\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***Section 1**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Run this before running hive commands:

set mapreduce.framework.name=local;

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 |

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**

CREATE DATABASE hive\_test;

CREATE TABLE salesman (

salesman\_id INT,

name STRING,

city STRING,

commission DOUBLE

)

CREATE TABLE customer (

customer\_id INT,

cust\_name STRING,

city STRING,

grade INT,

salesman\_id INT )

CREATE TABLE orders (

ord\_no INT,

purch\_amt DOUBLE,

ord\_date DATE,

customer\_id INT,

salesman\_id INT)

***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.

Solution:

SELECT s.name, c.cust\_name, c.city

FROM

salesman s

JOIN

customer c

ON s.city = c.city;

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

Solution:

SELECT c.cust\_name, s.name

FROM customer c

JOIN salesman s

ON c.salesman\_id = s.salesman\_id;

* 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.

Solution :

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

o.purch\_amt BETWEEN 500 AND 2000;

* 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%.

Solution:

SELECT 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;

* 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% .

Solution:

SELECT c.cust\_name , s.name,s.city, c.city, s.commission

FROM customer c

JOIN salesman s

ON c.salesman\_id = s.salesman\_id

WHERE s.city <> c.city

AND s.commission > 0.12;

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**

🡪CREATE TABLE 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 '\u0001'

COLLECTION ITEMS TERMINATED BY '\u0002'

MAP KEYS TERMINATED BY '\u0003'

STORED AS TEXTFILE;

|  |
| --- |
| 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**

**🡪**LOAD DATA LOCAL INPATH '/home/text/test/complex\_data.txt' INTO TABLE salesdetail\_complex;

|  |
| --- |
| 1,Cam1eras#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$2020-12-26 12:00  2,Cam`eras#Nikon$DLR#SmileDetector,Not Specified,Dealer,750001,5000561,1501.24,101,1001,1501.659744,FALSE,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,FALSE,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,FALSE,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,FALSE,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 |

**2. Write a sql to get the only 2 records from** salesdetail\_complex

🡪 SELECT \* FROM salesdetail\_complex LIMIT 2;

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

Table metadata for non partitioned table:

🡪CREATE TABLE non\_partitioned(

dateid SMALLINT,

caldate DATE,

day STRING,

week SMALLINT,

month STRING,

qtr STRING,

year SMALLINT,

holiday BOOLEAN

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE;

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 |

**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 |

🡪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;

**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 |

🡪CREATE TABLE date\_part\_bucketed (

dateid SMALLINT,

week SMALLINT,

month STRING,

qtr STRING,

year SMALLINT,

holiday BOOLEAN

)

PARTITIONED BY (caldate DATE)

CLUSTERED BY (dateid) INTO 10 BUCKETS

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE;