BDA Practical Question Bank

For practical exam you will be getting one problem from category I and other from category II

Category I

Solve using Sqoop/Hive/HBase   -- 10 marks

Q.1.   Employee(ename ,street ,city)

Works(ename ,cname ,salary)

Company(cname ,city)

Manages(ename ,mname)

Consider above database draw E-R Diagram

Apply following constraints

Workers salary should be within range 10000 to 35000

Customer id should begin with letter c.

Primary Key and foreign key.

And write SQL query for following Statements.

ü Find all employees from database who earn more than their manager.

ü List the names of company which has maximum number of employees.

ü Find cname and no of employees working only for those company in which at least 5 employees are working.

ü List all the employees who live in the same cities as the company for which they works in.

```
hive> create table employee(ename varchar(20),street varchar(20),city varchar(20));
hive> create table company (cname varchar(20),city varchar(20));
hive> create table works(ename varchar(20),cname varchar(20),salary int);
hive> create table manages(ename varchar(20),mname varchar(20));
hive> insert into employee
values('Roma','Chopra','Unr'),('Resham','Chopra','Pune'),('Richa','Punjabi
col','Mumbai'),('Vani','5no','Pune'),('Disha','1no','Mumbai'),('khatu','OT','Unr'),('Khuba','1no','Pune'
),('Manoj','kalyan','Mumbai'),('Mohit','nagina','Malad'),('Anjali','1no','Unr');

hive> insert into company values
('Accenture','Mumbai'),('JPMC','Powai'),('Dolat','Mumbai'),('JD','Malad');

hive> insert into manages
values('Roma','Divya'),('Khatu','Divya'),('Anjali','Heena'),('Vani','Yogita'),('Mohit','Raj'),('Manoj','XY
Z'),('Disha','XYZ'),('Khuba','Tarun'),('Resham','Tarun'),('Richa','ABC');

hive> insert into works
values('Roma','Accenture',30),('Khatu','Accenture',30),('Anjali','Accenture',30),('Divya','Accenture
',35),('Manoj','JPMC',85),('Disha','JPMC',85),('Heena','Accenture',25),('Khuba','Accenture',30),('V
ani','JPMC',85),('XYZ','JPMC',60),('Yogita','JPMC',40),('Resham','JPMC',45),('Mohit','JD',35),('Ra
j','JD',40),('Richa','Dolat',40),('ABC','Dolat',35),('Tarun','Accenture',35);
hive> insert into employee
values('Divya','XYZ','Unr'),('XYZ','XYZ','Unr'),('ABC','XYZ','Unr'),('Tarun','XYZ','Unr'),('Raj','XYZ','
Unr'),('Heena','XYZ','Unr'),('Yogita','XYZ','Unr');

hive> select * from employee;
hive> select * from company;
hive> select * from manages;
hive> select * from works;

hive> select w1.ename,w1.salary,w2.ename,w2.salary from works w1,manages,works w2
where w1.ename=manages.ename and manages.mname=w2.ename and w1.salary>w2.salary;
OK
Anjali  30      Heena 25
Manoj   85      XYZ     60
Disha   85      XYZ     60
Vani    85      Yogita  40
Resham          45      Tarun   35
Richa   40      ABC     35
hive> select company.cname,count(works.cname) as totalcount from company,works where
company.cname=works.cname group by company.cname order by totalcount desc limit 1 ;
Accenture       7
hive> select company.cname,count(works.cname) as totalcount from company,works where
company.cname=works.cname group by company.cname having count(works.cname)>=5;
Accenture       7
JPMC    6
```

```
hive> select employee.ename from employee,works,company where
employee.ename=works.ename and works.cname=company.cname and
employee.city=company.city;
Mohit
Richa
```

Q.2. Customer(cid,cname,city)

Deposits(cid,accno)

Account(bname,accno,balance)

Borrows(lno,cid)

Loan(lno,amount)

Consider above database draw E-R Diagram

ü Apply following constraints

Account balance should be within range 10000 to 25000

Customer id should begin with letter c.

Primary Key and foreign key.

write SQL query for following Statements

ü Find customer names those are having account balance more than loan .

ü Find branch name with minimum assets.

ü Find customer id, customer name of customers those are having at least 2 accounts and at least one loan.

ü Find those accounts whose balance is more than all accounts at dadar branch.

ü Delete all accounts which belongs to dadar branch and balance is more than account balance of accounts of john

```
 hive> create table customer(cid int,cname varchar(20),city varchar(20));

hive> create table deposits(cid int,accno int));
hive> create table account(bname varchar(20),accno int,balance int);

hive> create table borrows(lno int,cid int);

hive> create table loan(lno int,amount int);

hive> insert into customer values(1,'divya','mumbai'),(2,'khushboo','pune'),(3,'karishma','bangalore'),(4,'piku','chennai');

hive> insert into deposits values(1,120),(1,200),(2,200),(3,500),(4,1000);

hive> insert into account values('dadar',120,500),('dadar',200,500),('pune',200,600),('chennai',500,700),('kurla',1000,50);

hive> insert into borrows values(100,1),(101,2),(102,3);

hive> insert into loan values(100,100),(101,200),(102,300);

hive>select c.cname,a.accno,a.balance,l.amount from customer c,deposits d,account a,borrows b,loan l

    > where c.cid=d.cid and d.cid=b.cid and d.accno=a.accno and b.lno=l.lno and a.balance>l.amount;

divya           120     500     100

divya            200    500     100

divya            200    600     100

khushboo        200     500     200

khushboo        200     600     200

karishma        500     700     300

hive> select bname,sum(balance) as sm from account

    > group by bname

    > order by sm asc

    > limit 1;
```

```
kurla    50

hive> select c.cid,count(d.cid) as ct1,count(b.cid) as ct2 from
    > customer c,deposits d,borrows b where c.cid=b.cid and c.cid=d.cid
    > group by c.cid having ct1>1 and ct2>0;
1         2         2
```

4,5 Query not possible in hive as hive doesn’t support deletion of tuples and nested queries.

Q.3. A library has the following relations

Library(code ,name, noofbooks)

Person(id,name,age)

Imember(code,id,Dateofjoining)

Books(Accessionno,title,author,price)

Borrowedby(Accessionno,id,Date_of_borrow)

ü Consider above database draw E-R Diagram.

ü Apply following constraints

Personid should begin with letter P.

Primary Key and foreign key.

write SQL query for following Statements

ü Give details of person who has borrowed at least two books.

ü Give details of person who has borrowed at least one book along with database concepts.

ü Find name of book which has been borrowed minimum number of times

ü Delete all entries from borrowedby of database book borrowed by pid P101.

ü Find person details of persons who has borrowed database books with author korth and navathe.

Q.4 Employee(ssn ,ename ,salary ,superssn ,dno,pno)

Dept(dno,dname)

Project(pno,pname,dno)

Dependent(ssn,dependentname,relationship)

Apply following constraints

SSN should be exactly length 3.

Primary Key and foreign key.

Consider the above database draw E-R Diagram and write SQL statements for the following queries.

ü Retrieve employee name and supervisor name of employees those are earning salary more than their respective supervisors.

```
hive> select e1.ename,e2.ename from employee e1,employee e2 where
e1.superssn=e2.ssn and e1.salary>e2.salary;
```

ü Retrieve employee details of employees those are earning salary more than

Average salary of department for which employee is working.

```
hive> select ssn,ename,salary,superssn,employee.dno,pno from employee, (select
dno,AVG(salary) as average from employee group by dno) as t where
employee.dno=t.dno and employee.salary>t.average;
```

ü Give 15% raise in salary if salary is greater than 20000, 10% raise if salary

is within range 10000 to 20000 else 5% raise.

ü Retrieve employee details of employee those belongs to IT department and working on at least one project controlled by IT department.

ü Retrieve employee details those are working on Inventory project but does not Belongs to computer department.

ü Give one example of multiple table based view.

ü Consider schema

Branch (bname0, assets, city) and Accounts (accno,balance,bname)

Q.5. Customer(cid,cname,city,accno)

Account(bname,accno,balance)

Borrows(lno,cid)

Loan(lno,amount)

Consider above database draw E-R Diagram

ü Apply following constraints

Account balance should be within range 10000 to 25000

Customer id should begin with letter c.

Primary Key and foreign key.

Write SQL query for following Statements

ü Find customer details of customers those are having account balance more than

20000 at dadar branch and at least one loan. .

ü Find bname of branches those are having at least 2 accounts.

ü Find customer id, customer name those are having at least 1 accounts and at least 2 loan.

ü Give one example of left outer Join.

ü Delete all accounts which belongs to dadar branch and balance is more than account balance of accounts of john

üü Give example of multi table-based view and show it's updation

Q6. In sqoop

Create table in MySQL, import tables in sqoop , export tables from sqoop

```
sqoop export --connect jdbc:mysql://localhost/hue --username root --password cloudera
--export-dir=/user/cloudera/oozie_fs --table oozie_fs
```

Q7. Hue for data analysis

Category II

Solve **pyspark/mapreduce** program 15 marks

1. **Program to count 4-lettered words**

2. **Program to count 3-lettered words**

3. **Program to count 2-lettered words**

```
rdd1 = sc.textFile("file:/home/cloudera/Desktop/4letter.txt")

>>> rdd2 = rdd1.flatMap(lambda line:line.split())

>>> rdd3 = rdd2.filter(lambda word:len(word)==4)

>>> rdd4 = rdd3.map(lambda word:(word,1))

>>> rdd5 = rdd4.reduceByKey(lambda v1,v2:(v1+v2))

>>> rdd5.collect()

[(u'hell', 1), (u'diva', 1), (u'juhi', 1), (u'till', 1)]
```

4. **Program to count words starting with 'I'**

```
rdd1 = sc.textFile("file:/home/cloudera/Desktop/startI.txt")

>>> rdd2 = rdd1.flatMap(lambda line:line.split())

>>> rdd3 = rdd2.filter(lambda word :word[0]=="i" or word[0]=="I" )

>>> rdd4 = rdd3.map(lambda word:(word,1))

>>> rdd5 = rdd4.reduceByKey(lambda v1,v2:(v1+v2))

>>> rdd5.collect()

[(u'iqbaal', 1), (u'indu', 1), (u'ishq', 1), (u'illinois', 1)]
```

5. Program to give matrix-vector multiplication

```
from pyspark import SparkContext

from pyspark.mllib.linalg.distributed import *

import numpy as np

sc = SparkContext("local", "Matrix Vector Multiplication")

matrix = [[1, 2, 3],

          [4, 5, 6],

          [7, 8, 9]]

vector = np.array([17, 18, 19])

mat_rdd = sc.parallelize(matrix)

mul_rdd = mat_rdd.map(lambda x: x * vector)\

          .map(lambda x: sum(x))\

          .collect()

print(mul_rdd)
```

6. Program to implement join of tables:

```
 users = sc.parallelize([(0,"divya"),(1,"Khushboo"),(2,"Karishma")])

>>> hobbies = sc.parallelize([(0,"ghumna"),(0,"pareshani"),(1,"padhna")])
>>> users.join(hobbies).collect()

[(0, ('divya', 'ghumna')), (0, ('divya', 'pareshani')), (1, ('Khushboo', 'padhna'))]

>>> users.leftOuterJoin(hobbies).collect()

[(0, ('divya', 'ghumna')), (0, ('divya', 'pareshani')), (2, ('Karishma', None)), (1, ('Khushboo',
'padhna'))]

>>> users.join(hobbies).map(lambda x:x[1][0]+" likes "+x[1][1]).collect()

['divya likes ghumna', 'divya likes pareshani', 'Khushboo likes padhna']
```

7. Program to sort the given dataset

```
rdd1 = sc.textFile("file:/home/cloudera/Desktop/sort.txt")

>>> rdd2 = rdd1.flatMap(lambda line:line.split())

>>> rdd3 = rdd2.map(lambda word:(word,1)).reduceByKey(lambda v1,v2:(v1+v2))

>>> print rdd2.sortBy(lambda a:a[0]).collect()

[u'all', u'are', u'divya', u'hii', u'how', u'i', u'kaise', u'you']
```

8. Program to find given word string in the dataset

```
rdd1 = sc.textFile("file:/home/cloudera/Desktop/search.txt")

>>> searchTerm = "Divya"

>>> rdd2 = rdd1.filter(lambda line:(searchTerm in line)).collect()

>>> print rdd2

[u'Divya']
```

9. Program to find average temperature/user rating

```
rdd1 = sc.textFile("file:/home/cloudera/Desktop/average.txt")

>>> rdd2 = rdd1.map(lambda word:(word.split()[0],(int(word.split()[1]),1)))

>>> rdd3 = rdd2.reduceByKey(lambda v1,v2:((v1[0]+v2[0]),(v1[1]+v2[1])))

>>> rdd3.collect()
```

```
[(u'pune', (80, 1)), (u'delhi', (100, 2)), (u'mumbai', (90, 2))]

>>> rdd4 = rdd3.map(lambda word:(word[0],word[1][0]/word[1][1]))

>>> rdd4.collect()

[(u'pune', 80), (u'delhi', 50), (u'mumbai', 45)]
```

10. Program to implement k-means

11. Program to implement PageRank

HBASE

Create ‘tablename’,’columnfamily1’,’columnfamily2’
Or
Create ‘tablename’,{NAME=>’column family’}

put ‘tablename’,’row 1’,’columnfamily:column’,’value’
To update : put ‘tablename’,’row 1’,’columnfamily:column’,’new value’

To get one row : get ‘table name’,’row 1’
To get one column : get ’table name’,’row 1’,{COLUMN => ‘column family:column’}
To delete one whole row : deleteall ‘tablename’,’row1’

Display whole table : scan ‘table name’
Count number of rows : count ‘tablename’
To empty table : truncate ’tablename’

Before dropping a table disable it
disable ‘tablename’
drop ‘tablename’