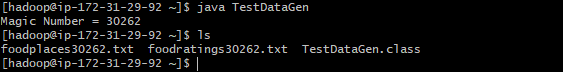
# CSP554—Big Data Technologies

## Assignment #4

Magic number screenshot

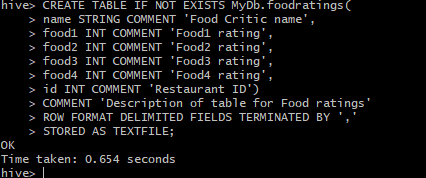


**Exercise 1:**

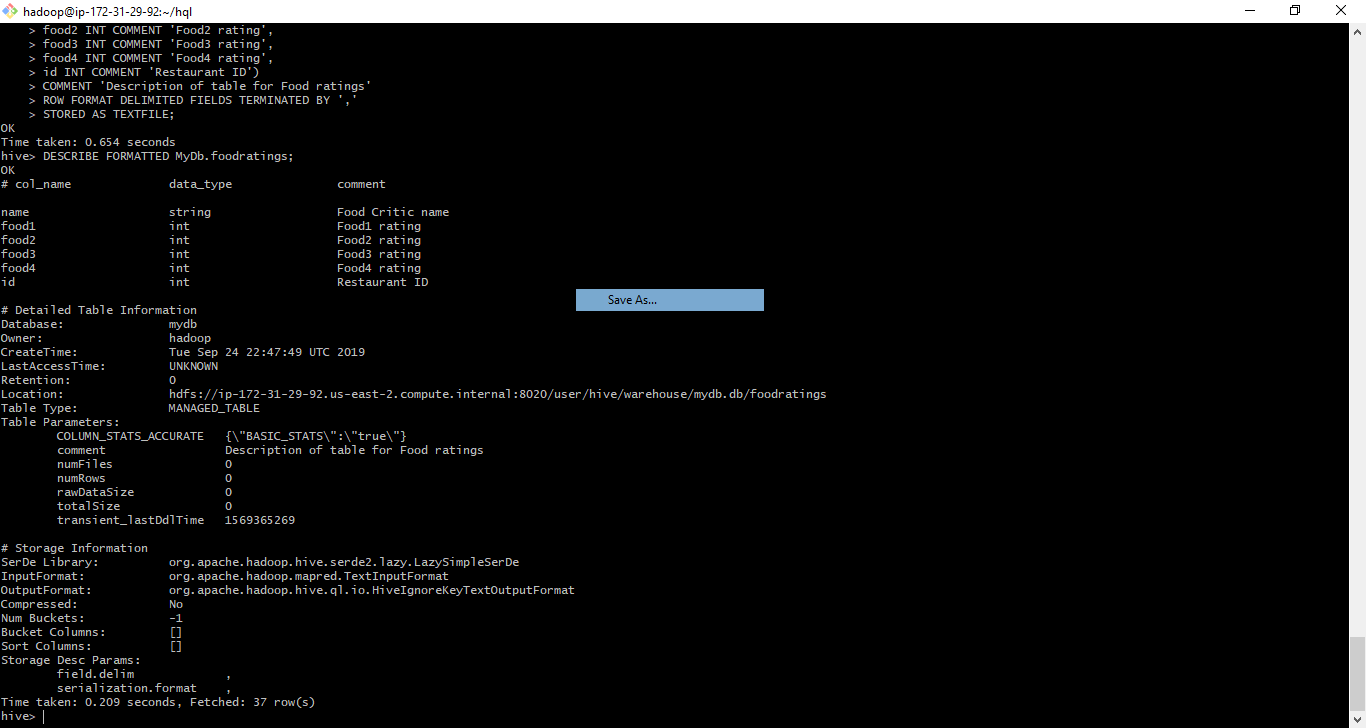
Database created command:



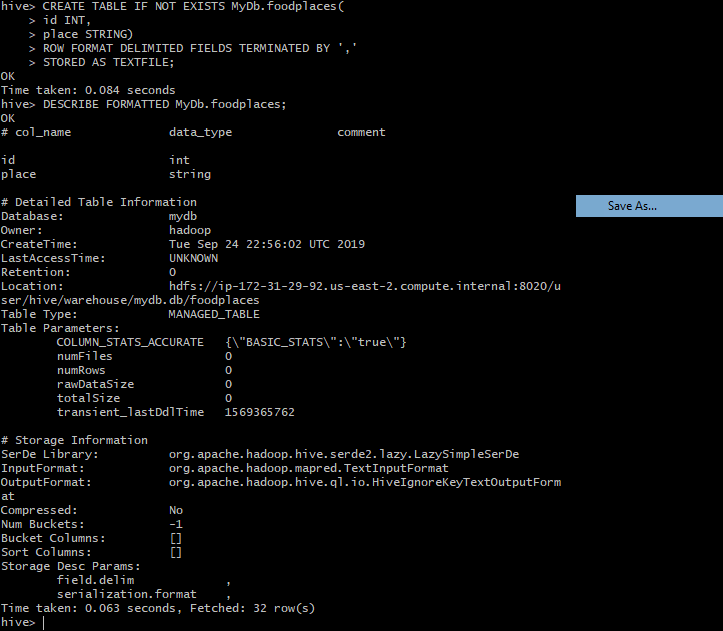
TABLE CREATION



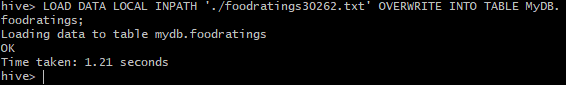
After describe table DESCRIBE FORMATTED MyDb.foodratings



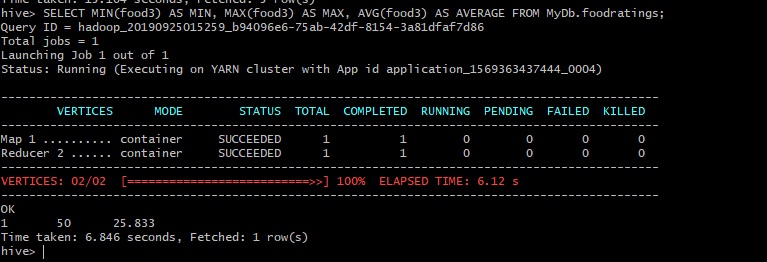
Creating table mYdb.foodplaces and ‘DESCRIBE FORMATTED MyDb.foodplaces’



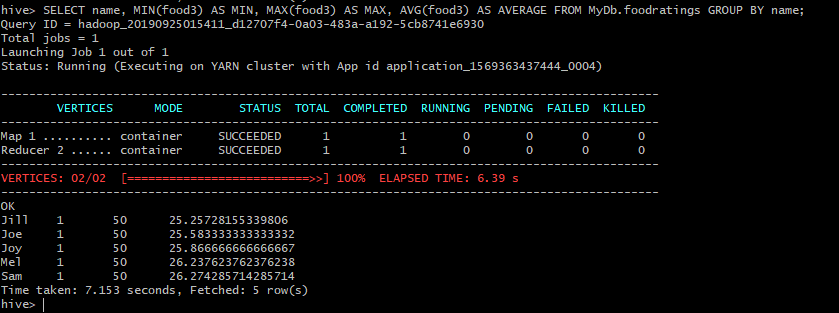
**Exercise 2:** The magic number is 30262



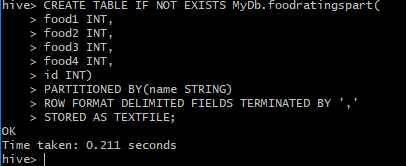
**For min average and maximum:**



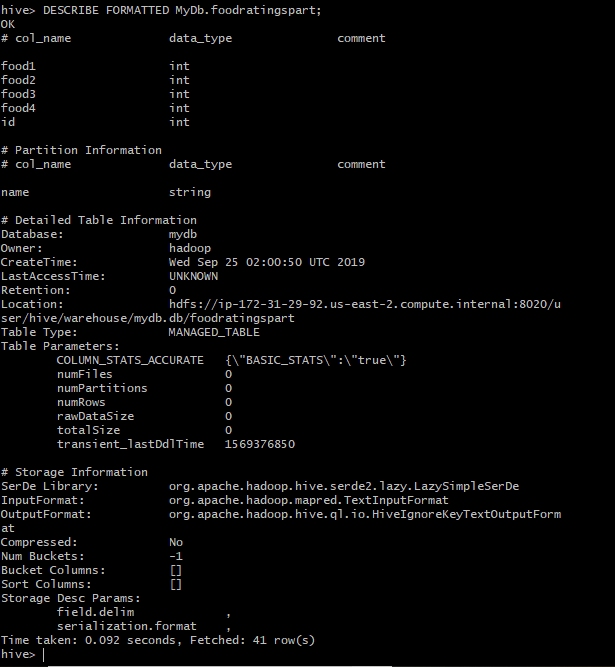
**Exercise 3:** The magic number is 30262



**Exercise 4:**



**After describe command:**

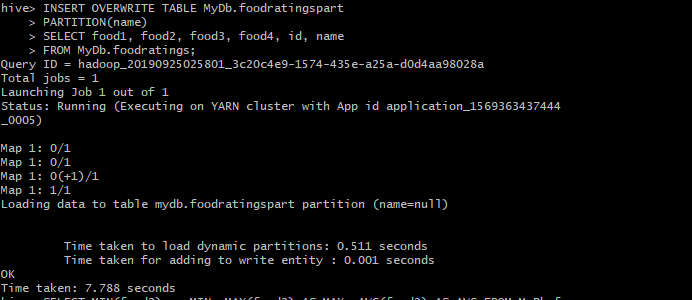


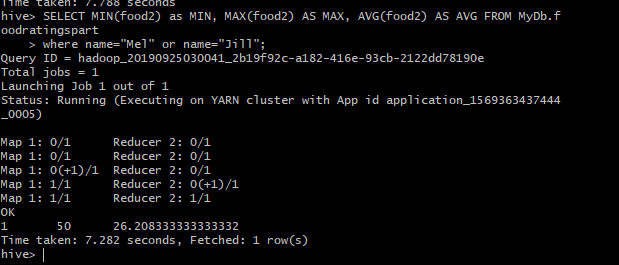
**Exercise 5:**

Partitioning helps in querying of data. It does so by selecting a column from the table

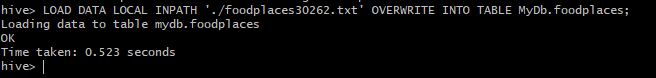
A partition created on critic name is much better than on place because it would be easier to understand. As there are few (less than 10) critics the result of partition would be a small number and we would get to know about the overall table. Creating a partition based on place id would lead to 10000 partitions which is not accepted and it would be difficult to interpret data. Also, if we query this partition it will take longer time to find the result than querying the food critic partition. Thus, partitioning on food critic helps in increasing the query performance as well as I/O time

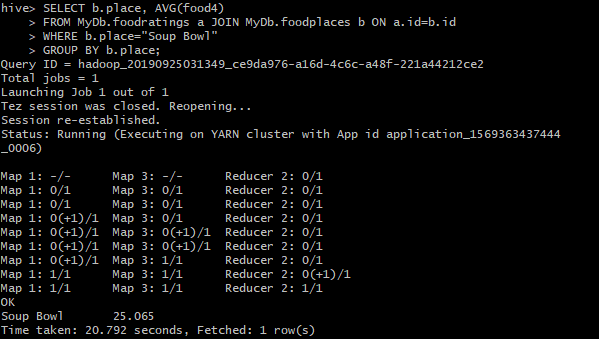
**Exercise6:** The magic number is 30262





**Exercise 7:** The magic number is 30262





**Exercise 8:**

The paper focuses on Apache Tez its open source library framework how it is scalable, reusable and flexible makes it different from MapReduce and Spark is given and also it is deployed at Yahoo! and LinkedIn. Tez helps in providing unifying framework to create purpose-built engines and customize data processing for their specific needs. It also provides elasticity, security and fault tolerance. The key benefits are improved performance, amortized development costs and enabling future pipelines. Tez uses DAG API to support high level engine. The structure of DAG helps in reusable and pluggable components. The Apache Tez consists of three main parts: 1) API library- providing DAG and Runtime APIs 2) Orchestration framework – used as YARN Application Master 3) Runtime library-providing inputs and outputs that can be used outside of the box. Limitations of Tez is that it is Java based and can run on JVM based applications only. Tez it can only be used in YARN based Hadoop cluster. There can be intermediate loss of data that leads to re-execution of tasks. Tez has been adopted in some of the following applications:

1) Apache MapReduce: Tez has MapProducer and ReduceProducer that do the tasks of map and reduce. The MapReduce does not require any changes to incorporate the Tez based version of MapReduce.

2) Apache Hive: Hive written in Hive SQL translates queries to trees but were inefficient due to limitation of MapReduce. This is fulfilled by Direct Acyclic Graph (DAG) of Tez and its edges in addition help in performing joins

3) Apache Pig: Being procedural the Pig Latin can use the DAG to create multiple output instead of the earlier use of tagging data or writing the side effect. The complex flow in graph can be represented by DAG

The experimental results have shown that in Hive Tez outperforms Mapreduce. Also in Yahoo Pig Production tests there were complex 20 to 50 vertices DAG having 100K+ tasks that had improved the performance from 1.5x to 2x against MapReduce. Thus, Tez’s customability and open source community has still research open to be done.