Apache Hive

Part 3

by Sumit Mittal

Apache Hive
Exercise 3

IMPORTANT

Copyright Infringement and Illegal Content Sharing Notice

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:

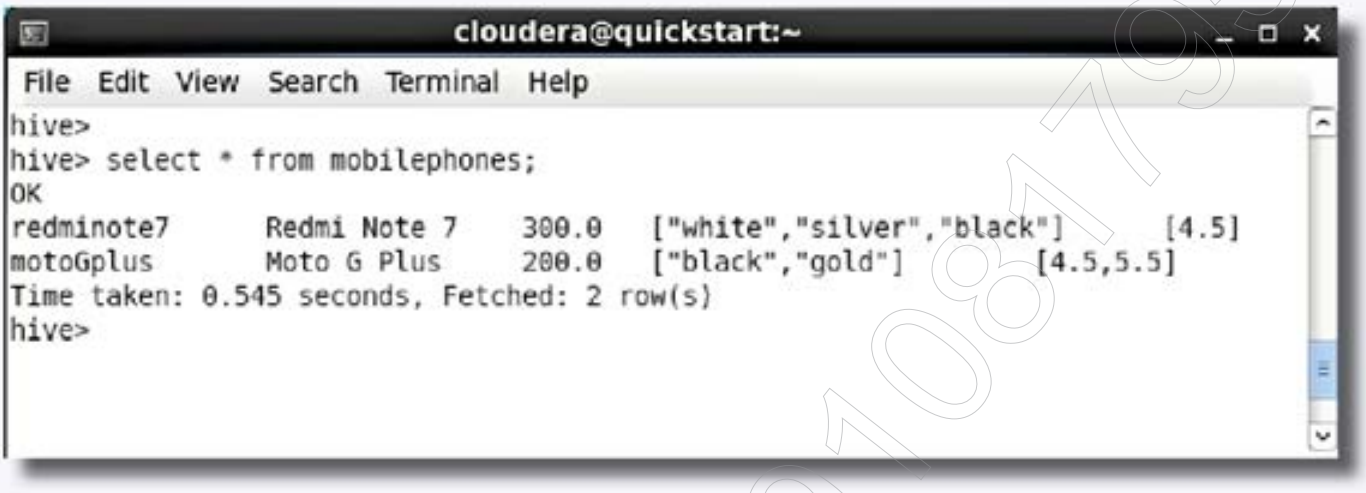
```
create table mobilephones (
   id string,
   title string,
   cost float,
   colors array<string>,
   screen_size array<float>
);
```

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));
```

Display all records of table:

```
select * from mobilephones;
```



Display records with all array elements:

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

Display records with 1st element of an array:

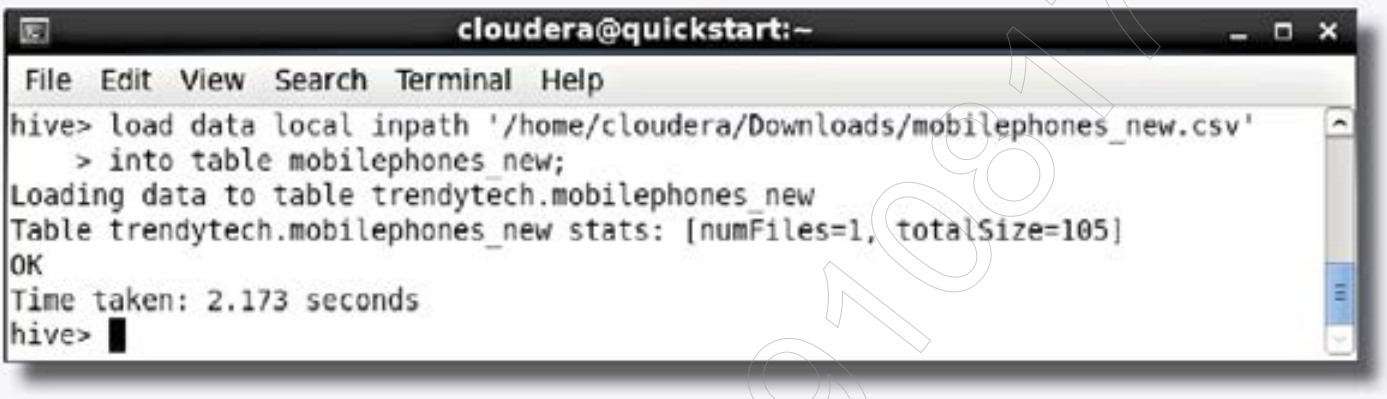
```
select id, colors[0] from mobilephones;
```

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 '#';
```

Load data to the above table:

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

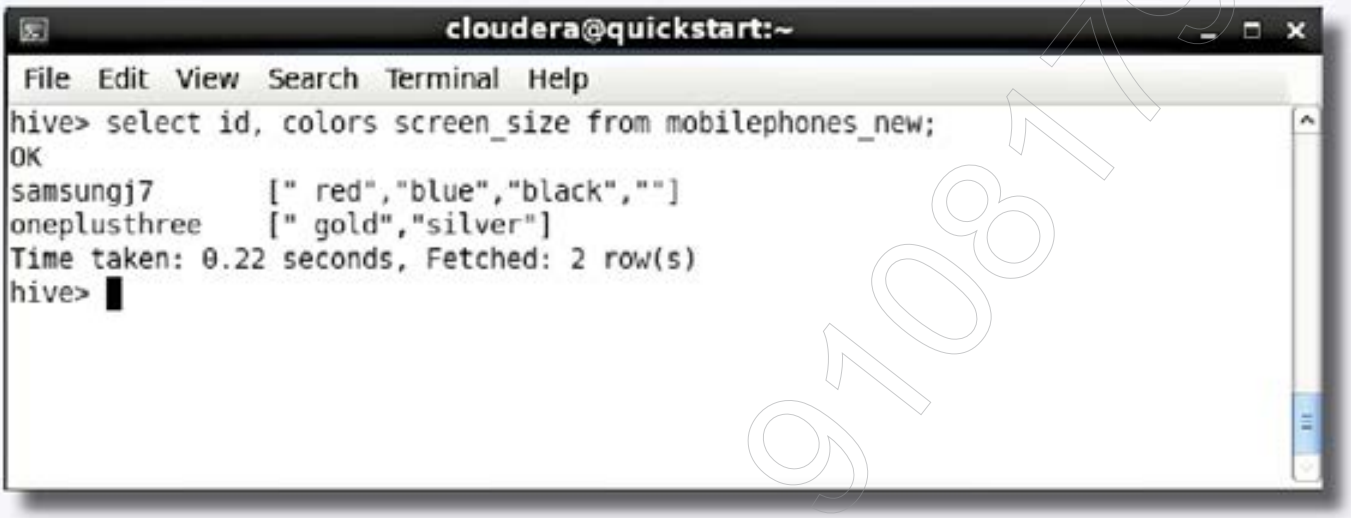


Display all records of table:

```
select * from mobilephones_new;
```

Display records with all array elements:

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



Display records of 1st elements of array:

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

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:

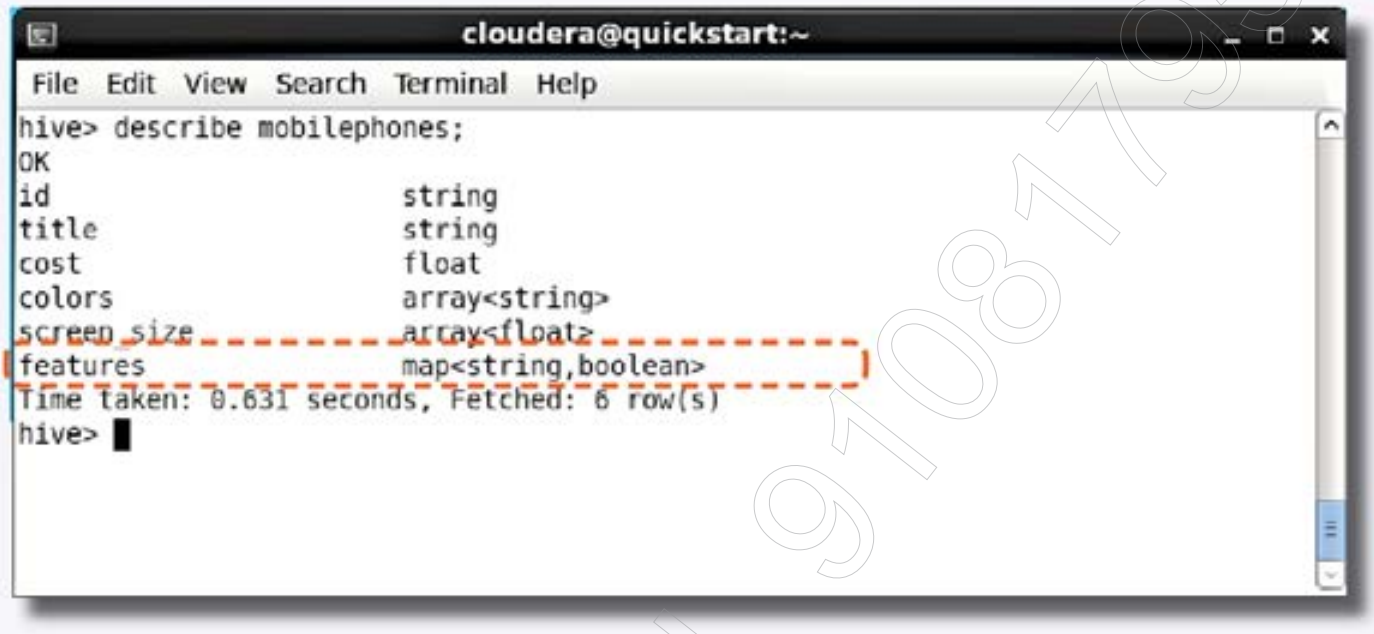
```
drop table mobilephones;
```

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 ':';
```

Describe to see the table schema:

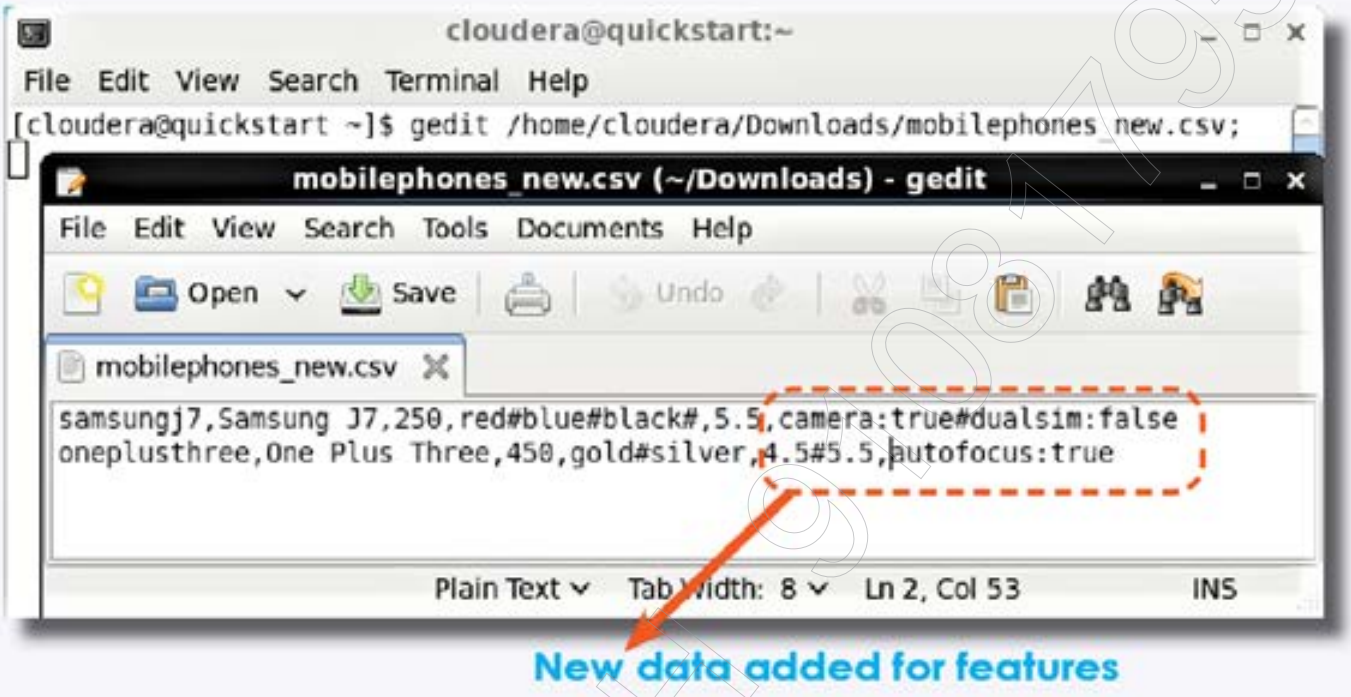
```
describe mobilephones;
```



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

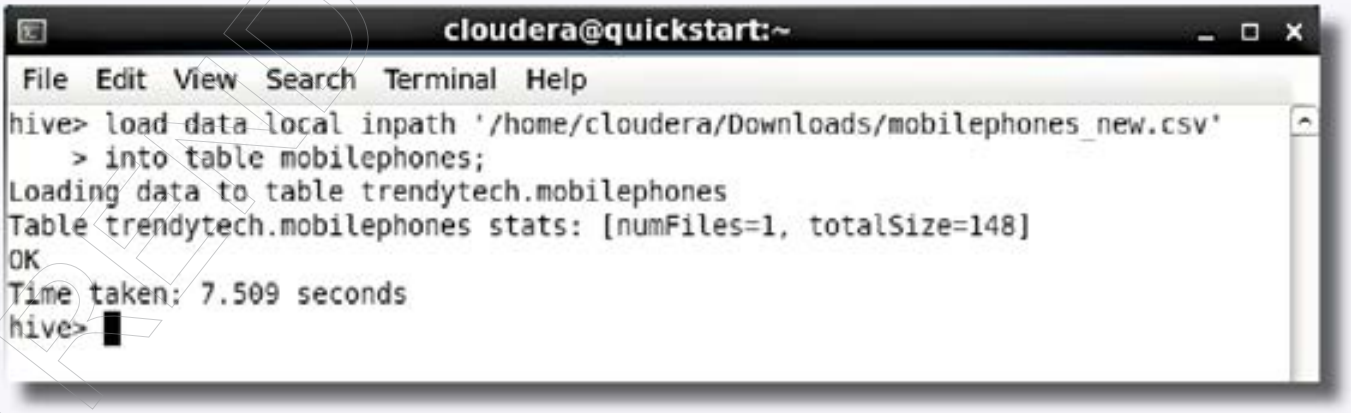
Edit the existing *mobilephones_new.csv* file:

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



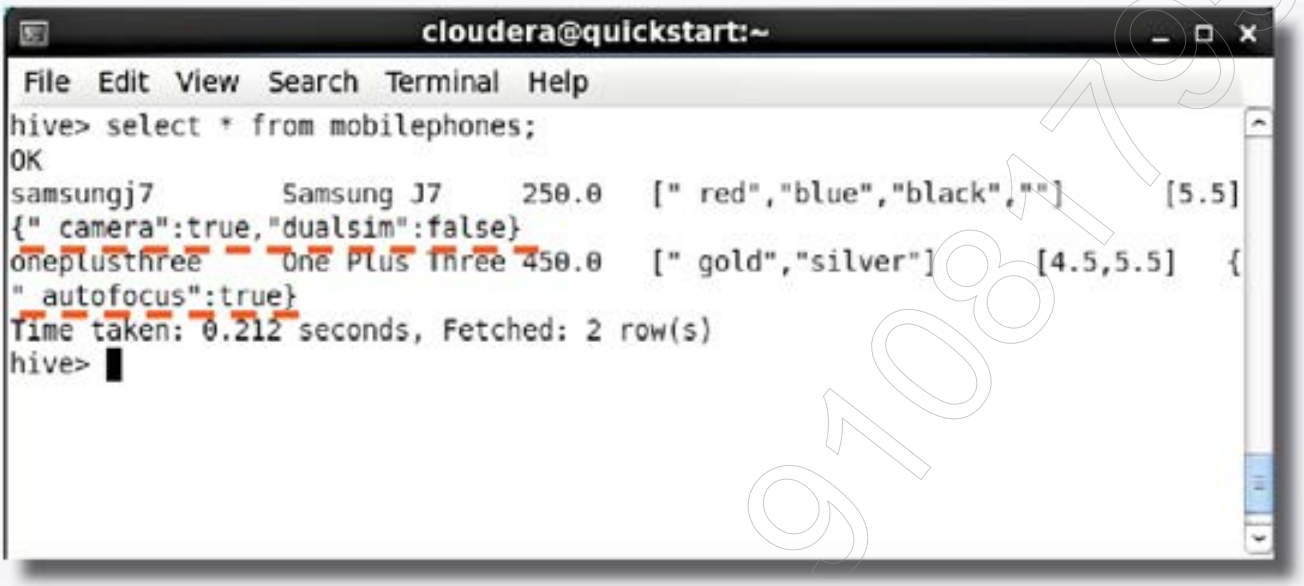
Now load data into the *mobilephones* table:

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



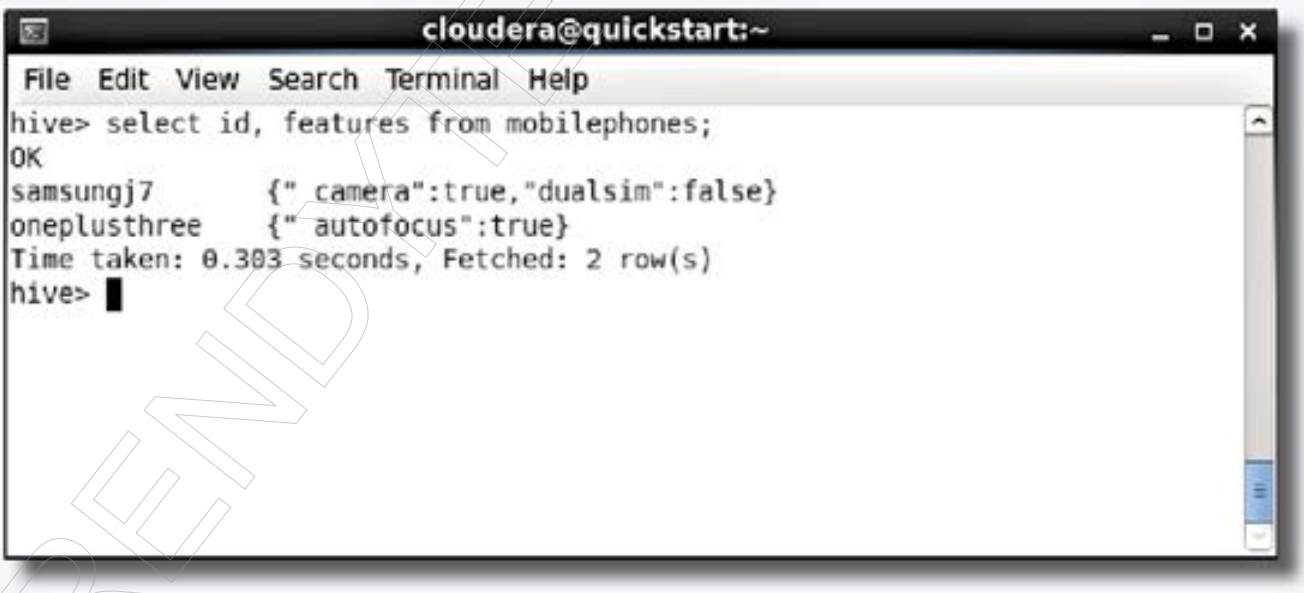
Display records of table:

```
select * from mobilephones;
```



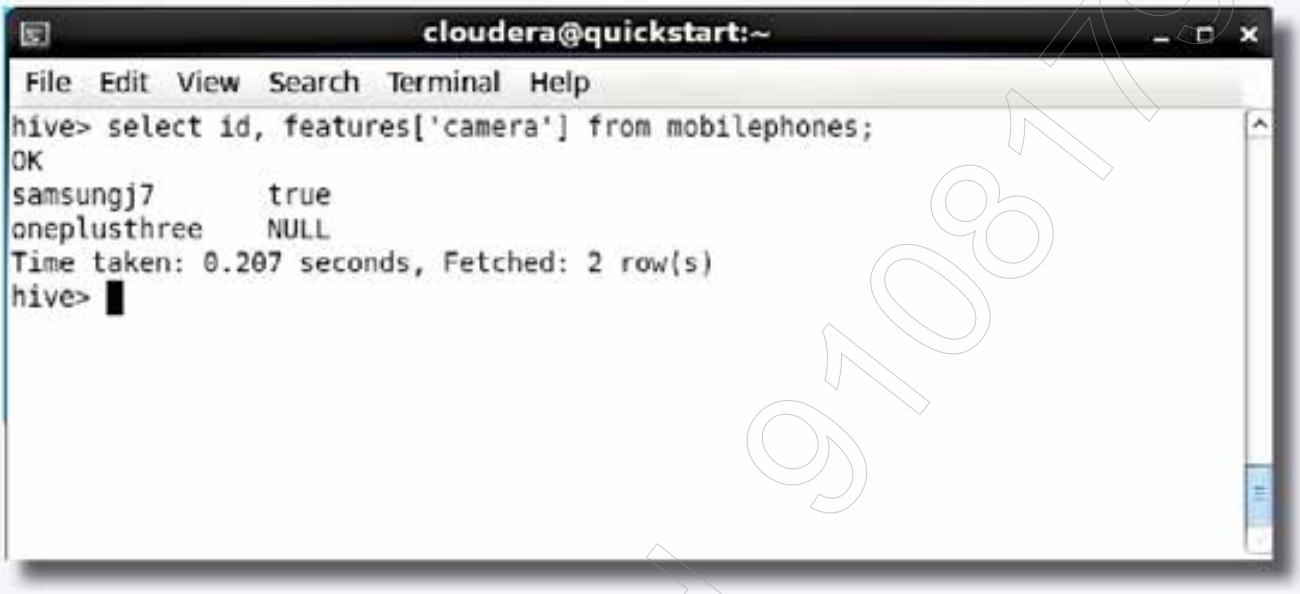
Display the *features* column data:

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



Display individual records of Map:

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

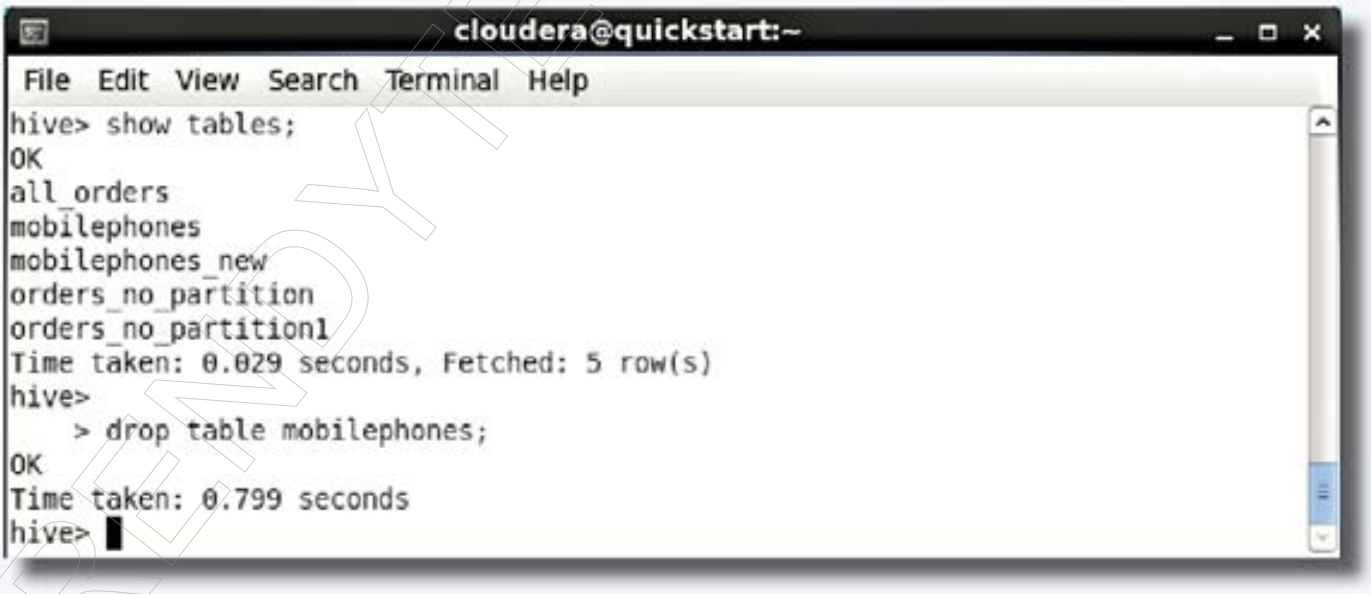


Struct

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;
```

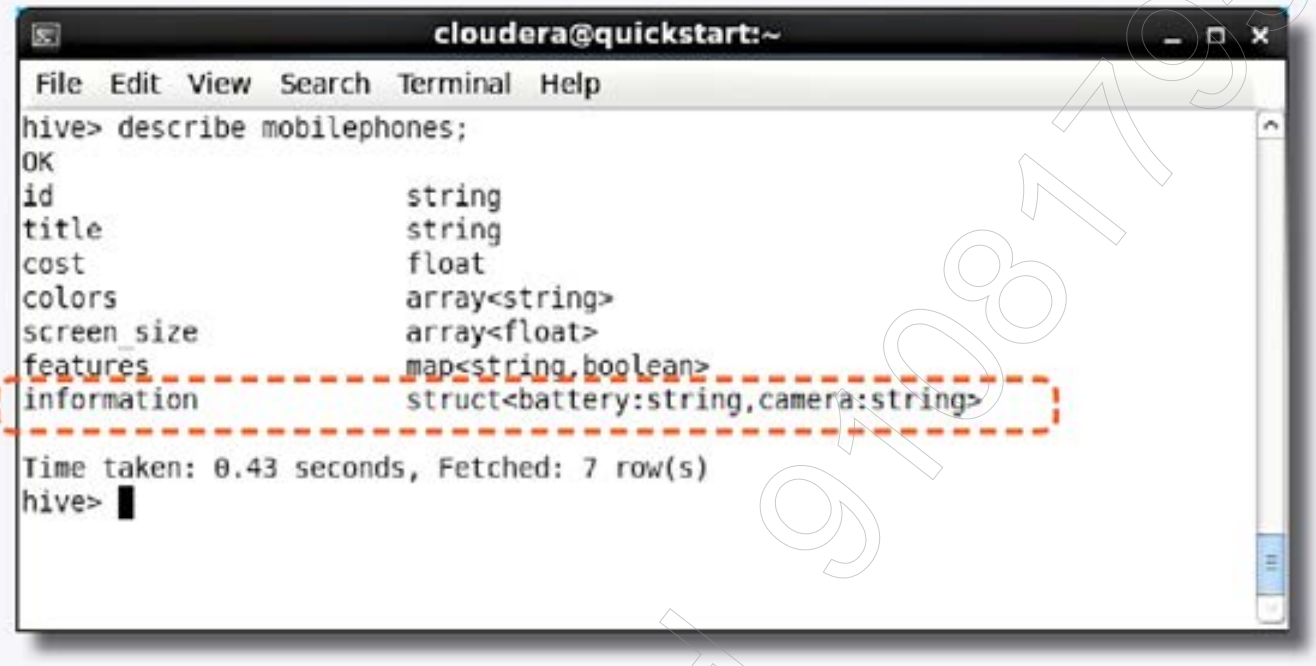


Create a table with struct data type:

```
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 ':';
```

Describe to see the table schema:

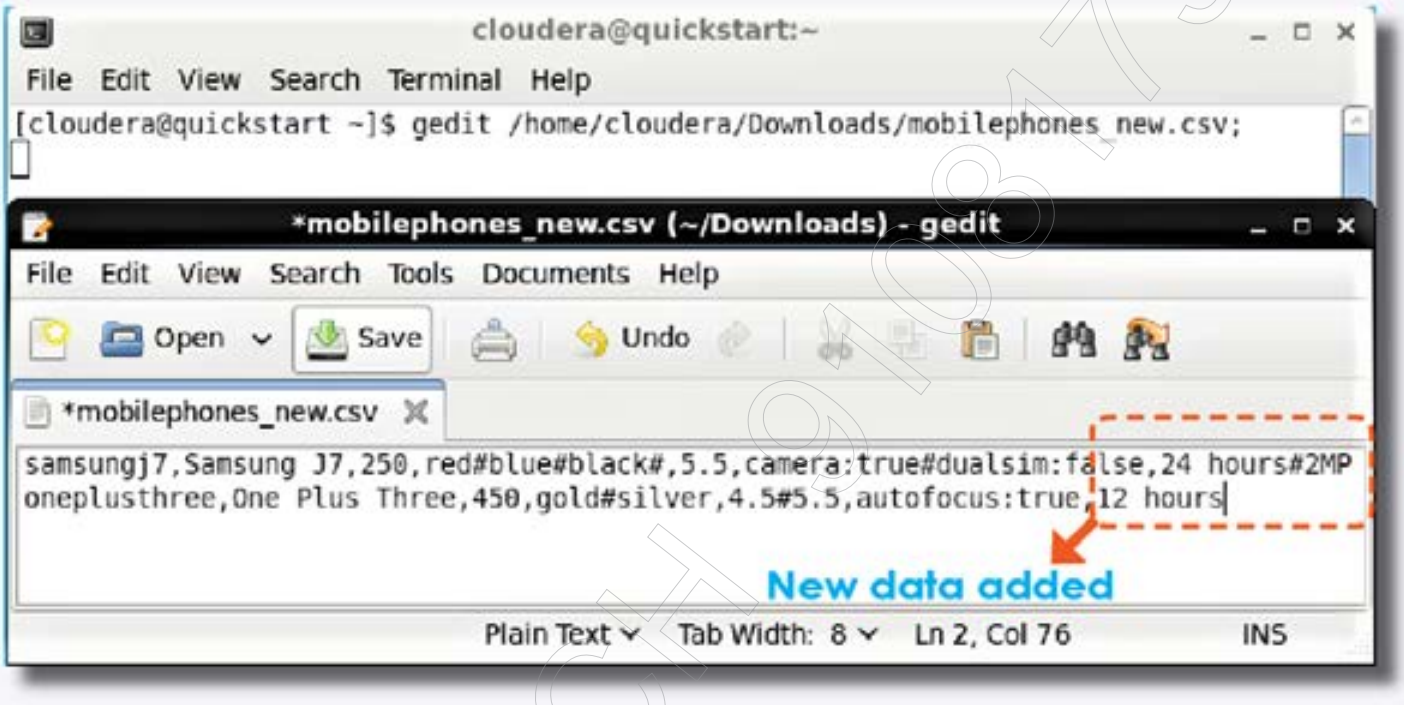
```
describe mobilephones;
```



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

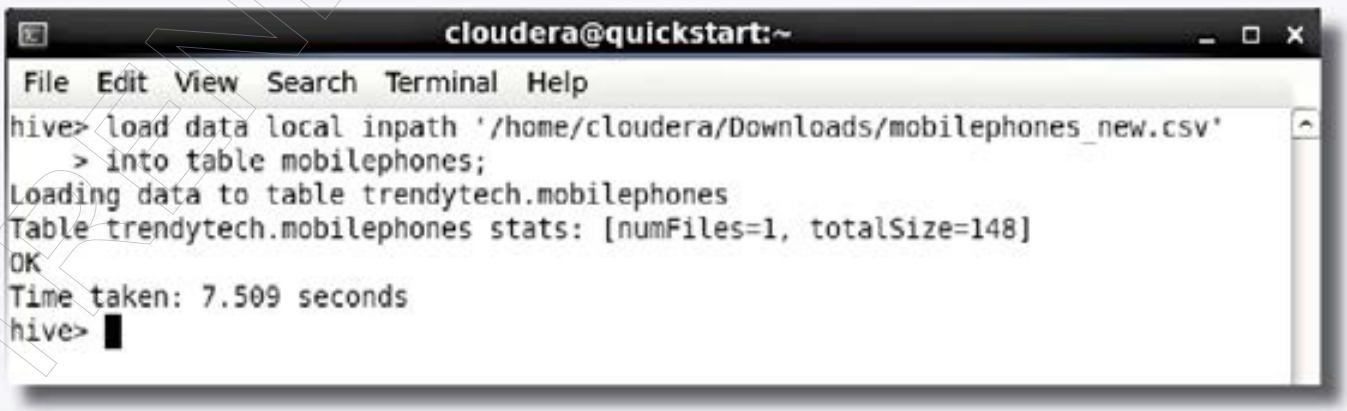
Edit the existing *mobilephones_new.csv* file to add new data for Struct column:

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



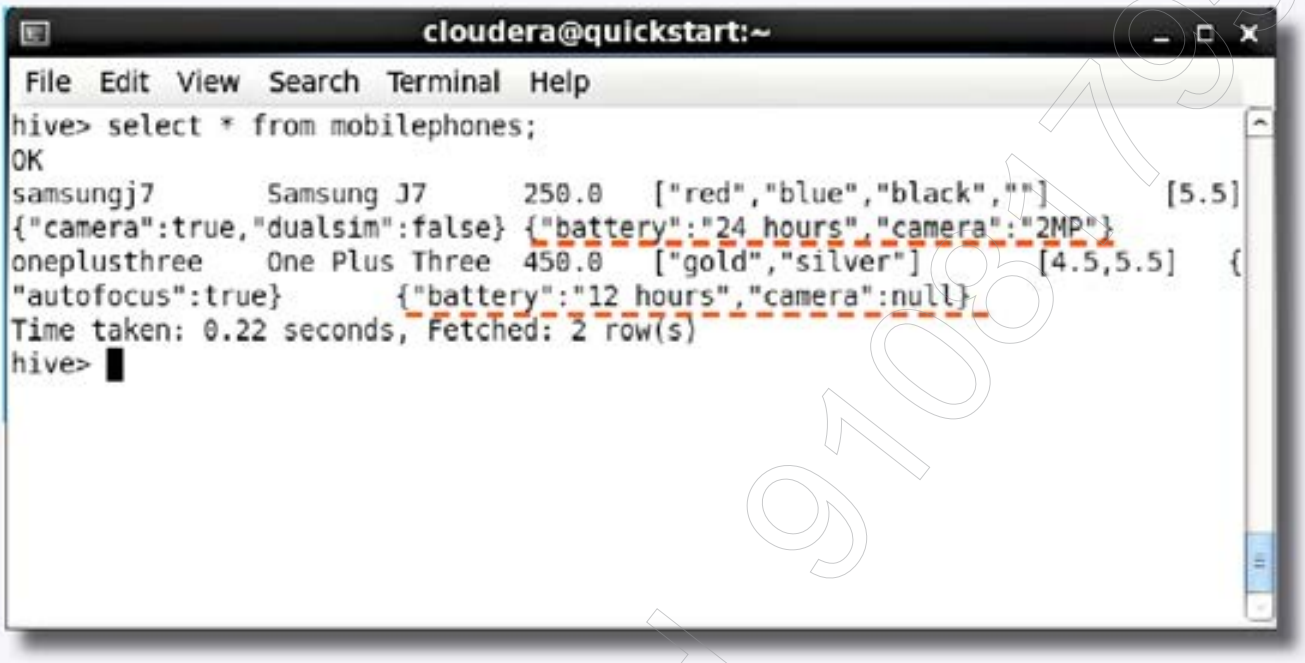
Now load data into the *mobilephones* table:

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



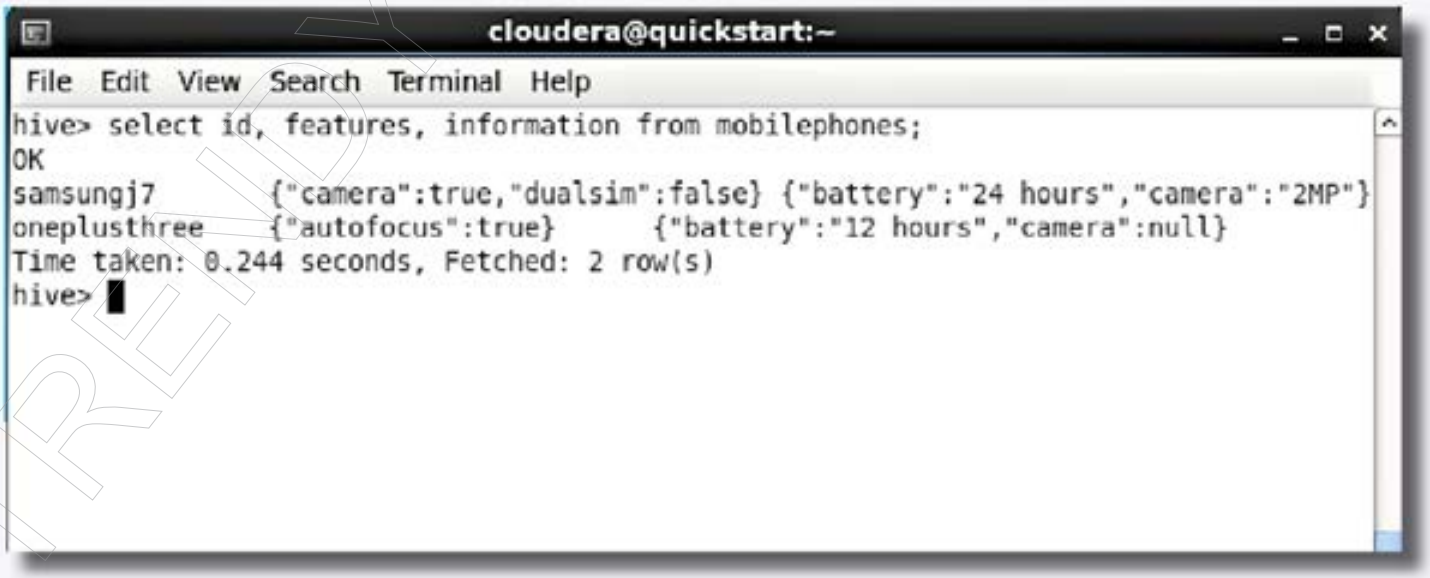
Display records of the Struct table:

```
select * from mobilephones;
```



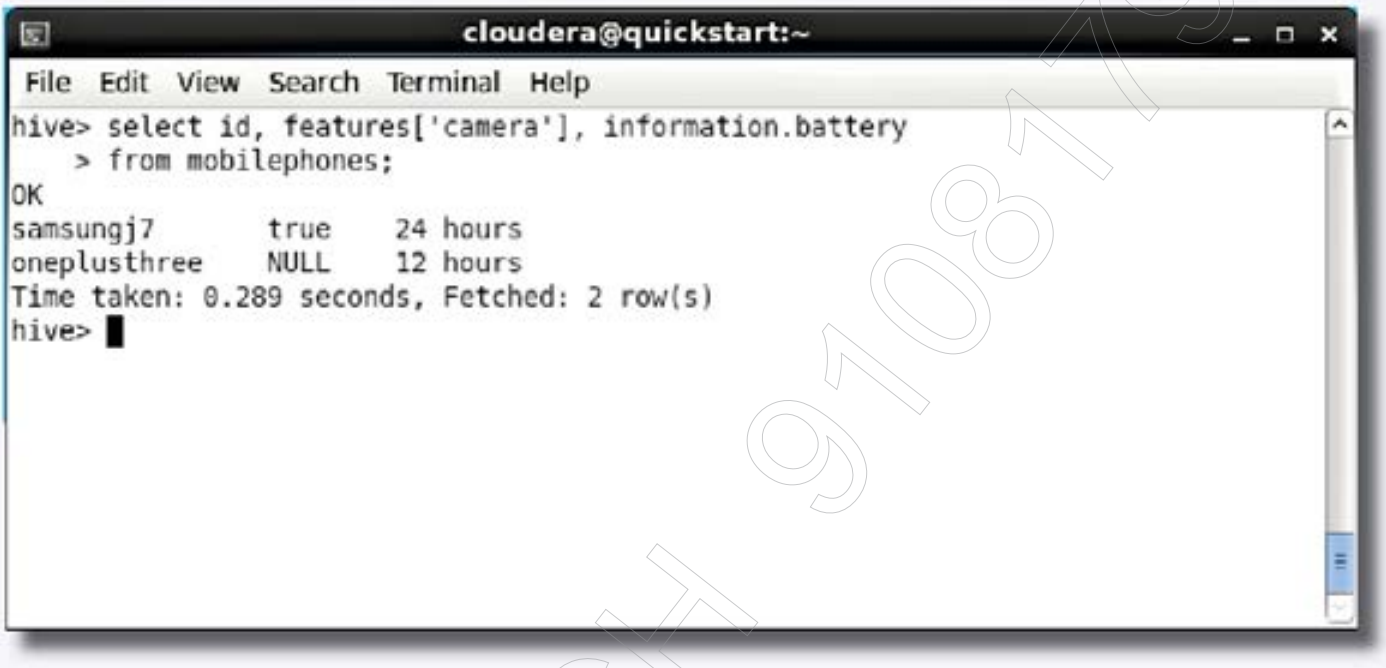
Display the *features* with *information* column data:

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



Display individual records of Map and Struct:

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



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/