PROPERTIES:

set hive.exec.dynamic.partition=true;

set hive.exec.dynamic.partition.mode=nonstrict;

TABLES:

CREATE TABLE `donations`(

`accountid` string COMMENT 'from deserializer',

`city` string COMMENT 'from deserializer',

`givings` string COMMENT 'from deserializer',

`avggivings` string COMMENT 'from deserializer',

`donorlevel` string COMMENT 'from deserializer')

PARTITIONED BY (

`year` int)

CLUSTERED BY (

accountid)

SORTED BY (accountid)

INTO 256 BUCKETS

ROW FORMAT SERDE

'org.apache.hadoop.hive.serde2.OpenCSVSerde'

WITH SERDEPROPERTIES (

'quoteChar'='\"',

'separatorChar'=',')

STORED AS INPUTFORMAT

'org.apache.hadoop.mapred.TextInputFormat'

OUTPUTFORMAT

'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'

LOCATION

'hdfs://localhost:9000/user/hive/warehouse/donations'

TBLPROPERTIES (

'skip.header.line.count'='1',

'transient\_lastDdlTime'='1517588204')

CREATE TABLE `tmp1`(

`accountid` string COMMENT 'from deserializer',

`city` string COMMENT 'from deserializer',

`year` string COMMENT 'from deserializer',

`givings` string COMMENT 'from deserializer',

`avggivings` string COMMENT 'from deserializer',

`donlevel` string COMMENT 'from deserializer')

ROW FORMAT SERDE

'org.apache.hadoop.hive.serde2.OpenCSVSerde'

WITH SERDEPROPERTIES (

'quoteChar'='\"',

'separatorChar'=',')

STORED AS INPUTFORMAT

'org.apache.hadoop.mapred.TextInputFormat'

OUTPUTFORMAT

'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'

LOCATION

'hdfs://localhost:9000/user/hive/warehouse/donations'

TBLPROPERTIES (

'COLUMN\_STATS\_ACCURATE'='true',

'numFiles'='0',

'numRows'='0',

'rawDataSize'='0',

'skip.header.line.count'='1',

'totalSize'='0',

'transient\_lastDdlTime'='1518264451')

load data local inpath '/home/student/share/project/FTDonations.csv' overwrite into table tmp1;

insert overwrite table donations PARTITION (year) select accountid,city,givings,avggivings,donlevel,year from tmp1

CREATE TABLE `accounts`(

`accountid` string COMMENT 'from deserializer',

`accountnum` string COMMENT 'from deserializer',

`subtype` string COMMENT 'from deserializer',

`type` string COMMENT 'from deserializer',

`assignedto` string COMMENT 'from deserializer',

`ismajor` string COMMENT 'from deserializer')

CLUSTERED BY (

accountid)

SORTED BY (accountid)

INTO 256 BUCKETS

ROW FORMAT SERDE

'org.apache.hadoop.hive.serde2.OpenCSVSerde'

WITH SERDEPROPERTIES (

'quoteChar'='\"',

'separatorChar'=',')

STORED AS INPUTFORMAT

'org.apache.hadoop.mapred.TextInputFormat'

OUTPUTFORMAT

'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'

LOCATION

'hdfs://localhost:9000/user/hive/warehouse/accounts'

TBLPROPERTIES (

'COLUMN\_STATS\_ACCURATE'='true',

'numFiles'='1',

'numRows'='0',

'rawDataSize'='0',

'skip.header.line.count'='1',

'totalSize'='3999550',

'transient\_lastDdlTime'='1517585808');

load data local inpath '/home/student/share/project/DTAccounts.csv' overwrite into table accounts;

CREATE TABLE `cities`(

`city` string COMMENT 'from deserializer',

`postcode` string COMMENT 'from deserializer')

PARTITIONED BY (

`country` string,

`state` string)

CLUSTERED BY (

city)

INTO 32 BUCKETS

ROW FORMAT SERDE

'org.apache.hadoop.hive.serde2.OpenCSVSerde'

WITH SERDEPROPERTIES (

'quoteChar'='\"',

'separatorChar'=',')

STORED AS INPUTFORMAT

'org.apache.hadoop.mapred.TextInputFormat'

OUTPUTFORMAT

'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'

LOCATION

'hdfs://localhost:9000/user/hive/warehouse/cities'

TBLPROPERTIES (

'skip.header.line.count'='1',

'transient\_lastDdlTime'='1517598895');

CREATE TABLE `tmp2`(

`city` string COMMENT 'from deserializer',

`country` string COMMENT 'from deserializer',

`postcode` string COMMENT 'from deserializer',

`gstate` string COMMENT 'from deserializer',)

ROW FORMAT SERDE

'org.apache.hadoop.hive.serde2.OpenCSVSerde'

WITH SERDEPROPERTIES (

'quoteChar'='\"',

'separatorChar'=',')

STORED AS INPUTFORMAT

'org.apache.hadoop.mapred.TextInputFormat'

OUTPUTFORMAT

'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'

LOCATION

'hdfs://localhost:9000/user/hive/warehouse/donations'

TBLPROPERTIES (

'COLUMN\_STATS\_ACCURATE'='true',

'numFiles'='0',

'numRows'='0',

'rawDataSize'='0',

'skip.header.line.count'='1',

'totalSize'='0',

'transient\_lastDdlTime'='1518264451')

load data local inpath '/home/student/share/project/DTCity.csv' overwrite into table tmp2;

insert overwrite table donations PARTITION (country,state) select city,postcode,country,state from tmp2;

CREATE TABLE `donorlevels`(

`donorlevel` string COMMENT 'from deserializer',

`minv` string COMMENT 'from deserializer',

`maxv` string COMMENT 'from deserializer')

ROW FORMAT SERDE

'org.apache.hadoop.hive.serde2.OpenCSVSerde'

WITH SERDEPROPERTIES (

'quoteChar'='\"',

'separatorChar'=',')

STORED AS INPUTFORMAT

'org.apache.hadoop.mapred.TextInputFormat'

OUTPUTFORMAT

'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'

LOCATION

'hdfs://localhost:9000/user/hive/warehouse/donorlevels'

TBLPROPERTIES (

'COLUMN\_STATS\_ACCURATE'='true',

'numFiles'='1',

'numRows'='0',

'rawDataSize'='0',

'skip.header.line.count'='1',

'totalSize'='120',

'transient\_lastDdlTime'='1517586027');

load data local inpath '/home/student/share/project/DonorLevels.csv' overwrite into table donorlevels;

VIEWS:

CREATE VIEW lvlyear as select donations.donlevel,donations.year,sum(donations.givings) as givings, avg(donations.avggivings) as avggivings from donations group by donations.donlevel, donations.year;

CREATE VIEW nameyear as select accounts.assignedto,donations.year,sum(donations.givings) as givings, avg(donations.avggivings) as avggivings from donations,accounts where donations.accountid = accounts.accountid group by accounts.assignedto,donations.year;

CREATE VIEW stat as select cities.state,cities.country,sum(donations.givings) as givings, avg(donations.avggivings) as avggivings from donations,cities where donations.city = cities.city group by cities.state,cities.country;

QUERIES:

Q1. select assignedto,year,givings from nameyear group by assignedto,year,givings order by assignedto,year;

Q2. select donlevel,year,givings from lvlyear order by donlevel,year;

Q3. select assignedto,year,givings, SUM(givings) over (partition by assignedto) as totperacc, givings/SUM(givings) over (partition by assignedto) as perc from nameyear group by assignedto,year,givings order by assignedto,year;

Q4. select year,sum(givings) as tot from donations group by year order by tot desc;

Q5. select state,country,givings from stat group by state,country,givings order by givings desc limit 10;

Q6. Select assignedto,year,givings, SUM(givings) over (partition by assignedto order by year rows unbounded preceding) as cumtot From nameyear Group by assignedto,year,givings Order by assignedto,year;

Q7. Select accountid,year,givings,AVG(givings) over (partition by accountid order by years rows 1 preceding) as mobavg From donations Group by accountid,year,givings Order by accountid,year;

Q8. Select donlevel,year,givings, SUM(givings) over (partition by donlevel) as totperacc, givings/SUM(givings) over (partition by donlevel) as perc From lvlyear Group by donlevel,year,givings Order by donlevel, year;

Q9. Select assignedto,givings, dense\_rank() over (order by givings desc) as ranking From nameyear Where year = 2016 Group by assignedto,givings;

H1. Select assignedto,first\_value(year) over (partition by assignedto order by year) as firstyear From nameyear Group by assignedto,year Order by assignedto;

H2. select count(distinct assignedto),sum(givings) as tot from donations,accounts,cities where donations.accountid = accounts.accountid and donations.city = cities.city and cities.country = 'USA' group by country;

H3. Select distinct assignedto from accounts where ismajor = ‘TRUE’;

H4. select assignedto,AVG(givings)as tot from donations,accounts where donations.accountid = accounts.accountid and donations.donorlevel = 5 group by assignedto order by tot desc;

H5. select donations.accountid,year,donlevel, sum(givings) over (partition by donations.accountid order by year rows unbounded preceding) as cumsum from donations,accounts where donations.accountid = accounts.accountid and accounts.ismajor = 'TRUE' group by donations.accountid,year,givings,donlevel order by donations.accountid,year;