Proposed Data Processing Solution Design for an Automotive Manufacturing Company

Joanne Senoren

Master of Science, Data Analytics

Table of Contents

Part I: Business Case Analysis

A. Business Case Overview

Precision Components, a manufacturer that provides parts throughout the supply chain for major automobile manufacturers, is acquiring a smaller manufacturing company. Both companies have home-grown systems, on-premise enterprise resource planning systems, and payroll systems. Precision Components, Inc. must find a seamless way to transition the data from the company acquisition into their own data environment across individual system entities. At the same time, Precision Components wants to build a centralized database system with reporting capabilities.

The company requested several high-level features to ensure a smooth data merge from SmallFirm, Inc., into their data systems and build a centralized data warehouse. Data for both companies should be centralized and integrated from the home-grown systems, the ERP system, and the HR/payroll system. The overall data engineering solution should be able to address scalability needs in the future and provide dashboards and reporting that unifies all systems' data, such as integrating sales data across various departments for an overall view of sales performance. Finally, the solution should also comply with privacy and data regulatory standards.

1. Problem Statement

In summary, the problem statement based on the overview above describes a critical point in data management and growth for Precision Components, Inc concerning isolated departmental and system data silos highlighted by the acquisition of SmallFirm, Inc. Integrating systems and new data from SmallFirm, Inc requires a data engineering solution that also addresses

centralization, scalability, automated updates based on schedules, thorough data cleaning, consolidated dashboards reports, and regulatory compliance.

2. Create Source to Target Mapping

Data mapping is a critical action point before enacting a data migration plan. It safeguards an accurate implementation by ensuring data quality, migration efficiency, maintaining data consistency, and eliminating data redundancy (Fatima, 2025). Given that we have virtually no information on Precision Components, Inc.'s data and attributes, let us proceed with the exercise assuming that they are like SmallFirm, Inc.'s attributes.

a. Define the Attributes

Table 1 below shows SmallFirm, Inc.'s source target columns and definitions based on the provided dictionary in Appendix A in the Business Scenario document.

Source Table	Source Column	Source Data Type	Source Description
Salary	ID	string	id of input row for salary
Salary	EmployeeID	string	employee id
Salary	Salary	number	value of salary amount
Salary	PayDate	datetime	date of salary paid
Personnel	ID	string	employee id
Personnel	First Name	string	employee first name
Personnel	Last Name	string	employee last name
Personnel	Position	string	employee position or title
Personnel	HireDate	string	employee hire date
Vendors	ID	string	vendor id
Vendors	Name	string	vendor company name
Vendors	AccountRep	string	vendor account representative name
Vendors	Status	string	vendor working status with company
Products	ProductID	string	product id
Products	Name	text	product name
Products	Cost	number	product cost
Products	SalePrice	number	product selling price
Product Batch Details	BatchNumber	GUID	product batch id
Product Batch Details	ProductionDate	datetime	production date

Product Batch Details	ProductID	text	product id of batched products
Product Batch Details	QuantityProduced	number	quantity of product batches
Product Batch Details	Year from File Name	text	year derived from the file name
Tooling Inventory	ID	text	tooling id
Tooling Inventory	ToolName	text	tool name
Tooling Inventory	Quantity	number	quantity
Tooling Inventory	Location	text	location of tool
Tooling Inventory	Cost	number	cost of tool
Tooling Inventory	RestockLevel	number	level when restock must be made
Tooling Inventory	VendorID	text	vendor id for the tool
Raw Materials Inventory	ID	text	raw materials id
Raw Materials Inventory	MaterialName	text	material name
Raw Materials Inventory	Quantity	number	quantity of available raw materials in inventory
Raw Materials Inventory	Unit	text	quantity of units of raw materials in inventory
Raw Materials Inventory	RestockLevel	number	level when raw materials restock must be made
Raw Materials Inventory	VendorID	text	raw materials vendor id
Products	RawMaterials	text	raw materials used to make product
Products	ProductID	string	product id
Products	Tooling	text	tooling used to make product
Products	ProductID	string	product id

The source target entities give rise to several questions about the data. Some considerations are not limited to, but include the following:

- What are the specific attributes within Precision Component's ERP system and HR/Payroll system?
- Are the personnel data and records considered personal identifiable information, and are there security or authentication measures that need to be acquired to access the information?
- What type of data format in Raw Materials and Tooling attributes within the Product table may affect entity relationships for the target columns?

- What are the required formats for all values and types?
- What is the frequency of SmallFirm, Inc.'s data updates?
- Are there default values implemented in specific columns?

Some of these questions can be answered if complete documentation of the source data is available. An ideal situation would include conferring with data owners and custodians to validate the source and target data, ensuring documentation and accurate information (Yaddow, 2019). For example, meeting with the IT team to get information on systems, assets, retention policy, archives, backup system, outsourcing, or data management is imperative to understanding the source and target environments (Yaddow, 2019). Meeting with team leaders and general data owners about their needs will provide more nuance in prioritizing specific data (Yaddow, 2019).

c. Map the Attributes

Project managers and relevant stakeholders should gather to provide key information on what should be included in data mapping (Yaddow, 2019). Examples include some of the questions listed above, such as consideration of which data sources will have accessibility parameters due to privacy constraints, and what should be protected (Yaddow, 2019). We will move forward with cautionary assumptions for this data mapping exercise since not all information is readily available.

We must consider two types of targets for SmallFirm's data. The first one is the specific request that data from SmallFirm, Inc. must be cleaned and merged into Precision Component's ERP databases and Oracle system for operational purposes. The Personnel and Payroll data will be mapped and ingested into the Oracle HR/Payroll system. Consequently, the Vendors and Products data will be cleansed and integrated into the SQL Server-based ERP system. This action

allows for business continuity while merging SmallFirm, Inc.'s data. We leave out Product Batch, Raw Materials, and Tooling since ERP systems are for OLTP purposes and these data are typically used for OLAP purposes. We can store them in the proposed cloud data warehouse instead.

Second, as a long-term data engineering solution, all data from SmallFirm, Inc. and Precision Components, Inc. will be centralized in a cloud data warehouse. All entities will be transformed and loaded into the new cloud data warehouse, which becomes the main analytics and report generation resource.

Table 2 below shows an overview of the file named SmallFirm_Data_Mapping, which provides the source-to-target data map for the cloud data warehouse. A similar process should be completed for Precision Components, Inc.'s source data. For the sake of the exercise, let us presume that Precision Components, Inc. has the same data attributes as SmallFirm, Inc.

The columns 'Source Table', 'Source Column', and 'Source Data Type' reiterate the information from Appendix A. The blue columns are the target tables, attributes, data types, and constraints that reflect the proposed entities within the cloud warehouse schema once the source data has been cleansed, transformed, and ingested into the centralized data system.

Table 2

Small Firm, Inc. Data Mapping

Source Table	Source Column	Source Data Type	Target Table	Target Column	Target Data Type & Length	Target Description	Target Constraints	Target Format Required
Payroll	ID	string	payroll	payroll_id	varchar(255)	id of input row for salary	not null, unique, primary key	
Payroll	EmployeeID	string	payroll	employee_id	varchar(255)	employee id	not null, foreign key to employee.employee_id	
Payroll	Salary	number	payroll	salary	numeric(18,2)	value of salary amount	not null	two decimal points, ex. $50,000.00
Payroll	PayDate	datetime	payroll	pay_date	datetime	date of salary paid	not null	MM/DD/YYYY
Personnel	ID	string	employee	employee_id	varchar(255)	employee id	not null, unique, primary key	
Personnel	First Name	string	employee	first_name	varchar(255)	employee first name	not null	
Personnel	Last Name	string	employee	last_name	varchar(255)	employee last name	not null	
Personnel	Position	string	employee	position	varchar(255)	employee position or title	not null	
Personnel	HireDate	string	employee	hire_date	datetime	employee hire date	not null	MM/DD/YYYY
Vendors	ID	string	vendor	vendor_id	varchar(255)	vendor id	not null, unique, primary key	
Vendors	Name	string	vendor	vendor_name	varchar(255)	vendor company name	not null	
Vendors	AccountRep	string	vendor	account_rep_name	varchar(255)	vendor account representative name	not null	
Vendors	Status	string	vendor	status	varchar(255)	vendor working status with company	not null	
Products	ProductID	string	product	product_id	varchar(255)	product id	not null, unique, primary key	
Products	Name	text	product	product_name	varchar(255)	product name	not null	
Products	Cost	number	product	cost	numeric(18,2)	product cost	not null	two decimal points, ex. $50,000.00
Products	SalePrice	number	product	sale_price	numeric(18,2)	product selling price	not null	two decimal points, ex. $50,000.00
Product Batch Details	BatchNumber	GUID	productBatch	batch_id	GUID	product batch id	not null, unique, primary key	
Product Batch Details	ProductionDate	datetime	productBatch	production_date	datetime	production date	not null	MM/DD/YYYY
Product Batch Details	ProductID	text	productBatch	product_id	varchar(255)	product id of batched products	not null, foreign key to product.product_id	
Product Batch Details	QuantityProduced	number	productBatch	quantity_produced	bigint	quantity of product batches	not null	
Product Batch Details	Year from File Name	text	productBatch	batch_year	bigint	product batch year derived from file name	not null	YYYY
Tooling Inventory	ID	text	tool	tool_id	varchar(255)	tooling id	not null, unique, primary key	
Tooling Inventory	ToolName	text	tool	tool_name	varchar(255)	tool name	not null	
Tooling Inventory	Quantity	number	tool	quantity	bigint	quantity	not null	
Tooling Inventory	Location	text	tool	location	varchar(255)	location of tool	not null	
Tooling Inventory	Cost	number	tool	cost	numeric(18,2)	cost of tool	not null	
Tooling Inventory	RestockLevel	number	tool	restock_level	bigint	level when restock must be made	not null	
Tooling Inventory	VendorID	text	tool	vendor_id	varchar(255)	vendor id for the tool	not null, foreign key to vendor.vendor_id	
Raw Materials Inventory	ID	text	material	material_id	varchar(255)	raw materials id	not null, unique, primary key	
Raw Materials Inventory	MaterialName	text	material	material_name	varchar(255)	material name	not null	
Raw Materials Inventory	Quantity	number	material	quantity	bigint	quantity of available raw materials in inventory	not null	
Raw Materials Inventory	Unit	text	material	unit	varchar(255)	quantity of units of raw materials in inventory	not null	
Raw Materials Inventory	RestockLevel	number	material	restock_level	bigint	level when raw materials restock must be made	not null	
Raw Materials Inventory	VendorID	text	material	vendor_id	varchar(255)	raw materials vendor id	not null, foreign key to vendor.vendor_id	
			productMaterial	material_id, product_id)		columns combined to make composite key as PK	composite key, PK	
Products	RawMaterials	text	productMaterial	material_id	varchar(255)	raw materials used to make product	not null, foreign key to raw_material.id	
Products	ProductID	string	productMaterial	product_id	varchar(255)	product id	not null, foreign key to product.product_id	
			productTooling	(tool_id, product_id)		columns combined to make composite key as PK	composite key, PK	
Products	Tooling	text	productTooling	tool_id	varchar(255)	tooling used to make product	not null, foreign key to tooling.tooling_id	
Products	ProductID	string	productTooling	product_id	varchar(255)	product id	not null, foreign key to product.product_id	

To reiterate, an extensive data map such as Table 2 would require multi-departmental validation across different teams and roles at Precision Components, Inc. and SmallFirm, Inc. before a complete migration occurs to ensure documentation and mapping of all relevant attributes are correct and agreed upon.

For full details of the proposed source-to-target data map, please review the attached file named “SmallFirm_DataMapping.”

c. Describe Necessary Data Transformations

The major transformations for all target tables and data consider Precision Components’ overall data environment and optimization for a SQL Server-compatible cloud data warehouse. All data sources with string and text types are transformed to variable characters (VARCHAR(255)), which allocates space for data to make storage more efficient. Source data with number formats were changed to data type NUMERIC(18,2) in attributes such as salary and sale_price, which require float type numbers. Target attributes, including quantity_produced,

batch_year, quantity, and restock_level, have the BIGINT data type to accommodate whole numbers. These data type constraints help maintain data consistency while addressing storage optimizations (Chu, 2024). There are also currency and date format constraints, such as sale_price requiring two decimal points, and any datetime type should have a format of 'MM/DD/YYYY.'

All target table names are in camelCase, a preferred naming convention for tables in SQL Server. The attribute naming convention is in snake case to accommodate descriptive language. All 'Id' columns were changed to '[table]_id' to maintain consistency and clarity.

There are explicit primary key and foreign key assignments based on table relationships. For example, the vendor_id columns in the material and tool tables are a foreign key related to the vendor table's vendor_id column.

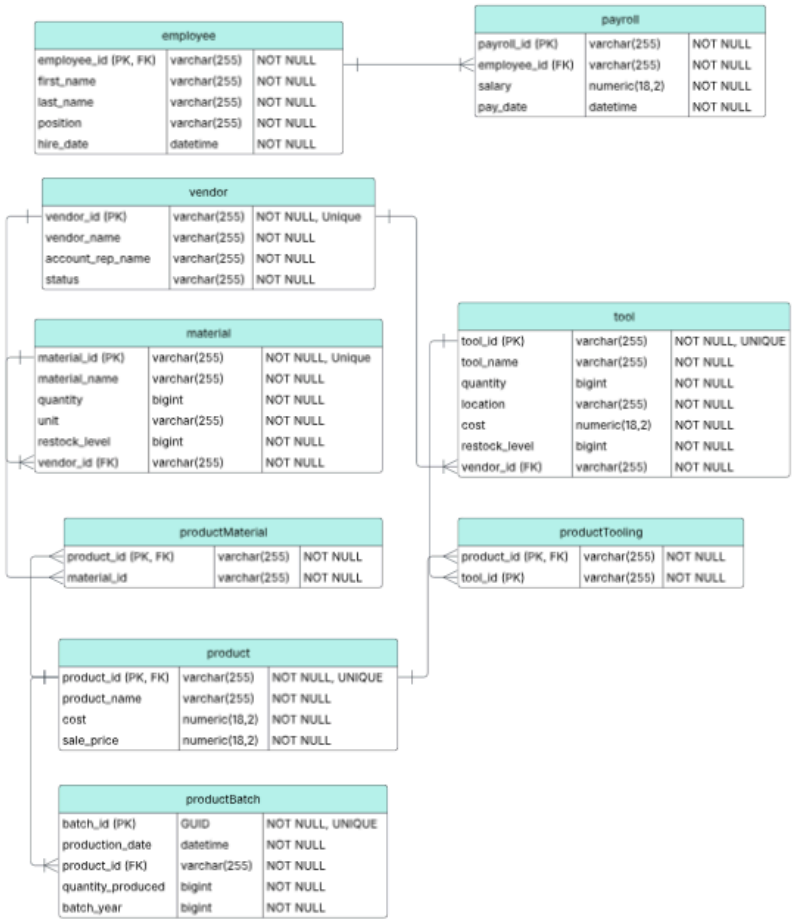
The most notable transformation is the creation of two link tables called 'productTooling' and 'productMaterial' to link the materials table and tooling table to the product table. They are normalized tables from the source attributes Products.RawMaterials and Products.Tooling. Given the limited information on these attributes, the text data could be single items, comma-separated items, or an array. To address this issue and accommodate scalability, these attributes are separated into their own tables, productMaterial and productTooling, respectively. They map back to the product table using the product ID as a foreign key. The primary key for these two tables is composite. For example, in the productMaterial table, the composite primary key is product_id and material_id. This improves data integrity by normalizing a many-to-many relationship.

The attribute batch_year is derived from the annual files provided by the Production Batch Details source table. The target table productBatch will consist of all data from each annual file, and the batch_year indicates which year the data came from. The year values from the separate files can be taken from the file name if labeled with the year or from the metadata.

Figure 1 shows a conceptual ERD of the entities and their relationships within the proposed cloud data warehouse once source data has been fully mapped and transformed into the target tables.

Figure 1

Conceptual Proposed ERD



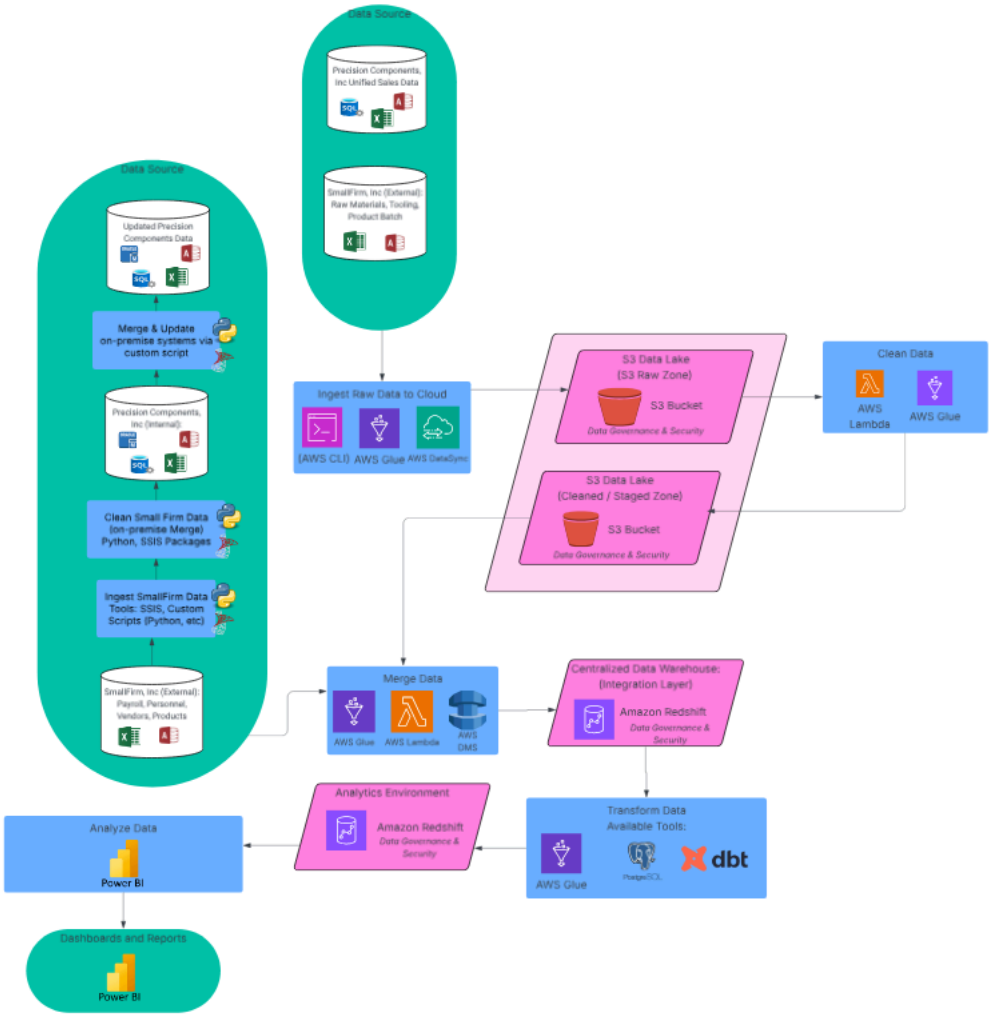
Part II: Recommended Design Solution

B. Data Engineering Design Outline

1. Process Flow Diagram

Figure 2

Process Flow Diagram



The process flow diagram in Figure 2 differentiates itself from the data flow diagram in Figure 3 to illustrate tools and software processes and emphasizes operational details (*Course* |, n.d.). It represents the major components of the entire data pipeline and their relationships to each other (*Course* |, n.d.). The diagram illustrates what parts and software are used for

integrating, processing, and analyzing data as Precision Components, Inc. requested. The process flow diagram starts with diverse raw data sources, including unified siloed sales data, external raw data from SmallFirm, Inc., and Precision Components, Inc.'s on-premise data. On-premise HR, products, and vendor data go through initial cleaning and merging with SmallFirm, Inc.'s Payroll, Personnel, Products, and Vendor data using tools such as Python, AWS Glue, and SSIS packages with scheduled updates to the on-premises system handled by custom scripts. These datasets go through a cleaning process and are merged.

Raw data from other sources, including SmallFirm's Product Batch, Materials, Tooling, and Precision Components, Inc.'s sales department data, are ingested directly into an Amazon S3 Data Lake raw zone to undergo a cleaning process that uses AWS CLI, AWS Glue, and AWS DataSync prior to being moved to a cleaned zone in S3. All data is merged into a centralized data warehouse within AWS Redshift. The data undergoes schema creation, normalization, data quality checks, and version-controlled transformations for analysis with the help of dbt, SQL, and AWS Glue. This allows users to access transformed data in a Redshift analytics environment that feeds Microsoft Power BI through built-in native connectivity, which allows business users to create dashboards and reports in a centralized space.

2. Data Flow Diagram

Figure 3

Data Engineering Solution L1 Data Flow Diagram

Figure 3 provides a Level 1 Data Flow Diagram (DFD) that describes Precision Components, Inc.'s data needs and specific requests. Data flow diagrams provide clarity on business operations and explain complicated data flow processes (Chi, 2024). The flowchart shows two main objectives: merging SmallFirm's data for immediate access and business continuity, and simultaneously establishing the centralized cloud-based engineering solution.

The diagram depicts the areas and processes of targeted data sources, including SmallFirm, Inc.'s external files, Precision Components, Inc.'s in-house ERP system, the Oracle HR System, and the Sales Department's siloed data. Each raw data source goes through an ingestion and staging process. Then the staged data is cleansed either for integration with current systems or for migration into the proposed distributed cloud-based data warehouse. The cloud-based warehouse should be designed to scale across different locations. Data transformation, which

includes normalization and schema creation, occurs before landing in the analytics environment for reporting purposes. Data consistency, privacy and governance compliance should be enforced throughout the pipeline for best practices. The final output of the diagram depicts prepared data for dashboards and reports to support OLAP purposes. The DFD symbols are based on Yourdon & De Marco notation.

Part III: Processing Design Evaluation

C. Define Design's Relevance to Business Scenario

1. Advantages of the Proposed Design

Precision Components, Inc. benefits from the specific engineering solution by leveraging the AWS suite of cloud services. The tools provide scalability and optimized performance for large-scale or incremental data processing. Since most of the tools reside within AWS, the company can seamlessly move from one environment to another without configuring connectivity manually. AWS also scales computing power and storage up and down dynamically, so the company can efficiently manage data costs.

Utilizing AWS Glue and Lambda lets Precision Components design with automation in mind. Data admin users can focus on monitoring data quality and scheduling trigger events for data processes instead of manually managing complicated scripts. Dbt within Redshift allows admins to manage version control, automate testing, generate documentation, and create SQL modules to manage data at the transformation stage better. Consolidated and referenceable module logic in dbt improves maintainability and productivity. Dbt also simplifies complicated scripted data mappings with CSV files called seeds, and dbt models can simply reference them (Manage Data Transformations With Dbt in Amazon Redshift | Amazon Web Services, 2022).

Most importantly, a self-service analytics space like Microsoft Power BI empowers non-technical users to harness data efficiently in business decision-making (Atlan, 2023). It allows teams to make decisions quickly by allowing them to focus on insights rather than accessing or organizing data for analytics (Atlan, 2023). Since most of the information lives in MS Access and Excel, Power BI also provides teams and users with a familiar environment to interact with the data.

Companies like Precision Components, Inc., with many moving pieces, benefit from a centralized system because it removes data siloes and allows data teams to have a unified view of all data and facilitates optimized business processes (Udt, 2025). A centralized data system supports task automation, reusability of data, simplifies data sharing, and provides a more productive work system with enhanced business intelligence and more informed decision-making (Udt, 2025).

2. Disadvantages of the Proposed Design

The engineering solution expects a high degree of AWS knowledge across its services. Since the proposed design utilizes many AWS suite tools, Precision Components, Inc. may need to hire a team of specialists to manage the entire system. This team would also need to create documentation and onboard other data specialists cross-functionally within the company. This could become a long-winded process of documentation, training, and onboarding across teams, potentially slowing down the company's centralization and migration goals.

Precision Components, Inc. wants to simultaneously integrate SmallFirm, Inc.'s data into their system and build a centralized database. This two-fold goal creates complexity that can easily cause data synchronization and latency issues. Given the numerous raw and siloed data sources, a more practical approach would include focusing on fully integrating and automating

SmallFirm's data first. Once the integration is stable, the company can consider establishing a centralized data warehouse. If Precision Components were to attempt both challenges at the same time, then it may be difficult to manage data consistency and connectivity within the AWS environment. Most notably, AWS's costs can quickly get out of control when multiple tools are 'turned on' for numerous actions. Running Glue jobs and executing Lambda Functions for ingestion, cleaning, and merging increases costs that quickly add up. They would require constant monitoring if services were continuously active for the two-fold purpose.

E. References (Citations)

3. Atlan, T. (2023, December 22). Self Service Analytics: What is It and Why is It Important? *Atlan*. https://atlan.com/what-is-self-service-analytics/#self-service-analytics-purpose-and-advantages
4. Chi, C. (2024, September 27). A Beginner's Guide to Data Flow Diagrams. *Hubspot*. https://blog.hubspot.com/marketing/data-flow-diagram#the-benefits-of-data-flow-diagrams
5. Chu, D. (2024, August 12). *In-Depth Guide to SQL Data Types: Differences and Best practices*. Secoda. https://www.secoda.co/learn/in-depth-guide-to-sql-data-types-differences-and-best-practices
6. *Course* |. (n.d.). https://apps.cgp-oex.wgu.edu/learning/course/course-v1:WGUx+OEX0420+v01/block-v1:WGUx+OEX0420+v01+type@sequential+block@77bacf8f30b4498f8fd5a696bb6cabdf/block-v1:WGUx+OEX0420+v01+type@vertical+block@bea700d62bbb43ad81deaf1873d18447

7. Fatima, N. (2025, February 4). Data Mapping 101: A Complete Guide. *Astera*. https://www.astera.com/type/blog/understanding-data-mapping-and-its-techniques/
8. *Manage data transformations with dbt in Amazon Redshift | Amazon Web Services*. (2022, August 3). Amazon Web Services. https://aws.amazon.com/blogs/big-data/manage-data-transformations-with-dbt-in-amazon-redshift/
9. Udt. (2025, May 14). *What are the benefits of centralized data management?* UDT. https://udtonline.com/what-are-the-benefits-of-centralized-data-management/
10. Yaddow, W. (2019). *The process of data mapping for data integration projects Data Mapping -A key work product for data warehouse, data integration, and data migration projects* [Online]. https://doi.org/10.13140/RG.2.2.10352.81925