WIE3007 DATA MINING AND WAREHOUSING 2023/2024 S1

Tutorial 2

Q1) Describe the importance of the Requirements Gathering phase in Data Warehouse Planning and Design. How does it impact subsequent phases, and what are the risks associated with inadequate requirements gathering?

The Requirements Gathering phase is a critical step in the planning and design of a data warehouse. It sets the foundation for the entire data warehouse project and has a significant impact on subsequent phases. Here's an overview of its importance, its impact on other phases, and the risks associated with inadequate requirements gathering:

Foundation for Decision-Making:

Requirements Gathering is where you identify and document the specific needs and objectives of the data warehouse project. This phase involves discussions with stakeholders to understand their business goals, reporting needs, data sources, and key performance indicators. Without a clear understanding of these requirements, it's challenging to make informed decisions in the subsequent phases.

Alignment with Business Goals:

Understanding the business requirements helps align the data warehouse project with the organization's strategic goals. It ensures that the data warehouse will support the decision-making process by providing the right data and insights to achieve those objectives.

Q2) You are tasked with designing the data model for a retail Data Warehouse. Define the entities, attributes, relationships, and schema type you would propose and justify your choices. Consider entities like Customers, Products, Orders, and any others you deem necessary.

Entities:

Customers:

Attributes: CustomerID, FirstName, LastName, Email, Phone, Address

Justification: Customers are a fundamental entity in retail, and their information is crucial for customer segmentation, marketing, and understanding buying behavior.

Products:

Attributes: ProductID, Name, Description, Category, Brand, Price, Stock

Justification: Products are what the retail business sells, and their attributes are essential for inventory management, sales analysis, and category performance.

Orders:

Attributes: OrderID, CustomerID, OrderDate, TotalAmount, PaymentMethod, ShippingAddress

Justification: Orders represent customer transactions, and their details are central for sales analysis, tracking customer orders, and measuring revenue.

Order Line Items:

Attributes: OrderLineItemID, OrderID, ProductID, Quantity, Price, TotalPrice

Justification: This entity is crucial to break down each order into individual items, enabling detailed sales analysis and stock management.

Vendors/Suppliers:

Attributes: VendorID, Name, ContactInfo

Justification: Vendors or suppliers are crucial for procurement and managing the supply chain.

Store Locations:

Attributes: StoreID, Name, Address, ContactInfo, Manager

Justification: Store locations are important for tracking sales performance by location, inventory management, and for understanding regional trends.

Relationships:

Customer-Orders:

One-to-Many relationship between Customers and Orders (a customer can place multiple orders).

Order-Order Line Items:

One-to-Many relationship between Orders and Order Line Items (an order can have multiple line items).

Product-Order Line Items:

One-to-Many relationship between Products and Order Line Items (each line item corresponds to a product).

Vendor-Products:

Many-to-Many relationship between Vendors and Products (a vendor can supply multiple products, and a product can be supplied by multiple vendors).

Store-Orders:

One-to-Many relationship between Store Locations and Orders (orders are made at specific store locations).

Schema Type Justification:

For a retail Data Warehouse, a star schema is often the most suitable choice. Here's why:

Simplicity: Star schemas are simple to understand and query. They consist of a central fact table (Orders) and dimension tables (Customers, Products, Vendors, Store Locations) that provide context.

Performance: Star schemas optimize query performance because they involve denormalization, making it easier to retrieve data quickly, especially for common queries related to sales, inventory, and customer behavior.

Analytical Power: Star schemas are well-suited for analytical purposes, such as data analysis, reporting, and business intelligence, which are common use cases in retail.

Scalability: Star schemas are scalable, and new dimensions can be added as needed to accommodate future business requirements without disrupting existing data.

Q3) Explain the ETL process in detail. Design a high-level ETL flow, considering a source system containing customer information in a relational database and transaction information in a NoSQL database, and elucidate how you would handle data transformation and loading into the Data Warehouse.

Extract: Design Extract Logic

- Determine how data will be extracted from source systems

Transform: Develop Transformation Logic:

- Define the transformations required to clean, enrich and format the data

Load: Design Load Strategy

- Plan how the transformed data will be loaded into the Data Warehouse.

High-Level ETL Flow:

Extraction:

- Extract customer information from the relational database (e.g., MySQL) using SQL queries.
- Extract transaction information from the NoSQL database (e.g., MongoDB) using appropriate connectors or APIs.
- Store the extracted data in a temporary storage area.

Transformation:

Perform data cleansing:

Handle missing values, duplicates, and anomalies in customer and transaction data.

Data validation:

Check for data integrity and constraints.

Data enrichment:

Merge customer data with transaction data to create a comprehensive dataset.

Data aggregation:

Summarize transaction data to facilitate analytics.

Data conversion:

Ensure data types and formats are consistent with the data warehouse schema.

Data quality checks:

Apply quality checks to ensure that the data meets predefined quality standards.

Loading:

- Stage the transformed data in the data warehouse staging area.
- Transform the data to match the data warehouse schema and business rules.
- Load the transformed data into the data warehouse's fact and dimension tables.
- Update or append data as required, maintaining data warehouse integrity.

Q4) How would you go about developing a prototype for your Data Warehouse project? Describe the steps involved, the scope of the prototype, and how you would gather and incorporate feedback from stakeholders to refine your Data Warehouse design.

- Create a Small-Scale Model:
    - Develop a prototype using a subset of the data to validate the design and gather feedback.
        - Define Scope: Select a subset of the data, focusing on the most critical and representative entities and relationships.
        - Use a development environment that is isolated from production.
        - Ensure the subset is diverse and large enough to validate the design effectively but manageable in terms of complexity and volume.
        - implement the designed Data Model, including tables, relationships, indexes, and any other database objects, in the prototype Data Warehouse.
        - Ensure these reports or dashboards align with the business requirements and objectives of the prototype.
- Refine Design:
    - Modify the design based on the feedback and any issues identified during prototyping.