Big Data Analytics

Exercise Sheet 8

Prof. Dr.Dr. Lars Schmidt-Thieme, Mohsan Jameel

Information Systems and Machine Learning Lab University of Hildesheim

June 16th, 2016

Submission until June 22nd, 2016, to mohsan.jameel@ismll.de

Setup and Reading:

Download employees' database from the link https://github.com/datacharmer/test_db. It contains `employees.sql` and `employees_partitioned.sql` file for loading data.

Read about different partitioning options for creating horizontal partitions.

http://www.arubin.org/files/PracticalPartitioning_Webinar.pdf

Exercise: Design Problem (8 marks)

A retail company has a number of franchises in a country. The data generated by collective sales of all the franchises per day has volume over 10GB per day. The nature of the business is such that company need to maintain quarterly information and take decisions accordingly.

The scheme of sales table is given as

```
<transaction ID>,<franchise ID>, <article ID>, <data of sale>, <possible return date>, <amount of sale>
```

Propose data distribution strategy for this case. In defining a strategy you are required to provide:

1) Which partitioning strategy you choose (provide argument in the given context why you choose one over another)
2) Based on the partitioning strategy you choose, further explain which technique you choose and its more suitable over others i.e. partitioning by List etc
3) Give your solution in form of sql query for creating partitions. How many partitions you will choose.
4) How you manage the increasing size of the data you received?

Exercise: Working with RDBMS (7 marks)

For downloaded database of employee you need to measure performance of difference partitioning strategies.

1) Load data into the table using employees.sql (without partitions)

2) Run query to get count of salaries transaction from year (from_date) 2001-01-01 to 2004-01-01. Measure time to run the query.
3) Run query to get sum of all the salaries payed to employ no = 206679. Measure time to run the query.
4) Now create partitions based on the from_date and run query 2 and 3 again and measure time.
5) If you see performance decrease or no change in performance after partitioning point of the problem with partitioning strategy and propose an alternative which will help improve performance.