UNIT - VI

1. Which of the following scripts that generate more than three MapReduce jobs ?

a)

```
a = load '/mapred/history/done' using HadoopJobHistoryLoader() as (j:map[], m:map[], r:m
ap[]);
b = group a by (j#'PIG_SCRIPT_ID', j#'USER', j#'JOBNAME');
c = for b generate group.$1, group.$2, COUNT(a);
d = filter c by $2 > 3;
dump d;
```

b)

```
a = load '/mapred/history/done' using HadoopJobHistoryLoader() as (j:map[], m:map[], r:m
ap[]);
b = display a by (j#'PIG_SCRIPT_ID', j#'USER', j#'JOBNAME');
c = foreach b generate group.$1, group.$2, COUNT(a);
d = filter c by $2 > 3;
dump d;
```

c)

```
a = load '/mapred/history/done' using HadoopJobHistoryLoader() as (j:map[], m:map[], r:m
ap[]);
b = group a by (j#'PIG_SCRIPT_ID', j#'USER', j#'JOBNAME');
c = foreach b generate group.$1, group.$2, COUNT(a);
d = filter c by $2 > 3;
dump d;
```

d) None of the mentioned

2. Point out the correct statement :

a) LoadPredicatePushdown is same as LoadMetadata.setPartitionFilter

b) getOutputFormat() is called by Pig to get the InputFormat used by the loader

c) Pig works with data from many sources

d) None of the mentioned

3. Which of the following find the running time of each script (in seconds) ?

a)

```
a = load '/mapred/history/done' using HadoopJobHistoryLoader() as (j:map[], m:map[], r:m
ap[]);
```

```
b = foreach a generate j#'PIG_SCRIPT_ID' as id, j#'USER' as user, j#'JOBNAME' as script_na
me,
        (Long) j#'SUBMIT_TIME' as start, (Long) j#'FINISH_TIME' as end;
c = group b by (id, user, script_name)
d = foreach c generate group.user, group.script_name, (MAX(b.end) - MIN(b.start)/1000;
dump d;
```

b)

```
a = load '/mapred/history/done' using HadoopJobHistoryLoader() as (j:map[], m:map[], r:m
ap[]);
b = for a generate j#'PIG_SCRIPT_ID' as id, j#'USER' as user, j#'JOBNAME' as script_name,
        (Long) j#'SUBMIT_TIME' as start, (Long) j#'FINISH_TIME' as end;
c = group b by (id, user, script_name)
d = for c generate group.user, group.script_name, (MAX(b.end) - MIN(b.start)/1000;
dump d;
```

c)

```
a = load '/mapred/history/done' using HadoopJobHistoryLoader() as (j:map[], m:map[], r:m
ap[]);
b = foreach a generate j#'PIG_SCRIPT_ID' as id, j#'USER' as user, j#'QUEUE_NAME' as queu
e;
c = group b by (id, user, queue) parallel 10;
d = foreach c generate group.user, group.queue, COUNT(b);
dump d;
```

d) All of the mentioned

4. Which of the following script determines the number of scripts run by user and queue on a cluster:

a)

```
a = load '/mapred/history/done' using HadoopJobHistoryLoader() as (j:map[], m:map[], r:m
ap[]);

b = foreach a generate (Chararray) j#'STATUS' as status, j#'PIG_SCRIPT_ID' as id, j#'USER'
as user, j#'JOBNAME' as script_name, j#'JOBID' as job;
c = filter b by status != 'SUCCESS';
dump c;
```

b)

```
a = load '/mapred/history/done' using HadoopJobHistoryLoader() as (j:map[], m:map[], r:m
ap[]);

b = foreach a generate j#'PIG_SCRIPT_ID' as id, j#'USER' as user, j#'JOBNAME' as script_na
me, (Long) r#'NUMBER_REDUCES' as reduces;
c = group b by (id, user, script_name) parallel 10;
```

```
d = foreach c generate group.user, group.script_name, MAX(b.reduces) as max_reduces;
e = filter d by max_reduces == 1;
dump e;
```

c)

```
a = load '/mapred/history/done' using HadoopJobHistoryLoader() as (j:map[], m:map[], r:m
ap[]);
b = foreach a generate j#'PIG_SCRIPT_ID' as id, j#'USER' as user, j#'QUEUE_NAME' as queu
e;
c = group b by (id, user, queue) parallel 10;
d = foreach c generate group.user, group.queue, COUNT(b);
dump d;
```

d) None of the mentioned

5. Point out the wrong statement :

a) Pig can invoke code in language like Java Only.

b) Pig enables data workers to write complex data transformations without knowing Java.

c) Pig's simple SQL-like scripting language is called Pig Latin, and appeals to developers already familiar with scripting languages and SQL.

d) Pig is complete, so you can do all required data manipulations in Apache Hadoop with Pig.

6. Which of the following script is used to check scripts that have failed jobs ?

a)

```
a = load '/mapred/history/done' using HadoopJobHistoryLoader() as (j:map[], m:map[], r:m
ap[]);
b = foreach a generate (Chararray) j#'STATUS' as status, j#'PIG_SCRIPT_ID' as id, j#'USER'
as user, j#'JOBNAME' as script_name, j#'JOBID' as job;
c = filter b by status != 'SUCCESS';
dump c;
```

b)

```
a = load '/mapred/history/done' using HadoopJobHistoryLoader() as (j:map[], m:map[], r:m
ap[]);
b = foreach a generate j#'PIG_SCRIPT_ID' as id, j#'USER' as user, j#'JOBNAME' as script_na
me, (Long) r#'NUMBER_REDUCES' as reduces;
c = group b by (id, user, script_name) parallel 10;
d = foreach c generate group.user, group.script_name, MAX(b.reduces) as max_reduces;
e = filter d by max_reduces == 1;
dump e;
```

c)

```
a = load '/mapred/history/done' using HadoopJobHistoryLoader() as (j:map[], m:map[], r:m
ap[]);
b = foreach a generate j#'PIG_SCRIPT_ID' as id, j#'USER' as user, j#'QUEUE_NAME' as queu
e;
c = group b by (id, user, queue) parallel 10;
d = foreach c generate group.user, group.queue, COUNT(b);
dump d;
```

d) None of the mentioned

7. Which of the following code is used to find scripts that use only the default parallelism ?

a)

```
a = load '/mapred/history/done' using HadoopJobHistoryLoader() as (j:map[], m:map[], r:m
ap[]);
b = foreach a generate (Chararray) j#'STATUS' as status, j#'PIG_SCRIPT_ID' as id, j#'USER'
as user, j#'JOBNAME' as script_name, j#'JOBID' as job;
c = filter b by status != 'SUCCESS';
dump c;
```

b)

```
a = load '/mapred/history/done' using HadoopJobHistoryLoader() as (j:map[], m:map[], r:m
ap[]);
b = foreach a generate j#'PIG_SCRIPT_ID' as id, j#'USER' as user, j#'JOBNAME' as script_na
me, (Long) r#'NUMBER_REDUCES' as reduces;
c = group b by (id, user, script_name) parallel 10;
d = foreach c generate group.user, group.script_name, MAX(b.reduces) as max_reduces;
e = filter d by max_reduces == 1;
dump e;
```

c)

```
a = load '/mapred/history/done' using HadoopJobHistoryLoader() as (j:map[], m:map[], r:m
ap[]);
b = foreach a generate j#'PIG_SCRIPT_ID' as id, j#'USER' as user, j#'QUEUE_NAME' as queu
e;
c = group b by (id, user, queue) parallel 10;
d = foreach c generate group.user, group.queue, COUNT(b);
dump d;
```

d) None of the mentioned

8. Pig Latin is _____ and fits very naturally in the pipeline paradigm while SQL is instead declarative.

a) functional

b) procedural

c) declarative

d) All of the mentioned

9. In comparison to SQL, Pig uses :

a) lazy evaluation

b) ETL

c) Supports pipeline splits

d) All of the mentioned

10. Which of the following is an entry in jobconf ?

a) pig.job

b) pig.input.dirs

c) pig.feature

d) None of the mentioned