La finalidad de esta práctica es familiarizarse con el lenguaje PIG.

Vete haciendo capturas de pantalla de todos los pasos que vayas dando, acompañándolas de comentarios descriptivos de los mismos.

INTRODUCCIÓN

A.- Trabajaremos sobre el conjunto de datos **Movielens** de **Kaggle:**

https://www.kaggle.com/datasets/prajitdatta/movielens-100k-dataset

Descarga ese *dataset* y copia el archivo ***u.user*** en el HDFS de la máquina con Hortonworks.

Si no tienes cuenta en Kaggle, data de alta en el servicio.

CONTENIDO

APARTADO A

Práctica con PIG

Trabajaremos sobre el fichero u.user. Las consultas han de hacerse con *mapreduce,* es decir, con los ficheros en HDFS, no en local.

u.user -- Demographic information about the users; this is a tab separated list of
user id | age | gender | occupation | zip code
The user ids are the ones used in the u.data data set.

1.- Muestra el total de hombres y mujeres que hay en el archivo *u.user*.

2.- Mediante instrucciones de PIG encontrar las 10 ocupaciones más frecuentes entre los usuarios.

3.- Muestra la edad media por géneros.

4.- Muestra la edad media por ocupaciones.

5.- Guarda el resultado de las cuatro consultas anteriores en un script de extensión "*.pig*". Ejecútalo. (recuerda, siempre en la carpeta `/user/maría_dev`)

6.- Almacena la salida de las cuatro consultas anteriores en una carpeta de HDFS llamada *`pig_usuarios`*.

INTRODUCCIÓN

A.- Ahora trabajaremos sobre el conjunto de datos ***Retail Sales Dataset*** de **Kaggle:**

https://www.kaggle.com/datasets/mohammadtalib786/retail-sales-dataset

Investiga su formato.

CONTENIDO

APARTADO B

1. **Carga y descripción del *dataset***
   - Carga el archivo (*retail_sales_dataset.csv*) usando PigStorage(',') y define un esquema correctamente para cada tipo de campo.
   - Usa DESCRIBE para ver el esquema y DUMP para ver las primeras tuplas.
   - Calcula cuántas transacciones totales tiene el *dataset* (COUNT).
2. **Filtrado por rango de edad**
   - Filtra los clientes con edad mayor de 30 años y guarda en alias `clientes_mayores30`.
   - Utiliza `LIMIT` para ver los primeros 10 resultados.
   - ¿Qué porcentaje del total de transacciones corresponde a clientes mayores de 30?
3. **Transformación de campos**
   - A partir del conjunto original, crea un alias donde generes:
     1. el género en mayúsculas (`UPPER(gender)`),
     2. una nueva columna `importe_descuento` que calcule, por ejemplo, `price_per_unit * quantity * 0.90` (aplicando un 10% de "descuento ficticio").
   - Muestra los primeros 20 registros resultantes.
4. **Agrupación y agregación por categoría de producto**
   - Agrupa por `product_category`.
   - Para cada categoría calcula: número de transacciones (`COUNT`), suma de `total_amount` (`SUM`), edad promedio de cliente (`AVG(age)`).
   - Ordena el resultado por la suma de `total_amount` descendente.
5. **Extracción de categorías distintas**
   - En este *dataset* extrae las categorías de producto distintas (`DISTINCT product_category`).
   - Pregunta: ¿Cuántas categorías diferentes hay?
6. **Ordenación y obtención de *top*-transacciones**
   - Ordena todas las transacciones por `total_amount` descendente.
   - Usa `LIMIT` para extraer, por ejemplo, las 5 transacciones con mayor `total_amount`.
   - Muestra: `transaction_id`, `customer_id`, `product_category`, `total_amount`.
7. **Uso de funciones de cadena**

- Añade una nueva columna al alias original donde el `product_category` se recorte a los primeros 3 caracteres (`SUBSTRING(product_category,0,3)`) y otra que sea la longitud del `product_category` (`SIZE(product_category)`).
- Muestra los primeros 15 registros resultantes.

8. **Filtrado por fecha y condiciones combinadas**
   - Filtra primero las transacciones que se han hecho **antes de** una determinada fecha, por ejemplo, `date < '2023-07-01'`. (Suponiendo que el campo `date` es tipo `chararray` con formato `'YYYY-MM-DD'`).
   - De ese conjunto, filtra adicionalmente las transacciones con `total_amount > 500`.
   - Muestra el resultado, y calcula la edad promedio (`AVG(age)`) de los clientes que cumplen estas condiciones.
9. ***Script* completo + almacenamiento**
   - Crea un script `.pig` que contenga los pasos: carga, filtrado, transformación, agrupación, ordenación, y finalmente almacenamiento (`STORE`) del resultado final en un directorio (por ejemplo `/usr/maria_dev/ventas_analisis`). Debes crear tu los filtros, transforamciones, etc. que desees.
   - Asegúrate de comentar la operación de cada bloque del *script* con `--comentario`.
   - Ejecuta el script en modo MapReduce estándar (`pig script.pig`).
   - Verifica los archivos de salida y comprueba que los resultados tienen sentido.

CONTENIDO
APARTADO C

INVESTIGA

Implementa un contador de palabras (cuantas veces aparece cada palabra en un texto)

1. Localiza en Internet una versión del ***Quijote*** en formato texto. Descárgala y cópiala a en tu sistema HDFS.
   - Implementa en PIG el script necesario para hacer dicha operación.
   - Muestra un ejemplo de ejecución sobre El Quijote en pantalla.
   - Almacena la salida en una carpeta de HDFS llamada `/usr/maria_dev/*pig_quijote*.`