Lab Assignment 08 - Batch 01

Aim:

IT Incident Analysis in Hive.

Objective:

In this exercise, you will work with a sample dataset containing IT incident records. You'll ingest the data into Hive, perform some basic transformations, and then execute queries to derive insights from the incident data.

Theory:

- Explain Hive and its commands

Dataset: You have been provided with a CSV file named **incident_data.csv**, which contains the following columns:

- **incident_id**: Unique identifier for each incident.
- **timestamp**: Date and time when the incident occurred.
- **severity**: Severity level of the incident (e.g., Critical, High, Medium, Low).
- **category**: Category of the incident (e.g., Network, Server, Application). ● **description**: Description of the incident.
- **assigned_to**: IT personnel assigned to resolve the incident.
- **resolved_timestamp**: Date and time when the incident was resolved (if applicable).

Tasks:

1. **Data Ingestion:**

○ Ingest the **incident_data.csv** file into a Hive table named **incident_data**.
○ Define appropriate data types for each column.

2. **Data Transformation:**

   ○ Extract the date from the **timestamp** column and store it in a new column named **incident_date**.

   ○ Calculate the resolution time for each incident (if resolved) by subtracting the **timestamp** from the **resolved_timestamp**, and store it in a new column named **resolution_time**.

3. **Querying:**

   ○ Retrieve the total number of incidents reported.

   ○ Calculate the average resolution time for incidents by severity level. ○ Identify the top 5 categories with the highest number of incidents. ○ Determine the distribution of incidents over time by plotting a time series chart.

4. **Advanced Querying (Optional):**

   ○ Analyze the correlation between incident severity and resolution time. ○ Identify recurring incidents or patterns by performing text analysis on the **description** column.

   ○ Implement a simple predictive model to forecast future incident volumes based on historical data.

Hints:

- Use the appropriate Hive commands (**CREATE TABLE**, **LOAD DATA**, **SELECT**, **GROUP BY**, **ORDER BY**, etc.) to perform each task.

- Utilize date and time functions in Hive to manipulate timestamps and calculate resolution times.

Outcome:

This exercise provides a practical scenario for analyzing IT incident data using Hive. Learners can gain insights into incident trends, resolution times, and patterns using Hive queries and data visualization techniques.

Lab Assignment 08 - Batch 02

Aim:

Analyzing Sales Data in Hive.

Objective:

In this exercise, you will work with a sample dataset containing sales records. You'll ingest the data into Hive, perform some basic transformations, and then execute queries to derive insights from the sales data.

Theory:

- Explain Hive and its commands

Dataset: You have been provided with a CSV file named **sales_data.csv**, which contains the following columns:

- **order_id**: Unique identifier for each order.
- **customer_id**: Unique identifier for each customer.
- **order_date**: Date of the order.
- **product_id**: Unique identifier for each product.
- **product_name**: Name of the product.
- **quantity**: Quantity of the product ordered.
- **unit_price**: Price per unit of the product.

Tasks:

1. **Data Ingestion:**

- ○ Ingest the **sales_data.csv** file into a Hive table named **sales_data**.
- ○ Define appropriate data types for each column.

2. **Data Transformation:**
   - ○ Calculate the total sales amount for each order by multiplying **quantity** and **unit_price**, and store it in a new column named **total_amount**.
   - ○ Extract the year and month from the **order_date** column and store them in separate columns.
   - ○ Round the **total_amount** to two decimal places.
3. **Querying:**
   - ○ Retrieve the total number of orders placed.
   - ○ Calculate the total sales amount for each month.
   - ○ Identify the top 5 best-selling products.
   - ○ Determine the total sales amount for each customer and rank them in descending order.
4. **Advanced Querying (Optional):**
   - ○ Calculate the average order value.
   - ○ Find out the month with the highest total sales amount.
   - ○ Analyze the distribution of sales across different product categories (if available).

Hints:

- ● Use the appropriate Hive commands (**CREATE TABLE**, **LOAD DATA**, **SELECT**, **GROUP BY**, **ORDER BY**, etc.) to perform each task.
- ● Refer to the Hive documentation for syntax and functions.
- ● Test your queries incrementally to ensure correctness.

Lab Assignment 08 - Practice

Aim:

Server Performance Analysis in Hive.

Objective:

In this exercise, you will analyze server performance metrics collected from various servers in a data center. You'll ingest the data into Hive, perform transformations, and execute queries to gain insights into server performance**.**

Theory:

- Explain Hive and its commands

Dataset: You have been provided with a CSV file named server_metrics.csv, containing the following columns:

- **server_id:** Unique identifier for each server.
- **timestamp:** Date and time when the metrics were collected.
- **cpu_usage:** CPU usage percentage at the time of measurement.
- **memory_usage:** Memory usage percentage at the time of measurement.
- **disk_usage:** Disk usage percentage at the time of measurement.
- **network_traffic:** Network traffic in bytes per second at the time of measurement.

Tasks:

1. **Data Ingestion:**
   - Ingest the server_metrics.csv file into a Hive table named server_metrics.
   - Define appropriate data types for each column.
2. **Data Transformation:**
   - Extract the date and hour from the timestamp column and store them in separate columns.
   - Round the cpu_usage, memory_usage, disk_usage, and network_traffic values to two decimal places.
3. **Querying:**
   - Calculate the average CPU, memory, disk, and network usage for each server.
   - Identify servers with the highest and lowest average CPU usage.
   - Determine the peak hours of network traffic across all servers.
   - Analyze trends in memory usage over time for a specific server.
4. **Advanced Querying (Optional):**
   - **I**mplement anomaly detection algorithms to identify servers exhibiting unusual behavior in terms of CPU, memory, or disk usage.

    - Perform predictive modeling to forecast future resource utilization based on historical data.
    - Incorporate additional server metrics (e.g., load average, disk I/O) into the analysis for a more comprehensive view of server performance.

Hints:

- Use appropriate Hive commands (CREATE TABLE, LOAD DATA, SELECT, GROUP BY, ORDER BY, etc.) to perform each task.
- Utilize date and time functions in Hive to extract date components and aggregate data over time intervals.

Outcome:

This exercise provides hands-on experience with analyzing server performance metrics using Hive. Learners can explore different aspects of server performance, such as resource utilization, trends over time, and anomaly detection, to gain insights into the health and efficiency of IT infrastructure.