

DAY- 8(Data Warehousing)

TOPIC: Data Warehousing Fundamentals

1. Design a data warehouse schema for a retail company that includes dimension tables for products, customers, and time. Implement the schema using a relational database management system (RDBMS) of your choice.
2. Create a fact table that captures sales data, including product ID, customer ID, date, and sales amount. Populate the fact table with sample data.
3. Write SQL queries to retrieve sales data from the data warehouse, including aggregations and filtering based on different dimensions.

TOPIC: ETL and Data Integration

1. Design an ETL process using a programming language (e.g., Python) to extract data from a source system (e.g., CSV files), transform it by applying certain business rules or calculations, and load it into a data warehouse.
2. Implement the ETL process by writing code that performs the extraction, transformation, and loading steps.

TOPIC: Dimensional Modeling and Schemas

1. Design a star schema for a university database, including a fact table for student enrollments and dimension tables for students, courses, and time. Implement the schema using a database of your choice.
2. Write SQL queries to retrieve data from the star schema, including aggregations and joins between the fact table and dimension tables.

TOPIC: Performance Optimization and Querying

1. Scenario: You need to improve the performance of your data loading process in the data warehouse. Write a Python script that implements the following optimizations:
 - a) Utilize batch processing techniques to load data in bulk instead of individual row insertion.
 - b) Implement multi-threading or multiprocessing to parallelize the data loading process.
 - c) Measure the time taken to load a specific amount of data before and after implementing these optimizations.

Submission Guidelines:

1. Answer all the questions in a single Jupyter Notebook file (.ipynb).
2. Include necessary code, comments, and explanations to support your answers and implementation.
3. Ensure the notebook runs without errors and is well-organized.
4. Create a GitHub repository to host your assignment files.
5. Rename the Jupyter Notebook file using the format "date_month_topic.ipynb" (e.g., "12_July_DataWarehousing.ipynb").
6. Place the Jupyter Notebook file in the repository.
7. Commit and push any additional files or resources required to run your code (if applicable) to the repository.
8. Ensure the repository is publicly accessible.
9. Submit the link to your GitHub repository as the assignment submission.

Grading Criteria:

1. Understanding and completeness of answers: 40%
2. Clarity and depth of explanations: 25%
3. Correct implementation and evaluation of matrix operations: 15%
4. Proper code implementation and organization: 10%
5. Overall presentation and adherence to guidelines: 10%

Note:- Create your assignment in Jupyter notebook and upload it to GitHub & share that uploaded assignment file link through your dashboard. Make sure the repository is public.