Data Warehouse Project

A **Data Warehouse** is a system for organising, structuring, and preparing data for data analysis, serving as the foundation for data analytics projects. It is subject-oriented, integrated, time-variant, and non-volatile.

* **Subject-oriented** means it focuses on business areas like sales, customers, or finance.
* **Integrated** because it combines data from multiple sources.
* **Time-variant** because it stores historical data.
* **Non-volatile** because data is not deleted or modified once it enters the data warehouse.

**The Importance of Data Warehousing**

Without a data warehouse, companies may face several issues:

* **Time-consuming data collection**: Data analysts spend weeks or months manually generating reports.
* **Inconsistent data**: Users consume reports with varying data freshness, making decision-making difficult.
* **Difficult data integration**: Integrating data from multiple sources manually is chaotic and risky.

A data warehouse solves these issues by:

* **Automating data collection**: Using ETL (Extract, Transform, Load) processes to extract data from sources, transform it, and load it into the data warehouse.
* **Providing a single source of truth**: The data warehouse becomes the central repository for analysis and reporting.
* **Enabling historical data access**: Storing historical data for trend analysis.
* **Ensuring data consistency**: Providing reports with the same data status.
* **Handling big data**: Modern data warehouses in cloud platforms can handle massive amounts of data.

**Roles in Building a Data Warehouse**

Different roles are involved in building and maintaining a data warehouse:

* **Data engineers**: Build the ETL components and the data warehouse database.
* **Data analysts**: Consume the data warehouse to build data models and reports for stakeholders.

**ETL Process**

The ETL process is a core element of the data warehouse, involving three key steps:

1. **Extract**: Extracting row data from various source systems.
   * Different extraction methods include pulling data from the source or having the source system push data.
   * Extraction types can be full (extracting all records) or incremental (extracting only new or changed data).
   * Techniques include manual extraction, database queries, file parsing, API connections, streaming (e.g., Kafka), change data capture (CDC), and web scraping.
2. **Transform**: Transforming the row data into a meaningful format for analysis and reporting.
   * Transformations include data enrichment, data integration, deriving new columns, data normalisation, applying business rules, data aggregation, and data cleansing.
   * Data cleansing involves removing duplicates, filtering data, handling missing or invalid values, removing unwanted spaces, and casting data types.
3. **Load**: Loading the transformed data into the target data warehouse.
   * Processing types include batch processing (loading data in one large batch) and stream processing (processing changes in real-time).
   * Load methods include full load (truncate and insert, update insert/upsert, drop create and insert) and incremental load (upsert, insert only, merge).
   * Slowly Changing Dimensions (SCDs) manage historisation. SCD0 means no historisation; SCD1 means overwriting old data; and SCD2 means adding new records for each change.

**Data Architecture Approaches**

When building a data warehouse, various approaches can be followed:

* **Inmon approach**: Uses a staged approach with a staging area, an enterprise data warehouse (EDW) using third normal form, and data marts for reporting.
* **Kimball approach**: Focuses on building data marts directly from the staging area for faster results.
* **Data Vault**: Similar to the Inmon approach but splits the EDW into row vault and business vault layers.
* **Medallion Architecture**: Uses Bronze, Silver, and Gold layers.
  + **Bronze Layer**: Stores raw, unprocessed data.
  + **Silver Layer**: Stores clean and standardised data.
  + **Gold Layer**: Contains business-ready data for consumption.
* **Data Mesh**: A decentralised approach where multiple departments build and share data products.

A key principle in data architecture is the **separation of concerns**, which breaks down complex systems into smaller, independent parts, each responsible for a specific task.

By understanding these components and processes, it is possible to build a data warehouse that provides a solid foundation for data analytics and informed decision-making.

**Data modelling** is a key process in building a data warehouse, especially in the gold layer. It involves organising and structuring raw data into a meaningful format that is optimised for reporting and analytics. The goal is to create a user-friendly data model that is flexible, scalable, and easy to understand.

Here's a breakdown of data modelling, based on the sources:

* **Objective:** To take raw, unorganised data and transform it into a structured and easily understandable format for business reporting and analysis.
* **Stages of Data Modelling**:
  + **Conceptual Data Model**: Focuses on identifying the main entities (e.g., customers, orders, products) and their relationships without detailed attributes.
  + **Logical Data Model**: Specifies columns and attributes within each entity, defines primary keys, and illustrates relationships between entities.
  + **Physical Data Model**: Adds technical details like data types, lengths, and database-specific configurations for implementation.
  + In projects, data architects commonly draw conceptual and logical data models.
* **Data modelling techniques**:
  + **Star Schema**: A central fact table surrounded by dimension tables. The fact table contains events or transactions, while dimensions hold descriptive information. Star schemas are easy to understand and query, making them ideal for analytics and reporting.
  + **Snowflake Schema**: Similar to the star schema but with dimensions broken down into smaller sub-dimensions. Snowflake schemas reduce data redundancy but are more complex to query.
* **Fact and dimension tables**:
  + **Fact tables** contain measures and numbers, answering "how much" or "how many."
  + **Dimension tables** hold descriptive information, answering "who," "what," or "where."
* **Process of building a data model:**
  + Understand the original data model and business objects.
  + Label tables with business objects.
  + Integrate data from different sources.
  + Rename columns into user-friendly names.
  + Validate the new data model and test data integration.
  + Document the data model and commit changes to Git.
  + When creating a fact table, include surrogate keys from dimension tables to connect the data model.
* **Data catalog**:
  + Create a data catalog to describe the data model, including table descriptions, column details, and relationships.
  + Document each column with descriptions and examples.

**Data transformation** is a core component of the ETL (Extract, Transform, Load) process used in building a data warehouse. It involves **manipulating and reshaping data** to make it suitable for analysis and reporting. Data transformation occurs after extracting the data from its source and before loading it into the target system, such as a data warehouse.

Here's a detailed look at data transformation, according to the sources:

* **Purpose**:
  + To convert raw data into a meaningful and usable format.
  + To ensure data quality, consistency, and compatibility across different sources.
  + To prepare data for specific analytical needs and business requirements.
* **Types of Transformations**:
  + **Data Cleansing**: Correcting or removing inaccurate, incomplete, or irrelevant data. This includes:
    - Removing duplicates.
    - Filtering data.
    - Handling missing data by filling in blanks with default values.
    - Handling invalid values.
    - Removing unwanted spaces or characters.
    - Casting data types to ensure data is in the correct format.
    - Detecting outliers.
  + **Data Standardisation**: Converting data into a consistent format. This may involve mapping coded values to user-friendly descriptions.
  + **Data Normalisation**: Organising data to reduce redundancy and improve data integrity.
  + **Data Enrichment**: Adding value to data sets by incorporating additional relevant information.
  + **Data Integration**: Combining data from multiple sources into a unified view.
  + **Data Aggregation**: Summarising data to a different level of granularity.
  + **Deriving New Columns**: Creating new columns based on existing data through calculations or transformations.
  + **Applying Business Rules**: Implementing specific business logic and criteria to create new columns or categorisations.
  + **Data Type Casting**: Converting data from one data type to another.
* **Layer-Specific Transformations**:
  + **Bronze Layer**: No transformations are applied in this layer to maintain raw, untouched data.
  + **Silver Layer**: Focuses on data cleansing and standardisation. Basic transformations are applied to prepare the data for the final layer.
  + **Gold Layer**: Focuses on business transformations needed for specific use cases. This includes data integration, aggregation, and applying business logic and rules.
* **Best Practices**:
  + **Separation of Concerns**: Ensure that each layer in the data warehouse has unique tasks and responsibilities. For example, data cleansing should be done in the silver layer, while business transformations should be done in the gold layer.
  + **Naming Conventions and Standards**: Follow consistent naming conventions and coding standards to improve code readability and maintainability.
  + **Data Quality Checks**: Implement data validation and quality checks at each stage of the transformation process to ensure data accuracy and completeness.
  + **Documentation**: Document all transformations performed on the data, including the purpose, logic, and dependencies.
  + **Error Handling and Logging**: Implement robust error handling and logging mechanisms to track and resolve any issues during the transformation process.
  + **Performance Optimisation**: Optimise transformation scripts for performance to ensure efficient data processing.

**Data integration** is a key aspect of building a data warehouse, especially in the transformation stage of the ETL process. It involves **combining data from multiple sources into a unified view** within the data warehouse. The goal is to create a **single, user-friendly data model** designed for analytics and reporting.

Here's a detailed breakdown of data integration, according to the sources:

* **Objective**:
  + To merge data from various source systems into a consistent and coherent data model.
  + To provide a unified view of data that enables comprehensive analysis and reporting.
* **Importance**:
  + **Addresses the issue of inconsistent data**: In organisations without proper data management, data is often scattered across multiple systems, leading to inconsistent reporting and difficulty in decision-making. Data integration solves this problem by bringing all data into one place.
  + **Enables integrated reports**: Data integration makes it possible to create reports that draw on data from multiple sources. Without it, merging data manually is chaotic, time-consuming, and prone to errors.
* **Techniques and methods**:
  + **ETL Process**: Data integration is typically performed during the transformation stage of the ETL process. This involves extracting data from various sources, transforming it to a common format, and loading it into the data warehouse.
  + **Data Standardisation and Normalisation**: These techniques are used to ensure that data from different sources is consistent and compatible.
  + **Data Enrichment**: Involves adding values to the dataset.
  + **Building a Central Data Warehouse**: Using approaches such as the Inmon, Kimball, or Data Vault to consolidate data.
  + **Medallion Architecture**: Following the Medallion architecture to organise data into bronze, silver, and gold layers.
  + **Star schema**: Use a star schema data model with fact and dimension tables.
* **Layer-Specific Integration**:
  + **Silver Layer**: In the silver layer, data integration involves basic transformations to standardise and clean the data.
  + **Gold Layer**: In the gold layer, data integration focuses on building business-ready data models by merging data from different sources.
* **Example**:
  + Combining customer data from CRM and ERP systems into a single customer dimension table. This involves resolving inconsistencies in customer IDs, names, and addresses to create a unified customer view.
* **Challenges and Considerations**:
  + **Data Quality**: Ensure data quality by cleansing and validating data before and during the integration process.
  + **Data Governance**: Implement data governance policies to ensure data consistency, accuracy, and compliance with regulatory requirements.
  + **Performance**: Optimise data integration processes for performance to handle large volumes of data efficiently.
  + **Metadata Management**: Manage metadata to provide a clear understanding of data lineage, transformations, and relationships.
* **Steps for Data Integration**:
  + Understand the original data model and business objects.
  + Label tables with business objects.
  + Rename columns into user-friendly names.
  + Validate the new data model and test data integration.
  + Document the data model and commit changes.

The **ETL (Extract, Transform, Load) process** is a crucial component in building a data warehouse and is used to get data from source systems into a data warehouse. It involves extracting data from source systems, transforming it into a usable format, and loading it into the target data warehouse. The ETL process is essential for data integration and ensures that data is consistent, high-quality, and ready for analysis and reporting.

Here's a detailed breakdown of the ETL process, according to the sources:

* **Extract**:
  + The initial stage involves **identifying and retrieving data** from various source systems. Sources can include databases, CSV files, APIs, and streaming data. The goal is to pull out the necessary data without altering it.
  + **Extraction Methods**:
    - **Pull vs. Push**: Data can be extracted by pulling it from the source system or having the source system push it to the data warehouse.
    - **Full vs. Incremental**: Data can be extracted fully (all records) or incrementally (only new or changed data).
    - **Techniques**: Manual extraction, direct database connections, file parsing, API connections, streaming, Change Data Capture (CDC), and web scraping are all methods of extracting data.
* **Transform**:
  + This stage involves **manipulating and converting the extracted data** into a format suitable for analysis and reporting. Transformations may include cleaning, standardising, normalising, and integrating data.
  + **Types of Transformations**:
    - **Data Cleansing**: Removing duplicates, filtering, handling missing or invalid data, and correcting inconsistencies.
    - **Data Standardisation**: Converting data into a consistent format.
    - **Data Normalisation**: Organising data to reduce redundancy and improve data integrity.
    - **Data Enrichment**: Adding relevant information to enhance the data.
    - **Data Integration**: Combining data from multiple sources.
    - **Data Aggregation**: Summarising data.
    - **Deriving New Columns**: Creating new columns based on existing data.
    - **Applying Business Rules**: Implementing business logic to transform data.
* **Load**:
  + The final stage involves **loading the transformed data into the target data warehouse**. The data is organised and stored for analysis and reporting.
  + **Loading Methods**:
    - **Batch vs. Stream Processing**: Data can be loaded in batches or processed in real-time via streaming.
    - **Full vs. Incremental**: Data can be loaded fully by truncating and inserting, updating and inserting (upsert), or dropping and recreating tables, or incrementally using upsert or merge.
    - **Slowly Changing Dimensions (SCD)**: SCDs manage how historical data is handled in dimension tables, with common types including SCD0 (no history), SCD1 (overwriting), and SCD2 (adding new records for each change).
* **Considerations for Real-World Projects**:
  + **Data Architecture**: ETL processes can vary depending on the data architecture (e.g., data warehouse, data lake, or data lakehouse).
  + **Techniques and Methods**: There are wide range of techniques, so it is important to make decisions about which one to apply to projects.
  + **Multiple Layers**: Data architectures often have multiple layers, and ETL processes can be tailored for moving data between these layers. For example, you might use a full ETL process between one layer and only the "L" (load) component between other layers.
  + **ETL vs ELT**: If transformations are done in the cloud, massive amounts of data can be handled.
* **Importance of ETL**:
  + **Single Source of Truth**: ETL creates a single, reliable source of truth for data.
  + **Improved Decision-Making**: By providing clean, integrated data, ETL enables better informed business decisions.
  + **Efficiency**: ETL automates data collection and preparation, freeing up data teams to focus on analysis.
  + **Data Quality**: ETL processes include data validation and quality checks.
  + **Historical Data**: ETL supports the maintenance of historical data for trend analysis.
* **Bronze Layer**:
  + **Purpose**: To store **raw, untouched data** directly from the source systems. It acts as a landing zone for the initial data ingestion.
  + **Use**:
    - **Traceability**: It provides a reference point for tracing data lineage and debugging issues.
    - **Data Analysis**: It allows data engineers to analyse the root cause of issues by examining the original data.
    - **No Transformations**: Data in this layer remains as is, without any transformations or modifications.
    - **Object type** tables
  + **Access**: Typically, access is restricted to data engineers to prevent accidental alterations or misuse of raw data.
* **Silver Layer:** 
  + **Purpose**: To store **clean and standardised data** that has undergone basic transformations.
  + **Use**:
    - **Data Preparation**: It serves as an intermediate layer where data is cleansed, standardised, and transformed to prepare it for the final gold layer.
    - **Basic Transformations**: This layer focuses on data cleansing, standardisation, and normalisation.
    - **Data Quality**: The goal is to improve data quality and consistency, making it easier to work with in subsequent layers.
    - **Object type** tables
  + **Transformations**: Data cleansing, standardisation, data normalisations, deriving new columns and data enrichment.
  + **Load method**: Full load
* **Gold Layer:** 
  + **Purpose**: To provide **business-ready data** that can be readily consumed by business users and analysts.
  + **Use**:
    - **Reporting and Analytics**: It offers a curated and transformed dataset optimised for reporting, analytics, and business intelligence.
    - **Business Logic**: It incorporates business rules, aggregations, and data integration to create a user-friendly data model.
    - **Data Integration**: It integrates data between source systems.
    - **Data Aggregations**: Data aggregations are applied to the data.
    - **Business Transformations**: Business transformations are applied to the data.
    - **Object type** Views
  + **Data Model**: The data model is built in this layer.
  + **Access**: Access is granted to business users and analysts to enable self-service reporting and data exploration.

A **data model** is a way of organising and structuring data in a meaningful way. It involves taking raw data and arranging it into **user-friendly objects** such as customers, orders, and products. The purpose of data modelling is to make data easier to understand, use, and manage.

Key aspects of data modelling include:

* **Logical data model**: Involves specifying columns and primary keys for each entity, and drawing relationships between those entities.
* **Physical data model**: Adds technical details to the data model like data types and lengths for each column.
* **Data integration**: Combining data from multiple sources into a single, user-friendly data model designed for analytics and reporting.
* **Metadata management**: Managing metadata to provide a clear understanding of data lineage, transformations, and relationships.
* **Naming conventions**: Table names should follow the convention of beginning with the source system, and then the entity name. Column names should be meaningful and business aligned.

Data models for analytics and data warehousing should be flexible, scalable, and easy to understand. Two common data models include the **star schema** and the **snowflake schema**.

* **Star Schema**:
  + Contains a central fact table surrounded by dimension tables.
  + Fact tables contain transactions and events, while dimension tables hold descriptive information.
  + The relationship between fact and dimension tables forms a star shape.
  + Star schemas are easier to understand and query but may contain data duplicates in the dimension tables.
* **Snowflake Schema**:
  + Similar to the star schema, but dimension tables are further broken down into smaller sub-dimensions.
  + The extended dimensions create a snowflake-like shape.
  + Snowflake schemas normalise data and optimise storage but are more complex to query.

Data models also contain two types of tables: **fact tables and dimension tables**.

* **Dimension tables**: Contain descriptive information or categories that provide context to the data (e.g., product information). Dimension tables answer who, what, and where questions.
* **Fact tables**: Contain events or transactions and include IDs from multiple dimensions, timestamps, and measures (e.g., sales data). Fact tables answer how much or how many questions.

The **key skills** and **topics** covered in the video include:

* **Data Architecture**: Designing a modern data architecture following best practices.
* **Data Engineering**: Writing code to clean, transform, load, and prepare data for analysis.
* **Data Modelling**: Learning the basics of data modelling and creating a new data model from scratch for analysis.
* **ETL/ELT Processing**: Implementing ETL (Extract, Transform, Load) and ELT processes using SQL to prepare data.
* **Data Integration**: Merging data from multiple sources.
* **Data Loading**: Loading data into the data warehouse.
* **SQL**: Using SQL Server; however, other databases like MySQL or BigQuery can also be used.
* **Data warehousing**: Projects focus on how to organise, structure, and prepare data for data analysis.
* **Data analysis**: After building a data warehouse, further analysis and reporting can be performed.

Furthermore, the video delves into specific techniques and methods within the ETL process:

* **Extraction**:
  + **Pulling data** from the source system.
  + **Full extraction**, extracting all records.
  + **File parsing** to extract data from files.
* **Transformation**:
  + **Data enrichment**, adding values to the datasets.
  + **Data integration**, bringing multiple sources into one data model.
  + **Data normalisation**, mapping code values to more user-friendly values.
  + **Data cleansing**, which includes:
    - Removing duplicates.
    - Filtering data.
    - Handling missing data.
    - Handling invalid values.
    - Removing unwanted spaces.
    - Casting data types.
    - Detecting outliers.
* **Loading**:
  + **Batch processing**.
  + **Full load**, which includes truncate and insert.
  + **Historisation**, using SCD1 (Slowly Changing Dimension Type 1) to update the content of the data warehouse.
* **Project planning**: Creating a project plan to organise project ideas, plans, and resources.
* **Defining naming conventions**: Establishing a set of rules for naming everything in the project, such as database schemas, tables, stored procedures, and folders.
* **Using Git for version control**: Implementing Git for code management, tracking changes, and collaboration.

Additionally, the video touches on data architecture approaches:

* **Data warehouse**: Organising and structuring data for reporting.
* **Data lake**: Storing various types of data in a less structured manner.
* **Data lakehouse**: A hybrid approach combining features of data warehouses and data lakes.
* **Data mesh**: A decentralised approach where multiple domains build and share data products.

The video uses a **Medallion architecture**, which includes Bronze, Silver, and Gold layers. Each layer has a specific purpose, transformation rules, and target audience.

* **Bronze layer**: Stores raw data.
* **Silver layer**: Stores clean and standardized data.
* **Gold layer**: Contains business-ready data for reporting and analytics.

Through the project, it is possible to develop skills as a data architect, data engineer, and data modeler. The end result is a professional portfolio project that can be showcased.

**There are three different types of SQL projects:**

* **Data warehousing**: Focuses on how to **organise, structure, and prepare data** for data analysis. This forms the foundation for any data analytics project.
* **Exploratory Data Analysis (EDA)**: Focuses on **understanding and uncovering insights** about datasets. Involves asking the right questions and finding answers using basic SQL skills.
* **Advanced analytics**: Focuses on using advanced SQL techniques to **answer business questions**. This includes finding trends over time, comparing performance, segmenting data, and generating reports for stakeholders.

The video also details a specific SQL data warehousing project:

* Skills gained include ETL/ELT processing using SQL, building data architecture, data integration, data loading, and data modelling.
* ETL (Extract, Transform, Load) is a process that extracts data from sources, transforms it, and loads it into a data warehouse. This data warehouse then becomes the single point of truth for analysis and reporting.
* The **data engineer** is typically responsible for building the ETL component and the data warehouse.
* Key steps include requirement analysis, designing data architecture, project initialisation, and building the different data layers (Bronze, Silver, Gold).
* **Bronze Layer**: Stores raw data.
  + Full load (truncate and insert).
  + No transformations.
* **Silver Layer**: Stores clean and standardised data.
  + Full load (truncate and insert).
  + Involves data transformations like data cleansing, normalisation, and standardisation.
* **Gold Layer**: Contains business-ready data.
  + Data integration.
  + Uses views.
  + Data modelling.

Based on the video transcript, numerous **SQL topics relevant to building a modern data** warehouse are covered. These topics span data architecture, ETL processes, data modelling, and data quality.

**Data Warehousing Fundamentals**

* **Data warehouse definition**: The data warehouse is defined by its subject orientation, integration of multiple data sources, time variance for historical data, and non-volatility to ensure data is not deleted or modified.
* **Importance of data warehousing**: Data warehousing is important because it streamlines data collection, ensures consistent data, and facilitates data integration, which leads to better decision-making.
* **Data architecture approaches**: Different approaches to designing a data warehouse include Inmon, Kimball, Data Vault, Medallion Architecture (Bronze, Silver, Gold layers), and Data Mesh.

**ETL (Extract, Transform, Load) Processes**

* **ETL overview**: ETL is central to data warehousing, involving extracting data from sources, transforming it, and loading it into the data warehouse.
* **Extraction methods**: Extraction involves either pulling data from the source system or having the source system push data. Extraction types include full extraction and incremental extraction. Techniques range from manual to automated, including database queries, API connections, and change data capture (CDC).
* **Transformation techniques**: The transformation phase includes various techniques such as:
  + **Data cleansing**: Addressing data quality issues like duplicates, missing or invalid data, and unwanted spaces.
  + **Data enrichment**: Enhancing data with additional relevant information.
  + **Data integration**: Combining data from multiple sources.
  + **Data normalisation**: Organising data to minimise redundancy and improve integrity.
  + **Data type casting**: Changing data types for consistency and accuracy.
  + **Handling invalid data**: Managing and correcting inaccurate or erroneous data entries.
  + **Deriving new columns**: Creating new data fields based on existing data.
* **Loading methods**: The loading phase involves inserting transformed data into the data warehouse. Methods include full load (truncate and insert, update insert/upsert, drop create and insert) and incremental load (upsert, insert only, merge).
* **Slowly Changing Dimensions (SCDs)**: Techniques for managing historical data, including SCD0 (no history), SCD1 (overwriting old data), and SCD2 (adding new records for each change).

**Data Modelling**

* **Data modelling stages**: The data modelling stages are conceptual, logical, and physical.
* **Star schema**: A data modelling approach with a central fact table surrounded by dimension tables.
* **Snowflake schema**: An alternative model that normalises dimension tables into smaller sub-dimensions.
* **Fact tables**: Fact tables contain measures and metrics related to business processes.
* **Dimension tables**: Dimension tables provide context and attributes for the facts, answering questions like who, what, and where.

**Data Quality**

* **Importance of data quality**: Ensuring data accuracy, completeness, and consistency is crucial before, during, and after ETL processes.
* **Data profiling**: Analysing data to understand its structure, content, and relationships.
* **Data validation**: Implementing checks and constraints to ensure data conforms to expected standards.
* **Data integrity**: Maintaining consistency and accuracy throughout the data lifecycle.

**SQL Techniques and Functions**

* **DDL (Data Definition Language) statements**: Creating and modifying database objects like tables, views, and schemas.
* **DML (Data Manipulation Language) statements**: Inserting, updating, and deleting data.
* **Bulk insert**: Efficiently loading large volumes of data from files into database tables.
* **String functions**: Manipulating text data using functions like TRIM, SUBSTRING, and REPLACE.
* **Date functions**: Handling date and time values, including data type conversions and range validation.
* **Aggregate functions**: Performing calculations like COUNT, SUM, AVG, MIN, and MAX on data.
* **Window functions**: Performing calculations across a set of table rows that are related to the current row, such as LEAD and LAG.
* **Data type conversions**: Converting data from one type to another, such as casting integers to dates.

**Development and Project Management**

* **Project planning**: Creating detailed project plans with epics and tasks to manage data warehouse development.
* **Naming conventions**: Establishing and following consistent naming standards for database objects.
* **Version control**: Using Git for version control, collaboration, and rollback capabilities.
* **Code commenting**: Adding comments to SQL scripts for clarity and maintainability.
* **Error handling**: Implementing TRY...CATCH blocks for error handling and logging.
* **Performance optimisation**: Measuring and optimising ETL process durations.

**Data Integration and Transformation**

* **Data lookup**: Joining tables to retrieve surrogate keys and related data.
* **Data standardisation**: Ensuring consistent data formats and values.
* **Business logic implementation**: Applying business rules and logic within SQL queries.
* **Surrogate key generation**: Creating unique identifiers for dimension and fact tables.
* **View creation**: Building virtual tables (views) in the gold layer for simplified data access.

A compilation of **SQL Topics** and functions that can be learned from this notes :

**Data Warehousing and ETL Concepts**

* **Data warehouse**: Understanding the definition of a data warehouse as subject-oriented, integrated, time-variant, and non-volatile collection of data.
* **ETL (Extract, Transform, Load)**: Learning the ETL process for extracting data from sources, transforming it, and loading it into a data warehouse. The source notes that 90% of the time spent on building a data warehouse project is spent on the ETL.
* **Data Architecture**: How to build data architecture and integrate data from multiple sources.
* **Data Modelling**: The basics of data modelling.
* **Data integration**: Combining data from multiple sources.
* **Data governance**: Ensuring data quality and consistency throughout the data warehouse.
* **Slowly Changing Dimensions (SCDs)**: Different methods for handling historical data in a data warehouse.
  + **SCD0**: No historisation, no changes allowed.
  + **SCD1**: Overwriting old data with new data.
  + **SCD2**: Adding new records for each change and inactivating old records.
* **Data Lakehouse**: A mix between a data warehouse and a data lake, combining the flexibility of a data lake with the structure of a data warehouse.
* **Enterprise Data Warehouse**: Modelling data using third normal form, building a new integrated data model from multiple sources.

**SQL Techniques and Functions**

* **Data Extraction**:
  + **Full Extraction**: Extracting all records from tables.
  + **Incremental Extraction**: Extracting only new or changed data.
  + **Techniques**: Connecting to databases with queries, parsing files, using APIs, event-based streaming, Change Data Capture (CDC), and web scraping.
* **Data Transformation**:
  + **Data Enrichment**: Adding values to data sets.
  + **Data Normalisation**: Mapping coded values to user-friendly values.
  + **Data Aggregation**: Aggregating data to different granularities.
  + **Data Cleansing**: Removing duplicates, filtering data, handling missing or invalid values, removing unwanted spaces, and casting data types.
  + **Business rules and logic**: Defining criteria to build new columns.
* **Data Loading**:
  + **Full Load**:
    - Truncate and insert.
    - Drop, create, and insert.
  + **Incremental Load**:
    - Upsert (update and insert).
    - Insert only (append data).
    - Merge (update, insert, delete).
* **Specific SQL commands and techniques**:
  + USE command: Switching between databases.
  + CREATE DATABASE: Creating new databases.
  + CREATE SCHEMA: Creating schemas to organise database objects.
  + **File Loading**: BULK INSERT for loading data from files.
  + CREATE OR ALTER PROCEDURE: Creating or modifying stored procedures.
  + EXECUTE: Executing stored procedures.
  + PRINT: Displaying messages in SQL.
  + Error Handling: Using TRY...CATCH blocks to handle errors.
  + **Variable Declaration**: Declaring variables to store values.
  + **Date Functions**: Using GETDATE() to get current timestamp and DATEDIFF() to calculate duration.
  + ROW\_NUMBER(): Generating surrogate keys.
  + **String manipulation**: Functions like TRIM and SUBSTRING for data cleansing.
  + ISNULL(): Handling null values.
  + **CAST()**: Converting data types.
* **Data Quality Checks**:
  + Checking for nulls and duplicates in primary keys.
  + Validating data ranges.
  + Ensuring data integrity by comparing values across tables.

**Data Modelling**

* **Star Schema**: Designing a data model with fact and dimension tables.
* **Snowflake Schema**: Discussing the complexity and normalisation of snowflake schemas.
* **Fact Tables**: Understanding that fact tables contain measures/numbers.
* **Dimension Tables**: Knowing that dimension tables answer "who, what, where" questions.

**Project Management and Development**

* **Project Initialisation**: Creating Git repository and preparing project structure.
* **Version Control**: Using Git for version control and collaboration.
* **Documentation**: Providing clear documentation of the data model.
* **Naming Conventions**: Establishing and following naming conventions for database objects.
* **Data Catalog**: Creating a data catalog to provide better understanding of the data.
* **Data Lineage**: Understanding and documenting data flow from sources to the data warehouse.