1. Create a database called movie_details and use it

```
hive> create database movie_details
. . > ;
OK
```

```
hive> use movie_details;
OK
```

2. Create tables to hold the movie and ratings data in Hive.

```
hive> create table movies (
. . > id bigint,
. . > title string,
. . > genres string
. . > )
. . > row format delimited fields terminated by ','
. . > location '/movies';
OK
```

```
hive> create table ratings (
. . > user_id bigint,
. . > movie_id bigint,
. . > ratings bigint,
. . > time bigint
. . > )
. . > row format delimited fields terminated by ','
. . > location '/ratings';
OK
```

3. Display 20 records of movies and ratings

```
hive> select * from movies limit 20;
OK
1  Toy Story (1995)  Animation|Children's|Comedy
2  Jumanji (1995)  Adventure|Children's|Fantasy
3  Grumpier Old Men (1995)  Comedy|Romance
4  Waiting to Exhale (1995)  Comedy|Drama
5  Father of the Bride Part II (1995)  Comedy
6  Heat (1995)  Action|Crime|Thriller
7  Sabrina (1995)  Comedy|Romance
8  Tom and Huck (1995)  Adventure|Children's
9  Sudden Death (1995)  Action
10  GoldenEye (1995)  Action|Adventure|Thriller
11  American President    The (1995)
12  Dracula: Dead and Loving It (1995)  Comedy|Horror
13  Balto (1995)  Animation|Children's
14  Nixon (1995)  Drama
15  Cutthroat Island (1995)  Action|Adventure|Romance
16  Casino (1995)  Drama|Thriller
17  Sense and Sensibility (1995)  Drama|Romance
18  Four Rooms (1995)  Thriller
19  Ace Ventura: When Nature Calls (1995)  Comedy
20  Money Train (1995)  Action
```

```
hive> select * from ratings limit 20;
OK
1 1193 5 978300760
1 661 3 978302109
1 914 3 978301968
1 3408 4 978300275
1 2355 5 978824291
1 1197 3 978302268
1 1287 5 978302039
1 2804 5 978300719
1 594 4 978302268
1 919 4 978301368
1 595 5 978824268
1 938 4 978301752
1 2398 4 978302281
1 2918 4 978302124
1 1035 5 978301753
1 2791 4 978302188
1 2687 3 978824268
1 2018 4 978301777
1 3105 5 978301713
1 2797 4 978302039
```

4. Find out number of non-adults as per Indian standard, who has rated movies.

```
hive> sel
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different
execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = rahul_20191121093720_438e40fd-7205-4d10-b513-936a77c2005c
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 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>
Starting Job = job_1574305760138_0002, Tracking URL = http://Jarvis:8088/proxy/application_1574305760138_0002/
Kill Command = C:\BigData\hadoop-2.9.1\bin\hadoop.cmd job  -kill job_1574305760138_0002
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different
execution engine (i.e. spark, tez) or using Hive 1.X releases.
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2019-11-21 09:37:45,700 Stage-1 map = 0%,  reduce = 0%
2019-11-21 09:37:59,779 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.983 sec
2019-11-21 09:38:12,409 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 10.075 sec
MapReduce Total cumulative CPU time: 10 seconds 75 msec
Ended Job = job_1574305760138_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 10.075 sec   HDFS Read: 118816 HDFS Write: 103 SUCCESS
Total MapReduce CPU Time Spent: 10 seconds 75 msec
OK
222
```

5. Find the age of the most rated user with counts of rating.

```
hive> select u.id, u.age, a.count from users u join ( select r.user_id, count(ratings) count from ratings r group by (r.
user_id) order by count DESC limit 1) a where u.id = a.user_id;
```

```
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 11.62 sec   HDFS Read: 21601067 HDFS Write: 131414 SUCCESS
Stage-Stage-3: Map: 1  Reduce: 1   Cumulative CPU: 8.982 sec   HDFS Read: 136139 HDFS Write: 119 SUCCESS
Stage-Stage-4: Map: 1   Cumulative CPU: 4.077 sec   HDFS Read: 5666 HDFS Write: 112 SUCCESS
Total MapReduce CPU Time Spent: 24 seconds 679 msec
OK
4169 50 2314
1 row selected (159.806 seconds)
```

6. Find the count of the ratings

```
hive> select count(r.ratings) from ratings r, users u where u.id = r.user_id group by u.age;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different
execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID
Total jobs = 1
```

```
MapReduce Total cumulative CPU time: 12 seconds 543 msec
Ended Job = job_1574305760138_0013
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 12.543 sec   HDFS Read: 21606476 HDFS Write: 216 SUCCESS
Total MapReduce CPU Time Spent: 12 seconds 543 msec
OK
27211
183536
395556
199003
83633
72490
38780
```

7. Find the movie with the maximum number of ratings.

```
hive> select count(r.ratings) from movies m, ratings r where m.id=r.movie_id;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different
execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID
```

```
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 14.136 sec   HDFS Read: 21605441 HDFS Write: 107 SUCCESS
Total MapReduce CPU Time Spent: 14 seconds 136 msec
OK
1000209
```

8. Find the movie with the lowest rating.

```
hive> select count(r.ratings) from movies m, ratings r where m.id=r.movie_id and r.ratings=1;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different
execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID
Total jobs = 1
```

```
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 12.574 sec   HDFS Read: 21605685 HDFS Write: 105 SUCCESS
Total MapReduce CPU Time Spent: 12 seconds 574 msec
OK
56174
1 row selected (60.685 seconds)
```

10. Create a database called user details and use it.

```
hive> create database user_details;
OK
No rows affected (2.635 seconds)
hive> use user_details;
OK
```

11. Create tables to hold user and occupation data in Hive.

```
hive> create table occupation (
. . > id bigint,
. . > occupation string
. . > )
. . > row format delimited fields terminated by ','
. . > location '/occupation';
OK
```

```
hive> create table users (
. . > id bigint,
. . > gender string,
. . > age bigint,
. . > occupation bigint,
. . > zip bigint
. . > )
. . > row format delimited fields terminated by ','
. . > location '/users';
OK
```

12. Find out occupation of all the users.

```
hive> select u.*, o.occupation from users u, occupation o where u.occupation=o.id limit 10;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different
execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID
Total jobs = 1
```

```
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1   Cumulative CPU: 4.139 sec   HDFS Read: 11700 HDFS Write: 523 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 139 msec
OK
1 F 1 10 48067    K-12 student
2 M 56 16 70072    self-employed
3 M 25 15 55117    scientist
4 M 45 7 2460    executive/managerial
5 M 25 20 55455    writer
6 F 50 9 55117    homemaker
7 M 35 1 6810    academic/educator
8 M 25 12 11413    programmer
9 M 25 17 61614    technician/engineer
10 F 35 1 95370    academic/educator
```

13. Find out the no of users with same occupation and having age more than 25 along with occupation details.

```
hive> select o.occupation, count(1) from users u join occupation o where u.occupation=o.id AND u.age>24 group by o.occup
ation;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different
execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID
Total jobs = 1
```

```
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 11.466 sec   HDFS Read: 123298 HDFS Write: 753 SUCCESS
Total MapReduce CPU Time Spent: 11 seconds 466 msec
OK
   K-12 student  3
   academic/educator  479
   artist  220
   clerical/admin  155
   college/grad student  222
   customer service  94
   doctor/health care  227
   executive/managerial  660
   farmer  15
   homemaker  86
   lawyer  121
   other or not specified  578
   programmer  328
   retired  141
   sales/marketing  263
   scientist  130
   self-employed  223
   technician/engineer  448
   tradesman/craftsman  60
   unemployed  30
   writer  232
```

14. Find the occupation of all female users.

```
hive> select u.*, o.occupation from users u, occupation o where u.occupation=o.id AND u.gender='F' limit 10;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different
execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID                                                                    01
Total jobs = 1
```

```
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1   Cumulative CPU: 4.81 sec   HDFS Read: 11942 HDFS Write: 564 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 810 msec
OK
1 F 1 10 48067    K-12 student
6 F 50 9 55117    homemaker
10 F 35 1 95370    academic/educator
11 F 25 1 4093    academic/educator
16 F 35 0 20670    other or not specified
18 F 18 3 95825    clerical/admin
24 F 25 7 10023    executive/managerial
28 F 25 1 14607    academic/educator
30 F 35 7 19143    executive/managerial
32 F 25 0 19355    other or not specified
```

15. Find the number of female and male working as doctor/health care.

```
hive> select count(*) from users u, occupation o where u.occupation=o.id and o.occupation='  doctor/health care';
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different
execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID
Total jobs = 1
```