet the specific needs of different customer groups can enhance market penetration and customer loyalty. Competitive analysis helps WWI stay ahead of rivals by adapting to market trends and leveraging its strengths.

Financial analysis provided insight into revenue streams, cost structures, and profitability. WWI can identify cost-saving opportunities and enhance financial stability by evaluating financial performance through metrics such as profit margins, ROI, and financial ratios. Effective fiscal management ensures that resources are allocated efficiently to support strategic initiatives and achieve long-term success.

3.10.7 Risk Management and Strategic Planning

Identifying and mitigating risks is essential for sustaining business operations and achieving strategic goals. The risk analysis highlighted potential operational, financial, market, and strategic risks and mitigation strategies to address these threats. Diversifying suppliers, optimising inventory management, and implementing hedging strategies are critical for managing operational and financial risks. Staying informed about regulatory changes and investing in technology can mitigate strategic risks, while continuous market analysis helps navigate economic fluctuations and competitive pressures.

Strategic planning involves developing actionable strategies and implementation plans to achieve business objectives. Focusing on market expansion, product diversification, operational efficiency, customer engagement, and sustainability initiatives can drive growth and enhance market presence. Regular performance reviews, feedback loops, and adaptability ensure that WWI remains aligned with its strategic goals and responsive to changing market conditions.

3.10.8 Continuous Improvement

Continuous monitoring and improvement are vital for maintaining operational excellence and achieving sustained growth. Regular performance and financial reviews and robust feedback mechanisms enable WWI to make data-driven decisions and implement necessary adjustments. Encouraging feedback from customers, employees, and partners fosters a culture of continuous improvement and innovation.

In conclusion, the comprehensive business analysis of Wide World Importers has provided valuable insights into the company's strategic and operational dynamics. By leveraging its strengths, addressing weaknesses, capitalising on opportunities, and mitigating threats, WWI can achieve its business objectives and sustain long-term growth. Continuous improvement through regular monitoring, feedback, and adaptability will ensure that WWI remains competitive and responsive to market changes, securing its position as a leading player in the wholesale industry.

4 KPIs and Data-Driven Infrastructure for the Purchasing

Defining and monitoring Key Performance Indicators (KPIs) are pivotal processes for organisations striving to measure and enhance their operational performance. At Wild World Importers (WWI), analysing financial data through well-defined KPIs is crucial. These indicators serve as quantifiable measures that reflect the success and progress towards achieving specific business objectives.

To effectively implement KPIs, WWI must establish specific metrics aligning with its strategic goals. For example, sales growth, inventory turnover, and customer satisfaction are critical metrics that can indicate the health and efficiency of the business. By setting these specific KPIs, WWI can focus on key areas of performance that directly impact its success. Each KPI should be tailored to reflect WWI's unique operational environment and strategic aims, ensuring the relevant measures drive the appropriate behaviours and outcomes.

Benchmarking these KPIs against industry standards and best practices is another essential step. This involves comparing WWI's performance with peers and industry leaders to identify performance gaps and areas for improvement. Benchmarking not only helps in setting realistic performance targets but also in adopting industry best practices that can enhance operational efficiency and competitiveness.

Regular review of these KPIs is vital for WWI to continually assess the effectiveness of its strategies and operational tactics. By monitoring these metrics regularly, WWI can make informed, data-driven decisions to refine its operations and respond to market dynamics. This ongoing evaluation helps maintain agility and ensures sustained improvements, adjusting strategies to meet changing conditions and goals.

Correct reporting and KPI management are particularly beneficial in the context of the Purchasing department at WWI. Effective KPIs for this department might include measures of cost savings, supplier performance, procurement cycle times, and contract compliance. The purchasing department can optimise its processes by focusing on these specific metrics, negotiating better terms with suppliers, and contributing significantly to cost management and overall organisational efficiency. Regular monitoring and reporting on these KPIs highlight areas of success and pinpoint opportunities for cost savings and efficiency gains, leading to more strategic purchasing decisions and improved departmental performance. This structured approach to defining and reviewing KPIs ensures that the Purchasing department aligns closely with WWI's broader strategic goals, driving value and efficiency across the organisation.

4.1 KPIs and Reporting Requirements for the Purchasing Department

A comprehensive dashboard for the purchasing department should include key performance indicators (KPIs) and reporting elements that provide insights into the efficiency, effectiveness, and strategic alignment of the procurement process. Here are the essential KPIs and reporting requirements:

4.2 KPI Dashboard

Several business intelligence tools can be considered alongside Microsoft Power BI to monitor KPIs in the purchasing department. Some popular alternatives include Tableau, Qlik Sense, Google Data Studio, and Looker. Each tool offers unique features, but Power BI is often preferred for several reasons:

1. Integration with Microsoft Products:

Power BI integrates seamlessly with other Microsoft products like Excel, Azure, SQL Server and SharePoint. This integration allows for a smoother workflow for businesses already embedded in the Microsoft ecosystem, making it easier to manage and analyse data across platforms.

2. Community and Support:

Power BI benefits from a large, active community. There are numerous forums, user groups, and resources available online. Microsoft's extensive network of developers continually contributes to the platform's development and problem-solving resources, making finding solutions and learning best practices easier.

3. Cost-Effectiveness:

Power BI provides a robust free version that is highly functional for individuals and small teams, making it accessible for study and initial development. The Pro and Premium plans are competitively priced for larger enterprises or advanced features, offering scalability and additional capabilities without a significant initial investment.

4. User-Friendly Interface:

Power BI is recognised for its user-friendly interface, particularly friendly to users familiar with Excel. It offers a short learning curve, with drag-and-drop features and an intuitive design, making it accessible to users of all skill levels.

5. Advanced Analytics and Customization:

Power BI supports advanced analytics capabilities, including AI-powered features, custom visualisations, and the ability to handle large datasets effectively. Users can also create customised dashboards and reports that meet specific organisational needs.

6. Regular Updates and Improvements:

Microsoft updates Power BI frequently, constantly adding new features and improving existing ones. This commitment to continuous improvement ensures that the tool remains at the cutting edge of technology and user needs.

Choosing Power BI to monitor KPIs in the Purchasing department leverages these advantages, ensuring robust data handling, seamless integration with existing systems, and a supportive community for ongoing development and troubleshooting.

4.3 Purchasing Dashboard Overview

The purchasing dashboard provides a comprehensive view of procurement performance, helping organizations monitor and optimize their purchasing processes. It focuses on key performance indicators (KPIs) and metrics that are critical for evaluating supplier performance, delivery efficiency, and overall procurement effectiveness.

4.3.1 KPIs

In purchasing, KPIs measure the efficiency and effectiveness of the procurement process. Examples include "Percentage of On-Time PO Received" and "Average PO Delivery Days." These indicators help assess supplier reliability and delivery performance, ensuring procurement aligns with organizational goals.

1. **Timely Delivered POs :**

This KPI tracks the total number of purchase orders (POs) that were delivered on or before the expected delivery date. It serves as a measure of supplier reliability and efficiency in meeting delivery commitments. A high number indicates that suppliers are consistently delivering products on time, which is essential for maintaining smooth operations and inventory levels.

2. **Delayed POs :**

This KPI measures the total number of purchase orders that were delivered after the expected delivery date. A high number of delayed POs can indicate issues with supplier performance or supply chain disruptions. Monitoring this KPI helps organizations identify problematic suppliers and take corrective actions to improve delivery timelines.

3. **Timely vs Delayed PO Ratio:**

This ratio compares the number of timely delivered POs to delayed POs. It provides a quick overview of the overall delivery performance. A higher ratio indicates that more orders are delivered on time compared to those that are delayed, reflecting positively on supplier performance and procurement efficiency.

4. **Average PO Delivered Days:**

This metric calculates the average number of days it takes for purchase orders to be delivered. It is crucial for understanding the lead times associated with different suppliers and categories. Shorter delivery times can lead to more agile operations and better inventory management.

5. **Total Quantity Variance:**

This KPI measures the difference between the current stock and the last stocktake, highlighting potential issues in inventory management, such as shrinkage or miscounts, which need to be addressed to maintain inventory integrity.

4.3.2 Key Metrics.

Key metrics provide detailed insights into purchasing operations. Metrics such as "Total Spend by Supplier," "Purchase Order Cycle Time," and "Quantity Variance Since Last Stocktake" offer valuable information on spending patterns, supplier performance, and inventory management. Analyzing these metrics allows organizations to identify cost-saving opportunities and improve supplier relationships.

1. Total Stock Value :

This metric represents the total value of the stock currently held by the organization. It reflects inventory investment and is essential for understanding the financial implications of inventory management. Keeping this value optimized ensures that capital is not tied up unnecessarily in excess inventory.

2. Total Procurement Spend :

This metric shows the total amount spent on procurement over a specific period. It helps organizations track their purchasing costs and identify opportunities for cost savings through better supplier negotiations or bulk purchasing strategies.

3. Total Number of POs :

This metric indicates the total number of purchase orders processed during the reporting period. It helps in assessing the procurement workload and understanding the volume of transactions handled by the purchasing team.

4. Stock at Last Stocktake:

This value indicates the quantity of stock recorded during the last stocktake. Comparing this with the current stock helps identify discrepancies and improve inventory accuracy.

5. Percentage of On-Time Deliveries :

This KPI measures the proportion of purchase orders delivered on or before the expected delivery date. It is a critical indicator of supplier performance and the efficiency of the procurement process. A higher percentage reflects well on the organization's ability to manage its suppliers effectively.

4.3.3 Operational Insights:

Operational insights derived from KPIs and metrics help organizations enhance their procurement process. For example, identifying trends in delayed deliveries or high cycle times can prompt process improvements and supplier negotiations. These insights enable the organization to proactively address issues, streamline operations, and ensure timely procurement.

- The dashboard features a breakdown of purchase orders by category and supplier, providing insights into spending patterns and supplier contributions. It helps

organizations identify top suppliers and categories, enabling better supplier relationship management and strategic sourcing decisions.

- The comparison between actual and expected delivery days for each supplier helps pinpoint areas where supplier performance may need improvement. It allows the organization to engage with suppliers to address delays and improve delivery times.
- The dashboard allows users to filter data by year, providing a temporal view of procurement trends. This enables organizations to analyze performance over time, adjust procurement strategies, and ensure alignment with business objectives.
- The comparison between total order quantity and spend helps identify cost-saving opportunities, optimize supplier selections, and enhance procurement strategies. Insights from quantity variance assist in addressing inventory management issues, ensuring more accurate stock levels and reducing discrepancies.

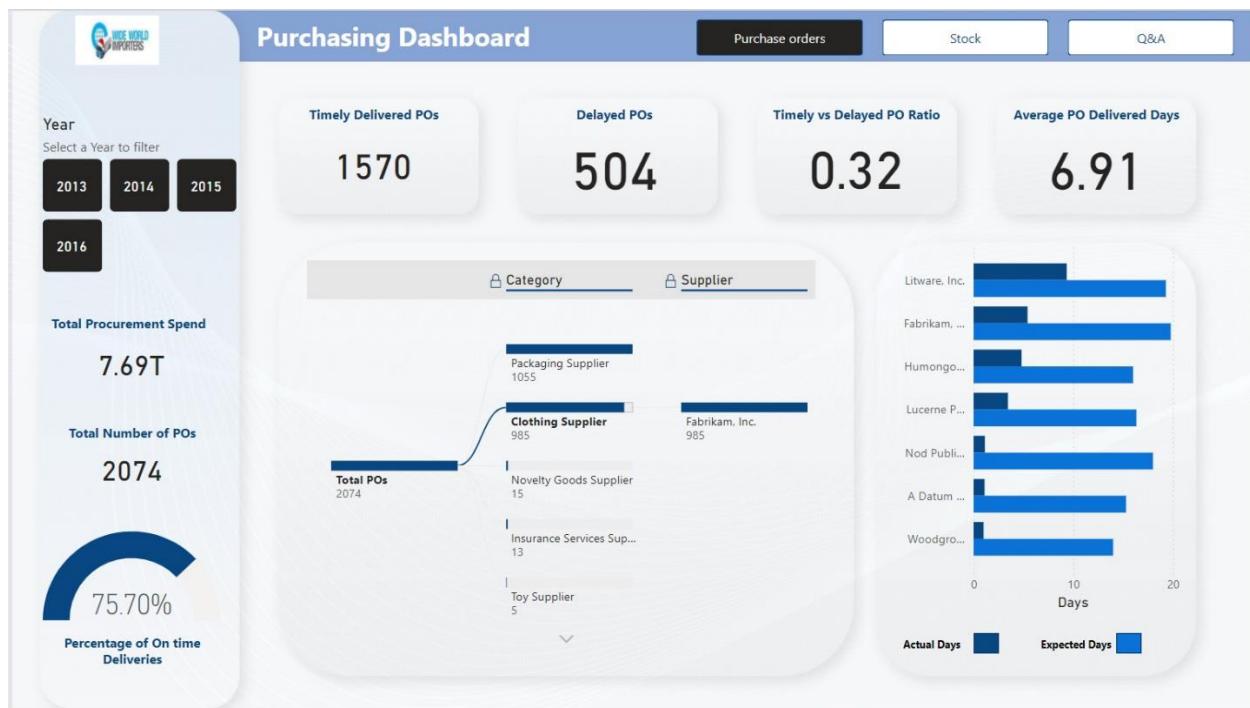


Figure 15: Purchasing Dashboard

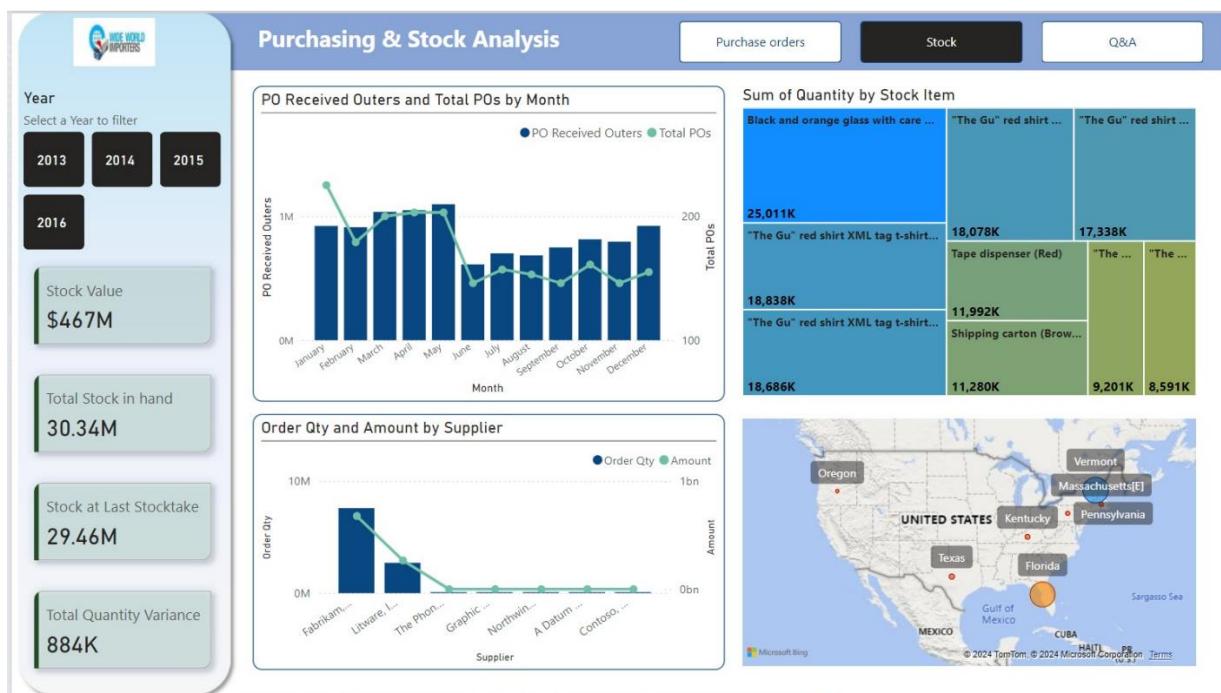


Figure 16 : Purchasing & Stock Analysis

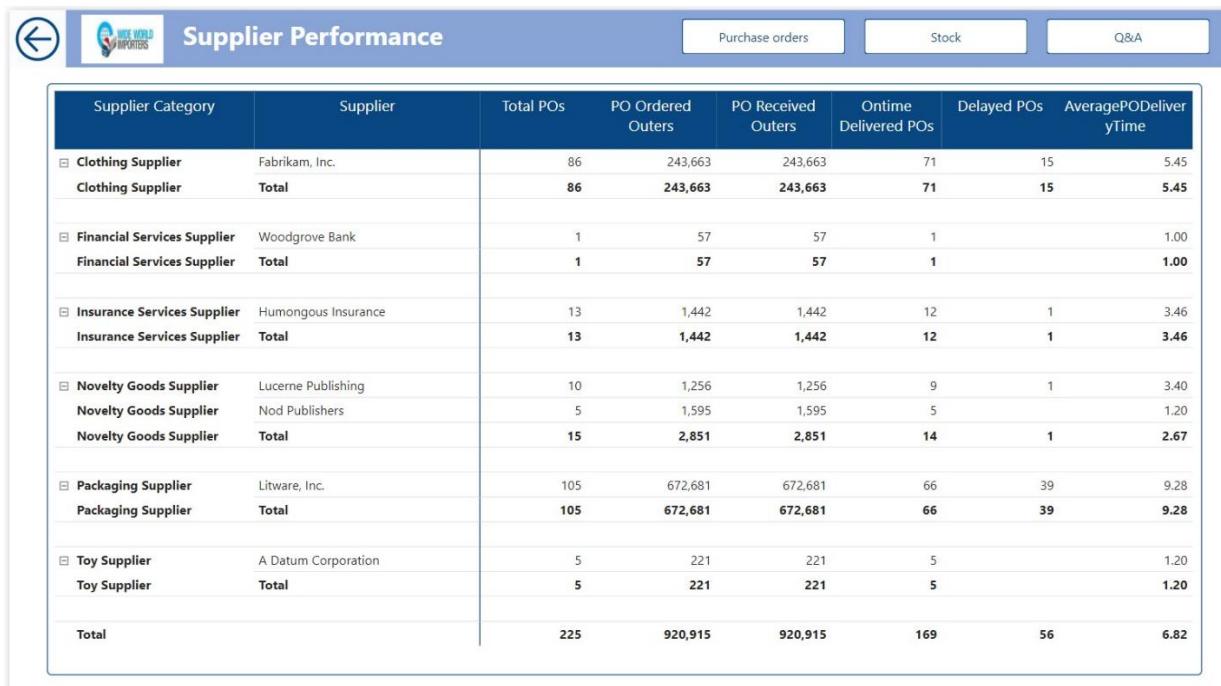


Figure 17: Drill Trough Dashboard - Supplier Performance

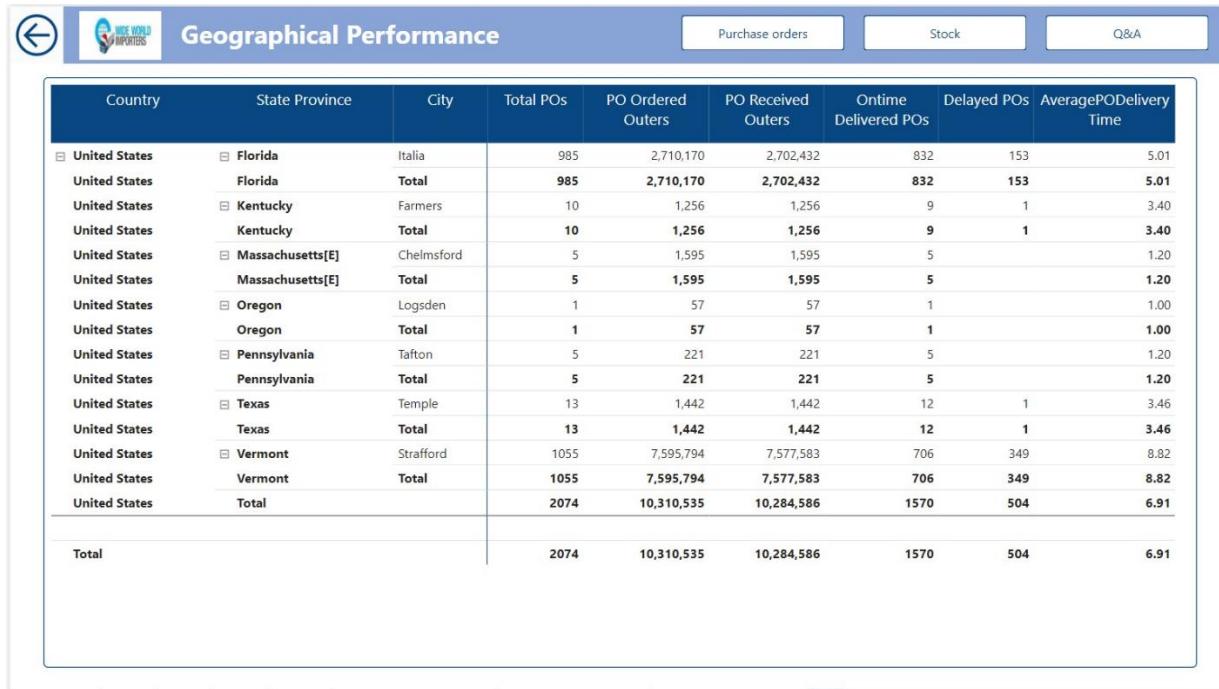


Figure 18 : Drill Through Dashboard - Geographical Performance



Figure 19: Drill Through Dashboard - Stock Group Performance

4.4 Predictive Model

4.4.1 Importance of Forecasting for Warehouse Management

Effective warehouse management offers several significant advantages that contribute to the overall success of retail operations. One notable benefit is the reduction in the cost of being out-of-stock, a critical issue highlighted by a RetailWire study which found that overstocks and out-of-stocks cost retailers \$1.1 trillion globally in lost revenue, with out-of-stocks alone accounting for 4.1% of lost revenue for the average retailer. Beyond addressing stock shortages, efficient warehouse management also improves operational efficiency by optimizing resource utilization, reducing handling time, and lowering operational costs. This leads to faster order fulfilment and higher accuracy in order processing, which in turn enhances customer service by ensuring timely and reliable delivery of products. Additionally, well-managed warehouses reduce lead times by streamlining supply chain processes, from procurement to distribution, and fostering better relationships with suppliers. These advantages collectively enable retailers to operate more profitably and maintain a competitive edge in the marketplace.

4.4.2 Data Analysis

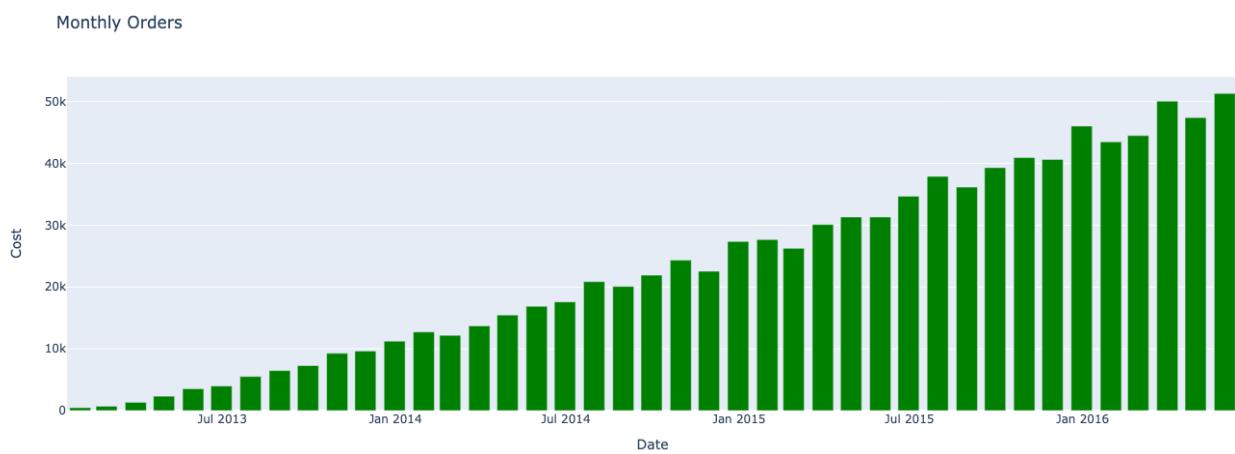


Figure 20: Last Five Years Trend

Over the last five years, purchasing has steadily increased, likely driven by rising consumer demand. In such scenarios, accurately predicting the cost of purchases becomes crucial for several reasons. First, it allows businesses to budget effectively, ensuring sufficient funds are allocated to meet increased demand without overspending. Second, accurate cost predictions help maintain optimal inventory levels, preventing overstocking or stockouts, which can lead to either tied-up capital or lost sales opportunities.

Furthermore, precise cost forecasting aids in negotiating better terms with suppliers, securing favourable prices, and managing cash flow more efficiently. It also enables businesses to plan for potential cost fluctuations, such as changes in raw material prices or transportation costs, mitigating the impact of these variables on the overall budget. By leveraging advanced predictive analytics and machine learning models, companies can analyse historical purchasing data, market trends, and other relevant factors to produce reliable cost forecasts. This enhances financial planning and operational efficiency and ensures that the supply chain remains robust and responsive to market demands, ultimately leading to improved customer satisfaction and business growth.

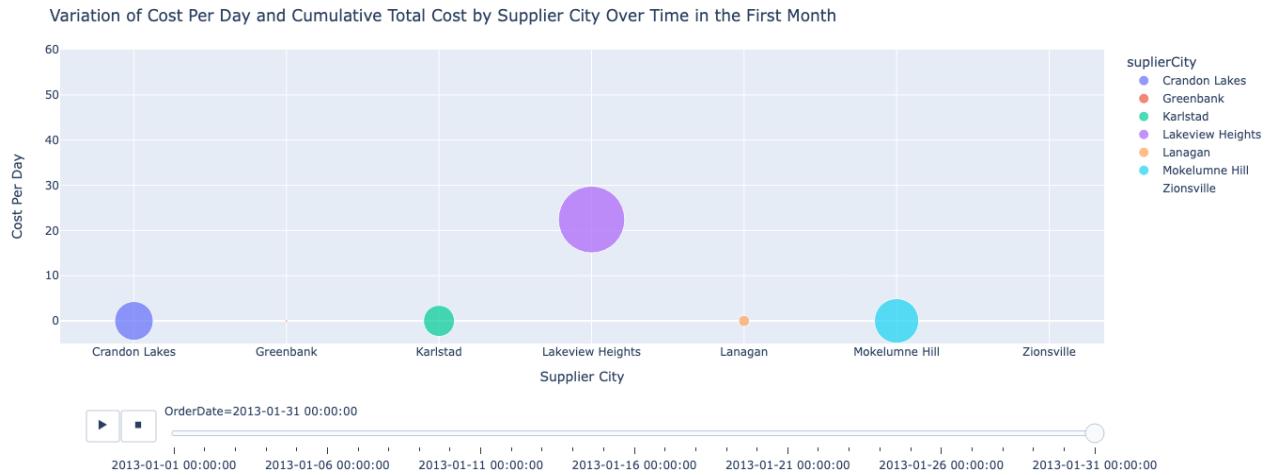


Figure 21: Purchase Volume by City

The analysis shows that Lake View Heights has consistently seen the highest volume of purchases, indicating a strong demand for products in this area. On the other hand, Greenbank and Zionsville have recorded the lowest purchase volumes, suggesting either a lower demand or potentially untapped markets that could benefit from targeted marketing strategies and sales efforts.

4.4.3 Machine Learning for Predicting Purchasing Amount

To predict the purchase amount for the next couple of weeks, we developed three time series forecasting models: Bi-LSTM, LSTM, and SARIMA. Among these, the Bi-LSTM model demonstrated the best performance compared to the other two. The evaluation metrics clearly highlight this distinction.

Evaluation Metric	LSTM	Bi-LSTM	SARIMA
Mean Absolute Error	256.7	109.82	798.12
Root-Mean-Square-Error	1257.54	861.31	2491.93
R2 Error	0.47	0.78	0.189

The Bi-LSTM model achieved a Mean Absolute Error (MAE) of 109.82, significantly lower than the LSTM's MAE of 256.7 and SARIMA's MAE of 798.12. Similarly, the Root-Mean-Square Error (RMSE) for Bi-LSTM was 861.31, outperforming the LSTM's RMSE of 1257.54 and SARIMA's RMSE of 2491.93. Furthermore, the R² Error for Bi-LSTM stood at 0.78, indicating a higher proportion of variance explained by the model, compared to 0.47 for LSTM and a mere 0.189 for SARIMA. These metrics underscore the Bi-LSTM model's enhanced accuracy and reliability in forecasting purchase amounts, making it the most effective model for our predictive needs.

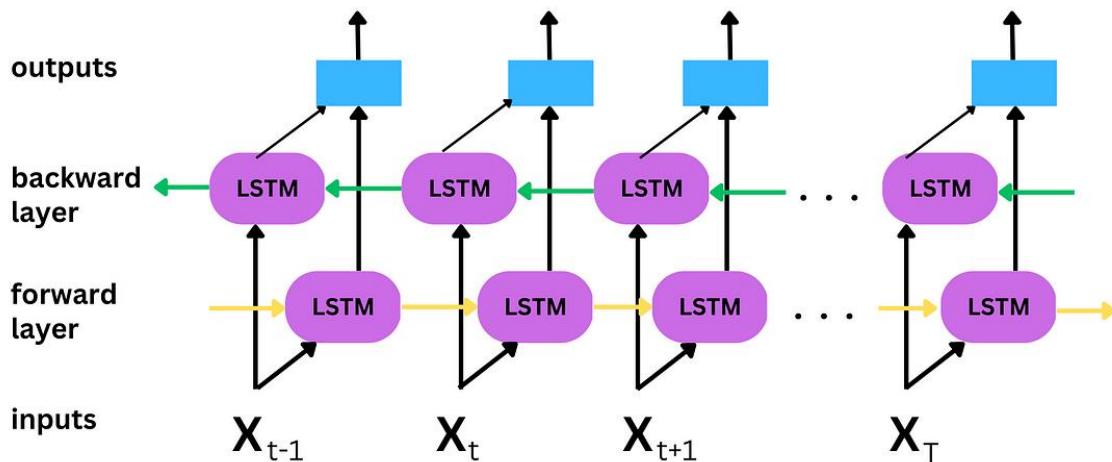


Figure 22: LSTM Architecture

The Bi-LSTM network we developed for predicting purchase amounts is composed of three bidirectional LSTM (Bi-LSTM) layers and two dropout layers. Each Bi-LSTM layer contains 120 nodes, which allows the network to effectively capture complex patterns and relationships in the time series data. The bidirectional nature of the LSTM layers enables the network to process information from both past and future contexts, enhancing its ability to make accurate predictions.

In addition to the Bi-LSTM layers, the network includes dropout layers, which are essential for preventing overfitting. By randomly deactivating a fraction of the nodes during training, the dropout layers ensure that the model generalises unseen data better. This architecture, combining the power of Bi-LSTM with the regularisation effect of dropout layers, contributes to the network's superior performance in forecasting purchase amounts, as evidenced by its lower error metrics and higher R² score compared to other models.



Figure 23: variation between the predicted and the actual purchase amounts

The diagram below illustrates the variation between the predicted purchase amounts and the actual purchase amounts over the forecasted period. It is evident from the graph that the predictions closely align with the actual values, demonstrating the model's accuracy. The Bi-LSTM network's ability to capture complex temporal patterns and relationships within the data is reflected in the minimal discrepancies between the predicted and actual amounts. This high level of precision is crucial for effective inventory and purchase planning, as it ensures that businesses can anticipate demand accurately, optimize stock levels, and minimize costs associated with overstocking or stockouts.

4.5 Data Architecture for Purchasing KPIs

Based on the KPIs and reporting requirements detailed in the document for the Purchasing department dashboard at Wild World Importers, we can create an effective entity-relationship diagram (ERD) that captures the necessary data from the OLTP database. The ERD will focus on the primary entities and their relationships essential for generating the required reports and metrics.

4.5.1 Entities and Tables to Use:

1. **PurchaseOrder:** This table is crucial as it houses all purchase order records. Key attributes include PurchaseOrderID, OrderDate, SupplierID, and TotalAmount. It supports KPIs like Total Procurement Spend, Purchase Order Cycle Time, and Order Accuracy Rate.
2. **Supplier:** This entity stores information about suppliers, including SupplierID, SupplierName, and ContactDetails. It is vital for Supplier Performance Rating, Supplier Lead Time, and Compliance Rate.
3. **PurchaseOrderDetail:** This table contains details about each purchase order, such as PurchaseOrderDetailID, PurchaseOrderID, ProductID, Quantity, and UnitPrice. It helps calculate Order Accuracy Rate, Total Procurement Spend, and Return Rate.
4. **Product:** Includes ProductID, ProductName, and StandardCost. This entity is essential for Inventory Turnover Rate and Stockout Rate.
5. **Inventory:** Holds records of inventory levels with fields like ProductID, WarehouseID, and QuantityAvailable. It supports KPIs such as Inventory Turnover Rate and Stockout Rate.
6. **Warehouse:** Contains details of each warehouse, such as WarehouseID and Location. This is important for managing inventory data.
7. **SupplierRating:** A table that might need to be created to track ratings given to suppliers based on quality, delivery, and service. Attributes include SupplierID, Rating, and RatingDate.

4.5.2 Entity-Relationship Diagram:

The ERD for the Purchasing department would typically include the following relationships:

- **PurchaseOrder to Supplier:** A many-to-one relationship, as each purchase order is associated with one supplier.
- **PurchaseOrderDetail to PurchaseOrder:** A many-to-one relationship, indicating that each purchase order can contain multiple line items.
- **PurchaseOrderDetail to Product:** A many-to-one relationship, connecting each line item in a purchase order to a specific product.

- **Product to Inventory:** A one-to-one relationship, linking each product to its inventory record.
- **Inventory to Warehouse:** A many-to-one relationship since multiple inventory records can be stored in one warehouse.

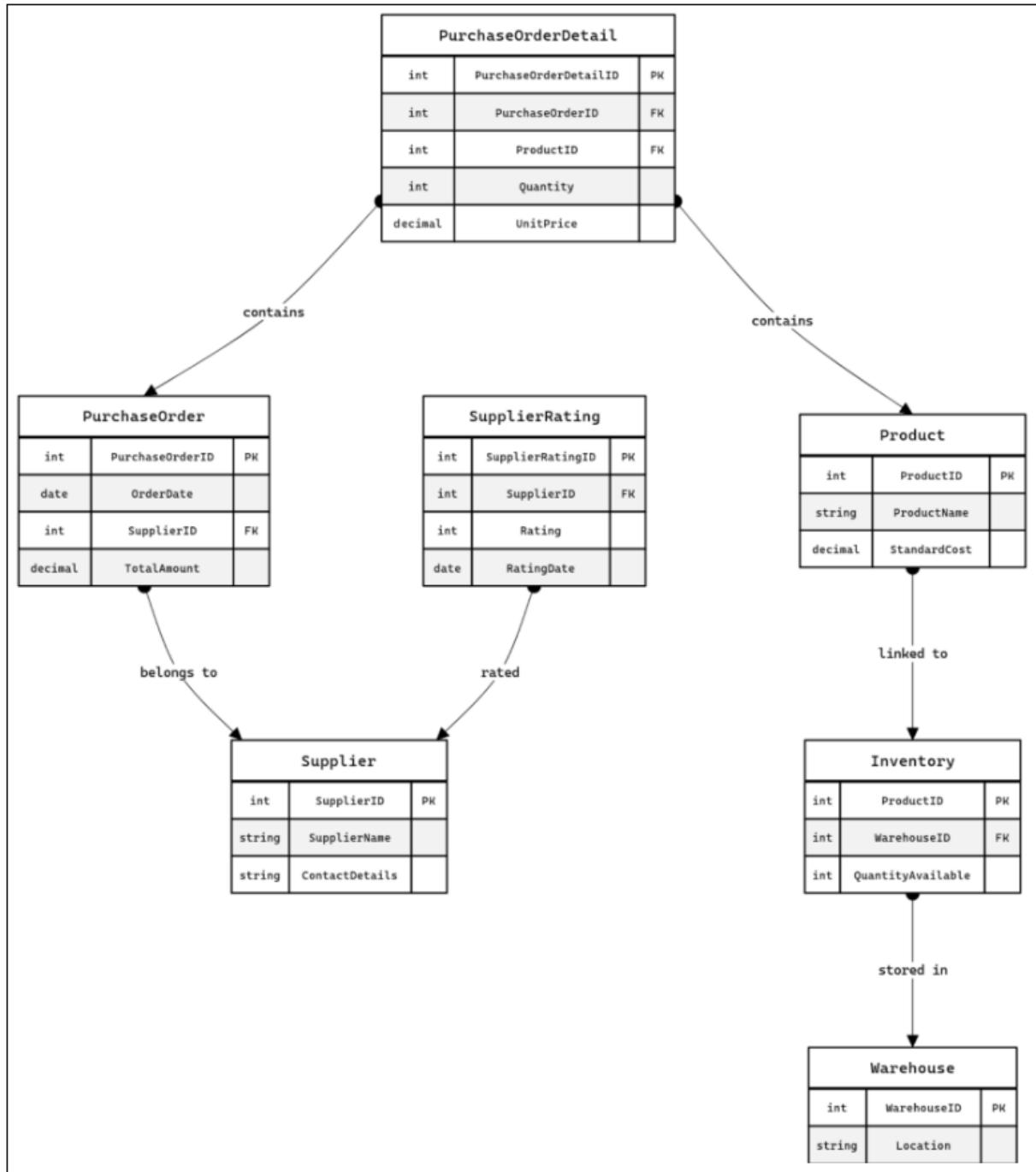


Figure 24: ERD for Purchase Functionality

This ERD will help pull together the data needed to calculate the KPIs and generate the required reports, such as monthly procurement summary, supplier performance report, inventory status report, cost analysis report, order cycle time report, and compliance audit report.

An entity-relationship diagram (ERD) plays a crucial role in designing and implementing a KPI monitoring system for the Purchasing department at Wild World Importers (WWI). The ERD serves as a visual representation of the relationships between different entities in the normalised OLTP database, helping to identify and link the necessary tables that support the required KPIs and reporting needs.

4.5.3 Understanding the ERD to the OLTP Database:

The ERD helps us understand how various tables in a normalised database relate to each other and how data flows between them. This is particularly important in a complex environment like WWI's OLTP database, which includes multiple related business functions. The tables identified through the ERD provide a structured way to access, update, and manage data efficiently while maintaining data integrity and reducing redundancy.

4.5.3.1 *Key Tables Identified from the WWI OLTP Database:*

1. **PurchaseOrders and PurchaseOrderLines:** These tables are central to tracking all details about purchase orders. ‘PurchaseOrders’ stores general information about each order, such as order date, supplier ID, and delivery method, while ‘PurchaseOrderLines’ details individual items within those orders, including product descriptions and quantities. These tables are essential for KPIs such as Total Procurement Spend and Purchase Order Cycle Time.
2. **Suppliers and SupplierTransactions:** ‘Suppliers’ contains data on each supplier, including contact details and performance metrics. ‘SupplierTransactions’ tracks transactions made with each supplier, useful for analysing Supplier Lead Time and Supplier Performance Rating.
3. **StockItems and StockItemHoldings:** These tables are vital for inventory management. ‘StockItems’ includes information on each item, such as name, color, and package type, while ‘StockItemHoldings’ provides real-time data on the quantity of each stock item available in different warehouses. This supports KPIs like Inventory Turnover Rate and Stockout Rate.
4. **StockItemTransactions:** Records transactions related to stock items, such as sales or restocking, crucial for understanding inventory flows and supporting the Stockout Rate KPI.
5. **Invoices:** Used to store invoice details for purchases, which can be linked to purchase orders and payments, essential for financial reporting and analysis.
6. **Customers, CustomerTransactions, and PaymentMethods:** While primarily related to sales, these tables can also be useful in understanding the overall financial health of WWI and its relationships with customers, which can indirectly impact purchasing strategies and decisions.

4.5.3.2 Relating ERD to Table Identification:

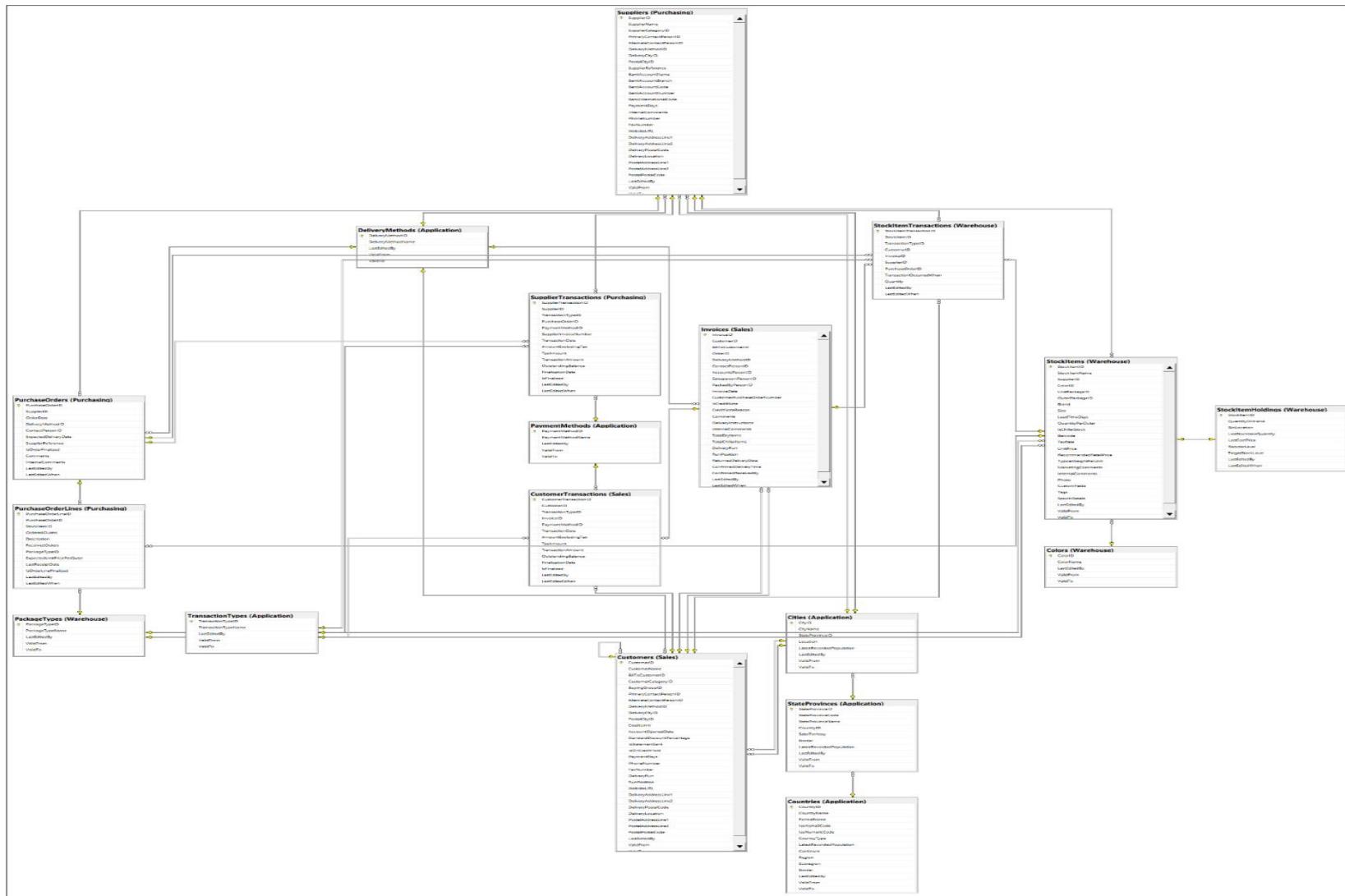


Figure 25: WWI OLTP Database for Purchasing

The ERD helps identify how these tables are interconnected through various primary key-foreign key relationships, ensuring that data pulled from one table can be accurately matched and integrated with data from others. For example, the relationship between ‘PurchaseOrders’ and ‘Suppliers’ via SupplierID ensures that each order can be traced back to its supplier, which is critical for supplier performance evaluations.

Moreover, the ERD also assists in understanding dependencies and hierarchies within the data, such as how ‘StockItems’ are linked to ‘Colors’ and ‘PackageTypes’, which are necessary for detailed inventory reports and analysis. Understanding these relationships is crucial for ensuring data integrity and executing complex queries supporting comprehensive reporting and analytics.

By utilising the ERD, WWI ensures that all relevant tables are identified and appropriately linked, forming a robust foundation for extracting, analysing, and reporting on purchasing data to support strategic decision-making and effective management of the Purchasing department’s operations.

4.6 Enhancing Business Intelligence

4.6.1 OLTP Systems and Their Limitations for Reporting

Online Transaction Processing (OLTP) systems are designed to manage day-to-day transactional operations in businesses. These systems are optimised for handling a large number of short, atomic transactions that require immediate processing and consistency. The primary focus of OLTP databases is to ensure data accuracy and quick processing times, which are essential for operational activities like order entry, financial transactions, and customer management.

4.6.1.1 *Characteristics of OLTP Systems:*

1. **High Transaction Volume:** OLTP systems are capable of handling thousands of transactions per second. This capability is crucial in environments where performance, speed, and accuracy are paramount.
2. **Complex Transactions:** Transactions in OLTP systems often involve complex queries with multiple table joins, inserts, updates, and deletes that need to be executed quickly and atomically to maintain data integrity.
3. **Normalisation:** OLTP databases typically use a highly normalised data schema to minimise redundancy and avoid data anomalies. This normalisation often extends to the third normal form (3NF), where data is separated into many smaller, related tables to reduce information duplication.
4. **Concurrency Control:** These systems are designed to support multiple users and applications accessing the database concurrently without performance degradation, ensuring that all transactions are processed reliably and without conflict.

4.6.1.2 Limitations of OLTP Systems for Reporting

While OLTP systems excel at managing transactional data, they encounter several limitations when used for reporting and analytical purposes:

1. **Complex Queries:** The highly normalised structure that benefits transaction processing becomes a hindrance when performing complex queries for reporting. The necessity to join multiple tables can severely impact the performance of analytical queries, making them slow and inefficient.
2. **Resource Intensive:** Running large, complex reports on OLTP systems can consume significant processing power and memory, potentially degrading the performance of transactional processing, which these systems are primarily built to support.
3. **Limited Historical Data:** OLTP systems are generally not designed to store historical data over long periods. They often only keep recent transaction data, which limits their usefulness for trend analysis and historical reporting.
4. **Inflexibility:** Changes to the schema for reporting purposes can be disruptive in an OLTP environment. Adding new dimensions or changing database structures can interfere with transaction processing efficiencies.
5. **Lack of Aggregation:** OLTP systems do not naturally support aggregated data storage, which is often necessary for effective reporting. Calculating aggregates on the fly from highly normalised data is resource-intensive and slow.

These limitations necessitate a different approach for organisations conducting extensive reporting and data analysis. The transition to systems specifically designed for these tasks, like Online Analytical Processing (OLAP) systems, becomes essential. OLAP systems are optimised for read-heavy operations and structured to support complex queries and large datasets, making them ideal for KPI monitoring, trend analysis, and decision support—all of which are challenging to manage efficiently with OLTP systems.

4.6.2 Transition to OLAP Systems for Enhanced Analytical Capabilities

Online Analytical Processing (OLAP) systems are designed specifically to address the limitations inherent in Online Transaction Processing (OLTP) systems, particularly in terms of reporting and analytical capabilities. OLTP systems, while excellent for handling high volumes of transactions, are not optimised for complex query processing and data analysis. This limitation is crucial in environments where decision-makers require rapid, extensive insight from large volumes of data.

OLAP systems, in contrast, are optimised for read-heavy operations that are typical in reporting, data mining, and analytical tasks. These systems enable the efficient execution of complex queries and large data sets, facilitating advanced data analysis tasks such as trend detection, forecasting, and multidimensional analysis. This capability significantly enhances business intelligence activities, supporting more informed and strategic decision-making processes.

4.6.2.1 Key Features of OLAP Systems:

1. **Advanced Data Aggregation:** OLAP systems pre-aggregate data, simplifying complex query processing. This feature allows for faster response times when accessing large volumes of data, making it easier to perform trend analysis and generate comprehensive reports.
2. **Multidimensional Analysis:** Unlike OLTP systems' flat, two-dimensional tables, OLAP allows for data storage in multidimensional cubes. This structure enables users to analyse data across multiple dimensions (e.g., time, geography, product), each providing a different perspective on the data.
3. **Ad-hoc Query Optimization:** OLAP systems are designed to handle ad-hoc queries effectively. Users can explore the data intuitively and flexibly without needing pre-defined paths or queries, which is crucial for exploratory data analysis and real-time decision-making.
4. **Scalable Performance:** As the volume of data grows, OLAP systems can scale to accommodate the increased load without significant degradation in performance. This scalability ensures that analytical capabilities remain robust even as business requirements expand.

4.6.2.2 The Role of Star Schema in OLAP

Within the architecture of OLAP systems, the star schema offers an effective method for organising data. This model is a part of the Kimball dimensional modelling methodologies and is highly regarded for its simplicity and efficiency in handling complex queries. The star schema organises data into two main types of tables: fact tables and dimension tables.

- **Fact Tables:** These tables are at the core of the star schema and store quantitative, measurable data that are essential for analysis. Common data stored in fact tables include metrics like sales revenue, cost, units sold, and profits. These tables contain keys that relate to dimension tables, facilitating the integration of descriptive data into the analysis.
- **Dimension Tables:** Surrounding the fact table, dimension tables contain descriptive attributes that provide context to the facts. For example, a product dimension table might detail attributes like product name, category, manufacturer, and price range. Each dimension table is linked to the fact table via a primary key to a foreign key relationship, enabling efficient data retrieval and slicing by various dimensions.

The star schema simplifies data access for analysts by reducing the number of joins needed in query operations, thus enhancing query performance and allowing for more straightforward data navigation and analysis. This organised structure accelerates data retrieval and intuitively aligns with business understanding, making it easier for business professionals to interact with the data without deep technical expertise.

By transitioning to OLAP systems and utilising the star schema, organisations can significantly enhance their analytical capabilities, supporting more robust business intelligence and strategic decision-making.

4.6.3 Advantages of Using a Star Schema

The star schema is a prevalent data modelling technique used in OLAP (Online Analytical Processing) data warehouses due to its simplicity and effectiveness in handling complex queries and large data sets. Below are detailed explanations of the key advantages of using a star schema in OLAP environments:

4.6.3.1 *Simplified Queries*

The star schema architecture significantly simplifies the data structure by organising data into a central fact table directly linked to surrounding dimension tables. This arrangement reduces the number of table joins required during queries. In traditional normalised databases, multiple joins across numerous tables are often necessary to compile the needed information for analysis, which can be time-consuming and computationally expensive. The star schema's straightforward design allows for easier and faster queries, enhancing the user's ability to retrieve information quickly and efficiently.

4.6.3.2 *Enhanced Performance*

The star schema, aggregations and calculations can be precomputed and stored in the fact table. This approach minimises the computational load during query execution, allowing for the rapid processing of complex analytical queries. By handling these intensive computations during the data loading phase rather than the query phase, the star schema enables analysts to access large volumes of summarised and detailed data with minimal delay. This feature is particularly beneficial in scenarios where time-sensitive decisions need to be made based on the latest data insights.

4.6.3.3 *Scalability*

The star schema is inherently scalable, making it ideal for growing organisations that continuously accumulate new data categories and measurements. New dimension tables can be easily linked to the existing central fact table without significantly disrupting the database structure or reporting processes. This flexibility ensures that the data warehouse can evolve with the business, accommodating new lines of business or changing analytical needs without extensive redesign.

4.6.3.4 *Intuitive Design*

The design of the star schema closely mirrors real-world business practices and concepts, which helps business users and analysts intuitively understand the database's layout and logic. Each dimension table represents a business entity (such as time, geography, products, or customers), and the fact table represents business events or transactions. This clear and logical structure helps users navigate the data more effectively, fostering a better understanding and quicker insight generation without requiring deep technical database knowledge.

4.6.3.5 Effective Data Segmentation

Dimension tables in a star schema allow analysts to “slice and dice” the data across multiple dimensions, facilitating robust multidimensional analysis. This capability is crucial for performing detailed business analysis and monitoring KPIs. Analysts can easily filter, group, or aggregate data along any dimension, enabling them to examine the data from various perspectives and uncover hidden patterns or trends that may not be apparent from a single-dimensional analysis.

4.6.3.6 Conclusion

While OLTP systems are indispensable for daily transaction processing, OLAP systems with star schema architectures provide the framework for effective reporting and advanced analytics. This strategic division allows organisations to maintain concurrently efficient operational systems and robust analytical platforms, supporting a comprehensive data management and decision-making approach. The star schema enhances OLAP capabilities, ensuring that businesses can leverage their data assets to the fullest extent, fostering informed strategies and competitive advantages.

4.6.4 Star Schema Design for the Purchasing Department

The OLAP star schema for the Purchasing department at Wild World Importers (WWI) is designed to streamline and enhance procurement data analysis. This schema will integrate various aspects of the purchasing process, including order details, supplier information, and inventory management. Here's how the star schema can be organised:

4.6.4.1 Fact Table: Purchase Order Facts

The central fact table in this schema will capture transactional data from the purchasing process and aggregate key numerical metrics crucial for analysis and reporting.

Attributes of the Fact Table:

- PurchaseOrderID (Link to PurchaseOrderDetail)
- OrderDate
- TotalAmount (Sum of all order line amounts, useful for calculating Total Procurement Spend)
- Number of Line Items (Count of all items in the purchase order helps in assessing order complexity)

4.6.4.2 Dimension Tables

The dimension tables provide descriptive attributes related to each dimension involved in purchasing. These tables allow for slicing and dicing the data in various ways to gain deeper insights.

Supplier Dimension

- SupplierID (Primary Key)

- SupplierName
- ContactDetails
- SupplierRating (Average rating from SupplierRating table)
- ComplianceRate (Calculated as a percentage of compliant orders)

Product Dimension

- ProductID (Primary Key)
- ProductName
- StandardCost
- Category (If available, categorise products to enhance analysis)

Inventory Dimension

- InventoryID (Primary Key, can be composed of ProductID and WarehouseID)
- CurrentQuantityAvailable
- WarehouseLocation (Link to Warehouse Dimension)

Warehouse Dimension

- WarehouseID (Primary Key)
- Location
- Capacity (If available, to analyse space utilisation)

Date Dimension (Time Intelligence)

- DateKey (Primary Key)
- FullDate
- DayOfWeek
- Month
- Year
- Quarter
- Fiscal Period (If fiscal reporting is required)

Purchase Order Detail Dimension

- PurchaseOrderDetailID (Primary Key)
- ProductID (Foreign Key, link to Product Dimension)
- UnitPrice
- QuantityOrdered
- TotalLineCost (Calculated as UnitPrice * Quantity)

4.6.4.3 Schema Diagram Layout

The schema would typically place the Purchase Order Facts table at the centre, surrounded by the dimension tables. Each dimension table is linked to the fact table via foreign keys corresponding to the primary keys in each dimension table. This layout simplifies the query process by

minimising the number of joins needed to gather insights across different dimensions, enhancing query performance and simplifying report generation.

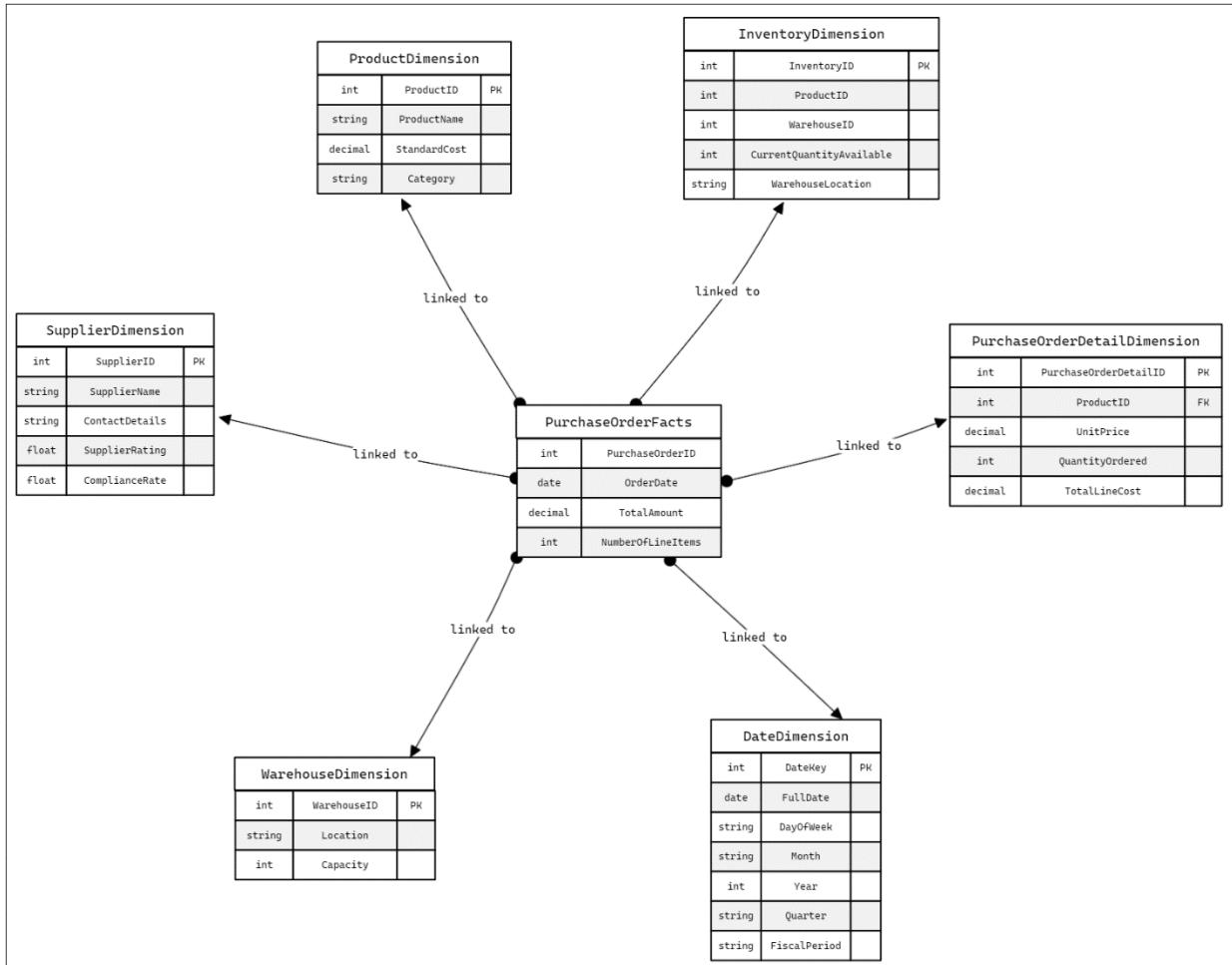


Figure 26: Star Schema for Purchase Functionality

4.6.4.4 WWI's Purchasing Department

Wild World Importers (WWI) is preparing to enhance the analytical capabilities of its Purchasing department by designing a comprehensive OLAP data warehouse. This initiative is driven by the need to support more robust decision-making processes and to accommodate future growth. Central to this effort are structuring appropriate fact and dimension tables that align with the department's operational needs while ensuring scalability and interdepartmental collaboration.

Overview of Fact Tables

The Purchasing department's data warehouse will feature a series of interconnected fact tables, each capturing different aspects of the procurement and inventory management processes. These tables include:

- Purchase:** This fact table tracks all purchase transactions, detailing the department's acquisitions of goods and services. It includes metrics such as purchase amount, quantity, and prices, directly impacting cost analysis and budgeting.
- Stock_Holding:** Essential for inventory management, this table records the current holdings within the company's inventory system. It tracks quantities of items, conditions, and locations, facilitating inventory valuation and space management.
- Movement:** Capturing the dynamics of inventory, the Movement table logs all actions that alter the quantity or location of stock items, such as restocking, transfers, and disposals. This table is crucial for tracking the efficiency of inventory logistics and identifying potential bottlenecks.
- Transaction:** This table records all financial transactions related to purchasing activities, including payments made, refunds received, and other monetary exchanges. It provides a comprehensive view of the financial flows within the purchasing operations.
- PurchaseOrderSummary:** Owned by the Purchasing department, this central fact table aggregates data related to purchase orders. It summarises information such as total orders made, orders completed, and average processing time, offering a macro view of procurement efficiency and supplier performance.

Overview of Dimension Tables

Dimension tables in the OLAP database provide the context needed to effectively analyse the facts. These tables include:

- Date:** This dimension is key for any temporal analysis, providing the necessary axes for trend analysis, seasonal adjustments, and time-based performance metrics. Attributes include date, day, month, year, and possibly fiscal periods if required.
- Payment_Method:** Details the methods by which purchases are paid, such as credit cards, bank transfers, or cash. This dimension helps analyse payment trends and preferences, which can influence payment negotiations with suppliers.
- Transaction_Type:** Categorizes the types of transactions recorded, such as purchases, refunds, or expenses. This dimension is vital for financial segmentation and regulatory reporting.
- Stock_Item:** Contains descriptive details about each stock item, such as name, category, supplier, and price. It's crucial for inventory analysis, cost control, and strategic purchasing decisions.
- Customer:** Though primarily a sales-related dimension, customer data can be crucial for understanding the end-use of purchased goods, especially in B2B environments where purchased items are directly resold.
- Supplier:** Fundamental to the purchasing process, this dimension includes data on suppliers like names, contact details, terms, and performance ratings. This is essential for supplier management and negotiation strategies.

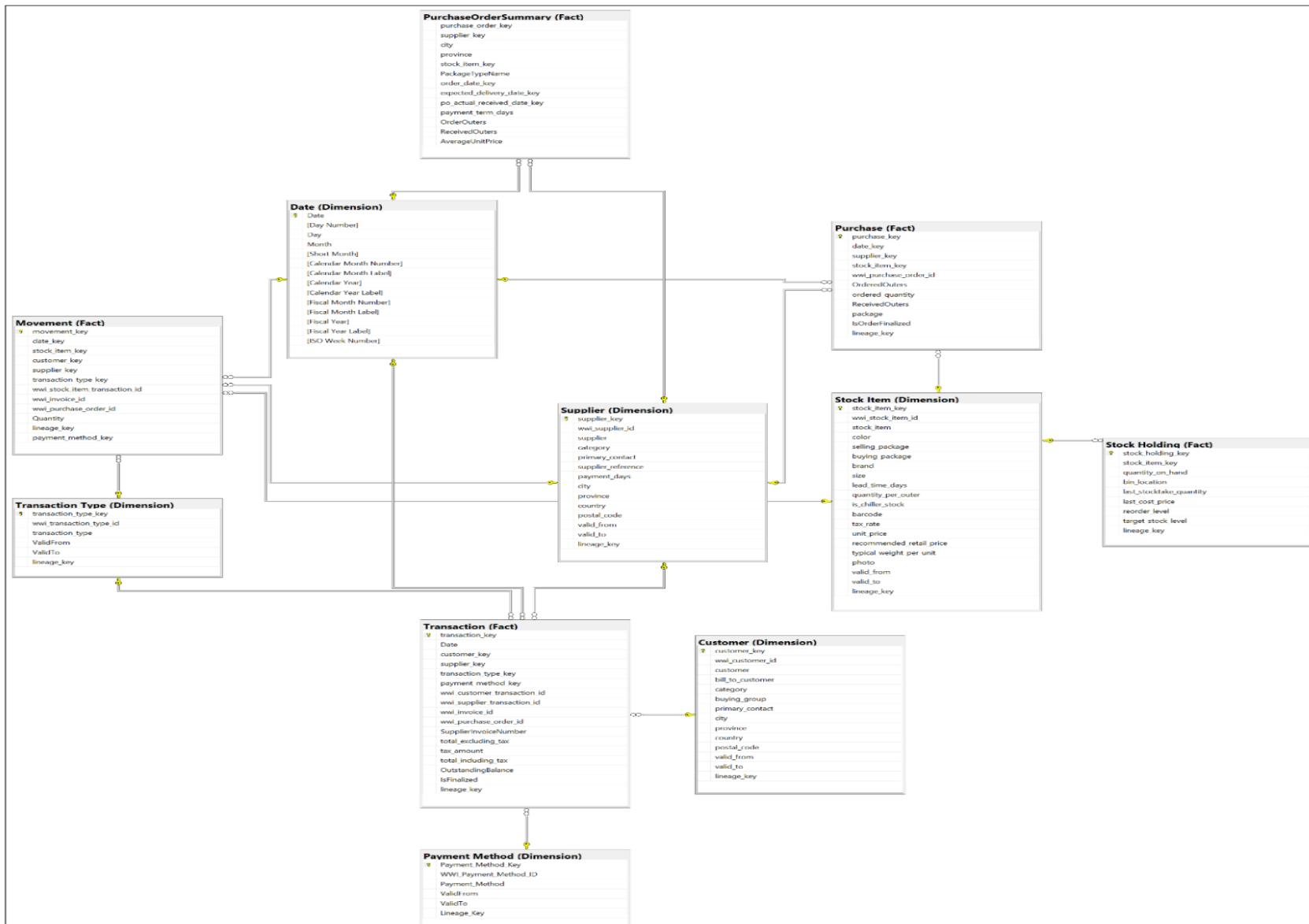


Figure 27: Facts & Dimensions for the Purchasing Department

Collaborative Validation

WWI's purchasing department will collaborate with other departments that own related data to ensure the accuracy and relevance of the data stored in the fact tables and undertake a collaborative validation process. This cross-functional engagement ensures that the data reflects the real-time and accurate state of affairs. It fosters a sense of ownership and mutual understanding among various organisational stakeholders.

4.6.4.5 Conclusion

By establishing a well-structured OLAP data warehouse with clearly defined fact and dimension tables, WWI's Purchasing department sets the stage for enhanced analytical capabilities. This infrastructure will support current operational needs and scale to meet future demands. The collaborative approach to data validation further ensures that the insights derived from the warehouse are accurate and comprehensive, enhancing the overall strategic decision-making processes at WWI.

5 Data Load Process for Purchase Data Mart

The data integration landscape has significantly shifted from traditional Extract, Transform, Load (ETL) methodologies to the more modern Extract, Load, Transform (ELT) approach. This transition is largely driven by the advent of cloud computing and the need for more agile and scalable data-handling processes. Understanding the evolution from ETL to ELT and its implications is crucial for organisations looking to optimise their data management and business intelligence capabilities.

5.1 What is ETL?

ETL has been the conventional data integration method used for decades, involving three sequential steps: Extracting data from various sources, Transforming it to fit operational needs, and Loading it into a destination database or data warehouse. This process is highly structured and often occurs in a staging area outside of the target database. While ETL has been effective for periodic batch processing, its limitations become apparent as data volumes grow and real-time analysis becomes critical.

5.1.1 Limitations of ETL

- **Complexity and Time-Consumption:** ETL processes are inherently complex and require significant time to manage, especially when dealing with large datasets.

- **Rigidity:** Once data is transformed and loaded, making changes to transformation logic or adapting to new business requirements can be cumbersome and resource-intensive.
- **Resource-intensive:** It requires substantial computational resources to perform transformations before loading, which can be costly and inefficient.

5.1.2 Why ETL Was Historically Preferred

ETL has been the foundational data integration and warehousing approach for many decades. Several factors contributed to its widespread adoption before more modern methodologies like ELT emerged. Understanding these historical reasons can help clarify why ETL was preferred and how technological advancements have shifted these preferences.

1. Cost of Storage

Historically, storage costs were significantly higher than they are today, especially when dealing with large volumes of data. ETL processes were designed to minimise the amount of data being stored by transforming data before it was loaded into the data warehouse. This transformation step often included filtering unnecessary data, aggregating information to reduce volume, and compressing data to save storage space. By applying these transformations upfront, businesses could substantially reduce the financial burden associated with data storage.

2. Limited Processing Power

Earlier computing systems had limited processing capabilities compared to modern standards. ETL processes allowed organisations to offload the heavy lifting of data transformation to separate processing environments (often on-premise servers) optimised explicitly for this purpose. This setup ensured that the data warehouse was not overwhelmed by storage and processing demands, thereby maintaining system performance and stability.

3. Data Quality and Consistency

ETL provided a controlled environment where data could be cleaned, validated, and transformed consistently before being loaded into the data warehouse. This approach was crucial for ensuring data quality and consistency, which are fundamental for accurate reporting and analysis. By centralising the transformation logic, ETL processes made enforcing business rules, data integrity, and standardisation easier across all data sources.

4. Security and Compliance

With ETL, sensitive data could be anonymised or encrypted during the transformation phase before it ever reaches the data warehouse. This aspect was particularly important for industries subject to strict data privacy regulations and security standards. Organisations could better manage risks associated with data breaches and compliance violations by processing and securing the data upstream.

5. Infrastructure Limitations

Historically, The IT infrastructure available to many organisations was not equipped to handle large-scale data processing within the database systems. ETL processes separated concerns, where dedicated ETL tools and servers handled data integration and

transformation tasks rather than the database systems, which were then primarily used for storage and querying.

5.2 Transition to ELT

The shift from ETL to ELT and other modern data integration practices has mainly been driven by advancements in cloud computing, significant reductions in storage costs, and improvements in processing power. Modern data warehouses and data lakes are designed to efficiently handle massive volumes of data and complex processing tasks. These technologies enable organisations to load raw data directly into the warehouse and perform transformations as needed, leveraging the scalability and flexibility of cloud resources.

This transition reflects a broader move towards more agile, cost-effective, and performance-optimized data management strategies that better align with today's dynamic business environments and real-time analytical needs.

5.2.1 Advantages of ELT

- **Faster Processing and Loading Speed:** By loading data directly into the data warehouse and utilising its computational power, ELT significantly reduces the time it takes to make data available for analysis.
- **Scalability and Lower Cost:** ELT is more scalable in cloud environments, where it can handle increasing data volumes without corresponding increases in infrastructure costs.
- **Greater Flexibility:** Since transformation rules are applied within the data warehouse, analysts can modify or rerun transformations as needed without extensive pre-planning or rerunning the entire pipeline.

5.2.2 Key Features of ELT

1. **Accessibility of Data:** ELT provides easier access to raw data, allowing for a broader range of data manipulation and exploration capabilities. It supports automatic schema migrations and standardised models across all data pipelines, simplifying maintenance.
2. **Data Integrity and Reliability:** Direct access to raw data ensures that analysts can always reference the original data, improving the reliability of insights derived and protecting against data transformation errors that might otherwise lead to data loss or corruption.
3. **Support for More Data Sources:** ELT's simplified architecture makes building and maintaining connectors easier, supporting a wider array of data sources and formats, including both structured and unstructured data.
4. **Enhanced Automation and Data Literacy:** ELT facilitates no-code or low-code data integration and fully managed services, promoting data literacy across the organisation by enabling more stakeholders to engage with data directly.

5.3 Medallion Architecture for ELT

The Medallion Architecture is a systematic approach to data warehousing and analytics that is especially well-suited to ELT (Extract, Load, Transform) processes. This architecture is characterised by dividing data into three distinct layers—Bronze, Silver, and Gold—each serving a specific purpose in managing and utilising data. Here's how it fits into the ELT framework and the benefits it offers.

5.3.1 Structure of Medallion Architecture

1. Bronze Layer (Raw Data Layer):

- **Function:** This is the foundational layer where data is loaded directly from various source systems in its raw, unprocessed form. In the context of ELT, raw data is extracted and loaded into this layer first without any transformations.
- **Benefits:** The Bronze layer acts as a single source of truth for all raw data, essential for audit trails, compliance, and data verification. It ensures that the original data can always be referenced, crucial for tracing data lineage and debugging.

2. Silver Layer (Cleansed Data Layer):

- **Function:** Data moved to the Silver layer undergoes initial cleaning and transformation processes. These might include standardisation, deduplication, validation, and sometimes light aggregation. The transformations are done directly in the data warehouse, utilising its processing power.
- **Benefits:** The Silver layer improves data quality and prepares data for more detailed analysis and reporting. It serves as an intermediate, reliable dataset that can be used for further refinement and exploration without impacting the raw data layer.

3. Gold Layer (Business-Ready Data Layer):

- **Function:** The final layer of the Medallion Architecture is where data is transformed and enriched to directly support business intelligence, analytics, and reporting. Data is often aggregated, enriched with business logic, and organised into dimensional models or other structures that facilitate easy consumption.
- **Benefits:** The Gold layer provides high-value, actionable insights tailored to business needs. It enables quick access to performance metrics and analytical outputs that decision-makers rely on to drive business strategies.

5.3.2 How Medallion Architecture Fits into ELT

In ELT processes, the power of modern cloud-based data warehouses is leveraged to perform transformations after the data is loaded, unlike traditional ETL, where transformations occur before loading. The Medallion Architecture is particularly effective in ELT settings for several reasons:

- **Performance:** Modern data warehouses' processing capabilities make transformations more efficient and can quickly handle large volumes of data. This setup reduces the need for separate transformation servers and simplifies the data pipeline.
- **Flexibility:** Since transformations occur within the data warehouse, it allows for more flexible and dynamic data manipulation. Analysts can modify or extend data models in the Silver and Gold layers without rerunning initial extraction and loading processes.
- **Scalability:** The separation into three layers enables scalable data management practices. As data volume or complexity grows, each layer can be scaled or optimised independently based on specific requirements.

5.3.3 Benefits of Medallion Architecture in ELT

- **Data Integrity and Transparency:** Keeping the raw data intact in the Bronze layer ensures that any transformations do not obscure the original data, maintaining transparency and integrity.
- **Iterative Refinement:** Data can be progressively refined through the layers, allowing for iterative data quality and usability improvements. This staged approach supports gradual enhancements without disrupting existing data workflows.
- **Reduced Load Times:** Separating the load and transform stages makes data available sooner in the Bronze layer, which can be critical for time-sensitive applications.
- **Enhanced Data Security:** Sensitive data can be more effectively managed and protected through controlled transformations and access policies at different layers.

The Medallion Architecture enhances ELT processes by providing a structured, efficient, and flexible data management approach that aligns with modern data warehouses' capabilities and advantages. This architecture streamlines the data transformation process and optimises data utilisation across the organisation.

5.4 Medallion Architecture for Purchasing Department

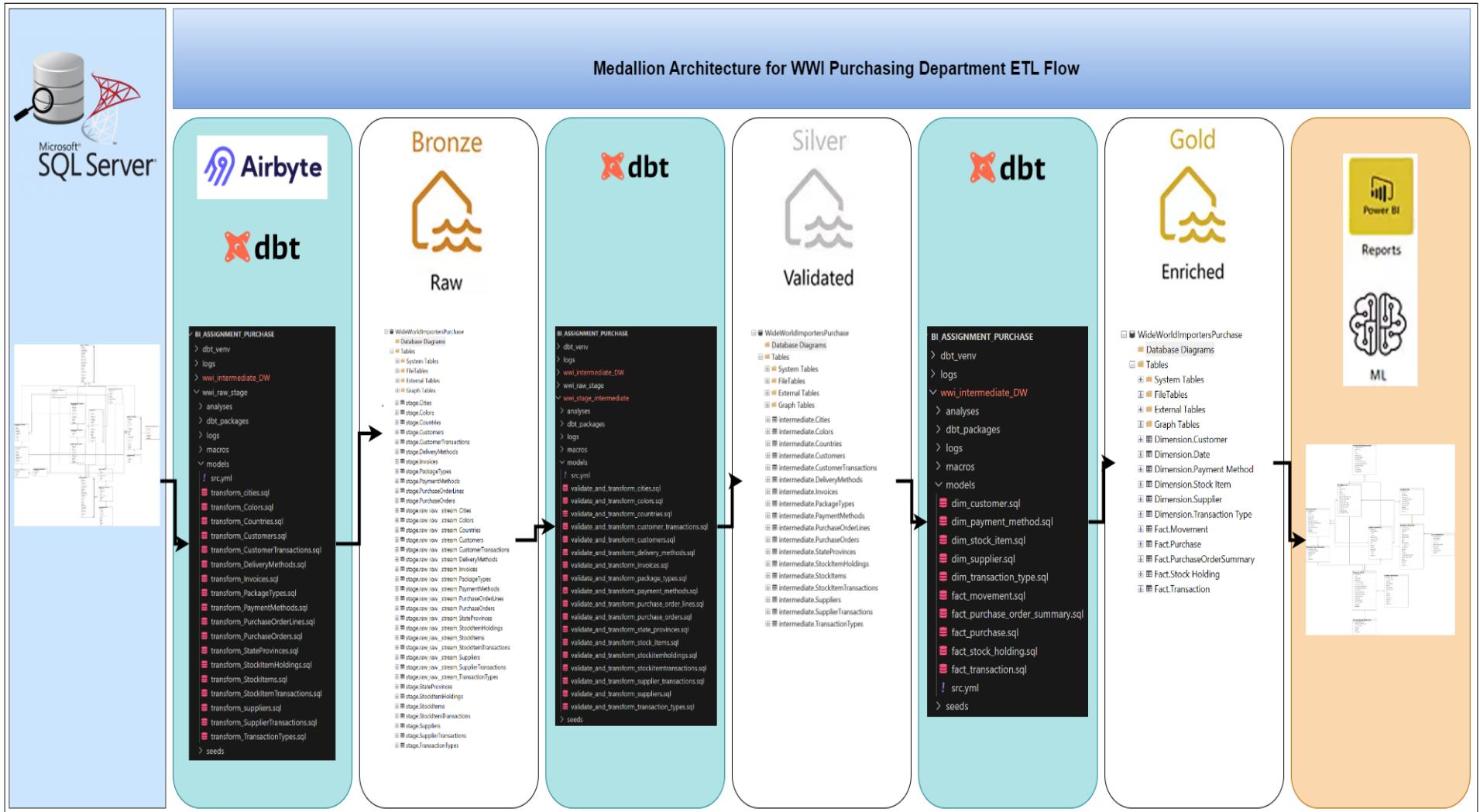


Figure 28: Medallion Architecture

In this project for the Wild World Importers (WWI) Purchasing Department, the Medallion Architecture was employed to facilitate the ELT process for creating a robust data warehouse. We explore the deployment of this architecture and the associated tools, highlighting the strategic adoption of modern data integration and transformation technologies.

5.4.1 Medallion Architecture in the ELT Framework

The Medallion Architecture, as utilised in this project, is segmented into three distinct layers, each serving a pivotal role in the management and transformation of data:

1. **Bronze Layer:** This initial layer serves as the repository for raw data extracted from the WWI OLTP database. The data remains untransformed in this layer, preserving its original state for completeness and integrity. This setup is crucial for maintaining a reliable audit trail and for ensuring compliance with data governance standards.
2. **Silver Layer:** The data in this intermediate layer undergoes preliminary cleaning processes to enhance data quality. These straightforward transformations aim to standardise, deduplicate, and validate data without applying complex business logic. This ensures that the data is reliable and prepared for more detailed analytics.
3. **Gold Layer:** Representing the culmination of the data transformation process, this layer contains data that has been fully transformed and structured according to the data warehouse's Star Schema design. This structured data is optimised for querying and analysis, providing valuable insights directly applicable to business decision-making.

5.4.2 Tools Employed in the Data Pipeline

To support the efficient operation of this data architecture, specific tools were chosen for their robust capabilities in data extraction, loading, and transformation:

- **Airbyte:** Selected for its powerful data extraction and loading capabilities, Airbyte is an open-source ELT tool that facilitates the seamless transfer of data from various sources into the data warehouse. Its extensive library of connectors allows for easy integration with multiple source systems, making it a versatile choice for diverse data environments.
- **dbt (data build tool):** dbt is employed for its excellence in transforming data directly within the data warehouse. This tool defines data transformations as code which supports version control and collaboration. dbt handles the transformation tasks in the Silver and Gold layers, enabling complex data modeling and business logic applications to prepare data for analytical use.

5.4.3 Alternative Tools Comparable to Airbyte and dbt

For organisations considering similar implementations, various alternative tools offer functionalities similar to Airbyte and dbt:

- **Fivetran:** Like Airbyte, Fivetran is a cloud-based data integration service that performs automated data extraction and loading. It supports various source and destination platforms and is known for its high reliability and extensive connector availability.
- **Stitch:** Another alternative to Airbyte, Stitch is an ETL service that integrates with various data sources and loads data into cloud data warehouses. It is designed for rapid setup and automated scaling, making it suitable for businesses of all sizes.
- **Apache Nifi:** A flexible and configurable tool, Apache Nifi supports data routing, transformation, and system mediation logic. While it is more complex to set up than Airbyte, it offers high customizability.
- **Matillion:** As an alternative to dbt, Matillion offers data transformation capabilities designed specifically for cloud data warehouses. It provides a visual interface that allows users to create data transformation jobs without extensive coding.
- **Informatica:** Informatica offers a range of data integration products that can be used for both ETL and ELT processes. It is particularly strong in enterprise-level data integration, data quality, and data governance.
- **Azure Data Factory (ADF):** Azure Data Factory is a strong alternative to tools like Airbyte and dbt, especially within environments anchored in Microsoft's cloud services. ADF is a cloud-based data integration service that allows users to create, schedule, and orchestrate ETL/ELT processes at scale. Unlike Airbyte, which specializes in data extraction and loading, ADF provides a comprehensive suite of integration services that can handle data extraction, workflow orchestration, and even transformation — though it typically orchestrates transformation processes by integrating with other services like Azure Data Lake Analytics or Azure Databricks. This makes ADF particularly valuable for enterprises that require robust data processing workflows and prefer a single, integrated platform to manage these processes. ADF's deep integration with other Azure services enhances its appeal, providing seamless connectivity and extending capabilities within the Azure ecosystem. Its visual tools and extensive monitoring capabilities allow users to configure, manage, and monitor data pipelines, ensuring data is reliably prepared and delivered for business analytics, reporting, and decision-making processes. Additionally, ADF supports a wide range of data sources and destinations, making it versatile for diverse data integration needs, similar to how dbt provides flexibility in data transformation within the data warehouse environment.

5.4.4 Rationale for Choosing Airbyte and dbt

The decision to implement the data pipeline using Airbyte and dbt was driven by several strategic considerations:

1. **Understanding Tool Implementation Steps:** Using Airbyte and dbt, the team aimed to gain a deep, hands-on understanding of each step involved in the tool implementation process. This practical experience was invaluable, allowing the team to learn the nuances of configuring and managing data pipelines intimately.
2. **Justification of Pre-Project Study:** Extensive learning was conducted to assess the capabilities of various tools before the project's initiation. Implementing Airbyte and dbt allows the project to leverage this pre-project study, ensuring that the investments in time and resources to understand these tools are capitalised upon.
3. **Applicability to Small & Medium Enterprises (SMEs):** Both Airbyte and dbt offer capabilities that are highly relevant to SMEs. They provide a hands-on, customisable approach to data integration and transformation, allowing smaller organisations to adapt these tools to their specific needs without the overhead of more complex, fully managed platforms like Azure Data Factory. The choice of these tools also offers an opportunity to explore and navigate the challenges of implementing more grassroots-level technologies that do not abstract the complexity of their environments, providing a richer learning and development experience.

5.4.5 High-Level Steps for Implementing Airbyte on Windows

Installing and Setting Up Airbyte:

1. **System Requirements:** Ensure your Windows system meets the minimum requirements, including Docker Desktop, as Airbyte runs on Docker.
2. **Install Docker:** Download and install Docker Desktop for Windows. During installation, enable the WSL 2 feature and ensure that Docker is set to use Linux containers.

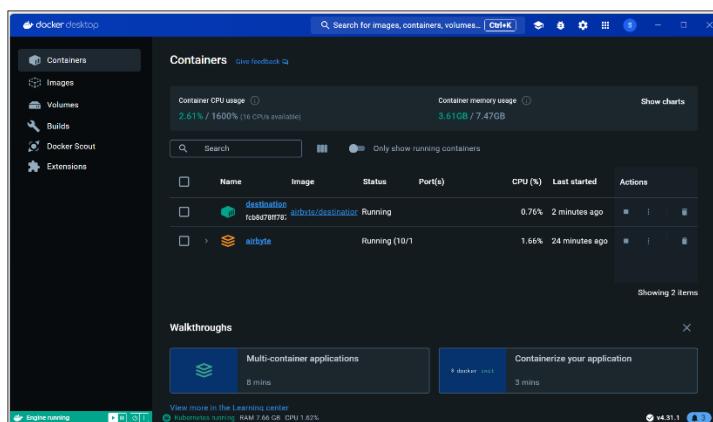


Figure 29: Docker Desktop for Airbyte

3. **Download Airbyte:** Once Docker is running, pull the Airbyte Docker image from Docker Hub using the Docker command line interface. The specific commands and details of the latest version are on the Airbyte documentation or Docker Hub page.

```
sachi@sACHIITH-DESKTOP:~/airbyte$ bash run-ab-platform.sh
AIRBYTE
Move Data

found docker-compose.yaml locally!
found docker-compose.debug.yaml locally!
found .env locally!
found .env.dev locally!
found flags.yaml locally!
found temporal/dynamicconfig/development.yaml locally!
Loading Shell Variables from .env...
Loading Shell Variables from .env.dev...

Starting Docker Compose
WARN[0000] The "GITHUB_STORE_BRANCH" variable is not set. Defaulting to a blank string.
WARN[0000] The "JOB_ERROR_REPORTING_SENTRY_DSN" variable is not set. Defaulting to a blank string.
WARN[0000] The "LAUNCHDARKLY_KEY" variable is not set. Defaulting to a blank string.
WARN[0000] The "NEW_SCHEDULER" variable is not set. Defaulting to a blank string.
WARN[0000] The "WORKER_ENVIRONMENT" variable is not set. Defaulting to a blank string.
WARN[0000] The "DEPLOYMENT_MODE" variable is not set. Defaulting to a blank string.
WARN[0000] The "TEMPORAL_HISTORY_RETENTION_IN_DAYS" variable is not set. Defaulting to a blank string.
WARN[0000] The "UPDATE_DEFINITIONS_CRON_ENABLED" variable is not set. Defaulting to a blank string.
WARN[0000] The "DEPLOYMENT_MODE" variable is not set. Defaulting to a blank string.
WARN[0000] The "APPLY_FIELD_SELECTION" variable is not set. Defaulting to a blank string.
WARN[0000] The "DEPLOYMENT_MODE" variable is not set. Defaulting to a blank string.
WARN[0000] The "FIELD_SELECTION_WORKSPACES" variable is not set. Defaulting to a blank string.
WARN[0000] The "JOB_ERROR_REPORTING_SENTRY_DSN" variable is not set. Defaulting to a blank string.
WARN[0000] The "LAUNCHDARKLY_KEY" variable is not set. Defaulting to a blank string.
WARN[0000] The "LOG_CONNECTOR_MESSAGES" variable is not set. Defaulting to a blank string.
WARN[0000] The "CDK_VERSION" variable is not set. Defaulting to a blank string.
WARN[0000] The "DEPLOYMENT_MODE" variable is not set. Defaulting to a blank string.
WARN[0000] The "PYTHON_VERSION" variable is not set. Defaulting to a blank string.
WARN[0000] /home/sachi/airbyte/docker-compose.yaml: 'version' is obsolete
[+] Running 15/15
✓ Network airbyte_default          Created      0.0s
✓ Network airbyte.airbyte_internal Created      0.0s
✓ Network airbyte.airbyte_public   Created      0.0s
✓ Container airbyte-temporal       Created      0.1s
✓ Container airbyte-docker-proxy-1 Created      0.1s
✓ Container init                   Created      0.1s
✓ Container airbyte-db             Created      0.1s
✓ Container airbyte-bootloader    Created      0.1s
✓ Container airbyte-server         Created      0.1s
✓ Container airbyte-worker         Created      0.1s
✓ Container airbyte-webapp        Created      0.1s
✓ Container airbyte-connector-builder-server Created  0.1s
✓ Container airbyte-api-server     Created      0.1s
✓ Container airbyte-cron           Created      0.1s
✓ Container airbyte-proxy          Created      0.0s
Attaching to airbyte-api-server, airbyte-bootloader, airbyte-connector-builder-server, airbyte-cron, airbyte-db, docker-pro
init          | MOUNT: /local_parent
init          | ROOT_PARENT: /tmp
init          | ROOT: /tmp/airbyte_local
init          | MOUNT_ROOT: /local_parent//airbyte_local
airbyte-temporal | Start init
airbyte-db    | PostgreSQL Database directory appears to contain a database; Skipping initialization
airbyte-db    |
airbyte-db    | PostgreSQL Database directory appears to contain a database; Skipping initialization
airbyte-db    | PostgreSQL Database directory appears to contain a database; Skipping initialization
```

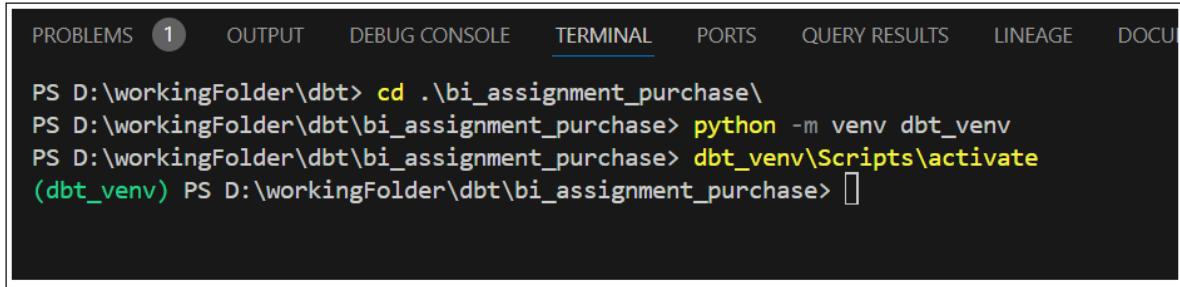
Figure 30: Airbyte Docker image running on WSL

4. **Launch Airbyte:** Run the Docker compose commands to start the Airbyte server and web app interface. The Airbyte interface will be accessible via a web browser, typically at `localhost:8000`.
 5. **Configure Data Sources and Destinations:** Use the Airbyte web interface to configure data sources and destinations. Airbyte offers a range of connectors for various data sources and warehouses. Setup involves entering the necessary connection details and credentials for each source and destination.
 6. **Create and Monitor Pipelines:** Set up data synchronisation pipelines within the Airbyte interface, specifying how often data should be replicated and any specific configurations related to data transformation or filtering. Monitor the operations and performance through the dashboard.

5.4.6 High-Level Steps for Implementing dbt on Windows

Setting Up dbt:

1. **Install Python:** dbt requires Python, so start by installing the latest version of Python for Windows. Ensure that Python and pip (Python's package installer) are correctly installed by checking their versions in the command prompt.

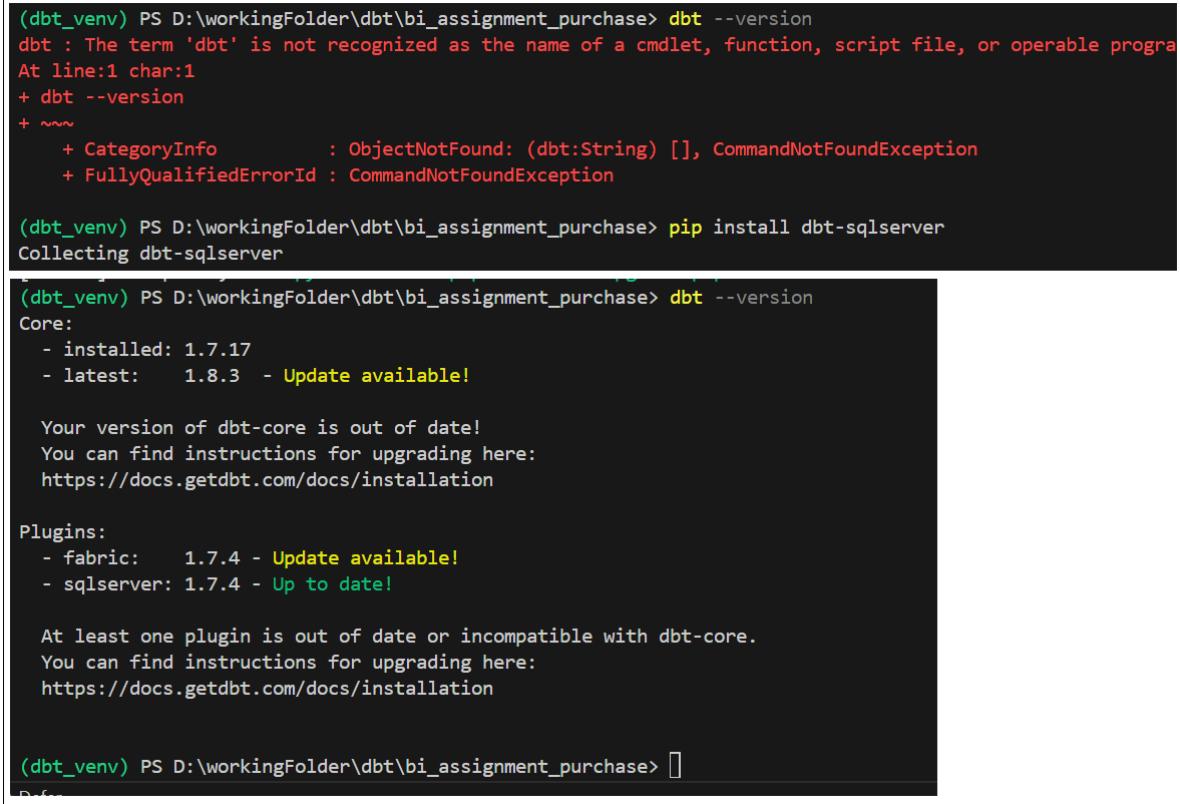


```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS LINEAGE DOCU

PS D:\workingFolder\dbt> cd .\bi_assignment_purchase\
PS D:\workingFolder\dbt\bi_assignment_purchase> python -m venv dbt_venv
PS D:\workingFolder\dbt\bi_assignment_purchase> dbt_venv\Scripts\activate
(dbt_venv) PS D:\workingFolder\dbt\bi_assignment_purchase> 
```

Figure 31: Python Virtual Environment

2. **Install dbt:** Use pip to install dbt. Open your Windows command prompt and run the pip install command provided by dbt documentation, such as pip install dbt.



```
(dbt_venv) PS D:\workingFolder\dbt\bi_assignment_purchase> dbt --version
dbt : The term 'dbt' is not recognized as the name of a cmdlet, function, script file, or operable program
At line:1 char:1
+ dbt --version
+ ~~~
+ CategoryInfo          : ObjectNotFound: (dbt:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

(dbt_venv) PS D:\workingFolder\dbt\bi_assignment_purchase> pip install dbt-sqlserver
Collecting dbt-sqlserver
  Using cached https://files.pythonhosted.org/packages/.../dbt-sqlserver-0.18.0-py3-none-any.whl
Requirement already satisfied: dbt-sqlserver in ./dbt-sqlserver-0.18.0-py3-none-any.whl

(dbt_venv) PS D:\workingFolder\dbt\bi_assignment_purchase> dbt --version
Core:
- installed: 1.7.17
- latest:    1.8.3 - Update available!

Your version of dbt-core is out of date!
You can find instructions for upgrading here:
https://docs.getdbt.com/docs/installation

Plugins:
- fabric:    1.7.4 - Update available!
- sqlserver: 1.7.4 - Up to date!

At least one plugin is out of date or incompatible with dbt-core.
You can find instructions for upgrading here:
https://docs.getdbt.com/docs/installation

(dbt_venv) PS D:\workingFolder\dbt\bi_assignment_purchase> 
```

Figure 32: dbt Installation

3. **Initialize dbt Project:** Create a new directory for your dbt project and navigate into it via the command line. Then, initialise a new dbt project using the `dbt init` command and follow the prompts to set up your project structure.
4. **Configure the `profiles.yml` File:** Set up your database connection in the `profiles.yml` file located in your dbt directory. This file contains all the necessary configurations for your target database.
5. **Develop dbt Models:** Develop your dbt models, which are SQL files stored in the `models` folder of your dbt project. These SQL files define the transformations that will be applied to your raw data.
6. **Run dbt:** Compile and run your dbt models using the `dbt run` command to apply transformations and load the transformed data into your data warehouse.
7. **Test and Document:** Use dbt's capabilities to perform tests and generate documentation on your models and data transformations, ensuring the integrity and maintainability of your data pipeline.

Choosing Airbyte and dbt balances hands-on control with the sophistication needed for robust data operations, making them ideal for projects that comprehensively understand data tool implementations.

5.5 Bronze Layer

The Bronze layer serves as the foundational tier in the Medallion Architecture, where data is transferred from the WideWorldImporters OLTP database, hosted on SQL Server, to a specifically designated data mart named WildWorldImportersPurchase, also on SQL Server. This initial layer is crucial as it ensures that raw data from various operational systems is consolidated into a single repository, maintaining its original form for accuracy and traceability.

5.5.1 Setup and Operation of Airbyte

Airbyte plays a pivotal role in this architecture's extract and load operations. The tool is configured to connect both the source (OLTP database) and the destination (Purchasing Department data mart) by setting up necessary connection details, including IP addresses, ports, and specific database names. The configuration also involves specifying the tables to be transferred, referred to as "connections" within Airbyte.

Source Configuration Details

The database name is specified and highlighted in red for the source configuration. The update method chosen is "Scan Changes with User Define Cursor," which differs from Change Data

Capture (CDC). While the CDC captures and transfers only the changes made to the database, thus requiring tracking changes at the source, the "Scan Changes" method with a user-defined cursor allows for more controlled data extraction based on custom criteria without altering the source tables. This method is particularly useful when the pipeline is restricted from changing source tables or introducing new objects, which is often the case with most source databases. The schemas of interest for the pipeline are configured and highlighted in orange in the setup.

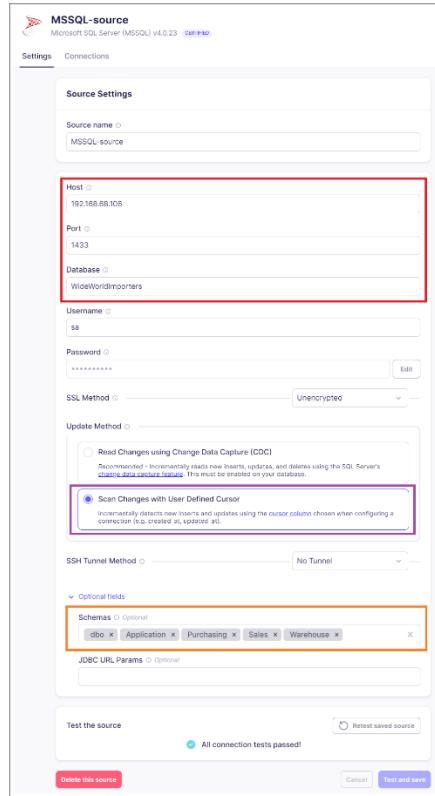


Figure 33: Airbyte Sourec Configuration

Destination Configuration Details

The destination settings include the IP, port, and database name, which are highlighted in orange. The target schema for the Extract and Load process is specified in the "Raw Table Schema Name" section, highlighted in red. This structured approach ensures the data is loaded into the correct SQL Server data mart schema.

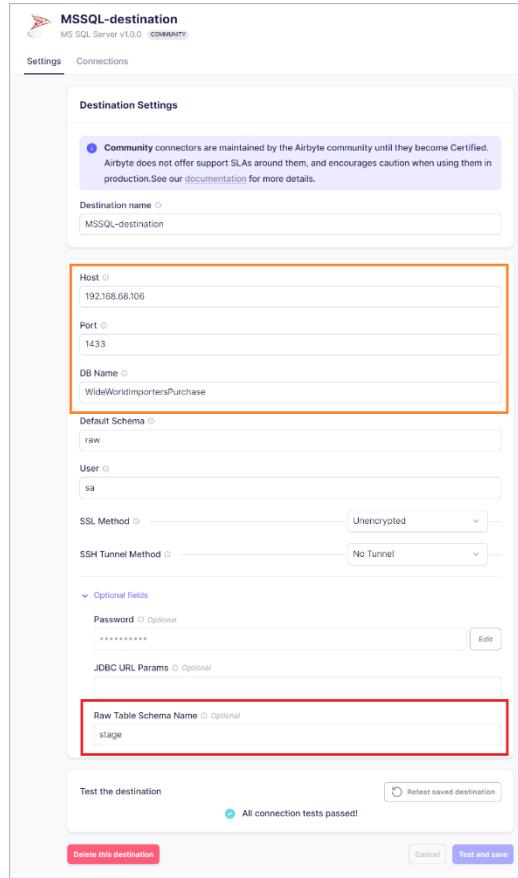


Figure 34: Airbyte Destination Configuration

Table Configuration and Data Refresh Strategy

For this project, tables such as CustomerTransactions, Countries, Customers, and others from the WildWorldImporter OLTP database are configured with the "Full Refresh/Override" option. Although this method is not always optimal due to its comprehensive data replacement approach, it is suitable given this database's manageable volume of data. This strategy ensures the entire dataset is refreshed during each execution, maintaining consistency and currency.

The screenshot shows the Airbyte Connection Setup interface for a connection named "BRONZE-WWI_OLTP_to_Staging". At the top, it displays the connection status as "MSSQL-source CERTIFIED" connected to "MSSQL-destination COMMUNITY". There are buttons for "Manual", "Sync now", and "ENABLED". Below this, the "Status" tab is selected, showing the "Active Streams" table.

Status	Stream name	Last record loaded
On time	Cities	14 hours ago
On time	Colors	14 hours ago
On time	Countries	14 hours ago
On time	Customers	14 hours ago
On time	CustomerTransactions	14 hours ago
On time	DeliveryMethods	14 hours ago
On time	Invoices	14 hours ago
On time	PackageTypes	14 hours ago
On time	PaymentMethods	14 hours ago
On time	PurchaseOrderLines	14 hours ago
On time	PurchaseOrders	14 hours ago
On time	StateProvinces	14 hours ago
On time	StockItemHoldings	14 hours ago
On time	StockItems	14 hours ago
On time	StockItemTransactions	14 hours ago
On time	Suppliers	14 hours ago
On time	SupplierTransactions	14 hours ago
On time	TransactionTypes	14 hours ago

Figure 35: Airbyte Connection Setup

Monitoring Execution

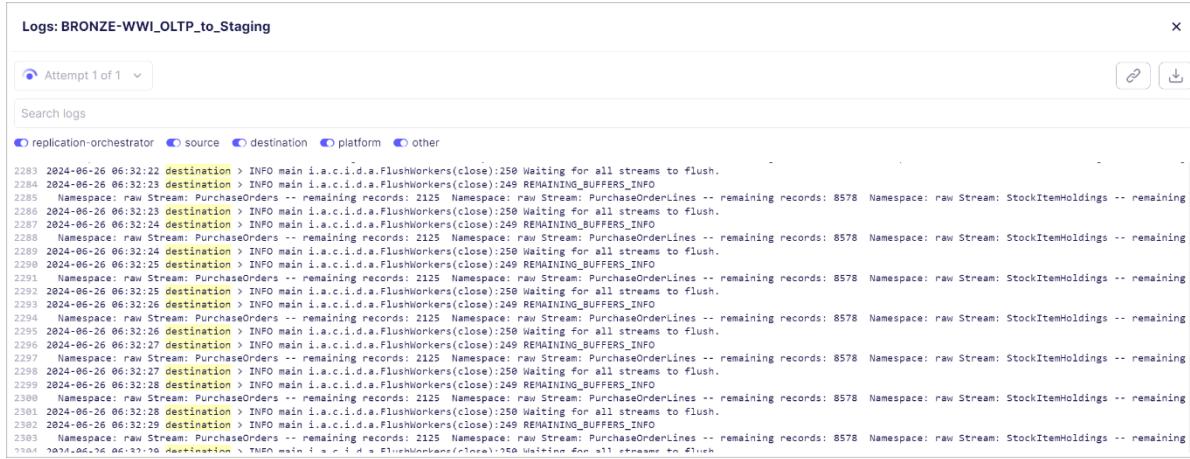
The execution of Airbyte processes can be actively monitored through the "Job History" tab available on the Airbyte dashboard. Logs can also be accessed through the dashboard or the console, providing real-time insights into the operational status and any issues encountered during the data transfer process.

The screenshot shows the Airbyte Job Monitoring interface for the same connection. The "Job History" tab is selected. It displays a table of recent sync events.

Action	Details	Timestamp	More Options
Sync Running		10:36AM 06/28/2024	⋮
Sync Succeeded	115.77 MB 395,658 records extracted 395,658 records loaded 45m 52s	7:41PM 06/27/2024	⋮
Sync Succeeded	115.37 MB 395,605 records extracted 395,605 records loaded 25m 28s	6:25PM 06/26/2024	⋮

Figure 36: Airbyte Job Monitoring

This structured approach to setting up and monitoring the data pipeline with Airbyte ensures efficient data transfer from the OLTP system to the Bronze layer of the data warehouse, supporting the overall data strategy for Wild World Importers.



```

Logs: BRONZE-WWI_OLTP_to_Staging

Attempt 1 of 1

Search logs

replication-orchestrator source destination platform other

2283 2024-06-26 06:32:22 destination > INFO main i.a.c.i.d.a.FlushWorkers(close):250 Waiting for all streams to flush.
2284 2024-06-26 06:32:23 destination > INFO main i.a.c.i.d.a.FlushWorkers(close):249 REMAINING_BUFFERS_INFO
2285 Namespace: raw Stream: PurchaseOrders -- remaining records: 2125 Namespace: raw Stream: PurchaseOrderLines -- remaining records: 8578 Namespace: raw Stream: StockItemHoldings -- remaining
2286 2024-06-26 06:32:23 destination > INFO main i.a.c.i.d.a.FlushWorkers(close):250 Waiting for all streams to flush.
2287 2024-06-26 06:32:24 destination > INFO main i.a.c.i.d.a.FlushWorkers(close):249 REMAINING_BUFFERS_INFO
2288 Namespace: raw Stream: PurchaseOrders -- remaining records: 2125 Namespace: raw Stream: PurchaseOrderLines -- remaining records: 8578 Namespace: raw Stream: StockItemHoldings -- remaining
2289 2024-06-26 06:32:24 destination > INFO main i.a.c.i.d.a.FlushWorkers(close):250 Waiting for all streams to flush.
2290 2024-06-26 06:32:25 destination > INFO main i.a.c.i.d.a.FlushWorkers(close):249 REMAINING_BUFFERS_INFO
2291 Namespace: raw Stream: PurchaseOrders -- remaining records: 2125 Namespace: raw Stream: PurchaseOrderLines -- remaining records: 8578 Namespace: raw Stream: StockItemHoldings -- remaining
2292 2024-06-26 06:32:25 destination > INFO main i.a.c.i.d.a.FlushWorkers(close):250 Waiting for all streams to flush.
2293 2024-06-26 06:32:26 destination > INFO main i.a.c.i.d.a.FlushWorkers(close):249 REMAINING_BUFFERS_INFO
2294 Namespace: raw Stream: PurchaseOrders -- remaining records: 2125 Namespace: raw Stream: PurchaseOrderLines -- remaining records: 8578 Namespace: raw Stream: StockItemHoldings -- remaining
2295 2024-06-26 06:32:26 destination > INFO main i.a.c.i.d.a.FlushWorkers(close):250 Waiting for all streams to flush.
2296 2024-06-26 06:32:27 destination > INFO main i.a.c.i.d.a.FlushWorkers(close):249 REMAINING_BUFFERS_INFO
2297 Namespace: raw Stream: PurchaseOrders -- remaining records: 2125 Namespace: raw Stream: PurchaseOrderLines -- remaining records: 8578 Namespace: raw Stream: StockItemHoldings -- remaining
2298 2024-06-26 06:32:27 destination > INFO main i.a.c.i.d.a.FlushWorkers(close):250 Waiting for all streams to flush.
2299 2024-06-26 06:32:28 destination > INFO main i.a.c.i.d.a.FlushWorkers(close):249 REMAINING_BUFFERS_INFO
2300 Namespace: raw Stream: PurchaseOrders -- remaining records: 2125 Namespace: raw Stream: PurchaseOrderLines -- remaining records: 8578 Namespace: raw Stream: StockItemHoldings -- remaining
2301 2024-06-26 06:32:28 destination > INFO main i.a.c.i.d.a.FlushWorkers(close):250 Waiting for all streams to flush.
2302 2024-06-26 06:32:29 destination > INFO main i.a.c.i.d.a.FlushWorkers(close):249 REMAINING_BUFFERS_INFO
2303 Namespace: raw Stream: PurchaseOrders -- remaining records: 2125 Namespace: raw Stream: PurchaseOrderLines -- remaining records: 8578 Namespace: raw Stream: StockItemHoldings -- remaining
2304 2024-06-26 06:32:30 destination > INFO main i.a.c.i.d.a.FlushWorkers(close):250 Waiting for all streams to flush.

```

Figure 37: Airbyte Logs

5.5.2 Data Transfer with Airbyte

While Airbyte supports SQL Server 2019 for data integration processes, our experience revealed certain limitations with the standard SQL Connector provided by Airbyte. Specifically, it was noted that although the Airbyte documentation suggested the capability to create normalized tables post-data transfer that mirrors the structure of the source tables, this functionality was not observed in practice with the version we utilized. This discrepancy meant that the data was loaded into the target database without the expected normalization, leading to additional steps in the data integration process.

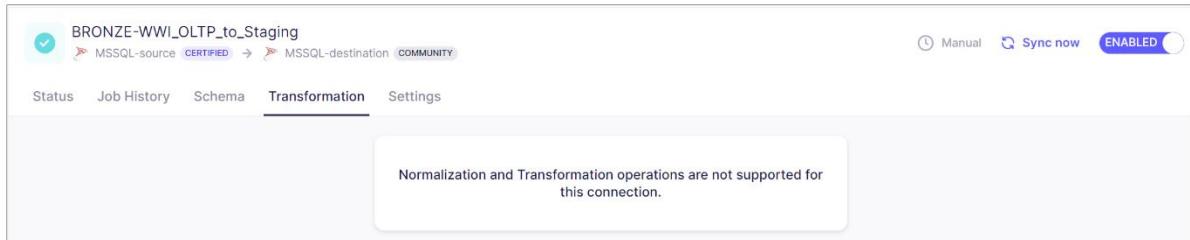


Figure 38: Airbyte Normalisation was Not Supported

To address this issue, we employed dbt to implement the necessary transformations. dbt was utilised for restructuring and normalising the data post-transfer, effectively creating tables in the target database that closely resemble the original schema of the source database. This step was crucial for maintaining the integrity and usability of the data within the Wild World Importers purchasing environment.

```

USE [WideWorldImportersPurchase];
DECLARE @SQL AS NVARCHAR(MAX);
SET @SQL = '';
-- Use the appropriate database
USE [WideWorldImportersPurchase];
-- Declare a variable to hold dynamically constructed SQL
-- Initialize the variable
SET @SQL = '';
-- Select statement to construct SQL for counting rows in each table in the 'stage' schema
SELECT @SQL = @SQL + 'SELECT ''' + TABLE_SCHEMA + '.' + TABLE_NAME + ''' AS Table_Name, COUNT(*) AS Row_Count FROM [' + TABLE_SCHEMA + '].[.' + TABLE_NAME + ']';
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_SCHEMA = 'stage' AND TABLE_TYPE = 'BASE TABLE';
-- Execute the dynamic SQL
EXEC sp_executesql @SQL;

```

Figure 39: Target DB after Airbyte Run

This image showcases the SQL Server target database following the data load via Airbyte. Notably, the record numbers in each raw table align precisely with those in the source table, indicating successful data replication in terms of volume.

Visualization of Raw Data and Metadata Management

Airbyte's mechanism for handling data involves transferring each record in a JSON format. This method encapsulates the data effectively and includes meta-information about each transaction. This metadata is crucial as it facilitates the tracking of changes and supports incremental loads in subsequent runs. Including metadata ensures that each data transfer is documented and traceable, enhancing the reliability and suitability of the data integration process.

```

USE [WideWorldImportersPurchase];
SELECT * FROM [stage].[raw_raw_stream_Cities];

```

	<u>_airbyte_raw_id</u>	<u>_airbyte_data</u>	<u>_airbyte_extracted_at</u>	<u>_airbyte_loaded_at</u>	<u>_airbyte_meta</u>
1	0001d290-6889-4916-ae05-cf3ed78bce88	{"CityID":7043,"CityName":"Coffee City","StateProvinceID":1}	2024-06-26 06:26:08.2310000 +00:00	NULL	NULL
2	0002de9d-ab3c-4205-920c-e957cb58fcf1	{"CityID":27747,"CityName":"Pottstown","StateProvinceID":2}	2024-06-26 06:26:08.2310000 +00:00	NULL	NULL
3	000350b9-6bf0-4ac8-8e05-c47acf3286fa	{"CityID":36397,"CityName":"Wendell","StateProvinceID":3}	2024-06-26 06:26:08.2310000 +00:00	NULL	NULL
4	000375c8-fc2c-488c-9608-7dac25b408f3	{"CityID":27416,"CityName":"Point Lookout","StateProvinceID":4}	2024-06-26 06:26:08.2310000 +00:00	NULL	NULL
5	00067bed-6483-4413-955b-a7160ffdd43f	{"CityID":8252,"CityName":"Curtiss","StateProvinceID":5}	2024-06-26 06:26:08.2310000 +00:00	NULL	NULL
6	0006889b-4660-40d3-83e7-b5d509933b93	{"CityID":6174,"CityName":"Chelmsford","StateProvinceID":6}	2024-06-26 06:26:08.2310000 +00:00	NULL	NULL

Figure 40: Airbyte Raw Table

The image below provides a visualization of the raw data in the SQL Server target database. The JSON format and the accompanying meta information are visible, illustrating how Airbyte structures data during the transfer. This format plays a vital role in the subsequent transformation

stages, where dbt leverages this metadata to efficiently track changes and update the data as needed.

```
{
  "CityID": 7043,
  "CityName": "Coffee City",
  "StateProvinceID": 45,
  "Location": "POINT(-95.4994013 32.1159915)",
  "LatestRecordedPopulation": 278,
  "LastEditedBy": 1,
  "ValidFrom": "2013-01-01T00:00:00.000000",
  "ValidTo": "9999-12-31T23:59:59.999999"
}
```

Figure 41: JSON Record

5.5.3 Normalisation of JSON Raw Data Using dbt Scripts

Transformation Process with dbt

Transferring JSON-formatted raw data to normalized tables is critical in refining data within our data architecture. This process was efficiently executed using dbt scripts, specifically designed to handle the transformation of raw data stored in JSON format into structured, normalized tables within the data warehouse. This transformation is fundamental to the Bronze layer of our Medallion Architecture, where it acts as a raw data repository and begins structuring this data for more detailed analysis and reporting.

```

models > transform_SupplierTransactions.sql
1   {{ config(materialized='table', alias='SupplierTransactions') }}
2
3   WITH source_data AS (
4     SELECT
5       CAST(_airbyte_data AS NVARCHAR(MAX)) AS json_data
6     FROM {{ source('raw', 'raw_raw_stream_SupplierTransactions') }}
7   )
8
9   SELECT
10    CAST(JSON_VALUE(json_data, '$.SupplierTransactionID') AS INT) AS SupplierTransactionID,
11    CAST(JSON_VALUE(json_data, '$.SupplierID') AS INT) AS SupplierID,
12    CAST(JSON_VALUE(json_data, '$.TransactionTypeID') AS INT) AS TransactionTypeID,
13    CAST(JSON_VALUE(json_data, '$.PurchaseOrderID') AS INT) AS PurchaseOrderID,
14    CAST(JSON_VALUE(json_data, '$.PaymentMethodID') AS INT) AS PaymentMethodID,
15    CAST(JSON_VALUE(json_data, '$.SupplierInvoiceNumber') AS NVARCHAR(20)) AS SupplierInvoiceNumber,
16    CAST(JSON_VALUE(json_data, '$.TransactionDate') AS DATE) AS TransactionDate,
17    CAST(JSON_VALUE(json_data, '$.AmountExcludingTax') AS DECIMAL(18, 2)) AS AmountExcludingTax,
18    CAST(JSON_VALUE(json_data, '$.TaxAmount') AS DECIMAL(18, 2)) AS TaxAmount,
19    CAST(JSON_VALUE(json_data, '$.TransactionAmount') AS DECIMAL(18, 2)) AS TransactionAmount,
20    CAST(JSON_VALUE(json_data, '$.OutstandingBalance') AS DECIMAL(18, 2)) AS OutstandingBalance,
21    CAST(JSON_VALUE(json_data, '$.FinalizationDate') AS DATE) AS FinalizationDate,
22    CAST(IIF(JSON_VALUE(json_data, '$.FinalizationDate') IS NOT NULL, 1, 0) AS BIT) AS IsFinalized,
23    CAST(JSON_VALUE(json_data, '$.LastEditedBy') AS INT) AS LastEditedBy,
24    CAST(JSON_VALUE(json_data, '$.LastEditedWhen') AS DATETIME2(7)) AS LastEditedWhen
25   FROM source_data;
26

```

Figure 42: dbt Script for Supplier Transaction Raw to Normalise Table

For the Supplier Transactions tables and the other 17 tables involved in this project, dbt scripts were developed to parse JSON data from Airbyte's raw output and systematically create

normalized tables in the data mart's stage schema. This schema effectively serves the role of a sophisticated Bronze layer, where data is collected and prepared for subsequent processing stages.

Figure 42 showcases a snippet from Visual Studio Code, illustrating the dbt script used for transforming raw JSON data into a normalized table for Supplier Transactions. In Figure 43, the script on the right-hand side lists all raw source tables, indicating a comprehensive setup that includes transformation logic for each table. All scripts and related files are shared with the project documentation as references for further review and replication.

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER:** Shows the project structure under "BI ASSIGNMENT PURCHASE". Key folders include "wwi_intermediate_DW", "wwi_raw_stage", "models", and "src.yml".
- CODE EDITOR:** Displays a dbt script named "validate_and_transform_transaction_types.sql". The script content is as follows:

```

version: 2
sources:
  - name: raw
    database: WideWorldImportersPurchase
    schema: stage
    tables:
      - name: raw_raw_stream_Suppliers
      - name: raw_raw_stream_Cities
      - name: raw_raw_stream_StockItemTransactions
      - name: raw_raw_stream_CustomerTransactions
      - name: raw_raw_stream_PaymentMethods
      - name: raw_raw_stream_PurchaseOrders
      - name: raw_raw_stream_Countries
      - name: raw_raw_stream_SupplierTransactions
      - name: raw_raw_stream_TransactionTypes
      - name: raw_raw_stream_StockItems
      - name: raw_raw_stream_Customers
      - name: raw_raw_stream_PurchaseOrderLines
      - name: raw_raw_stream_StockItemHoldings
      - name: raw_raw_stream_StateProvinces
      - name: raw_raw_stream_Colors
      - name: raw_raw_stream_DeliveryMethods
      - name: raw_raw_stream_PackageTypes
      - name: raw_raw_stream_Invoices

```

Figure 43: dbt Solution for Raw to Normalise

Monitoring Script Execution and Data Integrity

The execution of these transformation scripts is monitored to ensure accuracy and efficiency in data handling. Successful execution confirms that the scripts are correctly parsing and restructuring the raw data into the desired format without data loss or corruption.

The image below visualizes the successful execution of the dbt scripts, with logs indicating the seamless transition of data from its raw state to a more structured and normalized form within the Bronze layer.

```

BI_ASSIGNMENT_PURCHASE
> dbt_venv
> logs
> wwi_intermediate_DW
< wwi_raw_stage
> analyses
> dbt_packages
> logs
> macros
< models
  ! src.yml
  transform_Cities.sql
  transform_Colors.sql
  transform_Countries.sql
  transform_Customers.sql
  transform_CustomerTransactions.sql
  transform_DeliveryMethods.sql
  transform_Invoices.sql
  transform_PackageTypes.sql
  transform_PaymentMethods.sql
  transform_PurchaseOrderLines.sql
  transform_PurchaseOrders.sql
  transform_StateProvinces.sql
  transform_StockItemHoldings.sql
  transform_StockItems.sql
  transform_StockItemTransactions.sql
  transform_suppliers.sql
  transform_SupplierTransactions.sql
  transform_TransactionTypes.sql
> seeds
> snapshots
> target
> tests
  .gitignore
! dbt_project.yml
① README.md
> wwi_stage_intermediate

```

```

wwi_raw_stage > models > transform_CustomerTransactions.sql
1   { config(materialized='table', alias='CustomerTransactions') }
2
3   WITH source_data AS (
4     SELECT
5       CAST(_airbyte_data AS NVARCHAR(MAX)) AS json_data
6     FROM {{ source('raw', 'raw_raw__stream_CustomerTransactions') }}
7   )

```

TERMINAL

```

(dbt_venv) PS D:\workingFolder\dbt\bi_assignment_purchase\wwi_raw_stage> dbt run
00:39:31 Running with dbt=1.7.17
00:39:32 Registered adapter: sqlserver=1.7.4
00:39:32 [WARNING]: Configuration paths exist in your dbt_project.yml file which do not apply to any resources
There are 1 unused configuration paths:
- models.wwi_raw_stage.example
00:39:32 Found 18 models, 18 sources, 0 exposures, 0 metrics, 447 macros, 0 groups, 0 semantic models
00:39:32 Concurrency: 4 threads (target='dev')
00:39:32
00:39:32 1 of 18 START sql table model stage.Colors ..... [RUN]
00:39:32 2 of 18 START sql table model stage.Countries ..... [RUN]
00:39:32 3 of 18 START sql table model stage.CustomerTransactions ..... [RUN]
00:39:32 4 of 18 START sql table model stage.Customers ..... [RUN]
00:39:33 1 of 18 OK created sql table model stage.Colors ..... [OK in 1.30s]
00:39:33 5 of 18 START sql table model stage.DeliveryMethods ..... [RUN]
00:39:33 5 of 18 OK created sql table model stage.DeliveryMethods ..... [OK in 0.15s]
00:39:33 6 of 18 START sql table model stage.Invoices ..... [RUN]
00:39:34 3 of 18 OK created sql table model stage.CustomerTransactions ..... [OK in 2.28s]
00:39:34 7 of 18 START sql table model stage.PackageTypes ..... [RUN]
00:39:35 7 of 18 OK created sql table model stage.PackageTypes ..... [OK in 0.50s]
00:39:35 8 of 18 START sql table model stage.PaymentMethods ..... [RUN]
00:39:36 8 of 18 OK created sql table model stage.PaymentMethods ..... [OK in 1.02s]
00:39:36 9 of 18 START sql table model stage.PurchaseOrderLines ..... [RUN]
00:39:37 9 of 18 OK created sql table model stage.PurchaseOrderLines ..... [OK in 1.63s]
00:39:37 4 of 18 OK created sql table model stage.Customers ..... [OK in 5.44s]
00:39:37 10 of 18 START sql table model stage.PurchaseOrders ..... [RUN]
00:39:37 11 of 18 START sql table model stage.StateProvinces ..... [RUN]
00:39:38 6 of 18 OK created sql table model stage.Invoices ..... [OK in 4.43s]
00:39:38 12 of 18 START sql table model stage.StockItemHoldings ..... [RUN]
00:39:38 11 of 18 OK created sql table model stage.StateProvinces ..... [OK in 0.45s]
00:39:38 13 of 18 START sql table model stage.StockItemTransactions ..... [RUN]
00:39:38 10 of 18 OK created sql table model stage.PurchaseOrders ..... [OK in 0.47s]
00:39:38 14 of 18 START sql table model stage.StockItems ..... [RUN]
00:39:40 12 of 18 OK created sql table model stage.StockItemHoldings ..... [OK in 2.59s]
00:39:40 14 of 18 OK created sql table model stage.StockItems ..... [OK in 2.58s]
00:39:41 15 of 18 START sql table model stage.SupplierTransactions ..... [RUN]
00:39:41 16 of 18 START sql table model stage.TransactionTypes ..... [RUN]
00:39:41 16 of 18 OK created sql table model stage.TransactionTypes ..... [OK in 0.19s]
00:39:41 17 of 18 START sql table model stage.Cities ..... [RUN]
00:39:42 15 of 18 OK created sql table model stage.SupplierTransactions ..... [OK in 0.92s]
00:39:42 18 of 18 START sql table model stage.Suppliers ..... [RUN]
00:39:42 18 of 18 OK created sql table model stage.Suppliers ..... [OK in 0.19s]
00:39:45 2 of 18 OK created sql table model stage.Countries ..... [OK in 12.99s]
00:39:45 13 of 18 OK created sql table model stage.StockItemTransactions ..... [OK in 7.13s]
00:39:45 17 of 18 OK created sql table model stage.Cities ..... [OK in 4.12s]
00:39:45
00:39:45 Finished running 18 table models in 0 hours 0 minutes and 13.74 seconds (13.74s).
00:39:45 Completed successfully

```

OUTLINE

TIMELINE

Figure 44: dbt Execution Status for Raw Data

Verification of Data Transformation

To further validate the effectiveness of our data transformation efforts' effectiveness, we compare the record counts of raw tables and their corresponding normalized versions. This comparison helps to ensure that no data is lost during the transformation process and that the integrity of the data is maintained.

Airbyte Tables		dbt Tables	
Table_Name	Row_Count	Table_Name	Row_Count
stage.raw_raw_stream_Cities	37940	stage.Cities	37940
stage.raw_raw_stream_Colors	36	stage.Colors	36
stage.raw_raw_stream_Counties	190	stage.Counties	190
stage.raw_raw_stream_Customers	665	stage.Customers	665
stage.raw_raw_stream_CustomerTransactions	99906	stage.CustomerTransactions	99906
stage.raw_raw_stream_DeliveryMethods	10	stage.DeliveryMethods	10
stage.raw_raw_stream_Invoices	72466	stage.Invoices	72466
stage.raw_raw_stream_PackageTypes	14	stage.PackageTypes	14
stage.raw_raw_stream_PaymentMethods	4	stage.PaymentMethods	4
stage.raw_raw_stream_PurchaseOrderLines	8578	stage.PurchaseOrderLines	8578
stage.raw_raw_stream_PurchaseOrders	2125	stage.PurchaseOrders	2125
stage.raw_raw_stream_StateProvinces	53	stage.StateProvinces	53
stage.raw_raw_stream_StockItemHoldings	227	stage.StockItemHoldings	227
stage.raw_raw_stream_StockItems	227	stage.StockItems	227
stage.raw_raw_stream_StockItemTransactions	243220	stage.StockItemTransactions	243220
stage.raw_raw_stream_Suppliers	13	stage.Suppliers	13
stage.raw_raw_stream_SupplierTransactions	2497	stage.SupplierTransactions	2497
stage.raw_raw_stream_TransactionTypes	13	stage.TransactionTypes	13

Figure 45: Record Comparison - Airbyte & dbt

The first image compares the record numbers between the raw tables generated by Airbyte and the normalized tables transformed using dbt, confirming that the data volume remains consistent through the transformation process.

SQLQuery20.sql - S...ESKTOP\sachi (71)*											SQLQuery5.sql - SA...ESKTOP\sachi (78)*											SQLQuery1.sql - not connected*										
use [WideWorldImporters]											use [WideWorldImportersPurchase]											use [stage]										
select * from [Warehouse].StockItemHoldings;											select * from [stage].StockItemHoldings order by StockItemID;											select * from [stage].StockItemHoldings										
1	1	179929	L-1	171341	9.50	20	100	16	2016-06-30 12:00:00.0000000	1	1	179929	L-1	171341	9.50	20	100	16	2016-06-30 12:00:00.0000000	1	1	179929	L-1	171341	9.50	20	100	16	2016-06-30 12:00:00.0000000			
2	2	169589	L-1	161435	9.50	20	100	16	2016-06-30 12:00:00.0000000	2	2	169589	L-1	161435	9.50	20	100	16	2016-06-30 12:00:00.0000000	2	2	169589	L-1	161435	9.50	20	100	16	2016-06-30 12:00:00.0000000			
3	3	259510	L-2	246900	11.25	10	120	16	2016-06-30 12:00:00.0000000	3	3	259510	L-2	246900	11.25	10	120	16	2016-06-30 12:00:00.0000000	3	3	259510	L-2	246900	11.25	10	120	16	2016-06-30 12:00:00.0000000			
4	4	213256	L-3	202964	12.00	5	100	16	2016-06-30 12:00:00.0000000	4	4	213256	L-3	202964	12.00	5	100	16	2016-06-30 12:00:00.0000000	4	4	213256	L-3	202964	12.00	5	100	16	2016-06-30 12:00:00.0000000			
5	5	203951	L-3	194162	12.00	5	100	16	2016-06-30 12:00:00.0000000	5	5	203951	L-3	194162	12.00	5	100	16	2016-06-30 12:00:00.0000000	5	5	203951	L-3	194162	12.00	5	100	16	2016-06-30 12:00:00.0000000			
6	6	201784	L-3	192127	12.00	5	100	12	2016-06-30 07:00:00.0000000	6	6	201784	L-3	192127	12.00	5	100	12	2016-06-30 07:00:00.0000000	6	6	201784	L-3	192127	12.00	5	100	12	2016-06-30 07:00:00.0000000			
7	7	210308	L-3	200201	12.00	5	100	16	2016-06-30 12:00:00.0000000	7	7	210308	L-3	200201	12.00	5	100	16	2016-06-30 12:00:00.0000000	7	7	210308	L-3	200201	12.00	5	100	16	2016-06-30 12:00:00.0000000			
8	8	422537	L-3	401980	88.50	10	200	16	2016-06-30 12:00:00.0000000	8	8	422537	L-3	401980	88.50	10	200	16	2016-06-30 12:00:00.0000000	8	8	422537	L-3	401980	88.50	10	200	16	2016-06-30 12:00:00.0000000			
9	9	197464	L-3	187968	12.00	5	100	12	2016-06-30 07:00:00.0000000	9	9	197464	L-3	187968	12.00	5	100	12	2016-06-30 07:00:00.0000000	9	9	197464	L-3	187968	12.00	5	100	12	2016-06-30 07:00:00.0000000			
10	10	228050	L-3	217048	12.00	5	100	16	2016-06-30 12:00:00.0000000	10	10	228050	L-3	217048	12.00	5	100	16	2016-06-30 12:00:00.0000000	10	10	228050	L-3	217048	12.00	5	100	16	2016-06-30 12:00:00.0000000			

Figure 46: Side-by-Side Comparison of Data - Source & Bronze Layer

The second image provides a side-by-side comparison of the data from the source to the final tables in the Bronze layer, showcasing the accuracy and effectiveness of the dbt transformations. This visual confirmation ensures that the data not only retains its integrity but is also accurately reflected in the structured environment of the data warehouse.

5.6 Silver Layer

The Silver layer, positioned between the Bronze and Gold layers within the Medallion Architecture, plays a critical role in the data transformation. After data is initially ingested into the Bronze layer in its raw form, it progresses to the Silver layer, where further refinement and transformation take place. This middle layer is essential for enhancing data quality and preparing it for more complex analytics and reporting in the Gold layer.

In the Silver layer, data undergoes a series of transformations to clean, standardise, and validate the information to ensure it meets the organization's quality standards and is consistent across all sources. These transformations might include correcting data formats, enriching data with additional information, resolving discrepancies, and filtering out irrelevant or duplicate data entries. The Silver layer serves as a transitional stage where data is still mutable and adaptable, allowing for iterative refinement based on evolving business requirements or insights gained from initial analyses.

This layer not only improves the reliability and accuracy of the data but also structures it in a way that aligns with the business's analytical needs. By the time data moves to the Gold layer, it is in a state that readily supports decision-making processes, having been cleansed of inconsistencies and enriched with valuable context. The subsequent sections of this document will delve deeper into the validation logic employed within each script used in the Silver layer.

5.6.1 Validation for Cities

1. Validation of CityID and CityName:

- Checks for null values in both CityID and CityName fields to ensure that all entries have essential identifying information.
- Ensures that CityID can be correctly cast as an integer. This is important for maintaining data type consistency across the dataset.

2. Validation of Location Data:

- Check if the Location field is either null or contains a valid geography format. This is performed using the .STIsValid() method, which is specific to SQL Server geography data types.
- This validation ensures that any non-null geographic data entered into the database adheres to the correct format, thus preventing errors related to geographic data processing.

3. Validation of Date Fields (ValidFrom and ValidTo):

- Ensures that the ValidFrom date is logically consistent with the ValidTo date (i.e., ValidFrom should not be later than ValidTo).
- Check that the ValidFrom date is not set in the future relative to the current date (using GETDATE()), ensuring all date records pertain to the past or present, which is crucial for logical data consistency and usability.

4. Data Cleaning:

- Applies trimming functions (LTRIM and RTRIM) to the CityName field to remove any leading or trailing whitespace. This step is important for maintaining consistency and accuracy in string data, which may be used in searches or displayed in user interfaces.

5. Deduplication:

- Employs the ROW_NUMBER() function to assign a unique row number to each entry within groups defined by StateProvinceID and CityName, ordered by CityID.
- This deduplication step helps identify and retain only the first occurrence of a city within each state based on the CityID ordering, ensuring that the dataset contains unique city names within each state.

A significant observation was made regarding data loss in the ongoing efforts to refine the data transformation and validation processes for the Cities table. As detailed in the below figure, it was noted that the validation process led to the loss of 11,048 records. This loss was primarily attributed to NULL values in the LatestRecordedPopulation field.

The screenshot shows a SQL query window with two SELECT statements. The first statement counts rows in the [intermediate].[Cities] table, resulting in 26892 rows. The second statement counts rows in the [stage].[Cities] table, resulting in 37940 rows. The difference between these counts is 11,048, representing the data loss.

```
USE [WideWorldImportersPurchase]
GO

SELECT count(*)
FROM [intermediate].[Cities];
GO

SELECT count(*)
FROM stage.[Cities];
GO
```

(No column name)	1	26892
(No column name)	1	37940

Figure 47: Data Loss in Cities after Validation

Given the context of our data usage and the specific KPIs and reporting metrics requirements, the `LatestRecordedPopulation` field is not critical. Its primary function does not align closely with the core objectives of our data analysis, which focus on cities' geographical and administrative attributes rather than their demographic statistics. This realization prompted a strategic adjustment in our data validation approach. To mitigate the unintended data loss and maximize the utility of our dataset, the decision was made to adjust the validation script by removing the validation check for the `LatestRecordedPopulation` field. This adjustment will ensure that records are not excluded based solely on the absence of population data, thereby retaining a fuller dataset that remains highly relevant and valuable for our analytical needs.

```

wwi_stage_intermediate > models > validate_and_transform_cities.sql
  1  {{ config(materialized='table', alias='cities') }}
  2
  3  WITH source_data AS (
  4      SELECT
  5          CityID,
  6          CityName,
  7          StateProvinceID,
  8          Location,
  9          LatestRecordedPopulation,
 10         LastEditedBy,
 11         ValidFrom,
 12         ValidTo,
 13         -- Validate CityID and CityName for nulls and ensure CityID is an integer
 14         CASE
 15             WHEN CityID IS NOT NULL AND CityName IS NOT NULL AND TRY_CAST(CityID AS INT) IS NOT NULL THEN 1
 16             ELSE 0
 17         END AS is_valid,
 18         -- Validate Location for proper geography format if provided
 19         CASE
 20             WHEN Location IS NULL OR Location.STIsValid() = 1 THEN 1
 21             ELSE 0
 22         END AS is_valid_location,
 23         -- Validate ValidFrom and ValidTo for logical consistency and against future dates
 24         CASE
 25             WHEN ValidFrom <= ValidTo AND ValidFrom <= GETDATE() THEN 1
 26             ELSE 0
 27         END AS is_valid_dates,
 28         -- Ensure LatestRecordedPopulation is non-negative
 29         CASE
 30             WHEN LatestRecordedPopulation >= 0 THEN 1
 31             ELSE 0
 32         END AS is_valid_population
 33     FROM [stage].[Cities]
 34 ),
 35     valid_data AS (
 36     SELECT
 37         CityID,
 38         LTRIM(RTRIM(CityName)) AS CityName,
 39         StateProvinceID,
 40         Location,
 41         LatestRecordedPopulation,
 42         LastEditedBy,
 43         ValidFrom,
 44         ValidTo
 45     FROM source_data
 46     WHERE is_valid = 1 AND is_valid_location = 1 AND is_valid_dates = 1 AND is_valid_population = 1
 47 ),
 48     deduped_data AS (
 49     SELECT *,
 50         ROW_NUMBER() OVER (PARTITION BY StateProvinceID, CityName ORDER BY CityID) AS rn
 51     FROM valid_data
 52 )
 53     SELECT
 54         CityID,
 55         CityName,
 56         StateProvinceID,
 57         Location,
 58         LatestRecordedPopulation,
 59         LastEditedBy,
 60         ValidFrom,
 61         ValidTo
 62     FROM deduped_data
 63     WHERE rn = 1; -- This selects the first valid record of duplicates, ensuring unique city names within states
 64

```

Figure 48: Validation Script for Cities

5.6.2 Validation for State Provinces

1. Primary Key and Null Checks:

- Ensures that StateProvinceID, StateProvinceName, and CountryID are not null. These fields are essential for identifying records uniquely and correctly linking data across related tables.
- This check is crucial for maintaining referential integrity and avoiding issues that could arise from incomplete data.

2. Data Type Consistency:

- Verifies that StateProvinceID and CountryID can be correctly cast as integers. This validation step ensures that data types are consistent across the dataset, which is critical for maintaining database schema constraints and supporting accurate data operations.

3. Geography Data Integrity:

- Checks if the Border geography data is null or valid according to the validation method of the SQL Server geography data type.STIsValid().
- This ensures that any geographic data entered is structurally correct and usable for geographic information system (GIS) processes and analysis.

4. Range and Value Checks for Population:

- Validates that the LatestRecordedPopulation field is either null or a non-negative number. This prevents logically incorrect data (such as a negative population) from skewing demographic analyses and reports.

5. Temporal Validity:

- Confirms that the ValidFrom date is less than or equal to the ValidTo date, ensuring logical consistency in temporal data.
- Additionally, it checks that the ValidFrom date is not set in the future relative to the current date (GETDATE()), unless the ValidTo date is null, allowing for ongoing entries.

6. Deduplication Check:

- Implements a deduplication strategy by using the ROW_NUMBER() window function. This function assigns a unique row number to each record within partitions defined by CountryID and StateProvinceName, ordered by StateProvinceID.
- This step helps identify and retain only the first occurrence of a state province within each country, based on the StateProvinceID order. It ensures that the dataset contains unique state province names within each country.

5.6.3 Validation for Supplier Transactions

1. Uniqueness and Null Checks:

- Utilizes `ROW_NUMBER()` to ensure uniqueness within the `SupplierTransactionID`, especially considering updates or edits that might duplicate entries. The partition by `SupplierTransactionID` and order by `LastEditedWhen DESC` ensures that the most recent entry is considered first.
- This method helps identify and keep only the latest record for each `SupplierTransactionID`, thereby preventing duplication issues.

2. Data Type Validation:

- Checks if key fields (`SupplierTransactionID`, `SupplierID`, `TransactionTypeID`, `PurchaseOrderID`, `PaymentMethodID`, `AmountExcludingTax`, `TaxAmount`, `TransactionAmount`, and `OutstandingBalance`) are numeric, which is essential for maintaining data consistency and ensuring accurate calculations in financial reports.
- Validates that `TransactionDate` and `FinalizationDate` are not null, confirming the presence of essential temporal data for transaction tracking.

3. Referential Integrity Checks:

- Ensures that `SupplierID` exists in the `Suppliers` table, `TransactionTypeID` exists in the `TransactionTypes` table, and, if not null, that `PurchaseOrderID` exists in the `PurchaseOrders` table and `PaymentMethodID` in the `PaymentMethods` table.
- These checks confirm that all references in the transaction records are valid and correspond to existing records, which is crucial for maintaining the relational integrity of the database.

4. Financial and Logical Checks:

- Confirms that the sum of `AmountExcludingTax` and `TaxAmount` equals `TransactionAmount`, ensuring that the financial calculations within the transaction records are logically consistent.
- Verifies that `OutstandingBalance` is non-negative, which is important for the financial health of the transactions.
- Checks the finalization logic: if a transaction is finalized (`IsFinalized = 1`), then the `FinalizationDate` must be set, and the `OutstandingBalance` should be zero, confirming the transaction's completion and accuracy.

```

USE [WideWorldImportersPurchase]
GO

SELECT count(*)
FROM [intermediate].SupplierTransactions;
GO

SELECT count(*)
FROM stage.SupplierTransactions;

```

The screenshot shows two SQL queries in a query editor. The first query selects the count of rows from the [intermediate].SupplierTransactions table, resulting in 2117 rows. The second query selects the count of rows from the stage.SupplierTransactions table, resulting in 2497 rows. This indicates a significant discrepancy in the data being processed.

Figure 49: Data Loss for Supplier Transaction

In the process of validating the data for the `SupplierTransactions` and `CustomerTransactions` tables using dbt scripts, significant discrepancies were uncovered that led to the exclusion of numerous records. These discrepancies, identified through the stringent validation rules applied, highlight underlying issues in the transaction data that need to be addressed.

SupplierTransactionID	SupplierID	TransactionTypeID	PurchaseOrderID	PaymentMethodID	SupplierInvoiceNumber	TransactionDate	AmountExcludingTax	TaxAmount	TransactionAmount	OutstandingBalance	FinalizationDate	IsFinalized	LastEditedBy	LastEditedWhen
24_70501	4	7	NULL	4	NULL	2013-10-14	0.00	0.00	-1908165.00	0.00	2013-10-14	1	5	2013-10-14 09:00:00.0000000
24_88900	4	7	NULL	4	NULL	2013-10-07	0.00	0.00	-1858908.30	0.00	2013-10-07	1	7	2013-10-07 09:00:00.0000000
24_67454	4	7	NULL	4	NULL	2013-09-30	0.00	0.00	-1834151.10	0.00	2013-09-30	1	15	2013-09-30 09:00:00.0000000
24_85906	4	7	NULL	4	NULL	2013-09-23	0.00	0.00	-1762869.30	0.00	2013-09-23	1	12	2013-09-23 09:00:00.0000000
24_63016	4	7	NULL	4	NULL	2013-09-16	0.00	0.00	-1689009.60	0.00	2013-09-16	1	9	2013-09-16 09:00:00.0000000
24_60350	4	7	NULL	4	NULL	2013-09-09	0.00	0.00	-1615820.00	0.00	2013-09-09	1	17	2013-09-09 09:00:00.0000000
24_60350	4	7	NULL	4	NULL	2013-09-02	0.00	0.00	-1592291.10	0.00	2013-09-02	1	18	2013-09-02 09:00:00.0000000
24_59705	4	7	NULL	4	NULL	2013-08-26	0.00	0.00	-1536250.50	0.00	2013-08-26	1	8	2013-08-26 09:00:00.0000000
24_57242	4	7	NULL	4	NULL	2013-08-19	0.00	0.00	-1493436.00	0.00	2013-08-19	1	7	2013-08-19 09:00:00.0000000
24_55728	4	7	NULL	4	NULL	2013-08-12	0.00	0.00	-1445129.10	0.00	2013-08-12	1	3	2013-08-12 09:00:00.0000000
24_54051	4	7	NULL	4	NULL	2013-08-05	0.00	0.00	-1391040.00	0.00	2013-08-05	1	17	2013-08-05 09:00:00.0000000

Figure 50: Reason for Data Loss with Supplier Transaction

For the `SupplierTransactions` table, it was observed that 380 records failed to pass the financial logic validation. The primary reason for this failure is that these records exhibit financial discrepancies where both the `AmountExcludingTax` and `TaxAmount` are recorded as 0, yet the `TransactionAmount` shows a large negative number. This inconsistency suggests potential issues with how transactions are being recorded or processed in the source system.

5.6.4 Validation for Colors

1. Primary Key and Null Checks:

- Ensures that both `ColorID` and `ColorName` are not null. This step is crucial for maintaining the integrity of the dataset, ensuring that no essential identifiers or descriptors are missing from the records.

2. Data Type Consistency:

- Verifies that `ColorID` can be correctly cast as an integer. This check is vital for ensuring that the data types are consistent with the database schema, preventing errors during data operations that depend on type-specific functions.

3. Text and String Formatting:

- Checks that the length of `ColorName` does not exceed 4000 characters. This validation step is important for data consistency and to avoid exceeding field size limits that might be set at the database level.

4. Temporal Validity:

- Ensures that the `ValidFrom` date is less than or equal to the `ValidTo` date. This check is essential for maintaining logical consistency in temporal data, ensuring that the start date is not later than the end date.

5. Deduplication Logic

Applies two layers of deduplication using the `ROW_NUMBER()` window function:

- **By ColorName:** Ensures that each color name is unique within the dataset by assigning a row number to each entry partitioned by `ColorName` and ordered by `ColorID`. This step helps to prevent duplicate color names from appearing in the dataset.
- **By ColorID:** Confirms that each `ColorID` is unique by assigning a row number within entries partitioned by `ColorID` and ordered by `ColorID`. This ensures that there are no duplicate records for any given `ColorID`.

5.6.5 Validation for Countries

1. Primary Key and Null Checks:

- Ensures that `CountryID`, `CountryName`, and `IsoAlpha3Code` are not null. These fields are essential identifiers that must be present in every record to maintain data completeness and uniqueness.

2. Data Type Consistency and ISO Code Validation:

- Verifies that `CountryID` can be correctly cast as an integer, ensuring data type consistency across the dataset.
- Checks that the `IsoAlpha3Code` contains exactly 3 characters, aligning with international standards for country codes.
- Validates the `IsoNumericCode`, ensuring it is either null or within the valid range of 1 to 999, which is typical for numeric country codes.

3. Range and Value Checks for Population:

- Confirms that `LatestRecordedPopulation` is either null or a non-negative number. This validation ensures that population figures are logical and correctly recorded, preventing data entry errors that could impact demographic analyses.

4. Temporal Validity:

- Ensures that the `ValidFrom` date is on or before the `ValidTo` date, confirming the logical sequencing of time-bound data.
- Checks that the `ValidFrom` date is not set in the future compared to the current date (`GETDATE()`), unless the `ValidTo` is null, accommodating ongoing country entries.

5. Deduplication Logic

Implements deduplication strategies to ensure that each `CountryName` and `FormalName` appears only once within the dataset:

- Uses the `ROW_NUMBER()` window function to assign a unique row number to each entry within groups partitioned by `CountryName` and ordered by `CountryID`. This step helps identify and keep only the first occurrence of each country name, ensuring that the dataset contains unique names.
- Similarly, assigns row numbers to entries partitioned by `FormalName` and ordered by `CountryID` to ensure the uniqueness of formal names in the dataset.

5.6.6 Validation for Customer Transactions

1. Uniqueness and Null Checks:

- Uses the `ROW_NUMBER()` function to assign a unique row number to each record within groups defined by `CustomerTransactionID`, ordered by `LastEditedWhen` in descending order. This ensures that the most recent record for each transaction ID is selected, managing duplications effectively.

2. Data Type Validation:

- Checks if key transaction identifiers and amounts (`CustomerTransactionID`, `CustomerID`, `TransactionTypeID`, `InvoiceID`, `PaymentMethodID`, `AmountExcludingTax`, `TaxAmount`, `TransactionAmount`, `OutstandingBalance`) are numeric, which is crucial for maintaining data type consistency and ensuring database integrity.
- Verifies that `TransactionDate` is not null, which is essential for tracking when transactions occurred. Although the script currently does not validate the date format (`ISDATE(ct.TransactionDate) = 1` is commented out), it's typically used to ensure the date is in a valid format.

3. Referential Integrity Checks:

Ensures that references to other tables are valid:

- Verifies existence of `CustomerID` in the `Customers` table.

- Confirms that TransactionTypeID exists in the TransactionTypes table.
- Checks that if PaymentMethodID is not null, it exists in the PaymentMethods table. (Similar validation for InvoiceID is commented out but would ensure the invoice exists if it's specified.)

```
{
    config(materialized='table', alias='CustomerTransactions') {}

WITH validated_data AS (
    SELECT
        ct.CustomerTransactionID,
        ct.CustomerID,
        ct.TransactionTypeID,
        ct.InvoiceID,
        ct.PaymentMethodID,
        ct.TransactionDate,
        ct.AmountExcludingTax,
        ct.TaxAmount,
        ct.TransactionAmount,
        ct.OutstandingBalance,
        ct.FinalizationDate,
        ct.IsFinalized,
        ct.LastEditedBy,
        ct.LastEditedWhen,
        -- Uniqueness and Null Checks
        ROW_NUMBER() OVER (PARTITION BY ct.CustomerTransactionID ORDER BY ct.LastEditedWhen DESC) AS rn,
        -- Data Type Validation
        CASE WHEN ISNUMERIC(ct.CustomerTransactionID) = 1
            AND ISNUMERIC(ct.CustomerID) = 1
            AND ISNUMERIC(ct.TransactionTypeID) = 1
            AND (ct.InvoiceID IS NULL OR ISNUMERIC(ct.InvoiceID) = 1)
            AND (ct.PaymentMethodID IS NULL OR ISNUMERIC(ct.PaymentMethodID) = 1)
            AND ct.TransactionDate IS NOT NULL
            --AND ISDATE(ct.TransactionDate) = 1
            AND ISNUMERIC(ct.AmountExcludingTax) = 1
            AND ISNUMERIC(ct.TaxAmount) = 1
            AND ISNUMERIC(ct.TransactionAmount) = 1
            AND ISNUMERIC(ct.OutstandingBalance) = 1
            THEN 1 ELSE 0 END AS is_valid_types,
        -- Referential Integrity Checks
        CASE WHEN EXISTS (SELECT 1 FROM [stage].[Customers] WHERE CustomerID = ct.CustomerID)
            AND EXISTS (SELECT 1 FROM [stage].[TransactionTypes] WHERE TransactionTypeID = ct.TransactionTypeID)
            --AND (ct.InvoiceID IS NULL OR EXISTS (SELECT 1 FROM [stage].[Invoices] WHERE InvoiceID = ct.InvoiceID))
            AND (ct.PaymentMethodID IS NULL OR EXISTS (SELECT 1 FROM [stage].[PaymentMethods] WHERE PaymentMethodID = ct.PaymentMethodID))
            THEN 1 ELSE 0 END AS is_valid_references,
        -- Financial and Logical Checks
        CASE WHEN ct.AmountExcludingTax + ct.TaxAmount = ct.TransactionAmount
            AND ct.OutstandingBalance >= 0
            AND ((ct.IsFinalized = 1 AND ct.FinalizationDate IS NOT NULL AND ct.OutstandingBalance = 0) OR ct.IsFinalized = 0 )
            THEN 1 ELSE 0 END AS is_valid_financial_logic
    FROM [stage].[CustomerTransactions] ct
),
filtered_data AS (
    SELECT
        CustomerTransactionID,
        CustomerID,
        TransactionTypeID,
        InvoiceID,
        PaymentMethodID,
        TransactionDate,
        AmountExcludingTax,
        TaxAmount,
        TransactionAmount,
        OutstandingBalance,
        FinalizationDate,
        IsFinalized,
        LastEditedBy,
        LastEditedWhen
    FROM validated_data
    WHERE rn = 1
        AND is_valid_types = 1
        AND is_valid_references = 1
        AND is_valid_financial_logic = 1
)
SELECT * FROM filtered_data;
}
```

Figure 51: Validation for Customer Transactions

4. Financial and Logical Checks:

Confirms the financial accuracy and logic of the transaction records:

- Validates that the sum of AmountExcludingTax plus TaxAmount equals the TransactionAmount, ensuring the arithmetic correctness of transaction calculations.

- Ensures that the `OutstandingBalance` is non-negative, which is a logical check for financial transactions.
- Checks the finalization status of the transaction:
- If `IsFinalized` is set to 1 (true), then it verifies that the `FinalizationDate` is not null and the `OutstandingBalance` is zero, reflecting a properly closed transaction.
- If `IsFinalized` is 0 (false), the transaction is considered open, and no additional checks on `FinalizationDate` or `OutstandingBalance` are required for logical consistency.

5.6.7 Validation for Customers

1. Primary Key and Null Checks:

- Ensures that both `CustomerID` and `CustomerName` are not null to guarantee that each record can be uniquely identified and meaningfully represented.
- This step is critical to prevent records lacking essential identifiers from being included in the dataset.

2. Data Type and Range Checks:

- Confirms that `CustomerID` can be successfully cast as an integer, maintaining consistency in data types across the database.
- Validates that the `CreditLimit` is a non-negative decimal value, which is done by coalescing null values to 0 and ensuring it can be cast to a decimal.
- Ensures that `StandardDiscountPercentage` falls within a valid range of 0 to 100%, which is logical for percentage values.
- Checks that `PaymentDays` is a non-negative integer, ensuring logical consistency in terms describing credit terms.

3. Temporal Validity:

- Verifies that the `ValidFrom` date is not later than the `ValidTo` date, maintaining logical consistency in record validity periods.
- Confirms that the `AccountOpenedDate` is on or before the current date (`GETDATE()`), ensuring no future dates are entered that would conflict with real-time data processing.

4. Additional Transformations and Cleanup

- Imputation of null values for `CreditLimit` with 0 to maintain consistency and avoid calculation errors in downstream processes.
- Trimming whitespace from string fields like `CustomerName`, `PhoneNumber`, `FaxNumber`, etc., to ensure data uniformity and improve the accuracy of text-based queries.

5. Deduplication Strategy

- Applies a deduplication process using `ROW_NUMBER()` to manage potential duplicates:

- Assigns a row number within groups of records partitioned by `CustomerID` and ordered by `ValidFrom` in descending order.
- This approach ensures that if there are multiple entries for the same customer, only the most recent (based on `ValidFrom`) is retained in the final dataset.

5.6.8 Validation for Delivery Methods

1. Primary Key and Null Checks:

- Ensures that `DeliveryMethodID` and `DeliveryMethodName` are not null. This check is crucial to maintaining the integrity of primary identifiers and essential attributes, ensuring that each record can be uniquely and effectively identified.

2. Data Type Consistency:

- Verifies that `DeliveryMethodID` can be correctly cast as an integer. This validation is important for maintaining consistent data types across the database, facilitating accurate and efficient data processing.

3. Text and String Formatting:

- Checks that the length of `DeliveryMethodName` does not exceed 4000 characters. This step ensures that data conforms to database field size limitations and prevents potential errors related to excessive data length.

4. Temporal Validity:

- Ensures that the `ValidFrom` date is on or before the `ValidTo` date. This check is essential for maintaining logical consistency in the temporal data, confirming that the period of validity for each delivery method is chronologically coherent.

5. Additional Data Standardization

- The script standardizes the `DeliveryMethodName` by trimming any leading or trailing spaces and converting the text to uppercase. This transformation standardizes string formatting across the dataset, aiding in consistent data retrieval and comparison.

6. Deduplication Logic

Applies a dual-layered deduplication process using the `ROW_NUMBER()` function to manage potential duplicates effectively:

- **By DeliveryMethodID:** Ensures that each `DeliveryMethodID` is unique within the dataset by assigning a row number within groups of records partitioned by `DeliveryMethodID` and ordered by the same ID.
- **By DeliveryMethodName:** Assigns a row number within groups partitioned by `DeliveryMethodName` and ordered by `DeliveryMethodID`, ensuring that each delivery method name is unique.

5.6.9 Validation for Invoices

1. Primary Key and Null Checks:

- Ensures that `InvoiceID`, `CustomerID`, `InvoiceDate`, and `OrderID` are not null, affirming that each invoice record contains essential information for identification and linkage to other transaction-related data.

2. Data Type Consistency:

- Confirms that `InvoiceID` and `CustomerID` can be successfully cast as integers, which maintains data type consistency across the database and ensures that relational operations involving these IDs are error-free.

3. Text and String Formatting:

- Verifies that the length of the `CustomerPurchaseOrderNumber` does not exceed 4000 characters to ensure that data adheres to field size restrictions and maintains consistency in data formatting.

4. Temporal Validity:

- Checks that the `InvoiceDate` is on or before the `ConfirmedDeliveryTime`. If `ConfirmedDeliveryTime` is null, a default future date is used to validate the invoice date, ensuring that the temporal data within the records is logically consistent.

5. Business Logic Checks:

- Ensures that if an invoice is marked as a credit note (`IsCreditNote` = 1), it must have a non-null `CreditNoteReason`. This validation aligns with business rules that require a reason for issuing credit notes, enhancing data reliability and auditability.

6. Additional Adjustments and Considerations

- Default Value for `ConfirmedDeliveryTime`:** Assign a default future date if `ConfirmedDeliveryTime` is null, which helps handle potential data gaps and ensures continuity in validations involving date comparisons.
- String Standardization:** Applies trimming and conversion to uppercase for fields like `CreditNoteReason`, `Comments`, `DeliveryInstructions`, etc., to standardize text data, simplifying further text processing and analysis.

7. Deduplication Strategy

- Implements deduplication by using the `ROW_NUMBER()` function, ensuring each `InvoiceID` is unique within the dataset by assigning a row number within groups of records partitioned by `InvoiceID` and ordered by the `LastEditedWhen` in descending order.
- This strategy helps retain only the most recent record for each invoice when there are updates or corrections to the data, ensuring the dataset's accuracy and currency.

5.6.10 Validation for Package Types

1. **Primary Key and Null Checks:**
 - Ensures that `PackageTypeID` and `PackageName` are not null. This validation is crucial for maintaining the integrity of key identifiers and ensuring that each record can be uniquely identified.
2. **Data Type Consistency:**
 - Verifies that `PackageTypeID` can be correctly cast as an integer. This check is vital for maintaining consistent data types across the database, ensuring that relational operations involving these IDs are error-free.
3. **Text and String Formatting:**
 - Checks that the length of `PackageName` does not exceed 4000 characters. This step ensures that data conforms to database field size limitations and prevents potential errors related to excessive data length.
 - `PackageName` is trimmed of any leading or trailing spaces and converted to uppercase, standardising text data to improve readability.
4. **Temporal Validity:**
 - Ensures that the `ValidFrom` date is on or before the `ValidTo` date, confirming the logical consistency of the data's temporal aspects. This validation ensures that the durations represented in the data are chronologically coherent and accurate.
5. **Deduplication Strategy**

Applies a dual-layered deduplication process using the `ROW_NUMBER()` function to ensure the uniqueness of records:

 - **By PackageTypeID:** Assigns a row number to each record within groups partitioned by `PackageTypeID`, ordered by the same ID. This strategy helps to ensure that each package type ID is unique within the dataset.
 - **By PackageTypeName:** Similarly, assigns a row number to entries partitioned by `PackageName` and ordered by `PackageTypeID`. This helps prevent duplicate names in the dataset, ensuring that each package type name is unique.

5.6.11 Validation for Payment Methods

1. **Null Value and Data Type Checks:**
 - Ensures that `PaymentMethodID` and `PaymentMethodName` are not null, confirming that essential data is present in every record.
 - Verifies that `PaymentMethodID` can be successfully cast as an integer, ensuring that the payment method identifier adheres to the expected numeric data type, which is crucial for database integrity and relationships.
2. **Temporal Validity Check:**
 - Confirms that the `ValidFrom` date is on or before the `ValidTo` date, maintaining logical consistency within the temporal data. This check is essential to ensure that the duration of validity for each payment method is chronologically coherent.

3. Name Length Constraint Check:

- Validates that the length of `PaymentMethodName` does not exceed 50 characters. This constraint ensures that data adheres to predefined specifications, likely reflecting business rules or database design choices that optimize performance and storage.

4. Deduplication Strategy

- Implements a deduplication process using the `ROW_NUMBER()` function, which is critical for maintaining the uniqueness of payment method names within the dataset:
- Assigns a row number to each record within groups partitioned by `PaymentMethodName`, ordered by `PaymentMethodID`. This ordering ensures that the entry with the smallest `PaymentMethodID` (often indicating the first entry) is prioritized in cases where duplicate names exist.

5.6.12 Validation for Purchase Order Lines

1. Uniqueness and Null Checks:

- Utilizes `ROW_NUMBER()` to assign a unique row number to each record within groups defined by `PurchaseOrderLineID`, ordered by `LastEditedWhen` in descending order. This ensures that only the most recent record for each purchase order line ID is selected, which is crucial for handling updates and avoiding duplicates.

2. Data Type Validation:

- Ensures that key numeric fields (`PurchaseOrderLineID`, `PurchaseOrderID`, `StockItemID`, `OrderedOuters`, `ReceivedOuters`, and optionally `PackageTypeID`) are valid numbers. This check is essential for data integrity and proper relational database operations.
- Verifies that `LastReceiptDate` is not null, although the script currently does not explicitly validate the date format (`ISDATE(pol.LastReceiptDate) = 1` is commented out but would typically ensure the date is in a valid format).

3. Referential Integrity Checks:

- Confirms the existence of the `PurchaseOrderID` in the `PurchaseOrders` table, ensuring that each purchase order line is linked to a valid purchase order.
- Ensures that the `StockItemID` exists in the `StockItems` table, validating the linkage between purchase order lines and stock items.
- Checks that if `PackageTypeID` is provided, it exists in the `PackageTypes` table, thereby maintaining the integrity of packaging information associated with each order line.

4. Date Validity Checks:

- Validates that the `LastReceiptDate` is on or after the `OrderDate` of the corresponding purchase order. This ensures chronological consistency between when an order was placed and when goods were received, aligning with logical business processes and operational timelines.

5.6.13 Validation for Purchase Orders

1. Uniqueness and Null Checks:

- Uses `ROW_NUMBER()` to assign a unique row number to each record within groups defined by `PurchaseOrderID`, ordered by `LastEditedWhen` in descending order. This ensures that only the most recent record for each purchase order ID is selected, which is crucial for managing updates and maintaining data accuracy.

2. Data Type Validation:

- Ensures that `PurchaseOrderID` and `SupplierID` can be successfully cast as integers, confirming their validity and consistency with expected data types.
- Validates that `OrderDate` and `ExpectedDeliveryDate` are not null, although the script currently does not explicitly validate the date format (`ISDATE()` is commented out but would typically ensure the dates are in a valid format).
- Confirms that optional fields like `DeliveryMethodID` and `ContactPersonID`, if not null, can also be cast as integers. This maintains data type consistency and supports accurate relational database operations.

3. Referential Integrity Checks:

- Checks for the existence of `SupplierID` in the `Suppliers` table to ensure each purchase order is linked to a valid supplier.
- Verifies that `DeliveryMethodID`, if provided, exists in the `DeliveryMethods` table, thereby maintaining the integrity of linkage to delivery methods.
- Optionally, checks that `ContactPersonID`, if provided, exists in the `Contacts` table (currently commented out), ensuring that contact references are valid.

4. Date Validity Checks:

- Validates that the `OrderDate` is on or before the `ExpectedDeliveryDate`. This check ensures chronological consistency, confirming that orders are not expected to be delivered before they are placed, which aligns with logical business processes.

5.6.14 Validation for Stock Items

1. Null Checks and Primary Key Uniqueness:

- Ensures that `StockItemID` and `StockItemName` are not null, confirming the presence of essential data for each record.
- This validation is critical for maintaining the primary key integrity and ensuring that each stock item can be uniquely identified.

2. Data Type and Range Checks:

- Validates that `StockItemID` can be successfully cast as an integer to maintain data type consistency.

- Confirms that `UnitPrice` and `TaxRate` can be cast as decimals and checks that these values are non-negative, which is crucial for financial calculations.
- Ensures `LeadTimeDays` is a non-negative integer, which is logical for inventory and supply chain management.

3. Temporal Validity:

- Checks that `ValidFrom` is on or before `ValidTo`, ensuring chronological consistency across the data. This validation is essential to maintain the logical integrity of the temporal aspects of the data.

4. Referential Integrity and Foreign Key Checks:

- The script includes a placeholder for foreign key validations (assumed to be always valid in the script with `1 AS is_valid_fk`). In practice, this would involve checks to ensure that values like `SupplierID`, `ColorID`, `UnitPackageID`, and `OuterPackageID` correspond to valid entries in their respective reference tables.

5. Deduplication Strategy

Applies a deduplication strategy using the `ROW_NUMBER()` function:

- Assigns a row number within groups partitioned by `StockItemID` and ordered by `ValidFrom` in descending order. This ensures that if there are multiple records for the same stock item (due to updates or corrections), only the most recent record is retained in the final dataset.

5.6.15 Validation for Stock Item Holdings

1. Null and Primary Key Checks:

- Ensures that `StockItemID`, `QuantityOnHand`, and `LastCostPrice` are not null. These fields are critical as they represent the essential attributes of stock inventory records, including the item identifier and key financial data.

2. Data Type and Range Checks:

- Confirms that `StockItemID` can be successfully cast as an integer, maintaining consistency in data types across the database.
- Verifies that `QuantityOnHand` is a non-negative integer, which is logical for inventory quantities.
- Ensures that `LastCostPrice` can be cast as a decimal and is non-negative, reflecting valid cost pricing data.

3. Logical Consistency:

- Checks that `ReorderLevel` is less than or equal to `TargetStockLevel`. This validation is crucial for maintaining logical consistency in inventory management practices, ensuring that reorder levels are set appropriately relative to target stock levels.

4. Temporal Validity:

- Validates that the `LastEditedWhen` timestamp is on or before the current date (`GETDATE()`). This check ensures that the record editing dates are chronologically plausible and prevent future-dated records from being erroneously entered.

5. Deduplication Strategy

Implements a deduplication process using the `ROW_NUMBER()` function to manage potential duplicates:

- Assigns a row number within groups of records partitioned by `StockItemID` and `BinLocation`, ordered by `LastEditedWhen` in descending order. This approach ensures that only the most recent record is retained if there are multiple records for the same stock item and bin location (due to updates or corrections).

5.6.16 Validation for Stock Item Transactions

1. Primary Key and Null Checks:

- Ensures that essential fields (`StockItemTransactionID`, `StockItemID`, `TransactionTypeID`, and `TransactionOccurredWhen`) are not null. This step is critical to guarantee that each record can be uniquely identified and contains necessary transaction details.

2. Data Type and Range Checks:

- Confirms that `StockItemTransactionID`, `StockItemID`, and `TransactionTypeID` can be correctly cast as integers, maintaining data type consistency across the dataset.
- Validates that `Quantity` can be cast as a decimal, allowing for precision in transaction quantity reporting. The script notes the absence of a non-negative check on `Quantity`, which could be adjusted based on specific business rules (e.g., allowing negative quantities for returns).
- Ensures that the `TransactionOccurredWhen` timestamp is on or before the current date (`GETDATE()`), ensuring temporal accuracy and preventing future-dated records.

3. Referential Integrity and Logical Consistency:

Checks for the existence of related records in other tables to ensure all foreign keys in the transaction record are valid:

- Verifies `StockItemID` against the `StockItems` table to ensure the stock item exists.
- Confirms `TransactionTypeID` against the `TransactionTypes` table to validate the type of transaction.
- Optionally checks `CustomerID` against the `Customers` table, ensuring the customer exists if provided.
- Optionally checks `SupplierID` against the `Suppliers` table, ensuring the supplier exists if provided.
- Optionally checks `InvoiceID` against the `Invoices` table, validating the invoice exists if provided.

- Optionally checks PurchaseOrderID against the PurchaseOrders table, validating the purchase order exists if provided.

4. Deduplication Strategy

Implements a deduplication strategy using the `ROW_NUMBER()` function:

- Assigns a row number within groups of records partitioned by StockItemTransactionID and ordered by LastEditedWhen in descending order. This strategy ensures that only the most recent record is retained if there are multiple records for the same transaction ID (due to updates or corrections).

5.6.17 Validation for Suppliers

1. Primary Key and Null Checks:

- Ensures that SupplierID, SupplierName, and SupplierCategoryID are not null. These fields are essential for identifying suppliers uniquely and categorizing them correctly, so their presence is critical for maintaining data integrity.

2. Data Type and Range Checks:

- Confirms that SupplierID can be cast as an integer, ensuring consistency in the primary key's data type across the database.
- Validates that PaymentDays is a non-negative integer. This check ensures that the terms of payment are logical and correctly formatted, reflecting sound business practices.

3. Temporal Validity:

- Checks that ValidFrom is on or before ValidTo. This validation ensures the chronological accuracy of the validity period for supplier records, preventing logical inconsistencies in temporal data.

4. Additional Data Standardization

- To remove extraneous spaces, ensuring consistency and preventing discrepancies in string comparisons or searches, applies trimming to text fields like SupplierName, BankAccountName, PhoneNumber, WebsiteURL, etc..

5. Deduplication Strategy

Implements a deduplication strategy using the `ROW_NUMBER()` function:

- Assigns a row number within groups of records partitioned by SupplierID and ordered by ValidFrom in descending order. This ensures that only the most recent entry is retained if there are multiple entries for the same supplier (perhaps due to updates or corrections).

5.6.18 Validation for Transaction Types

1. Primary Key and Null Checks:

- Ensures that both `TransactionTypeID` and `TransactionTypeName` are not null. This step is essential to maintain the integrity of the dataset, ensuring every transaction type can be uniquely identified and described.

2. Data Type Consistency:

- Confirms that `TransactionTypeID` can be successfully cast as an integer, maintaining consistency in data types across the database. This check is critical for ensuring that foreign key relationships involving this field are stable and error-free.

3. Text and String Formatting:

- Validates that the length of `TransactionTypeName` does not exceed 50 characters. This limit ensures that data adheres to predefined specifications, likely reflecting business rules or database design choices that optimise performance and storage.

4. Temporal Validity:

- Checks that the `ValidFrom` date is on or before the `ValidTo` date. This validation is essential for ensuring that the record's active duration is logically coherent, preventing any transaction type from having an end date that precedes its start date.

5. Deduplication Strategy

Implements a deduplication process using the `ROW_NUMBER()` function:

- Assigns a row number within groups of records partitioned by `TransactionTypeName`, ordered by `TransactionTypeID`. This strategy helps ensure that if there are duplicate names for transaction types (possibly due to data entry errors or updates), only the first entry based on the transaction type ID is retained.

5.7 Gold Layer

The Gold layer is a crucial component of the data architecture, particularly within the context of a Medallion architecture, which typically includes Bronze, Silver, and Gold layers. Each layer progressively refines the data, with the Gold layer representing the most processed and business-ready data. This layer is specifically designed to support high-level analytical and decision-making processes.

Key Features of the Gold Layer:

- Highly Curated:** Data in the Gold layer undergoes extensive cleansing, transformation, and enrichment in the preceding Silver layer, ensuring it is of the highest quality and relevance.

- **Optimized for Analytics:** The structure and schema of the Gold layer are optimised for efficient querying and analysis, making it ideal for generating insights quickly and reliably.
- **Consolidated:** This layer typically consolidates data from multiple sources into a format that is directly usable by business users and decision-makers without further transformation.

For the WWI Purchasing department, the Gold layer serves as the foundation for various critical functions:

- **KPI Tracking:** It supports visualising and monitoring key performance indicators (KPIs) that reflect the purchasing operations' efficiency, cost-effectiveness, and overall performance.
- **Reporting:** The Gold layer provides a ready-to-use dataset for generating detailed reports on procurement activities, supplier performance, cost analysis, and other aspects crucial for the purchasing department.
- **Predictive Modeling:** Utilizing clean and processed data from the Gold layer enhances the accuracy and reliability of predictive models. These models can forecast future trends, demand, and supply chain issues, helping the department to make proactive decisions.

5.7.1 Dimension Customer

Mappings and Transformations:

1. Customer Identity and Core Attributes:

- `CustomerID` is mapped to `wwi_customer_id` to serve as a unique identifier for each customer within the transformed dataset.
- `CustomerName` is transformed to `customer`, simplifying the column name while maintaining its essential descriptive value.

2. Geographical Information:

- `city`, `province`, and `country` are derived by joining the `Customers` table with `Cities`, `StateProvinces`, and `Countries` respectively. This enriches each customer record with detailed geographical context, providing insights into the customer's location.
- The joins ensure that each customer's location details are accurately matched based on their respective city, state, and country IDs.

3. Categorical and Relationship Dimensions:

- `BillToCustomerID`, `CustomerCategoryID`, `BuyingGroupID`, and `PrimaryContactPersonID` are initially set to placeholders like `bill_to_customer_id`, `category_id`, `buying_group_id`, and `primary_contact_id`. These fields represent relational links to other entities or classifications within the business model (e.g., billing accounts, customer categories).

- In a fully developed model, these IDs can be further joined to their respective tables to fetch more descriptive labels or names, enhancing the readability and usability of the data.

4. Temporal Dimensions:

- `ValidFrom` and `ValidTo` are directly mapped without transformation but are crucial for understanding each customer record's temporal validity, indicating the data's active period.

5. Lineage Tracking:

- A lineage key is included (`lineage_key`) using a templated schema notation to track the source and transformation lineage of the data, aiding in data governance and auditability.

6. Unique Key Assignment:

- A unique `customer_key` is generated using the `ROW_NUMBER()` function, ordered by `wwi_customer_id`. This key is a synthetic primary key for potential use in fact tables or other analytic constructs requiring a surrogate key.

```

1  {{ config(materialized='table', alias='Customer') }}
2
3  WITH customer_data AS (
4      SELECT
5          c.[CustomerID] AS wwi_customer_id,
6          c.[CustomerName] AS customer,
7          c.BillToCustomerID AS bill_to_customer_id, -- placeholder for actual name
8          c.CustomerCategoryID AS category_id, -- placeholder for actual category
9          c.BuyingGroupID AS buying_group_id, -- placeholder for actual group
10         c.PrimaryContactPersonID AS primary_contact_id, -- placeholder for actual contact
11         city.[CityName] AS city,
12         sp.[StateProvinceName] AS province,
13         co.[CountryName] AS country,
14         c.PostalPostalCode AS postal_code, -- No addresses table available
15         c.[ValidFrom] AS valid_from,
16         c.[ValidTo] AS valid_to,
17         '{{ this.schema }}.CustomerKey' AS lineage_key
18     FROM {{ source('intermediate', 'Customers') }} c
19     LEFT JOIN {{ source('intermediate', 'Cities') }} city ON c.DeliveryCityID = city.CityID
20     LEFT JOIN {{ source('intermediate', 'StateProvinces') }} sp ON city.StateProvinceID = sp.StateProvinceID
21     LEFT JOIN {{ source('intermediate', 'Countries') }} co ON sp.CountryID = co.CountryID
22 )
23
24
25  SELECT
26      ROW_NUMBER() OVER (ORDER BY wwi_customer_id) AS customer_key,
27      wwi_customer_id,
28      customer,
29      bill_to_customer_id AS bill_to_customer, -- Set to ID as placeholder
30      category_id AS category, -- Set to ID as placeholder
31      buying_group_id AS buying_group, -- Set to ID as placeholder
32      primary_contact_id AS primary_contact, -- Set to ID as placeholder
33      city,
34      province,
35      country,
36      postal_code,
37      valid_from,
38      valid_to,
39      lineage_key
40  FROM customer_data

```

Figure 52: Dimension Creation for Customer

5.7.2 Fact Purchase

Key Transformations and Mappings in the dbt Script:

1. Source Data Integration:

- **PurchaseOrders:** Pulls essential fields from the `PurchaseOrders` table such as `PurchaseOrderID`, `SupplierID`, `OrderDate`, etc. These fields are aliased to more descriptive names and include lineage tracking.
- **PurchaseOrderLines:** Retrieves details from the `PurchaseOrderLines` table, including the `StockItemID`, quantity details, and joins with the `PackageTypes` table to enrich the line items with package type descriptions.

2. Joining Orders and Lines:

- Combines data from the `PurchaseOrders` and `PurchaseOrderLines` through a join on `PurchaseOrderID`, creating a unified view that includes both order-level and line-item level details.
- Calculates the `ordered_quantity` by multiplying `OrderedOuters` with `ExpectedUnitPricePerOuter`, providing a monetary value for the ordered items, assuming a straightforward cost calculation without discounts or tax considerations.

3. Integration with Dimension Tables:

- **Date Dimension:** Joins with a date dimension table to translate the `OrderDate` into a surrogate key (`order_date_key`), facilitating time-based analysis and enabling the use of time intelligence functions in BI tools.
- **Stock Item Dimension:** Maps `StockItemID` to a dimension surrogate key (`stock_item_key`), enhancing the ability to slice and dice data based on stock item attributes stored in the dimension table.
- **Supplier Dimension:** Links `SupplierID` to a dimension surrogate key (`supplier_key`), which allows for aggregating and analyzing data based on supplier characteristics.

4. Final Fact Table Construction:

- Uses a `ROW_NUMBER()` window function to generate a unique `purchase_key` for each record, based on ordering by `order_date_key`, `stock_item_key`, and `supplier_key`. This key serves as a unique identifier within the fact table.
- Selects and renames fields to align with the dimensional model's requirements, ensuring that each column clearly represents its content and purpose.

5. Output Selection:

- The final `SELECT` operation extracts fields that constitute the fact table, including keys linking to dimensions (`date_key`, `supplier_key`, `stock_item_key`), quantitative metrics (`OrderedOuters`, `ordered_quantity`, `ReceivedOuters`), and descriptors (`package`).

- Includes transaction state (`IsOrderFinalized`) and lineage information (`lineage_key`) to provide context for the data and support traceability.

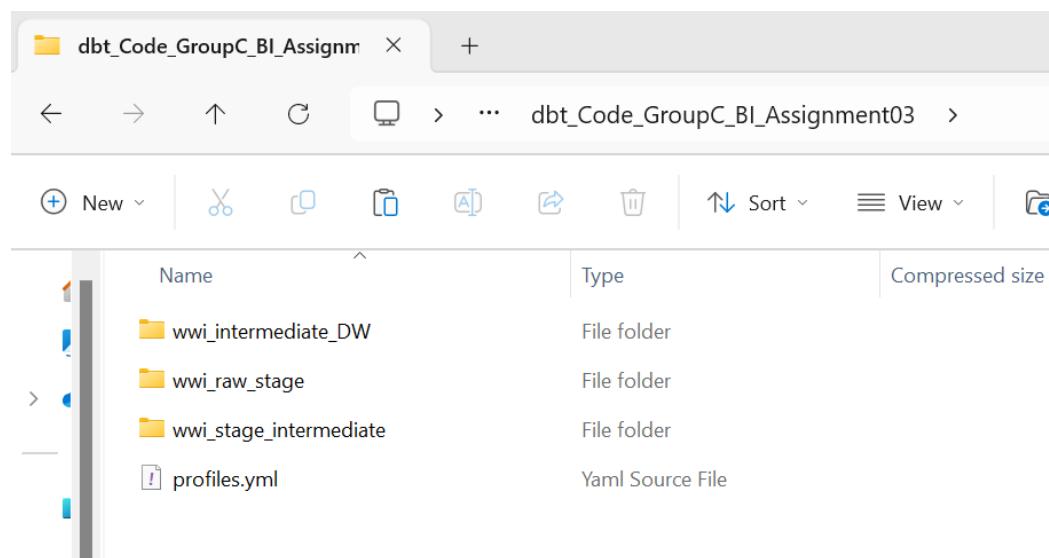
```
wwi_intermediate_DW > models > fact_purchase.sql
 2
 3
 4 WITH PurchaseOrders AS (
 5     SELECT
 6         po.PurchaseOrderID AS wwi_purchase_order_id,
 7         po.SupplierID AS supplier_id,
 8         po.OrderDate,
 9         po.ExpectedDeliveryDate,
10         po.IsOrderFinalized,
11         po.LastEditedBy AS lineage_key
12     FROM {{ source('intermediate', 'PurchaseOrders') }} AS po
13 ),
14 +
15 PurchaseOrderLines AS (
16     SELECT
17         pol.PurchaseOrderID,
18         pol.StockItemID AS stock_item_id,
19         pol.OrderedOuters,
20         pol.Description,
21         pol.ReceivedOuters,
22         pol.PackageTypeID,
23         pol.ExpectedUnitPricePerOuter,
24         pol.LastEditedBy AS pol_lineage_key,
25         pt.PackageTypeName AS package
26     FROM {{ source('intermediate', 'PurchaseOrderLines') }} AS pol
27     LEFT JOIN {{ source('intermediate', 'PackageTypes') }} AS pt ON pol.PackageTypeID = pt.PackageTypeID
28 ),
29
30 JoinedOrders AS (
31     SELECT
32         po.wwi_purchase_order_id,
33         po.supplier_id,
34         po.OrderDate,
35         po.IsOrderFinalized,
36         po.lineage_key AS po_lineage_key,
37         pol.stock_item_id,
38         pol.OrderedOuters,
39         pol.ReceivedOuters,
40         pol.package,
41         pol.ExpectedUnitPricePerOuter,
42         pol.pol_lineage_key,
43         (pol.OrderedOuters * pol.ExpectedUnitPricePerOuter) AS ordered_quantity -- Assuming 'expected_units_per_outer'
44     FROM PurchaseOrders AS po
45     JOIN PurchaseOrderLines AS pol ON po.wwi_purchase_order_id = pol.PurchaseOrderID
46 ),
47
48 DimensionIntegration AS (
49     SELECT
50         jo.wwi_purchase_order_id,
51         d.Date AS order_date_key,
52         si.stock_item_key,
53         s.supplier_key,
54         jo.OrderedOuters,
55         jo.ReceivedOuters,
56         jo.ordered_quantity,
57         jo.package,
58         jo.IsOrderFinalized,
59         jo.po_lineage_key AS lineage_key
60     FROM JoinedOrders AS jo
61     LEFT JOIN {{ source('Dimension', 'Date') }} AS d ON d.Date = jo.OrderDate
62     LEFT JOIN {{ source('Dimension', 'Stock Item') }} AS si ON si.wwi_stock_item_id = jo.stock_item_id
63     LEFT JOIN {{ source('Dimension', 'Supplier') }} AS s ON s.wwi_supplier_id = jo.supplier_id
64 )
```

Figure 53: Fact Purchasing

6 Appendix

6.1 dbt Script Files

dbt_Code_GroupC_BI_Assignment03.zip is submitted.



The file dbt_Code_GroupC_BI_Assignment03.zip has been successfully submitted. Inside the zip file, you will find the following folder structure:

- **profile.yml**: Contains all the connection configurations used for the project.
- **Bronze layer transformation**: The folder wwi_raw_stage includes scripts for transforming raw Airbyte data into a normalized structure (tables).
- **Silver layer transformation**: The folder wwi_stage_intermediate contains scripts for data validation.
- **Gold layer transformation**: The folder wwi_intermediate_DW holds scripts dedicated to creating the star schema.

Name	Type	Compressed size	Password pr...	Size	Ratio	Date modified
dim_customer.sql	ssms.sql.19.0	1 KB	No	2 KB	66%	28-Jun-24 14:10
dim_payment_method.sql	ssms.sql.19.0	1 KB	No	1 KB	51%	28-Jun-24 04:45
dim_stock_item.sql	ssms.sql.19.0	1 KB	No	2 KB	66%	28-Jun-24 13:46
dim_supplier.sql	ssms.sql.19.0	1 KB	No	2 KB	64%	28-Jun-24 14:39
dim_transaction_type.sql	ssms.sql.19.0	1 KB	No	1 KB	62%	28-Jun-24 10:07
fact_movement.sql	ssms.sql.19.0	1 KB	No	2 KB	71%	28-Jun-24 16:48
fact_purchase.sql	ssms.sql.19.0	1 KB	No	3 KB	73%	30-Jun-24 19:20
fact_purchase_order_summary.sql	ssms.sql.19.0	1 KB	No	2 KB	69%	28-Jun-24 18:01
fact_stock_holding.sql	ssms.sql.19.0	1 KB	No	2 KB	67%	28-Jun-24 17:03
fact_transaction.sql	ssms.sql.19.0	1 KB	No	4 KB	77%	28-Jun-24 18:05
src.yml	Yaml Source File	1 KB	No	1 KB	70%	28-Jun-24 16:12

6.2 SQL Script Files

The file `DB_Scripts.zip` includes two SQL files:

- **Create_Date_Dimension.sql**: This file is used to create the Date dimension.
- **Fact_Dimension_Constraints.sql**: This file is designed to implement table constraints in the final data warehouse.

Name
Create_Date_Dimension.sql
Fact_Dimension_Constraints.sql