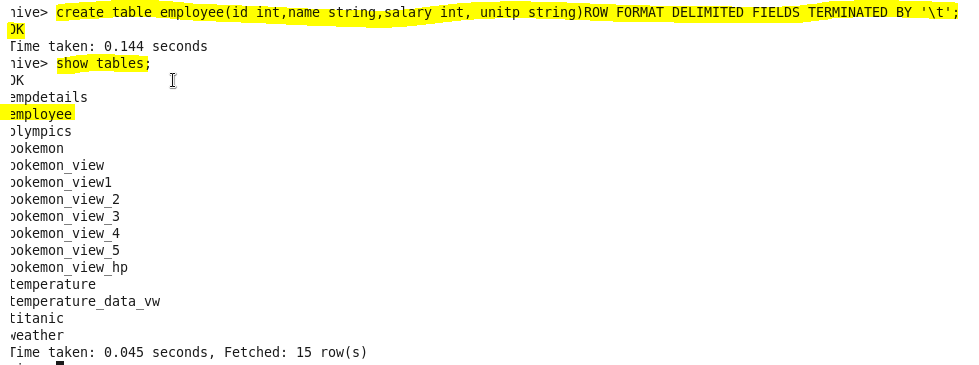
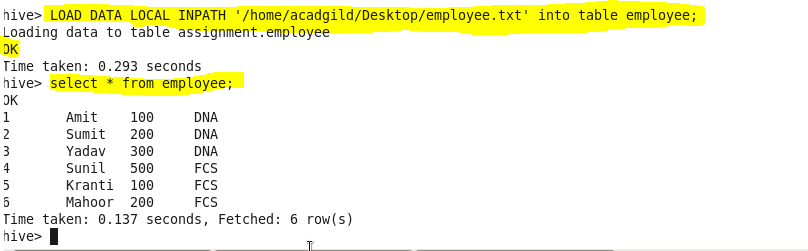
**ASSIGNMENT28.3**

* **First we will create table employee using create command**



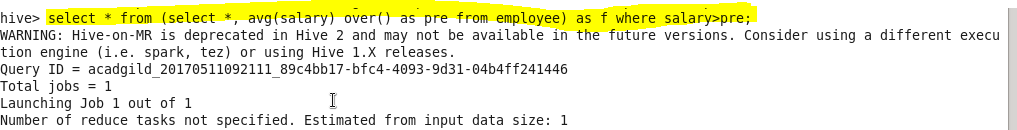
* **now we will load our dataset employee.txt using LOAD command**



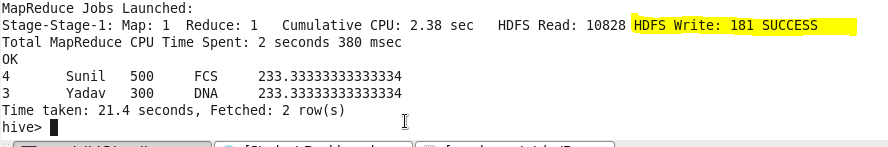
* **Get a list of employees who receive a salary less than 100, compared to their immediate employee with higher salary in the same unit**



* **List of all employees who draw higher salary than the average salary of that department**



**Output:**



* **Store the above dataset into the following file formats.**
* **SEQUENCEFILE**
* **RCFILE**
* **ORC FILE**

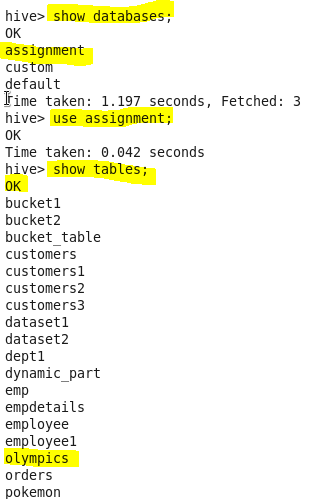
Share the commands used with their results

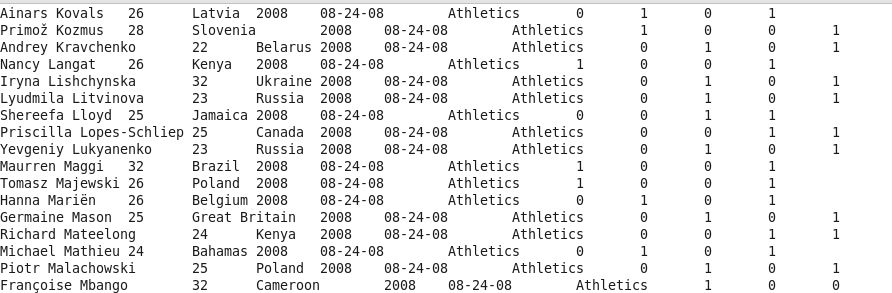
* **SEQUENCEFILE**

We know that Hadoop’s performance is drawn out when we work with a small number of files with big size rather than a large number of files with small size. If the size of a file is smaller than the typical block size in Hadoop, we consider it as a small file. Due to this, a number of metadata increases which will become an overhead to the NameNode. To solve this problem sequence files are introduced in Hadoop. Sequence files act as a container to store the small files.

Sequence files are flat files consisting of binary key-value pairs. When Hive converts queries to MapReduce jobs, it decides on the appropriate key-value pairs to be used for a given record. Sequence files are in the binary format which are able to split and the main use of these files is to club two or more smaller files and make them as a one sequence file.

First we create the database and the table for olympics dataset.

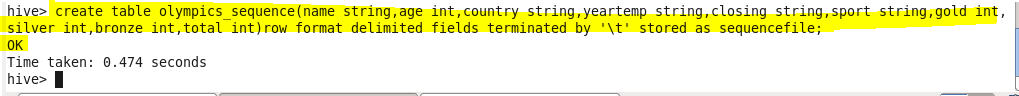




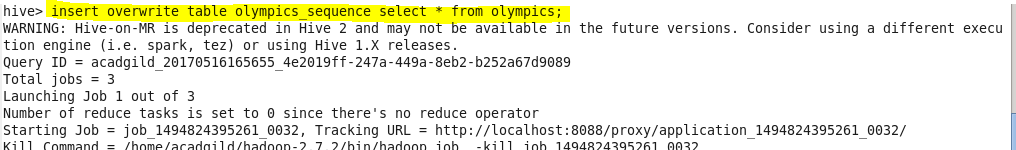
Hive uses the SEQUENCEFILE input and output formats from the following packages:

|  |  |
| --- | --- |
| 1  2 | org.apache.hadoop.mapred.SequenceFileInputFormat  org.apache.hadoop.hive.ql.io.HiveSequenceFileOutputFormat |

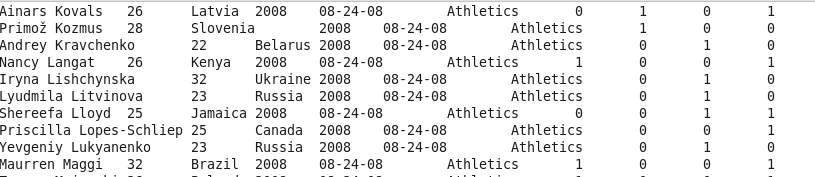
Creating table to store sequence data:



Inserting data into sequence table from another text.



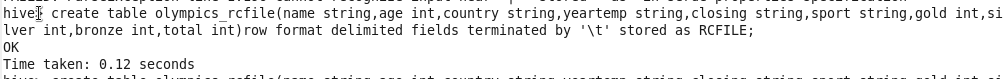
Output:

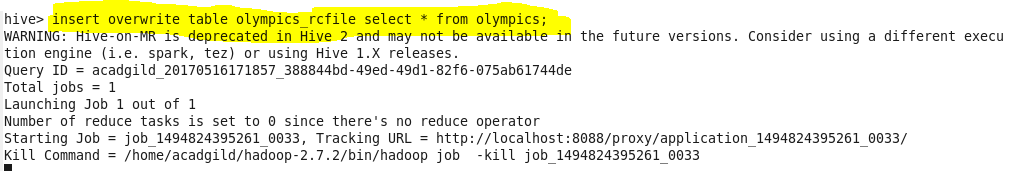


* **RCFILE**

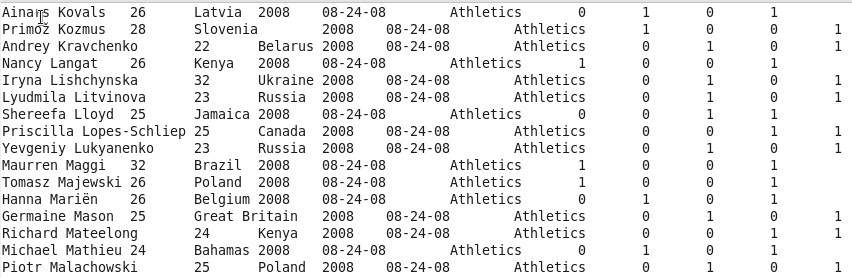
RCFILE stands of Record Columnar File which is another type of binary file format which offers high compression rate on the top of the rows.  
RCFILE is used when we want to perform operations on multiple rows at a time.  
RCFILEs are flat files consisting of binary key/value pairs, which shares much similarity with SEQUENCEFILE. RCFILE stores columns of a table in form of record in a columnar manner. It first partitions rows horizontally into row splits and then it vertically partitions each row split in a columnar way. RCFILE first stores the metadata of a row split, as the key part of a record, and all the data of a row split as the value part. This means that RCFILE encourages column oriented storage rather than row oriented storage.

Creating rcfile:





**Output:**

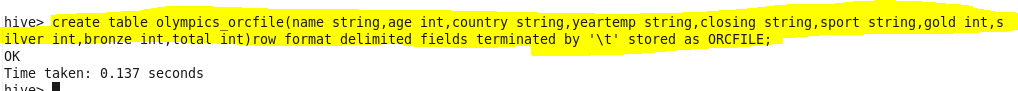


* **ORC FILE**

ORC stands for Optimized Row Columnar which means it can store data in an optimized way than the other file formats. ORC reduces the size of the original data up to 75%. As a result the speed of data processing also increases. ORC shows better performance than Text, Sequence and RC file formats.

An ORC file contains rows data in groups called as Stripes along with a file footer. ORC format improves the performance when Hive is processing the data

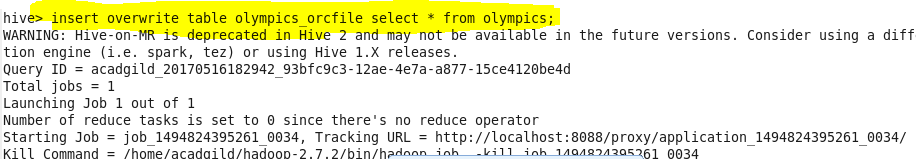
**Creating orc file:**



Hive has its own ORCFILE Input format and ORCFILE output format in its default package:

|  |  |
| --- | --- |
| 1 | org.apache.hadoop.hive.ql.io.orc |

**Inserting data into orc table**



**Output:**

