Lesson-4
Lesson-4 ICP

Source Code: https://umkc.box.com/s/e26jmxsli6nnbb9vy2pncvd4cfst2q96

Complete the following:

Name: Andrew Poitras
Email: ap3h7@mail.umkc.edu
vidoe link if available:

Write your report with screenshots here or in a PDF format file.
1. To start the ICP I had to create the petrol table and load the petrol.txt file into it using the following queries.

```
hive> create table petrol (distributor_id STRING, distributor_name STRING, amt_I
N STRING, amy_OUT STRING, vol_IN INT, vol_OUT INT, year INT) row format delimite
d fields terminated by ',' stored as textfile;
OK
Time taken: 0.166 seconds
```

```
hive> load data local inpath "/home/cloudera/Documents/petrol.txt"  into table p
etrol;
Loading data to table work.petrol
Table work.petrol stats: [numFiles=1, totalSize=19215]
OK
Time taken: 0.415 seconds
hive> show tables;
OK
petrol
Time taken: 0.065 seconds, Fetched: 1 row(s)
```

1-1. Next I showed the total amount of petrol in volume sold by each distributor using the following query
Query:

```
1 select distributor_name,
2 sum(vol_OUT)
3 from petrol
4 group by distributor_name;
```

Results:

	distributor_name	_c1
1	Bharat	83662
2	Distributer name	NULL
3	hindustan	71767
4	reliance	76558
5	shell	69266

1-2. I showed the top 10 distributor IDs along with the petrol sold by them.

Query:

```
select distributor_id,
vol_OUT
from petrol
order by vol_OUT desc limit 10;
```

Results:

	distributor_id	vol_out
1	T1A 9W4	899
2	S8W 0P4	899
3	V8U 2T6	898
4	O9P 9S3	897
5	O8A 6Z5	897
6	F6W 6H3	896
7	N5Q 8E5	895
8	E6O 9P1	895
9	M6S 1P4	895
10	J4M 4G3	895

1-3. I showed the bottom 10 distributor IDs along with the petrol sold by them.

Query:

```
select distributor_id,
vol_OUT
from petrol
order by vol_OUT limit 10;
```

Results:

	distributor_id	vol_out
1	District.ID	NULL
2	F4D 6K2	602
3	H7M 4M4	603
4	G9F 6U7	607
5	R3W 2E3	608
6	O5D 2R6	610
7	H4P 6A9	610
8	V0Z 0F6	612
9	O0D 0L1	612
10	W0M 8R7	612

1-4. I listed all distributors that had a volume in - volume out difference greater than 500 which turned out to be none of them as it is illegal.

Query:

```
select distributor_id,
year,
(vol_IN - vol_OUT) as difference
from petrol
where (vol_IN-vol_OUT)>500;
```

Results:

```
✔ Done. 0 results.
```

2. For question 2 I had to start by creating the olympic table and loading the olympic.csv file into it.
Query:

```
create table olympic (athlete STRING, age INT, country STRING, year STRING,
closing STRING, sport STRING, gold INT, silver INT, bronze INT, total INT)
row format delimited fields terminated by '\t' stored as textfile;
```

```
hive> load data local inpath "/home/cloudera/Documents/olympic_data.csv" into ta
ble olympic;
Loading data to table work.olympic
Table work.olympic stats: [numFiles=1, totalSize=518669]
OK
Time taken: 0.455 seconds
```

2-1. I listed the number of medals won by each country in swimming.
Query:

```
select country,
SUM(total)
from olympic
where sport = "Swimming"
group by country;
```

Results:

	country	_c1
1	Argentina	1
2	Australia	163
3	Austria	3
4	Belarus	2
5	Brazil	8
6	Canada	5
7	China	35
8	Costa Rica	2
9	Croatia	1
10	Denmark	1
11	France	39
12	Germany	32
13	Great Britain	11
14	Hungary	9
15	Italy	16
16	Japan	43
17	Lithuania	1
18	Netherlands	46

2-2. I displayed the real life number of medals India won year wise.
Query:

```
select year,
SUM(total)
from olympic
where country = "India"
group by year;
```

Results:

	year	_c1
1	2000	1
2	2004	1
3	2008	3
4	2012	6

2-3. I found the total number of medals that each country won.
Query:

```
select country,
SUM(total)
from olympic
group by country;
```

Results:

	country	_c1
1	Afghanistan	2
2	Algeria	8
3	Argentina	141
4	Armenia	10
5	Australia	609
6	Austria	91
7	Azerbaijan	25
8	Bahamas	24
9	Bahrain	1
10	Barbados	1
11	Belarus	97
12	Belgium	18
13	Botswana	1
14	Brazil	221
15	Bulgaria	41
16	Cameroon	20
17	Canada	370
18	Chile	22
19	China	530

2-4. I found the number of gold medals that each country won.
Query:

```
select country,
sum(gold)
from olympic
group by country;
```

Results:

	country	_c1
1	Afghanistan	0
2	Algeria	2
3	Argentina	49
4	Armenia	0
5	Australia	163
6	Austria	36
7	Azerbaijan	6
8	Bahamas	11
9	Bahrain	0
10	Barbados	0
11	Belarus	17
12	Belgium	2
13	Botswana	0
14	Brazil	46
15	Bulgaria	8
16	Cameroon	20
17	Canada	168
18	Chile	3
19	China	234

2-5. I found which country got medals for shooting with year wise classification.
Query:

```
select country,
total,
year
from olympic
where sport="Shooting"
order by year desc,
total desc;
```

Results:

	country	total	year
1	Italy	2	2012
2	South Korea	2	2012
3	Ukraine	2	2012
4	United States	1	2012
5	Qatar	1	2012
6	Kuwait	1	2012
7	Slovakia	1	2012
8	United States	1	2012
9	Poland	1	2012
10	Croatia	1	2012
11	China	1	2012
12	South Korea	1	2012
13	Belgium	1	2012
14	Sweden	1	2012
15	Slovenia	1	2012
16	China	1	2012
17	United States	1	2012
18	Italy	1	2012
19	France	1	2012

3.
3-1. I created the movies, ratings, and users tables and loaded their respective files into them.
Queries:

```
create table movies (movieID INT, title STRING, genres STRING)
row format delimited fields terminated by ',' stored as textfile;
```

```
create table ratings (userId INT, movieId INT, rating FLOAT, timestamp TIMESTAMP)
row format delimited fields terminated by ',' stored as textfile;
```

```
create table users (userId INT, gender STRING, occupation INT, zipCode INT)
row format delimited fields terminated by ',' stored as textfile;
```

```
hive> load data local inpath '/home/cloudera/Documents/movies.csv' into table mo
vies;
Loading data to table work.movies
Table work.movies stats: [numFiles=1, totalSize=494431]
OK
Time taken: 0.478 seconds
hive> load data local inpath '/home/cloudera/Documents/ratings.csv' into table r
atings;
Loading data to table work.ratings
Table work.ratings stats: [numFiles=1, totalSize=2483723]
OK
Time taken: 0.244 seconds
hive> load data local inpath '/home/cloudera/Documents/users.txt' into table use
rs;
Loading data to table work.users
Table work.users stats: [numFiles=1, totalSize=116282]
OK
Time taken: 0.238 seconds
```

3-2. I listed all movies with "Action" or "Drama" genres.
Query:

```
1 select title
2 from movies
3 where genres like '%Action%'
4 or genres like '%Drama%';
```

Results:

	title
1	Waiting to Exhale (1995)
2	Heat (1995)
3	Sudden Death (1995)
4	GoldenEye (1995)
5	Nixon (1995)
6	Cutthroat Island (1995)
7	Casino (1995)
8	Sense and Sensibility (1995)
9	Money Train (1995)
10	Copycat (1995)
11	Assassins (1995)
12	Powder (1995)
13	Leaving Las Vegas (1995)
14	Othello (1995)
15	Now and Then (1995)
16	Persuasion (1995)
17	Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)
18	Dangerous Minds (1995)
19	Babe (1995)

3-3. I listed all movie ids that had a 5 rating from the ratings table.
Query:

```
select movieid
from ratings
where rating = 5;
```

Results:

	movieid
1	47
2	50
3	101
4	151
5	157
6	163
7	216
8	231
9	260
10	333
11	362
12	457
13	527
14	553
15	596
16	608
17	661

3-4. I found the top 11 average rated action movies in descending order of rating.
Query:

```
select movies.title,
sum(ratings.rating) / count(ratings.movieid) as averageRating
from movies
join ratings on movies.movieid=ratings.movieid
where movies.genres like '%Action%'
group by movies.title
order by averageRating desc
limit 11;
```

Results:

	movies.title	averagerating
1	Knock Off (1998)	5
2	Justice League: Doom (2012)	5
3	Faster (2010)	5
4	Wonder Woman (2009)	5
5	Branded to Kill (Koroshi no rakuin) (1967)	5
6	Battle Royale 2: Requiem (Batoru rowaiaru II: Chinkonka) (2003)	5
7	Galaxy of Terror (Quest) (1981)	5
8	Crippled Avengers (Can que) (Return of the 5 Deadly Venoms) (1981)	5
9	Love Exposure (Ai No Mukidashi) (2008)	5
10	Tokyo Tribe (2014)	5
11	Alien Contamination (1980)	5