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

Big Data 2023

Entregable 3 - Hive

Realizado por José Antonio Zamudio Amaya

UNIVERSIDAD D SEVILLA

Índice de contenidos

1. Propuesta

La franquicia de Star Wars ha sido un fenómeno de la cultura popular durante décadas, cautivando a audiencias con su rico universo y personajes icónicos. En este proyecto, exploramos los conjuntos de datos de Star Wars en Apache Hive, un sistema de almacén de datos distribuido que facilita la consulta y el análisis de grandes conjuntos de datos. Trabajaremos con tres conjuntos de datos: personajes, planetas y especies, todos ellos contienen información sobre el universo de Star Wars.

El objetivo para este proyecto es realizar consultas y análisis complejos en los conjuntos de datos de Star Wars utilizando Apache Hive. Aprovecharemos el poder de Hive para unir múltiples tablas, filtrar y transformar datos, y obtener información sobre el universo de Star Wars. Específicamente, trataremos los siguientes temas:

1. Análisis de personajes: Analizaremos las características de los personajes de Star Wars, como su altura, peso, género y especie. Usaremos Hive para filtrar y agrupar personajes en función de estas características, y visualizamos los resultados utilizando herramientas como Excel o Tableau.
2. Análisis de planetas: Explicaremos los diversos planetas del universo de Star Wars y analizaremos sus propiedades, como su clima, diámetro y población. Usaremos Hive para unir la tabla de planetas con otras tablas para obtener información sobre las relaciones entre personajes, especies y planetas.
3. Análisis de especies: Analizaremos las diversas especies en el universo de Star Wars y exploramos sus propiedades, como su altura promedio, esperanza de vida promedio y clasificación. Usaremos Hive para unir la tabla de especies con otras tablas para obtener información sobre las relaciones entre personajes, especies y planetas.

2. Dataset

En este proyecto usaremos un dataset adicional a los vistos en la práctica 2. Los datasets utilizados en este proyecto son tres archivos CSV: "characters.csv", "species.csv" y “planets.csv”. Los tres archivos contienen información sobre personajes, especies y planetas 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".

El dataset planets.csv contiene información sobre los planetas del universo de Star Wars. Tiene las siguientes columnas:

1. name: el nombre del planeta.
2. rotation_period: la cantidad de horas que tarda el planeta en rotar sobre su eje.
3. orbital_period: la cantidad de días que tarda el planeta en orbitar alrededor de su estrella.
4. diameter: el diámetro del planeta en kilómetros.
5. climate: el clima predominante del planeta.
6. gravity: la fuerza gravitatoria del planeta.
7. terrain: el tipo de terreno predominante en el planeta.
8. surface_water: el porcentaje de la superficie del planeta cubierta por agua.
9. population: la cantidad de habitantes del planeta (en caso de haberlos).

3. Carga de datos

Para cargar los datos, se han almacenado los ficheros csvs en la carpeta /tmp/hivetest de nuestro almacenamiento local de hadoop. Con esto, podemos crear una nueva base de datos desde Hive View en Ambari. Creamos las tablas en nuestra base de datos con:

```
CREATE DATABASE IF NOT EXISTS starwars;
USE starwars;
```

Ahora cargaremos los datasets con las siguientes queries:

```
CREATE TABLE characters (
  name STRING,
```

```
  height INT,
  mass INT,
  hair_color STRING,
  skin_color STRING,
  eye_color STRING,
  birth_year STRING,
  gender STRING,
  homeworld STRING,
  species STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';

LOAD DATA LOCAL INPATH '/tmp/hivetest/characters.csv' INTO TABLE
characters;
```

```
CREATE TABLE species (
      name STRING,
      classification STRING,
      designation STRING,
      average_height INT,
      skin_colors STRING,
      hair_colors STRING,
      eye_colors STRING,
      average_lifespan INT,
      language STRING,
      homeworld STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
LOAD DATA LOCAL INPATH '/tmp/hivetest.csv' INTO TABLE species;
```

```
CREATE TABLE planets (
  name STRING,
  rotation_period INT,
  orbital_period INT,
  diameter INT,
  climate STRING,
  gravity STRING,
  terrain STRING,
  surface_water INT,
  population BIGINT
)
ROW FORMAT DELIMITED
```

```
FIELDS TERMINATED BY ',';
LOAD DATA LOCAL INPATH '/tmp/hivetest/planets.csv' INTO TABLE planets;
```

4. Consultas

En esta sección se muestran las consultas realizadas junto a explicaciones detalladas de su funcionamiento.

4.1. Nivel básico

```
SELECT name, height FROM characters ORDER BY height DESC;
```

Muestra el nombre y altura de todos los personajes de Star Wars, ordenados de mayor a menor altura.

```
SELECT species, COUNT(*) as num_characters FROM characters GROUP BY
species;
```

Cuenta el número de personajes de Star Wars de cada especie y muestra la cantidad junto al nombre de la especie.

```
SELECT name FROM planets WHERE population > 1000000000;
```

Selecciona los nombres de los planetas de Star Wars que tienen una población mayor a mil millones de habitantes.

name
Eriadu

```
SELECT name FROM characters WHERE hair_color = 'blond' AND eye_color =
'blue';
```

Selecciona los nombres de los personajes de Star Wars que tienen cabello rubio y ojos azules.

name
Luke Skywalker
Anakin Skywalker
Finis Valorum

4.2. Nivel intermedio

```
SELECT c.name, c.height, s.classification
FROM characters c
JOIN species s ON c.species = s.name
WHERE c.gender = 'female' AND c.species = 'Human';
```

Esta consulta selecciona el nombre, la altura y la clasificación de especies de todos los personajes femeninos humanos en la tabla "characters" y su correspondiente información en la tabla "species". Utiliza la cláusula JOIN para unir ambas tablas por el campo "species". La condición WHERE filtra los resultados para que solo incluya personajes femeninos y humanos.

c.name	c.height	s.classification
Leia Organa	150	mammal
Beru Whitesun lars	165	mammal

```
SELECT p.name, COUNT(c.name) as num_characters
FROM planets p
LEFT JOIN characters c ON p.name = c.homeworld
GROUP BY p.name
HAVING COUNT(c.name) > 5;
```

Esta consulta utiliza una operación JOIN para combinar información de las tablas "planets" y "characters". En particular, se utiliza un LEFT JOIN para incluir todas las filas de la tabla "planets" incluso si no tienen una coincidencia en la tabla "characters". Luego, se cuenta el número de personajes asociados con cada planeta utilizando la función COUNT y se agrupa por el nombre del planeta. Finalmente, se utiliza la cláusula HAVING para filtrar solo los planetas con más de 5 personajes asociados.

p.name	num_characters
NA	16
Naboo	10
Tatooine	8

```
SELECT s.name as species_name, p.name as planet_name, p.climate
FROM species s
JOIN planets p ON s.homeworld = p.name
WHERE s.classification = 'mammal' AND p.climate LIKE '%temperate%';
```

Esta consulta utiliza una operación JOIN para combinar información de las tablas "species" y "planets". En particular, se busca una coincidencia en el nombre del planeta en la tabla "planets" para cada especie en la tabla "species". Luego, se utiliza la cláusula WHERE para filtrar solo las especies con una clasificación de "mammal" y los planetas con un clima que contenga la palabra "temperate". Por lo tanto, la consulta devuelve una lista de especies con el nombre del planeta donde se encuentran y el clima de ese planeta.

species_name	planet_name	p.climate
Pau'an	Utapau	"temperate

4.3. Nivel avanzado

```
SELECT c.name as character_name, p.name as planet_name, s.classification
FROM characters c
JOIN planets p ON c.homeworld = p.name
JOIN species s ON c.species = s.name
WHERE p.climate LIKE '%desert%' AND s.classification = 'mammal';
```

Esta consulta selecciona el nombre del personaje, el nombre del planeta y la clasificación de especies de todos los personajes en la tabla "characters" que pertenecen a una especie de mamíferos y tienen su hogar en un planeta con un clima de desierto.

Utiliza la cláusula JOIN para unir las tres tablas: "characters", "planets" y "species", utilizando las claves "homeworld" y "name". La condición WHERE filtra los resultados para que solo incluya personajes que pertenezcan a una especie de mamíferos y cuyo planeta de origen tenga un clima de desierto.

character_name	planet_name	s.classification

```
SELECT c.name as character_name, s.name as species_name, p.name as
planet_name
FROM characters c
JOIN species s ON c.species = s.name
LEFT JOIN planets p ON c.homeworld = p.name
WHERE s.classification = 'droid' OR (s.classification = 'mammal' AND
p.surface_water > 60);
```

Esta consulta une las tres tablas characters, species y planets. La cláusula JOIN une characters con species utilizando el campo species de la tabla characters y el campo name de la tabla species. Luego, se utiliza un LEFT JOIN para unir la tabla planets con characters utilizando el campo homeworld de la tabla characters y el campo name de la tabla planets.

La consulta selecciona el nombre del personaje (c.name), el nombre de la especie (s.name) y el nombre del planeta (p.name). El WHERE se compone de dos condiciones unidas por un operador lógico OR. La primera condición busca todos los personajes cuya especie sea "droid". La segunda condición busca todos los personajes cuyo planeta tenga más de 60% de agua en su superficie y cuya especie sea "mammal".

character_name	species_name	planet_name
Boba Fett	Human	Kamino

5. Puntos de mejora y trabajo futuro

En este proyecto se ha realizado un análisis de datos utilizando Hive, una herramienta de procesamiento de big data muy utilizada en el ecosistema de Hadoop. Se trabajó con tres datasets diferentes: characters.csv, planets.csv y species.csv. Cada uno de ellos fue cargado en una tabla de Hive y se realizaron consultas complejas para extraer información interesante. A continuación, se presentan algunas ideas para continuar trabajando en este proyecto:

1. Visualización de datos: Una forma de presentar de manera más efectiva los resultados del análisis es a través de visualizaciones de datos. En este sentido, se podría utilizar herramientas como Tableau o Power BI para crear gráficos y dashboards que muestren la información extraída de las consultas en una forma más accesible.

2. Machine learning: Se podría entrenar un modelo de machine learning para predecir, por ejemplo, la especie de un personaje a partir de sus características físicas. Para ello, se podría utilizar el dataset characters.csv y técnicas de machine learning como árboles de decisión o redes neuronales.
3. Integración con otras fuentes de datos: Se podría integrar los datos de este proyecto con otras fuentes de datos para enriquecer aún más el análisis. Por ejemplo, se podría utilizar datos de redes sociales para analizar el impacto de los personajes de Star Wars en la cultura popular.