CDAC MUMBAI

PG-DBDA SEP 2022 BATCH KHARGHAR

MODULE: BIG DATA ANALYTICS

DATE : 14TH DEC, 2022

MARKS : 40 MARKS

Please create a doc/txt/pdf file with 12 digits student id, which will contain the code along with the screenshots of the output or result. While taking the screenshot make sure that you are visible in all the images.

Q1.

MapReduce

Problem Statement **[10 marks]**

Here, we have chosen the stock market dataset on which we have performed map-reduce operations. Following is the structure of the data. Kindly Find the solutions to the questions below.

Data Structure

1. Exchange Name

2 Stock symbol

3. Transaction date

4. Opening price of the stock

5. Intra day high price of the stock

6. Intra day low price of the stock

7. Closing price of the stock

8. Total Volume of the stock on the particular day

9. Adjustment Closing price of the stock
Field Separator – comma

Question 2 : Find all time High price for each stock (Hadoop Question)

hive[15 marks]

Please find the customer data set.

cust id

firstname

lastname

age

profession

1) Write a program to find the count of customers for each profession.

Please find the sales data set.

txn id

txn date

cust id

amount

category

product

city

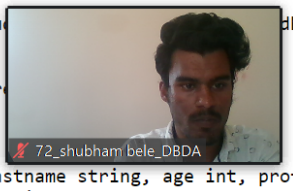
state

spendby

```
npbdh login: bigcdac432522
bigcdac432522@npbdh.cloudloka.com's password:
Last login: Wed Dec 14 07:28:18 2022 from ec2-65-1-45-35.ap-south-1.compute.amazonaws.com
[bigcdac432522@ip-10-1-1-204 ~]$ hive
WARNING: Use "yarn jar" to launch YARN applications.
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/jars/log4j-slf4j-impl-2.
SLF4J: Found binding in [jar:file:/opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/jars/slf4j-log4j12-1.7.2
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
2022-12-14 08:53:56,031 main WARN JNDI lookup class is not available because this JRE does not support JNDI. JND
guration. Ignoring java.lang.ClassNotFoundException: org.apache.logging.log4j.core.lookup.JndiLookup

Logging initialized using configuration in jar:file:/opt/clou                                    dh6.2.1.p0.1425774/jars/h
ync: false

WARNING: Hive CLI is deprecated and migration to Beeline is r
hive> create database hiveexm;
OK
Time taken: 1.804 seconds
hive> create table customer(cust_id int ,firstname string, lastname string, age int, profession string);
FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.DDLTask. AlreadyExistsException(messa
hive> use hiveexm
    > ;
```

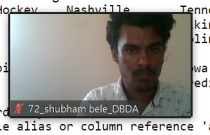



```
Time taken: 0.227 seconds
hive> create table customer(cust_id int ,firstname string, lastname string, age int, profession string) row format delimited fields terminated by ',' stored as textfi
;
OK
Time taken: 0.157 seconds
hive> desc customer
    > ;
OK
cust_id                 int
firstname               string
lastname                string
age                     int
profession              string
Time taken: 0.113 seconds, Fetched: 5 row(s)
hive> load data local inpath 'custs.txt' overwrite into table customer;
Loading data to table hiveexm.customer
OK
Time taken: 1.241 seconds
hive> select * from customer limit 10;
OK
4000001 Kristina        Chung   55      Pilot
4000002 Paige   Chen    74      Teacher
4000003 Sherri  Melton  34      Firefighter
4000004 Gretchen        Hill    66      Computer hardware engineer
4000005 Karen   Puckett 74      Lawyer
4000006 Patrick Song    42      Veterinarian
4000007 Elsie   Hamilton        43      Pilot
4000008 Hazel   Bender  63      Carpenter
4000009 Malcolm Wagner  39      Artist
4000010 Dolores McLaughlin      60      Writer
```

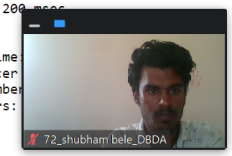
```
hive> select count(cust_id),profession from customer group by profession;
Query ID = bigcdac432522_20221214091037_eefcd2fa-9ff8-417d-8cdf-498fc4d4038e
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load 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>
22/12/14 09:10:38 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032
22/12/14 09:10:39 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032
Starting Job = job_1663041244711_22607, Tracking URL = http://ip-10-1-1-204.ap-south-1.compute.internal:6066/proxy/application_1663041244711_226
Kill Command = /opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/bin/hadoop job  -kill job_1663041244711_22607
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2022-12-14 09:10:55,772 Stage-1 map = 0%,  reduce = 0%
2022-12-14 09:11:05,137 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.22 sec
2022-12-14 09:11:19,583 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 5.17 sec
MapReduce Total cumulative CPU time: 5 seconds 170 msec
Ended Job = job_1663041244711_22607
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 5.17 sec   HDFS Read: 400772 HDFS Write: 1584 HDFS EC Read: 0 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 170 msec
OK
199     Accountant
202     Actor
195     Agricultural and food scientist
203     Architect
175     Artist
196     Athlete
193     Automotive mechanic
181     Carpenter
209     Chemist
207     Childcare worker
193     Civil engineer
201     Coach
204     Computer hardware engineer
```

2) Write a program to find the top 10 products sales wise

```
hive> create table txn(txn_id int ,txn_date string,cust_id int,amount float,category string, product string,city string,state string, spendby string) row format delimit
ed fields terminated by ',' stored as textfile;
OK
Time taken: 0.172 seconds
hive> load data local inpath 'txns1.txt' overwrite into table txn;
Loading data to table hiveexm.txn
OK
Time taken: 1.185 seconds
hive> select * from txn limit 10;
OK
0       06-26-2011      4007024 40.33   Exercise & Fitness      Cardio Machine Accessories      Clarksville     Tennessee       credit
1       05-26-2011      4006742 198.44  Exercise & Fitness      Weightlifting Gloves    Long Beach      California      credit
2       06-01-2011      4009775 5.58    Exercise & Fitness      Weightlifting Machine Accessories       Anaheim California      credit
3       06-05-2011      4002199 198.19  Gymnastics      Gymnastics Rings        Milwaukee       Wisconsin       credit
4       12-17-2011      4002613 98.81   Team Sports     Field Hockey    Nashville       Tennessee       credit
5       02-14-2011      4007591 193.63  Outdoor Recreation                              king  Chicago Illinois        credit
6       10-28-2011      4002190 27.89   Puzzles Jigsaw Puzzles                          lina  credit
7       07-14-2011      4002964 96.01   Outdoor Play Equipment                          Ohio    credit
8       01-17-2011      4007361 10.44   Winter Sports   Snowmobi                        wa    credit
9       05-17-2011      4004798 152.46  Jumping Bungee Jumping                          edit
Time taken: 0.253 seconds, Fetched: 10 row(s)
hive> select count(amount),product from txn group by product ord
FAILED: SemanticException [Error 10004]: Line 1:70 Invalid table alias or column reference 'amount': (possible column names are: _c0, product)
hive> select sum(amount),product from txn group by product order by sum(amount) desc;
FAILED: SemanticException [Error 10004]: Line 1:66 Invalid table alias or column reference 'amount': (possible column names are: _c0, product)
hive> select amount from txn limit 10;
OK
40.33
198.44
5.58
198.19
98.81
193.63
27.89
96.01
10.44
```



```
hive> select sum(amount) as t, product from txn group by product order by t desc limit 10 ;
Query ID = bigcdac432522_20221214095305_636072c6-c144-4a8b-ace7-16a5332d2851
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 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>
22/12/14 09:53:06 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032
22/12/14 09:53:06 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032
Starting Job = job_1663041244711_22849, Tracking URL = http://ip-10-1-1-204.ap-south-1.compute.internal:6066/proxy/application_1663041244711_22849/
Kill Command = /opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/bin/hadoop job  -kill job_1663041244711_22849
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2022-12-14 09:54:07,957 Stage-1 map = 0%,  reduce = 0%
2022-12-14 09:54:29,998 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.6 sec
2022-12-14 09:55:30,561 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.6 sec
2022-12-14 09:55:50,450 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 7.2 sec
MapReduce Total cumulative CPU time: 7 seconds 200 msec
Ended Job = job_1663041244711_22849
Launching Job 2 out of 2
Number of reduce tasks determined at compile time
In order to change the average load for a reducer
  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>
22/12/14 09:55:54 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032
22/12/14 09:55:54 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032
Starting Job = job_1663041244711_22870, Tracking URL = http://ip-10-1-1-204.ap-south-1.compute.internal:6066/proxy/application_1663041244711_22870/
Kill Command = /opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/bin/hadoop job  -kill job_1663041244711_22870
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
```




```
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
22/12/14 09:55:54 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032
22/12/14 09:55:54 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032
Starting Job = job_1663041244711_22870, Tracking URL = http://ip-10-1-1-204.ap-south-1.compute.internal:6066/proxy/application_1663041244711_22870/
Kill Command = /opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/bin/hadoop job  -kill job_1663041244711_22870
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2022-12-14 09:57:19,660 Stage-2 map = 0%,  reduce = 0%
2022-12-14 09:58:08,781 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 1.86 sec
2022-12-14 09:58:25,639 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 5.45 sec
MapReduce Total cumulative CPU time: 5 seconds 450 msec
Ended Job = job_1663041244711_22870
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 7.2 sec   HDFS Read: 4427073 HDFS Write: 4865 HDFS EC Read: 0 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 5.45 sec   HDFS Read: 10363 HDFS Write: 528 HDFS EC Read: 0 SUCCESS
Total MapReduce CPU Time Spent: 12 seconds 650 msec
OK
47804.94006872177       Yoga & Pilates
47204.14009475708       Swing Sets
46828.43995523453       Lawn Games
46577.6800737381        Golf
46485.54000759125       Cardio Machine Accessories
45143.839904785156      Exercise Balls
45111.67999744415       Weightlifting Belts
44995.20000743866       Mahjong
44954.680015563965      Basketball
44890.67011499405       Beach Volleyball
Time taken: 324.062 seconds, Fetched: 10 row(s)
hive>
```

3) Write a program to create partiioned table on category

```
hive> create table txn(txn_id int ,txn_date string,cust_id int,amount float, product string,city string,state string, spendby string) partitioned by (category string) r
ow format delimited fields terminated by ',' stored as textfile;
OK
Time taken: 0.136 seconds
hive> desc txn;
OK
txn_id                  int
txn_date                string
cust_id                 int
amount                  float
product                 string
city                    string
state                   string
spendby                 string
category                string

# Partition Information
# col_name              data_type               comment

category                string
Time taken: 0.131 seconds, Fetched: 14 row(s)
```

QUESTION 3 [15 marks]
PySpark

Please find the AIRLINES data set

Year

Quarter

Average revenue per seat

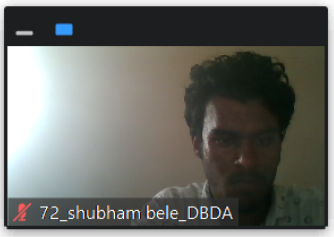
Total number of booked seats

```
LongTypeyspark.sql.types import StructType, StringType, IntegerType, DoubleType,
File "<stdin>", line 1
  from pyspark.sql.types import StructType, StringType, IntegerType, DoubleType, LongTypefrom pyspark.sql.types import StructType, StringType,
LongType

ntaxError: invalid syntax
> from pyspark.sql.types import StructType, StringType, IntegerType, DoubleType, LongType
> schema2 = StructType().add("Year",StringType(),True).add("Quarter",StringType(),True).add("ARPS",DoubleType(),True).add("Booked_seats",IntegerType(),True)
>
```


```
>>> from pyspark.sql.types import StructType, StringType, IntegerType, DoubleType, LongType
>>> schema2 = StructType().add("Year",StringType(),True).add("Quarter",StringType(),True).add("A
>>> df_with_schema = spark.read.format("csv").option("header","False").schema(schema9).load("hdf
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'schema9' is not defined
>>> df_with_schema = spark.read.format("csv").option("header","False").schema(schema2).load("hdf
>>> df_with_schema = spark.read.format("csv").option("header","False").schema(schema9).load("hdf
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'schema9' is not defined
>>> df_with_schema.printSchema()
root
 |-- Year: string (nullable = true)
 |-- Quarter: string (nullable = true)
 |-- ARPS: double (nullable = true)
 |-- Booked_seats: integer (nullable = true)

>>>
```



1) What was the highest number of people travelled in which

Year?
Year = spark.sql("select year, sum(booked_seats) as total_seats from airlines group by

```
year order by total_seats desc limit 1")
```

2) Identifying the highest revenue generation for which year

```
max_rev = spark.sql("select year, round(sum(arps*booked_seats)/1000000,2) as total
from airlines group by year order by total  desc limit 1")
```

3) Identifying the highest revenue generation for which year and quarter (Common group)

```
max_rev = spark.sql("select year, quarter,round(sum(arps*booked_seats)/1000000,2) as
total from airlines group by year,order order by total  desc limit 1")
```