



Apache Hive

by Sumit Mittal



IMPORTANT

Copyright Infringement and Illegal Content Sharing Notice

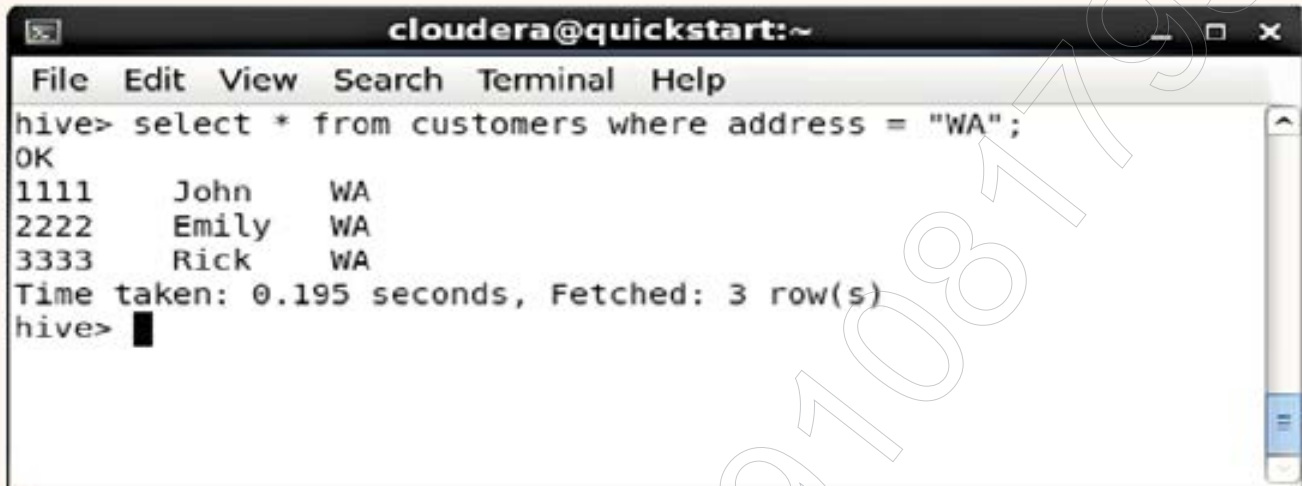
All course content designs, video, audio, text, graphics, logos, images are Copyright© and are protected by India and international copyright laws. All rights reserved.

Permission to download the contents (wherever applicable) for the sole purpose of individual reading and preparing yourself to crack the interview only. Any other use of study materials – including reproduction, modification, distribution, republishing, transmission, display – without the prior written permission of Author is strictly prohibited.

Trendytech Insights legal team, along with thousands of our students, actively searches the Internet for copyright infringements. Violators subject to prosecution.

Display table data with **where** condition:

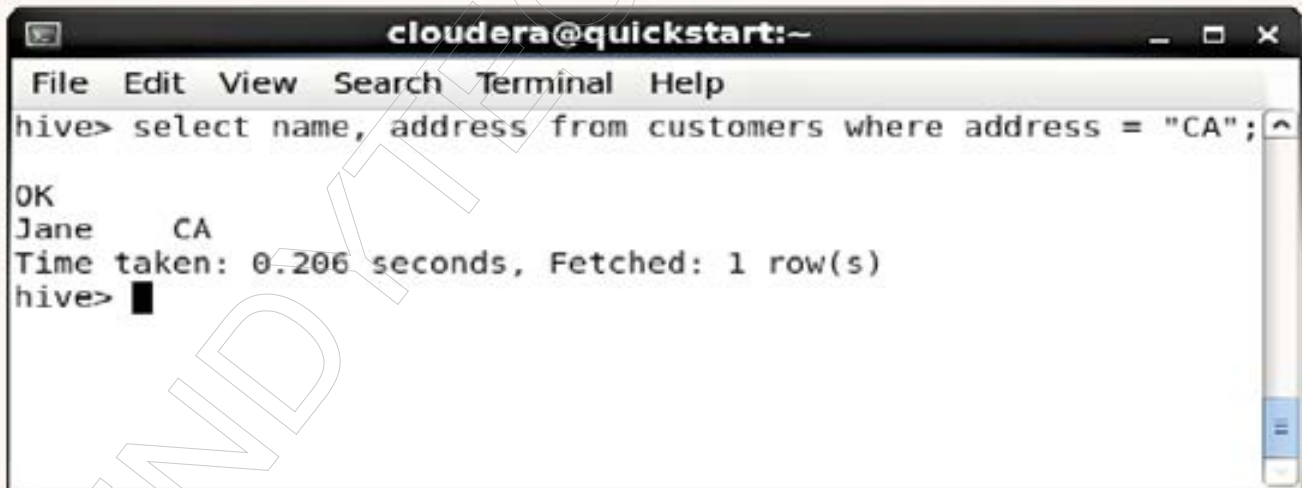
```
select * from customers where address = "WA";
```



A terminal window titled 'cloudera@quickstart:~' showing a Hive query execution. The query is 'select * from customers where address = "WA";'. The output shows three rows of data: (1111, John, WA), (2222, Emily, WA), and (3333, Rick, WA). The time taken is 0.195 seconds and 3 rows were fetched.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive> select * from customers where address = "WA";  
OK  
1111      John      WA  
2222      Emily     WA  
3333      Rick      WA  
Time taken: 0.195 seconds, Fetched: 3 row(s)  
hive>
```

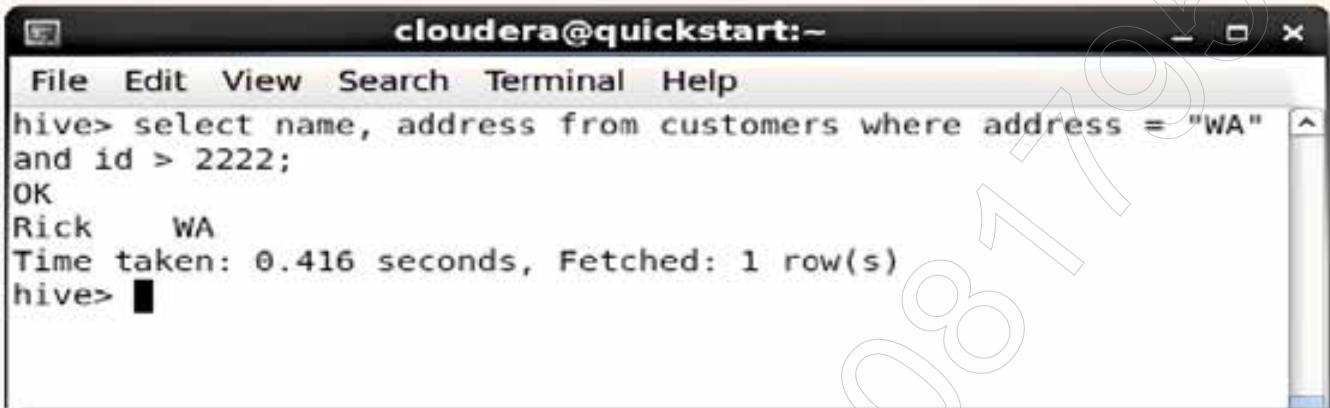
```
select name, address from customers  
where address = "CA";
```



A terminal window titled 'cloudera@quickstart:~' showing a Hive query execution. The query is 'select name, address from customers where address = "CA";'. The output shows one row of data: (Jane, CA). The time taken is 0.206 seconds and 1 row was fetched.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive> select name, address from customers where address = "CA";  
OK  
Jane      CA  
Time taken: 0.206 seconds, Fetched: 1 row(s)  
hive>
```

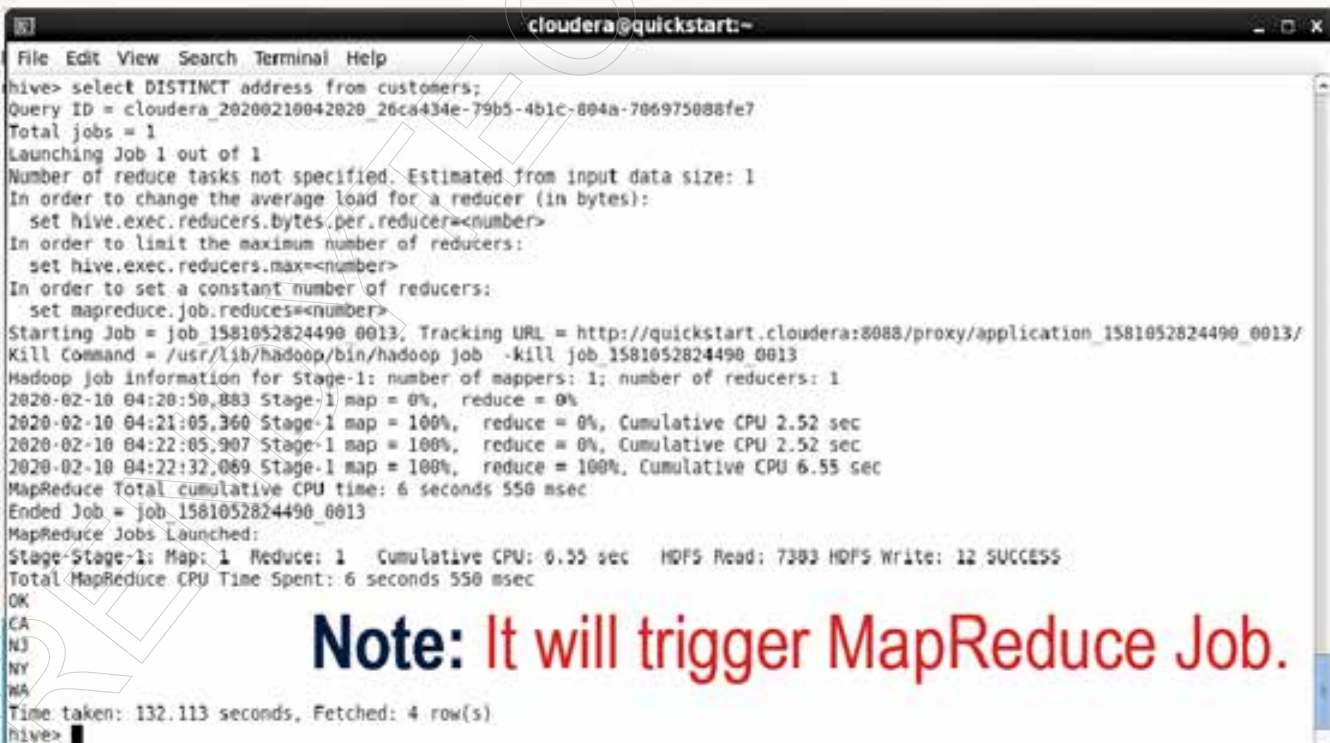
```
select name, address from customers where  
address = "WA" and id > 2222;
```

A terminal window titled 'cloudera@quickstart:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows a Hive prompt 'hive>' followed by the query 'select name, address from customers where address = "WA" and id > 2222;'. The output is 'OK', 'Rick WA', and 'Time taken: 0.416 seconds, Fetched: 1 row(s)'. The prompt 'hive>' is followed by a cursor.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive> select name, address from customers where address = "WA"  
and id > 2222;  
OK  
Rick WA  
Time taken: 0.416 seconds, Fetched: 1 row(s)  
hive> █
```

To display distinct values:

```
select DISTINCT address from customers;
```


A terminal window titled 'cloudera@quickstart:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows a Hive prompt 'hive>' followed by the query 'select DISTINCT address from customers;'. The output is 'OK', 'CA', 'NJ', 'NY', 'WA', and 'Time taken: 132.113 seconds, Fetched: 4 row(s)'. The prompt 'hive>' is followed by a cursor. Below the query, there is a large block of text showing job details, including 'Query ID', 'Total jobs', 'Launching Job 1 out of 1', 'Number of reduce tasks not specified', 'In order to change the average load for a reducer (in bytes):', 'In order to limit the maximum number of reducers:', 'In order to set a constant number of reducers:', 'Starting Job = job 1581052824490_0013', 'Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job 1581052824490_0013', 'Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1', '2020-02-10 04:20:50.883 Stage-1 map = 0%, reduce = 0%', '2020-02-10 04:21:05.360 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.52 sec', '2020-02-10 04:22:05.907 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.52 sec', '2020-02-10 04:22:32.069 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 6.55 sec', 'MapReduce Total cumulative CPU time: 6 seconds 550 msec', 'Ended Job = job 1581052824490_0013', 'MapReduce Jobs Launched:', 'Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 0.55 sec HDFS Read: 7303 HDFS Write: 12 SUCCESS', 'Total MapReduce CPU Time Spent: 6 seconds 550 msec', 'OK', 'CA', 'NJ', 'NY', 'WA', 'Time taken: 132.113 seconds, Fetched: 4 row(s)', 'hive> █'.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive> select DISTINCT address from customers;  
Query ID = cloudera_20200210042020_26ca434e-79b5-4b1c-804a-706975088fe7  
Total jobs = 1  
Launching Job 1 out of 1  
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>  
Starting Job = job 1581052824490_0013, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1581052824490_0013/  
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job 1581052824490_0013  
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1  
2020-02-10 04:20:50.883 Stage-1 map = 0%, reduce = 0%  
2020-02-10 04:21:05.360 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.52 sec  
2020-02-10 04:22:05.907 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.52 sec  
2020-02-10 04:22:32.069 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 6.55 sec  
MapReduce Total cumulative CPU time: 6 seconds 550 msec  
Ended Job = job 1581052824490_0013  
MapReduce Jobs Launched:  
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 0.55 sec HDFS Read: 7303 HDFS Write: 12 SUCCESS  
Total MapReduce CPU Time Spent: 6 seconds 550 msec  
OK  
CA  
NJ  
NY  
WA  
Time taken: 132.113 seconds, Fetched: 4 row(s)  
hive> █
```

Note: It will trigger MapReduce Job.

To display records with order by clause:

```
select name, address from customers
order by address;
```

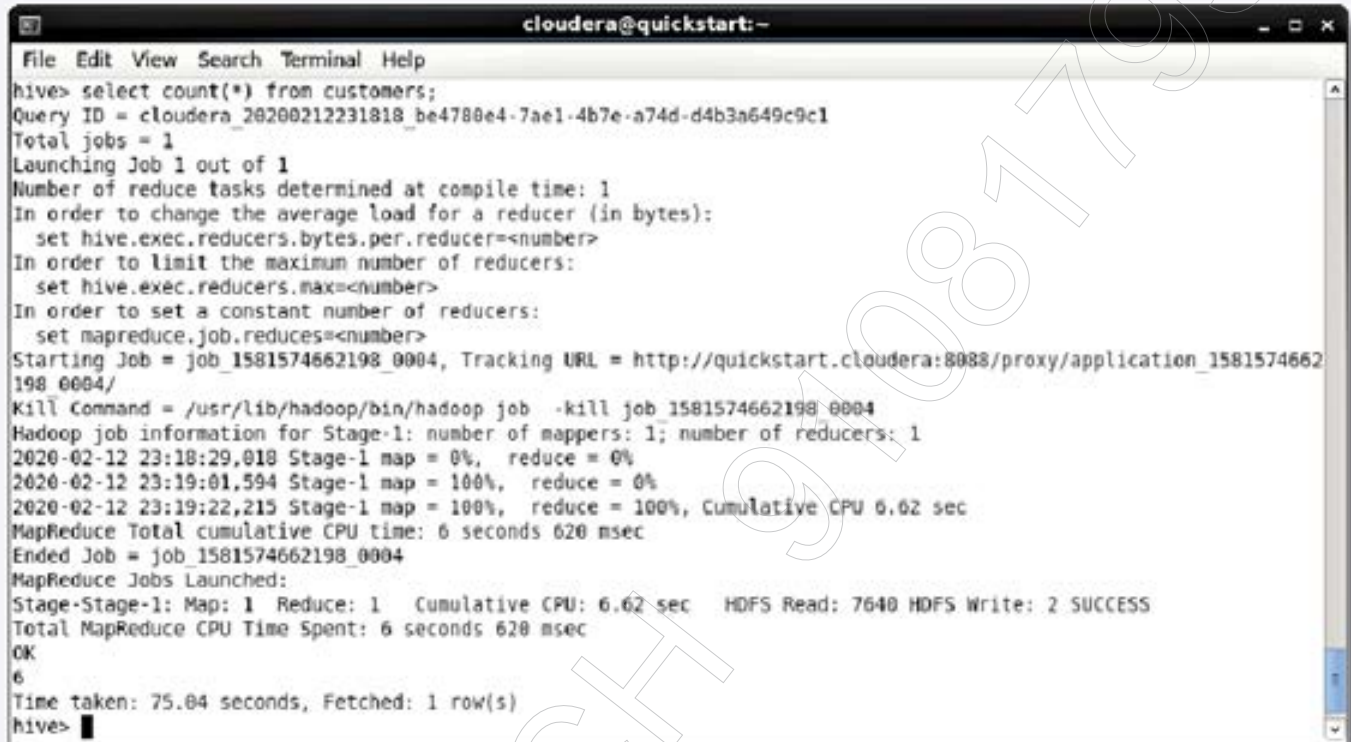


```
cloudera@quickstart:~$
File Edit View Search Terminal Help
hive> select name, address from customers
> order by address;
Query ID = cloudera_20200212231010_849b0d9d-4c8a-4001-970b-b0c9db3ef1ba
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to 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>
Starting Job = job_1581574662198_0002, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1581574662198_0002/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1581574662198_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2020-02-12 23:10:51,030 Stage-1 map = 0%, reduce = 0%
2020-02-12 23:11:12,618 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.71 sec
2020-02-12 23:11:28,520 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 5.96 sec
MapReduce Total cumulative CPU time: 5 seconds 960 msec
Ended Job = job_1581574662198_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 5.96 sec HDFS Read: 6985 HDFS Write: 49 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 960 msec
OK
Jane      CA
Amit      NJ
Nina      NY
Rick      WA
Emily     WA
John      WA
Time taken: 56.901 seconds, Fetched: 6 row(s)
hive>
```

Note: It will trigger MapReduce Job.

To display no. of records in a table:

```
select count(*) from customers;
```



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive> select count(*) from customers;  
Query ID = cloudera_20200212231818_be4780e4-7ae1-4b7e-a74d-d4b3a649c9c1  
Total jobs = 1  
Launching Job 1 out of 1  
Number of reduce tasks determined at compile time: 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>  
Starting Job = job_1581574662198_0004, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1581574662198_0004/  
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1581574662198_0004  
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1  
2020-02-12 23:18:29,010 Stage-1 map = 0%, reduce = 0%  
2020-02-12 23:19:01,594 Stage-1 map = 100%, reduce = 0%  
2020-02-12 23:19:22,215 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 6.62 sec  
MapReduce Total cumulative CPU time: 6 seconds 620 msec  
Ended Job = job_1581574662198_0004  
MapReduce Jobs Launched:  
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 6.62 sec HDFS Read: 7640 HDFS Write: 2 SUCCESS  
Total MapReduce CPU Time Spent: 6 seconds 620 msec  
OK  
6  
Time taken: 75.04 seconds, Fetched: 1 row(s)  
hive>
```

Note: It will trigger MapReduce Job.

To display records with group by clause:

```
select address, count(*) from customers  
group by address;
```

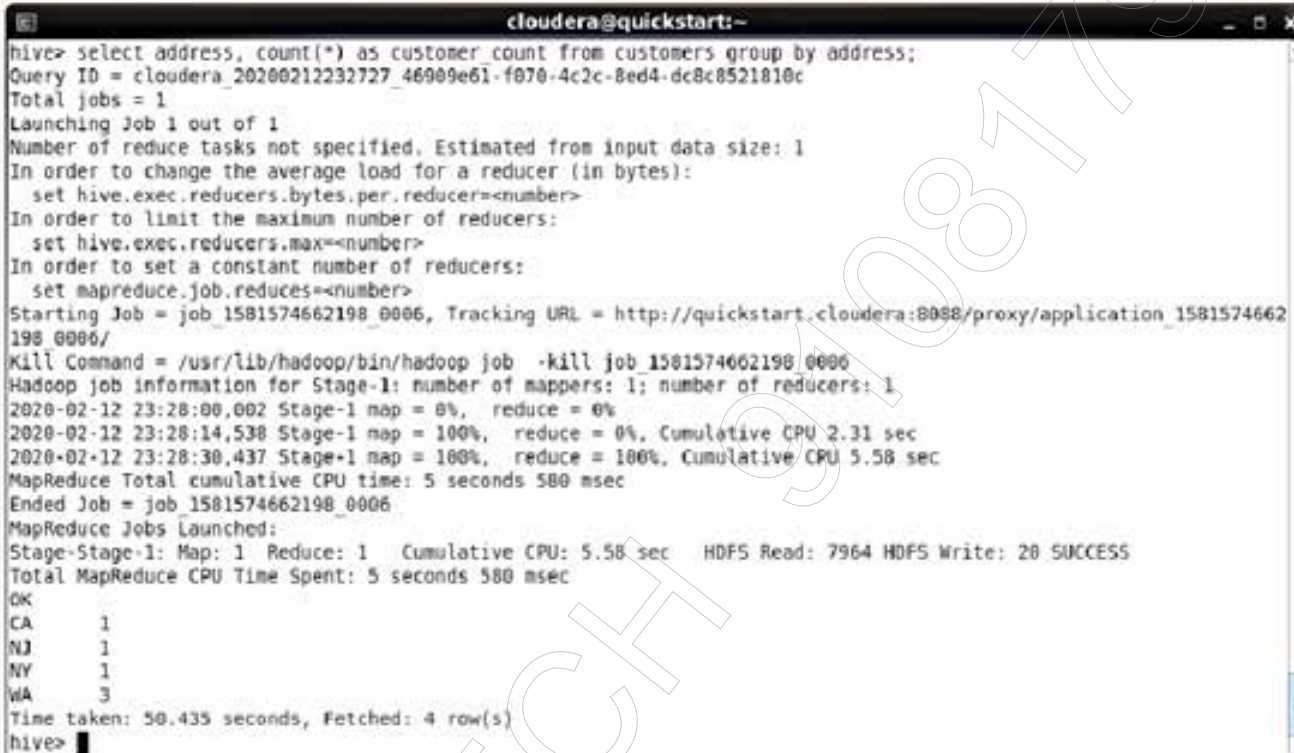


```
cloudera@quickstart:-  
File Edit View Search Terminal Help  
hive> select address, count(*) from customers group by address;  
Query ID = cloudera_20200212232222_f7d69e0e-97c9-4757-9664-7da12b330265  
Total jobs = 1  
Launching Job 1 out of 1  
Number of reduce tasks not specified. Estimated from 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>  
Starting Job = job_1581574062198_0005, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1581574062198_0005/  
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1581574062198_0005  
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1  
2020-02-12 23:22:27,406 Stage-1 map = 0%, reduce = 0%  
2020-02-12 23:22:40,950 Stage-1 map = 100%, reduce = 8%, Cumulative CPU 2.54 sec  
2020-02-12 23:22:57,784 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 5.76 sec  
MapReduce Total cumulative CPU time: 5 seconds 760 msec  
Ended Job = job_1581574062198_0005  
MapReduce Jobs Launched:  
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 5.76 sec HDFS Read: 7964 HDFS Write: 20 SUCCESS  
Total MapReduce CPU Time Spent: 5 seconds 760 msec  
OK  
CA      1  
NJ      1  
NY      1  
WA      3  
Time taken: 50.614 seconds, Fetched: 4 row(s)  
hive>
```

Note: It will trigger MapReduce Job.

To display records with group by clause:

```
select address, count(*) as customer_count
from customers group by address;
```

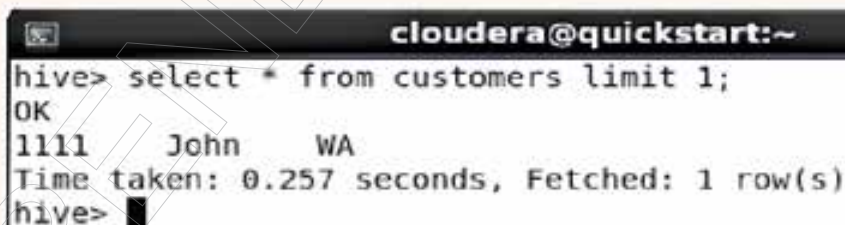


The terminal window shows the execution of a Hive query. The query is: `select address, count(*) as customer_count from customers group by address;`. The output shows the job details, including the number of mappers and reducers, and the final result set.

```
cloudera@quickstart:~
hive> select address, count(*) as customer_count from customers group by address;
Query ID = cloudera_20200212232727_46909e61-f070-4c2c-8ed4-dc8c8521810c
Total jobs = 1
Launching Job 1 out of 1
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>
Starting Job = job_1581574662198_0006, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1581574662198_0006/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1581574662198_0006
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2020-02-12 23:28:00,002 Stage-1 map = 0%, reduce = 0%
2020-02-12 23:28:14,538 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.31 sec
2020-02-12 23:28:30,437 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 5.58 sec
MapReduce Total cumulative CPU time: 5 seconds 580 msec
Ended Job = job_1581574662198_0006
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 5.58 sec HDFS Read: 7964 HDFS Write: 20 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 580 msec
OK
CA      1
NJ      1
NY      1
WA      3
Time taken: 50.435 seconds, Fetched: 4 row(s)
hive>
```

Display records using limit clause:

```
select * from customers limit 1;
```

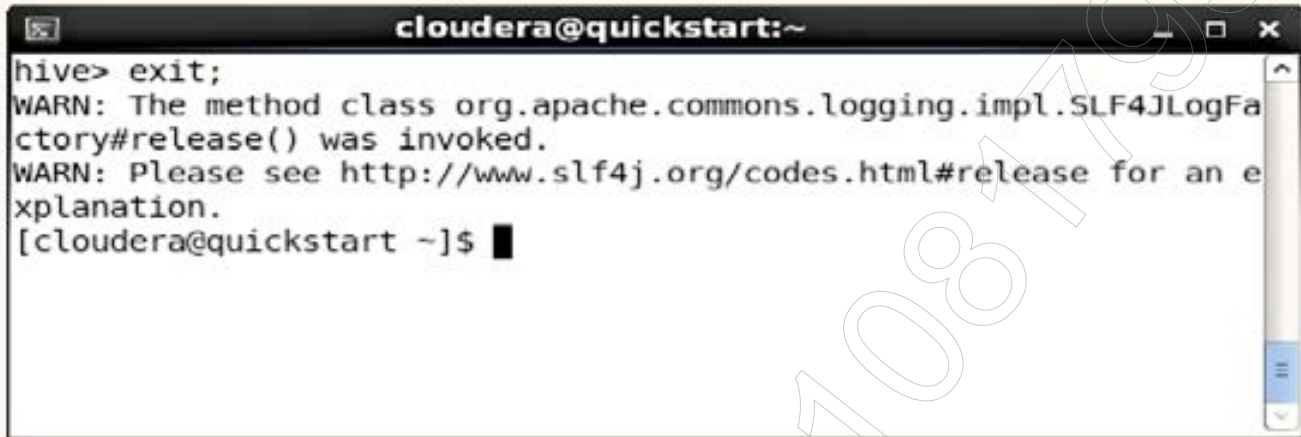


The terminal window shows the execution of a Hive query. The query is: `select * from customers limit 1;`. The output shows the first record from the customers table.

```
cloudera@quickstart:~
hive> select * from customers limit 1;
OK
1111    John    WA
Time taken: 0.257 seconds, Fetched: 1 row(s)
hive>
```


To exit from Hive shell:

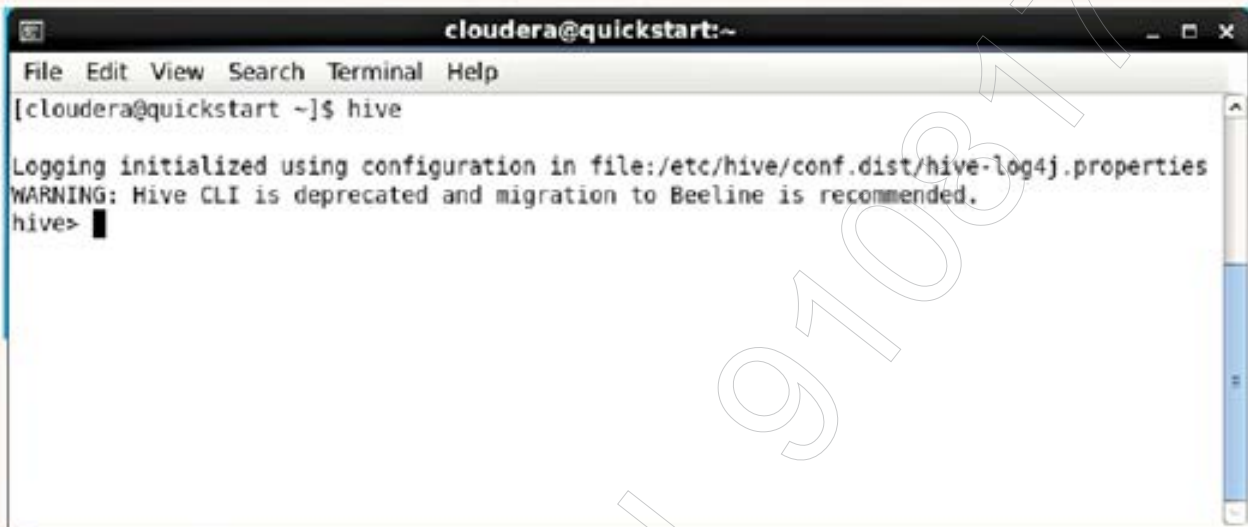
`exit;`

A terminal window titled 'cloudera@quickstart:~' showing the execution of the 'exit;' command. The output includes two warning messages from SLF4J and the shell returning to the prompt '[cloudera@quickstart ~]\$'.

```
cloudera@quickstart:~  
hive> exit;  
WARN: The method class org.apache.commons.logging.impl.SLF4JLogFa  
ctory#release() was invoked.  
WARN: Please see http://www.slf4j.org/codes.html#release for an e  
xplanation.  
[cloudera@quickstart ~]$
```

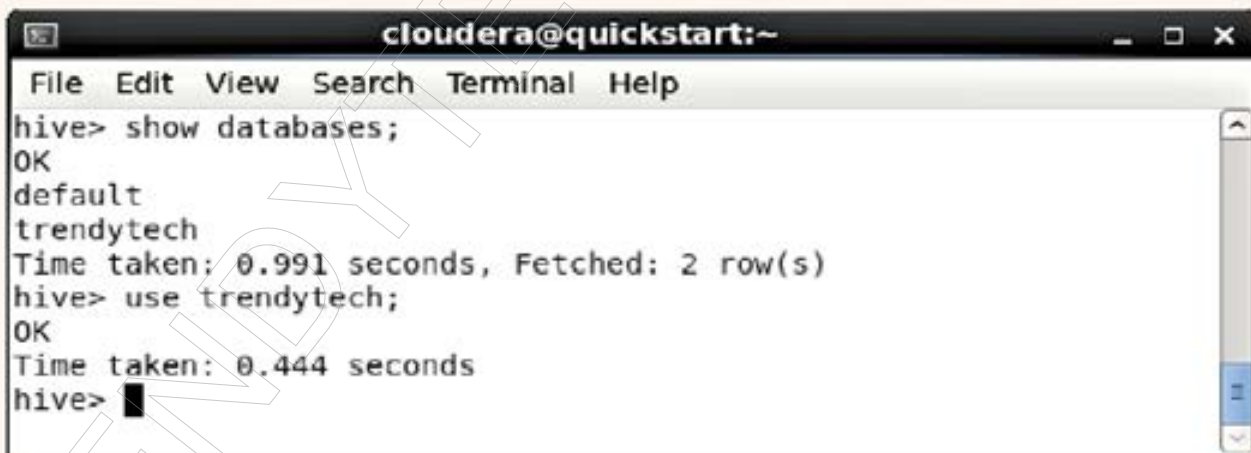
Create a new hive table with **<if not exist>** statement:

- Open hive:



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ hive  
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties  
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.  
hive>
```

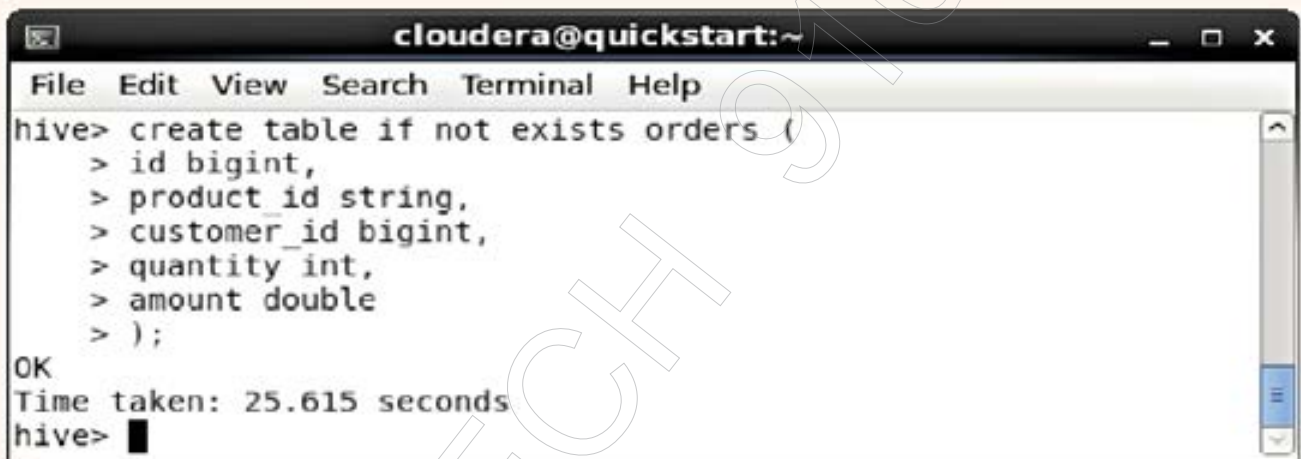
- use database trendytech;



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive> show databases;  
OK  
default  
trendytech  
Time taken: 0.991 seconds, Fetched: 2 row(s)  
hive> use trendytech;  
OK  
Time taken: 0.444 seconds  
hive>
```

Create a table with **<if not exist>** statement:

```
create table if not exists orders(  
id bigint,  
product_id string,  
customer_id bigint,  
quantity int,  
amount double  
);
```



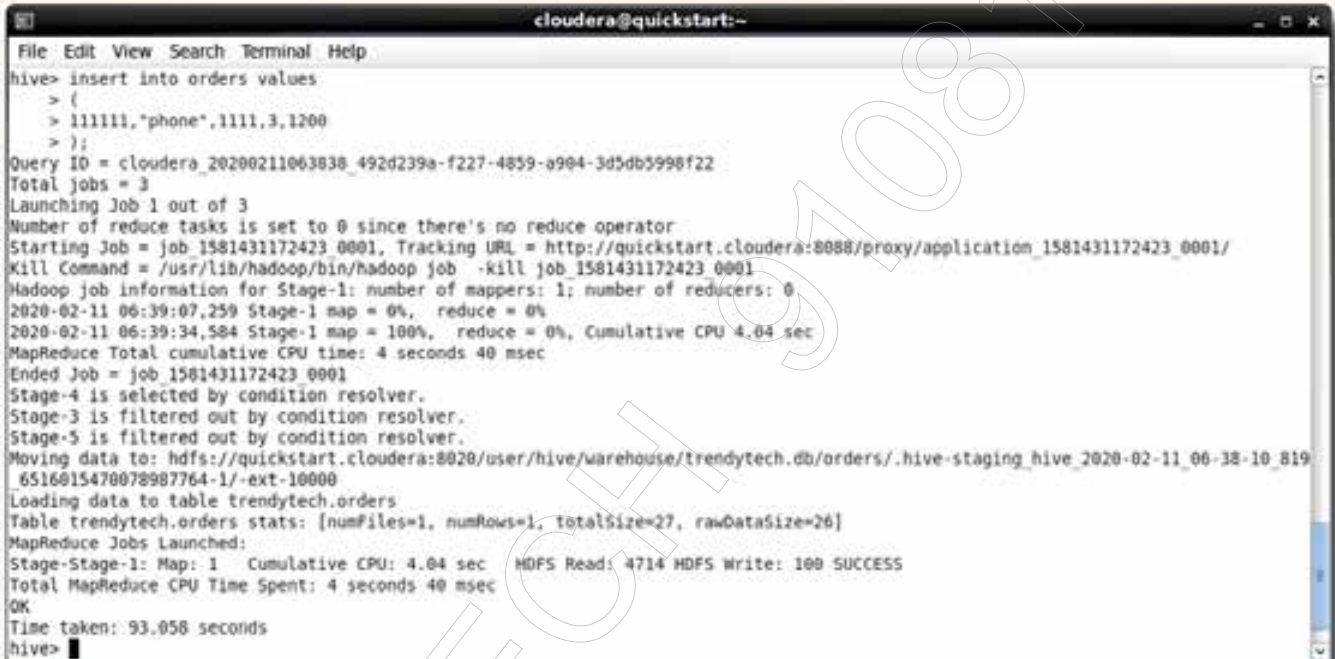
The screenshot shows a terminal window titled "cloudera@quickstart:~". The terminal has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The command prompt is "hive>". The user enters the command: "create table if not exists orders (
> id bigint,
> product_id string,
> customer_id bigint,
> quantity int,
> amount double
>);". The terminal shows "OK" and "Time taken: 25.615 seconds". The prompt returns to "hive>".

Note: If the table with same name already exists, the above statement won't do anything.

If the table does not exist, then it will create a new table.

Insert a record into *orders* table:

```
insert into orders values  
(  
111111,"phone",1111,3,1200  
);
```



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive> insert into orders values  
> (  
> 111111,"phone",1111,3,1200  
> );  
Query ID = cloudera_20200211063838_492d239a-f227-4859-a904-3d5db5998f22  
Total jobs = 3  
Launching Job 1 out of 3  
Number of reduce tasks is set to 0 since there's no reduce operator  
Starting Job = job_1581431172423_0001, Tracking URL = http://quickstart.cloudera:8080/proxy/application_1581431172423_0001/  
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1581431172423_0001  
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0  
2020-02-11 06:39:07,259 Stage-1 map = 0%, reduce = 0%  
2020-02-11 06:39:34,584 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 4.04 sec  
MapReduce Total cumulative CPU time: 4 seconds 40 msec  
Ended Job = job_1581431172423_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: hdfs://quickstart.cloudera:8020/user/hive/warehouse/trendytech.db/orders/.hive-staging_hive_2020-02-11_06-38-10_819_6516015470078987764-1/-ext-10000  
Loading data to table trendytech.orders  
Table trendytech.orders stats: [numFiles=1, numRows=1, totalSize=27, rawDataSize=26]  
MapReduce Jobs Launched:  
Stage-Stage-1: Map: 1 Cumulative CPU: 4.04 sec HDFS Read: 4714 HDFS Write: 100 SUCCESS  
Total MapReduce CPU Time Spent: 4 seconds 40 msec  
OK  
Time taken: 93.058 seconds  
hive>
```

Note: It will trigger MapReduce job.

Insert multiple records into orders table

```
insert into orders values
(
111112,"camera",1111,1,5200), (
111113,"broom",1111,1,10), (
111114,"broom",2222,2,20), (
111115,"t-shirt",4444,2,66
);
```



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive> insert into orders values  
> {  
> 111112,"camera",1111,1,5200},{  
> 111113,"broom",1111,1,10},{  
> 111114,"broom",2222,2,20},{  
> 111115,"t-shirt",4444,2,66  
> };  
Query ID = cloudera_20200211064242_a3995ce4-e3a6-433c-99e5-9ccb78e7c62c  
Total jobs = 3  
Launching Job 1 out of 3  
Number of reduce tasks is set to 0 since there's no reduce operator  
Starting Job = job_1581431172423_0002, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1581431172423_0002/  
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1581431172423_0002  
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0  
2020-02-11 06:44:22,873 Stage-1 map = 0%, reduce = 0%  
2020-02-11 06:44:44,156 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 3.76 sec  
MapReduce Total cumulative CPU time: 3 seconds 760 msec  
Ended Job = job_1581431172423_0002  
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: hdfs://quickstart.cloudera:8020/user/hive/warehouse/trendytech.db/orders/.hive-staging_hive_2020-02-11_06-42-01_321_3671192994152449539-1/-ext-10000  
Loading data to table trendytech.orders  
Table trendytech.orders stats: [numFiles=2, numRows=5, totalSize=132, rawDataSize=127]  
MapReduce Jobs Launched:  
Stage-Stage-1: Map: 1 Cumulative CPU: 3.76 sec HDFS Read: 4902 HDFS Write: 179 SUCCESS  
Total MapReduce CPU Time Spent: 3 seconds 760 msec  
OK  
Time taken: 169.354 seconds
```

Note: It will trigger MapReduce job.

Working with

HUE

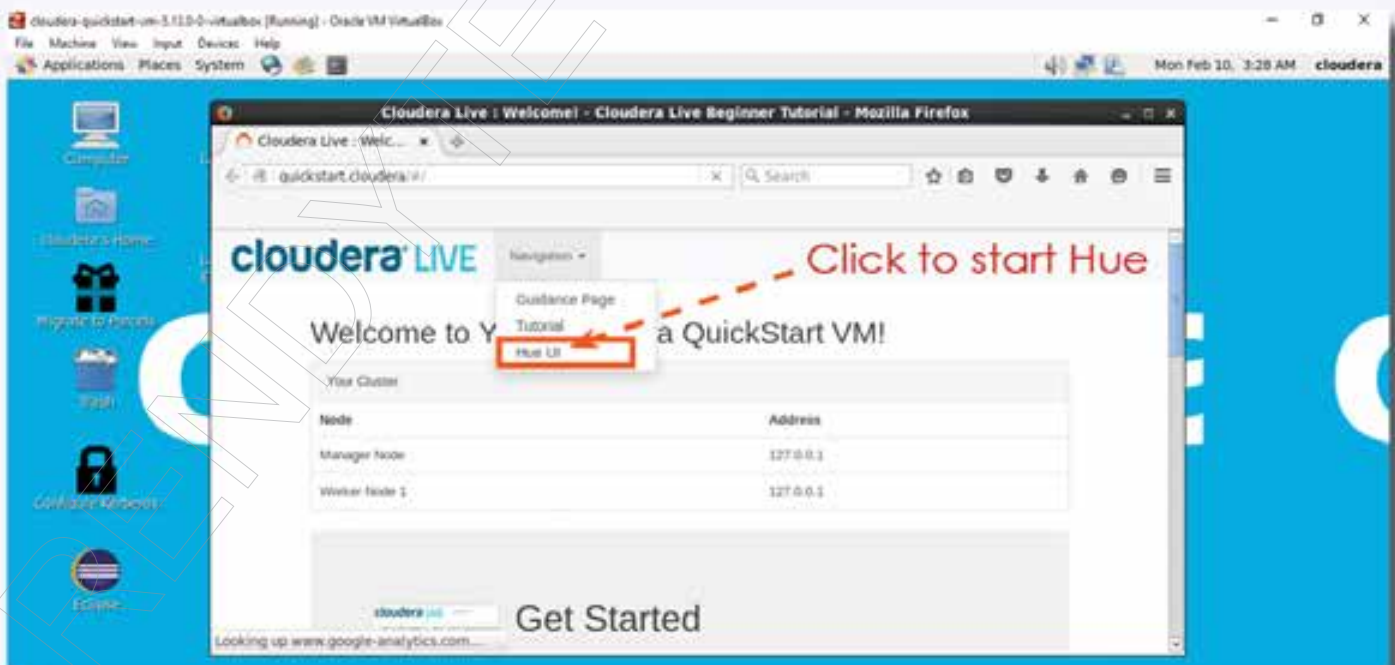


Invoking Hue:

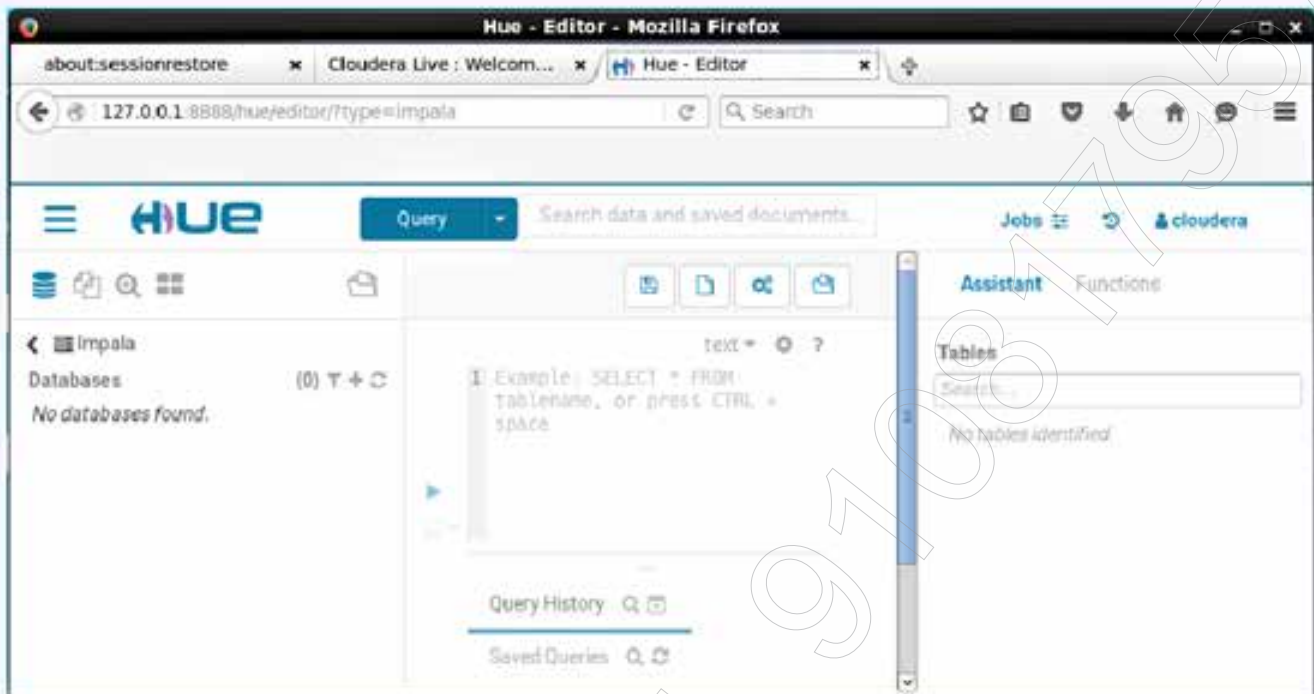
- Open cloudera browser



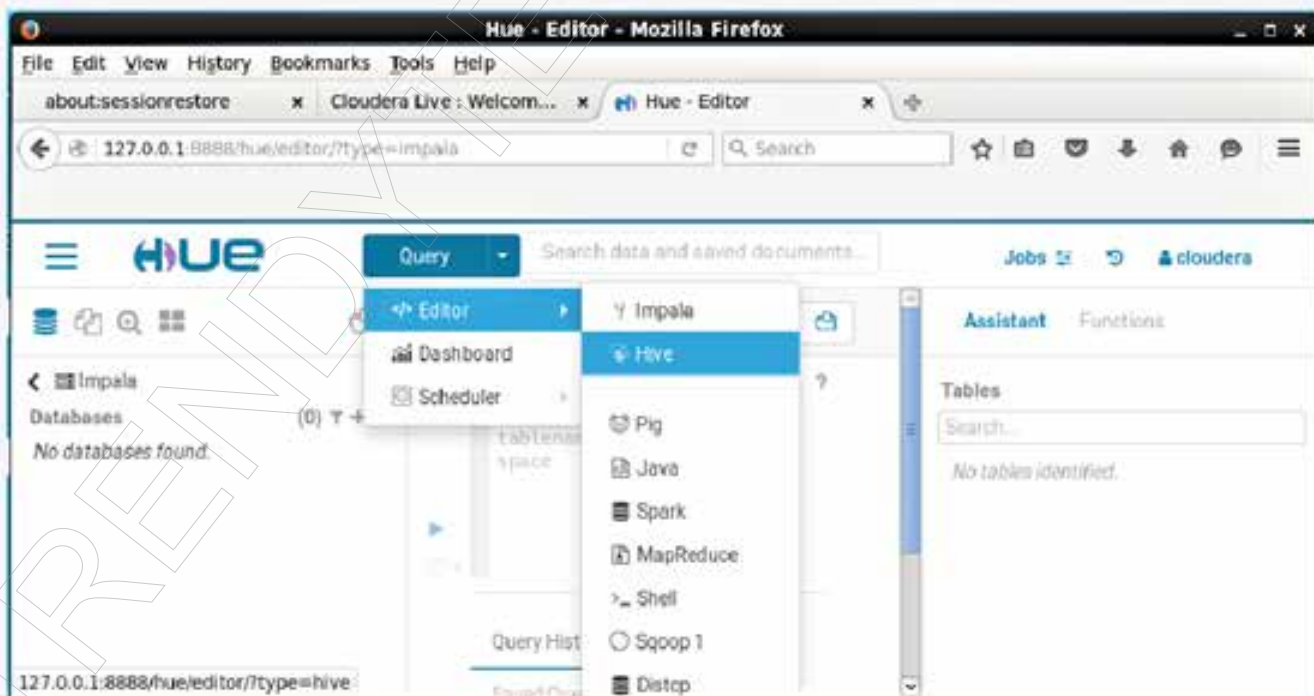
- In the navigation drop-down menu choose **Hue UI**



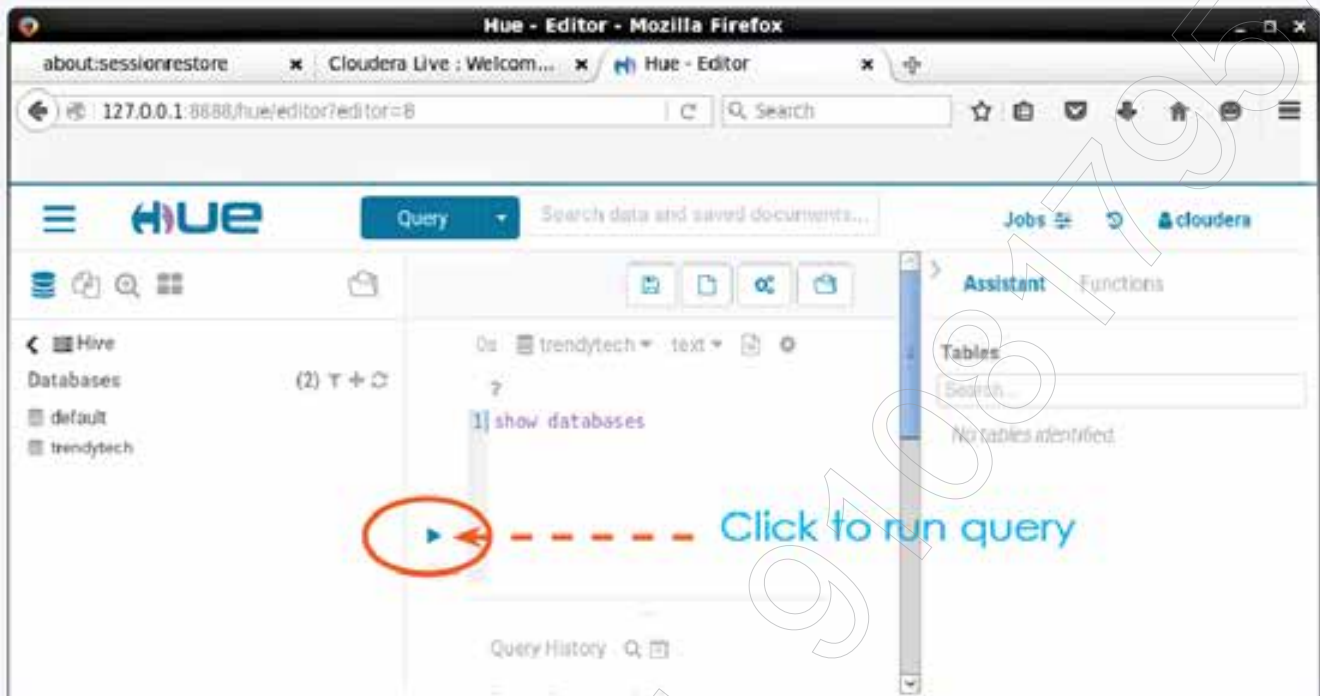
The Hue web UI:



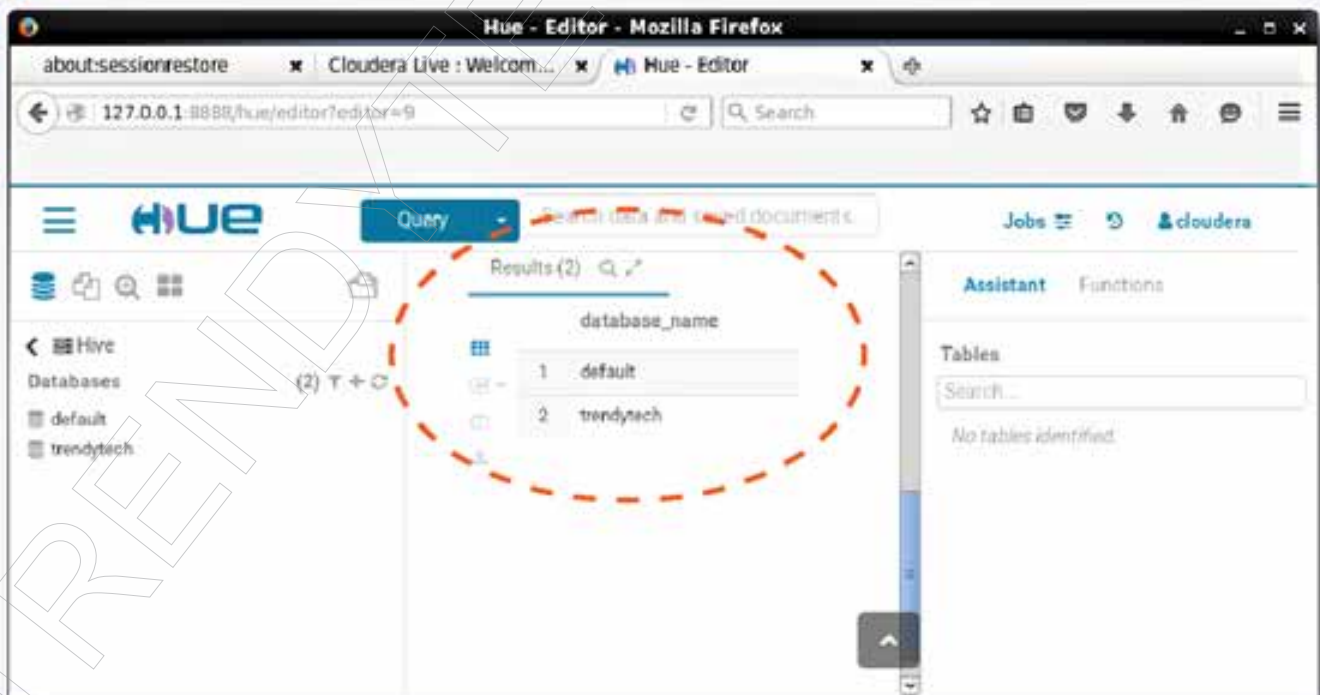
Select Hive from the Query menu:



Type Hive command on query area and run:



The result will be displayed in the result area:

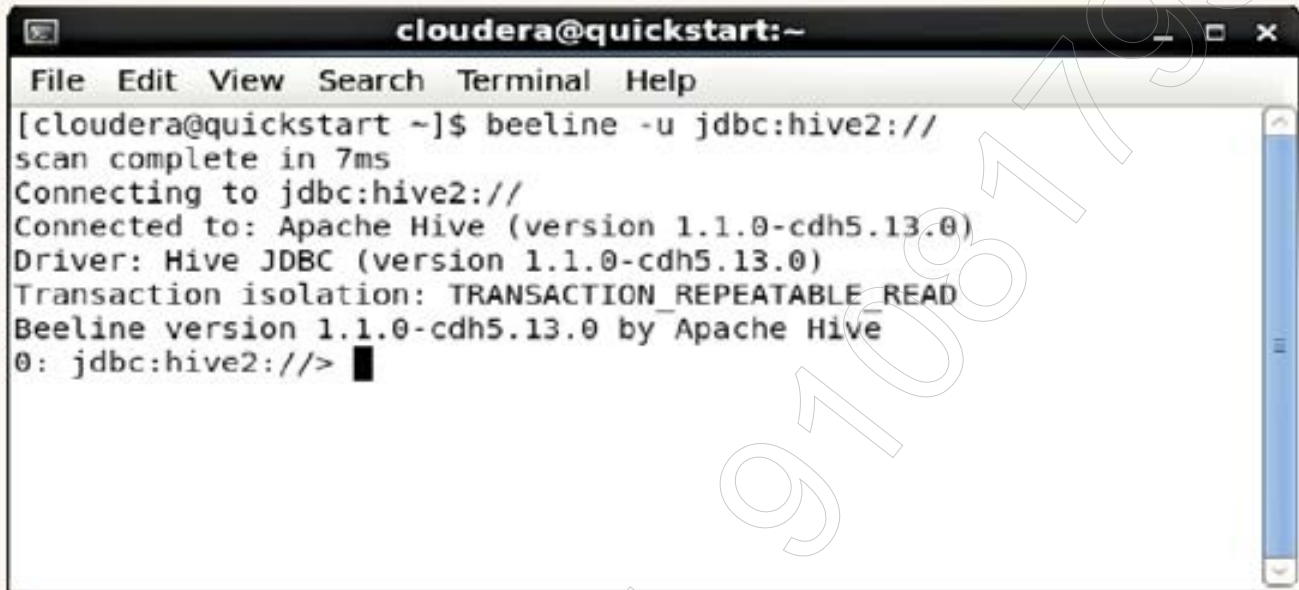


Connecting Hive with Beeline



To enter Beeline:

```
beeline -u jdbc:hive2://
```

A terminal window titled 'cloudera@quickstart:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the execution of 'beeline -u jdbc:hive2://'. The output includes: 'scan complete in 7ms', 'Connecting to jdbc:hive2://', 'Connected to: Apache Hive (version 1.1.0-cdh5.13.0)', 'Driver: Hive JDBC (version 1.1.0-cdh5.13.0)', 'Transaction isolation: TRANSACTION_REPEATABLE_READ', 'Beeline version 1.1.0-cdh5.13.0 by Apache Hive', and the prompt '0: jdbc:hive2://>'.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ beeline -u jdbc:hive2://  
scan complete in 7ms  
Connecting to jdbc:hive2://  
Connected to: Apache Hive (version 1.1.0-cdh5.13.0)  
Driver: Hive JDBC (version 1.1.0-cdh5.13.0)  
Transaction isolation: TRANSACTION_REPEATABLE_READ  
Beeline version 1.1.0-cdh5.13.0 by Apache Hive  
0: jdbc:hive2://>
```

Run hive commands from Beeline:

```
show databases;
```

A terminal window titled 'cloudera@quickstart:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the execution of 'show databases;' from the Beeline prompt. The output is a table with two rows: 'default' and 'trendytech'. Below the table, it says '2 rows selected (0.223 seconds)'.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
0: jdbc:hive2://> show databases;  
OK  
+-----+  
| database name |  
+-----+  
| default      |  
| trendytech   |  
+-----+  
2 rows selected (0.223 seconds)  
0: jdbc:hive2://>
```

Run few more hive commands from Beeline:

describe customers;

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
0: jdbc:hive2://> describe customers;  
OK  
+-----+-----+-----+  
| col_name | data_type | comment |  
+-----+-----+-----+  
| id       | bigint   |         |  
| name     | string   |         |  
| address  | string   |         |  
+-----+-----+-----+  
3 rows selected (0.878 seconds)  
0: jdbc:hive2://> █
```

describe **formatted** customers;

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
0: jdbc:hive2://> describe formatted customers;  
OK  
20/02/06 02:36:12 [main]: WARN lazy.LazyStruct: Extra bytes detected at the end of the row! Ignoring similar problems.  
+-----+-----+-----+  
| col_name | data_type | comment |  
+-----+-----+-----+  
# col_name | data_type | comment |  
id         | bigint   |         |  
name       | string   |         |  
address    | string   |         |  
# Detailed Table Information | NULL | NULL |  
Database:  | trendytech | NULL |  
Owner:     | cloudera   | NULL |  
CreateTime: | Mon Feb 03 20:15:37 PST 2020 | NULL |  
LastAccessTime: | UNKNOWN   | NULL |  
Protect Mode: | None      | NULL |  
Retention:  | 0         | NULL |  
Location:   | hdfs://quickstart.cloudera:8020/user/hive/warehouse/trendytech.db/customers | NULL |  
Table Type: | MANAGED_TABLE | NULL |  
Table Parameters: | NULL | NULL |  
COLUMN_STATS_ACCURATE | true |  
numFiles | 2 |  
numRows  | 6 |  
rawDataSize | 73 |  
totalSize  | 79 |  
transient_lastDdlTime | 1580790920 |  
# Storage Information | NULL | NULL |  
SerDe Library: | org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe | NULL |
```


To exit from Beeline:

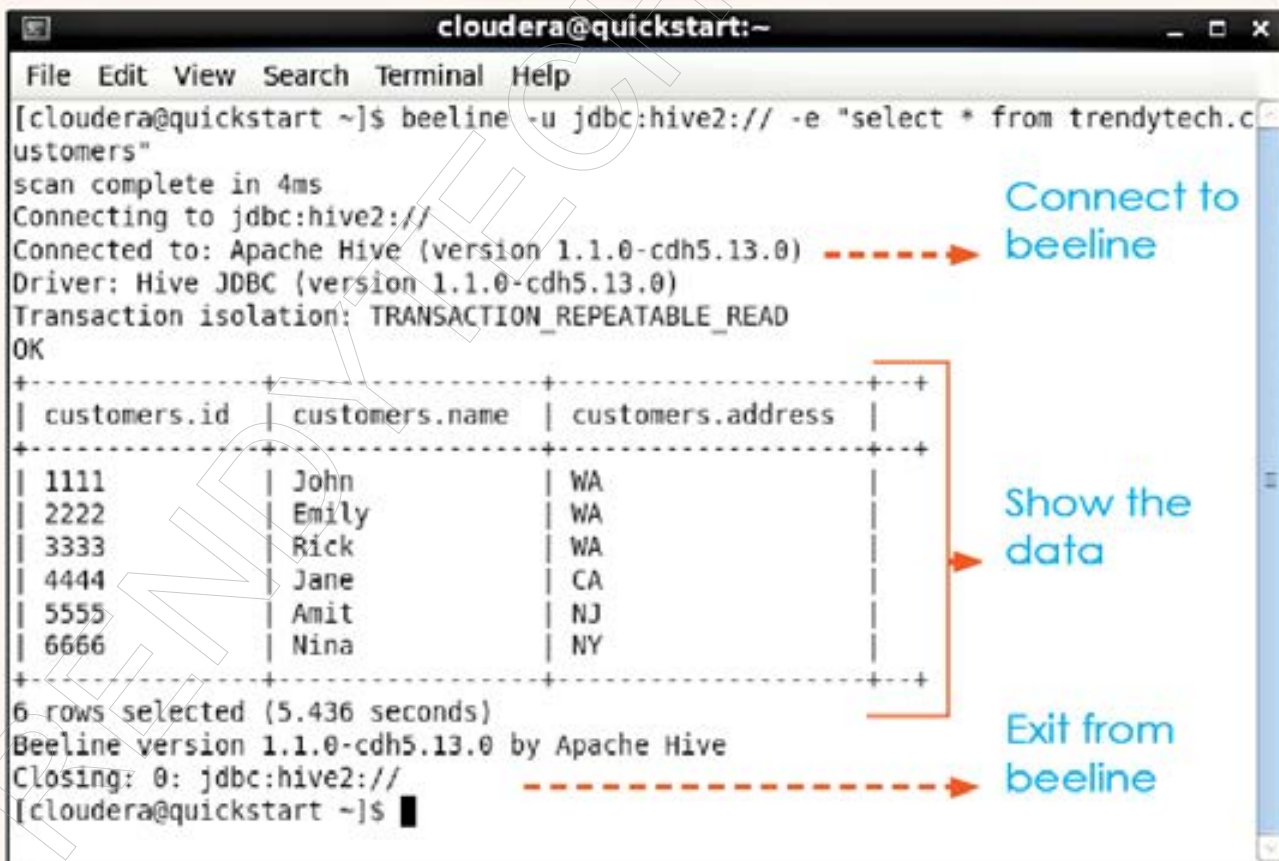
!q



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
0: jdbc:hive2://> !q  
Closing: 0: jdbc:hive2://  
[cloudera@quickstart ~]$
```

Execute beeline query from terminal:

```
beeline -u jdbc:hive2:// -e "select * from trendytech.customers"
```



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ beeline -u jdbc:hive2:// -e "select * from trendytech.c  
ustomers"  
scan complete in 4ms  
Connecting to jdbc:hive2://  
Connected to: Apache Hive (version 1.1.0-cdh5.13.0)  
Driver: Hive JDBC (version 1.1.0-cdh5.13.0)  
Transaction isolation: TRANSACTION_REPEATABLE_READ  
OK  
+-----+-----+-----+  
| customers.id | customers.name | customers.address |  
+-----+-----+-----+  
| 1111         | John          | WA                |  
| 2222         | Emily         | WA                |  
| 3333         | Rick          | WA                |  
| 4444         | Jane          | CA                |  
| 5555         | Amit          | NJ                |  
| 6666         | Nina          | NY                |  
+-----+-----+-----+  
6 rows selected (5.436 seconds)  
Beeline version 1.1.0-cdh5.13.0 by Apache Hive  
Closing: 0: jdbc:hive2://  
[cloudera@quickstart ~]$
```

Connect to beeline

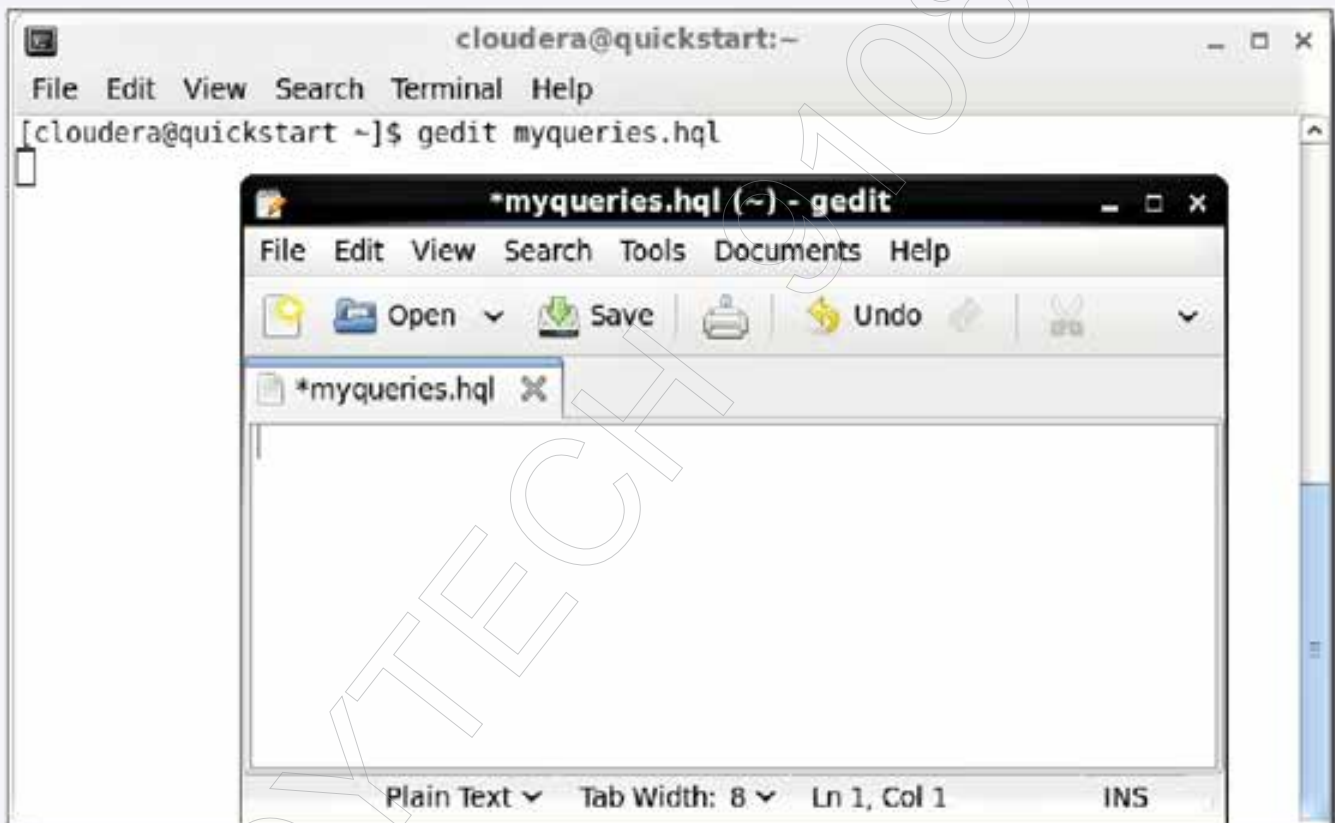
Show the data

Exit from beeline

Run Beeline script file from terminal

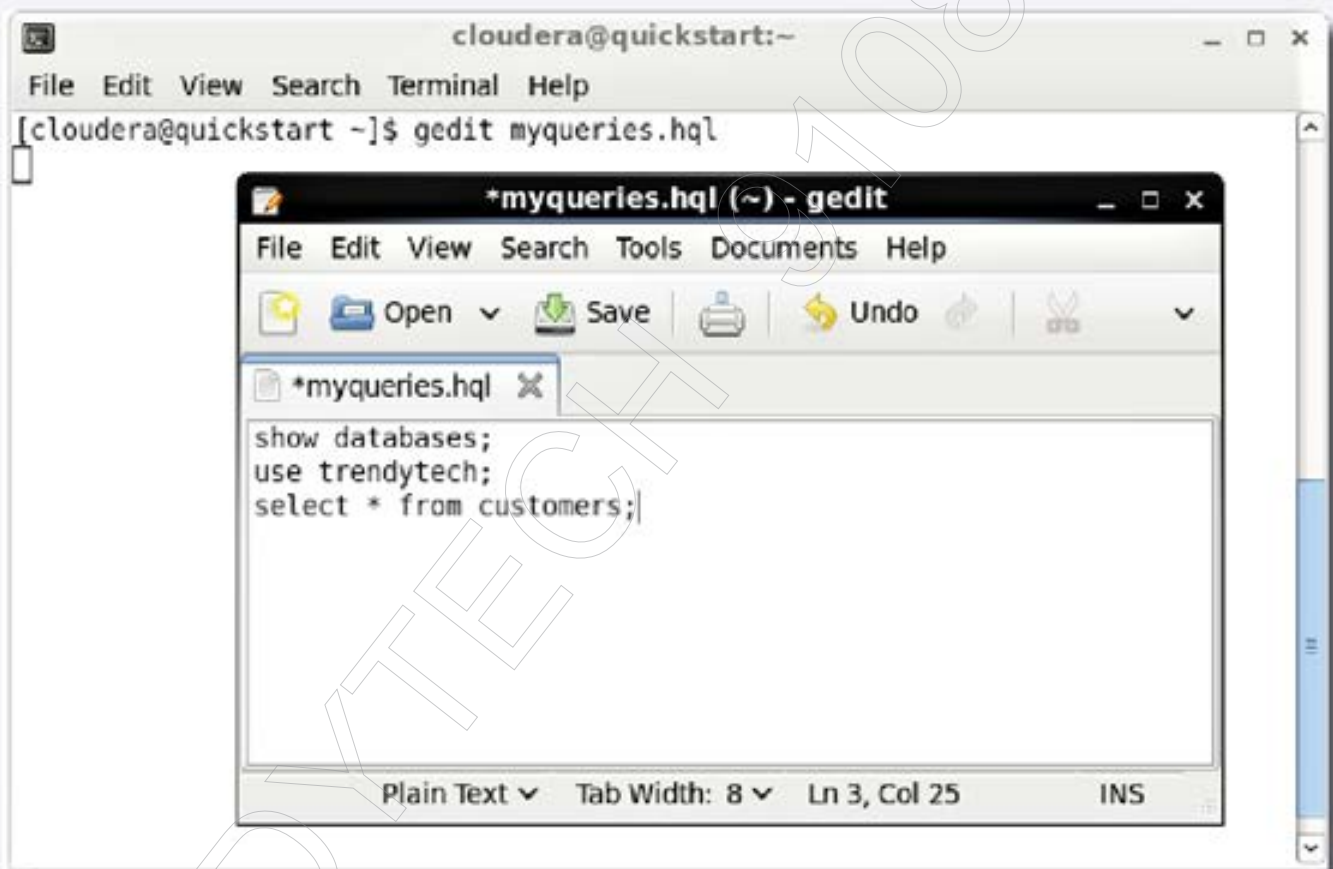
Create a file *myqueries.hql* using gedit from terminal:

```
gedit test.hql
```



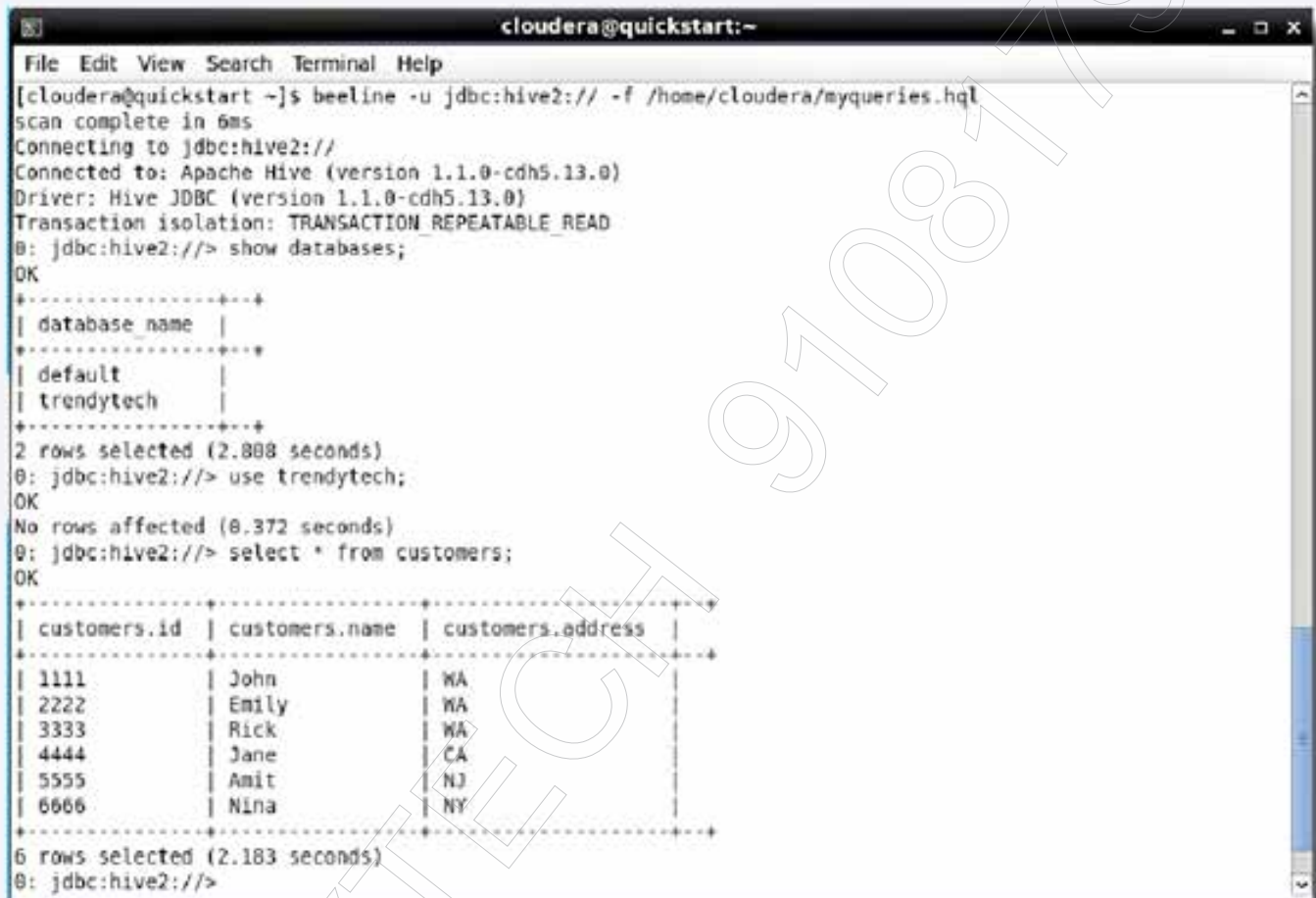
Enter following beeline queries inside *myqueries.hql* and save:

```
show databases;  
use trendytech;  
select * from customers;
```



Execute the beeline script file from terminal:

```
beeline -u jdbc:hive2:// -f /home/cloudera/myqueries.hql
```



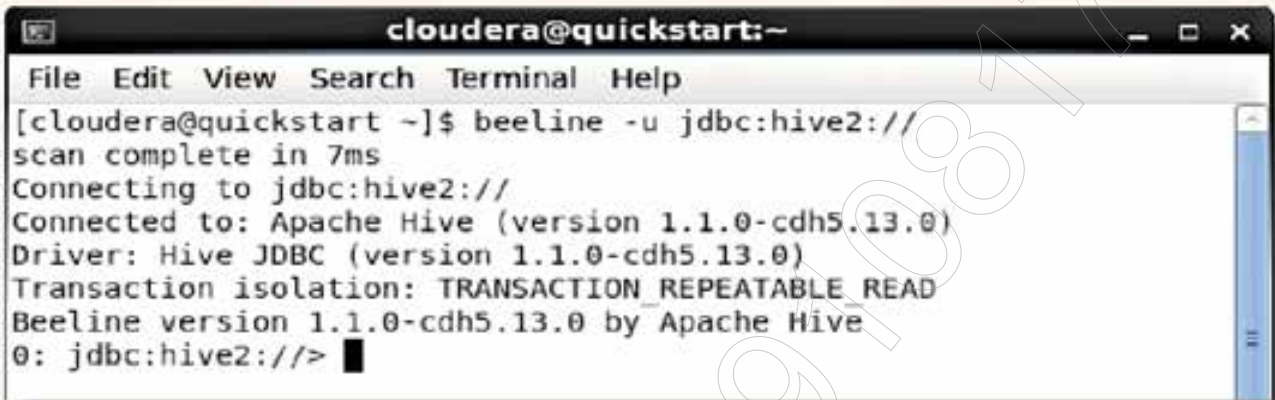
A terminal window titled 'cloudera@quickstart:~' showing the execution of a beeline script. The script connects to a Hive instance and runs two queries. The first query, 'show databases;', returns two rows: 'default' and 'trendytech'. The second query, 'select * from customers;', returns six rows of customer data.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ beeline -u jdbc:hive2:// -f /home/cloudera/myqueries.hql  
scan complete in 6ms  
Connecting to jdbc:hive2://  
Connected to: Apache Hive (version 1.1.0-cdh5.13.0)  
Driver: Hive JDBC (version 1.1.0-cdh5.13.0)  
Transaction isolation: TRANSACTION_REPEATABLE_READ  
0: jdbc:hive2://> show databases;  
OK  
+-----+  
| database name |  
+-----+  
| default       |  
| trendytech    |  
+-----+  
2 rows selected (2.808 seconds)  
0: jdbc:hive2://> use trendytech;  
OK  
No rows affected (0.372 seconds)  
0: jdbc:hive2://> select * from customers;  
OK  
+-----+-----+-----+  
| customers.id | customers.name | customers.address |  
+-----+-----+-----+  
| 1111         | John          | WA                |  
| 2222         | Emily         | WA                |  
| 3333         | Rick          | WA                |  
| 4444         | Jane          | CA                |  
| 5555         | Amit          | NJ                |  
| 6666         | Nina          | NY                |  
+-----+-----+-----+  
6 rows selected (2.183 seconds)  
0: jdbc:hive2://>
```

Run beeline script file from beeline itself

Enter into beeline:

```
beeline -u jdbc:hive2://
```



A terminal window titled 'cloudera@quickstart:~' showing the execution of the 'beeline -u jdbc:hive2://' command. The output shows a successful connection to Apache Hive (version 1.1.0-cdh5.13.0) using the Hive JDBC driver (version 1.1.0-cdh5.13.0). The transaction isolation is set to TRANSACTION_REPEATABLE_READ. The prompt changes to '0: jdbc:hive2://>'.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ beeline -u jdbc:hive2://  
scan complete in 7ms  
Connecting to jdbc:hive2://  
Connected to: Apache Hive (version 1.1.0-cdh5.13.0)  
Driver: Hive JDBC (version 1.1.0-cdh5.13.0)  
Transaction isolation: TRANSACTION_REPEATABLE_READ  
Beeline version 1.1.0-cdh5.13.0 by Apache Hive  
0: jdbc:hive2://>
```

Execute the beeline script:

```
source /home/cloudera/myqueries.hql
```



A terminal window titled 'cloudera@quickstart:~' showing the execution of the 'source /home/cloudera/myqueries.hql;' command. The output shows two queries being executed. The first query selects from a table named 'default.trendytech', returning 2 rows. The second query selects from a table named 'customers', returning 6 rows. The results are displayed in a table format.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
0: jdbc:hive2://> source /home/cloudera/myqueries.hql;  
OK  
+-----+  
| database_name |  
+-----+  
| default      |  
| trendytech   |  
+-----+  
2 rows selected (0.145 seconds)  
OK  
No rows affected (0.085 seconds)  
OK  
+-----+-----+-----+  
| customers.id | customers.name | customers.address |  
+-----+-----+-----+  
| 1111         | John          | WA                |  
| 2222         | Emily         | WA                |  
| 3333         | Rick          | WA                |  
| 4444         | Jane          | CA                |  
| 5555         | Amit          | NJ                |  
| 6666         | Nina          | NY                |  
+-----+-----+-----+  
6 rows selected (0.34 seconds)  
0: jdbc:hive2://>
```




5 Star Google Rated
Big Data Course

LEARN FROM THE EXPERT



9108179578

Call for more details

Follow US

Trainer Mr. Sumit Mittal

LinkedIn <https://www.linkedin.com/in/bigdatabysumit/>

Website <https://trendytech.in/courses/big-data-online-training/>

Phone 9108179578

Email trendytech.sumit@gmail.com

Youtube TrendyTech

Twitter @BigdataBySumit

Instagram bigdatabysumit

Facebook <https://www.facebook.com/trendytech.in/>

