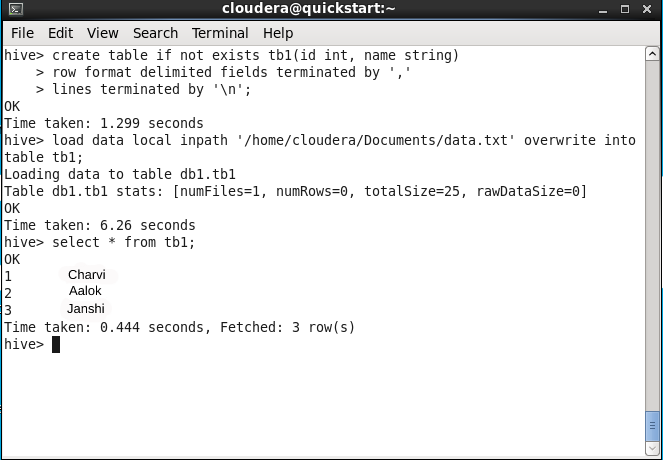
**#HIVE DAY-1 ASSIGNMENT-1**

**SIRSS1024**

**Charvi Dosi**

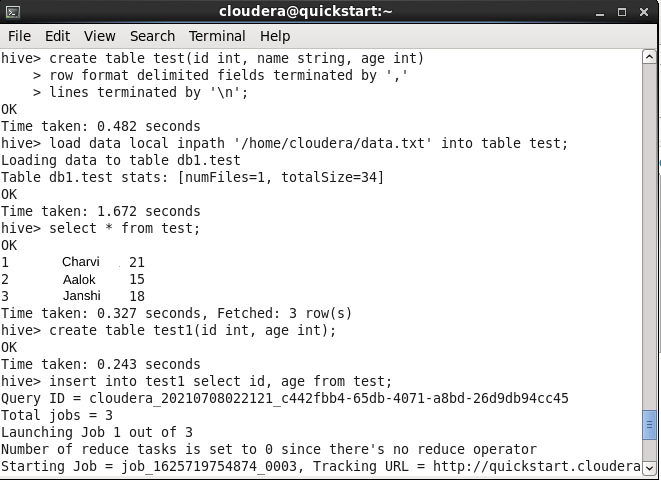
**Q.1. Try to load the data from the file located in /home/cloudera/Documents folder - Load the data into the table created**

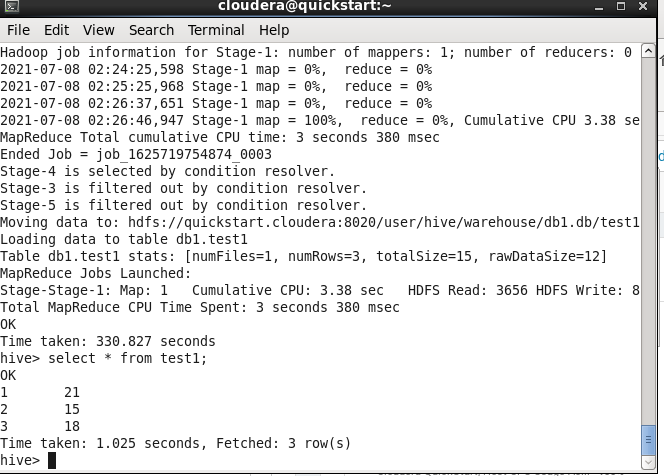
**- Try to load data into table again but make sure you don't append the data into the table**



**Q.2. Take the file data.txt inside this we have (3 columns)**

**- Load the data of data.txt into the hive table but take only 2 columns while loading the data into the table from data.txt**





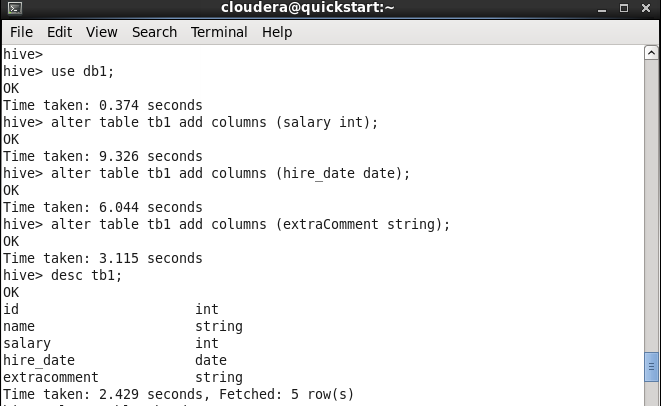
**Q.3. Let assume we have a table**

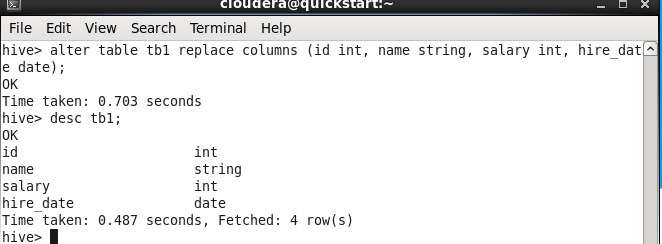
**- we need to add new column as salary with int data type**

**- hire\_date column as date datatype**

**- add new columns extraComment with string datatype**

**- remove the extraComment column from the table using the alter command**





**Q.4. What is partitioning & bucketing in hive**

**- Why bucketing is useful or partitioning with their limitations (theory)**

**Hive Partitioning:**

Partition divides large amount of data into multiple slices based on value of a table column(s).

Assume that you are storing information of people in entire world spread across 196+ countries spanning around 500 crores of entries. If you want to query people from a particular country (Vatican city), in absence of partitioning, you have to scan all 500 crores of entries even to fetch thousand entries of a country. If you partition the table based on country, you can fine tune querying process by just checking the data for only one country partition. Hive partition creates a separate directory for a column(s) value.

**Pros:**

Distribute execution load horizontally

Faster execution of queries in case of partition with low volume of data. e.g. Get the population from "Vatican city" returns very fast instead of searching entire population of world.

**Cons:**

Possibility of too many small partition creations - too many directories.

Effective for low volume data for a given partition. But some queries like group by on high volume of data still take long time to execute. e.g. Grouping of population of China will take long time compared to grouping of population in Vatican city. Partition is not solving responsiveness problem in case of data skewing towards a particular partition value.

**Hive Bucketing:**

Bucketing decomposes data into more manageable or equal parts.

With partitioning, there is a possibility that you can create multiple small partitions based on column values. If you go for bucketing, you are restricting number of buckets to store the data. This number is defined during table creation scripts.

**Pros**

Due to equal volumes of data in each partition, joins at Map side will be quicker.

Faster query response like partitioning

**Cons**

You can define number of buckets during table creation but loading of equal volume of data has to be done manually by programmers.