End To End Data Warehousing Project

I. Project Planning and Requirements Analysis

- **Defining Epics and Sub-tasks**: Break down the project into epics (large tasks) and sub-tasks to manage complexity. Example epics include:
    - Requirements Analysis
    - Data Architecture Design
    - Project Initialization
    - Bronze Layer Build
    - Silver Layer Build
    - Gold Layer Build
- **Requirements Analysis**: Understand the project's needs by consulting with stakeholders. Key considerations include:
    - **Data Sources**: Identify sources like CRM and ERP systems, specifying data formats such as CSV files.
    - **Data Quality**: Determine necessary data cleaning and fixing.
    - **Data Integration**: Plan how to combine data sources into a unified data model.
    - **Historization**: Determine whether historical data needs to be maintained. The example project does not require historization.

II. Data Architecture Design

- **Data Management Approach**: Choose an approach such as a data warehouse, data lake, or data lakehouse. The example project focuses on building a data warehouse.
- **Data Warehouse Layer Definition**: Define the purpose and tasks of each layer:
    - **Bronze Layer**: Stores raw, unprocessed data directly from the source systems. This layer maintains a history of data for traceability and debugging.
    - **Silver Layer**: Contains clean, standardized data after applying basic transformations such as data cleansing, normalization, and standardization.
    - **Gold Layer**: Stores business-ready data structured for reporting and analytics. This layer involves data integration and may use a star schema.
- **Data Source Specification**: Document all data sources, including databases, APIs, and files. The video project uses CSV files as data sources.

III. Development Environment Setup

- **Tool Downloads**: Download necessary tools for development:
    - SQL Server Express: A local server for the database.
    - SQL Server Management Studio (SSMS): A client for interacting with the database.

IV. Project Initialization

- **Detailed Project Plan**: Expand the initial project plan with more epics and tasks for each layer (Bronze, Silver, Gold).
- **Naming Conventions**: Create rules for naming database schemas, tables, stored procedures, and folders.
    - **Consistency**: Apply naming conventions consistently across all project elements. Examples include prefixes for table types (e.g., `dim_` for dimension tables, `fact_` for fact tables).
- **Git Repository**: Create and structure a Git repository to store and track code.
    - **Folder Setup**: Establish folders for datasets, documents, scripts, and tests.
- **Database and Schema Creation**: Write a script to create the database and schemas (Bronze, Silver, Gold).

V. Building the Bronze Layer

- **Source System Analysis**: Understand the source systems thoroughly.
    - **Expert Interviews**: Interview source system experts.
    - **Documentation**: Document business context, data ownership, and supported business processes.

  - **Data Model Understanding**: Understand table and column descriptions.
  - **Integration Capabilities**: Determine integration methods like APIs, CSV files, or direct database connections.
  - **Performance Considerations**: Understand data volume limitations and their effect on performance.
- **DDL Script Creation**: Define table structures in the Bronze layer based on source system metadata.
  - **Naming Convention**: Follow the naming convention: `SourceSystem_Entity`.
  - **Column Names**: Ensure column names match the source system.
- **Data Ingestion Script Development**: Write scripts to load data from the source into the Bronze layer.
  - **Full Load**: Perform a full load by truncating the table and inserting the data.
  - **Stored Procedure**: Create a stored procedure to automate the data loading process.
- **Error Handling and Logging Implementation**: Incorporate error handling and logging in the stored procedure.
  - **TRY-END Blocks**: Use `BEGIN TRY` and `END TRY` blocks to catch errors.
  - **Print Messages**: Print messages to indicate loading status and any errors.
- **Data Validation**: Ensure data completeness and schema accuracy.
- **Data Flow Diagram Creation**: Visualize data flow from source systems to the Bronze layer.
- **Code Commitment to Git**: Store DDL scripts and stored procedures in the Git repository.

VI. Building the Silver Layer

- **Data Exploration and Analysis**: Explore data in the Bronze layer to understand its content and relationships.
  - **Data Integration Diagrams**: Create diagrams to document relationships between tables.
- **Data Cleansing Script Development**: Write scripts to clean and transform data.
  - **Handling Data Quality Issues**: Address duplicates, missing data, and invalid values.
  - **Data Transformations**: Perform normalization, standardization, and data enrichment.
- **DDL Script Creation**: Define table structures in the Silver layer, mirroring the Bronze layer.
- **Data Loading Script Development**: Write scripts to load data from the Bronze layer into the Silver layer.
  - **Full Load**: Perform a full load by truncating the table and inserting transformed data.
  - **Stored Procedure**: Automate data loading with a stored procedure.
- **Error Handling and Logging**: Implement error handling and logging in the stored procedure.
- **Data Validation**: Validate data quality in the Silver layer.
- **Data Flow Diagram Extension**: Update the data flow diagram to include the Silver layer, showing data lineage from source to Silver layer.
- **Code Commitment to Git**: Store DDL scripts, stored procedures, and quality checks in the Git repository.

VII. Building the Gold Layer

- **Business Object Exploration**: Identify main business objects (e.g., customers, products, sales) within source systems.
- **Data Modeling**: Design a data model (e.g., star schema) optimized for reporting and analytics.
  - **Dimension Table Creation**: Create dimension tables for descriptive information (who, what, where).
  - **Fact Table Creation**: Create fact tables for events and measures (how much, how many).
- **View Creation**: Write SQL views to transform and combine data from the Silver layer into business-ready datasets.
- **Data Validation**: Test the data model and ensure data integration is correct.
- **Data Model Diagram**: Create a diagram of the star schema, including tables, columns, primary keys, and foreign keys. Show relationships between fact and dimension tables.
- **Data Catalog Creation**: Document the data model, including descriptions of tables and columns.
- **Data Flow Diagram Finalization**: Update the data flow diagram to include the Gold layer, completing the data lineage.
- **Code Commitment to Git**: Store SQL views, quality checks, and documentation in the Git repository.