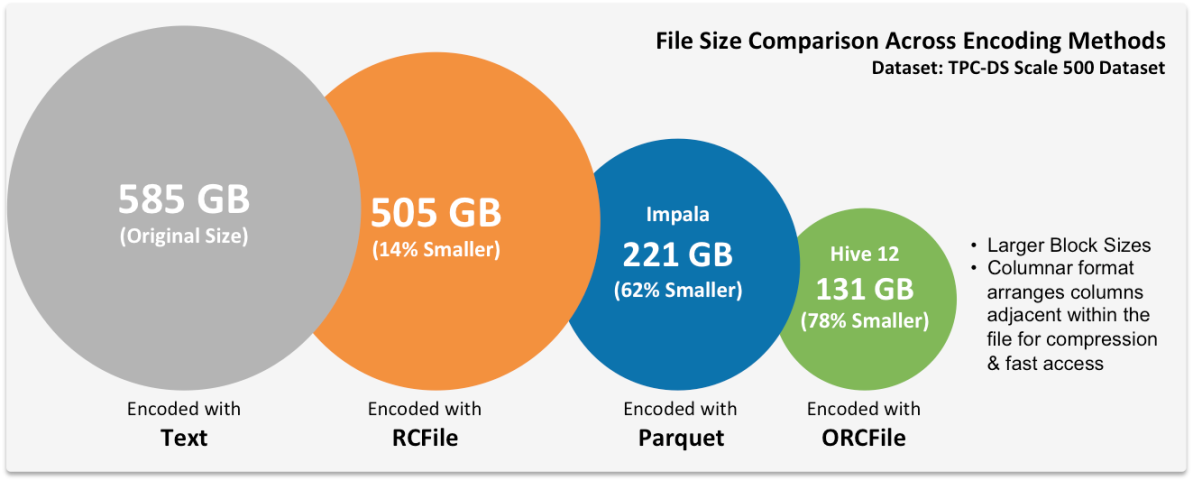
Hive Compression Codecs:

****

**4mc, snappy, lzo, lz4, bzip2, and gzip**.

* 4mc com.hadoop.compression.fourmc.FourMcCodec
* gzip org.apache.hadoop.io.compress.GzipCodec
* lzo com.hadoop.compression.lzo.LzopCodec
* Snappy org.apache.hadoop.io.compress.SnappyCodec
* bzip2 org.apache.hadoop.io.compress.BZip2Codec
* lz4 org.apache.hadoop.io.compress.Lz4Codec

**Types of UDFs in HIVE:**

Hive Functions: User Defined Functions (UDFs) in hive are used to plug in our own logic in terms of code into hive when we are not able to get the desired result from hive's built in functions. We can invoke the UDFs from hive query.

There are three kind of UDFs in Hive:

**1.Regular UDF**, **2. User Defined Aggregate Function (UDAF),** **3.User Defined Tabular Function (UDTF)**

1.Regular UDF: UDFs works on a single row in a table and produces a single row as output. Its one to one relationship between input and output of a function. e.g Hive built in TRIM() function.

Hive allows us to define our own UDFs as well. Lets take an example of student record.

Problem Statement: Find the maximum marks obtained out of four subject by an student.

Sample Input: STUD\_ID, NAME, CLASS, TOT\_MARKS, MATH, ENGLISH, PHYSICS, SOCIAL\_STUDY, YEAR

Steps to Create & Execute UDFs in Java:Step1: We have to extend a base Class UDF to write our business logic in Java. Step2: In order to write business logic , we have to overload a method called evaluate() inside our class.

Step3: We need to export the JAR files to HDFS where hive is running. In my case , I am running hive from my local directory path /home/GAURAV/HIVE, so I have copied the exported JAR file in this path.

Step4: Add the exported JAR file to hive classpath using below command from hive terminal: ADD JAR EXPORTED\_FILE\_NAME.jar Alternate: You can add exported JAR files in bashrc file using command "nano ~/.bashrc" as HIVE\_AUX\_JAR\_PATH = '/home/GAURAV/HIVE/HIVE-UDF-MAXMARKS.jar'. It will avoid adding your hive jar to class path each time you login to hive session or hadoop as it will be loaded during hadoop cluster loading by the framework itself.

Step5: In order to apply business logic on top of hive column using our UDF, we need to create a temporary function for the exported jar file. CREATE temporary function func\_name as 'absolute\_class\_path\_name'.

**2.UDAF:** User defined aggregate functions works on more than one row and gives single row as output. **e.g Hive built in MAX() or COUNT() functions**. here the relation is many to one. Lets say you

have a table with students name, id and total marks, so here if I have 10 rows in the table and if I have to find student who got maximum number then our query need to check each 10 row to find the maximum

but ultimately we get only one output which is the maximum. Hope this justifies the many to one relationship.

Hive allows us to define our own UDAFs. Lets take an example of student record as explained in UDFs example with same input.

3.UDTF: User defined tabular function works on one row as input and returns multiple rows as output. So here the relation in one to many. e.g Hive built in EXPLODE() function. Now lets take an array column USER\_IDS as ARRAY10,12,5,45> then SELECT EXPLODE(USER\_IDS) as ID FROM T\_USER. will give 10,12,5,45 as four different rows in output. UDTF can be used to split a column into multiple column as well which we will look in below example. Here alias "AS" clause is mandatory .