A. Provide an overview of the business case by doing the following:
1. Provide a problem statement based on the attached "Data Engineering Scenario" document indicating the business requirements to be addressed by the new data engineering design.

Precision Components Inc. is undergoing a significant transformation due to the acquisition of SmallFirm, Inc. The merger presents challenges related to data integration, as the two companies utilize different and incompatible systems. Currently, Precision Components Inc. operates a mix of home-grown MS Access databases, Excel files, an on-premise SQL Server ERP system, and a hosted Oracle-based HR/payroll system. SmallFirm, Inc. primarily relies on MS Access databases and Excel files.
The company requires a modernized data engineering solution to address the following business requirements:

1. **Data Cleansing and Merging** – The existing data from SmallFirm, Inc. must be cleansed and merged into Precision Components Inc.'s ERP and HR/payroll systems to ensure consistency and accuracy.

2. **Data Ingestion** – The new system must support automated daily or nightly data ingestion from SmallFirm, Inc. into Precision Components Inc.'s systems.

3. **Data Integration** – A centralized data warehouse should be implemented to integrate data from home-grown systems, the on-premise ERP system, and the HR/payroll system.

4. **Dashboard and Reporting** – The company requires real-time dashboards and reports to provide insights into business operations.

5. **Scalability and Flexibility** – The solution must accommodate future growth, including international expansion and increased data volume.

6. **Unified Sales Data** – Sales departments currently operate in silos. A unified data model is required to integrate sales data for comprehensive performance analysis.

7. **Compliance and Security** – The data solution must comply with industry standards for security and regulatory compliance.

By addressing these requirements, Precision Components Inc. will enhance data-driven decision-making, streamline operations, and maintain a competitive edge in the automotive supply chain industry.

2. Create source-to-target data mapping by doing the following:
a. Define the attributes.
b. Map the attributes.
c. Describe necessary data transformations.

A. Define the Attributes

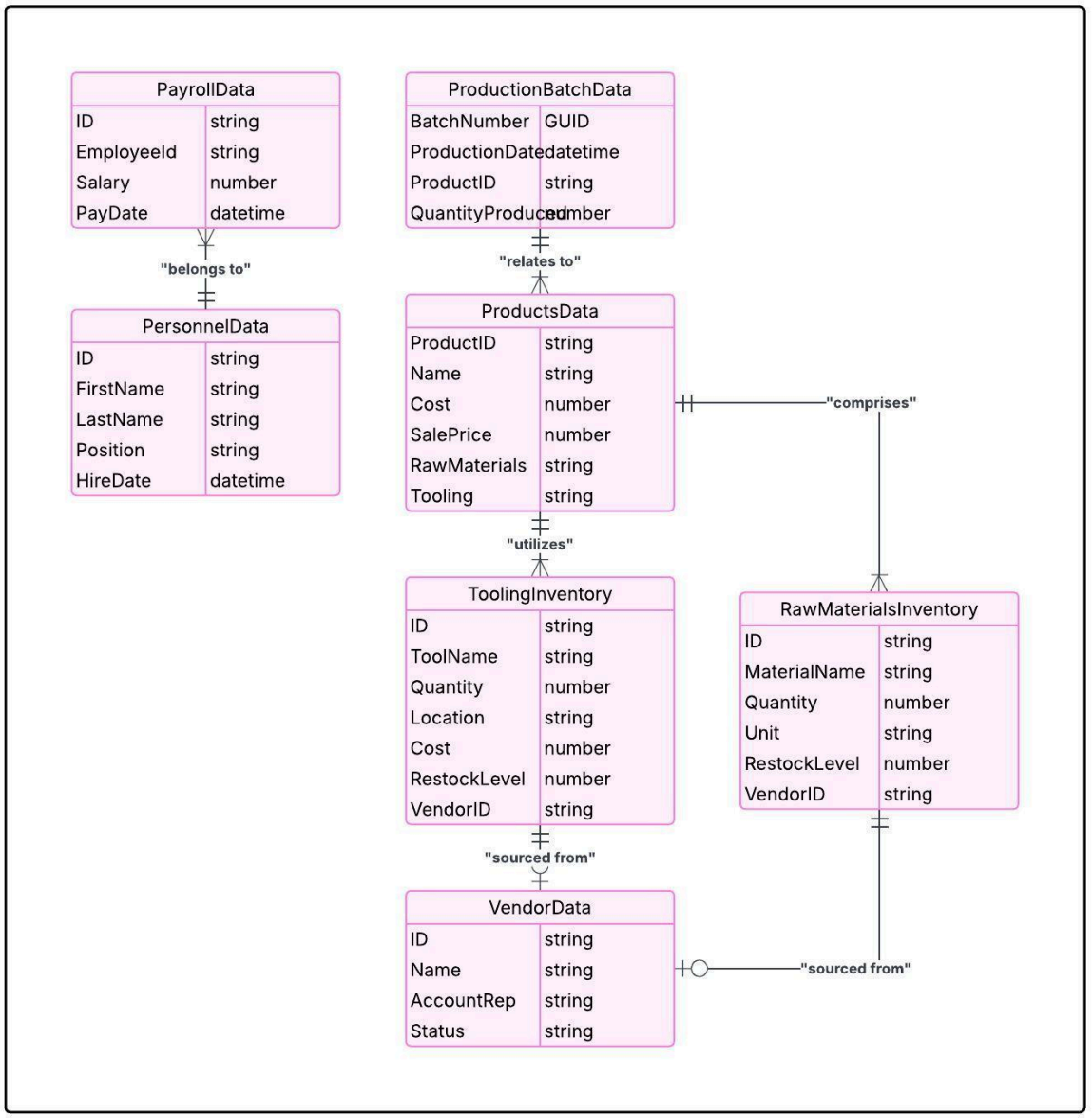
Source Data Attributes (SmallFirm, Inc.)
SmallFirm, Inc. currently stores data in MS Access and Excel files. The key data entities and their attributes are:

1. **Payroll Data (MS Access)**
    - **ID** (string) → Unique identifier
    - **EmployeeId** (string) → Employee unique ID
    - **Salary** (number) → Employee salary
    - **PayDate** (datetime) → Date of salary payment
2. **Personnel Data (MS Access)**
    - **ID** (string) → Unique identifier
    - **First Name** (string) → Employee first name
    - **Last Name** (string) → Employee last name
    - **Position** (string) → Job title
    - **HireDate** (datetime) → Date of hiring
3. **Vendor Data (MS Access)**
    - **ID** (string) → Vendor unique ID
    - **Name** (string) → Vendor name
    - **AccountRep** (string) → Account representative name
    - **Status** (string) → Vendor status (Active/Inactive)
4. **Products Data (Excel)**
    - **ProductID** (string) → Product unique ID
    - **Name** (string) → Product name
    - **Cost** (number) → Manufacturing cost
    - **SalePrice** (number) → Selling price
    - **RawMaterials** (string) → Materials used
    - **Tooling** (string) → Tools required
5. **Production Batch Data (Excel)**
    - **BatchNumber** (GUID) → Unique production batch ID
    - **ProductionDate** (datetime) → Date of production
    - **ProductID** (string) → Related product ID
    - **QuantityProduced** (number) → Number of items produced
6. **Tooling Inventory (Excel)**
    - **ID** (string) → Unique tooling ID
    - **ToolName** (string) → Tool name
    - **Quantity** (number) → Quantity available
    - **Location** (string) → Storage location
    - **Cost** (number) → Cost per unit
    - **RestockLevel** (number) → Threshold for reordering
    - **VendorID** (string) → Related vendor ID
7. **Raw Materials Inventory (Excel)**
    - **ID** (string) → Unique material ID

- **MaterialName** (string) → Name of raw material
- **Quantity** (number) → Quantity available
- **Unit** (string) → Measurement unit
- **RestockLevel** (number) → Threshold for reordering
- **VendorID** (string) → Related vendor ID

B. Map the Attributes to the Target System

The target system is a **centralized cloud-based data warehouse** (AWS Redshift, Google BigQuery, or Azure Synapse Analytics). Below is the **source-to-target attribute mapping**:



C. Necessary Data Transformations

The following transformations are required to standardize and cleanse the data:

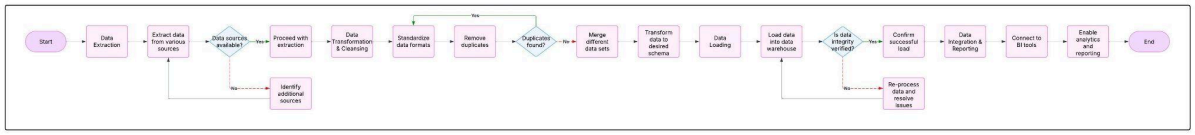
1. **Data Cleansing**
    - Remove duplicates and correct inconsistencies in Employee, Vendor, and Product IDs.
    - Standardize date formats across all datasets (e.g., convert `MM/DD/YYYY` to `YYYY-MM-DD`).
2. **Data Type Conversion**
    - Convert `ID` fields to **STRING** format for consistency.
    - Convert `Salary`, `Cost`, and `SalePrice` to **FLOAT**.
    - Convert `PayDate`, `HireDate`, and `ProductionDate` to **DATETIME**.
3. **Data Merging**
    - Merge **Payroll** and **Personnel** data using `EmployeeID` to form a complete employee profile.
    - Integrate **Tooling Inventory** and **Raw Materials Inventory** with **Vendors** using `VendorID`.
4. **Data Enrichment**
    - Generate unique **Batch_ID** using GUIDs to prevent conflicts.
    - Add calculated fields such as **Profit Margin (Selling Price - Manufacturing Cost)** in the `Products_Dim` table.
5. **Data Aggregation**
    - Summarize **monthly payroll expenses** in the `Payroll_Fact` table.
    - Compute **total sales per product** in the `Products_Dim` table.

By applying these transformations, the data warehouse will ensure accurate, reliable, and standardized data for reporting and analytics.

B. Outline a data engineering design that meets the business needs by doing the following:

1. Provide a process flow diagram based on the scenario.

2. Provide a Level 1 or higher data flow diagram showing the logical organization and movement of data for the proposed design as described in part B1.



The Process Flow Diagram visually represents how data is extracted, transformed, and loaded into the data warehouse for analysis. The key steps include:

1. **Data Extraction**
    - Extract data from SmallFirm, Inc.'s MS Access and Excel files.
    - Extract data from Precision Components Inc.’s ERP (SQL Server) and HR/Payroll system (Oracle).
    - Extract data from home-grown MS Access databases.
2. **Data Transformation & Cleansing**
    - Standardize data types, formats, and structures.
    - Remove duplicates, inconsistencies, and errors.
    - Merge Employee, Payroll, and Vendor data into a unified format.
3. **Data Loading**
    - Store processed data in a cloud-based Data Warehouse (e.g., AWS Redshift, Google BigQuery, or Azure Synapse Analytics).
    - Load cleaned data into separate fact and dimension tables for analytics.
4. **Data Integration & Reporting**
    - Connect the data warehouse to BI tools (e.g., Tableau, Power BI) for visualization.
    - Generate real-time dashboards and reports for business insights.

2. Level 1 Data Flow Diagram

The Level 1 Data Flow Diagram details the logical organization and movement of data across systems. The key components include:

Entities and Data Sources

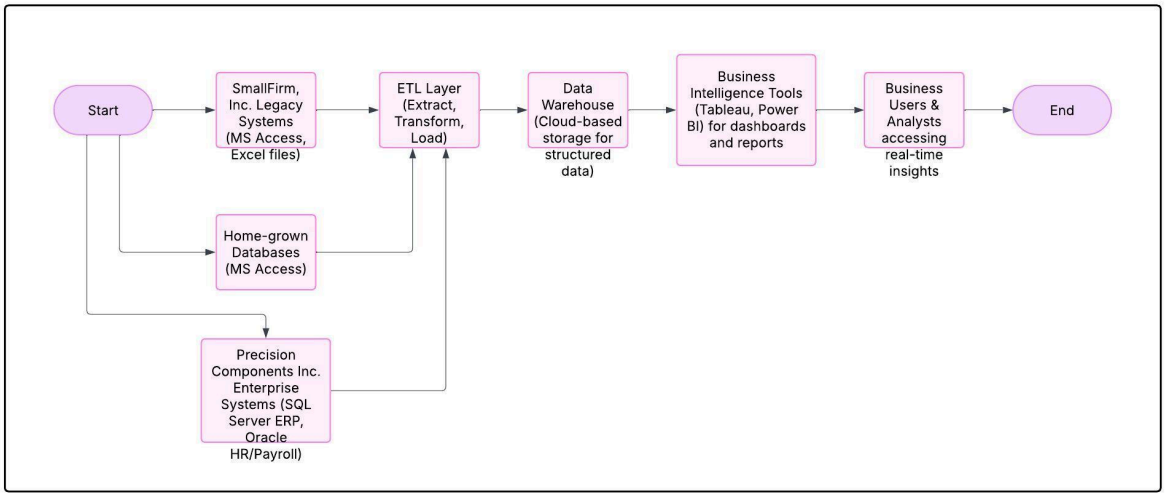
- SmallFirm, Inc. Legacy Systems (MS Access, Excel files)
- Precision Components Inc. Enterprise Systems (SQL Server ERP, Oracle HR/Payroll)
- Home-grown Databases (MS Access)

Data Processing Layers

- ETL Layer (Extract, Transform, Load) using tools like Apache NiFi, Talend, AWS Glue, or Azure Data Factory
- Data Warehouse (Cloud-based storage for structured data)

End Users & Applications

- Business Intelligence Tools (Tableau, Power BI) for dashboards and reports
- Business Users & Analysts accessing real-time insights



Part III: Processing Design Evaluation
C. Define the design's relevance to stakeholders by doing the following:
1. Describe two or more advantages to the specified approach in your recommended design compared to others you considered that would satisfy the business requirements.
2. Describe two or more disadvantages to the specified approach in your recommended design compared to others you considered that would satisfy the business requirements.

1. Advantages of the Recommended Design

Advantage 1: Scalability & Future Growth Readiness

- The recommended cloud-based data warehouse (e.g., AWS Redshift, Google BigQuery, or Azure Synapse Analytics) is designed to scale dynamically as data volume increases.
- This approach supports Precision Components Inc.'s international expansion plans, ensuring smooth data integration from multiple geographic locations.
- Alternative approaches, such as an on-premise data warehouse, would be more rigid and require costly hardware upgrades as the business grows.

Advantage 2: Automated ETL for Real-time Data Processing

- By using ETL tools like Apache NiFi, Talend, AWS Glue, or Azure Data Factory, automated data ingestion and transformation will ensure timely updates from different source systems.
- This enables daily or nightly data ingestion from SmallFirm, Inc. into the Precision Components Inc. ERP and HR systems.
- Alternative methods, such as manual data integration or batch file transfers, would be prone to errors, delays, and require more human effort.

Advantage 3: Centralized, Unified Data for Decision-Making

- The data warehouse consolidates data from all business units, including finance, HR, sales, and manufacturing, ensuring a single source of truth.
- This integration eliminates data silos, allowing holistic insights into company performance via real-time dashboards in Power BI, Tableau, or Looker.
- Without a centralized data warehouse, reporting would continue to be fragmented, requiring analysts to pull reports separately from different enterprise systems.

2. Disadvantages of the Recommended Design

Disadvantage 1: Implementation Complexity & Time-Intensive Setup

- Migrating data from SmallFirm, Inc.'s MS Access and Excel files into a cloud-based data warehouse requires significant data cleansing and transformation efforts.
- The integration of home-grown MS Access systems, SQL Server ERP, and Oracle HR/payroll systems will involve complex schema mapping and ETL processes.
- Alternative approaches, such as keeping the existing MS Access and Excel-based reporting structure, would require minimal initial effort but would not support business growth.

Disadvantage 2: Increased Costs Compared to On-Premise Solutions

- Cloud data warehouses and ETL tools involve ongoing subscription costs for storage, compute power, and API calls.
- While cloud-based infrastructure is more flexible, it can become expensive if query optimization is not properly managed.
- Alternative approaches, such as expanding the existing on-premise SQL Server system, may have lower recurring costs but lack scalability and flexibility.

Conclusion:

While the recommended cloud-based data warehouse approach provides scalability, automation, and unified reporting, it comes with higher initial implementation complexity and costs. However, these short-term challenges are outweighed by the long-term benefits, ensuring Precision Components Inc. remains competitive in the automotive supply chain industry.