1. **Run PIG-**

```
[training@localhost ~]$ pig
2019-08-07 02:49:53,549 [main] INFO  org.apache.pig.Main - Logging error messages to:
/home/training/pig_1565171393548.log
2019-08-07 02:49:53,954 [main]
INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to
hadoop file system at: hdfs://localhost:8020
2019-08-07 02:49:54,145 [main]
INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to map-
reduce job tracker at: localhost:8021
```

2. **fs -ls-**

```
grunt> fs -ls
Found 16 items
-rw-r--r--   1 training supergroup       1390 2015-10-02 20:20 /user/training/Books
drwxr-xr-x   - training supergroup          0 2014-08-17 03:5/user/training/WeatherData
drwxr-xr-x   - training supergroup          0 2015-10-17 02:34 /user/training/_sqoop
-rw-r--r--   1 training supergroup       2944 2015-09-26 02:37 /user/training/bookinfo
drwxr-xr-x   - training supergroup          0 2015-09-20 01:38/user/training/class2_dir1
-rw-r--r--   1 training supergroup         81 2015-09-18 21:05 /user/training/deptinfo
drwxr-xr-x   - training supergroup          0 2014-08-17 01:59 /user/training/dir1
-rw-r--r--   1 training supergroup       1390 2015-10-02 20:22 /user/training/dirbook
drwxr-xr-x   - training supergroup          0 2015-10-17 02:14 /user/training/emp
drwxr-xr-x   - training supergroup          0 2015-10-17 02:34 /user/training/emp2
drwxr-xr-x   - training supergroup          0 2015-10-10 23:18 /user/training/emp_dir
drwxr-xr-x   - training supergroup          0 2015-10-17 02:26 /user/training/employee
drwxr-xr-x   - training supergroup          0 2015-10-17 02:01 /user/training/student
-rw-r--r--   1 training supergroup         86 2015-10-10 03:40 /user/training/table2
drwxr-xr-x   - training supergroup          0 2015-09-12 02:33 /user/training/user
-rw-r--r--   1 training supergroup         59 2015-10-10 00:56 /user/training/wordcount
```

3. **copyFromLocal-**

```
[training@localhost ~]$ cat skstudent.txt
1,sushil, 68.2,F,male
2,deepak,70.33,F,male
3,siddhesh, 80.21,F,male
4,vikas,70.3,F,male
5,sumit,62.00,F,male
6,ankita,68.00,F,female

[training@localhost ~]$ hadoop fs -copyFromLocal skstudent.txt students.txt
[training@localhost ~]$ pig

grunt> cat students.txt
1,sushil, 68.2,F,male
2,deepak,70.33,F,male
3,siddhesh, 80.21,F,male
4,vikas,70.3,F,male
5,sumit,62.00,F,male
6,ankita,68.00,F,femalegrunt>
```

4. **Dump-**

```
grunt> student = load '/user/training/students.txt' USING PigStorage(',') as
(rollno:int,name:chararray,percentage:float,sex:chararray);
```

```
grunt> dump student;
(1,sushil,68.2,F)
(2,deepak,70.33,F)
(3,siddhesh,80.21,F)
(4,vikas,70.3,F)
(5,sumit,62.0,F)
(6,ankita,68.0,F)
```

5. Projection-

```
grunt> Studentname = foreach student generate name;
grunt> Dump Studentname;
(sushil)
(deepak)
(siddhesh)
(vikas)
(sumit)
(ankita)
```

6. Join-

```
[training@localhost ~]$ cat > dept.txt
10,sushil,engineer,50000
10,nikhil,engineer,45000
10,supriya,engineer,50000
10,siddesh,engineer,44000
20,manish,clerk,20000
20,mahesh,clerk,25000
30,minal,scientist,60000
30,krishna,scientist,80000
30,govind,scientist,60000
50,rahul,chemist,80000
40,neha,biochemist,90000
[training@localhost ~]$ hadoop fs -copyFromLocal dept.txt department.txt
grunt> dept = load '/user/training/department.txt' USING PigStorage(',') as
(id:int,name:chararray,deptname:chararray,sal:int);
grunt> dump dept;
 (10,sushil,engineer,50000)
(10,nikhil,engineer,45000)
(10,supriya,engineer,50000)
(10,siddesh,engineer,44000)
(20,manish,clerk,20000)
(20,mahesh,clerk,25000)
(30,minal,scientist,60000)
(30,krishna,scientist,80000)
(30,govind,scientist,60000)
(50,rahul,chemist,80000)
(40,neha,biochemist,90000)
```

7. Relational Operators

a. Cross: The cross operator is userd to calculate the cross product of two or more relations.

```
grunt> x = cross student,dept;
grunt> dump x;
 (6,ankita,68.0,F,10,sushil,engineer,50000)
(6,ankita,68.0,F,20,manish,clerk,20000)
(6,ankita,68.0,F,10,siddesh,engineer,44000)
(6,ankita,68.0,F,10,nikhil,engineer,45000)
```

```
(6,ankita,68.0,F,50,rahul,chemist,80000)
(6,ankita,68.0,F,30,govind,scientist,60000)
(6,ankita,68.0,F,30,minal,scientist,60000)
(6,ankita,68.0,F,10,supriya,engineer,50000)
(6,ankita,68.0,F,20,mahesh,clerk,25000)
(6,ankita,68.0,F,40,neha,biochemist,90000)
(6,ankita,68.0,F,30,krishna,scientist,80000)
(2,deepak,70.33,F,10,siddesh,engineer,44000)
(2,deepak,70.33,F,20,manish,clerk,20000)
(2,deepak,70.33,F,10,sushil,engineer,50000)
(4,vikas,70.3,F,10,siddesh,engineer,44000)
(4,vikas,70.3,F,20,manish,clerk,20000)
(4,vikas,70.3,F,10,sushil,engineer,50000)
(3,siddhesh,80.21,F,10,siddesh,engineer,44000)
(3,siddhesh,80.21,F,20,manish,clerk,20000)
(3,siddhesh,80.21,F,10,sushil,engineer,50000)
(2,deepak,70.33,F,10,nikhil,engineer,45000)
(3,siddhesh,80.21,F,10,nikhil,engineer,45000)
(4,vikas,70.3,F,10,nikhil,engineer,45000)
(2,deepak,70.33,F,50,rahul,chemist,80000)
(2,deepak,70.33,F,30,govind,scientist,60000)
(3,siddhesh,80.21,F,50,rahul,chemist,80000)
(3,siddhesh,80.21,F,30,govind,scientist,60000)
(4,vikas,70.3,F,50,rahul,chemist,80000)
(4,vikas,70.3,F,30,govind,scientist,60000)
(2,deepak,70.33,F,30,minal,scientist,60000)
(2,deepak,70.33,F,10,supriya,engineer,50000)
(4,vikas,70.3,F,30,minal,scientist,60000)
(4,vikas,70.3,F,10,supriya,engineer,50000)
(3,siddhesh,80.21,F,30,minal,scientist,60000)
(3,siddhesh,80.21,F,10,supriya,engineer,50000)
(4,vikas,70.3,F,20,mahesh,clerk,25000)
(3,siddhesh,80.21,F,20,mahesh,clerk,25000)
(2,deepak,70.33,F,20,mahesh,clerk,25000)
(3,siddhesh,80.21,F,40,neha,biochemist,90000)
(4,vikas,70.3,F,40,neha,biochemist,90000)
(2,deepak,70.33,F,40,neha,biochemist,90000)
(2,deepak,70.33,F,30,krishna,scientist,80000)
(3,siddhesh,80.21,F,30,krishna,scientist,80000)
(4,vikas,70.3,F,30,krishna,scientist,80000)
(5,sumit,62.0,F,10,siddesh,engineer,44000)
(5,sumit,62.0,F,20,manish,clerk,20000)
(5,sumit,62.0,F,10,sushil,engineer,50000)
(1,sushil,68.2,F,10,siddesh,engineer,44000)
(1,sushil,68.2,F,20,manish,clerk,20000)
(1,sushil,68.2,F,10,sushil,engineer,50000)
(5,sumit,62.0,F,10,nikhil,engineer,45000)
(1,sushil,68.2,F,10,nikhil,engineer,45000)
(1,sushil,68.2,F,30,govind,scientist,60000)
(1,sushil,68.2,F,50,rahul,chemist,80000)
(5,sumit,62.0,F,30,govind,scientist,60000)
(5,sumit,62.0,F,50,rahul,chemist,80000)
(5,sumit,62.0,F,10,supriya,engineer,50000)
(5,sumit,62.0,F,30,minal,scientist,60000)
(1,sushil,68.2,F,10,supriya,engineer,50000)
```

```
(1,sushil,68.2,F,30,minal,scientist,60000)
(1,sushil,68.2,F,20,mahesh,clerk,25000)
(5,sumit,62.0,F,20,mahesh,clerk,25000)
(1,sushil,68.2,F,40,neha,biochemist,90000)
(5,sumit,62.0,F,40,neha,biochemist,90000)
(5,sumit,62.0,F,30,krishna,scientist,80000)
(1,sushil,68.2,F,30,krishna,scientist,80000)
```

b. Distinct: This operator is used to remove the duplicate tuples in a relation. It does not preserve the original order of the contents.

```
Distinct A = load '/user/training/dummy.txt' USING PigStorage(',') as
(a1:int,a2:int,a3:int,a4:int);
grunt> A = load '/user/training/dumA.txt' USING PigStorage(',') as (a1:int,a2:int,a3:int,a4:int);
grunt> dump A;
(1,2,2,3)
(2,4,5,65)
(2,4,5,65)
(23,44,3,2)
(23,44,3,2)
(1,2,3,4)
(4,3,2,1)
(5,6,7,8)
grunt> z = distinct A;
grunt> dump z;
 (1,2,2,3)
(1,2,3,4)
(2,4,5,65)
(4,3,2,1)
(5,6,7,8)
(23,44,3,2)
```

c. Filter: This operator can be used to select the required data based on someconditions

```
grunt> y = FILTER A By a2 == 2;
grunt> dump y;
 (1,2,2,3)
(1,2,3,4)
grunt> y2 = FILTER dept By deptname == 'engineer';
grunt> dump y2;
 (10,sushil,engineer,50000)
(10,nikhil,engineer,45000)
(10,supriya,engineer,50000)
(10,siddesh,engineer,44000)
```

d. FOREACH:- This operator is used to generate data transformation based on column data

```
grunt> X = Foreach dept GENERATE id,name;
grunt> dump X;
(10,sushil)
(10,nikhil)
(10,supriya)
(10,siddesh)
(20,manish)
(20,mahesh)
(30,minal)
(30,krishna)
```

```
(30,govind)
(50,rahul)
(40,neha)
```

e. COGROUP operator:-

```
grunt> A = load '/user/training/dumA.txt' USING PigStorage(',') as (a1:int,a2:int,a3:int,a4:int);
grunt> dump A;
 (1,2,2,3)
(2,4,5,65)
(2,4,5,65)
(23,44,3,2)
(23,44,3,2)
(1,2,3,4)
(4,3,2,1)
(5,6,7,8)
grunt> copyFromLocal /home/training/dumB.txt /user/training/
grunt> B = load '/user/training/dumB.txt' USING PigStorage(',') as (b1:int,
b2:int);
grunt> dump B;
 (2,3)
(4,5)
(5,6)
(4,3)
(2,3)
(3,7)
grunt> C = COGROUP A By a1 inner, B BY b1 Inner;
grunt> dump C;
 (2,{(2,4,5,65),(2,4,5,65)},{(2,3),(2,3)})
(4,{(4,3,2,1)},{(4,5),(4,3)})
(5,{(5,6,7,8)},{(5,6)})
```

8. Nested Projection:

```
grunt> Z = FOREACH C GENERATE group, B.b2;
grunt> dump Z;
(2,{(3),(3)})
(4,{(5),(3)})
(5,{(6)})
grunt> Z = FOREACH C GENERATE group,A.(a1,a2);
grunt> dump Z;
(2,{(2,4),(2,4)})
(4,{(4,3)})
(5,{(5,6)})
```