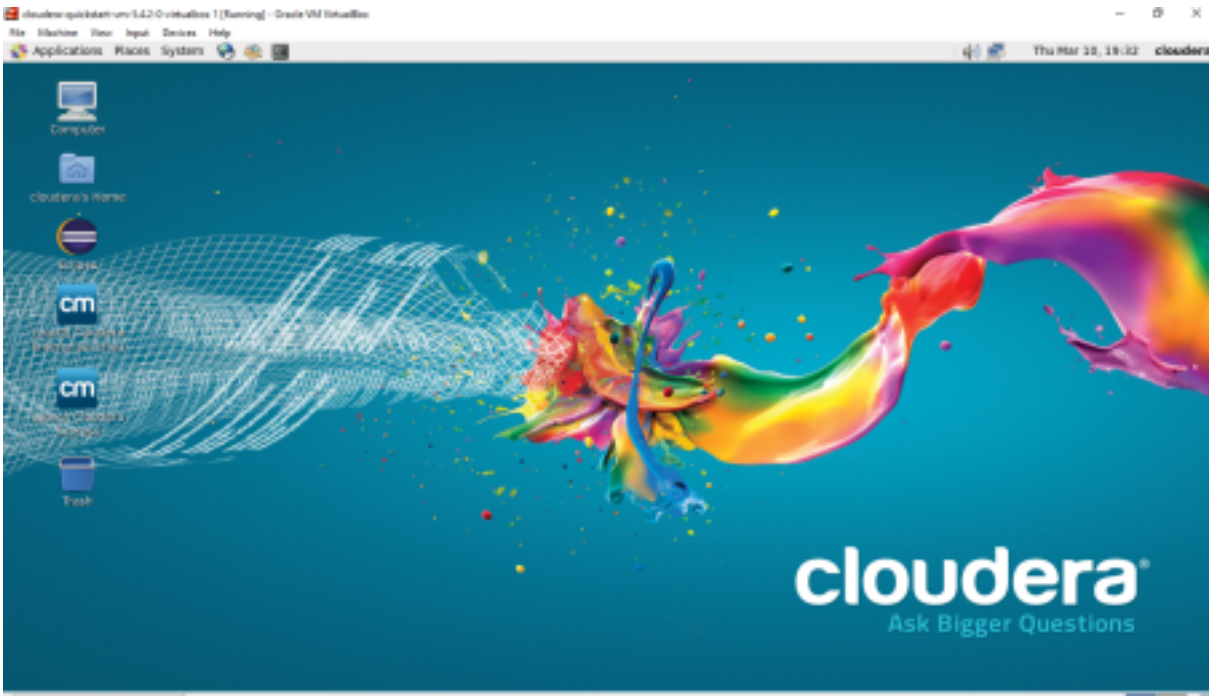
Name: Pooja Pathak
Roll No:14
MSc Part-I Sem-2

Subject: Big Data Technology
Practical_7: Querying, Sorting, Aggregating data using HiveQL

Steps:Querying, Sorting, Aggregating data using HiveQL
1. Open the cloudera.



2. Open the terminal, Now we use hive command to enter the hive shell prompt and in
hive shell we could execute all of the hive commands.

3. Now we will see the databases which are already existing using below command.
Show databases;

Name: Pooja Pathak
Roll No:14
MSc Part-I Sem-2

```
hive> show databases;
OK
default
rjc
Time taken: 0.551 seconds, Fetched: 2 row(s)
hive>
```

4.

If we want to drop the database with the entire data (rows) then we will use below
command. Here we don't have any existing database rather than default so if for example
I have 'office' as a database then will drop the database along with the data using
command as,
drop database rjc;

```
hive> drop database rjc;
OK
Time taken: 0.169 seconds
hive>
```

5. Creating a database name 'RJC' using below
command. Create database RJC;

```
hive> create database rjc;
OK
Time taken: 0.058 seconds
hive>
```

6. So if we want to see whether this RJC database is created or not we will use below
command,
show databases;

```
hive> show databases;
OK
default
rjc
Time taken: 0.029 seconds, Fetched: 2 row(s)
hive>
```

Name: Pooja Pathak
Roll No:14
MSc Part-I Sem-2

7. Now we want to check whether we have any tables inside this rjc database or not. So first
we will move to this databse rjc using below command,
Use rjc;

```
hive> use rjc;
OK
Time taken: 0.014 seconds
hive>
```

Now we have moved inside this rjc database. Now we will check out which are the tables
available using below command,
show tables;

```
hive> show tables;
OK
Time taken: 0.04 seconds
hive>
```

9. Now we will create a table inside this rjc database named as employee using below
command,
create table employee(ID int, name string, salary float, age int) After this we will not put semicolon , When we will be loading the data from some
existing csv file or maybe some other text files so we have to mention that how that data
has to be loaded here. We are simply creating the schema of the table with some certain
fields or attributes along with their datatypes and then I'm mentioning

- **row format delimited**
- **fields terminated by ',';**

```
hive> create table employee(ID int, name string, salary float, age int)
    > row format delimited
    > fields terminated by ',';
OK
Time taken: 0.327 seconds
```

10. So now we will see the schema of the table using below command, describe employee;
It will give different fields of table employee along with their respective datatypes.

```
hive> describe employee;
OK
id                      int
name                    string
salary                  float
age                     int
Time taken: 0.133 seconds, Fetched: 4 row(s)
```

So Now we will check how the table which we will be created is internal table or external
table using below command,
describe formatted employee;

Roll No:14
MSc Part-I Sem-2

```
hive> describe formatted employee;
OK
# col_name              data_type               comment

id                      int
name                    string
salary                  float
age                     int

# Detailed Table Information
Database:               rjc
Owner:                  cloudera
CreateTime:             Thu Mar 10 19:46:08 PST 2022
LastAccessTime:         UNKNOWN
Protect Mode:           None
Retention:              0
Location:               hdfs://quickstart.cloudera:8020/user/hive/warehouse/rjc.
db/employee
Table Type:             MANAGED_TABLE
Table Parameters:
        transient_lastDdlTime   1646970368

# Storage Information
SerDe Library:          org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
InputFormat:            org.apache.hadoop.mapred.TextInputFormat
OutputFormat:           org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputForm
at
Compressed:             No
Num Buckets:            -1
Bucket Columns:         []
Sort Columns:           []
Storage Desc Params:
        field.delim             ,
        serialization.format    ,
Time taken: 0.091 seconds, Fetched: 30 row(s)
```

By default hive createsInternal table or Managed Table.

12. Now we will create the external table using below command, create external table emloyee2 (ID int, name string, salary float, age int)

- **➢ row format delimited**
- **➢ fields terminated by ‘,’**
- **➢ stored as textfile;**

```
hive> create external table employee2(ID int, name string, salary float, age int
)
    > row format delimited
    > fields terminated by ','
    > stored as textfile;
OK
Time taken: 0.061 seconds
hive>
```

13. Checking the schema of the table using below command, BDT - Practical_7

Name: Pooja Pathak
Roll No:14

describe employee2;

```
hive> describe employee2;
OK
id                      int
name                    string
salary                  float
age                     int
Time taken: 0.1 seconds, Fetched: 4 row(s)
```

14. So Now we will check how the table which we will be created is internal table or external
table using below command,
describe formatted employee2;

```
hive> describe formatted employee2;
OK
# col_name              data_type               comment

id                      int
name                    string
salary                  float
age                     int

# Detailed Table Information
Database:               rjc
Owner:                  cloudera
CreateTime:             Thu Mar 10 19:59:41 PST 2022
LastAccessTime:         UNKNOWN
Protect Mode:           None
Retention:              0
Location:               hdfs://quickstart.cloudera:8020/user/hive/warehouse/rjc.db/employee2
Table Type:             EXTERNAL_TABLE
Table Parameters:
        EXTERNAL                TRUE
        transient_lastDdlTime   1646971181

# Storage Information
SerDe Library:          org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
InputFormat:            org.apache.hadoop.mapred.TextInputFormat
OutputFormat:           org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
Compressed:             No
Num Buckets:            -1
Bucket Columns:         []
Sort Columns:           []
Storage Desc Params:
        field.delim             ,
        serialization.format    ,
Time taken: 0.111 seconds, Fetched: 31 row(s)
hive>
```

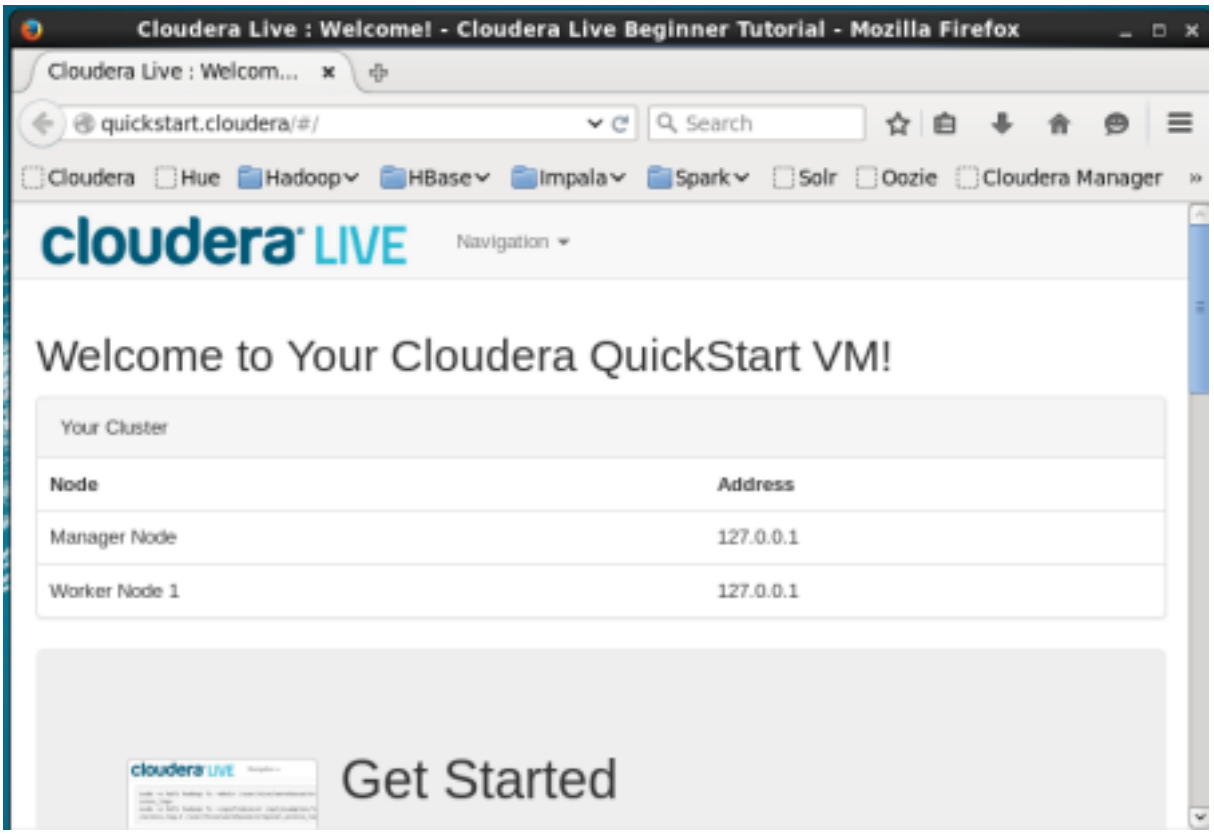
It is an External table.

15. So whatever we have done in terminal the same thing we can see in the browser as well.
Open the browser → click on Hue

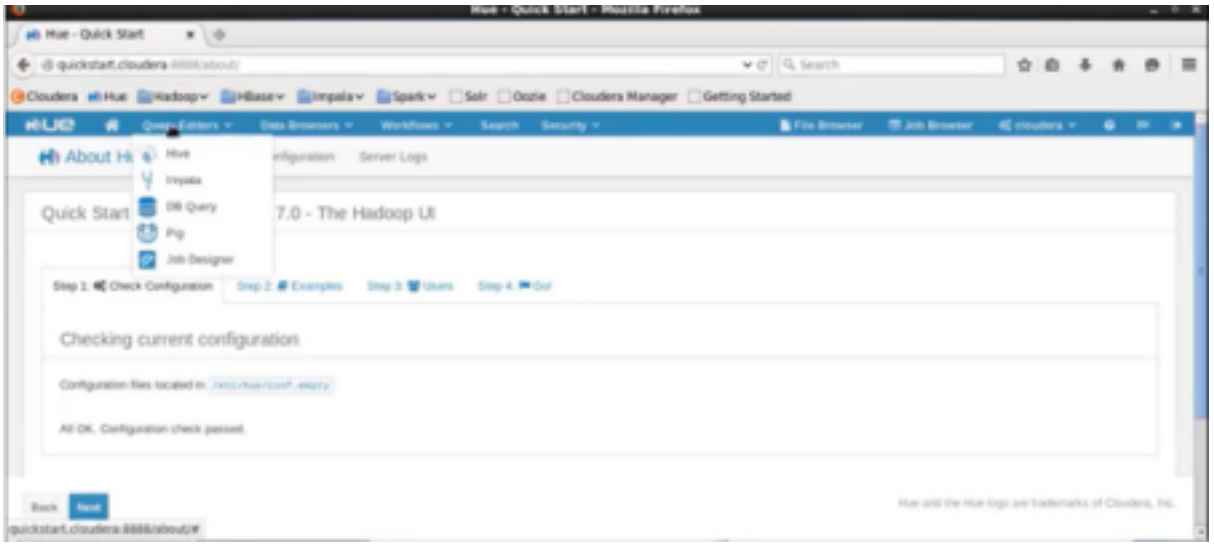
BDT - Practical_7

Name: Pooja Pathak
Roll No:14

MSc Part-I Sem-2



Then click on Query Editor → select Hive

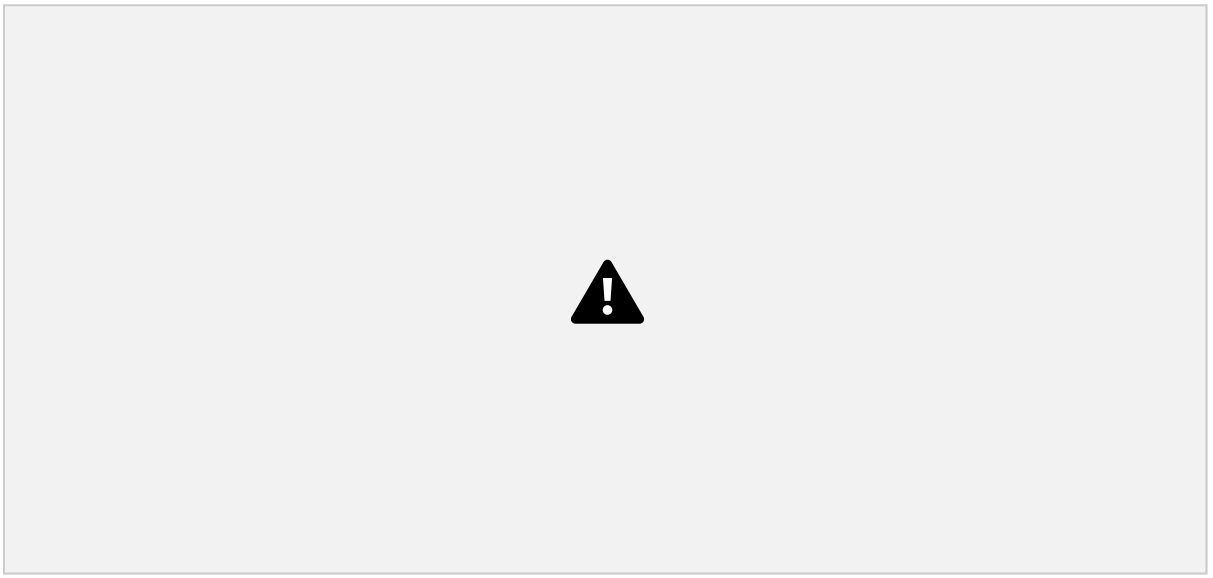


Write the query in query editor. Here we are showing the databases by using below
command,
show databases;

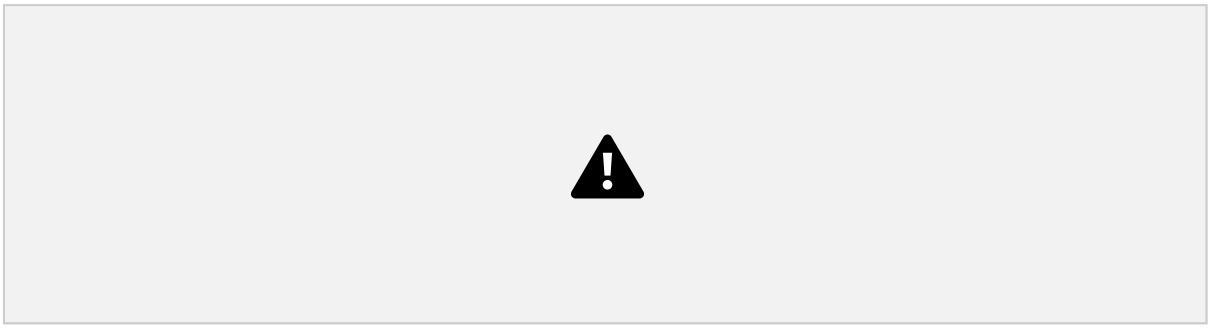
BDT - Practical_7

Name: Pooja Pathak
Roll No:14
MSc Part-I Sem-2

It will give the list of databases which are present.
There are only two databases present i.e. default and rjc which we created earlier.
We can also see the list of databases in left side corner. And after clicking on the database
name here it is “rjc” it will give the list of all tables present inside “rjc” database.



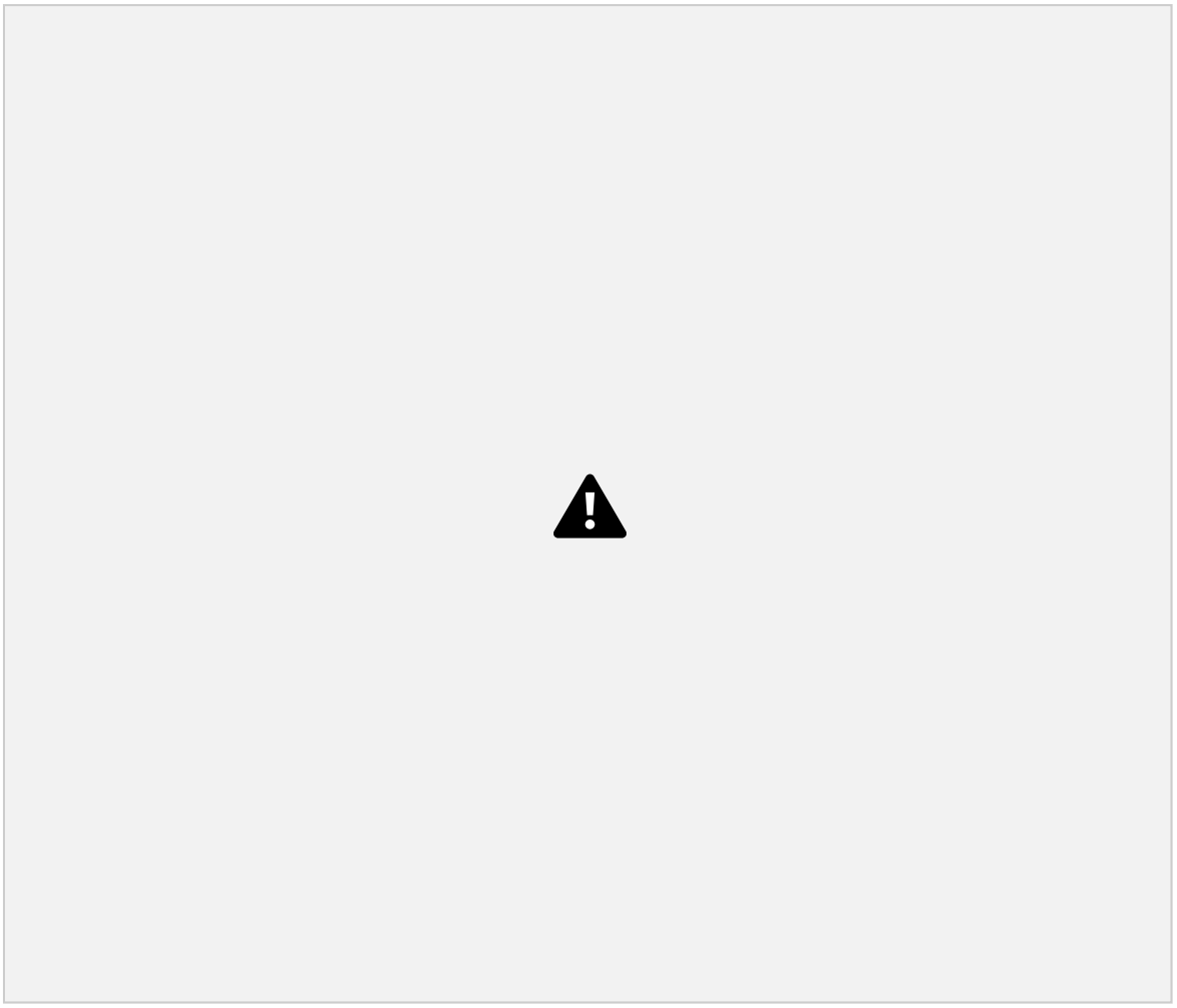
We can also Preview the sample data. Here it is showing blank because we have not
inserted anything or data inside this employee table.



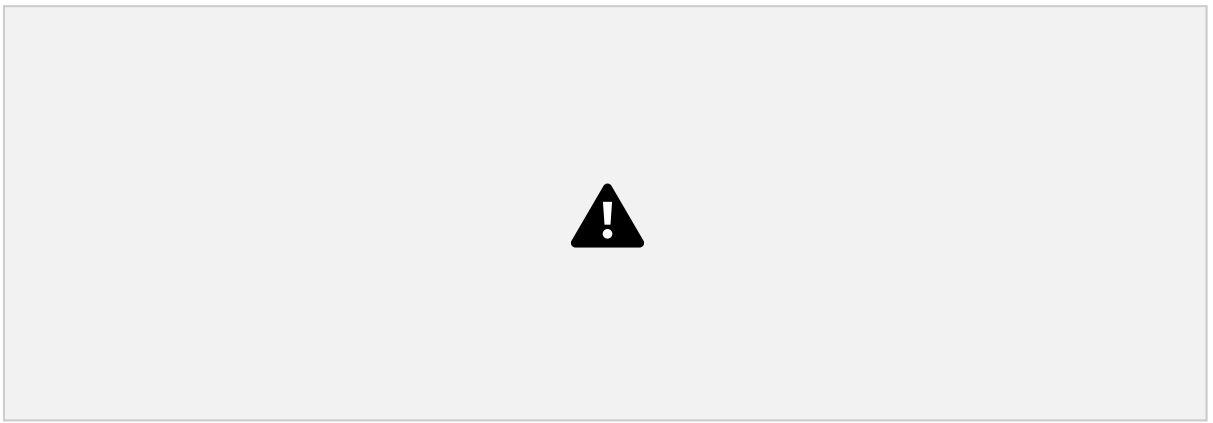
When we click on employee table it will give the fields of employee table.

BDT - Practical_7

Name: Pooja Pathak
Roll No:14
MSc Part-I Sem-2



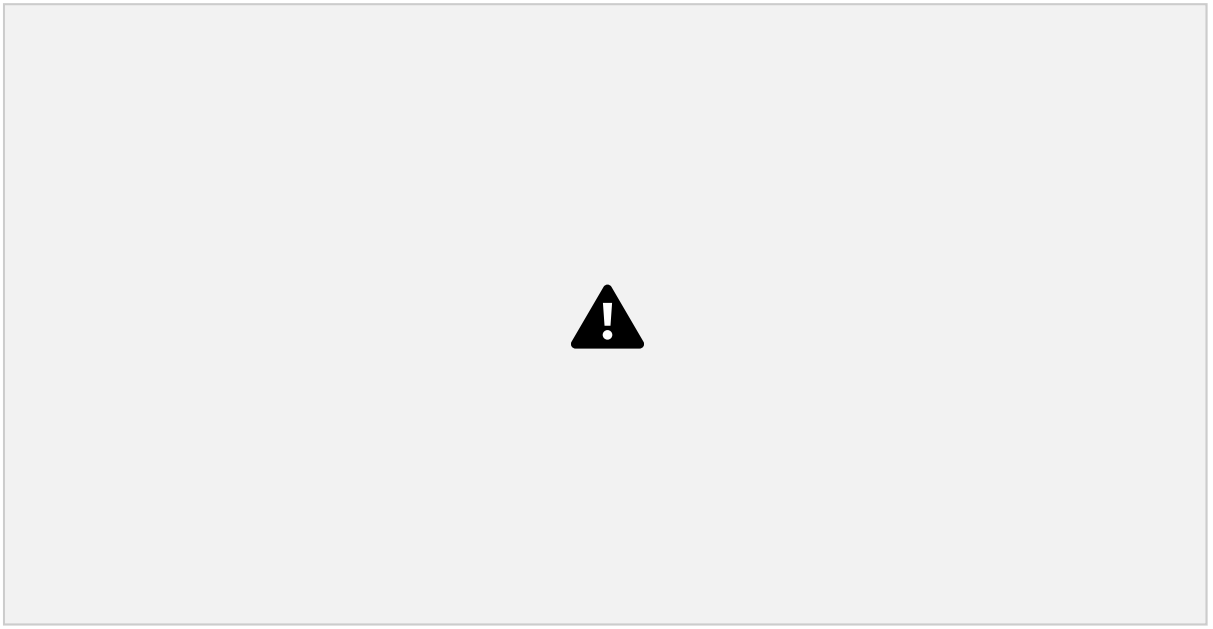
Now click on employee2 table which is an external table then click on eye type icon it
will give the employee2 table Metastore information like fields and their respective
Datatypes.



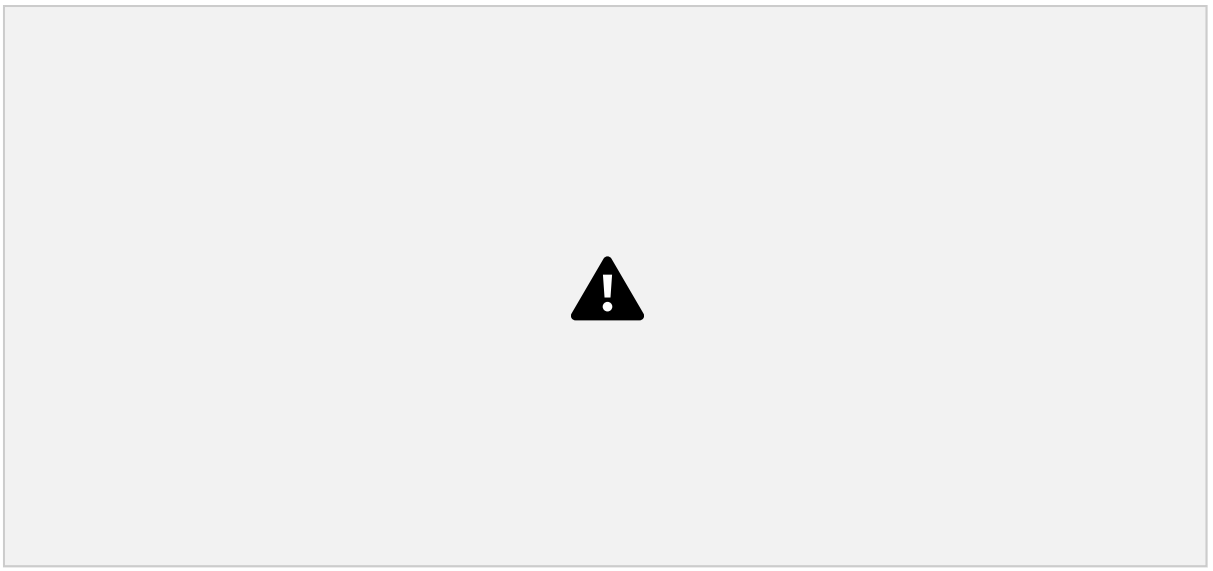
BDT - Practical_7

Name: Pooja Pathak
Roll No:14
MSc Part-I Sem-2

And click on properties it will give the properties of employee2 table.



here we can also see that the External is set to TRUE.



16. Creating a new external table named as employee3 in the specific location using below
command,
create external table emloyee3 (ID int, name string, salary float, age int)
➢ row format delimited

BDT - Practical_7

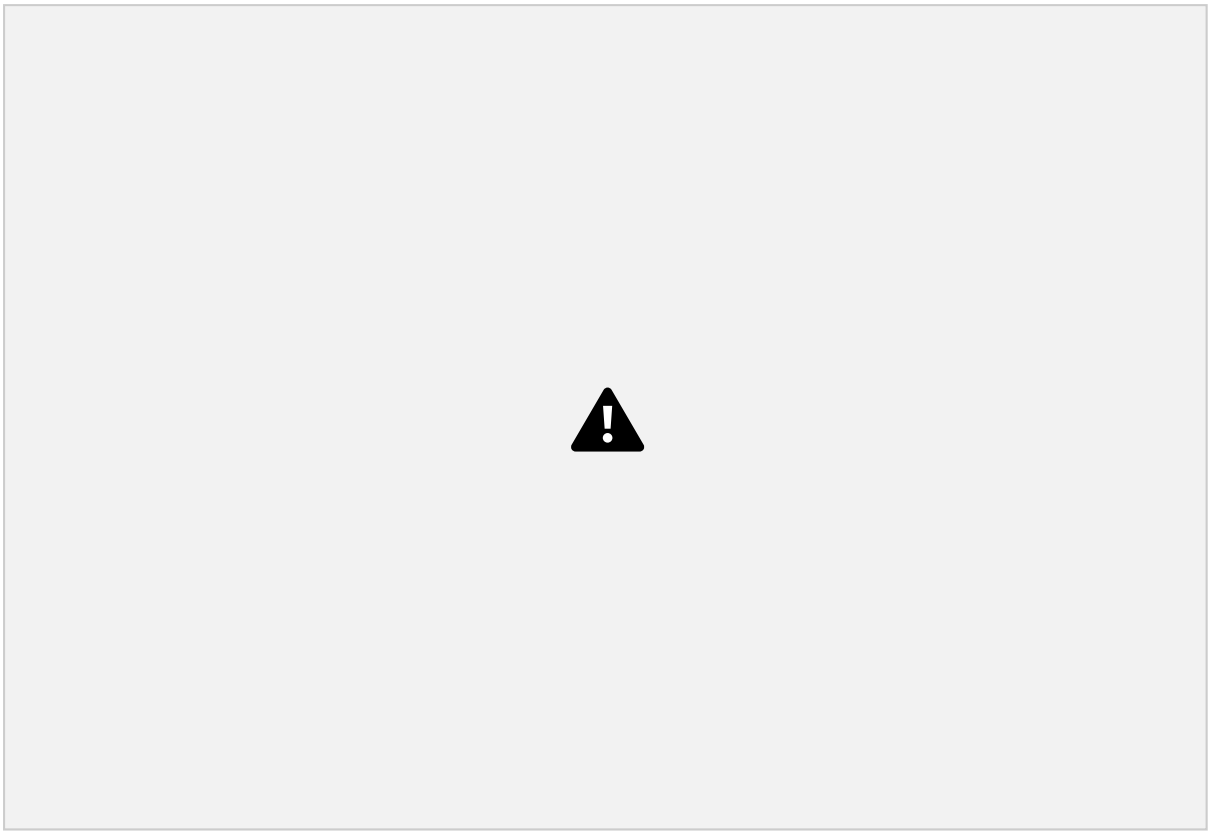
Name: Pooja Pathak
Roll No:14
MSc Part-I Sem-2

➢ **fields terminated by ‘,’**
➢ **location '/user/cloudera/vj';**

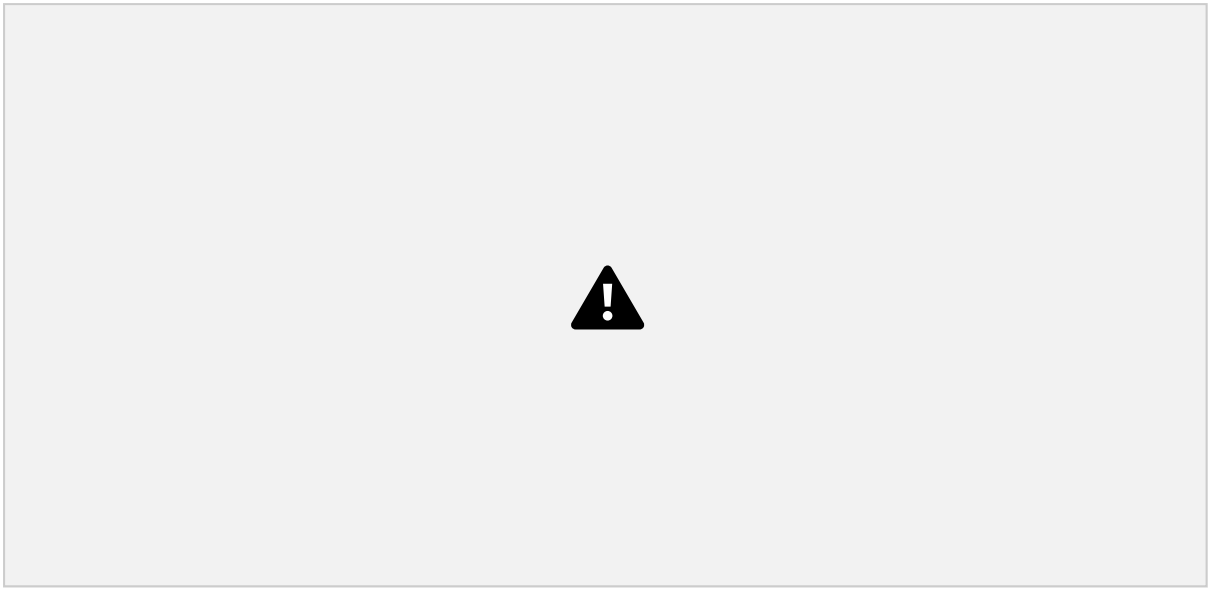
17. To see the schema of the employee3 table we use below command, describe employee3;

18. Then switch to browser and Refresh and select the rjc database and refresh, Now it will display
the employee3 table along with other two tables which we have created earlier.

BDT - Practical_7
Name: Pooja Pathak
Roll No:14
MSc Part-I Sem-2

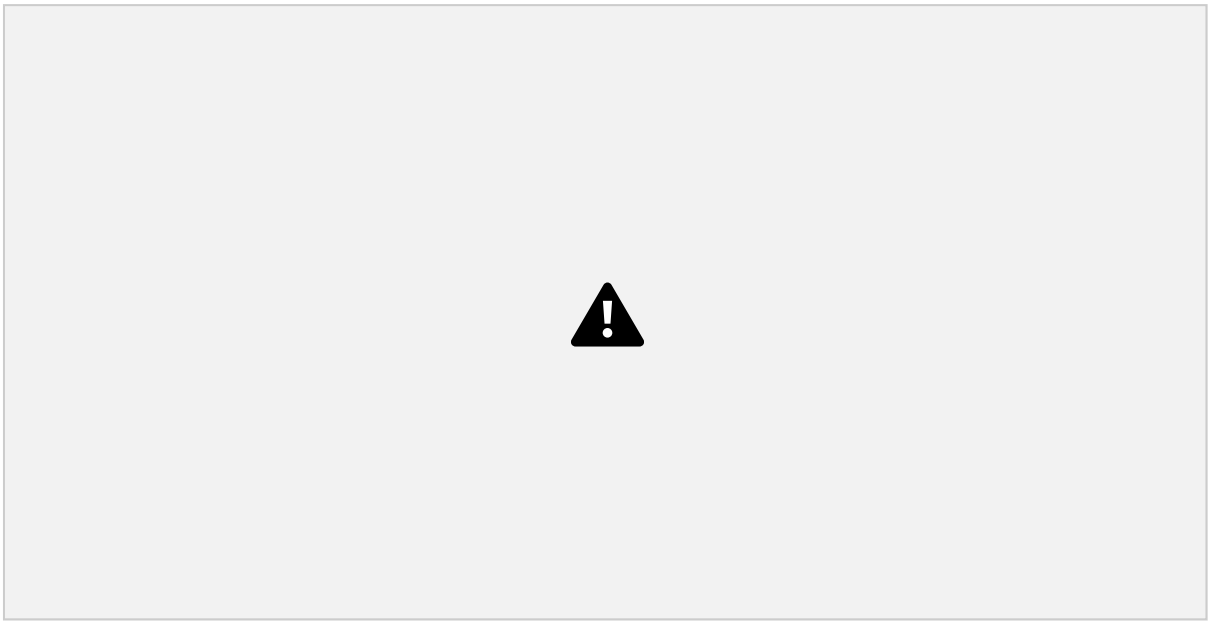


Here we will see the properties of employee3 table here we can also see the location of the
table where it is stored.

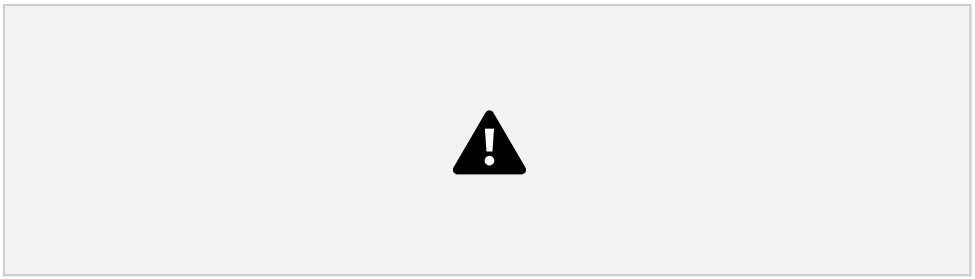


BDT - Practical_7

Name: Pooja Pathak
Roll No:14
MSc Part-I Sem-2



19. Now move to terminal and listing out all the tables using below command;
show tables;



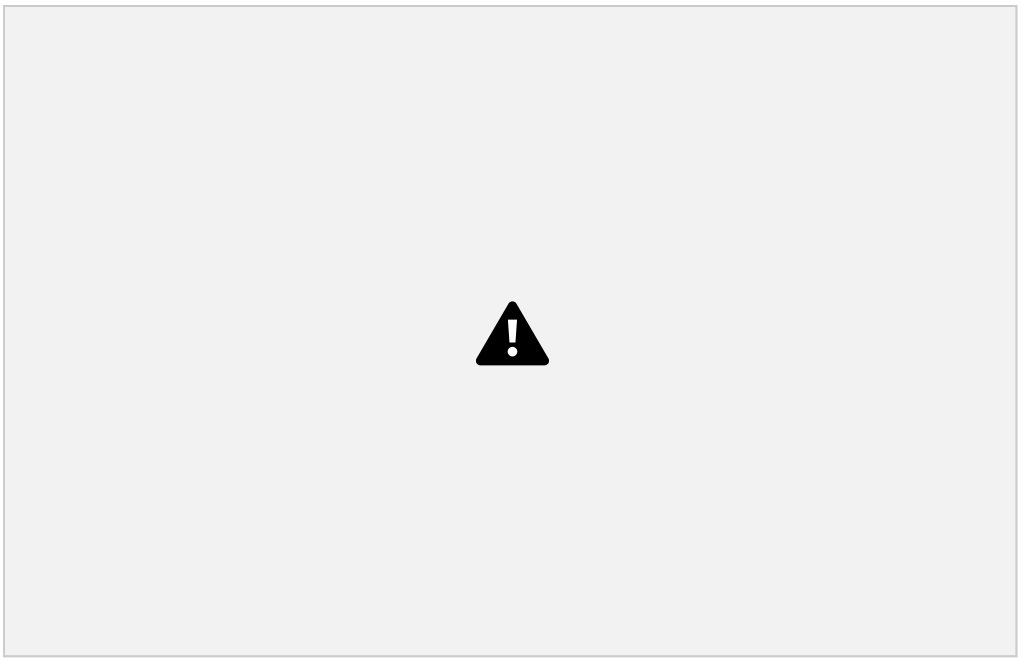
20. ALTER COMMANDS
Now we are changing the name of the employee3 table to emptable using below command,

21. Now we will check whether the name of the employee3 table changes to emptable or not
using below command,
show tables;

BDT - Practical_7

Name: Pooja Pathak
Roll No:14
MSc Part-I Sem-2

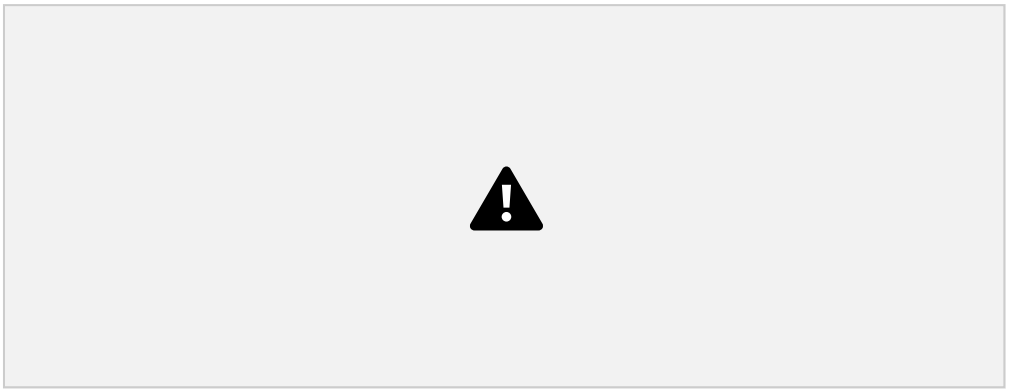
22. First we will see the fields of emptable then we will add new column as surname in emptable
using below command,
describe emptable;
Alter table emptable add columns (surname string);
describe emptable;



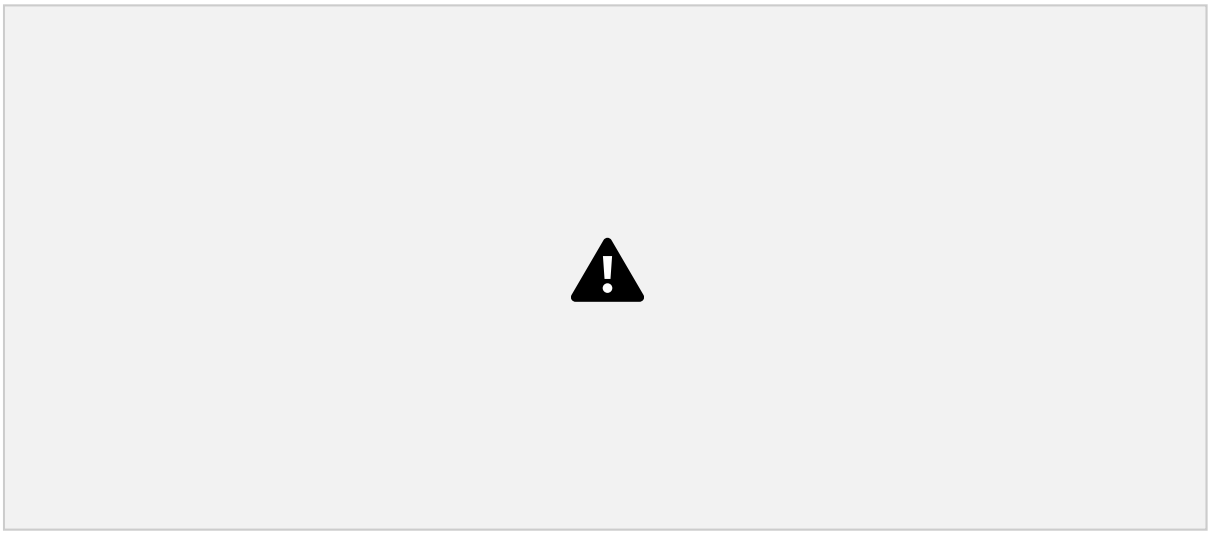
23. Now we will change field name of the emptable to first_name using alter command,
Alter table emptable change name first_name string;
describe emptable;

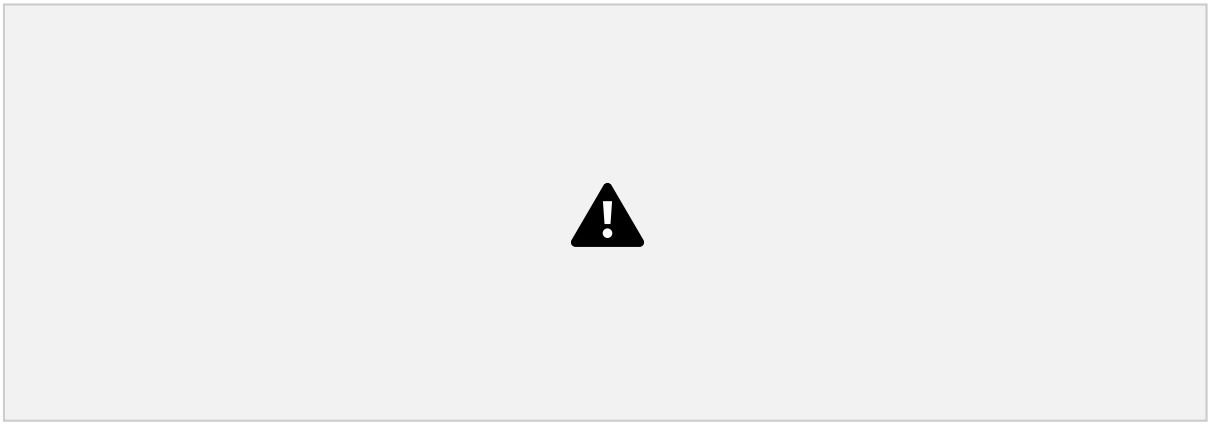
BDT - Practical_7

Name: Pooja Pathak
Roll No:14
MSc Part-I Sem-2



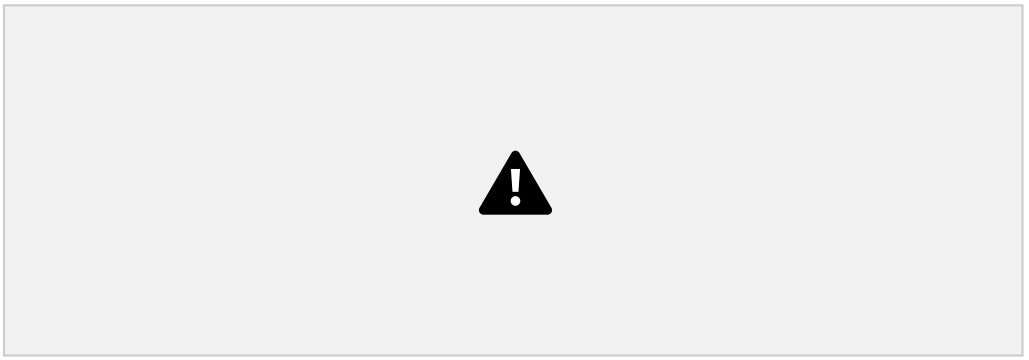
24. Before loading the data in the table we will first create the csv file. Now open the new terminal ,
using ls command list out all the directories --> change the directory to document directory -->
use ls command to list all the files present inside the document folder or directory



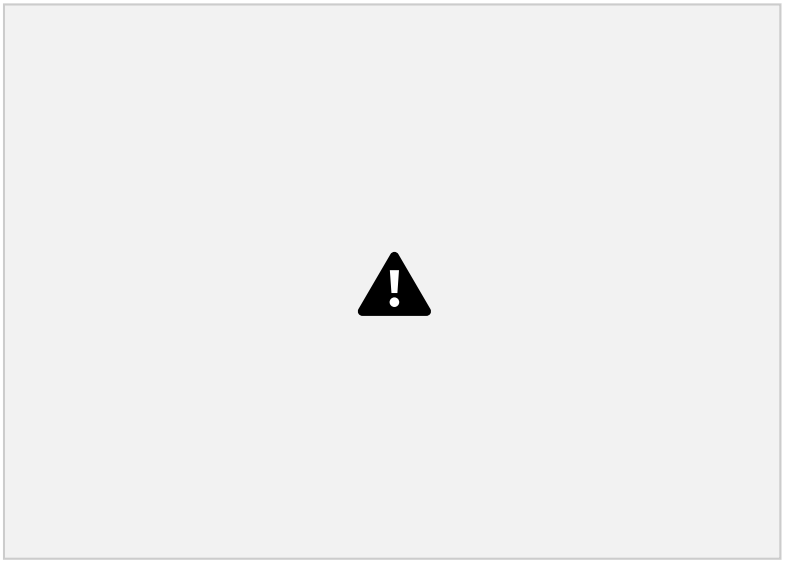


25. Now creating new file as Student.csv using below command, gedit Student.csv

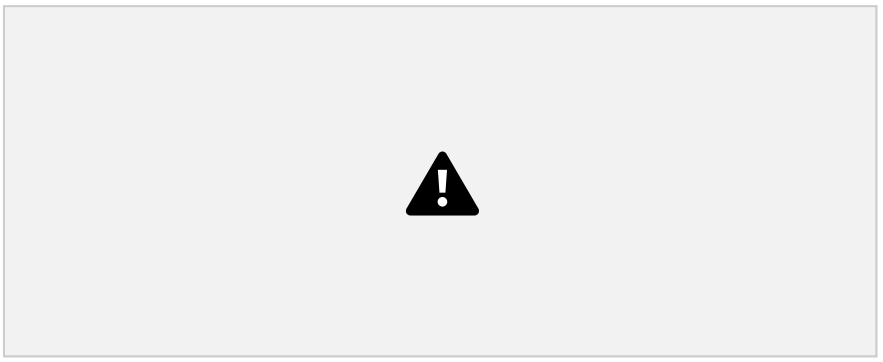
Name: Pooja Pathak
Roll No:14
MSc Part-I Sem-2



Now we are populating the Student.csv file with some data and save this file.



26. Creating a new database as rjcstudent.
create database rjcstudent;
show databases;



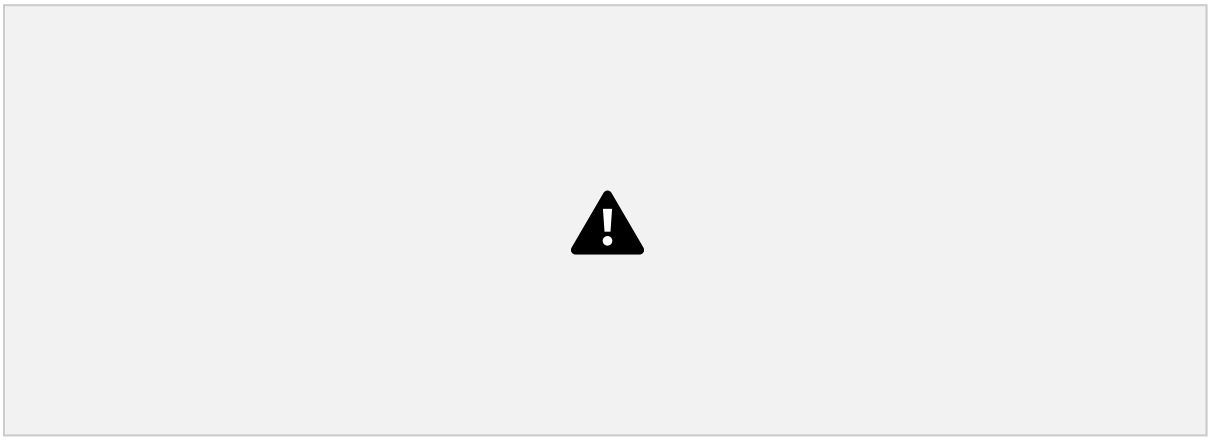
27. Using rjcstudent database.
use rjcstudent;

28. Creating new table student inside rjcstudent database. create table student (ID int, Name string, Age int)

- **partitioned by(Course string)**
- **row format delimited**
- **fields terminated by ‘,’;**

29. To see the structure or schema of the table,
describe student;

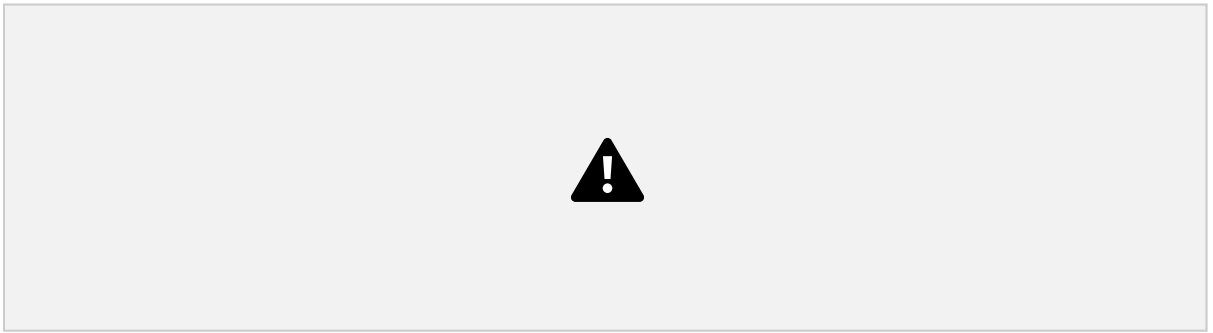
Name: Pooja Pathak
Roll No:14
MSc Part-I Sem-2



30. Loading data in the student table from Student.csv file which we have created in
document directory. Here we are partitioning based on course = 'Hadoop'.
load data local inpath '/home/cloudera/Student.csv' into table student

- **partitioned by(Course string)**
- **row format delimited**

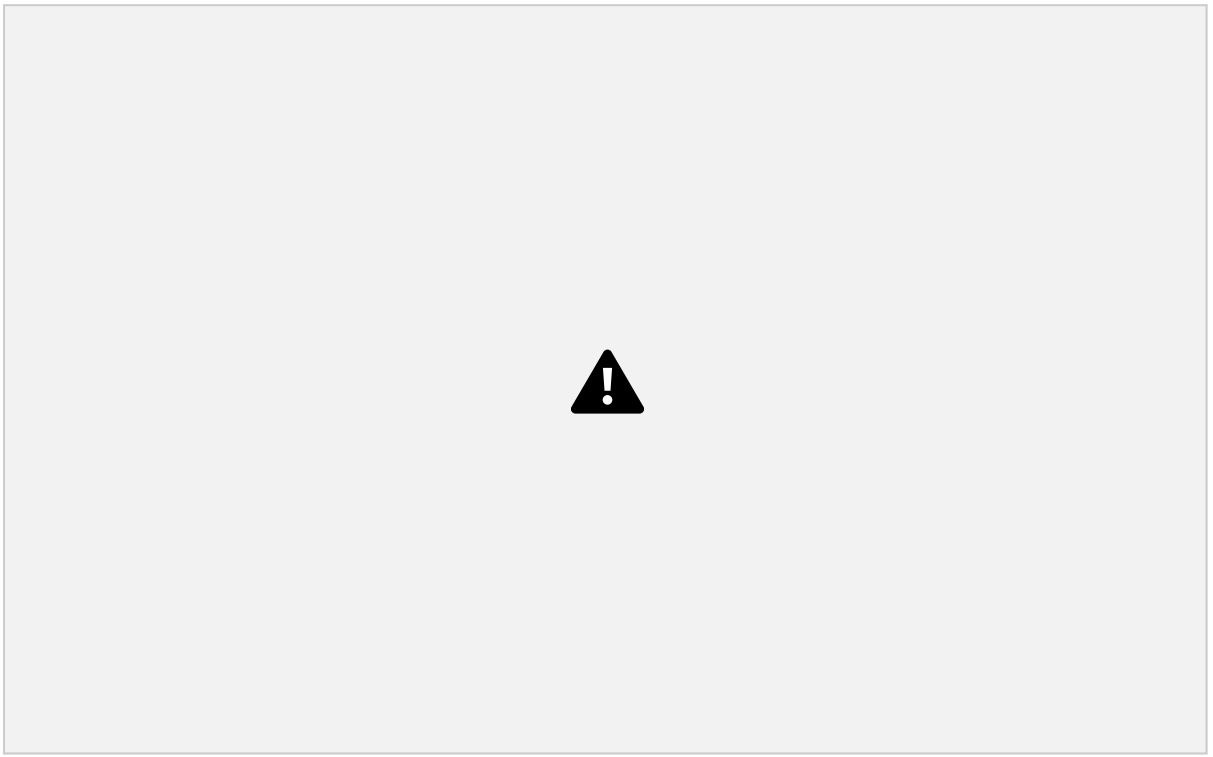
31. Now we similarly partition for course = Java and course = Python.



32. Now go to browser refresh the page and select database as rjcstudent and click in
preview student table.

BDT - Practical_7

Name: Pooja Pathak
Roll No:14
MSc Part-I Sem-2



33. Now dropping the table student and creating the student table again normally (i.e. without
partitioning).
drop table student;

create table student (ID int, Name String, Course string, Age int)
➢ row format delimited
➢ fields terminated by ‘,’

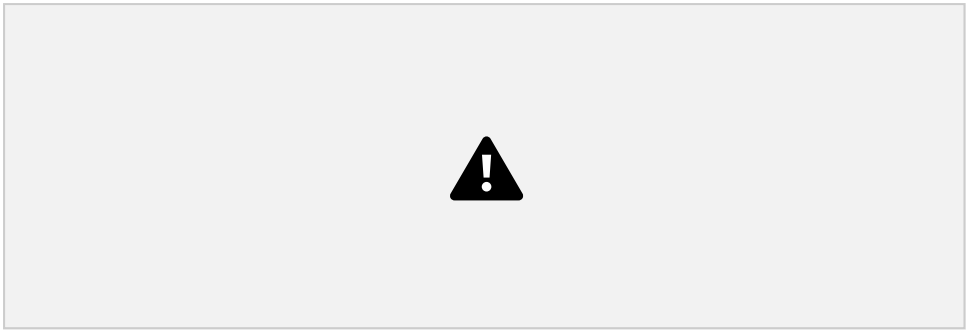
BDT - Practical_7

Name: Pooja Pathak
Roll No:14
MSc Part-I Sem-2

➢ tblproperties("skip.header.line.count" ="1");

34. Loading data in the student table from Student.csv file which present inside
/home/cloudera.
load data local inpath '/home/cloudera/Student.csv' into table student

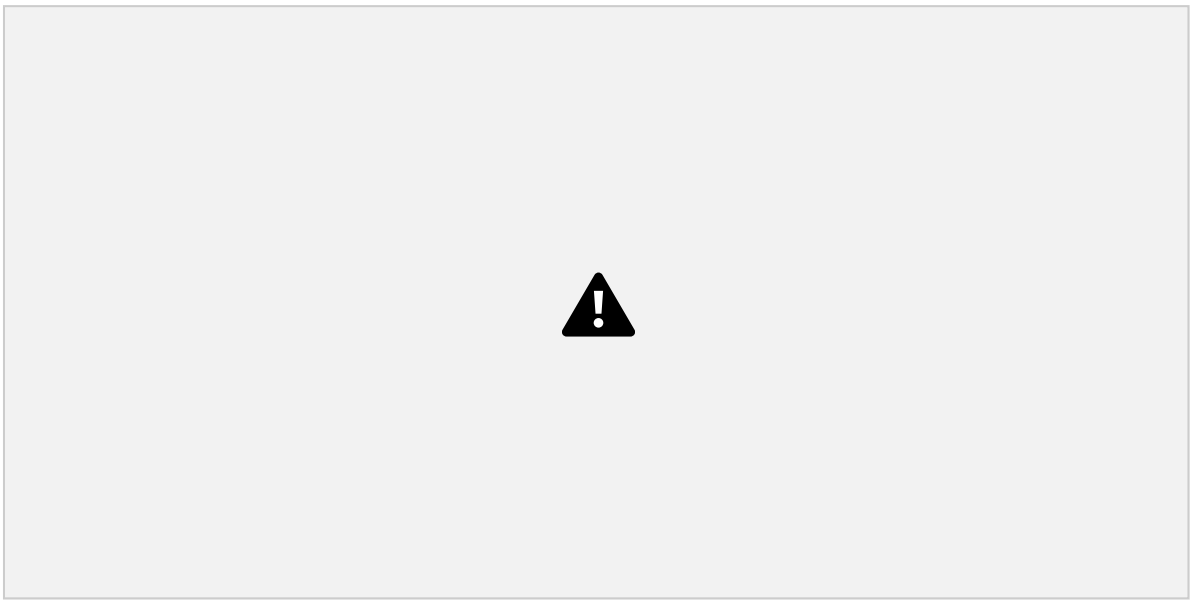
select * from student;



35. Now creating employee.csv file.
gedit employee.csv

BDT - Practical_7

Roll No:14
MSc Part-I Sem-2



Querying :

36. Creating new database for performing querying operations. Create database hiveql;

37. Using database hiveql and creating table employee inside the hiveql datanase.

create table employee(ID int, Name string, Department string, YOJ int, Salary float)

- **row format delimited**
- **fields terminated by ‘,’;**
- **tblproperties(“skip.header.line.count” =”1”);**

Now we will see the schema of employee table using below command, describe employee;

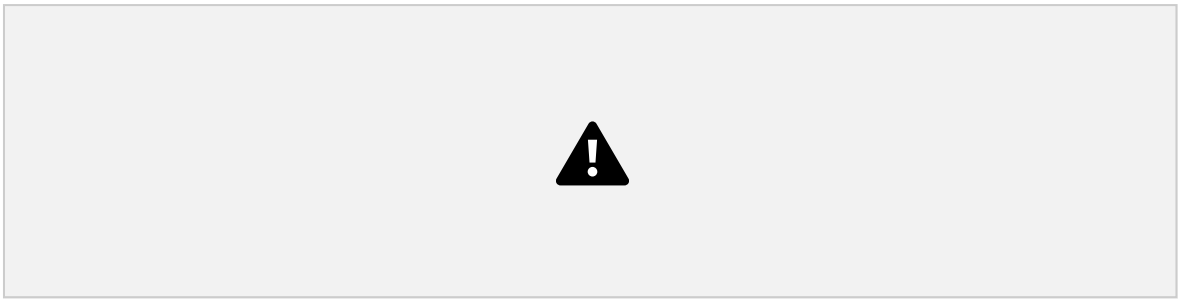
BDT - Practical_7

Name: Pooja Pathak
Roll No:14

38. Loading the data into employee table from employee.csv file which we have created
earlier and it is present in /home/cloudera.
load data local inpath '/home/cloudera/employee.csv' into table Employee

Displaying the table using below command,

select * from employee;



Now we Perform some operations on this employee table. 39. We are printing or displaying the rows or records of employees whose salary is greater
than and equal to 25000.
select * from employee where salary >=25000;

BDT - Practical_7

Name: Pooja Pathak
Roll No:14
MSc Part-I Sem-2

The Rose and Mike are two employees whose salaries are greater than and equal to 25000.

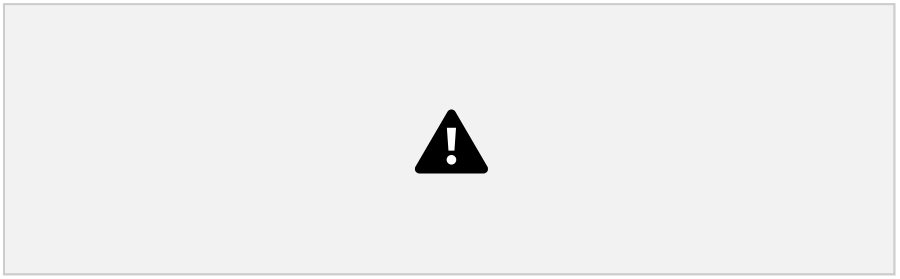
40. Now we are printing or displaying the rows or records of employees whose salary is less than 25000.

select * from employee where salary <25000;

Aggregating

41.Arithmetic operations:

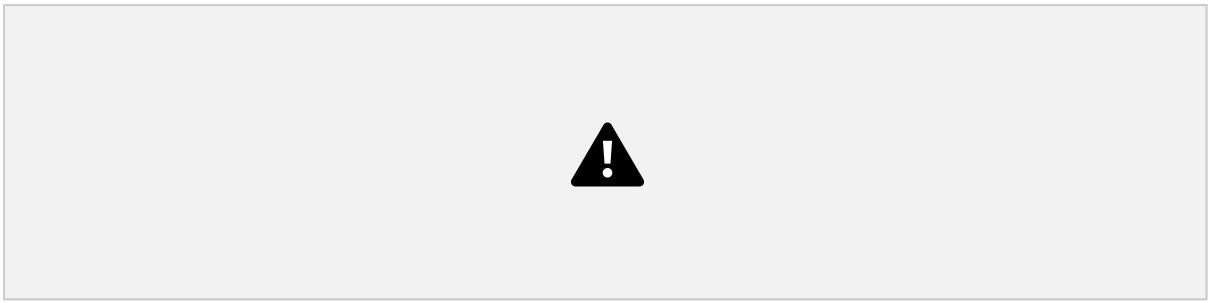
We are adding 5000 in existing salary and displaying specific columns using below command,



42. Displaying the maximum salary using below command, select max(salary) from employee;

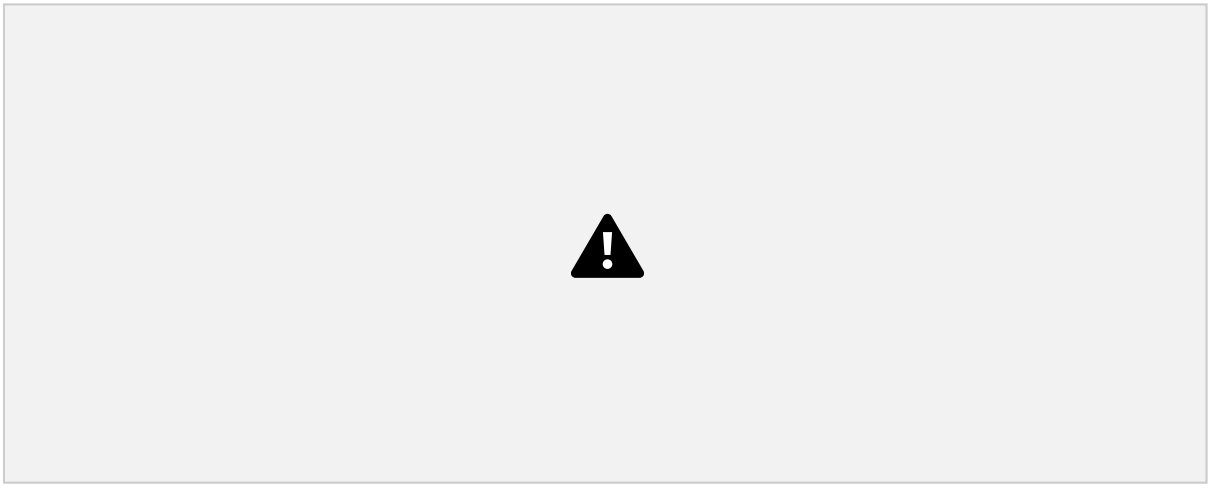
BDT - Practical_7

Name: Pooja Pathak
Roll No:14
MSc Part-I Sem-2



Here we can see that 26000 is maximum salary.

43. Displaying the minimum salary using below command, select min(salary) from employee;

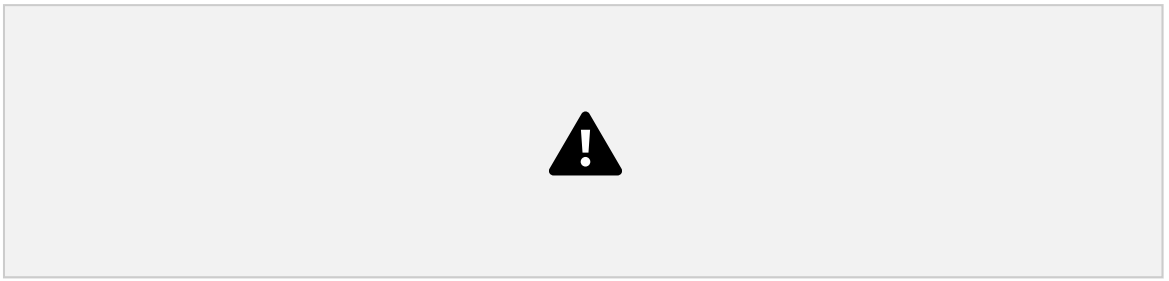


Here we can see that 22000 is minimum salary.

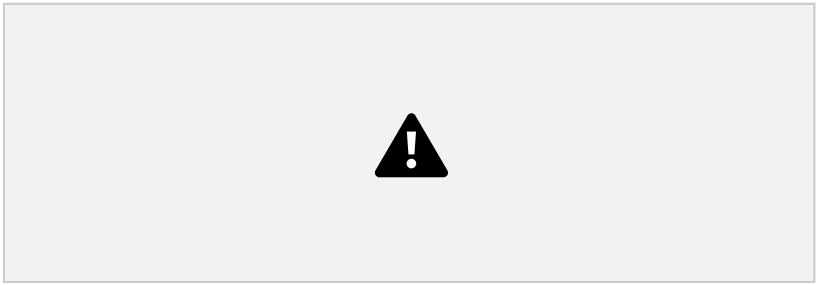
44. Now finding the square root of salary using below command, Select ID, name, sqrt(salary) from employee;

BDT - Practical_7

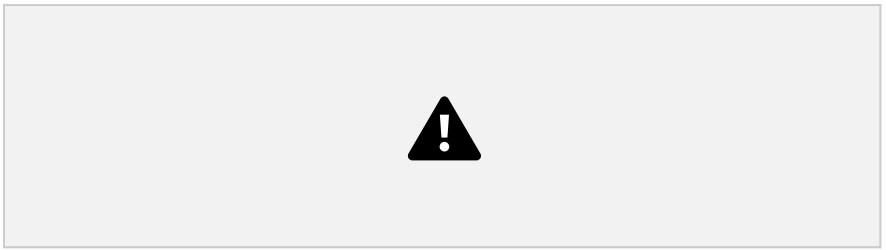
Name: Pooja Pathak
Roll No:14
MSc Part-I Sem-2



45. Now we are showing the name column in upper case using below command,
select ID, upper(name) from employee;



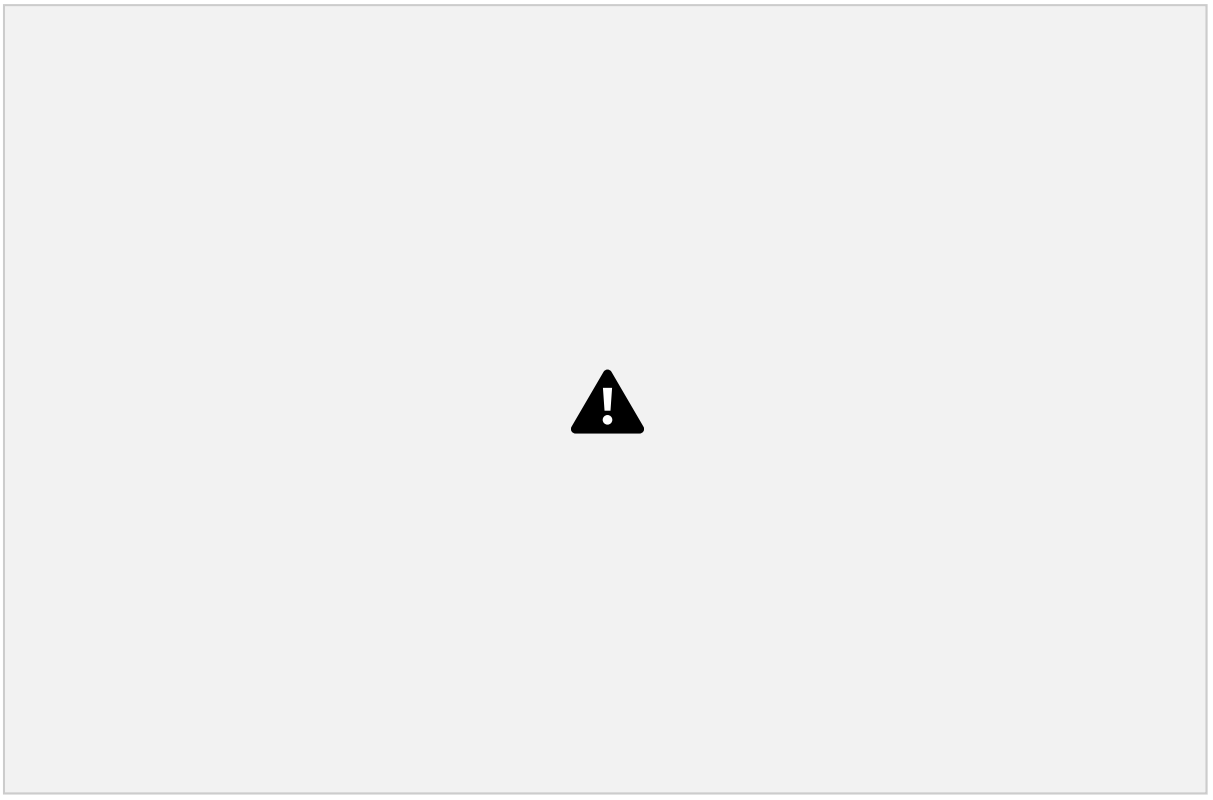
46. Now we are showing the name column in lower case using below command,
select ID, lower(name) from employee;

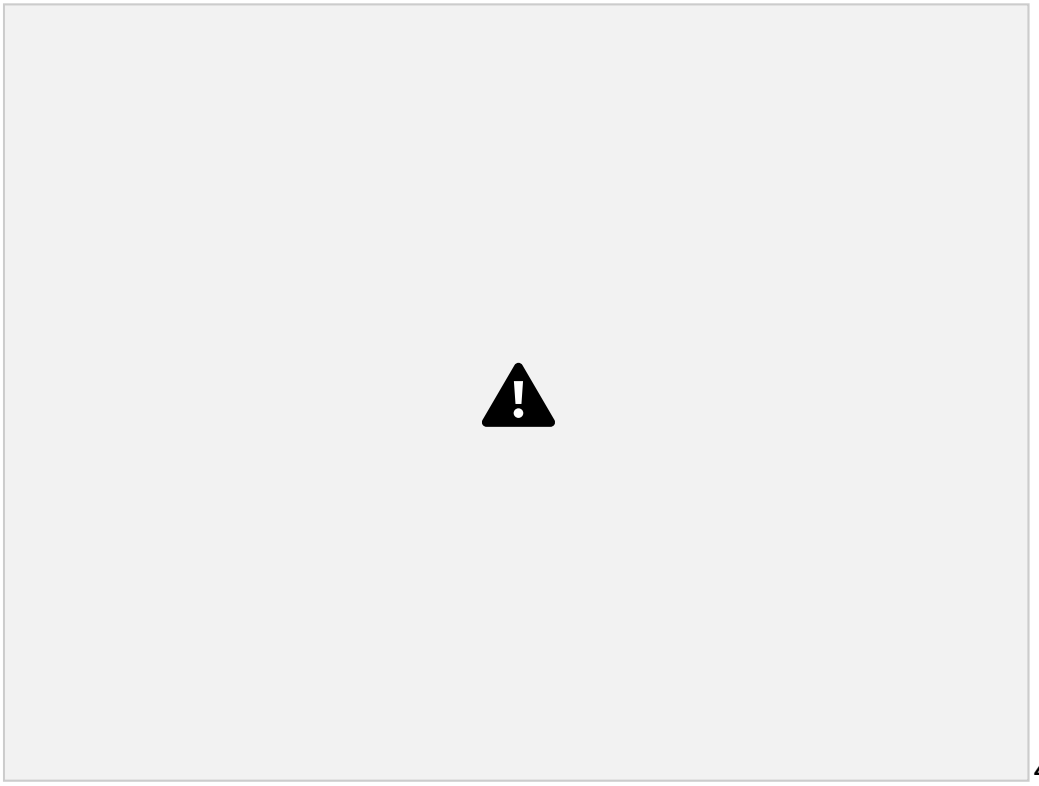


47. Creating another employee2.csv file with the same data as employee.csv but adding one
more column as Country to it.

BDT - Practical_7

Name: Pooja Pathak
Roll No:14
MSc Part-I Sem-2





48. Creating a new table as empgroup.

BDT - Practical_7

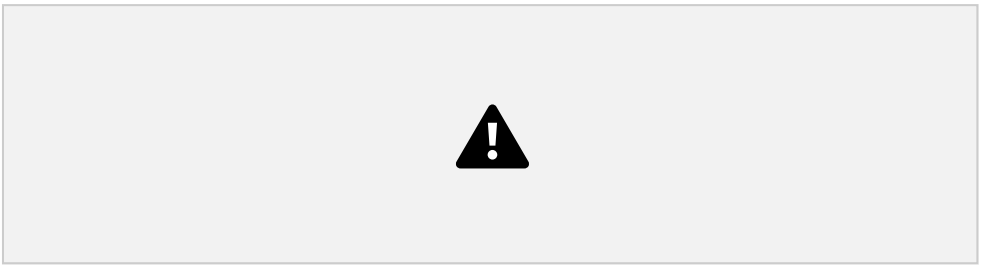
Name: Pooja Pathak
Roll No:14
MSc Part-I Sem-2

create table empgroup (ID int, Name string, Dept string, yoj int, salary float,
Country string)

- **row format delimited**
- **fields terminated by ‘,’;**
- **tblproperties(“skip.header.line.count” =”1”);**

49. Loading the data into empgroup table from employee2.csv file which we have created
and it is present in /home/cloudera/Documents directory. load data local inpath ‘/home/cloudera/Documents/employee2.csv’ into table
Empgroup

Displaying the table using below command,
select * from empgroup;

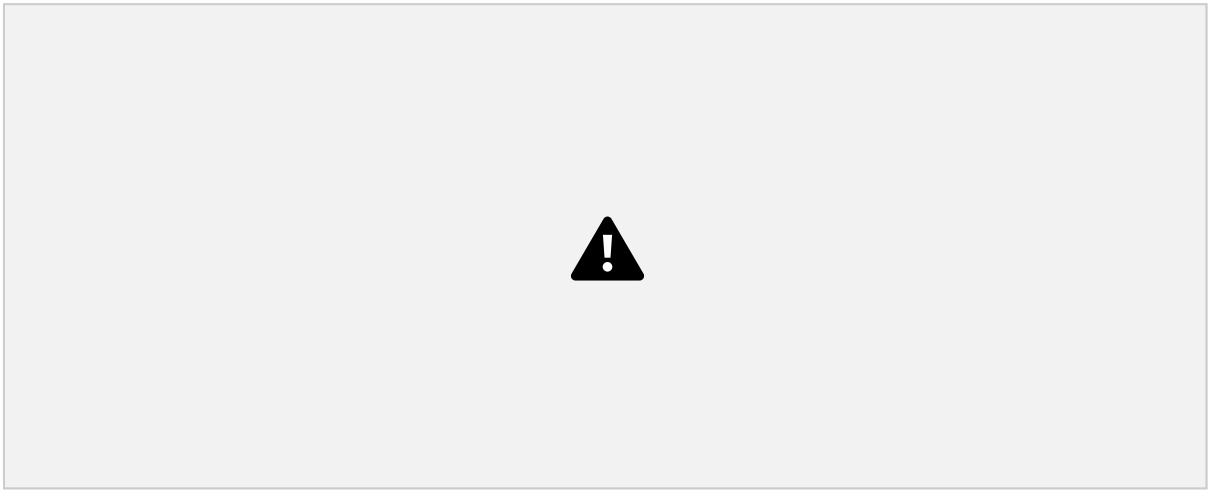


50.Groupby clause
Now we display the total sum of salary of employees country wise using below
command,
select country, sum(salary) from empgroup group by

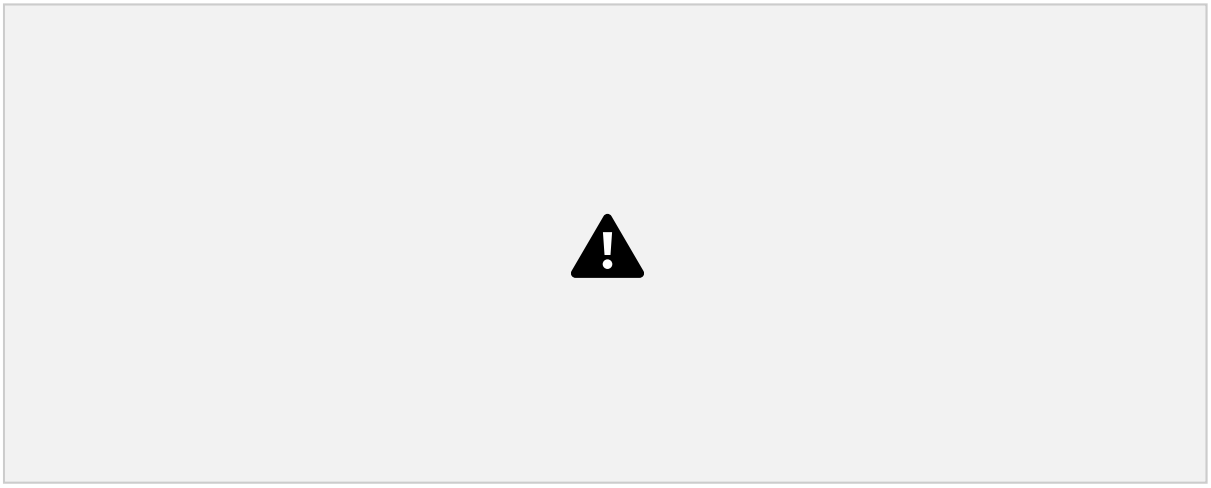
country; BDT - Practical_7

Name: Pooja Pathak

Roll No:14
MSc Part-I Sem-2

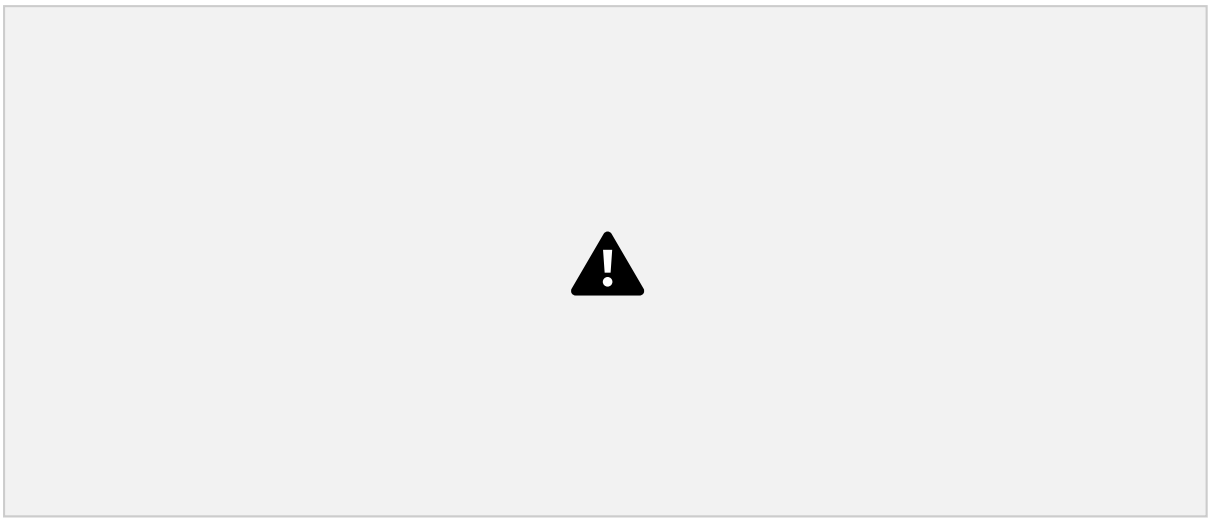


51. Groupby clause along with the having clause
Taking the total sum of salary countrywise using groupby clause and from that selecting
or displaying those country whose total sum of salary>50000 using having clause.
select country, sum(salary) from empgroup group by country having sum(salary)>50000;



Sorting : Order by
52. Now we are displaying the table by sorting the salary in descending order.
Select * from empgroup order by salary desc;

BDT - Practical_7
Name: Pooja Pathak
Roll No:14



53. Instead of order by if we have sort by,
Select * from empgroup sort by salary desc;