**ACD\_BDD\_Session\_16\_Assignment\_2\_Main**

**Problem Statement**

Link: https://acadgild.com/blog/indexing-in-hive/

Follow the blog link given above and implement the concepts of Indexing.

Perform the operations given in the above blog using your own sample dataset and send us the code snippets and output screenshot.

**Solution**

There are 2 types of Indexes in Hive:

1. Compact Indexing

2. Bitmap Indexing

The following are the different operations that can be performed on Hive indexes:

1. Creating index

2. Showing index

3. Alter index

4. Dropping index

**Creating Index in Hive**

Syntax for creating a compact index in Hive is as follows:

CREATE INDEX index\_name

ON TABLE table\_name (columns,....)

AS 'org.apache.hadoop.hive.ql.index.compact.CompactIndexHandler'

WITH DEFERRED REBUILD;

**Table Creation**

CREATE TABLE olympic\_index(

Athlete STRING,

Age INT,

Country STRING,

Year INT,

Closing\_Date STRING,

Sport STRING,

Gold\_Medal INT,

Silver\_Medal INT,

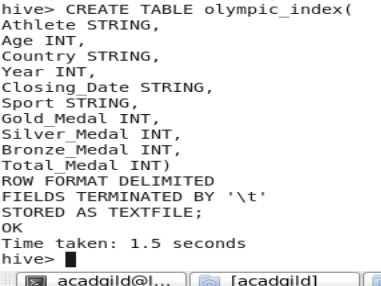
Bronze\_Medal INT,

Total\_Medal INT)

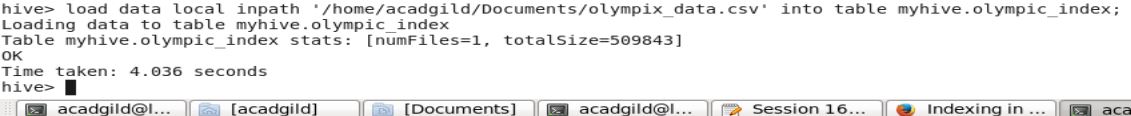
ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t'

STORED AS TEXTFILE;



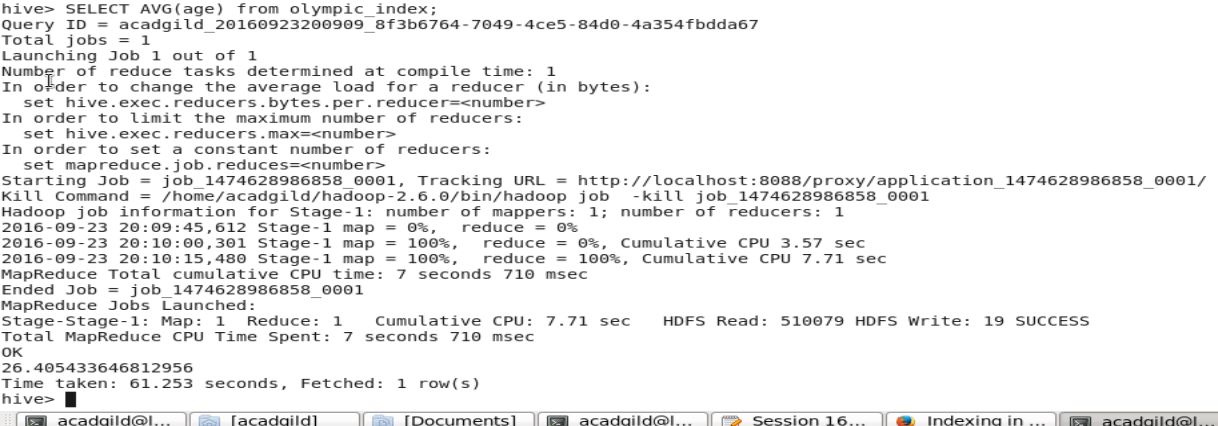
load data local inpath '/home/acadgild/Documents/olympix\_data.csv' into table myhive.olympic\_index;



SELECT AVG(age) from olympic\_index;

26.405433646812956

Time taken: 61.253 seconds, Fetched: 1 row(s)



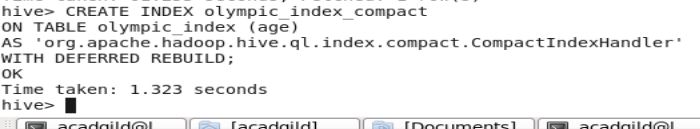
**Creating index for the table olympic\_index:**

CREATE INDEX olympic\_index\_compact

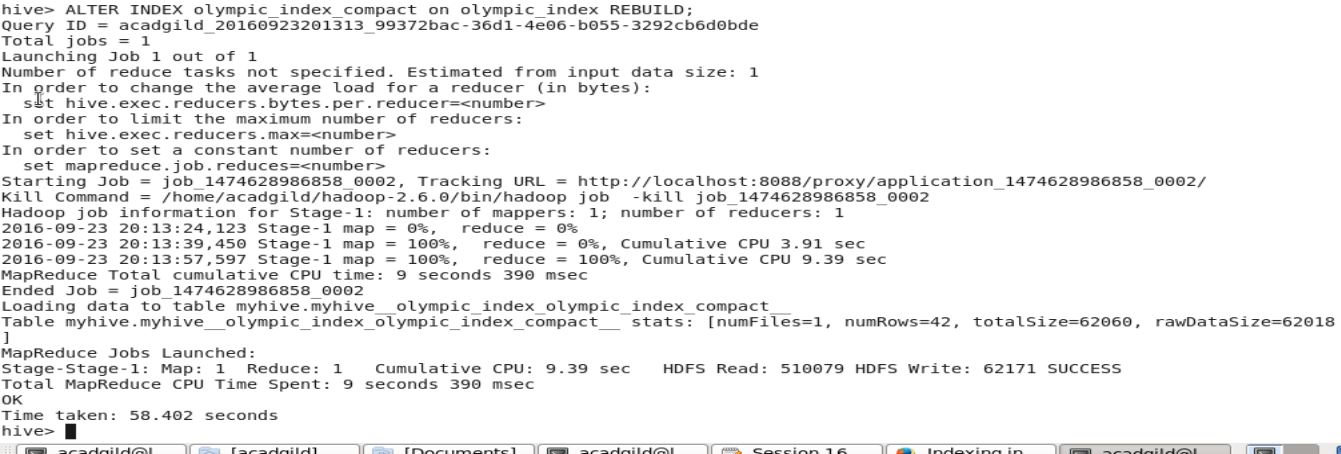
ON TABLE olympic\_index (age)

AS 'org.apache.hadoop.hive.ql.index.compact.CompactIndexHandler'

WITH DEFERRED REBUILD;



ALTER INDEX olympic\_index\_compact on olympic\_index REBUILD;



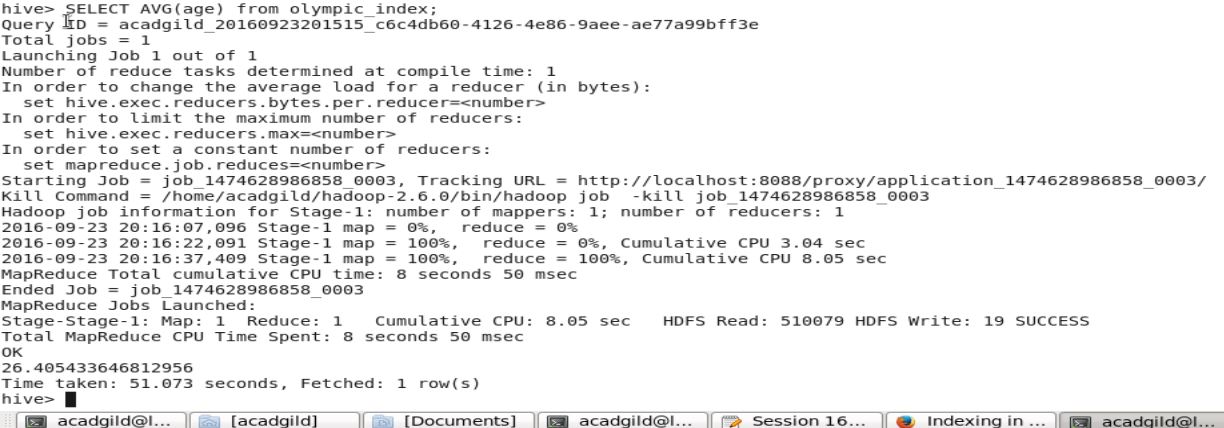
show formatted index on olympic\_index;



SELECT AVG(age) from olympic\_index;

26.405433646812956

Time taken: 51.073 seconds, Fetched: 1 row(s)



**Creating a Bitmap index for the same table:**

CREATE INDEX olympic\_index\_bitmap

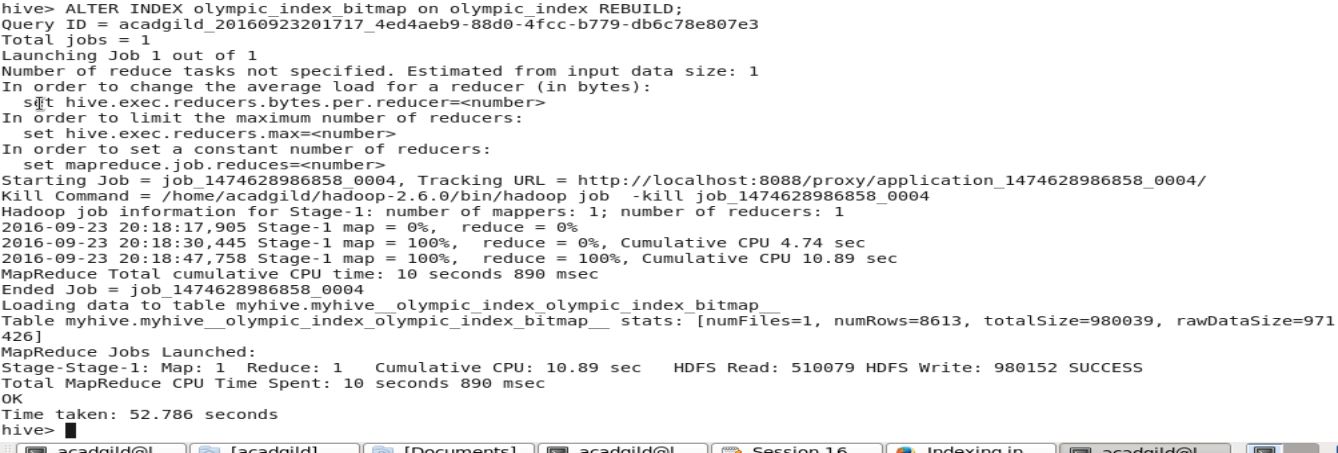
ON TABLE olympic\_index (age)

AS 'BITMAP'

WITH DEFERRED REBUILD;



ALTER INDEX olympic\_index\_bitmap on olympic\_index REBUILD;



show formatted index on olympic\_index;

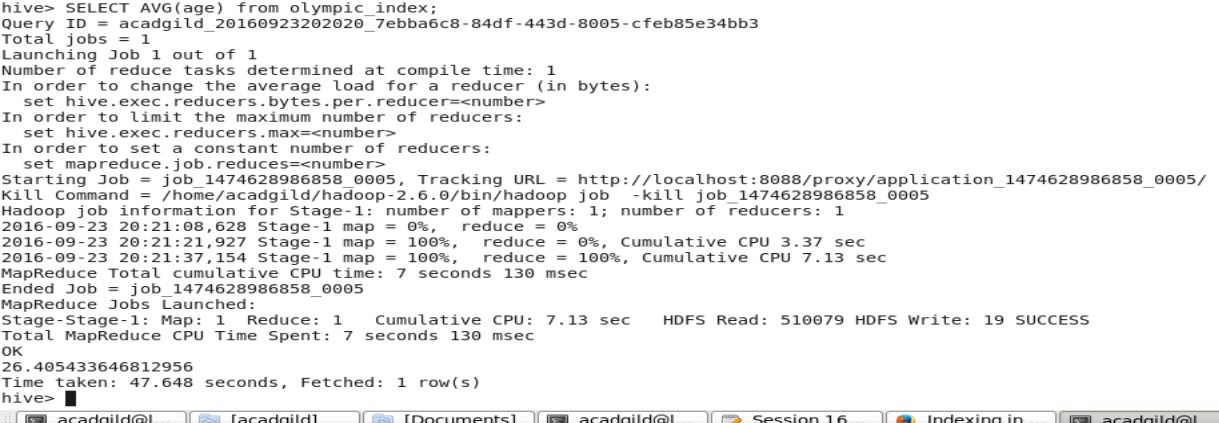


Now, performing select operation having 2 indexes:

SELECT AVG(age) from olympic\_index;

26.405433646812956

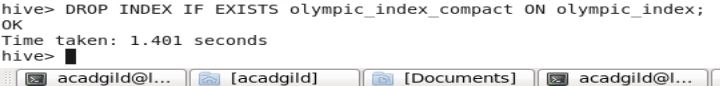
Time taken: 47.648 seconds, Fetched: 1 row(s)



**Note:** With different types (compact,bitmap) of indexes on the same columns, for the same table, the index which is created first is taken as the index for that table on the specified columns.

Now let's delete one index using the following command:

DROP INDEX IF EXISTS olympic\_index\_compact ON olympic\_index;



Average Operation with Bitmap Index:

SELECT AVG(age) from olympic\_index;

26.405433646812956

Time taken: 48.216 seconds, Fetched: 1 row(s)

