CSE 512 Assignment 1

Maximum points Possible – 10

The required task is to understand data fragmentation technique on-top of an open-source relational database management system (i.e., PostgreSQL) and write some SQL queries to verify the understanding of various properties of fragmentation.

Input Data: The input data is a Movie Rating data set collected from the MovieLens website (http://movielens.org). The raw data is available in the given file ratings.dat. You can download the full rating.dat file from the MovieLens website: https://files.grouplens.org/datasets/movielens/ml-10m.zip. You can also use the partial dataset provided with assignment instructions for testing purpose. For your testing, we provide test_data.txt which provides a small fraction of rating.dat file. Be noted that we will use a larger dataset during evaluation.

The rating.dat file contains 10 million ratings and 100,000 tag applications applied to 10,000 movies by 72,000 users. Each line of this file represents one rating of one movie by one user, and has the following format:

UserID::MovieID::Rating::Timestamp

Ratings are made on a 5-star scale, with half-star increments. Timestamps represent seconds since midnight Coordinated Universal Time (UTC) of January 1, 1970. A sample of the file contents is given below:

1::122::5::838985046

1::185::5::838983525

We load all ratings into a table (saved in PostgreSQL) named *ratings* that has the following schema (note that we don't load timestamps of ratings):

userid(int) – movieid(int) – rating(float)

Required Tasks

Below are the steps you need to follow to fulfill this assignment requirements:

Part A: Write five SQL queries on top of ratings table that can be executed in a PostgreSQL database. Each query should return some tuples from the ratings table.

Part B: Provide a fragmentation for each of the following scenario of fragmentation properties such that the provided fragmentation decomposes the ratings table into 3 fragments. For each of the given scenario, write one or more SQL queries that can decompose the ratings table into 3 fragments satisfying the given scenario. Your SQL queries for each of the fragmentations should create three tables.

- B1): The fragmentation satisfies reconstruction and completeness properties but does not satisfy the disjointness property. After writing the queries for fragmentation, write another query on top of constructed fragments which shows that the ratings table can be reconstructed from the fragments. Explain how your fragmentation satisfies reconstruction and completeness but not disjointness. You can provide explanations as comments in the script.

- B2) The fragmentation satisfies completeness and disjointness properties but does not satisfy the reconstruction property. Explain how your fragmentation satisfies completeness and disjointness but not reconstruction. You can provide explanations as comments in the script.

- B3) The fragmentation satisfies all completeness, reconstruction and disjointness properties. After writing the queries for fragmentation, write another query on top of constructed fragments which shows that the ratings table can be reconstructed from the fragments. Explain how your fragmentation satisfies all three properties. You can provide explanations as comments in the script.

Part C: For the third fragmentation (B3) which satisfies all three properties, physically allocate the fragments into three tables named f1, f2 and f3. For the five queries that you wrote in Part A, rewrite all those queries in terms of the fragments instead of the original rating table.

Assignment Tips!

- You can write comment in SQL in two ways: 1) Begin the comment with a slash and an asterisk (/*). Proceed with the text of the comment. This text can span multiple lines. End the comment with an asterisk and a slash (*/). 2) Begin the comment with -- (two hyphens). Proceed with the text of the comment. This text cannot extend to a new line. End the comment with a line break.

- Please do not change the table name and its structure as provided in the assignment to keep it consistent.

- If you change the path to the dataset according to your local path, change it back to default provided path before you submit the solution.

- Please make sure that there is no syntax or other errors in your submission. In case of **any syntax error, 0 marks will be given.**

- For any case of doubt in the assignment, please make use of office hours and discussion board. Individual emails would not be entertained.

- Use PostgreSQL DBMS only. Use the version PostgreSQL-13.

Submission Instructions:

- Only submit the Assignment1.sql file. Do not change the file name. Do not put it into a folder or upload a zip.
- Multiple submissions are allowed. Only the latest submission will be graded. No late submission is accepted.

Note:- Failure to follow the instructions provided in the document will result in the loss of the points.