

```
0: jdbc:hive2://> select address, count(*)as customer_count from customers group by address;
```

```
+-----+-----+--+
| address | customer_count |
+-----+-----+--+
| CA | 1 |
| WA | 3 |
+-----+-----+--+
```

```
0: jdbc:hive2://> select * from customers limit 1;
```

OK

```
+-----+-----+-----+--+
| customers.id | customers.name | customers.address |
+-----+-----+-----+--+
| 1111 | John | WA |
+-----+-----+-----+--+
```

Single line execution without login

```
[root@localhost apache-hive-2.1.1-bin]# bin/beeline -u jdbc:hive2:// -e "use pluralsight; select*from customers"
```

execution from file

```
[root@localhost apache-hive-2.1.1-bin]# nano read_customers.hql
```

```
use pluralsight;
select * from customer;
```

ctrl+x and save

```
[root@localhost apache-hive-2.1.1-bin]# bin/beeline -u jdbc:hive2:// -f read_customers.hql
```

```
0: jdbc:hive2://> create table if not exists orders (
```

```
..... > id bigint,
..... > product_id string,
..... > customer_id bigint,
..... > quantity int,
..... > amount double
..... > );
```

```
0: jdbc:hive2://> insert into orderin values (
```

```
..... > 111111,
```

```

..... > "phone",
..... > 1111,
..... > 3,
..... > 1200
..... > );

```

```

0: jdbc:hive2://> select * from orderin;
OK

```

orderid	product	customerid	quantity	amount
111111	phone	1111	3	1200.0

```

0: jdbc:hive2://> insert into orderin values (

```

```

..... > 111112,
..... > "camera",
..... > 1111,
..... > 1,
..... > 5200
..... > ), (
..... > 11113,
..... > "broom",
..... > 1111,
..... > 1,
..... > 10
..... > ), (
..... > 11114,
..... > "broom",
..... > 2222,
..... > 2,
..... > 20
..... > ), (
..... > 11115,
..... > "t-shirt",
..... > 4444,
..... > 2,
..... > 66
..... > );

```

```

0: jdbc:hive2://> select * from orderin;
OK

```

orderid	product	customerid	quantity	amount
111111	phone	1111	3	1200.0
111112	camera	1111	1	5200.0
111113	broom	1111	1	10.0
111114	broom	2222	2	20.0
111115	t-shirt	4444	2	66.0

111111	phone	1111	3	1200.0
111112	camera	1111	1	5200.0
111113	broom	1111	1	10.0
111114	broom	2222	2	20.0
111115	t-shirt	4444	2	66.0

```
0: jdbc:hive2://> select customers.id,name,product_id
..... > from customers join orderin
..... > where customers.id = ordrs.customer_id;
```

2018-02-19 19:31:08 End of local task; Time Taken: 1.917 sec.

customers.id	name	product_id
1111	John	phone
1111	John	camera
1111	John	broom
2222	Emily	broom
4444	Jane	t-shirt

5 rows selected (32.1 seconds)

Database and Tables:

```
[keerthan@localhost hive-install]$ cd apache-hive-2.1.1-bin
[keerthan@localhost apache-hive-2.1.1-bin]$ hadoop fs -ls /user/hive/warehouse
Found 3 items
drwxrwxr-x - root supergroup 0 2018-02-19 19:11 /user/hive/warehouse/orders
drwxrwxr-x - root supergroup 0 2018-02-19 19:56 /user/hive/warehouse/plural.db
drwxrwxr-x - root supergroup 0 2018-02-19 19:15 /user/hive/warehouse/pluralsight.db
```

External storage

```
0: jdbc:hive2://> create external table if not exists products (
..... > id string,
..... > title string,
..... > cost float
..... > )
..... > comment "Table to store product information sold in stores"
..... > location '/data/';
```

OK

create a csv in hadoop file system

```
[root@localhost apache-hive-2.1.1-bin]# hadoop fs -mkdir /dat
[root@localhost apache-hive-2.1.1-bin]# hadoop fs -mkdir /data
[root@localhost apache-hive-2.1.1-bin]# hadoop fs -copyFromLocal product.csv /data/
copyFromLocal: `product.csv': No such file or directory
[root@localhost apache-hive-2.1.1-bin]# hadoop fs -copyFromLocal products.csv /data/
[root@localhost apache-hive-2.1.1-bin]# hadoop fs -ls /data/
Found 1 items
-rw-r--r-- 1 root supergroup 116 2018-02-19 21:07 /data/products.csv
```

```
0: jdbc:hive2://> create external table if not exists products (
..... > id string,
..... > title string,
..... > cost float
..... > )
..... > comment "Table to store product information sold in stores"
..... > row format delimited fields terminated by ','
..... > stored as textfile
..... > location '/data/';
OK
```

```
: jdbc:hive2://> select * from products;
OK
```

```
+-----+-----+-----+--+
| products.id | products.title | products.cost |
+-----+-----+-----+--+
| iphone7 | iPhone 7 | 950.0 |
| camera_cannon | Canon 570x | 1000.0 |
| washingmachine_samsung | Samsung Swift | 400.0 |
| tv_vu | VU 56 Inch | 600.0 |
+-----+-----+-----+--+
```

```
0: jdbc:hive2://> alter table fresh_products
..... > rename to freshproducts;
```

```
0: jdbc:hive2://> create table if not exists fresh_products like products2;
```

```
0: jdbc:hive2://> alter table freshproducts add columns (
..... > expiry_date date
..... > comment "Expiry date of fresh produce"
..... > );
```

```
: jdbc:hive2://> alter table freshproducts
..... > change column id id string
..... > after title;
```

Temporary table:

```
0: jdbc:hive2://> create temporary table test_customers
..... > like customers;
OK
No rows affected (0.147 seconds)
```

```
0: jdbc:hive2://> insert into test_customers values (
..... > 9999,
..... > "Jill",
..... > "MN"
..... > );
```

Temporary table with same name as existing table

```
0: jdbc:hive2://> create temporary table customers
..... > like orderin;
OK
No rows affected (0.706 seconds)
```

```
0: jdbc:hive2://> show tables;
```

OK

```
+-----+---+
| tab_name |
+-----+---+
| customers |
| freshproducts |
| orderin |
| products2 |
| test |
+-----+---+
```

```
: jdbc:hive2://> describe customers;
OK
```

```
+-----+-----+-----+---+
| col_name | data_type | comment |
```

```

+-----+-----+-----+--+
| id | bigint | |
| product_id | string | |
| customer_id | bigint | |
| quantity | int | |
| amount | double | |
+-----+-----+-----+--+

```

```

0: jdbc:hive2://> !q
0: jdbc:hive2://> describe customers;
OK

```

```

+-----+-----+-----+--+
| col_name | data_type | comment |
+-----+-----+-----+--+
| id | bigint | |
| name | string | |
| address | string | |
+-----+-----+-----+--+
3 rows selected (1.074 seconds)

```

create a file
[keerthan@localhost apache-hive-2.1.1-bin]\$ nano freshproducts3.csv

feed data

```

broccoli, Broccoli, 5
spinach, Spinach, 7
carrot, Local Carrots, 4
potato, Idaho Potatoes, 4

```

```

load data to table
0: jdbc:hive2://> load data local inpath 'freshproducts3.csv'
. . . . . > into table freshproducts;
Loading data to table pluralsight.freshproducts
OK
No rows affected (0.85 seconds)

```

```

0: jdbc:hive2://> select * from freshproducts;
18/02/20 11:01:00 [7ec656d4-a360-4176-8f6c-f86019c94765 main]: ERROR
hdfs.KeyProviderCache: Could not find uri with key [dfs.encryption.key.provider.uri] to
create a keyProvider !!
OK

```

```

+-----+-----+-----+-----+-----+--+
| freshproducts.id | freshproducts.title | freshproducts.cost | freshproducts.expiry_date |

```

broccoli	Broccoli	5.0	NULL
spinach	Spinach	7.0	NULL
carrot	Local Carrots	4.0	NULL
potato	Idaho Potatoes	4.0	NULL

4 rows selected (1.252 seconds)

```
[root@localhost apache-hive-2.1.1-bin]# hadoop fs -cat
/user/hive/warehouse/pluralsight.db/freshproducts/*
broccoli, Broccoli, 5
spinach, Spinach, 7
carrot, Local Carrots, 4
potato, Idaho Potatoes, 4
[root@localhost apache-hive-2.1.1-bin]# hadoop fs -copyFromLocal freshproducts.csv
/data/
```

Loading data to table pluralsight.freshproducts

OK

No rows affected (0.426 seconds)

```
[root@localhost apache-hive-2.1.1-bin]# hadoop fs -ls /data/
Found 2 items
-rw-r--r-- 1 root supergroup 95 2018-02-20 11:06 /data/freshproducts.csv
-rw-r--r-- 1 root supergroup 116 2018-02-19 21:07 /data/products.csv
```

Load csv from hdfs file system

0: jdbc:hive2://> load data inpath '/data/freshproducts.csv' into table freshproducts;

HDFS data appended to the existing table

0: jdbc:hive2://> select * from freshproducts;

OK

freshproducts.id	freshproducts.title	freshproducts.cost	freshproducts.expiry_date
broccoli	Broccoli	5.0	NULL
spinach	Spinach	7.0	NULL
carrot	Local Carrots	4.0	NULL
potato	Idaho Potatoes	4.0	NULL
	NULL	NULL	NULL
broccoli	Broccoli	5.0	NULL
spinach	Spinach	7.0	NULL
carrot	Local Carrots	4.0	NULL

potato	Idaho Potatoes	4.0	NULL
--------	----------------	-----	------

overwrite table contents

0: jdbc:hive2://> load data local inpath 'freshproducts.csv'

..... > overwrite into table freshproducts;

Loading data to table pluralsight.freshproducts

OK

No rows affected (0.462 seconds)

0: jdbc:hive2://> select * from freshproducts;

OK

freshproducts.id	freshproducts.title	freshproducts.cost	freshproducts.expiry_date
broccoli	Broccoli	5.0	NULL
spinach	Spinach	7.0	NULL
carrot	Local Carrots	4.0	NULL
potato	Idaho Potatoes	4.0	NULL
	NULL	NULL	NULL

5 rows selected (0.279 seconds)

Merge two tables

0: jdbc:hive2://> insert into table products2

..... > select id, title, cost from freshproducts;

0: jdbc:hive2://> select * from products2;

OK

products2.title	products2.id	products2.cost
broccoli	Broccoli	5.0
spinach	Spinach	7.0
carrot	Local Carrots	4.0
potato	Idaho Potatoes	4.0
	NULL	NULL
iphone7	iPhone 7	950.0
camera_cannon	Canon 570x	1000.0
washingmachine_samsung	Samsung Swift	400.0
tv_vu	VU 56 Inch	600.0

Load single table data to multiple table

0: jdbc:hive2://> describe products2;

OK

col_name	data_type	comment
title	string	
id	string	
cost	float	

3 rows selected (0.135 seconds)

0: jdbc:hive2://> alter table products2

..... > change column title title string

..... > after id;

OK

No rows affected (0.229 seconds)

split products2 table data to 2 tables

0: jdbc:hive2://> create table product_name (id string, name string);

0: jdbc:hive2://> select * from product_name;

OK

product_name.id	product_name.name
broccoli	Broccoli
spinach	Spinach
carrot	Local Carrots
potato	Idaho Potatoes
	NULL
iphone7	iPhone 7
camera_cannon	Canon 570x
washingmachine_samsung	Samsung Swift
tv_vu	VU 56 Inch

9 rows selected (0.268 seconds)

0: jdbc:hive2://> select * from product_cost;

OK

product_cost.id	product_cost.cost
broccoli	5.0

```
| spinach | 7.0 |  
| carrot | 4.0 |  
| potato | 4.0 |  
| | NULL |  
| iphone7 | 950.0 |  
| camera_cannon | 1000.0 |  
| washingmachine_samsung | 400.0 |  
| tv_vu | 600.0 |  
+-----+-----+--+  
9 rows selected (0.236 seconds)
```

OK
No rows affected (0.396 seconds)

```
0: jdbc:hive2://> create table product_cost (id string, cost float);  
OK  
No rows affected (0.151 seconds)
```

merge query

```
0: jdbc:hive2://> from products2  
..... > insert overwrite table product_name  
..... > select id,title  
..... > insert into table product_cost  
..... > select id,cost;
```

delete rows using truncate

```
0: jdbc:hive2://> truncate table freshproducts;
```