

Assignment Submission

*****Section 1*****

Sample data to be loaded in **Hive Tables**

salesman.csv

```
5001,James Hoog>New York,0.15
5002,Nail Knite>Paris,0.13
5005,Pit Alex>London,0.11
5006,Mc Lyon>Paris,0.14
5003,Lauson, Hen,0.12
5007,Paul Adam>Rome,0.13
```

customer.csv

```
3002,Nick Rimando>New York,100,5001
3005,Graham Zusi>California,200,5002
3001,Brad Guzan>London,,5005
3004,Fabian Johns>Paris,300,5006
3007,Brad Davis>New York,200,5001
3009,Geoff Camero>Berlin,100,5003
3008,Julian Green>London,300,5002
3003,Jozy Altidor>Moscow,200,5007
```

orders.csv

```
70001,150.5,2012-10-05,3005,5002  
70009,270.65,2012-09-10,3001,5005  
70002,65.26,2012-10-05,3002,5001  
70004,110.5,2012-08-17,3009,5003  
70007,948.5,2012-09-10,3005,5002  
70005,2400.6,2012-07-27,3007,5001  
70008,5760,2012-09-10,3002,5001  
70010,1983.43,2012-10-10,3004,5006  
70003,2480.4,2012-10-10,3009,5003  
70012,250.45,2012-06-27,3008,5002  
70011,75.29,2012-08-17,3003,5007  
70013,3045.6,2012-04-25,3002,5001
```

Answer :

- creating customer.csv , orders.csv and salesman.csv using touch command in shweta_folder

```
takeo@011970f20bdb:~/shweta_folder$ ls  
bootcamp.csv  customer.csv  department.txt  employee.txt  orders.csv  salesman.csv
```

- inserting data using vi command and viewing the result using cat command

```
takeo@011970f20bdb:~/shweta_folder$ vi salesman.csv  
takeo@011970f20bdb:~/shweta_folder$ cat salesman.csv  
5001,James Hoog,New York,0.15  
5002,Nail Knite,Paris,0.13  
5005,Pit Alex,London,0.11  
5006,Mc Lyon,Paris,0.14  
5003,Lauson, Hen,0.12  
5007,Paul Adam,Rome,0.13
```

```
takeo@011970f20bdb:~/shweta_folder$ cat customer.csv
3002,Nick Rimando>New York,100,5001
3005,Graham Zusi>California,200,5002
3001,Brad Guzan>London,,5005
3004,Fabian Johns>Paris,300,5006
3007,Brad Davis>New York,200,5001
3009,Geoff Camero>Berlin,100,5003
3008,Julian Green>London,300,5002
3003,Jozy Altidore>Moscow,200,5007
```

```
takeo@011970f20bdb:~/shweta_folder$ vi orders.csv
takeo@011970f20bdb:~/shweta_folder$ cat orders.csv
70001,150.5,2012-10-05,3005,5002
70009,270.65,2012-09-10,3001,5005
70002,65.26,2012-10-05,3002,5001
70004,110.5,2012-08-17,3009,5003
70007,948.5,2012-09-10,3005,5002
70005,2400.6,2012-07-27,3007,5001
70008,5760,2012-09-10,3002,5001
70010,1983.43,2012-10-10,3004,5006
70003,2480.4,2012-10-10,3009,5003
70012,250.45,2012-06-27,3008,5002
70011,75.29,2012-08-17,3003,5007
70013,3045.6,2012-04-25,3002,5001
```

Hive Tables Metadata

salesman

```
salesman_id int,
name string,
city string,
commission double
```

customer

```
customer_id int,  
cust_name string,  
city string,  
grade int,  
salesman_id int
```

orders

```
ord_no int,  
purch_amt double,  
ord_date date,  
customer_id int,  
salesman_id int
```

Create a database named **hive_test** and create three tables

salesman

customer

Orders

Answer:

– creating the database

```
[hive (default)> create database hive_test;  
OK  
Time taken: 1.273 seconds  
[hive (default)> show databases;  
OK  
default  
demo  
demotest  
hive_test  
xyz
```

– creating and loading data into table : salesman

```
[hive (default)> use hive_test;
OK
Time taken: 0.087 seconds
hive (hive_test)> Create table salesman ( salesman_id int , name string , city string , commission double) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
OK
Time taken: 0.111 seconds
hive (hive_test)> LOAD DATA LOCAL INPATH '/home/takeo/shweta_folder/salesman.csv' INTO TABLE salesman;
Loading data to table hive_test.salesman
OK
Time taken: 0.165 seconds
hive (hive_test)> select * from salesman;
OK
5001  James Hoog      New York      0.15
5002  Nail Knite     Paris 0.13
5005  Pit Alex       London 0.11
5006  Mc Lyon Paris 0.14
5003  Lauson Hen    0.12
5007  Paul Adam      Rome 0.13
Time taken: 0.202 seconds, Fetched: 6 row(s)
```

– creating and loading data into table customer:

```
[hive (hive_test)> create table customer (customer_id int , cust_name string, city string, grade int, salesman_id int) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
OK
Time taken: 0.123 seconds
hive (hive_test)> LOAD DATA LOCAL INPATH '/home/takeo/shweta_folder/customer.csv' INTO TABLE customer;
Loading data to table hive_test.customer
OK
Time taken: 0.233 seconds
hive (hive_test)> select * from customer;
OK
3002  Nick Rimando  New York   100  5001
3005  Graham Zusi   California 200  5002
3001  Brad Guzan    London    NULL 5005
3004  Fabian Johns  Paris    300  5006
3007  Brad Davis    New York   200  5001
3009  Geoff Camero  Berlin   100  5003
3008  Julian Green   London   300  5002
3003  Jozy Altidor  Moscow   200  5007
Time taken: 0.126 seconds, Fetched: 8 row(s)
```

– creating and loading data into table orders:

```
[hive (hive_test)> create table orders (ord_no int, purch_amt double, ord_date date, customer_id int, salesman_id int) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
OK
Time taken: 0.136 seconds
hive (hive_test)> LOAD DATA LOCAL INPATH '/home/takeo/shweta_folder/orders.csv' INTO TABLE orders;
Loading data to table hive_test.orders
OK
Time taken: 0.618 seconds
hive (hive_test)> select * from orders;
OK
70001  150.5  2012-10-05  3005  5002
70009  270.65 2012-09-10  3001  5005
70002  65.26  2012-10-05  3002  5001
70004  110.5  2012-08-17  3009  5003
70007  948.5   2012-09-10  3005  5002
70005  2400.6  2012-07-27  3007  5001
70008  5760.0  2012-09-10  3002  5001
70010  1983.43 2012-10-10  3004  5006
70003  2480.4  2012-10-10  3009  5003
70012  258.45  2012-06-27  3008  5002
70011  75.29   2012-08-17  3003  5007
70013  3045.6  2012-04-25  3002  5001
Time taken: 0.101 seconds, Fetched: 12 row(s)
```

Create above tables with the help of given data and metadata.

*****Write SQL For Followings*****

- Write a SQL statement to prepare a list with salesman name, customer name and their cities for the salesmen and customer who belongs to the same city.

Select s.name as salesman_name , c.cust_name as customer_name, c.city as city from salesman s join customer c on s.city = c.city order by salesman_name;

```
hive (hive_test)> SET hive.exec.mode.local.auto=true;
hive (hive_test)> Select s.name as salesman_name , c.cust_name as customer_name, c.city as city from salesman s join customer c on s.city = c.city order by salesman_name;
Automatically selecting local only mode for query
Query ID = takeo_20251005190649_141502a-76eb-4510-885a-80c736f0816d
Total jobs = 1
SLF4J: Found binding in [jar:file:/home/takeo/bootcamp/installshive/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/takeo/bootcamp/installshive/hadoop-3.3.3/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.
2025-10-05 19:06:52 Starting to launch local task to process map join; maximum memory = 239075328
2025-10-05 19:06:53 Uploaded 1 File to: file:/tmp/takeo/abd4bb14-2842-48da-92f2-12edeff55381/hive_2025-10-05_19-06-49_419_3267241577321256126-1-local-18005/HashTable-Stage-2/Map
hashtable (443 bytes)
Execution completed successfully
MapredLocal task succeeded
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>
Job running in-process (local Hadoop)
2025-10-05 19:06:55,318 Stage-2 map = 100%, reduce = 100%
Ended Job = job_local169349284_0001
MapReduce Jobs Launched:
Stage-Stage-2: HDFS Read: 2868 HDFS Write: 84520720 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
James Hoog      Brad Davis      New York
James Hoog      Nick Rimando    New York
Mc Lyon Fabian Johns Paris
Nail Knite     Fabian Johns Paris
Pit Alex       Julian Green   London
Pit Alex       Brad Guzan    London
Time taken: 5.949 seconds, Fetched: 6 row(s)
```

- Write a SQL statement to know which salesman are working for which customer.

Select c.customer_id , c.cust_name, s.name from customer c join salesman s on c.salesman_id = s.salesman_id order by s.name;

```
hive (hive_test)> Select c.customer_id , c.cust_name, s.name from customer c join salesman s on c.salesman_id = s.salesman_id order by s.name;
Automatically selecting local only mode for query
Query ID = takeo_20251005191303_9c134256-58d5-49ac-8f2c-cbd247d43ff7
Total jobs = 1
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type org.apache.logging.slf4j.Log4jLoggerFactory
2025-10-05 19:13:07 Dump the side-table for tag: i with group count: 6 into file: file:/tmp/takeo/abd4bb14-2842-48da-92f2-12edeff55381/hive_2025-10-05_19-13-03_452_440
005/HashTable-Stage-2/MapJoin-mapfile21--.hashtable
2025-10-05 19:13:07 End of local task; Time Taken: 1.11 sec.
Execution completed successfully
MapredLocal task succeeded
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>
Job running in-process (local Hadoop)
2025-10-05 19:13:09,628 Stage-2 map = 100%, reduce = 100%
Ended Job = job_local785859352_0002
MapReduce Jobs Launched:
Stage-Stage-2: HDFS Read: 4082 HDFS Write: 84521456 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
3007      Brad Davis      James Hoog
3002      Nick Rimando    James Hoog
3009      Geoff Camero    Lauson
3004      Fabian Johns    Mc Lyon
3008      Julian Green    Nail Knite
3005      Graham Zusi     Nail Knite
3003      Jozy Altidore   Paul Adam
3001      Brad Guzan     Pit Alex
Time taken: 6.244 seconds, Fetched: 8 row(s)
```

- Write a SQL statement to make a list with order no, purchase amount, customer name and their cities for those orders which order amount between 500 and 2000.

Select o.ord_no, o.purch_amt, c.cust_name, c.city from orders o join customer c on o.customer_id = c.customer_id where purch_amt between 500 and 2000 order by o.purch_amt;

```

hive (hive_test)> Select o.ord_no, o.purch_amt, c.cust_name, c.city from orders o join customer c on o.customer_id = c.customer_id where purch_amt between 500 and 2000 order by o.purch_amt;
Automatically selecting local only mode for query
Query ID = takeo_20251005192302_45ee835c-3e9b-4244-ba20-7cd8be2062d9
Total jobs: 1
SLF4J: Found binding in [jar:file:/home/takeo/bootcamp/installshive/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
2025-10-05 19:23:06   Dump the side-table for tag: i with group count: 8 into file: file:/tmp/takeo/ab4bb14-2842-48da-92f2-12edeff55381/hive_2025-10-05_19-23-02_792_8831730152285816275-1/-lo
005/HashTable-Stage-2/MapJoin-mapfile41--.hashtable
2025-10-05 19:23:06   Uploaded 1 File to: file:/tmp/takeo/ab4bb14-2842-48da-92f2-12edeff55381/hive_2025-10-05_19-23-02_792_8831730152285816275-1/-local-10005/HashTable-Stage-2/MapJoin-mapfil
hashtable (593 bytes)
Execution completed successfully
MapredLocal task succeeded
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>
Job running in-process (local Hadoop)
2025-10-05 19:23:08,496 Stage-2 map = 100%,  reduce = 100%
Ended Job = job_local10005598469_0004
MapReduce Jobs Launched:
Stage-Stage-2:  HDFS Read: 6848 HDFS Write: 84522393 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
78007 948.5 Graham Zusi    California
78010 1983.43 Fabian Johns  Paris
Time taken: 5.757 seconds, Fetched: 2 rows(s)

```

- Write a SQL statement to find the list of customers who appointed a salesman for their jobs who gets a commission from the company is more than 12%.

Select c.customer_id, c.cust_name,s.name,s.commission from customer c join salesman s on c.salesman_id = s.salesman_id where s.commission > 0.12 order by s.commission;

```

hive (hive_test)> Select c.customer_id, c.cust_name,s.name,s.commission from customer c join salesman s on c.salesman_id = s.salesman_id where s.commission > 0.12 order by s.commission;
Automatically selecting local only mode for query
Query ID = takeo_20251005193513_760831d9-17fe-4c47-8200-ed1b5526932
Total jobs: 1
SLF4J: Found binding in [jar:file:/home/takeo/bootcamp/installshive/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
2025-10-05 19:35:18   Dump the side-table for tag: i with group count: 4 into file: file:/tmp/takeo/ab4bb14-2842-48da-92f2-12edeff55381/hive_2025-10-05_19-35-13_048_516735646008590605-1/-lo
005/HashTable-Stage-2/MapJoin-mapfile51--.hashtable
2025-10-05 19:35:18   Uploaded 1 File to: file:/tmp/takeo/ab4bb14-2842-48da-92f2-12edeff55381/hive_2025-10-05_19-35-13_048_516735646008590605-1/-local-10005/HashTable-Stage-2/MapJoin-mapfil
ashtable (416 bytes)
2025-10-05 19:35:18   End of local task; Time Taken: 1.071 sec.
Execution completed successfully
MapredLocal task succeeded
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>
Job running in-process (local Hadoop)
2025-10-05 19:35:19,528 Stage-2 map = 100%,  reduce = 100%
Ended Job = job_local10005620006_0005
MapReduce Jobs Launched:
Stage-Stage-2:  HDFS Read: 8056 HDFS Write: 84522928 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
3003 Jozy Altidor  Paul Adam    0.13
3008 Julian Green  Nail Knite   0.13
3005 Graham Zusi   Nail Knite   0.13
3004 Fabian Johns  Mc Lyon 0.14
3007 Brad Davis     James Hoog    0.15
3002 Nick Rimando   James Hoog    0.15
Time taken: 6.534 seconds, Fetched: 6 row(s)

```

- Write a SQL statement to find the list of customers who appointed a salesman for their jobs who does not live in the same city where their customer lives, and gets a commission above 12% .

Select c.customer_id, c.cust_name,c.city,s.name,s.commission from customer c join salesman s on c.salesman_id = s.salesman_id where c.city<>s.city and s.commission > 0.12 order by s.commission;

```

hive (hive_test)> Select c.customer_id, c.cust_name,c.city,s.name,s.commission from customer c join salesman s on c.salesman_id = s.salesman_id where c.city<>s.city and s.commission > 0.12 order by
s.commission;
Automatically selecting local only mode for query
Output to takeo_20251005193846_74bd85b6-cd29-41c9-91b7-9e52cdf295d9
Total jobs: 1
SLF4J: Found binding in [jar:file:/home/takeo/bootcamp/installshive/apache-hive-3.1.3-bin/lib@Slf4j-slf4j-impl-2.17.1.jar!/org/slf4j/jimpl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/takeo/bootcamp/installshive/hadoop-3.3/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/jimpl/StaticLoggerBinder.class]
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
2025-10-05 19:38:49      Starting to launch local task to process map join;           maximum memory = 239075328
2025-10-05 19:38:50      Uploaded 1 file to: /file:/tmp/takeo/ab4bb14-2842-48da-92f2-12edeff56381/hive_2025-10-05_19-38-46_593_7886246583723647997-1/_local-10005/HashTable-Stage-2/MapJoin-mapfile61--.
MapJoin task launched.
Execution completed successfully
MapReduce task succeeded
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 limit the minimum number of reducers:
  set napreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2025-10-05 19:39:52.345 Stage-2 map = 100%,  reduce = 100%
Ended Job = job_local1875437673_0006
MapReduce Jobs Launched:
Stage-Stage-2:  HDFS Read: 9310  HDFS Write: 84623532 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
0002  Jazy Altidor  Moscow  Paul Adam    0.13
3000  Julian Green  London  Neil Knite    0.13
3005  Graham Zusi  California  Neil Knite    0.13
Time taken: 5.799 seconds, Fetched: 3 row(s)

```

Please submit sql's for above five questions as your assignment answers.

*****Section 2*****

1. Please complete the below create table commands on complex data and create the table in hive.

Replace ? with appropriate delimiter according to the given data in below create table sql

```

salesdetail_complex (Product_ID INT,productdetails map<String,String>,Order_Priority
VARCHAR(4),merchantType CHAR(4),Sale_Amount DOUBLE,Order_Quantity
BIGINT,Discount FLOAT,Salaryhike TINYINT,companyprofit SMALLINT,
financeDeficit DECIMAL(8,2),indian BOOLEAN,saledate array<date>,saleyear
array<int>,selleramountfile array<DOUBLE>,orderQuantityfile array<BIGINT>,costlist
map<int,int>,strutureType struct<city:string,state:string,pin:bigint>,systemdatetime array<String>)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '?' collection items terminated by '?'
map keys terminated by '?';

```

And load below sample data in above table

1,Cameras#Cameras#Cameras,Medium,Seller,750000,5000000,1500.24,100,1000,1500.659744,TRUE,2012-12-21\$2013-12-26,2012\$1998,750700.00\$850000.01,500000065\$500056458,401#901\$1200#5410,ap\$mp\$500001,2019-12-21\$2020-12-26 12:00
 2,Cameras#Nikon\$DLR#SmileDetector,Not Specified,Dealer,750001,5000561,1501.24,101,1001,1501.659744,2012-12-22\$2012-12-27,2012\$1999,750001.00\$750000.02,500056187\$500056458,,ap\$mp\$500002,2019-12-21\$2020-12-26 12:01
 3,,High,Seller,750002,5000562,1502.24,105,1002,1502.659744,2012-12-23\$2012-12-28,2012\$2000,750002.00\$780000.03,500056245\$500056458,400#902\$1200#5413,ap\$mp\$500003,2019-12-21\$2020-12-26 12:02
 4,Accessories#PentaLite\$TV Accessories#Wall Mount,Critical,Dealer,750003,5000563,1503.24,106,1003,1503.659744,2012-12-24\$2012-12-29,2012\$2001,750003.00\$790000.04,500056345\$500056458,400#903\$1200#5414,ap\$mp\$500004,2019-12-21\$2020-12-26 12:03
 5,Camera1s#Timbre\$Video#Lens Kit,Low,Dealer,750004,5000564,1504.24,108,1004,1504.659744,2012-12-25\$2012-12-30,2012\$2002,750004.00\$800000.05,500056458\$500056458,400#904\$1200#5415,ap\$mp\$500005,2019-12-21\$2020-12-26 12:04

Answer:

– Creating file and loading in docker

```

takeo@011970f20bdb:~/shweta_folder$ vi salesdetail_complex.csv
takeo@011970f20bdb:~/shweta_folder$ cat salesdetail_complex.csv
1,Cameras#Cameras#Cameras,Medium,Seller,750000,5000000,1500.24,100,1000,1500.659744,TRUE,2012-12-21$2013-12-26,2012$1998,750700.00$850000.01,500000065$500056458,401#901$1200#5410,ap$mp$500001,2019-12-21$2020-12-26 12:00
2,Cameras#Nikon$DLR#SmileDetector,Not Specified,Dealer,750001,5000561,1501.24,101,1001,1501.659744,2012-12-22$2012-12-27,2012$1999,750001.00$750000.02,500056187$500056458,,ap$mp$500002,2019-12-21$2020-12-26 12:01
3,,High,Seller,750002,5000562,1502.24,105,1002,1502.659744,2012-12-23$2012-12-28,2012$2000,750002.00$780000.03,500056245$500056458,400#902$1200#5413,ap$mp$500003,2019-12-21$2020-12-26 12:02
4,Accessories#PentaLite$TV Accessories#Wall Mount,Critical,Dealer,750003,5000563,1503.24,106,1003,1503.659744,2012-12-24$2012-12-29,2012$2001,750003.00$790000.04,500056345$500056458,400#903$1200#5414,ap$mp$500004,2019-12-21$2020-12-26 12:03
5,Camera1s#Timbre$Video#Lens Kit,Low,Dealer,750004,5000564,1504.24,108,1004,1504.659744,2012-12-25$2012-12-30,2012$2002,750004.00$800000.05,500056458$500056458,400#904$1200#5415,ap$mp$500005,2019-12-21$2020-12-26 12:04
  
```

– Creating table and pulling data from docker in hive

```

hive (hive_test)> CREATE TABLE salesdetail_complex (
>   ProductCategory STRING,
>   ProductSubCategory MAP<STRING,STRING>,
>   Order_Priority VARCHAR(20),
>   merchantType CHAR(4),
>   Sale_Amount DOUBLE,
>   Order_Quantity BIGINT,
>   Discount FLOAT,
>   Discounthike TINYINT,
>   companyprofit SMALLINT,
>   itemOfferPrice DECIMAL(8,2),
>   Indian BOOLEAN,
>   saledate ARRAY<DATE>,
>   salyear ARRAY<INT>,
>   selleramountfile ARRAY<DOUBLE>,
>   orderQuantityfile ARRAY<BIGINT>,
>   costlist MAP<INT,INT>,
>   struturetype STRUCT<city:STRING,state:STRING,pin:BIGINT>
>   ,systemdatetime ARRAY<STRING>
> );
> ROW FORMAT DELIMITED
> FIELDS TERMINATED BY ''
> COLLECTION ITEMS TERMINATED BY '$'
> MAP KEYS TERMINATED BY '#';
OK
Time taken: 0.431 seconds
hive (hive_test)> LOAD DATA LOCAL INPATH '/home/takeo/shweta_folder/salesdetail_complex.csv' into TABLE salesdetail_complex;
Loading data to table hive_test.salesdetail_complex
OK
Time taken: 1.194 seconds
hive (hive_test)> select * from salesdetail_complex;
OK
1   {"Cameras": "Cameras", "Cameras": "Cameras"}   Medium Sell 750000.0 5000000 1500.24 100 1000 1500.659744 true ["2012-12-21", "2013-12-26"] [2012, 1998] [750700.0, 850000.01]
0.01] [50000065, 500056458] {401#901, 1200#5410} {"city": "ap", "state": "mp", "pin": 500001} ["2019-12-21", "2020-12-26 12:00"]
2   {"Cameras": "Nikon", "DLR": "SmileDetector"} Not Specified Deal 750001.0 5000561 1501.24 101 1001 1501.66 false ["2012-12-22", "2012-12-27"] [2012, 1999] [750000.02, 500056187$500056458,,]
0.750000.02] [500056197, 500056458] {} {"city": "ap", "state": "mp", "pin": 500002} ["2019-12-21", "2020-12-26 12:01"]
3   High Sell 750002.0 5000562 1502.24 105 1002 1502.66 false ["2012-12-23", "2012-12-28"] [2012, 2000] [750002.0, 780000.03] [500056245, 500056458] {400:90}
2,1200#5413] [{"city": "ap", "state": "mp", "pin": 500003} ["2019-12-21", "2020-12-26 12:02"]
4   {"Accessories": "PentaLite", "TV Accessories": "Wall Mount"} Critical Deal 750003.0 5000563 1503.24 106 1003 1503.66 false ["2012-12-24", "2012-12-29"] [2012, 2001]
0.01] [750003.0, 790000.04] [500056345, 500056458] {400#902, 1200#5414} {"city": "ap", "state": "mp", "pin": 500004} ["2019-12-21", "2020-12-26 12:03"]
5   {"Cameras": "Timbre", "Video": "Lens Kit"} Low Deal 750004.0 5000564 1504.24 108 1004 1504.66 false ["2012-12-25", "2012-12-30"] [2012, 2002] [750004.0, 800000.05]
0.05] [500056458, 500056458] {400#904, 1200#5415} {"city": "ap", "state": "mp", "pin": 500005} ["2019-12-21", "2020-12-26 12:04"]
Time taken: 0.294 seconds, Fetched: 5 row(s)
  
```

2. Write a sql to get the only 2 records from salesdetail_complex

```
select * from salesdetail_complex limit 2;
```

```
hive (hive_test)> select * from salesdetail_complex limit 2;
OK
1   {"Cameras": "Cameras", "Cameras": "Cameras"}      Medium Sell    750000.0     5000000 1500.24 100    1000    1500.66 true    ["2012-12-21", "2013-12-26"]    [2012,1998]    [750700.0,85000
0.01]  [500000065,500056458]  {{"city": "ap", "state": "mp", "pin": 500001} ["2019-12-21", "2020-12-26 12:00"]
2   {"Cameras": "Nikon", "DLR": "SmileDetector"} Not Specified Deal    750001.0     5000561 1501.24 101    1001    1501.66 false   ["2012-12-22", "2012-12-27"]    [2012,1999]    [750001
.0,750000.02]  [500056187,500056458]  {}    {"city": "ap", "state": "mp", "pin": 500002} ["2019-12-21", "2020-12-26 12:01"]
Time taken: 0.288 seconds. Fetched: 2 row(s)
```

3. Create a non **partitioned table** named as **non_part** on below data.

Table metadata for non partitioned table:

Table columns

```
dateid smallint ,
caldate date ,
day string ,
week smallint ,
month string ,
qtr string ,
year smallint ,
holiday boolean
```

Sample data from above table.

```
1827|2008-01-01|WE|1|JAN|1|2008|TRUE
1828|2008-01-02|TH|1|JAN|1|2008|FALSE
1829|2008-01-03|FR|1|JAN|1|2008|FALSE
1830|2008-01-04|SA|2|JAN|1|2008|FALSE
1831|2008-01-05|SU|2|JAN|1|2008|FALSE
1832|2008-01-06|MO|2|JAN|1|2008|FALSE
1833|2008-01-07|TU|2|JAN|1|2008|FALSE
1834|2008-01-08|WE|2|JAN|1|2008|FALSE
1835|2008-01-09|TH|2|JAN|1|2008|FALSE
1836|2008-01-10|FR|2|JAN|1|2008|FALSE
1837|2008-01-11|SA|3|JAN|1|2008|FALSE
1838|2008-01-12|SU|3|JAN|1|2008|FALSE
1839|2008-01-13|MO|3|JAN|1|2008|FALSE
1840|2008-01-14|TU|3|JAN|1|2008|FALSE
1841|2008-01-15|WE|3|JAN|1|2008|FALSE
1842|2008-01-16|TH|3|JAN|1|2008|FALSE
1843|2008-01-17|FR|3|JAN|1|2008|FALSE
```

1844 2008-01-18 SA 4 JAN 1 2008 FALSE
1845 2008-01-19 SU 4 JAN 1 2008 FALSE
1846 2008-01-20 MO 4 JAN 1 2008 FALSE
1847 2008-01-21 TU 4 JAN 1 2008 FALSE
1848 2008-01-22 WE 4 JAN 1 2008 FALSE
1849 2008-01-23 TH 4 JAN 1 2008 FALSE
1850 2008-01-24 FR 4 JAN 1 2008 FALSE
1851 2008-01-25 SA 5 JAN 1 2008 FALSE
1852 2008-01-26 SU 5 JAN 1 2008 FALSE

Answer:

```

hive (hive_test)>
    > CREATE TABLE non_part (
    >     datedid SMALLINT,
    >     caldate DATE,
    >     day STRING,
    >     week SMALLINT,
    >     month STRING,
    >     qtr STRING,
    >     year SMALLINT,
    >     holiday BOOLEAN
    > )
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '|';
OK
Time taken: 0.2 seconds
hive (hive_test)> LOAD DATA LOCAL INPATH '/home/takeo/shweta_folder/non_part.txt' into TABLE non_part;
Loading data to table hive_test.non_part
OK
Time taken: 0.798 seconds
hive (hive_test)> select * from non_part;
OK
1827 2008-01-01 WE 1 JAN 1 2008 true
1828 2008-01-02 TH 1 JAN 1 2008 false
1829 2008-01-03 FR 1 JAN 1 2008 false
1830 2008-01-04 SA 2 JAN 1 2008 false
1831 2008-01-05 SU 2 JAN 1 2008 false
1832 2008-01-06 MO 2 JAN 1 2008 false
1833 2008-01-07 TU 2 JAN 1 2008 false
1834 2008-01-08 WE 2 JAN 1 2008 false
1835 2008-01-09 TH 2 JAN 1 2008 false
1836 2008-01-10 FR 2 JAN 1 2008 false
1837 2008-01-11 SA 3 JAN 1 2008 false
1838 2008-01-12 SU 3 JAN 1 2008 false
1839 2008-01-13 MO 3 JAN 1 2008 false
1840 2008-01-14 TU 3 JAN 1 2008 false
1841 2008-01-15 WE 3 JAN 1 2008 false
1842 2008-01-16 TH 3 JAN 1 2008 false
1843 2008-01-17 FR 3 JAN 1 2008 false
1844 2008-01-18 SA 4 JAN 1 2008 false
1845 2008-01-19 SU 4 JAN 1 2008 false
1846 2008-01-20 MO 4 JAN 1 2008 false
1847 2008-01-21 TU 4 JAN 1 2008 false
1848 2008-01-22 WE 4 JAN 1 2008 false
1849 2008-01-23 TH 4 JAN 1 2008 false
1850 2008-01-24 FR 4 JAN 1 2008 false
1851 2008-01-25 SA 5 JAN 1 2008 false
1852 2008-01-26 SU 5 JAN 1 2008 false
Time taken: 0.208 seconds, Fetched: 26 row(s)

```

4. Now create a partitioned table on column caldate named as date_part from non partitioned table created above

Partitioned table metadata

Table Columns

```
dateid smallint,  
       day string ,  
       week smallint ,  
       month string ,  
       qtr string ,  
       year smallint ,  
       holiday boolean
```

Partition column

```
caldate
```

```
hive (hive_test)> CREATE TABLE date_part (  
    >     dateid SMALLINT,  
    >     day STRING,  
    >     week SMALLINT,  
    >     month STRING,  
    >     qtr STRING,  
    >     year SMALLINT,  
    >     holiday BOOLEAN  
    > )  
    > PARTITIONED BY (caldate DATE)  
    > ROW FORMAT DELIMITED  
    > FIELDS TERMINATED BY '|'  
    > STORED AS TEXTFILE;  
OK
```

– Loading data from previously non partitioned data into partitioned table

```
hive (hive_test)> LOAD DATA LOCAL INPATH '/home/takeo/shweta_folder/non_part.txt' into table date_part;  
Automatically selecting local only mode for query  
Query ID = takeo_20251005210715_f38ffb72-b4a3-4789-9a07-f3820708239d  
Total jobs = 3
```

5. Now create a partitioned and bucketed table on above table

Table Columns

```
dateid smallint,  
       week smallint ,  
       month string ,  
       qtr string ,  
       year smallint ,  
       holiday boolean
```

Partition column

```
caldate
```

Bucket column

```
dateid
```

```
hive (hive_test)> CREATE TABLE datepart_bucketed (
>     dateid SMALLINT,
>     week SMALLINT,
>     month STRING,
>     qtr STRING,
>     year SMALLINT,
>     holiday BOOLEAN
> )
> PARTITIONED BY (caldate DATE)
> CLUSTERED BY (dateid) INTO 4 BUCKETS
> ROW FORMAT DELIMITED
> FIELDS TERMINATED BY '|';
OK
Time taken: 0.168 seconds
```