Practica con Pig

1) APARTADO A

Mediante un script de PIG, encontrar las cinco películas (código, título y número de votos) más votadas (recuento de votos, no media).

1) Describe informalmente los pasos que darás para llegar a la solución, por ejemplo:
    I. Extraemos todos los registros de u.data.
    II. Extraemos el identificador y título de u.item.
    III. Agrupamos los datos por id y contamos el número de entradas.
    IV. Los combinamos con los ítems.
    V. Lo organizamos y lo recortamos.
2) Implementa en PIG el script necesario para obtener la información deseada.

```
datos = LOAD '/user/maria_dev/u.data'
    USING PigStorage('\t')
    AS (
        user_id:int,
        item_id:int,
        rating:int,
        timestamp:datetime
    );

items = LOAD '/user/maria_dev/u.item'
    USING PigStorage('|')
    AS (
        movie_id:int,
        movie_title:chararray,
        release_date:long,
        video_release_date:long,
        IMDb_URL:chararray,

        unknown:boolean,
        action:boolean,
        adventure:boolean,
        animation:boolean,
        childrens:boolean,
        comedy:boolean,
        crime:boolean,
        documentary:boolean,
```

```
        drama:boolean,
        fantasy:boolean,
        film_noir:boolean,
        horror:boolean,
        musical:boolean,
        mystery:boolean,
        romance:boolean,
        sci_fi:boolean,
        thriller:boolean,
        war:boolean,
        western:boolean,
    );

grouped = GROUP datos BY item_id;
votes = FOREACH grouped GENERATE
    group AS movie_id,
    COUNT(data) AS num_votos;

joined = JOIN votes BY movie_id, items BY movie_id;

filtramos = FOREACH joined GENERATE
    votes::movie_id AS movie_id,
    items::movie_title AS title,
    votes::num_votes AS votes;

ordenado = ORDER filtramos BY votes DESC;
top5 = LIMIT ordenado 5;

STORE top5 INTO '/user/maria_dev/resultados/tarea2/parte1' USING
PigStorage('\t');
```

3) Muestra la salida del mismo por pantalla y almacena también su resultado en un archivo en HDFS

```
[maria_dev@sandbox-hdp tarea2]$ hdfs dfs -cat /user/maria_dev/resultados/tarea2/parte1/part-r-00000
50      Star Wars (1977)        583
258     Contact (1997)  509
100     Fargo (1996)    508
181     Return of the Jedi (1983)       507
294     Liar Liar (1997)        485
```

2) APARTADO B

Mediante un script de PIG, encontrar las diez películas mejor valoradas (código, título y media de puntuación) por los usuarios (ahora sí, media de todos los votos recibidos).

1) Describe informalmente los pasos que darás para llegar a la solución.

   I. Extraemos todos los registros de u.data.

   II. Extraemos el identificador y título de u.item.

III. Agrupamos los datos por id y calculamos la media de calificaciones.

IV. Los combinamos con los ítems.

V. Lo organizamos y lo recortamos.

2) Implementa en PIG el script necesario para obtener la información deseada.

```
datos = LOAD '/user/maria_dev/u.data'
    USING PigStorage('\t')
    AS (
        user_id:int,
        item_id:int,
        rating:int,
        timestamp:datetime
    );

items = LOAD '/user/maria_dev/u.item'
    USING PigStorage('|')
    AS (
        movie_id:int,
        movie_title:chararray,
        release_date:long,
        video_release_date:long,
        IMDb_URL:chararray,

        unknown:boolean,
        action:boolean,
        adventure:boolean,
        animation:boolean,
        childrens:boolean,
        comedy:boolean,
        crime:boolean,
        documentary:boolean,
        drama:boolean,
        fantasy:boolean,
        film_noir:boolean,
        horror:boolean,
        musical:boolean,
        mystery:boolean,
        romance:boolean,
        sci_fi:boolean,
        thriller:boolean,
        war:boolean,
        western:boolean
    );

grouped = GROUP datos BY item_id;
votos = FOREACH grouped GENERATE
```

```
    group AS movie_id,
    AVG(datos.rating) AS media_votos;

joined = JOIN votos BY movie_id, items BY movie_id;

filtramos = FOREACH joined GENERATE
    votos::movie_id AS movie_id,
    items::movie_title AS title,
    votos::media_votos AS votos;

ordenado = ORDER filtramos BY votos DESC;
top10 = LIMIT ordenado 10;

STORE top10 INTO '/user/maria_dev/resultados/tarea2/parte2' USING
PigStorage('\t');
```

3) Muestra la salida del mismo por pantalla y almacena también su resultado en un archivo en HDFS.

```
[maria_dev@sandbox-hdp tarea2]$ hdfs dfs -cat /user/maria_dev/resultado
1536    Aiqing wansui (1994)    5.0
1122    They Made Me a Criminal (1939)  5.0
1653    Entertaining Angels: The Dorothy Day Story (1996)       5.0
814     Great Day in Harlem, A (1994)   5.0
1189    Prefontaine (1997)      5.0
1293    Star Kid (1997) 5.0
1500    Santa with Muscles (1996)       5.0
1201    Marlene Dietrich: Shadow and Light (1996)       5.0
1599    Someone Else's America (1995)   5.0
1467    Saint of Fort Washington, The (1993)    5.0
```

3) APARTADO C

Mediante un script de PIG, encontrar las cinco películas más antiguas con una valoración media por encima de 4 puntos.

1) Implementa en PIG el script necesario para obtener la información deseada.

```
datos = LOAD '/user/maria_dev/u.data'
    USING PigStorage('\t')
    AS (
        user_id:int,
        item_id:int,
        rating:int,
        timestamp:datetime
    );

items = LOAD '/user/maria_dev/u.item'
    USING PigStorage('|')
    AS (
        movie_id:int,
```

```
        movie_title:chararray,
        release_date:long,
        video_release_date:long,
        IMDb_URL:chararray,

        unknown:boolean,
        action:boolean,
        adventure:boolean,
        animation:boolean,
        childrens:boolean,
        comedy:boolean,
        crime:boolean,
        documentary:boolean,
        drama:boolean,
        fantasy:boolean,
        film_noir:boolean,
        horror:boolean,
        musical:boolean,
        mystery:boolean,
        romance:boolean,
        sci_fi:boolean,
        thriller:boolean,
        war:boolean,
        western:boolean
    );

grouped = GROUP datos BY item_id;
votos = FOREACH grouped GENERATE
    group AS movie_id,
    AVG(datos.rating) AS media_votos;

joined = JOIN votos BY movie_id, items BY movie_id;

filtramos = FOREACH joined GENERATE
    votos::movie_id AS movie_id,
    items::movie_title AS title,
    items::release_date AS release_date,
    votos::media_votos AS votos;

refiltramos = FILTER filtramos BY votos > 4;

ordenado = ORDER refiltramos BY release_date ASC;
dataset_final = LIMIT ordenado 5;

STORE dataset_final INTO '/user/maria_dev/resultados/tarea2/parte3' USING
PigStorage('\t');
```

2) Muestra la salida del mismo por pantalla y almacena también su resultado en un archivo en HDFS

```
1594    Everest (1998)          4.5
1599    Someone Else's America (1995)           5.0
1639    Bitter Sugar (Azucar Amargo) (1996)             4.333333333333333
1642    Some Mother's Son (1996)                4.5
1653    Entertaining Angels: The Dorothy Day Story (1996)               5.0
```

4) APARTADO D

Mediante un script de PIG, encontrar la película mejor valorada por cada una de las ocupaciones (student, writer, doctor, etc.)

1) Implementa en PIG el script necesario para obtener la información deseada.

```
data = LOAD '/user/maria_dev/u.data'
    USING PigStorage('\t')
    AS (
        user_id:int,
        item_id:int,
        rating:int,
        timestamp:datetime
    );

items = LOAD '/user/maria_dev/u.item'
    USING PigStorage('|')
    AS (
        movie_id:int,
        movie_title:chararray,
        release_date:long,
        video_release_date:long,
        IMDb_URL:chararray,

        unknown:boolean,
        action:boolean,
        adventure:boolean,
        animation:boolean,
        childrens:boolean,
        comedy:boolean,
        crime:boolean,
        documentary:boolean,
        drama:boolean,
        fantasy:boolean,
        film_noir:boolean,
        horror:boolean,
        musical:boolean,
        mystery:boolean,
        romance:boolean,
        sci_fi:boolean,
        thriller:boolean,
        war:boolean,
        western:boolean
```

```
    );

users = LOAD '/user/maria_dev/ml-100k/u.user'
    USING PigStorage('|')
    AS (
        user_id:int,
        age:int,
        gender:chararray,
        occupation:chararray,
        zip:chararray
    );

data_users = JOIN data BY user_id, users BY user_id;
data_occupation = GROUP data_users BY (users::occupation, data::item_id);

data_occupation_avg = FOREACH data_occupation GENERATE
    group.occupation AS occupation,
    group.item_id AS movie_id,
    AVG(data_users.data::rating) AS media_votos;

grp = GROUP data_occupation_avg BY occupation;

max_por_ocupacion = FOREACH grp {
    ordenado = ORDER data_occupation_avg BY media_votos DESC;
    top1 = LIMIT ordenado 1;
    GENERATE FLATTEN(top1);
};

max_con_titulo = JOIN max_por_ocupacion BY top1::movie_id, items BY movie_id;

resultado_final = FOREACH max_con_titulo GENERATE
    max_por_ocupacion::top1::occupation AS occupation,
    max_por_ocupacion::top1::movie_id AS movie_id,
    items::movie_title AS movie_title,
    max_por_ocupacion::top1::media_votos AS media_votos;

STORE resultado_final INTO '/user/maria_dev/resultados/tarea2/parte4' USING
PigStorage('\t');
```

2) Muestra la salida del mismo por pantalla y almacena también su resultado en un archivo en HDFS

```
salesman        114     Wallace & Gromit: The Best of Aardman Animation (1996)  5.0
writer  130     Kansas City (1996)      5.0
doctor  132     Wizard of Oz, The (1939)        5.0
healthcare      148     Ghost and the Darkness, The (1996)      5.0
librarian       149     Jude (1996)     5.0
entertainment   150     Swingers (1996) 5.0
executive       155     Dirty Dancing (1987)    5.0
scientist       156     Reservoir Dogs (1992)   5.0
retired 158     Weekend at Bernie's (1989)      5.0
technician      166     Manon of the Spring (Manon des sources) (1986)  5.0
artist  169     Wrong Trousers, The (1993)      5.0
marketing       169     Wrong Trousers, The (1993)      5.0
lawyer  169     Wrong Trousers, The (1993)      5.0
homemaker       222     Star Trek: First Contact (1996) 5.0
administrator   224     Ridicule (1996) 5.0
other   247     Turbo: A Power Rangers Movie (1997)      5.0
educator        263     Steel (1997)    5.0
student 279     Once Upon a Time... When We Were Colored (1995) 5.0
programmer      320     Paradise Lost: The Child Murders at Robin Hood Hills (1996)     5.0
engineer        611     Laura (1944)    5.0
none    874     Career Girls (1997)     5.0
```

5) APARTADO E

Mediante un script de PIG, encontrar el promedio de valoraciones por décadas, guardarlo en HDFS como un archivo csv. Posteriormente lo descargaremos a nuestro ordenador y con EXCEL hacer un gráfico de barras con los datos del fichero

1) Implementa en PIG el script necesario para obtener la información deseada.
    i. Analizar el archivo u.item (información de películas) y extraer la fecha de estreno utilizando el operador SUBSTRING.
    ii. Agrupar las películas por década (por ejemplo: 1970, 1980, 1990, etc.).
    iii. Calcular el promedio de rating por década.
    iv. Guardar los resultados en CSV para graficar posteriormente.

```
items = LOAD '/user/maria_dev/u.item'
    USING PigStorage('|')
    AS (
        movie_id:int,
        movie_title:chararray,
        release_date:chararray,
        video_release_date:chararray,
        IMDb_URL:chararray,

        unknown:boolean,
        action:boolean,
        adventure:boolean,
        animation:boolean,
        childrens:boolean,
        comedy:boolean,
        crime:boolean,
```

```
        documentary:boolean,
        drama:boolean,
        fantasy:boolean,
        film_noir:boolean,
        horror:boolean,
        musical:boolean,
        mystery:boolean,
        romance:boolean,
        sci_fi:boolean,
        thriller:boolean,
        war:boolean,
        western:boolean
    );

ratings = LOAD '/user/maria_dev/u.data'
    USING PigStorage('\t')
    AS (
        user_id:int,
        movie_id:int,
        rating:int,
        timestamp:chararray
    );

items_year = FOREACH items GENERATE
    movie_id,
    (int)SUBSTRING(release_date, 7, 11) AS year;

items_year_valid = FILTER items_year BY year IS NOT NULL;

items_decade = FOREACH items_year_valid GENERATE
    movie_id,
    (year / 10 * 10) AS decade;

ratings_with_decade = JOIN ratings BY movie_id, items_decade BY movie_id;

ratings_clean = FOREACH ratings_with_decade GENERATE
    ratings::rating AS rating,
    items_decade::decade AS decade;

group_by_decade = GROUP ratings_clean BY decade;

avg_rating_by_decade = FOREACH group_by_decade GENERATE
    group AS decade,
    AVG(ratings_clean.rating) AS avg_rating;

STORE avg_rating_by_decade INTO '/user/maria_dev/resultados/tarea2/parte5'
USING PigStorage(',');
```

2) Abrir el archive en Excel y generar un gráfico de barras con los datos

1920	3,53571428571428
1930	3,92513368983957
1940	4,01067140951534
1950	3,93725042783799
1960	3,88154838709677
1970	3,86652780003204
1980	3,74979376340537
1990	3,39886436226843