**Problem Statement:**

**Explain File Formats in Hive in Brief with an example.**

**1. TEXTFILE:**

TEXTFILE format is a famous input/output format used in Hadoop. In Hive if we define a table as TEXTFILE it can load data of from CSV (Comma Separated Values), delimited by Tabs, Spaces, and JSON data. This means fields in each record should be separated by comma or space or tab or it may be JSON (JavaScript Object Notation) data.  
By default, if we use TEXTFILE format then each line is considered as a record.

**We can create a TEXTFILE format in Hive as follows:**

|  |  |
| --- | --- |
| 1 | create table table\_name (schema of the table) row format delimited fields terminated by ',' | stored as TEXTFILE. |

At the end, we need to specify the type of **file format**. If we do not specify anything it will consider the file format as TEXTFILE format.  
The TEXTFILE input and TEXTFILE output format are present in the Hadoop package as shown below:

|  |  |
| --- | --- |
| 1  2 | org.apache.hadoop.mapred.TextInputFormat  org.apache.hadoop.mapred.TextOutputFormat |

Let us see one example in Hive about how to create TEXTFILE table format, how to load data into TEXTFILE format and perform one basic select operation in Hive.

**Creating TEXTFILE**

|  |  |
| --- | --- |
| 1 | create table olympic(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 textfile; |

Here we are creating a table with name “olympic” and the schema of the table is as specified above. The data inside the above input file is delimited by tab space. As explained earlier the file format is specified as TEXTFILE at the end. The schema of the table created above can be checked using describe olympic;

**2. SEQUENCEFILE:**

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.

In Hive we can create a sequence file by specifying STORED AS SEQUENCEFILE in the end of a CREATE TABLE statement.  
There are three types of sequence files :  
• Uncompressed key/value records.  
• Record compressed key/value records – only ‘values’ are compressed here  
• Block compressed key/value records – both keys and values are collected in ‘blocks’ separately and compressed. The size of the ‘block’ is configurable.

**Creating SEQUENCEFILE**

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

**To load the data into SEQUENCEFILE we need to use the following approach:**

|  |  |
| --- | --- |
|  | INSERT OVERWRITE TABLE olympic\_sequencefile  SELECT \* FROM olympic; |

**3.RC FILE:**

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.

This column oriented storage is very useful while performing analytics. It is easy to perform analytics when we “hive’ a column oriented storage type.

Facebook uses RCFILE as its default file format for storing of data in their data warehouse as they perform different types of analytics using Hive.

In Hive we can create a RCFILE format as follows:

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

Hive has its own RCFILE Input format and RCFILE output format in its 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**

Here we are creating a table with name olympic\_rcfile and the schema of the table is as specified above. The data inside the input file is delimited by 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;**

We cannot load data into RCFILE directly. First we need to load data into another table and then we need to overwrite it into our newly created RCFILE as shown below:

**INSERT OVERWRITE TABLE olympic\_rcfile**

**SELECT \* FROM olympic;**

We have already created table by name olympic which is of TEXTFILE format and then we are writing the contents of the olympic table into olympic\_rcfile table.

Now let us perform the same basic SELECT operation which we have performed on the TEXTFILE format on RCFILE format as well as shown in Figure below:

**select athelete from olympic\_rcfile;**

**4. ORCFILE:**

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.

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

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

**Creating ORCFILE**

|  |  |
| --- | --- |
|  | 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; |