



Apache Hive

Part 3

by Sumit Mittal



TRENDY TECH

UPLIFT YOUR CAREER!

Apache Hive

Exercise 3



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.

Complex data types in Hive

Hive has support for 4 types of Complex Data:

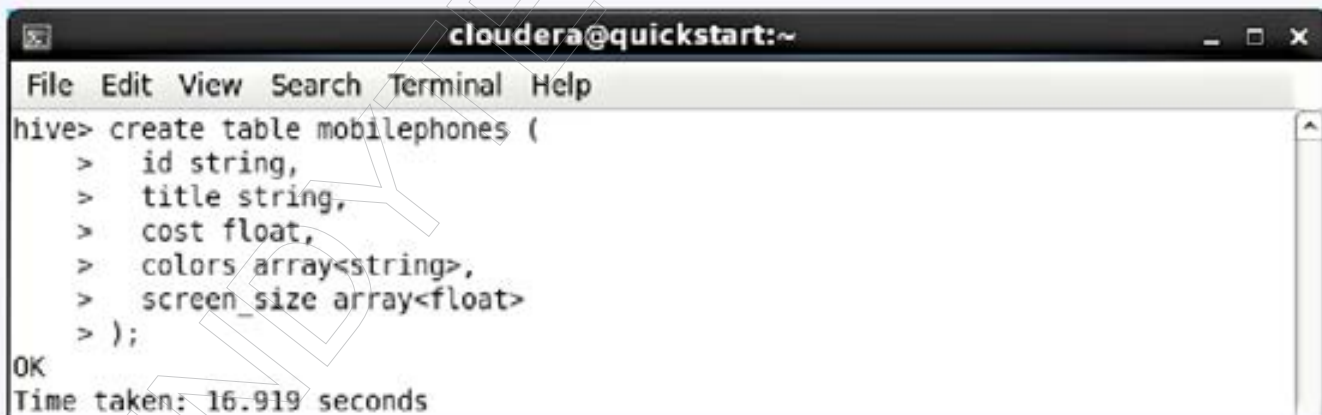
1. Array
2. Map
3. Struct
4. Union (Rarely used because of incomplete support in Hive)

Arrays

Collection of items of similar data type. An array can contain one or more values of the same data type.

Create a table with array data types:

```
create table mobilephones (  
  id string,  
  title string,  
  cost float,  
  colors array<string>,  
  screen_size array<float>  
);
```



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 being executed is:

```
hive> create table mobilephones (  
>   id string,  
>   title string,  
>   cost float,  
>   colors array<string>,  
>   screen_size array<float>  
> );
```

The output of the command is:

```
OK  
Time taken: 16.919 seconds
```


Load data into the array data type table:

```
insert into table mobilephones  
select "redminote7", "Redmi Note 7", 300,  
array("white", "silver", "black"),  
array(float(4.5))
```

UNION ALL

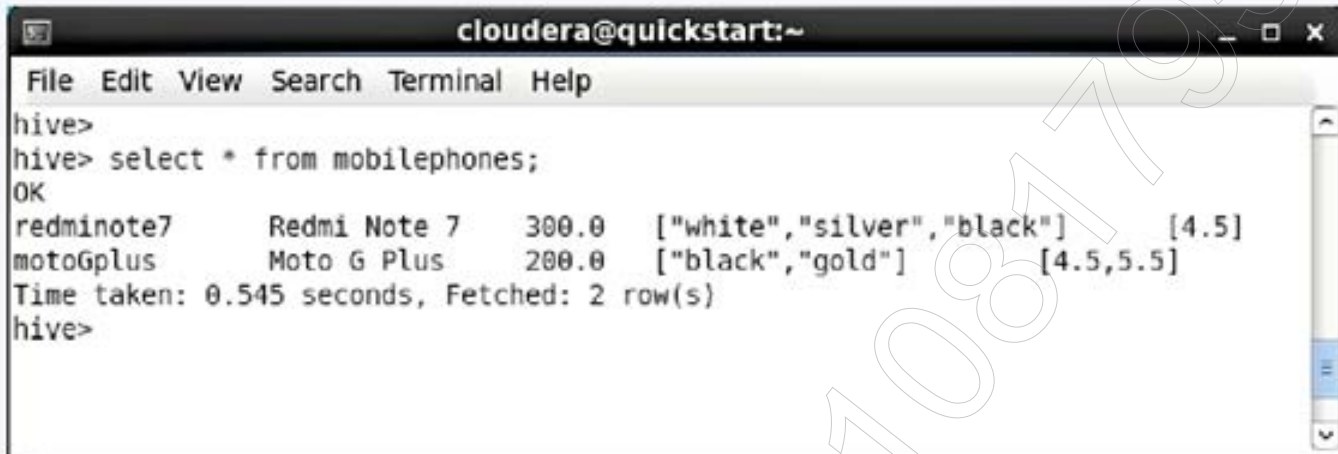
```
select "motoGplus", "Moto G Plus", 200,  
array("black", "gold"),  
array(float(4.5), float(5.5));
```



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive> insert into table mobilephones  
> select "redminote7", "Redmi Note 7", 300,  
> array("white", "silver", "black"), array(float(4.5))  
> UNION ALL  
> select "motoGplus", "Moto G Plus", 200, array("black", "gold"),  
> array(float(4.5), float(5.5));  
Query ID = cloudera_20200414045151_2f80b8ab-4c04-48b8-8a4c-18d018a44950  
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_1586509874827_0002, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1586509874827_0002/  
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1586509874827_0002  
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0  
2020-04-14 04:51:10,400 Stage-1 map = 0%, reduce = 0%  
2020-04-14 04:51:32,997 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 3.88 sec  
MapReduce Total cumulative CPU time: 3 seconds 888 msec  
Ended Job = job_1586509874827_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/mobilephones/.hive-staging_hive_2020-04-14_04-51-0  
2_370_3876187256621881706-1/-ext-10000  
Loading data to table default.mobilephones  
Table default.mobilephones-stats: [numFiles=1, numRows=2, totalSize=100, rawDataSize=98]  
MapReduce Jobs Launched:  
Stage-Stage-1: Map: 1 Cumulative CPU: 3.88 sec HDFS Read: 6098 HDFS Write: 176 SUCCESS  
Total MapReduce CPU Time Spent: 3 seconds 888 msec  
OK  
Time taken: 35.033 seconds  
hive>
```

Display all records of table:

```
select * from mobilephones;
```

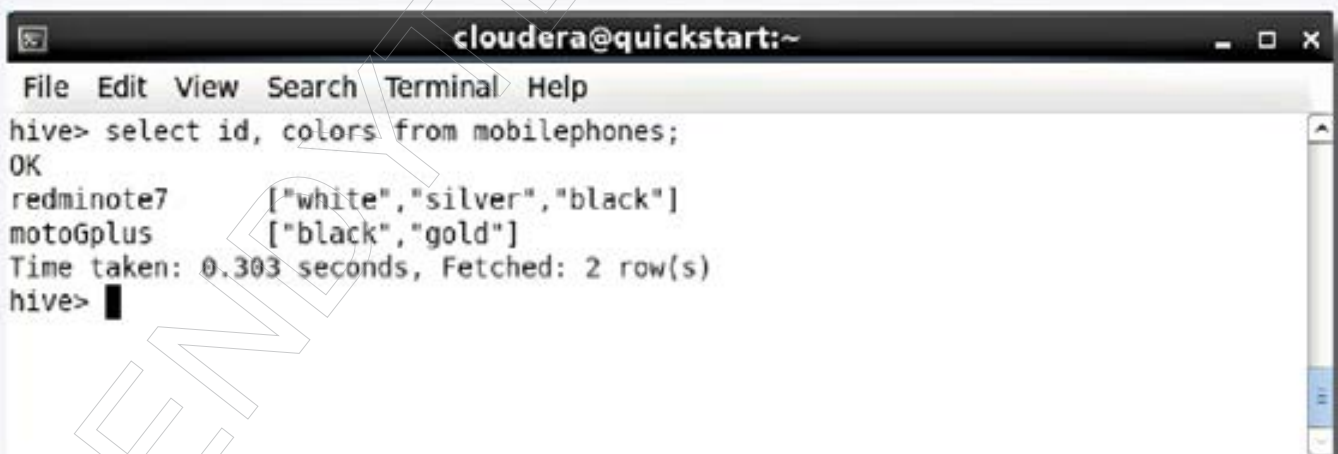


A terminal window titled 'cloudera@quickstart:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'hive>'. The user enters 'select * from mobilephones;'. The prompt changes to 'OK'. The output shows two rows of data: 'redminote7' with 'Redmi Note 7', '300.0', and an array of colors ['white','silver','black']; and 'motoGplus' with 'Moto G Plus', '200.0', and an array of colors ['black','gold']. The output also includes 'Time taken: 0.545 seconds, Fetched: 2 row(s)'. The prompt returns to 'hive>'.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive>  
hive> select * from mobilephones;  
OK  
redminote7      Redmi Note 7      300.0      ["white","silver","black"]      [4.5]  
motoGplus       Moto G Plus       200.0      ["black","gold"]      [4.5,5.5]  
Time taken: 0.545 seconds, Fetched: 2 row(s)  
hive>
```

Display records with all array elements:

```
select id, colors from mobilephones;
```

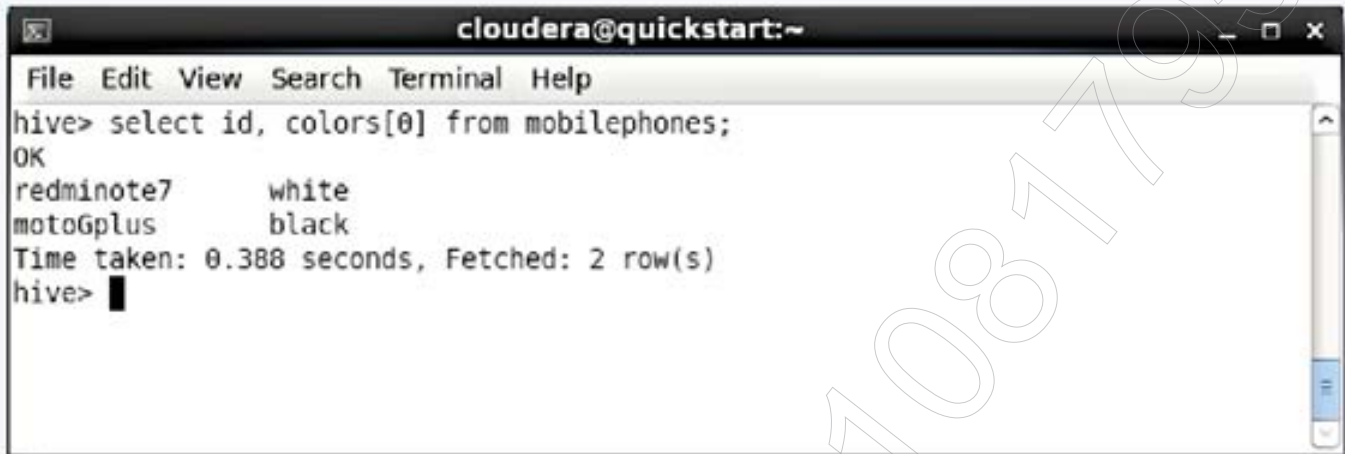


A terminal window titled 'cloudera@quickstart:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'hive>'. The user enters 'select id, colors from mobilephones;'. The prompt changes to 'OK'. The output shows two rows of data: 'redminote7' with an array of colors ['white','silver','black']; and 'motoGplus' with an array of colors ['black','gold']. The output also includes 'Time taken: 0.303 seconds, Fetched: 2 row(s)'. The prompt returns to 'hive>' with a cursor.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive> select id, colors from mobilephones;  
OK  
redminote7      ["white","silver","black"]  
motoGplus       ["black","gold"]  
Time taken: 0.303 seconds, Fetched: 2 row(s)  
hive> █
```

Display records with 1st element of an array:

```
select id, colors[0] from mobilephones;
```



A screenshot of a terminal window titled "cloudera@quickstart:~". The terminal shows a Hive query execution. The prompt is "hive>". The query entered is "select id, colors[0] from mobilephones;". The output shows "OK" followed by two rows of data: "redminote7" with "white" and "motoGplus" with "black". Below the data, it says "Time taken: 0.388 seconds, Fetched: 2 row(s)". The prompt "hive>" is followed by a cursor.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive> select id, colors[0] from mobilephones;  
OK  
redminote7      white  
motoGplus      black  
Time taken: 0.388 seconds, Fetched: 2 row(s)  
hive> █
```


Create a new table with array data type:

```
create table mobilephones_new (  
  id string,  
  title string,  
  cost float,  
  colors array<string>,  
  screen_size array<float>  
)
```

row format delimited fields terminated by ','
collection items terminated by '#';



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 being executed is:

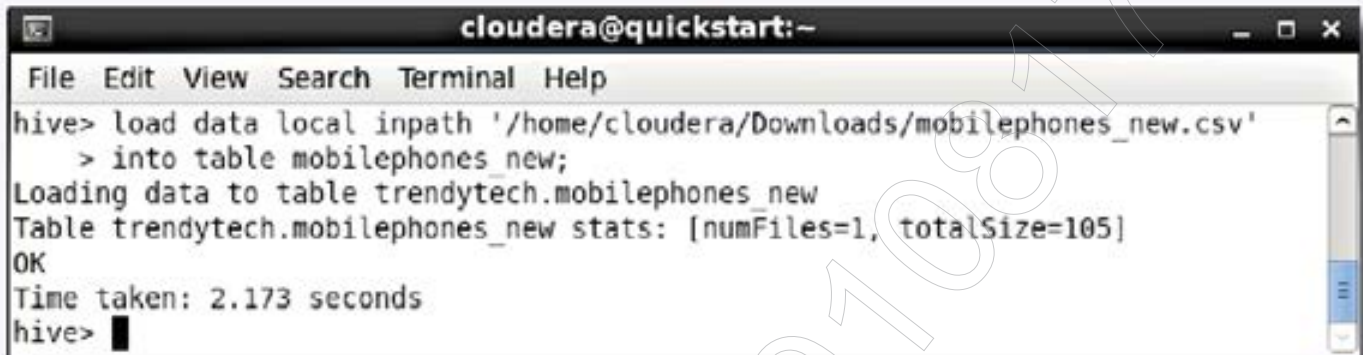
```
hive> create table mobilephones_new (  
  > id string,  
  > title string,  
  > cost float,  
  > colors array<string>,  
  > screen_size array<float>  
  > )  
  > row format delimited fields terminated by ','  
  > collection items terminated by '#';
```

The output of the command is:

```
OK  
Time taken: 0.342 seconds  
hive> █
```

Load data to the above table:

```
load data local inpath '/home/cloudera/Downloads/  
mobilephones_new.csv'  
into table mobilephones_new;
```



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive> load data local inpath '/home/cloudera/Downloads/mobilephones_new.csv'  
> into table mobilephones_new;  
Loading data to table trendytech.mobilephones_new  
Table trendytech.mobilephones_new stats: [numFiles=1, totalSize=105]  
OK  
Time taken: 2.173 seconds  
hive> █
```

Display all records of table:

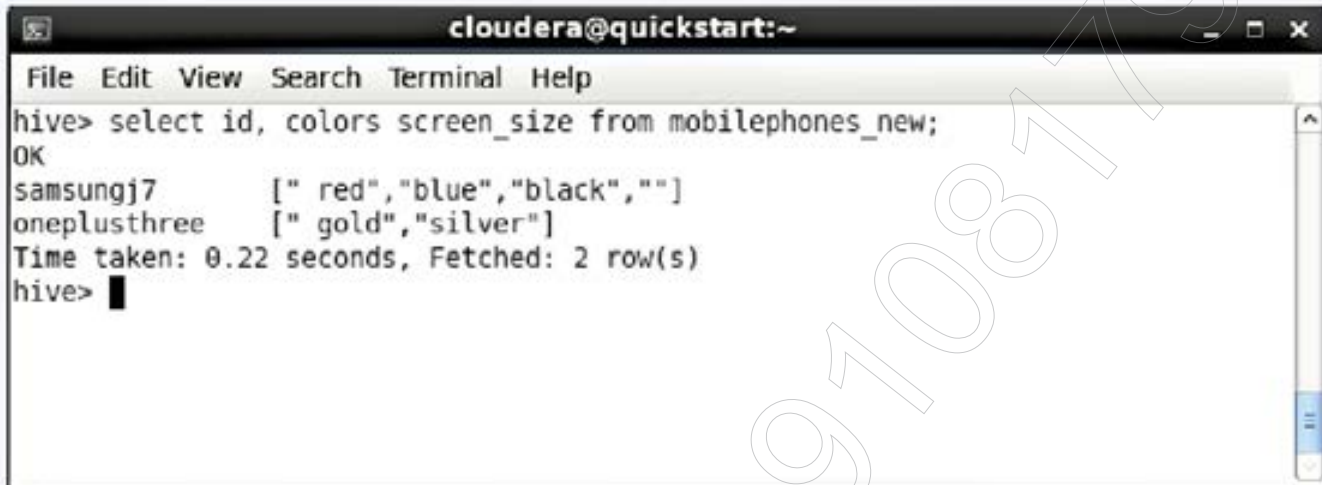
```
select * from mobilephones_new;
```



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive> select * from mobilephones_new;  
OK  
samsungj7          Samsung J7      250.0  [" red","blue","black",""]  [5.5]  
oneplusthree       One Plus Three 450.0  [" gold","silver"]          [4.5,5.5]  
Time taken: 0.194 seconds, Fetched: 2 row(s)  
hive> █
```

Display records with all array elements:

```
select id, colors screen_size from  
mobilephones_new;
```



A terminal window titled 'cloudera@quickstart:~' showing a Hive query. The query is 'select id, colors screen_size from mobilephones_new;'. The output shows two rows: 'samsungj7' with colors ['red','blue','black',''] and 'oneplusthree' with colors ['gold','silver']. The time taken is 0.22 seconds and 2 rows were fetched.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive> select id, colors screen_size from mobilephones_new;  
OK  
samsungj7      [" red","blue","black",""]  
oneplusthree   [" gold","silver"]  
Time taken: 0.22 seconds, Fetched: 2 row(s)  
hive>
```

Display records of 1st elements of array:

```
select id, colors[0] from mobilephones_new;
```



A terminal window titled 'cloudera@quickstart:~' showing a Hive query. The query is 'select id, colors[0] from mobilephones_new;'. The output shows two rows: 'samsungj7' with the first color 'red' and 'oneplusthree' with the first color 'gold'. The time taken is 0.221 seconds and 2 rows were fetched.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive> select id, colors[0] from mobilephones_new;  
OK  
samsungj7      red  
oneplusthree   gold  
Time taken: 0.221 seconds, Fetched: 2 row(s)  
hive>
```


Map

The Map is collection of key-value pairs or unordered collection of pairs. Map has no fixed size. The value is accessed using a unique key. Keys and values have their own data types.

Drop the *mobilephones* table which we have created earlier:

```
drop table mobilephones;
```



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive> show tables;  
OK  
all orders  
mobilephones  
mobilephones_new  
orders_no_partition  
orders_no_partition1  
Time taken: 0.029 seconds, Fetched: 5 row(s)  
hive>  
    > drop table mobilephones;  
OK  
Time taken: 0.799 seconds  
hive> █
```


Create a table with map data types:

```
create table mobilephones (  
  id string,  
  title string,  
  cost float,  
  colors array<string>,  
  screen_size array<float>,  
  features map<string, boolean>  
)  
row format delimited fields terminated by ','  
collection items terminated by '#'  
map keys terminated by ':';
```



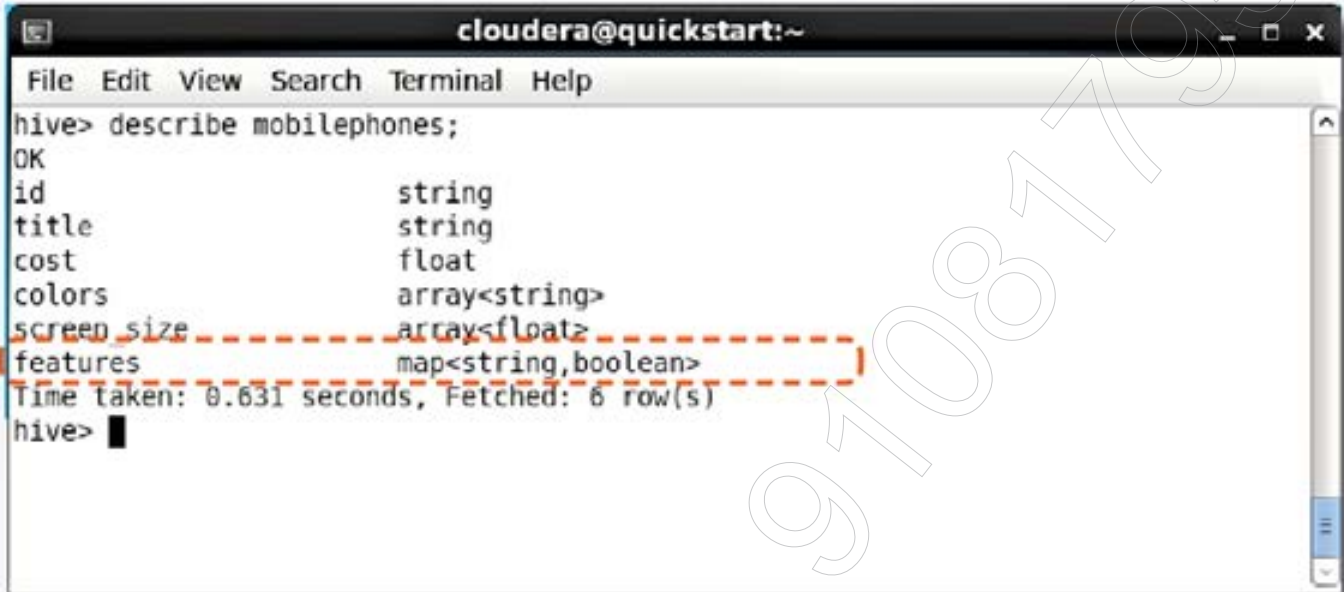
The screenshot shows a terminal window titled "cloudera@quickstart:~". The terminal displays the execution of a Hive command to create a table named "mobilephones". The command is as follows:

```
hive> create table mobilephones (  
  > id string,  
  > title string,  
  > cost float,  
  > colors array<string>,  
  > screen_size array<float>,  
  > features map<string, boolean>  
  > )  
  > row format delimited fields terminated by ','  
  > collection items terminated by '#'  
  > map keys terminated by ':';
```

The terminal output shows "OK" and "Time taken: 0.269 seconds". The prompt "hive>" is visible at the bottom of the terminal window.

Describe to see the table schema:

```
describe mobilephones;
```



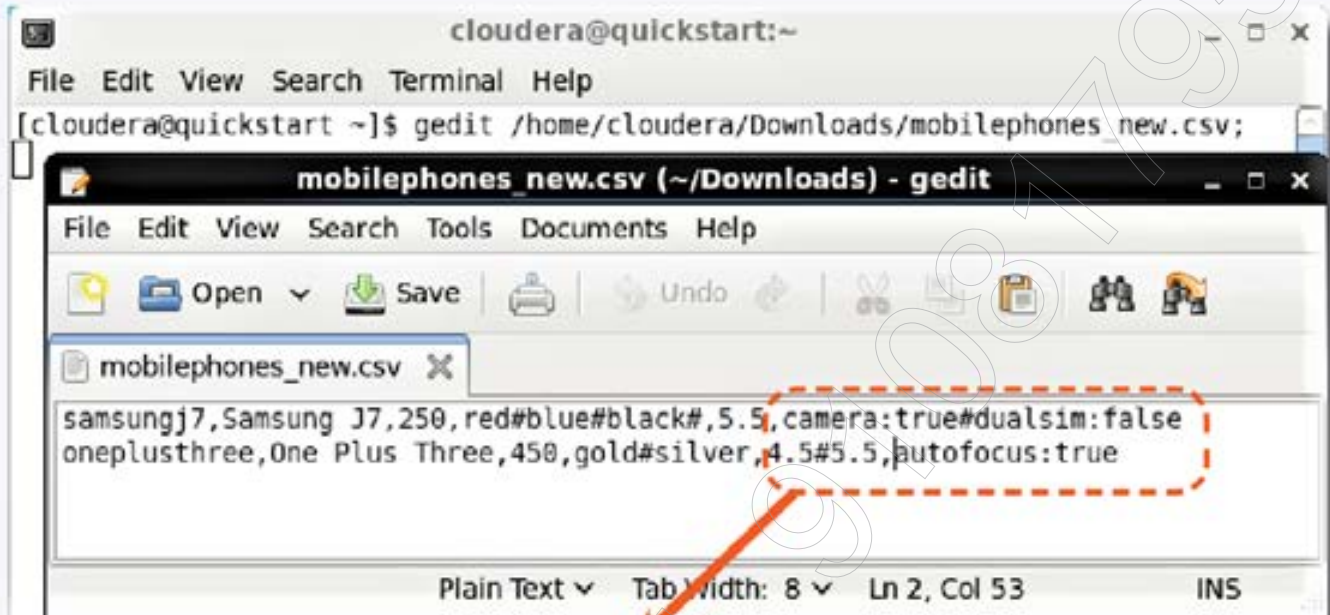
A terminal window titled 'cloudera@quickstart:~' showing the execution of the Hive command 'describe mobilephones;'. The output displays the schema of the 'mobilephones' table. The 'features' field is highlighted with a dashed orange box. Below the schema, the execution time and the number of rows fetched are shown.

```
File Edit View Search Terminal Help
hive> describe mobilephones;
OK
id                string
title             string
cost              float
colors            array<string>
screen_size       array<float>
features          map<string,boolean>
Time taken: 0.631 seconds, Fetched: 6 row(s)
hive> █
```

Note: Now we have to edit the mobilephones_new.csv file to add features

Edit the existing *mobilephones_new.csv* file:

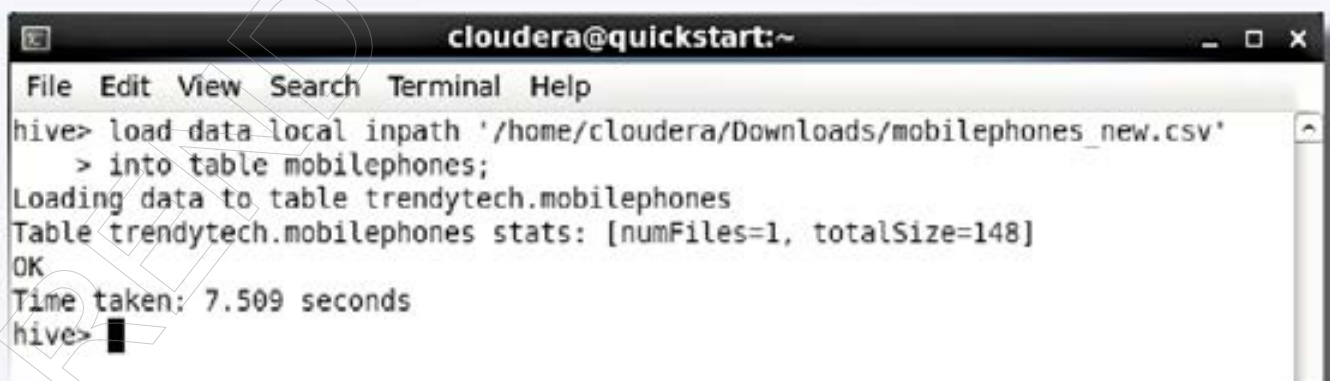
```
gedit /home/cloudera/Downloads/mobilephones_new.csv;
```



New data added for features

Now load data into the *mobilephones* table:

```
load data local inpath '/home/cloudera/  
Downloads/mobilephones_new.csv'  
into table mobilephones;
```



Display records of table:

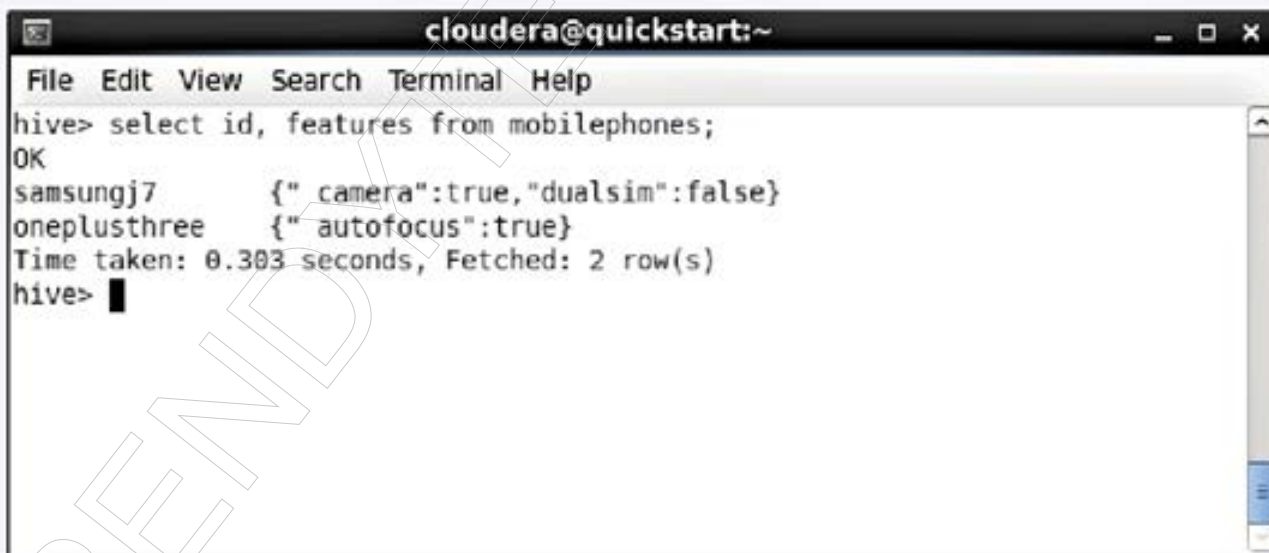
```
select * from mobilephones;
```



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive> select * from mobilephones;  
OK  
samsungj7      Samsung J7      250.0  [" red","blue","black",""]  [5.5]  
{" camera":true,"dualsim":false}  
oneplusthree   One Plus Three 450.0  [" gold","silver"]  [4.5,5.5]  {  
" autofocus":true}  
Time taken: 0.212 seconds, Fetched: 2 row(s)  
hive>
```

Display the *features* column data:

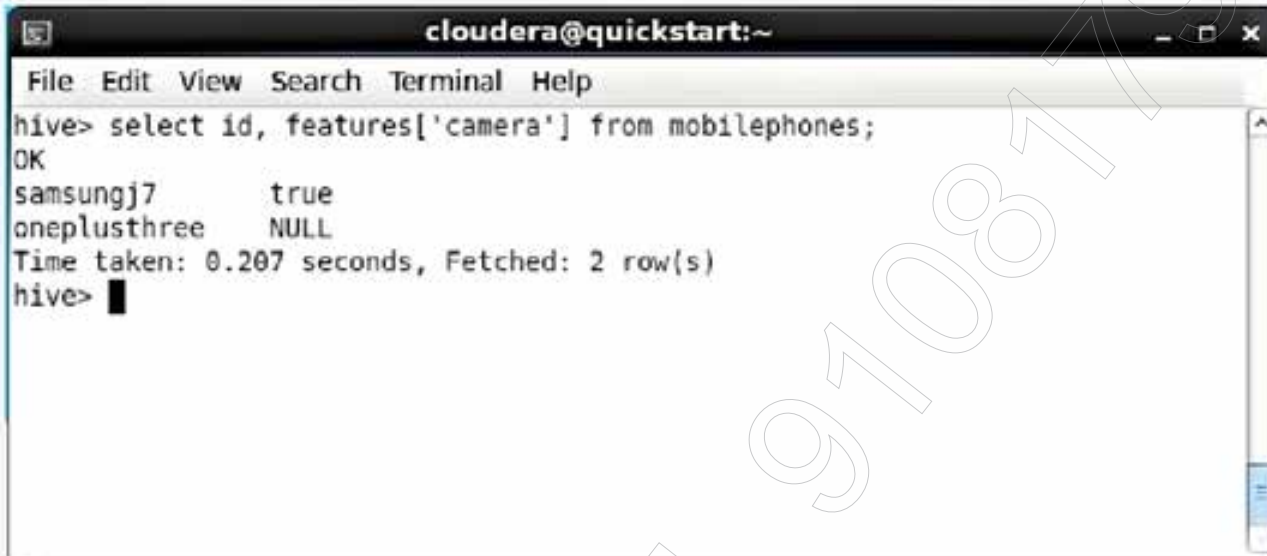
```
select id, features from mobilephones;
```



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive> select id, features from mobilephones;  
OK  
samsungj7      {" camera":true,"dualsim":false}  
oneplusthree   {" autofocus":true}  
Time taken: 0.303 seconds, Fetched: 2 row(s)  
hive>
```


Display individual records of Map:

```
select id, features['camera'] from  
mobilephones;
```



A screenshot of a terminal window titled "cloudera@quickstart:~". The window shows a Hive query execution. The query is "select id, features['camera'] from mobilephones;". The output shows two rows: "samsungj7" with "true" and "oneplusthree" with "NULL". Below the output, it says "Time taken: 0.207 seconds, Fetched: 2 row(s)". The prompt "hive>" is visible at the bottom.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive> select id, features['camera'] from mobilephones;  
OK  
samsungj7      true  
oneplusthree   NULL  
Time taken: 0.207 seconds, Fetched: 2 row(s)  
hive> █
```

Struct

Struct logically groups different types of data together into one entity. Each individual bit of data within a Struct can have different data types. Struct can hold any number of values. Each value referenced by a name.

Drop again the *mobilephones* table which we have created earlier:

```
drop table mobilephones;
```



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive> show tables;  
OK  
all_orders  
mobilephones  
mobilephones_new  
orders_no_partition  
orders_no_partition1  
Time taken: 0.029 seconds, Fetched: 5 row(s)  
hive>  
    > drop table mobilephones;  
OK  
Time taken: 0.799 seconds  
hive> █
```

Create a table with struct data type:

```
create table mobilephones (  
  id string,  
  title string,  
  cost float,  
  colors array<string>,  
  screen_size array<float>,  
  features map<string, boolean>,  
  information struct<battery:string,camera:string>  
)  
row format delimited fields terminated by ','  
collection items terminated by '#'  
map keys terminated by ':';
```

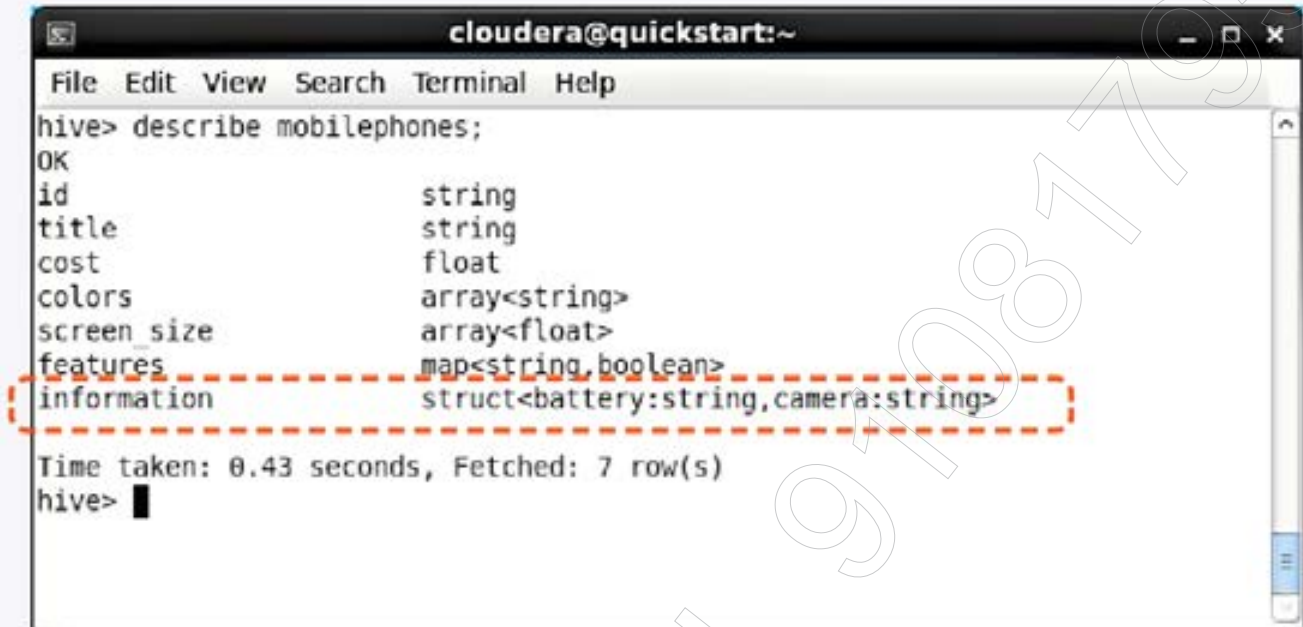


The screenshot shows a terminal window titled "cloudera@quickstart:~". The terminal contains the following text:

```
File Edit View Search Terminal Help  
hive> create table mobilephones (  
  > id string,  
  > title string,  
  > cost float,  
  > colors array<string>,  
  > screen_size array<float>,  
  > features map<string, boolean>,  
  > information struct<battery:string,camera:string>  
  > )  
  > row format delimited fields terminated by ','  
  > collection items terminated by '#'  
  > map keys terminated by ':';  
OK  
Time taken: 0.971 seconds  
hive>
```


Describe to see the table schema:

`describe mobilephones;`

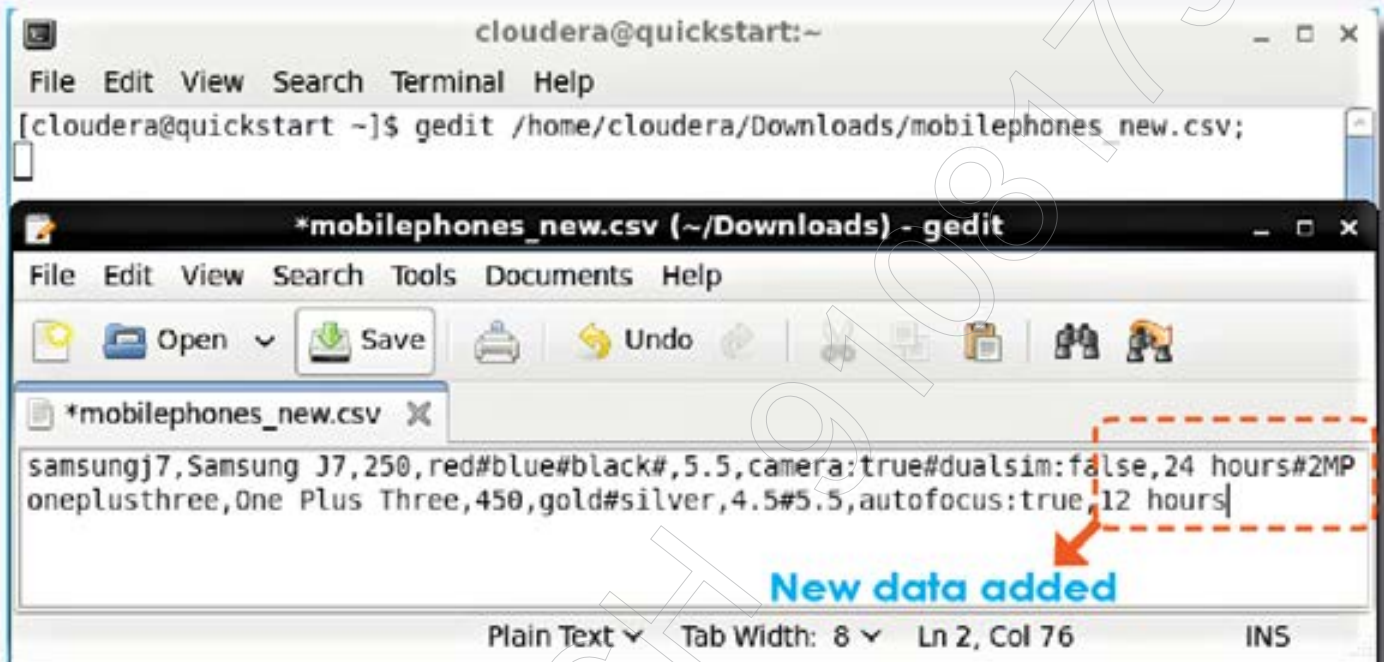


```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive> describe mobilephones;  
OK  
id                string  
title             string  
cost              float  
colors            array<string>  
screen_size       array<float>  
features          map<string,boolean>  
information       struct<battery:string,camera:string>  
Time taken: 0.43 seconds, Fetched: 7 row(s)  
hive> █
```

Note: Now we have to edit the *mobilephones_new.csv* file to add information column data.

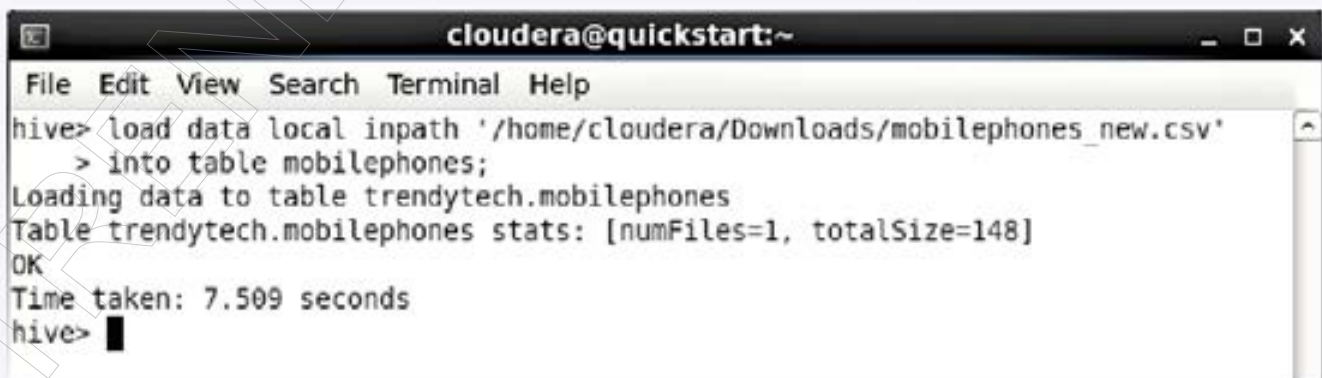
Edit the existing *mobilephones_new.csv* file to add new data for Struct column:

```
gedit /home/cloudera/Downloads/mobilephones_new.csv;
```



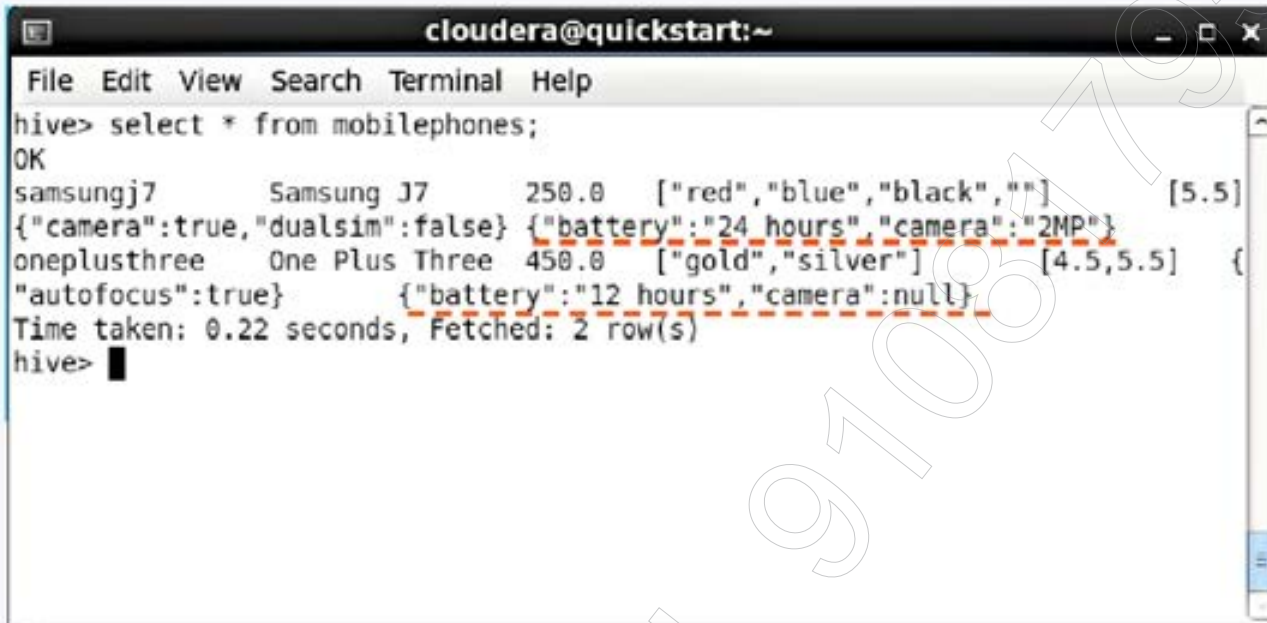
Now load data into the *mobilephones* table:

```
load data local inpath '/home/cloudera/
Downloads/mobilephones_new.csv'
into table mobilephones;
```



Display records of the Struct table:

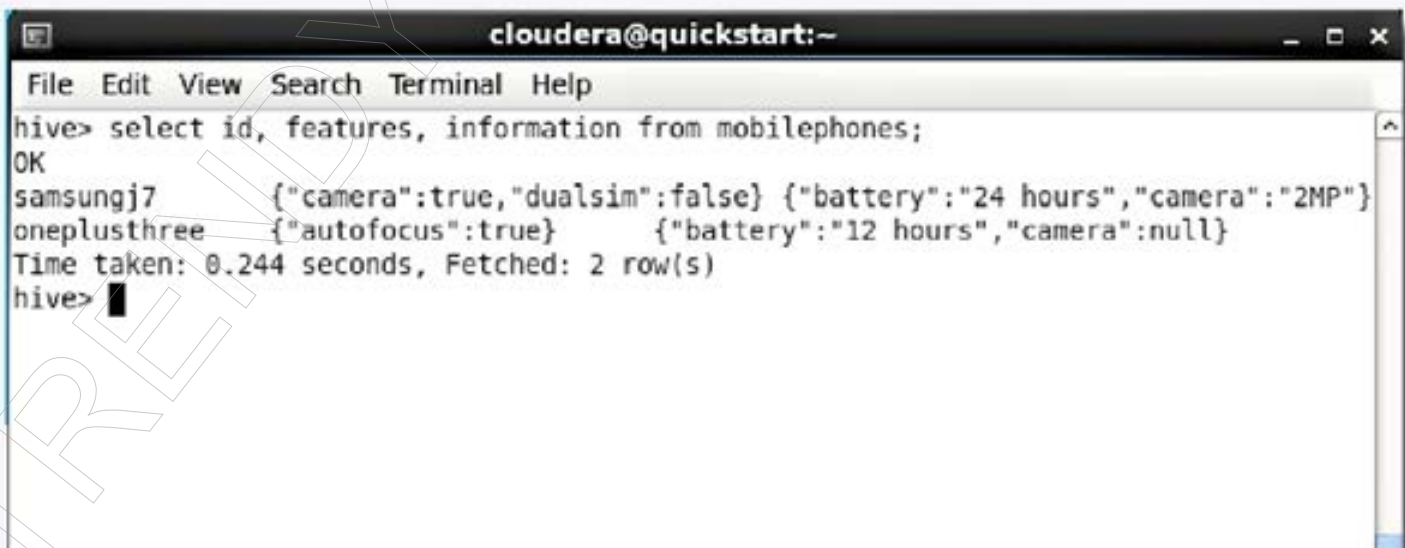
```
select * from mobilephones;
```



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive> select * from mobilephones;  
OK  
samsungj7      Samsung J7      250.0  ["red","blue","black",""]  [5.5]  
{"camera":true,"dualsim":false} {"battery":"24 hours","camera":"2MP"}  
oneplusthree   One Plus Three  450.0  ["gold","silver"]  [4.5,5.5]  {  
"autofocus":true} {"battery":"12 hours","camera":null}  
Time taken: 0.22 seconds, Fetched: 2 row(s)  
hive>
```

Display the *features* with *information* column data:

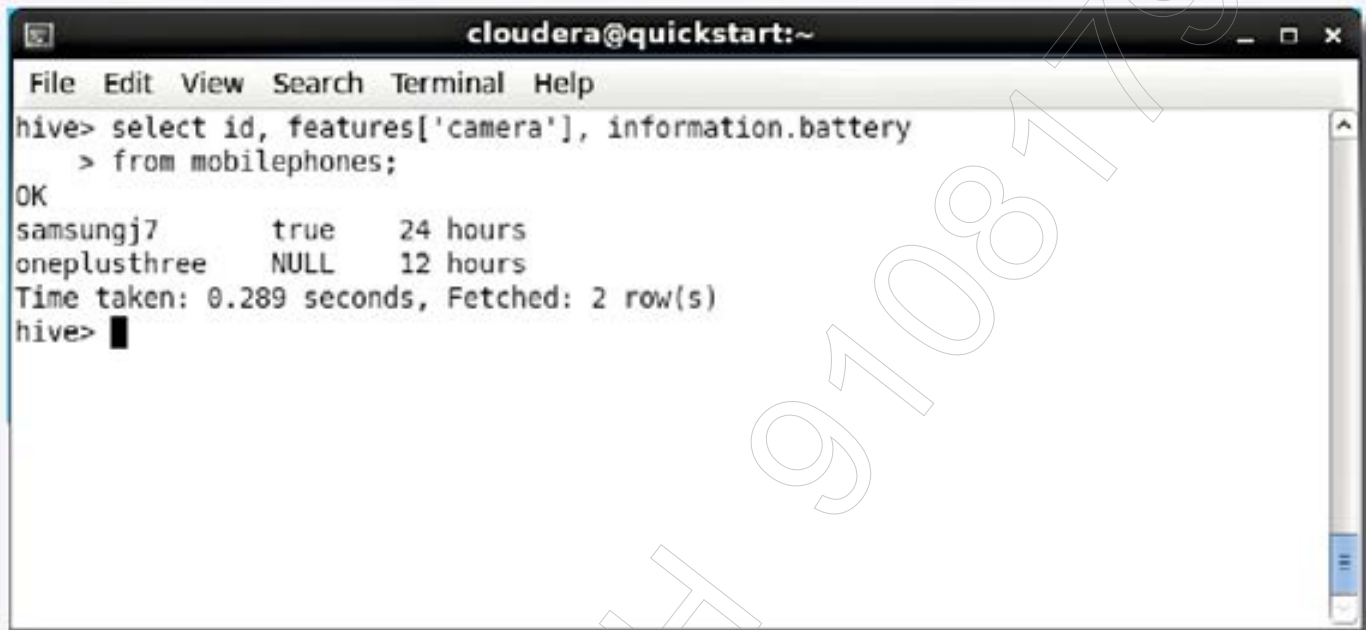
```
select id, features, information from  
mobilephones;
```



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive> select id, features, information from mobilephones;  
OK  
samsungj7      {"camera":true,"dualsim":false} {"battery":"24 hours","camera":"2MP"}  
oneplusthree   {"autofocus":true} {"battery":"12 hours","camera":null}  
Time taken: 0.244 seconds, Fetched: 2 row(s)  
hive>
```

Display individual records of Map and Struct:

```
select id, features['camera'], information.battery  
from mobilephones;
```



The screenshot shows a terminal window titled 'cloudera@quickstart:~'. The terminal displays the execution of a Hive query. The query is: `hive> select id, features['camera'], information.battery > from mobilephones;`. The output shows two rows of data: `samsungj7` with `true` for the camera feature and `24 hours` for the battery, and `oneplusthree` with `NULL` for the camera feature and `12 hours` for the battery. The terminal also shows the execution time and the number of rows fetched: `Time taken: 0.289 seconds, Fetched: 2 row(s)`. The prompt `hive>` is visible at the bottom of the terminal window.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
hive> select id, features['camera'], information.battery  
  > from mobilephones;  
OK  
samsungj7      true      24 hours  
oneplusthree   NULL      12 hours  
Time taken: 0.289 seconds, Fetched: 2 row(s)  
hive> █
```




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/>

