Máster en Ingeniería del Software: Cloud, Datos y Gestión TI

Big Data 2023

Entregable 2 - Pig

Realizado por José Antonio Zamudio Amaya

UNIVERSIDAD Ð SEVILLA

Índice de contenidos

1. Propuesta

La saga de películas de Star Wars es una de las franquicias más icónicas de la cultura pop, y cuenta con una gran cantidad de personajes y especies fascinantes. En este proyecto, se utilizará la herramienta Pig para realizar un análisis de datos de personajes y especies de Star Wars. El objetivo de esta práctica es analizar los datos de personajes y especies de Star Wars para obtener información relevante y significativa. Algunos de los objetivos específicos son:

1. Unir los datos de personajes, especies y planetas de Star Wars utilizando Pig.
2. Realizar agregaciones de datos para obtener información estadística, como la altura y el peso promedio de los personajes por especie.
3. Crear filtros complejos para obtener información específica, como los personajes de una especie en particular con una altura superior a cierto valor.

2. Dataset

Los datasets utilizados en este proyecto son dos archivos CSV: "characters.csv" y "species.csv". Ambos archivos contienen información sobre personajes y especies del universo de Star Wars. El archivo "characters.csv" contiene los siguientes campos:

1. "name": el nombre del personaje.
2. "height": la altura del personaje en centímetros.
3. "mass": la masa del personaje en kilogramos.
4. "hair_color": el color del cabello del personaje.
5. "skin_color": el color de piel del personaje.
6. "eye_color": el color de ojos del personaje.
7. "birth_year": el año de nacimiento del personaje en formato "ABY" (After Battle of Yavin) o "BBY" (Before Battle of Yavin).
8. "gender": el género del personaje.
9. "homeworld": el planeta de origen del personaje.
10. "species": la especie del personaje.

El archivo "species.csv" contiene los siguientes campos:

1. "name": el nombre de la especie.
2. "classification": la clasificación de la especie.
3. "designation": la designación de la especie.
4. "average_height": la altura promedio de la especie en centímetros.
5. "skin_colors": los colores de piel de la especie.
6. "hair_colors": los colores de cabello de la especie.
7. "eye_colors": los colores de ojos de la especie.
8. "average_lifespan": la esperanza de vida promedio de la especie en años.
9. "language": el idioma de la especie.
10. "homeworld": el planeta de origen de la especie.

Ambos archivos se relacionan mediante el campo "species" del archivo "characters.csv" y el campo "name" del archivo "species.csv".

3. Código

En esta sección se muestran los fragmentos de código utilizados para las funciones junto a explicaciones detalladas de su funcionamiento.

3.1. Unión - Conjuntos de datos de Star Wars

```
-- Load the character and species datasets
characters = LOAD '/home/maria_dev/pig/characters.csv' USING
PigStorage(',') AS (name:chararray, height:int, mass:int,
hair_color:chararray, skin_color:chararray, eye_color:chararray,
birth_year:chararray, gender:chararray, homeworld:chararray,
species:chararray);
species = LOAD '/home/maria_dev/pig/species.csv' USING PigStorage(',')
AS (name:chararray, classification:chararray, designation:chararray,
average_height:int, skin_colors:chararray, hair_colors:chararray,
eye_colors:chararray, average_lifespan:int, language:chararray,
homeworld:chararray);

-- Join the datasets on the species name column
character_species = JOIN characters BY species, species BY name;

-- Filter out unnecessary columns
character_species_filtered = FOREACH character_species GENERATE
characters::name AS name, height, mass, hair_color, skin_color,
eye_color, birth_year, gender, homeworld, species, classification,
designation, average_height, skin_colors, hair_colors, eye_colors,
average_lifespan, language;

-- Display the joined dataset
DUMP character_species_filtered;
```

Este código carga dos conjuntos de datos, characters y species, desde dos archivos CSV diferentes. La sintaxis USING PigStorage(',') especifica que el delimitador de campos en los archivos es una coma. A continuación, los campos de cada registro se asignan a variables con nombres significativos usando la cláusula AS.

Luego, los dos conjuntos de datos se unen usando la cláusula JOIN que combina registros con la misma especie. La cláusula BY especifica que la columna para unir es el nombre de la especie. El resultado de la unión se almacena en character_species.
Después, se eliminan las columnas innecesarias del conjunto de datos resultante usando la función FOREACH y la cláusula GENERATE. Las columnas se seleccionan y se asignan a nuevas variables con nombres específicos. El resultado se almacena en

character_species_filtered. Finalmente, la función DUMP se utiliza para mostrar el resultado de la operación en la consola.

```
(Arvel Crynyd,,,brown,fair,brown,NA,male,Human,mammal,sentient,180,"caucasian, black, asian,,"blonde)
(Ki-Adi-Mundi,198,82,white,pale,yellow,92BBY,male,Cerean,mammal,sentient,200,pale pink,"red, blond,, white")
(Rugor Nass,206,,none,green,orange,NA,male,Gungan,amphibian,sentient,190,"brown, green",none,,NA)
(Roos Tarpals,224,82,none,grey,orange,NA,male,Gungan,amphibian,sentient,190,"brown, green",none,,NA)
(Jar Jar Binks,196,66,none,orange,orange,52BBY,male,Gungan,amphibian,sentient,190,"brown, green",none,,NA)
(Tion Medon,206,80,none,grey,black,NA,male,Pau'an,mammal,sentient,190,grey,none,black,700,Utapese)
(Greedo,173,74,NA,green,black,44BBY,male,Rodian,sentient,reptilian,170,"green, blue",NA,,NA)
(Eeth Koth,171,,black,brown,brown,NA,male,Zabrak,mammal,sentient,180,"pale, brown, red,, yellow")
(Darth Maul,175,80,none,red,yellow,54BBY,male,Zabrak,mammal,sentient,180,"pale, brown, red,, yellow")
(Plo Koon,188,80,none,orange,black,22BBY,male,Kel Dor,NA,sentient,180,"peach, orange, red",,"black)
(Bib Fortuna,180,,none,pale,pink,NA,male,Twi'lek,mammals,sentient,200,"orange, yellow, blue,, pink)
(Ayla Secura,178,55,none,blue,hazel,48BBY,female,Twi'lek,mammals,sentient,200,"orange, yellow, blue,, pink)
(Chewbacca,228,112,brown,NA,blue,200BBY,male,Wookiee,mammal,sentient,210,gray,"black, brown",, green)
(Tarfful,234,136,brown,brown,blue,NA,male,Wookiee,mammal,sentient,210,gray,"black, brown",, green)
(Dexter Jettster,198,102,none,brown,yellow,NA,male,Besalisk,amphibian,sentient,178,brown,none,yellow,75,besalisk)
(Mas Amedda,196,,none,blue,blue,NA,male,Chagrian,amphibian,sentient,190,blue,none,blue,,Chagria)
(Saesee Tiin,188,,none,pale,orange,NA,male,Iktotchi,NA,sentient,180,pink,none,orange,,Iktotchese)
(Taun We,213,,none,grey,black,NA,female,Kaminoan,amphibian,sentient,220,"grey, blue",none,,80)
(Lama Su,229,88,none,grey,black,NA,male,Kaminoan,amphibian,sentient,220,"grey, blue",none,,80)
(Barriss Offee,166,50,black,yellow,blue,40BBY,female,Mirialan,mammal,sentient,180,"yellow, green","black,,"blue)
(Luminara Unduli,170,56,black,yellow,blue,58BBY,female,Mirialan,mammal,sentient,180,"yellow, green","black,,"blue)
(Kit Fisto,196,87,none,green,black,NA,male,Nautolan,amphibian,sentient,180,"green, blue, brown,,none)
(Yarael Poof,264,,none,white,yellow,NA,male,Quermian,mammal,sentient,240,white,none,yellow,86,Quermian)
```

3.2. Agregación - Agrupación por especies y recuento del número de caracteres

```
-- Load the character and species datasets
characters = LOAD '/home/maria_dev/pig/characters.csv' USING
PigStorage(',') AS (name:chararray, height:int, mass:int,
hair_color:chararray, skin_color:chararray, eye_color:chararray,
birth_year:chararray, gender:chararray, homeworld:chararray,
species:chararray);
species = LOAD '/home/maria_dev/pig/species.csv' USING PigStorage(',')
AS (name:chararray, classification:chararray, designation:chararray,
average_height:int, skin_colors:chararray, hair_colors:chararray,
eye_colors:chararray, average_lifespan:int, language:chararray,
homeworld:chararray);

-- Join the character and species datasets
character_species = JOIN characters BY species, species BY name;

-- Group by species and count the number of characters
character_count = GROUP character_species BY species;
character_count = FOREACH character_count GENERATE group AS species,
COUNT(character_species) AS character_count;

-- Display the results
DUMP character_count;
```

Este código carga dos conjuntos de datos, characters y species, que están almacenados en archivos CSV separados en la ruta /home/maria_dev/pig/. Ambos conjuntos de datos se

cargan utilizando PigStorage(',') para indicar que los campos están separados por comas y se especifica la estructura de los datos utilizando AS. Luego, los dos conjuntos de datos se unen por la columna de nombre de especie. A continuación, se agrupa el conjunto de datos combinado character_species por la columna de especie y se cuenta el número de personajes que pertenecen a cada especie con la función COUNT(). Finalmente, se muestra el resultado utilizando la función DUMP().

```
(Ewok,1)
(Muun,1)
(Droid,3)
(Human,32)
(Cerean,1)
(Gungan,3)
(Pau'an,1)
(Rodian,1)
(Zabrak,2)
(Kel Dor,1)
(Twi'lek,2)
(Wookiee,2)
(Besalisk,1)
(Chagrian,1)
(Iktotchi,1)
(Kaminoan,2)
(Mirialan,2)
(Nautolan,1)
(Quermian,1)
(Geonosian,1)
(Neimodian,1)
(Sullustan,1)
(Tholothian,1)
(Trandoshan,1)
(Mon Calamari,1)
(Yoda's species,1)
```

3.3. Agregación - Calcular la altura y la masa medias por especie

```
-- Load character and species data
characters = LOAD '/home/maria_dev/pig/characters.csv' USING
PigStorage(',')
            AS (name:chararray, height:int, mass:int,
hair_color:chararray, skin_color:chararray, eye_color:chararray,
birth_year:chararray, gender:chararray, homeworld:chararray,
species:chararray);

species = LOAD '/home/maria_dev/pig/species.csv' USING PigStorage(',')
            AS (name:chararray, classification:chararray,
```

```
designation:chararray, average_height:int, skin_colors:chararray,
hair_colors:chararray, eye_colors:chararray, average_lifespan:int,
language:chararray, homeworld:chararray);

-- Join character and species data on species name
character_species = JOIN characters BY species, species BY name;

-- Compute average height and mass per species
species_stats = FOREACH (GROUP character_species BY species) GENERATE
group AS species,  AVG(character_species.height) AS avg_height,
AVG(character_species.mass) AS avg_mass;

-- Output the results
DUMP species_stats;
```

Este código carga dos conjuntos de datos, 'characters.csv' y 'species.csv', y los almacena en las relaciones 'characters' y 'species', respectivamente. Luego une las dos relaciones por la columna 'species' del conjunto de datos 'characters'.

A continuación, se calcula la altura y el peso promedio de cada especie. Esto se hace mediante la agrupación de la relación resultante por la columna 'species' y el cálculo de la media de la altura y el peso de cada grupo. Los resultados se almacenan en la relación 'species_stats'. Finalmente, los resultados se muestran con el comando 'DUMP'.

```
(Ewok,88.0,20.0)
(Muun,191.0,)
(Droid,183.5,107.5)
(Human,176.28571428571428,81.2)
(Cerean,198.0,82.0)
(Gungan,208.66666666666666,74.0)
(Pau'an,206.0,80.0)
(Rodian,173.0,74.0)
(Zabrak,173.0,80.0)
(Kel Dor,188.0,80.0)
(Twi'lek,179.0,55.0)
(Wookiee,231.0,124.0)
(Besalisk,198.0,102.0)
(Chagrian,196.0,)
(Iktotchi,188.0,)
(Kaminoan,221.0,88.0)
(Mirialan,168.0,53.0)
(Nautolan,196.0,87.0)
(Quermian,264.0,)
(Geonosian,183.0,80.0)
(Neimodian,191.0,90.0)
(Sullustan,160.0,68.0)
(Tholothian,184.0,50.0)
(Trandoshan,190.0,113.0)
(Mon Calamari,180.0,83.0)
(Yoda's species,66.0,17.0)
```

3.4. Filtrado - Caracteres por especie y color de ojos

```
-- Load the CSV files
characters = LOAD '/home/maria_dev/pig/characters.csv' USING
PigStorage(',')
AS (name:chararray, height:int, mass:int, hair_color:chararray,
skin_color:chararray, eye_color:chararray, birth_year:chararray,
gender:chararray, homeworld:chararray, species:chararray);

species = LOAD '/home/maria_dev/pig/species.csv' USING PigStorage(',')
AS (name:chararray, classification:chararray, designation:chararray,
average_height:int, skin_colors:chararray, hair_colors:chararray,
eye_colors:chararray, average_lifespan:int, language:chararray,
homeworld:chararray);

-- Filter the characters by species and eye color
human_characters = FILTER characters BY species == 'Human' AND eye_color
== 'blue';
```

```
-- Display the filtered results
DUMP human_characters;
```

Este código carga 'characters.csv' y 'species.csv' en las relaciones 'characters' y 'species' respectivamente. Luego, filtra los personajes por especie y color de ojos, restringiendo la selección a personajes humanos con ojos azules. Finalmente, imprime en la pantalla los personajes filtrados.

El operador FILTER es usado en este código para aplicar dos condiciones al conjunto de datos. La primera condición es que la especie sea 'Human' y la segunda es que el color de ojos sea 'blue'. El operador AND se utiliza para combinar estas dos condiciones. Los resultados se almacenan en la relación 'human_characters' que luego se muestra con el operador DUMP.

```
(Luke Skywalker,172,77,blond,fair,blue,19BBY,male,Tatooine,Human)
(Beru Whitesun lars,165,75,brown,light,blue,47BBY,female,Tatooine,Human)
(Anakin Skywalker,188,84,blond,fair,blue,41.9BBY,male,Tatooine,Human)
(Jek Tono Porkins,180,110,brown,fair,blue,NA,male,Bestine IV,Human)
(Lobot,175,79,none,light,blue,37BBY,male,Bespin,Human)
(Mon Mothma,150,,auburn,fair,blue,48BBY,female,Chandrila,Human)
(Qui-Gon Jinn,193,89,brown,fair,blue,92BBY,male,NA,Human)
(Finis Valorum,170,,blond,fair,blue,91BBY,male,Coruscant,Human)
(Cliegg Lars,183,,brown,fair,blue,82BBY,male,Tatooine,Human)
(Jocasta Nu,167,,white,fair,blue,NA,female,Coruscant,Human)
```

3.5. Filtrado - Personajes nacidos antes del 20BBY, de pelo castaño o negro y de estatura superior a 180 cm

```
-- Load the data
characters = LOAD '/path/to/characters.csv' USING PigStorage(',')
AS (name:chararray, height:int, mass:int, hair_color:chararray,
skin_color:chararray,
      eye_color:chararray, birth_year:chararray, gender:chararray,
homeworld:chararray, species:chararray);

-- Filter out characters who have brown or black hair and are taller
than 180cm
filtered_characters = FILTER characters BY (hair_color == 'brown' OR
hair_color == 'black') AND height > 180;

-- Filter out characters who were born before 20BBY
final_characters = FILTER filtered_characters BY birth_year >= '20BBY';
```

```
-- Output the final result
DUMP final_characters;
```

Este código carga "characters.csv" y define el esquema de la tabla utilizando la función AS. El archivo contiene información sobre personajes ficticios y sus atributos como el nombre, altura, masa, color del cabello, color de piel, color de ojos, año de nacimiento, género, hogar y especie.

A continuación, el código filtra los personajes que cumplen con ciertas condiciones. Primero, se filtran los personajes que pertenecen a la especie humana y tienen ojos azules. Luego, se filtran los personajes que tienen cabello marrón o negro y miden más de 180cm. Finalmente, se filtran los personajes que nacieron después del año 20BBY (Before the Battle of Yavin). Los resultados filtrados se almacenan en la variable final_characters y se muestran en la pantalla utilizando la función DUMP.

```
(Biggs Darklighter,183,84,black,light,brown,24BBY,male,Tatooine,Human)
(Boba Fett,183,78,black,fair,brown,31.5BBY,male,Kamino,Human)
(Qui-Gon Jinn,193,89,brown,fair,blue,92BBY,male,NA,Human)
(Ric Olié,183,,brown,fair,blue,NA,male,Naboo,NA)
(Quarsh Panaka,183,,black,dark,brown,62BBY,male,Naboo,NA)
(Gregar Typho,185,85,black,dark,brown,NA,male,Naboo,Human)
(Cliegg Lars,183,,brown,fair,blue,82BBY,male,Tatooine,Human)
(Bail Prestor Organa,191,,black,tan,brown,67BBY,male,Alderaan,Human)
(Jango Fett,183,79,black,tan,brown,66BBY,male,Concord Dawn,Human)
(Tarfful,234,136,brown,brown,blue,NA,male,Kashyyyk,Wookiee)
(Raymus_Antilles,188,79,brown,light,brown,NA,male,Alderaan,Human)
```

4. Puntos de mejora y trabajo futuro

Durante esta práctica, hemos trabajado con una serie de ejemplos de código en el lenguaje Pig, que es utilizado para procesar grandes conjuntos de datos en Apache Hadoop.

En primer lugar, cargamos los conjuntos de datos de personajes y especies de Star Wars desde archivos CSV, los unimos en una tabla única y filtramos los datos para obtener solo las columnas relevantes. También utilizamos la cláusula GROUP BY para contar el número de personajes por especie y para calcular estadísticas como la altura y la masa promedio por especie Luego, implementamos filtros adicionales para encontrar solo los personajes humanos con ojos azules, así como aquellos con cabello marrón o negro y una altura mayor a 180 cm, nacidos después del año 20BBY.

Para trabajo futuro, podríamos explorar otros aspectos de Pig, como el uso de funciones de usuario y la integración con otros lenguajes de programación, como Python y Java. También podríamos explorar más casos de uso y ejemplos de cómo Pig se puede utilizar en conjunción con otras herramientas de big data, como Apache Spark y Apache Hive.