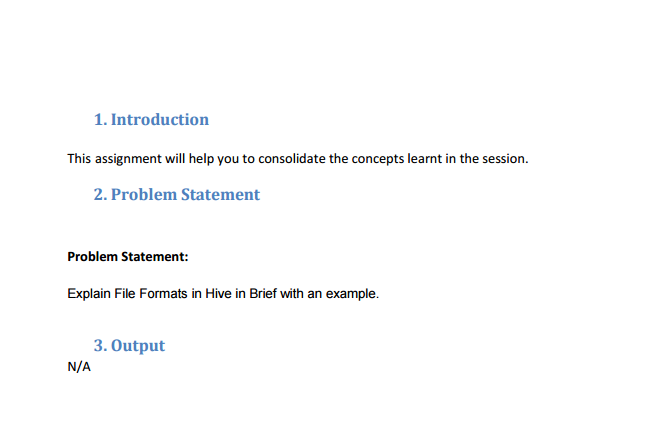
**Assignment 28.2**

****

* **Hive File Format:**
* File format – means a way by which all the information will be stored or can be encoded as a file in a computer.
* In case of Hive it means how records will stored inside a file.
* For structured data, each record will have its own structure.
* records are encoded within a file will be defined by the file format.
* file formats will vary mainly between data encoding, compression rate, usage of space and disk I/O.
* Hive will never check the data which is loaded match the table schema or not.
* Instead , it verify file format match to the table definition or not.
* **TEXT FILE:**
* TEXTFILE- is a input/output format which is being used in Hadoop.
* In Hive if a table is defined as TEXTFILE then it will load data from CSV (Comma Separated Values).
* They will be delimited by means of Tabs, Spaces, and JSON data.
* So fields inside each record will be separated by means of comma or space or tab or sometimes by JSON (JavaScript Object Notation) data.
* By default, if we use TEXTFILE format then each line is considered as a record.
* TEXTFILE format can be created in in Hive as follows:

***create table table\_name (schema of the table) row format delimited fields terminated by ',' | stored as TEXTFILE.***

* By the end, file format must be specified.
* If the format is not specified then it will take the file format as TEXTFILE format.
* The TEXTFILE input and TEXTFILE output in Hadoop package are:

***org.apache.hadoop.mapred.TextInputFormat***

***org.apache.hadoop.mapred.TextOutputFormat***

example - In Hive creating the TEXTFILE table format, and loading data to TEXTFILE format as well performing basic select operation.

**Creating TEXT FILE**

A table is created by using the below command:

create table olympic(athlete STRING, age INT, country STRING, year STRING, closing STRING, sport STRING, gold INT, silver INT, bronze INT, total INT) row format delimited fields terminated by '\t' stored as textfile;

* Table was named “olympic” and the table schema is specified.
* data in the input file is delimited by using tab space.
* And the file format is specified as TEXTFILE.
* By using describe Olympic the schema of the table can be viewed.
* data can be loaded into table as follows:

***load data local inpath ‘path of your file’ into table olympic;***

* SELECT operation is shown below:

***Select \* from olympic;***

* Will select all the data inside the table.

**SEQUENCE FILE:**

* When we work in Hadoop with a small number of files with big size then its performance will be drawn out .
* Instead when a large number of files with small size.
* A file is taken as small one only when the size of the file is smaller than typical block size.
* When metadata count increases then there will be a overhead of the NameNode.
* For this kind of problem sequence files were used in Hadoop.
* Sequence files will act like a container in order to store those small files.
* Sequence files will be like flat files which has binary key-value pairs.
* When query is converted to MapReduce jobs, so appropriate key-value pairs will be decided for the given record.
* Sequence files will be in binary format
* So one can split those files and can club to two or more small files to make them as one sequence file.
* One can create a sequence file in HIVE by using STORED AS SEQUENCEFILE at the end of CREATE TABLE.
* Sequence files are of 3 types:

***1. Uncompressed key/value records.***

***2.Record compressed key/value records –*** only ‘values’ will be compressed here.

***3. Block compressed key/value records –*** both keys and values will be collected in these ‘blocks’ separately and will be compressed.

* The size of the ‘block’ will be configured.
* Hive will have its own SEQUENCEFILE reader as well SEQUENCEFILE writer for reading and writing purpose through sequence files.
* Creating sequence file format in Hive:

***create table table\_name (schema of the table) row format delimited fileds terminated by ',' | stored as SEQUENCEFILE***

Hive will use the SEQUENCEFILE input and output formats from these packages:

***org.apache.hadoop.mapred.SequenceFileInputFormat***

***org.apache.hadoop.hive.ql.io.HiveSequenceFileOutputFormat***

**Creating SEQUENCE FILE**

***create table olympic\_sequencefile(athelete STRING,age INT,country STRING,year STRING,closing STRING,sport STRING,gold INT,silver INT,bronze INT,total INT) row format delimited fields terminated by '\t' stored as sequencefile***

* A table is created with name olympic\_sequencefile .
* Data inside the file is delimited by tab space.
* At the end the file format is specified as SEQUENCEFILE format.
* You can check the schema of your created table using:

***describe olympic\_sequencefile;***

* Load data into the table will be different from loading the table created by TEXTFILE format.
* The data will be compressed later will be stored into the table.
* If the data need to be loaded directly in TEXTFILE format then it will not be possible.
* because one cannot insert a compressed files into tables.
* In order to load the data into SEQUENCEFILE:

***INSERT OVERWRITE TABLE olympic\_sequencefile***

***SELECT \* FROM olympic;***

* Already a table named olympic is created .
* It is of TEXTFILE format .
* Later the contents of the olympic table were written into olympic\_sequencefile table.

***select athelete from olympic\_sequencefile;***

**RC FILE:**

* RCFILE - Record Columnar File it is of binary file format which is provided with high compression rate on the top rows.
* RCFILE is used to perform operations at a time on multiple rows.
* RCFILEs acts as flat files which consist of binary key/value pairs.
* And will be similar to SEQUENCEFILE. RCFILE which will store columns of table in form of record in a columnar manner.
* During first partitions of rows horizontally into row will get splits
* Later will vertically partitions each row split in a columnar way.
* RCFILE will first stores metadata row split as key part of each record
* And all those data of row split as the value part.
* column oriented storage is important while performing the analytics.
* Analytics is easy to perform while we “hive’ as a column oriented storage type.
* Facebook makes use of RCFILE as default file format in order to store the data in data warehouse to perform different types of analytics by making use of Hive.
* In Hive RCFILE format is created as:

***create table table\_name (schema of the table) row format delimited fields terminated by ',' | stored as RCFILE***

Hive will have a own RCFILE Input format as well RCFILE output format in default package:

***org.apache.hadoop.hive.ql.io.RCFileInputFormat***

***org.apache.hadoop.hive.ql.io.RCFileOutputFormat***

**Creating RC FILE**

***create table olympic\_rcfile(athelete STRING,age INT,country STRING,year STRING,closing STRING,sport STRING,gold INT,silver INT,bronze INT,total INT) row format delimited fields terminated by '\t' stored as rcfile***

* A table is created with name olympic\_rcfile
* And the schema of the table is as specified.
* The data inside the input file is delimited by means of tab space.
* At the end the file format is specified as RCFILE format.
* You can check the schema of your created table using:

***describe olympic\_rcfile;***

* one cannot directly load data into RCFILE.
* First one should load the data into another table later the data need to be overwritten into newly created RCFILE as shown below:

***INSERT OVERWRITE TABLE olympic\_rcfile***

***SELECT \* FROM olympic;***

Already a table was named as olympic which is in the format of TEXTFILE .

While writing the Olympic contents into olympic\_rcfile table.

***select athelete from olympic\_rcfile;***

* **ORC FILE:**
* ORC - Optimized Row Columnar which will store data in a optimized way rather like other file formats.
* ORC will reduce the size of original data up to 75%.
* Which will result in increase of speed of data processing.
* ORC will show better performance than Text, Sequence and RC file formats.
* ORC file will contain rows data in groups called Stripes along with file footer. ORC format tends to improve the performance .
* While Hive is processing with the data.
* In Hive one can create a RCFILE format as follows:

***create table table\_name (schema of the table) row format delimited fields terminated by ',' | stored as ORC***

* Hive will have its own ORCFILE Input format as well ORCFILE output format in as default package:

***org.apache.hadoop.hive.ql.io.orc***

**Creating ORC FILE**

***create table olympic\_orcfile(athelete STRING,age INT,country STRING,year STRING,closing STRING,sport STRING,gold INT,silver INT,bronze INT,total INT) row format delimited fields terminated by '\t' stored as orcfile;***

* A table is created with name olympic\_orcfile as well the schema of the table specified.
* The data inside the input file will be delimited by means of tab space.
* End the file format is specified as ORCFILE format.
* One can check the schema of created table by making use of:

***describe olympic\_orcfile;***

* We cannot load data into ORCFILE directly. First we need to load data into another table and then we need to overwrite it into our newly created ORCFILE.

***INSERT OVERWRITE TABLE olympic\_orcfile***

***SELECT \* FROM olympic;***

* A table was created on the name olympic of TEXTFILE format and to write the contents of the olympic table into olympic\_orcfile table.
* This is how we have loaded data into the ORCFILE.
* In order to perform SELECT operation on TEXTFILE format, on ORCFILE format as well.

***select athlete from olympic\_orcfile;***

* Thus you can use the above four file formats depending on your data.

**For example,**

* When there is delimited data then one can use TEXT FILE format.
* If data is of small files where size is less than block size then one can use SEQUENCE FILE format.
* When on need to perform analytics of data as well to store the data efficiently then one can use RC FILE format.
* To store data in a optimized way to decrease the storage and to increases the performance then one can make use of ORC FILE format.