IMAAD IMRAN HAJWANE

202101132 / 21

BIG DATA ANALYTICS – LAB 5

Hive Partitioning

Partitioning is a method of dividing a large dataset into smaller, manageable parts, based on the values of one or more columns. It is an optimization technique that helps improve query performance in Hive by reducing the amount of data read during query execution. When a query is run on partitioned data, only the relevant partitions are scanned, leading to faster processing.

Types of Partitioning:

1. **Static Partitioning**:
   - In static partitioning, the partitions are explicitly defined by the user when loading data into a table.
   - For example, consider a table storing sales data where the partition is based on the `year` column. The user must specify the partition value when inserting data, such as `year=2020`.
   - It requires more manual intervention but allows for precise control over how data is organized.
2. **Dynamic Partitioning**:
   - Dynamic partitioning automatically creates partitions based on the data being loaded.
   - Instead of manually specifying the partition key, Hive determines the partitions dynamically based on the column values at runtime.
   - For example, if the sales data includes different years (2018, 2019, 2020) in one dataset, Hive will automatically create partitions for each year.
   - It is useful when data for multiple partitions is being loaded simultaneously.

Advantages of Partitioning:

- **Improved Query Performance**: When querying partitioned data, Hive scans only the required partitions instead of the entire dataset. This reduces the input/output (I/O) operations and speeds up query execution.
- **Organized Data**: Partitioning helps organize large datasets logically, making it easier to manage and retrieve specific data subsets.

Hive Bucketing

Bucketing is another technique in Hive to optimize query performance by dividing data into more manageable segments (buckets). Unlike partitioning, bucketing distributes data within a partition into equally-sized buckets using a hashing function. Each bucket can then be treated as a separate file.

How Bucketing Works:

- In bucketing, Hive divides the data into buckets based on the hash value of the bucketed column (often an integer).
- For example, if a table is bucketed by the `emp_id` column into 4 buckets, Hive applies a hash function to `emp_id` values to distribute the rows across 4 different buckets. Rows with similar hash values end up in the same bucket.
- This makes operations like joins more efficient, as similar data is stored in the same bucket across tables.

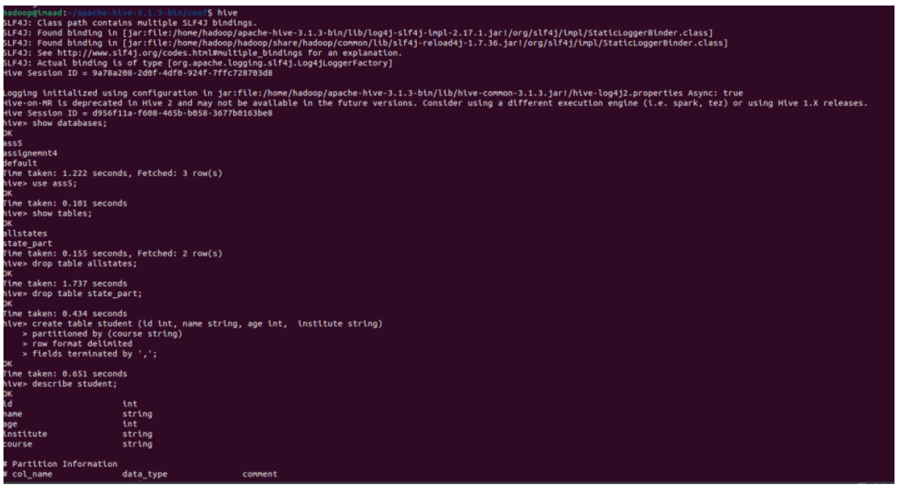
Advantages of Bucketing:

- **Efficient Joins**: Bucketing is particularly useful in join operations, as data with the same bucket value can be co-located. This minimizes the amount of data shuffled across the cluster during joins.
- **Sampling**: Bucketing enables Hive to perform sampling efficiently. Since the data is evenly distributed across buckets, Hive can select a subset of data by sampling specific buckets.

PARTITIONING

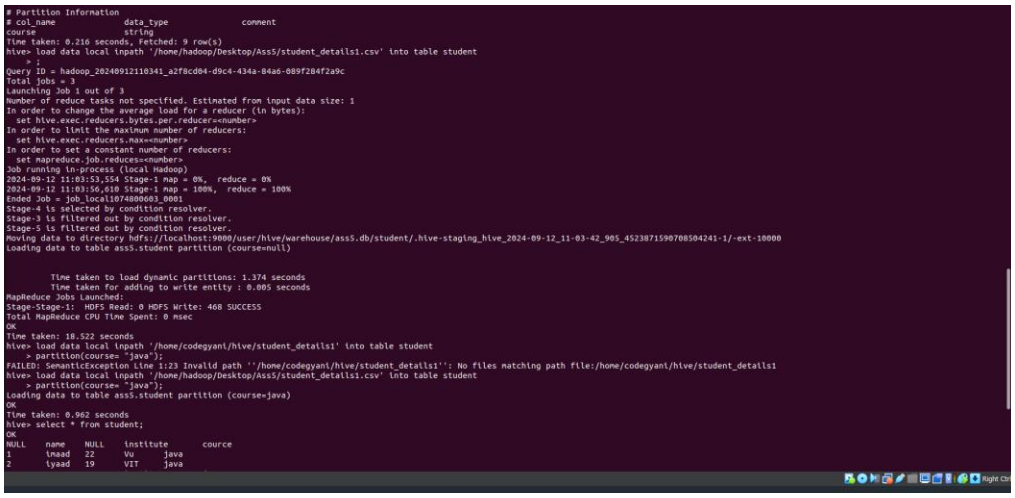
Static

- Initialising, Displaying & Creating table in database



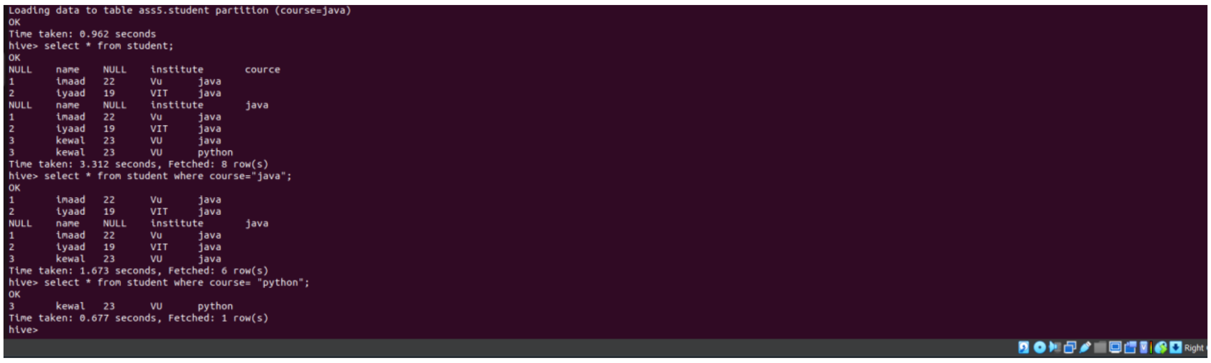
```
hadoop@imaad:~/apache-hive-3.1.3-bin/conf$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoop/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = 9a78a208-2d0f-4df0-924f-7ffc728703d8

Logging initialized using configuration in jar:file:/home/hadoop/apache-hive-3.1.3-bin/lib/hive-common-3.1.3.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Hive Session ID = d956f11a-f608-465b-b058-3677b0163be8
hive> show databases;
OK
ass5
assignemnt4
default
Time taken: 1.222 seconds, Fetched: 3 row(s)
hive> use ass5;
OK
Time taken: 0.101 seconds
hive> show tables;
OK
allstates
state_part
Time taken: 0.155 seconds, Fetched: 2 row(s)
hive> drop table allstates;
OK
Time taken: 1.737 seconds
hive> drop table state_part;
OK
Time taken: 0.434 seconds
hive> create table student (id int, name string, age int,  institute string)
    > partitioned by (course string)
    > row format delimited
    > fields terminated by ',';
OK
Time taken: 0.651 seconds
hive> describe student;
OK
id                      int
name                    string
age                     int
institute               string
course                  string

# Partition Information
# col_name              data_type               comment
```

- Loading data using “LOAD” commands and displaying data stored



```
# Partition Information
# col_name              data_type               comment
course                  string
Time taken: 0.216 seconds, Fetched: 9 row(s)
hive> load data local inpath '/home/hadoop/Desktop/Ass5/student_details1.csv' into table student
    > ;
Query ID = hadoop_20240912110341_a2f8cd04-d9c4-434a-84a6-089f284f2a9c
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2024-09-12 11:03:53,554 Stage-1 map = 0%,  reduce = 0%
2024-09-12 11:03:56,610 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local1074800603_0001
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/ass5.db/student/.hive-staging_hive_2024-09-12_11-03-42_905_4523871590708504241-1/-ext-10000
Loading data to table ass5.student partition (course=null)


         Time taken to load dynamic partitions: 1.374 seconds
         Time taken for adding to write entity : 0.005 seconds
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 0 HDFS Write: 468 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 18.522 seconds
hive> load data local inpath '/home/codegyani/hive/student_details1' into table student
    > partition(course= "java");
FAILED: SemanticException Line 1:23 Invalid path ''/home/codegyani/hive/student_details1'': No files matching path file:/home/codegyani/hive/student_details1
hive> load data local inpath '/home/hadoop/Desktop/Ass5/student_details1.csv' into table student
    > partition(course= "java");
Loading data to table ass5.student partition (course=java)
OK
Time taken: 0.962 seconds
hive> select * from student;
OK
NULL    name    NULL    institute       cource
1       imaad   22      Vu      java
2       iyaad   19      VIT     java
```

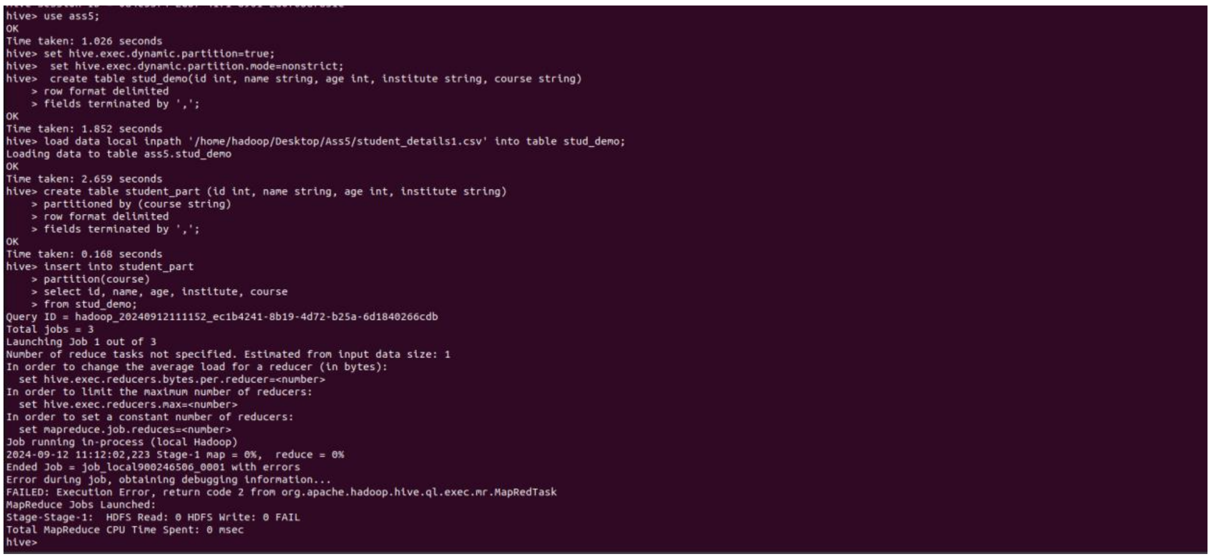
- Final display of stored data



```
Loading data to table ass5.student partition (course=java)
OK
Time taken: 0.962 seconds
hive> select * from student;
OK
NULL    name    NULL    institute       cource
1       imaad   22      Vu      java
2       iyaad   19      VIT     java
NULL    name    NULL    institute       java
1       imaad   22      Vu      java
2       iyaad   19      VIT     java
3       kewal   23      VU      java
3       kewal   23      VU      python
Time taken: 3.312 seconds, Fetched: 8 row(s)
hive> select * from student where course="java";
OK
1       imaad   22      Vu      java
2       iyaad   19      VIT     java
NULL    name    NULL    institute       java
1       imaad   22      Vu      java
2       iyaad   19      VIT     java
3       kewal   23      VU      java
Time taken: 1.673 seconds, Fetched: 6 row(s)
hive> select * from student where course= "python";
OK
3       kewal   23      VU      python
Time taken: 0.677 seconds, Fetched: 1 row(s)
hive>
```

Dynamic

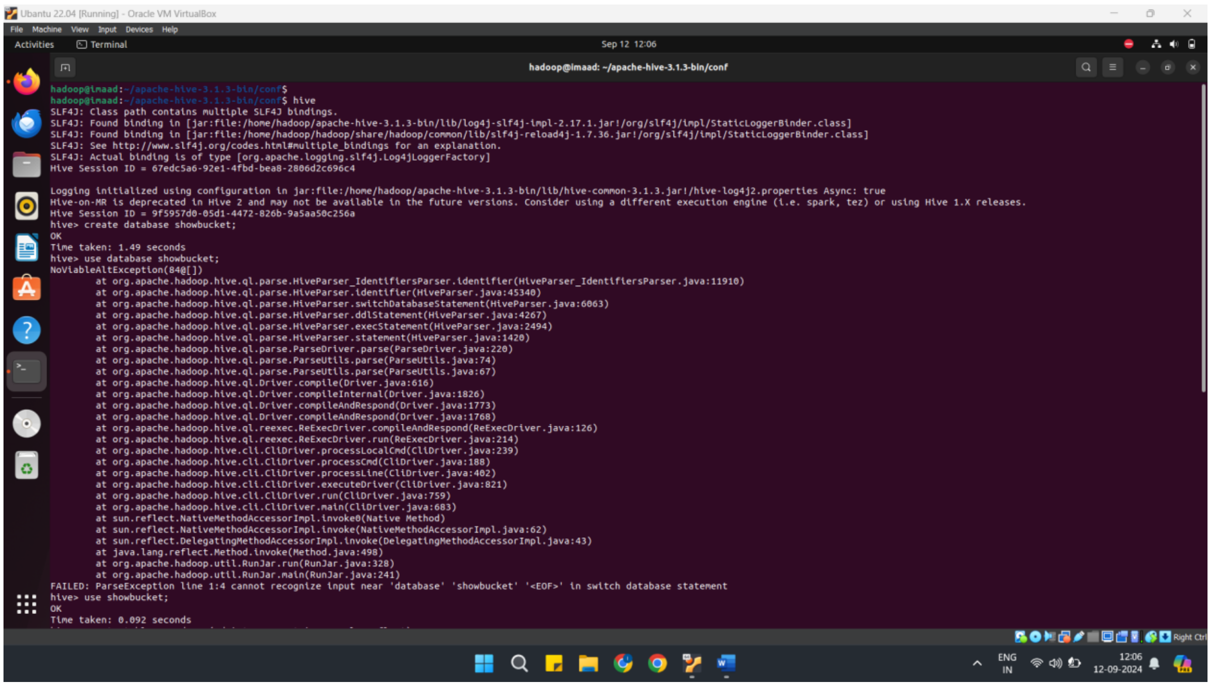
- Creation, storing & Execution



```
hive> use ass5;
OK
Time taken: 1.026 seconds
hive> set hive.exec.dynamic.partition=true;
hive>  set hive.exec.dynamic.partition.mode=nonstrict;
hive>  create table stud_demo(id int, name string, age int, institute string, course string)
    > row format delimited
    > fields terminated by ',';
OK
Time taken: 1.852 seconds
hive> load data local inpath '/home/hadoop/Desktop/Ass5/student_details1.csv' into table stud_demo;
Loading data to table ass5.stud_demo
OK
Time taken: 2.659 seconds
hive> create table student_part (id int, name string, age int, institute string)
    > partitioned by (course string)
    > row format delimited
    > fields terminated by ',';
OK
Time taken: 0.168 seconds
hive> insert into student_part
    > partition(course)
    > select id, name, age, institute, course
    > from stud_demo;
Query ID = hadoop_20240912111152_ec1b4241-8b19-4d72-b25a-6d1840266cdb
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2024-09-12 11:12:02,223 Stage-1 map = 0%,  reduce = 0%
Ended Job = job_local900246506_0001 with errors
Error during job, obtaining debugging information...
FAILED: Execution Error, return code 2 from org.apache.hadoop.hive.ql.exec.mr.MapRedTask
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 0 HDFS Write: 0 FAIL
Total MapReduce CPU Time Spent: 0 msec
hive>
```

BUCKETING

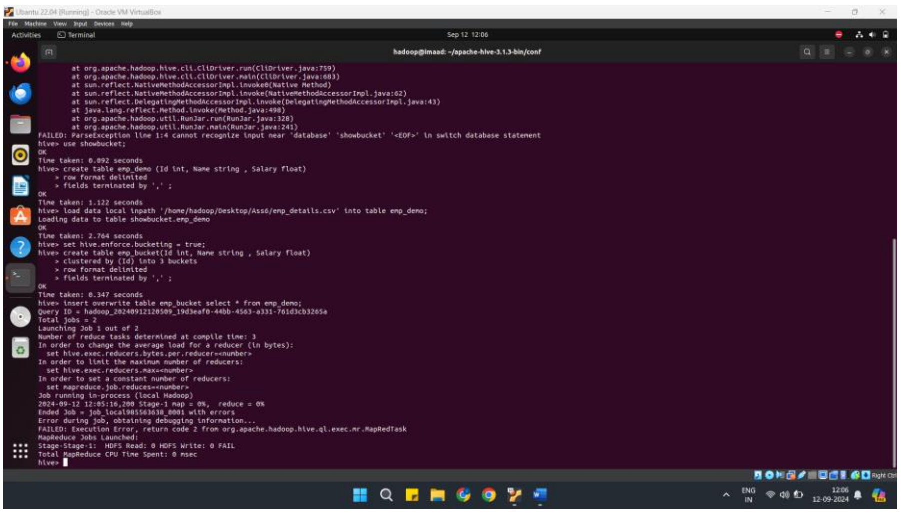
- Initializing, Creating & Displaying buckets



Ubantu 22.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Sep 12 12:06
hadoop@imaad: ~/apache-hive-3.1.3-bin/conf
ENG IN
12:06 12-09-2024

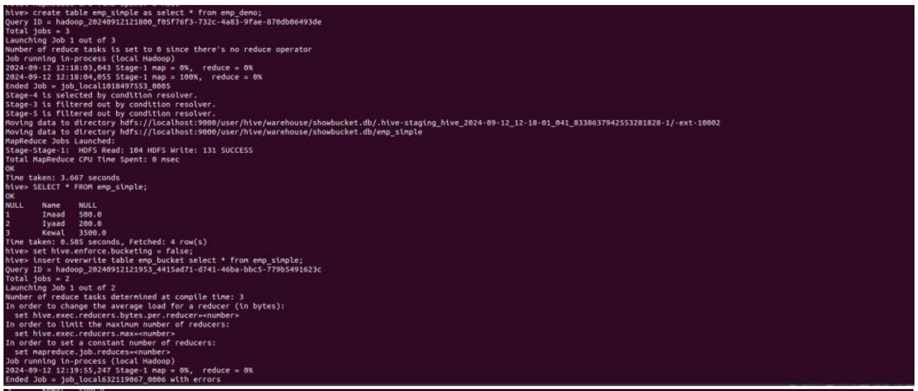
```
hadoop@imaad:~/apache-hive-3.1.3-bin/conf$
hadoop@imaad:~/apache-hive-3.1.3-bin/conf$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoop/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = 67edc5a6-92e1-4fbd-bea8-2806d2c696c4

Logging initialized using configuration in jar:file:/home/hadoop/apache-hive-3.1.3-bin/lib/hive-common-3.1.3.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Hive Session ID = 9f5957d0-05d1-4472-826b-9a5aa50c256a
hive> create database showbucket;
OK
Time taken: 1.49 seconds
hive> use database showbucket;
NoViableAltException(84@[])
        at org.apache.hadoop.hive.ql.parse.HiveParser_IdentifiersParser.identifier(HiveParser_IdentifiersParser.java:11910)
        at org.apache.hadoop.hive.ql.parse.HiveParser.identifier(HiveParser.java:45340)
        at org.apache.hadoop.hive.ql.parse.HiveParser.switchDatabaseStatement(HiveParser.java:6063)
        at org.apache.hadoop.hive.ql.parse.HiveParser.ddlStatement(HiveParser.java:4267)
        at org.apache.hadoop.hive.ql.parse.HiveParser.execStatement(HiveParser.java:2494)
        at org.apache.hadoop.hive.ql.parse.HiveParser.statement(HiveParser.java:1420)
        at org.apache.hadoop.hive.ql.parse.ParseDriver.parse(ParseDriver.java:220)
        at org.apache.hadoop.hive.ql.parse.ParseUtils.parse(ParseUtils.java:74)
        at org.apache.hadoop.hive.ql.parse.ParseUtils.parse(ParseUtils.java:67)
        at org.apache.hadoop.hive.ql.Driver.compile(Driver.java:616)
        at org.apache.hadoop.hive.ql.Driver.compileInternal(Driver.java:1826)
        at org.apache.hadoop.hive.ql.Driver.compileAndRespond(Driver.java:1773)
        at org.apache.hadoop.hive.ql.Driver.compileAndRespond(Driver.java:1768)
        at org.apache.hadoop.hive.ql.reexec.ReExecDriver.compileAndRespond(ReExecDriver.java:126)
        at org.apache.hadoop.hive.ql.reexec.ReExecDriver.run(ReExecDriver.java:214)
        at org.apache.hadoop.hive.cli.CliDriver.processLocalCmd(CliDriver.java:239)
        at org.apache.hadoop.hive.cli.CliDriver.processCmd(CliDriver.java:188)
        at org.apache.hadoop.hive.cli.CliDriver.processLine(CliDriver.java:402)
        at org.apache.hadoop.hive.cli.CliDriver.executeDriver(CliDriver.java:821)
        at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:759)
        at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:683)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:328)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:241)
FAILED: ParseException line 1:4 cannot recognize input near 'database' 'showbucket' '<EOF>' in switch database statement
hive> use showbucket;
OK
Time taken: 0.092 seconds
```

- Execution of bucketing

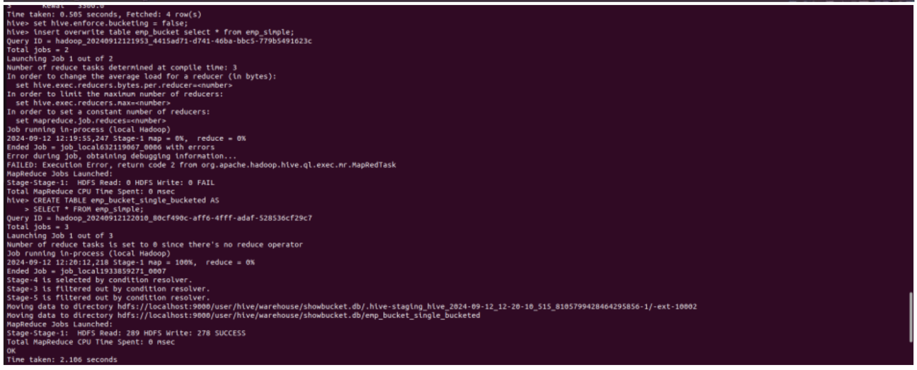


```
        at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:759)
        at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:683)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:328)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:241)
FAILED: ParseException line 1:4 cannot recognize input near 'database' 'showbucket' '<EOF>' in switch database statement
hive> use showbucket;
OK
Time taken: 0.092 seconds
hive> create table emp_demo (Id int, Name string , Salary float)
    > row format delimited
    > fields terminated by ',' ;
OK
Time taken: 1.122 seconds
hive> load data local inpath '/home/hadoop/Desktop/Ass6/emp_details.csv' into table emp_demo;
Loading data to table showbucket.emp_demo
OK
Time taken: 2.764 seconds
hive> set hive.enforce.bucketing = true;
hive> create table emp_bucket(Id int, Name string , Salary float)
    > clustered by (Id) into 3 buckets
    > row format delimited
    > fields terminated by ',' ;
OK
Time taken: 0.347 seconds
hive> insert overwrite table emp_bucket select * from emp_demo;
Query ID = hadoop_20240912120509_19d3eaf0-44bb-4563-a331-761d3cb3265a
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks determined at compile time: 3
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2024-09-12 12:05:16,200 Stage-1 map = 0%,  reduce = 0%
Ended Job = job_local985563638_0001 with errors
Error during job, obtaining debugging information...
FAILED: Execution Error, return code 2 from org.apache.hadoop.hive.ql.exec.mr.MapRedTask
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 0 HDFS Write: 0 FAIL
Total MapReduce CPU Time Spent: 0 msec
hive> 
```

- Final Display of buckets values



```
hive> create table emp_simple as select * from emp_demo;
Query ID = hadoop_20240912121800_f05f76f3-732c-4a83-9fae-870db06493de
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Job running in-process (local Hadoop)
2024-09-12 12:18:03,043 Stage-1 map = 0%,  reduce = 0%
2024-09-12 12:18:04,055 Stage-1 map = 100%,  reduce = 0%
Ended Job = job_local1018497553_0005
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/showbucket.db/.hive-staging_hive_2024-09-12_12-18-01_041_8338637942553281828-1/-ext-10002
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/showbucket.db/emp_simple
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 104 HDFS Write: 131 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 3.667 seconds
hive> SELECT * FROM emp_simple;
OK
NULL    Name    NULL
1       Imaad   500.0
2       Iyaad   200.0
3       Kewal   3500.0
Time taken: 0.505 seconds, Fetched: 4 row(s)
hive> set hive.enforce.bucketing = false;
hive> insert overwrite table emp_bucket select * from emp_simple;
Query ID = hadoop_20240912121953_4415ad71-d741-46ba-bbc5-779b5491623c
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks determined at compile time: 3
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2024-09-12 12:19:55,247 Stage-1 map = 0%,  reduce = 0%
Ended Job = job_local632119067_0006 with errors
```



```
Time taken: 0.505 seconds, Fetched: 4 row(s)
hive> set hive.enforce.bucketing = false;
hive> insert overwrite table emp_bucket select * from emp_simple;
Query ID = hadoop_20240912121953_4415ad71-d741-46ba-bbc5-779b5491623c
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks determined at compile time: 3
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2024-09-12 12:19:55,247 Stage-1 map = 0%,  reduce = 0%
Ended Job = job_local632119067_0006 with errors
Error during job, obtaining debugging information...
FAILED: Execution Error, return code 2 from org.apache.hadoop.hive.ql.exec.mr.MapRedTask
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 0 HDFS Write: 0 FAIL
Total MapReduce CPU Time Spent: 0 msec
hive> CREATE TABLE emp_bucket_single_bucketed AS
    > SELECT * FROM emp_simple;
Query ID = hadoop_20240912122010_80cf490c-aff6-4fff-adaf-528536cf29c7
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Job running in-process (local Hadoop)
2024-09-12 12:20:12,218 Stage-1 map = 100%,  reduce = 0%
Ended Job = job_local1933859271_0007
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/showbucket.db/.hive-staging_hive_2024-09-12_12-20-10_515_8105799428464295856-1/-ext-10002
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/showbucket.db/emp_bucket_single_bucketed
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 289 HDFS Write: 278 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 2.106 seconds
```