Hive Assignment 1

Car Insurance Cold Calls Data Analysis

Problem 1: Data Loading

1. Load the data into a Hive table. Create an external table with the given schema and load the data into the table from a text file or HDFS path.

Problem 2: Data Exploration

1. How many records are there in the dataset?
2. How many unique job categories are there?
3. What is the age distribution of customers in the dataset? Provide a breakdown by age group: 18-30, 31-45, 46-60, 61+.
4. Count the number of records that have missing values in any field.
5. Determine the number of unique 'Outcome' values and their respective counts.
6. Find the number of customers who have both a car loan and home insurance.

Problem 3: Aggregations

1. What is the average, minimum, and maximum balance for each job category?
2. Find the total number of customers with and without car insurance.
3. Count the number of customers for each communication type.
4. Calculate the sum of 'Balance' for each 'Communication' type.
5. Count the number of 'PrevAttempts' for each 'Outcome' type.
6. Calculate the average 'NoOfContacts' for people with and without 'CarInsurance'.

Problem 4: Partitioning and Bucketing

1. Create a partitioned table on 'Education' and 'Marital' status. Load data from the original table to this new partitioned table.

2. Create a bucketed table on 'Age', bucketed into 4 groups (as per the age groups mentioned above). Load data from the original table into this bucketed table.
3. Add an additional partition on 'Job' to the partitioned table created earlier and move the data accordingly.
4. Increase the number of buckets in the bucketed table to 10 and redistribute the data.

Problem 5: Optimized Joins

1. Join the original table with the partitioned table and find out the average 'Balance' for each 'Job' and 'Education' level.
2. Join the original table with the bucketed table and calculate the total 'NoOfContacts' for each 'Age' group.
3. Join the partitioned table and the bucketed table based on the 'Id' field and find the total balance for each education level and marital status for each age group.

Problem 6: Window Function

1. Calculate the cumulative sum of 'NoOfContacts' for each 'Job' category, ordered by 'Age'.
2. Calculate the running average of 'Balance' for each 'Job' category, ordered by 'Age'.
3. For each 'Job' category, find the maximum 'Balance' for each 'Age' group using window functions.
4. Calculate the rank of 'Balance' within each 'Job' category, ordered by 'Balance' descending.

Problem 7: Advanced Aggregations

1. Find the job category with the highest number of car insurances.
2. Which month has seen the highest number of last contacts?
3. Calculate the ratio of the number of customers with car insurance to the number of customers without car insurance for each job category.

4. Find out the 'Job' and 'Education' level combination which has the highest number of car insurances.
5. Calculate the average 'NoOfContacts' for each 'Outcome' and 'Job' combination.
6. Determine the month with the highest total 'Balance' of customers.

Problem 8: Complex joins and aggregations

1. For customers who have both a car loan and home insurance, find out the average 'Balance' for each 'Education' level.
2. Identify the top 3 'Communication' types for customers with 'CarInsurance', and display their average 'NoOfContacts'.
3. For customers who have a car loan, calculate the average balance for each job category.
4. Identify the top 5 job categories that have the most customers with a 'default', and show their average 'balance'.

Problem 9: Advanced Window Functions

1. Calculate the difference in 'NoOfContacts' between each customer and the customer with the next highest number of contacts in the same 'Job' category.
2. For each customer, calculate the difference between their 'balance' and the average 'balance' of their 'job' category.
3. For each 'Job' category, find the customer who had the longest call duration.
4. Calculate the moving average of 'NoOfContacts' within each 'Job' category, using a window frame of the current row and the two preceding rows.

Problem 10: Performance Tuning

1. Experiment with different file formats (like ORC, Parquet) and measure their impact on the performance of your Hive queries.
2. Use different levels of compression and observe their effects on storage and query performance.

3. Compare the execution time of join queries with and without bucketing.
4. Optimize your Hive queries using different Hive optimization techniques (for example, predicate pushdown, map-side joins, etc.). Discuss the difference in performance.

Please make sure to submit the HiveQL queries, along with their results and your observations. This assignment not only tests your understanding of Apache Hive but also requires you to derive meaningful insights from the data.