The objective of this assignment is to design and implement data pipelines using Hive and Pig to analyze data efficiently. You will define a schema, load data into Hive tables, perform analytical queries, optimize query execution using partitioning and bucketing, and compare performance with Pig Latin queries. Each problem carries 8 points.

Problem 1. Based on the three data files `Course_Attendance.csv`, `Enrollment_Data.csv`, and `GradeRosterReport.csv` provided on the LMS, define an appropriate schema for each dataset. The schema should effectively capture the structure and relationships among the data elements.

Once the schema is defined, create three Hive tables corresponding to these data files. Next, write Hive scripts to load the data from the given files into the respective Hive tables, ensuring data integrity and correctness.

If required, define a schema for storing erroneous or missing value tuples separately. This schema should capture key metadata about the errors, including the source table, column names, and type of issue (e.g., missing values, format inconsistencies).

Problem 2. After loading the data, define a data warehouse schema in Hive that integrates the three tables into a unified format. This schema should be designed to facilitate efficient querying and analysis.

Create a pipeline to map the three source tables into the data warehouse schema using appropriate transformations. Ensure that erroneous or missing value tuples are captured and stored in the previously defined error schema, preventing them from affecting analytical results.

Once the data is consolidated, define three complex analytical queries in HiveQL that extract meaningful insights from the data warehouse. The queries should be designed to showcase different aspects of the dataset and involve aggregations, joins, and filtering.

Problem 3. To enhance query performance, implement partitioning and bucketing on the required tables. Partitioning helps in logically dividing the data for faster access, while bucketing distributes data into smaller, more manageable files.

Create a new Hive table with partitioning and bucketing applied and then run the three analytical queries on the optimized table. Compare the execution time of these queries with the execution time on the original data warehouse table, documenting the improvements in performance.

Problem 4. Export the data from the data warehouse table into a CSV file. Then, load the exported data into a Pig table, ensuring that the schema and data format are correctly maintained.

Redefine the three analytical queries using Pig Latin and execute them. Compare the runtime of the Pig Latin queries with the HiveQL queries and analyze the differences in performance, highlighting the strengths and weaknesses of each approach.

Deliverables

At the end of the assignment, you are required to submit the following:

- Schema definitions and Hive table creation scripts.

- Hive scripts for data loading and query execution.
- Analytical query results along with runtime comparisons.
- Pig Latin scripts and performance analysis.
- A report summarizing observations, performance comparisons, and conclusions.

Design your own complex queries—here, complexity does not imply using advanced functions in HiveQL/PigLatin. The relational operations that you perform should be complex. Kindly refrain from using LLMs wherever possible. You are expected to report genuine challenges faced while implementing the data pipelines.