Práctica 1 Sqoop

Objetivo: Que el alumno importe datos de una base de datos hacia un HDFS con Sqoop.

ENTREGABLES: Entregar vía correo electrónico, el día estipulado como fecha de entrega, lo siguiente:

1. Archivo que contenga las capturas de pantalla mostrando la evidencia de realización de todas y cada una de las actividades, así como de sus correspondientes resultados.

NOTA: Los archivos deben estar adjuntos al correo electrónico, NO deben ser parte del texto del mensaje.

Datos:
Ubicación de archivos: /root/devph/labs/Lab3.1
Salida esperada: Usted habrá importado datos de MySQL hacia folders en HDFS
Pre-requisito: Cluster HDP2.1 debe estar arriba y ejecutando en la VM.

NOTA: La MV se encuentra configurada para conectarse automáticamente al nodo donde se realizarán las prácticas.

Actividad 1: Crear una tabla salaries en MySQL.

a) En la línea de comandos, ubicarse en el directorio /root/devph/labs/Lab3.1/
b) Observar el contenido del archivo salaries.txt que tiene campos gender, age, salary, zipcode.
c) Copiar el archivo salaries.txt al directorio /tmp
d) Revisar y ejecutar en MySQL el script de sql: salaries.sql en la base de datos test.

Actividad 2: Verificar la existencia de la tabla salaries.

a) Entrar a sesión de mysql.
b) Ubicarse en la base de datos test.
c) Mostrar las tablas existentes en esa base de datos y verificar que se encuentre salaries.
d) Obtenga los 10 primeros registros de la tabla salaries para verificar que ésta contenga datos.
e) Salir de MySQL.

Actividad 3: Importar la tabla salaries hacia HDFS.

a) INVESTIGUE que comando de sqoop permite importar la tabla salaries hacia HDFS
b) Un job de MapReduce debería empezar a ejecutarse y puede que tarde un par de minutos.

Actividad 4: Verificar que el proceso de importación se haya realizado exitosamente.

a) Verificar el contenido en el folder HDFS.
b) Usted deberá ver un nuevo folder con nombre salaries. Observe su contenido.
c) Note que existen cuatro nuevos archivos en el folder. ¿Por qué hay cuatro de estos archivos?
d) Con el comando cat observe el contenido de los archivos. Note que el contenido de estos archivos son registros de la tabla salaries. De ser así, la importación de la tabla con todas sus columnas fue exitosa.

Actividad 5: Especifique de una tabla las columnas a importar.

a) Use el parámetro –columns en el comando sqoop que importe las columnas salary y age (en ese orden) de la tabla salaries a un directorio en HDFS con nombre salaries2. Además, asigne el parámetro –m a 1 de tal forma que el resultado sea en un solo archivo.
b) Posteriormente verifique que solo se haya creado un archivo en salaries2.
c) Verifique que el contenido corresponda con las columnas especificadas.

Actividad 6: Importar datos a partir de una consulta

a) Escriba un comando sqoop que permita importar registros de la tabla salaries en MySQL, cuyo salario sea mayor que 90,000.00. Use el campo gender como el valor --split-by, especifique solo dos mappers e importe los datos hacia el archivo salaries3 en el folder HDFS.
b) Verifique y muestre el resultado esperado.
c) Observe el contenido de los archivos generados.
d) Verifique que los archivos de salida contienen solo los registros cuyo salario sea mayor a 90,000.00.

Práctica 2 Sqoop.

Objetivo: Que el alumno exporte datos de HDFS a una tabla de base de datos MySQL.

ENTREGABLES: Entregar vía correo electrónico, el día estipulado como fecha de entrega, lo siguiente:

1. Archivo que contenga las capturas de pantalla mostrando la evidencia de realización de todas y cada una de las actividades, así como de sus correspondientes resultados.

NOTA: Los archivos deben estar adjuntos al correo electrónico, NO deben ser parte del texto del mensaje.

Datos:
Ubicación de archivos: /root/devph/labs/Lab3.2
Salida esperada: Usted habrá importado datos de HDFS hacia MySQL.
Pre-requisito: Cluster HDP2.1 debe estar arriba y ejecutando en la VM.

Actividad 1: Agregar datos a HDFS.

a) En la línea de comandos, ubicarse en el directorio /root/devph/labs/Lab3.2
b) Observar el contenido del archivo salarydata.txt que tiene campos gender, age, salary, zipcode.
c) Crear un directorio en HDFS de nombre salarydata.
d) Copiar el archivo salarydata.txt en el directorio salarydata.

Actividad 2: Crear una tabla en la Base de datos test.

a) Despliegue, comprenda y ejecute el script sql salaries2.sql en MySQL.
b) Verifique que la tabla salaries2 se haya creado satisfactoriamente.

Actividad 3: Exportar los datos.

a) INVESTIGUE que comando de sqoop permite exportar el folder salarydata hacia la tabla salaries2 en MySQL. Al final de la salida de MapReduce, usted deberá ver un evento en la bitácora que diga que 10,000 registros fueron exportados.
b) Verifique que el proceso se ejecutó correctamente, al obtener el contenido de la tabla desde el prompt de MySQL.
c) Cierre sesión de MySQL.

Práctica 1 MapReduce

Objetivo: Que el alumno comprenda el funcionamiento de MapReduce.

ENTREGABLES: Entregar vía correo electrónico, el día estipulado como fecha de entrega, lo siguiente:

1. Archivo que contenga las capturas de pantalla mostrando la evidencia de realización de todas y cada una de las actividades, así como de sus correspondientes resultados.

NOTA: Los archivos deben estar adjuntos al correo electrónico, NO deben ser parte del texto del mensaje.

Datos:
Ubicación de archivos: /root/devph/labs/demos

Actividad 1: Agregar datos a HDFS.

a) En la línea de comandos, ubicarse en el directorio /root/devph/labs/demos.
b) Use el comando more para ubicar el archivo constitution.txt.
c) Agregue el archivo a HDFS.

Actividad 2: Ejecute el Job WordCount.

a) El siguiente comando ejecuta un job contador de palabras sobe el archivo constitution.txt y escribe la salida a wordcount_output.
    i. yarn jar /usr/hdp/current/hadoop-mapreduce-historyserver/hadoop-mapreduce-examples.jar wordcount constitution.txt wordcount_output
b) Note que un job MapReduce se somete al cluster para ejecución. Espere a que el job se termine de ejecutar.

Actividad 3: Observar los resultados.

a) Vea el contenido del folder wordcount_output.
b) ¿Por qué hay un archivo *part-r en este directorio?
c) ¿Qué significa la "r" en el nombre del archivo?
d) Observe el contenido de part-r-00000
e) ¿Por qué las palabras están ordenadas alfabéticamente?
f) ¿Cuál fue la clave de salida del reductor WordCount?
g) ¿Cuál fue el valor de salida del reductor WordCount?
h) Con base en la salida del reductor, ¿Qué piensa usted que hayan sido las parejas key/value?

Actividad 4: Ejecución de un Job MapReduce
El job MapReduce que se ejecutará es una aplicación de Indice invertido, uno de los primeros casos de uso de MapReduce.

Ruta de archivos: /root/devph/labs/Lab4.1
Salida esperada: Usted podrá observar los resultados de un job de índice invertido en el folder inverted/output en HDFS

a) En la línea de comandos, ubicarse en el directorio /root/devph/labs/Lab4.1
b) Use el comando more para ubicar el archivo hortonworks.txt. Cada línea de texto consiste de una URL de página web, seguida de una lista de palabras clave separadas por coma encontradas en esa página.
c) Cree un nuevo folder en HDFS de nombre inverted-input
d) Agregue el archivo hortonworks a HDFS en el folder inverted-input. Este archivo será la entrada al job MapReduce.

Actividad 5: Ejecute el Job Inverted Index
a) Desde el folder /root/devph/labs/Lab4.1 ejecute el siguiente comando
    i. hadoop jar invertedindex.jar inverted.IndexInverterJob inverted-input inverted-output
b) Espere a que el job se termine de ejecutar. La salida deberá indicar
"File Input Format Counters
Bytes Read= 1126…"

Actividad 6: Observe los resultados
a) Liste el contenido del folder inverted-output con hadoop fs –ls ..
b) ¿Cuántos reductores realizaron esta tarea?
c) ¿Cómo puede usted determinar lo anterior desde el contenido de inverted-output?
d) Con el comando cat observe el contenido de inverted-output/part-r-00000

Actividad 7: Especifique el número de reductores
a) Ejecute el job de nuevo, pero esta vez especifique el número de reductores a 3
b) Observe el contenido de los tres archivos. ¿Cómo el marco de trabajo de MapReduce determinó cuales pares de key/value enviar a cada reductor?

NOTA: Usted ahora ha ejecutado un job MapReduce desde la línea de comandos que toma un archivo de texto como entrada y obtiene como resultado los índices invertidos de las líneas de texto. Esta tarea común es la que las máquinas de búsqueda como Google y Yahoo utilizan para determinar las páginas asociadas con argumentos de búsqueda.

Práctica 1 Pig

Objetivo: Que el alumno entienda los conceptos de los scripts Pig y las relaciones Pig

ENTREGABLES: Entregar vía correo electrónico, el día estipulado como fecha de entrega, lo siguiente:

1. Archivo que contenga las capturas de pantalla mostrando la evidencia de realización de todas y cada una de las actividades, así como de sus correspondientes resultados.

NOTA: Los archivos deben estar adjuntos al correo electrónico, NO deben ser parte del texto del mensaje.

Tarea 1: Entender conceptos y comportamiento de Pig.

Datos:
Ubicación de archivos: /root/devph/demos/

Actividad 1: Arrancar el Grunt Shell

a) En la línea de comandos, ubicarse en el directorio /root/devph/demos/
b) Observar el contenido del archivo pigdemo.txt
c) Ejecute el binario pig para arrancar el Grunt Shell
d) Note que la salida incluye la ubicación de la sesión de pig y la conexión al sistema de archivos Hadoop

Actividad 2: Crear un nuevo directorio

a) Note que puede ejecutar comandos HDFS fácilmente desde el Grunt Shell, ejecute ls
b) Cree un nuevo directorio de nombre demos
c) Use copyFromLocal para copiar el archivo pigdemo.txt al folder demos
d) Verifique que el archivo fue cargado satisfactoriamente con ls
e) Ubíquese al nuevo directorio de trabajo demos
f) Observe el contenido del directorio demos con cat

Actividad 3: Defina una relación

a) Defina la relación employees, usando el esquema (state, name)
b) Con describe observe como es la relación employees. Note que los campos tienen un tipo de dato.
c) Observe los registros de la relación employees con DUMP. Note que esto requiere un job MapReduce y el resultado es un conjunto de tuplas

Actividad 4: Filtre la relación por campo

a) Filtre la información de la relación employees para aquellas tuplas cuyo campo state sea igual a ‘CA’, la relación resultante tendrá nombre ca_only, observe el resultado con DUMP

Actividad 5: Crear un grupo

a) Defina la relación emp_group que agrupe los empleados por estado
b) Las cajas representan grupos en Pig. Una caja es una colección desordenada de tuplas, observe

el contenido de emp_group con DUMP.

c) Todos los registros con el mismo estado deberán estar agrupados. Note que las tuplas se despliegan entre paréntesis, las llaves representan cajas.

Actividad 6: El comando STORE

a) El comando DUMP vacia el contenido de una relación a la consola. El comando STORE envía la salida a un folder en HDFS. Por ejemplo: `grunt> STORE emp_group INTO 'emp_group';`
b) Verifique que un nuevo folder se ha creado con ls.
c) Observe el contenido del folder con cat.

Actividad 7: Muestre todos los alias

a) El comando aliases muestra una lista de los alias definidos, ejecútelo.

Actividad 8: Monitoreo de Jobs Pig

a) Con el navegador observe el historial de Jobs a traces de una interface de usuario en http://sandbox:19888/
b) Observe la lista de Jobs, los cuales deben contener los Jobs MapReduce que han sido ejecutados desde el Pig Latin en el Grunt Shell
c) Note que puede observar las bitácoras del ApplicationMaster y también cada tarea de mapeo y reducción. Nota existen tres comando que disparan Jobs MapReduceSTORE, DUMP e ILLUSTRATE.

Tarea 2: Introducción a Pig

Datos:
Ubicación de archivos: /root/devph/labs/Lab5.1
Salida: Tendrá programas Pig que cargaran datos de visitantes de la Casa Blanca con y sin esquema, almacenar la salida a una relación hacia un folder en HDFS.

Actividad 1: Observar los datos crudos

a) Cambie de directorio a /root/devph/labs/Lab5.1
b) Descomprima el archivo whitehouse_visits.txt
c) Observe el contenido de este archivo con tail.

Actividad 2: Cargue los datos a HDFS

a) Arranque el Grunt Shell
b) Cree un nuevo directorio en HDFS con nombre whitehouse
c) Use el comando copyFromLocal para copiar el archivo whitehouse_visits.txt al folder whitehouse en HDFS, renombrando el archivo a visits.txt (asegúrese que todo el comando este en una sola línea).
d) Con el comando ls verifique que el archivo fue cargado satisfactoriamente.

Actividad 3: Defina una relación

a) Use TextLoader para cargar el archivo visits.txt a una relación con nombre A. Nota: El TextLoader simplemente crea una tupla por cada línea de texto y usa un solo campo chararray que contiene la línea completa. Esto permite cargar líneas de texto y no preocuparse por el formato o esquema aun.
b) Use DESCRIBE para verificar que A no tiene esquema
c) Use el operador LIMIT para definir una nueva relación A_limit que contenga solo 10 registros de A
d) Use el operador DUMP para ver el contenido de A_limit.

Actividad 4: Defina un esquema

a) Cargue los datos de la Casa Blanca de nuevo en B, pero esta vez use el cargador PigStorage y defina un esquema parcial:

```
grunt> B = LOAD '/user/root/whitehouse/visits.txt' USING PigStorage(',') AS (
lname:chararray,
fname:chararray,
mname:chararray,
id:chararray,
status:chararray,
state:chararray,
arrival:chararray
   );
```

b) Use el comando DESCRIBE para ver el esquema

Actividad 5: Comando STORE

a) Con el comando STORE para almacenar la relación B a un folder de nombre whose_tab y separe los campos de cada registro con tabulador.
b) Verifique que el folder whouse_tab haya sido creado con ls
c) Observe uno de los archivos salida para verificar que contienen la relación B en formato por tabuladores, con fs –tail
d) Cada registro debe contener siete campos. Que paso con el resto de los campos de los datos crudos del archivo whitehouse/visitors.txt?___

Actividad 6: Use JsonStorage

a) Ejecute el siguiente comando para almacenar la misma relación pero ahora en formato JSON:

```
grunt> store B into 'whouse_json' using JsonStorage();
```

b) Verifique que el folder whouse_json fue creado con ls
c) Observe uno de los archivos salida con fs –tail.

Práctica 2 Pig.

Objetivo: Que el alumno explore relaciones Pig.

ENTREGABLES: Entregar vía correo electrónico, el día estipulado como fecha de entrega, lo siguiente:

1. Archivo que contenga las capturas de pantalla mostrando la evidencia de realización de todas y cada una de las actividades, así como de sus correspondientes resultados.

NOTA: Los archivos deben estar adjuntos al correo electrónico, NO deben ser parte del texto del mensaje.

Actividad 1: Cargue los datos de White House

a) Defina la relación A= LOAD '/user/root/whitehouse/' USING TextLoader();

Actividad 2: Contar el número de lineas

a) Defina una nueva relación B que contenga la agrupación de todos los registros de A
b) Muestre el esquema
c) ¿Cuál es el tipo de dato del campo de agrupación?
d) ¿De dónde proviene este tipo de dato?
e) ¿Por qué el campo A de B no tiene esquema?
f) ¿Cuántos grupos hay en la relación B?
g) El campo A de la tupla B es una Bag de todos los registros en visits.txt. Use la función COUNT en esta Bag para determinar cuantas líneas de texto hay en visits.txt y guárdelo en A_count
h) Use DUMP en A_count para ver el resultado.

Actividad 3: Analice el contenido de los datos

a) Observe los datos de los campos de cada registro, Cargue los datos con PigStorage y déjelo en visits con delimitador por coma.
b) Use FOREACH… GENERATE para definir una relación que sea una proyección de los primeros 10 campos de la relación en firstten
c) Use LIMIT para desplegar solo 50 registros en firstten_limit y despliegue con DUMP firstten_limit

Actividad 4: Localice al presidente de USA

Existen 26 campos en cada registro, uno de ellos representa al visitado. El objetivo es ubicar esta columna y determinar quién ha visitado al presidente de USA, (President Of The US, POTUS)

a) Defina una relación lastfields que proyecte los últimos siete campos (19 a 25) de visits.
b) Use LIMIT para obtener solo 500 registros y guárdelo en lastfields_limit
c) Despliegue el contenido de lastfields_limit
d) Use FILTER para definir una relación potus que contenga solamente los registros de visits donde el campo del visitado corresponda con "POTUS", limite la salida a 500 registros en potus_limit, muestre la salida de potus_limit que debe incluir solo visitantes que se reunieron con el presidente.

Actividad 5: Cuente el número de Visitantes de POTUS

a) Genere la relación potus con FILTER
b) Agrupe potus en potus_group
c) Genere relación potus_count con FOREACH potus_group
d) Despliegue el contenido, deberían salir 21,819 visitantes

Actividad 6: Observe los archivos bitácora de Pig

Cada vez que usted ejecuta DUMP o STORE, se ejecuta un job MapReduce en su cluster. Usted puede observar las bitácoras en el JobHistory con el browser en http://sandobox:19888/

Práctica 3 Pig

Objetivo: Que el alumno separe un conjunto de datos en dos a fin de optimizar rendimiento y realice joins entre dos fuentes de datos con Pig.

ENTREGABLES: Entregar vía correo electrónico, el día estipulado como fecha de entrega, lo siguiente:

1. Archivo que contenga las capturas de pantalla mostrando la evidencia de realización de todas y cada una de las actividades, así como de sus correspondientes resultados.

NOTA: Los archivos deben estar adjuntos al correo electrónico, NO deben ser parte del texto del mensaje.

Actividad 1: Explore el campo Comments

a) Desde el Pig Grunt Shell genere una relación visits a partir de /whitehouse/visits.txt con PigStorage(',')
b) El campo $25 corresponde a comentarios. Genere una relación not_null_25 filtrando visits para que obtengan solo las tuplas cuyo campo Comments no tenga nulos.
c) Genere una relación comments que contenga solo la columna $25 partir de not_null_25
d) Muestre el esquema de la relación comments

Actividad 2: Prueba de consistencia sobre comments

a) Una tarea de Pig muy común es checar que una relación sea consistente con lo que se pretende hacer con ella. Pero usando DUMP en una relación Big Data puede tomar mucho tiempo. Por tanto, genere una relación comments_sample que sea una muestra del 1% de comments
b) Ejecute un DUMP sobre comments_sample y coteje que no contiene valores nulos.
c) Agrupe comments en comments_all y cuente el número de tuplas de comments_all y genere con esto la relación comments_count
d) Dump comments_count debe resultar en 222,839 tuplas

Actividad 3: Separe una fuente de datos en dos; una cuyo comments contenga el valor "CONGRESS" y otra fuente de datos cuyo comments no contenga valor "CONGRESS"

a) Utilice el comando SPLIT para generar dos relaciones congress y not_congress a partir de la relación visits.
b) Use STORE para almacenar la relación congress en el folder congress usando formato JSON
c) Similarmente almacene la relación no_congress en el folder not_congress
d) Observe los folders generados con ls y escriba los tamaños correspondientes.
e) Despliegue la salida de las dos relaciones y coteje que los resultados sean los esperados.

Actividad 4: Juntando fuentes de datos con Pig con Hash-join

Ubicación de archivos: /root/devph/labs/Labs6.2

a) Suba el archivo /root/devph/labs/Labs6.2/congress.txt hacia whitehouse en HDFS
b) Use hadoop fs –ls para verificar que el archivo congress.txt esta en Whitehouse y hadoop fs –cat para observar su contenido.
c) Usted debe crear un archivo script de Pig con el editor de textos gedit con el icono que se encuentra del lado izquierdo de su VM, el script se llamara join.pig en el folder

/devph/labs/Lab6.2.

d) Defina las siguientes relaciones que contendrán los campos lname: chararray y fname: chararray
```
visitors = LOAD 'whitehouse/visits.txt' USING PigStorage(',') AS
(lname:chararray, fname:chararray);
```
e) Defina una proyección de los datos de Congreso:
```
congress = LOAD 'whitehouse/congress.txt' AS (
full_title:chararray,
district:chararray,
title:chararray,
fname:chararray,
lname:chararray,
party:chararray
);
```
f) Los nombres en visits.txt están todos en mayúsculas, pero los nombres en congress.txt no. Por tanto, defina una proyección en la relación congress que consista de los siguientes campos:
```
congress_data = FOREACH congress GENERATE
district,
UPPER(lname) AS lname,
UPPER(fname) AS fname,
party;
```
g) Defina una nueva relación con nombre join_contact_congress que corresponda al join de visitors y congress_data. Realice el join sobre el nombre y el apellido.
h) Con el comando STORE almacene el resultado de join_contact_congress en un directorio joinresult
i) Abra una ventana de terminal y cambie de directorio a /root/devph/labs/Lab6.2
j) Ejecute el script join.pig
k) Espere a que el job MapReduce termine y apunte el número de segundos que tomó el join, a través de la diferencia entre StartedAt y FinishedAt
l) El tipo de join usado es también salida en el job de estadísticas. Note la salida de estadísticas que tiene HASH_JOIN debajo de la columna Features, lo cual significa que se ejecutó un hashjoin.
m) Observe el resultado dentro del folder joinresult con hadoop fs –cat joinresult/part-r-00000

Actividad 5: Juntando fuentes de datos con Pig con Replicated-join

a) Borre el directorio joinresult en HDFS con hadoop fs .-rm
b) Modifique el script join.pig y corra el script de nuevo
c) Note que la salida de las estadísticas usaron REPLICATED_JOIN
d) Compare el tiempo de ejecución entre los dos joins. Y anote el valor
¿Hubo una mejora o no en el rendimiento?

Práctica 4 Pig

Objetivo: Que el alumno transforme y exporte conjunto de datos con Pig para su uso con Hive.

ENTREGABLES: Entregar vía correo electrónico, el día estipulado como fecha de entrega, lo siguiente:

1. Archivo que contenga las capturas de pantalla mostrando la evidencia de realización de todas y cada una de las actividades, así como de sus correspondientes resultados.

NOTA: Los archivos deben estar adjuntos al correo electrónico, NO deben ser parte del texto del mensaje.
Datos:
Ubicación de archivos: /root/devph/labs/Lab6.3
Salida satisfactoria: Un script Pig que almacene una proyección de visits.txt en un folder en el warehouse de Hive de nombre wh_visits.

Actividad 1: Leer y comprender el script Pig wh_visits.pig

a) Desde la línea de comandos, ubíquese en el directorio /root/devph/labs/Lab6.3/. Observe el contenido del script wh_visits.pig
b) Note que todos los visitantes de la Casa Blanca que se reunieron con el Presidente están en la relación potus.
c) Note que la relación Project_potus es una proyección de last_name, first_name, time y comments acerca de la visita

Actividad 2: Almacene la proyección en el warehouse de Hive

a) Use gedit para editar el script wh_visits.pig
b) Agregue el siguiente comando al final del archivo, éste comando almacena la relación Project_potus en un folder específico.

```
STORE project_potus INTO '/apps/hive/warehouse/wh_visits/';
```

Actividad 3: Ejecute el script

a) Guarde los cambios realizados al script.
b) Ejecute el script con el siguiente comando

```
# pig wh_visits.pig
```

Actividad 4: Observe los resultados

a) Presente la vista del contenido del folder con ls.
b) Presente el contenido de uno de los archivos generados con cat

Práctica 5 Pig

Objetivo: Que el alumno utilice la función PageRank, analice datos de sesión Clickstream y calcule cuantiles a través de la librería DataFu

ENTREGABLES: Entregar vía correo electrónico, el día estipulado como fecha de entrega, lo siguiente:

1. Archivo que contenga las capturas de pantalla mostrando la evidencia de realización de todas y cada una de las actividades, así como de sus correspondientes resultados.

NOTA: Los archivos deben estar adjuntos al correo electrónico, NO deben ser parte del texto del mensaje.
Datos:
Ubicación de archivos: /root/devph/labs/demos

Tarea 1: Utilizar función PageRank
Actividad 1: Familiarizarse con los datos del archivo edges.txt

a) Desde la línea de comandos, ubíquese en el directorio //root/devph/labs/demos. Observe el contenido del archivo edges.txt
b) La primer columna es el tópico, dado que solo se tiene una sola gráfica, el tópico 0 es para todos los bordes.
c) La segunda y tercera columnas son el inicio y fin de cada borde. Por ejemplo, en el primer registro, hay un borde de 2 a 3.
d) La cuarta columna es el peso del borde. La gráfica esta tiene a todos los bordes igualmente ponderados.
e) Con base en los datos, ¿cuáles paginas deberían ser clasificadas en primer lugar?

Actividad 2: Cargar los datos a HDFS

a) Cargue el archivo edges.txt a HDFS

Actividad 3: Defina la UDF PageRank

a) Observe el contenido de `/root/devph/labs/demos/pagerank.pig` con cat o gedit. Las primeras dos líneas registran la librería DataFu y definen la función PageRank.

```
register /root/devph/labs/Lab6.5/datafu-1.2.0.jar;
define PageRank datafu.pig.linkanalysis.PageRank();
```

b) Los bordes son cargados y agrupados por tópico y fuente.

```
topic_edges = LOAD '/user/root/edges.txt' as
(topic:INT,source:INT,dest:INT,weight:DOUBLE);
topic_edges_grouped = GROUP topic_edges by (topic, source);
```

c) Los datos son preparados para la función PageRank, que espera un tópico, una fuente y sus bordes:

```
topic_edges_data = FOREACH topic_edges_grouped GENERATE
group.topic as topic,
group.source as source,
topic_edges.(dest,weight) as edges;
```

d) En el ejemplo se tiene un solo tópico, pero los bordes necesitan ser agrupados por tópico:
```
topic_edges_data_by_topic = GROUP topic_edges_data
BY topic;
```

e) Se invoca la función PageRank:
```
topic_ranks = FOREACH topic_edges_data_by_topic GENERATE
group as topic,
FLATTEN(PageRank(topic_edges_data.(source,edges)));
```

f) Los resultados se almacenan en HDFS
```
store topic_ranks into 'topicranks';
```

Actividad 4: Ejecute el script

a) Desde la línea de comandos ejecute el script pagerank.pig, el job tardará unos minutos en ejecutarse.

Actividad 5: Observe los resultados

a) Desde la línea de comandos observe el contenido del archivo topicranks en HDFS con ls.
b) Observe el contenido del archivo de salida con cat.

Actividad 6: Analice los resultados

a) ¿Cuál página fue clasificada en primer lugar?
b) ¿Cuál página fue clasificada en último lugar?

Tarea 2: Analizar datos de sesión en páginas web (Clickstream)

Datos:
Ubicación de archivos: /root/devph/labs/lab6.4
Salida esperada: Se calculará la longitud de cada sesión junto con sus valores promedio y mediana.

Actividad 1: Observe los datos

a) Ubíquese en el directorio /root/devph/labs/lab6.4
b) Observe el contenido del archivo clicks.csv. La primer columna es el identificador de usuario, la segunda columna es el tiempo de realización de click, almacenado como tipo long, la tercer columna es la dirección URL visitada.
c) Ponga el archivo en HDFS

Actividad 2: Defina función Sessionize

a) Use el editor gedit y edite el archivo /root/devph/labs/lab6.4/sessions.pig
b) Note que se registran dos archivos JAR: `datafu-1.2.0.jar` y `piggybank.jar`. El JAR contiene la función Sessionize que se usara y el jar piggybank contiene una función de tiempo de nombre UnixToISO que está ya definida en el script Pig.
c) Agregue la siguiente sentencia de definición de la UDF Sessionize:
```
DEFINE Sessionize datafu.pig.sessions.Sessionize('8m');
```

d) ¿Qué significa el '8m' en el constructor?

Actividad 3: Aplique la función Sessionize

a) Note que el archivo clicks.csv es cargado en sessions.pig

```
clicks = LOAD 'clicks.csv' USING PigStorage(',')
AS (id:int, time:long, url:chararray);
```

b) Note que la relación clicks es proyectada a clicks_iso con el tipo de dato long convertido a formato de tiempo ISO y posteriormente agrupado en la relación clicks_group

```
clicks_iso = FOREACH clicks GENERATE UnixToISO(time)
AS isotime, time, id;
clicks_group = GROUP clicks_iso BY id;
```

c) Aplique la función Sessionize a los clicks agregando el siguiente ciclo anidado FOREACH

```
clicks_sessionized = FOREACH clicks_group {
sorted = ORDER clicks_iso BY isotime;
GENERATE FLATTEN(Sessionize(sorted))
AS (isotime, time, id, sessionid);
}
```

d) En el script, agregue la línea que corresponda para realizar un DUMP a los datos clicks_sessionized

e) Guarde los cambios en sessions.pig

Actividad 4: Ejecute el script

a) Verifique que la función Sessionize está trabajando ejecutando el siguiente script:

```
# pig sessions.pig
```

b) Verifique que la última sesión de la salida sea similar a lo siguiente:

```
(2013-01-10T07:15:20.520Z,1357802120520,2,51d89b38-b14a-4158-8703-
724525d9f787)
(2013-01-10T07:15:39.797Z,1357802139797,2,51d89b38-b14a-4158-8703-
724525d9f787)
(2013-01-10T07:26:30.602Z,1357802790602,2,711525c4-eff6-4697-ade7-
e2ad5ec555e5)
(2013-01-10T07:26:53.357Z,1357802813357,2,711525c4-eff6-4697-ade7-
e2ad5ec555e5)
(2013-01-10T07:26:58.800Z,1357802818800,2,711525c4-eff6-4697-ade7-
e2ad5ec555e5)
(2013-01-10T07:27:05.253Z,1357802825253,2,711525c4-eff6-4697-ade7-
e2ad5ec555e5)
(2013-01-10T07:27:57.844Z,1357802877844,2,711525c4-eff6-4697-ade7-
e2ad5ec555e5)
(2013-01-10T07:28:20.610Z,1357802900610,2,711525c4-eff6-4697-ade7-
e2ad5ec555e5)
(2013-01-10T07:29:01.128Z,1357802941128,2,711525c4-eff6-4697-ade7-
e2ad5ec555e5)
```

Actividad 5: Calcule la longitud de la sesión con el script sessions.pig

a) En el script, invalide la línea DUMP clicks_sessionized con –
b) Agregue código para definir una relación de nombre sessions que sea la proyección de los campos time y sessionid de la relación clicks_sessionized con FOREACH
c) Agregue código para definir una relación de nombre sessions_group a partir de sessions pero que agrupe por sessionid
d) Agregue código para definir una relación session_times usando la siguiente proyección que calcula la longitud de cada sesión.

```
session_times = FOREACH sessions_group
GENERATE group as sessionid,
(MAX(sessions.time) - MIN(sessions.time)) / 1000.0 / 60
as session_length;
```

e) Realice un DUMP a session_times
f) Guarde los cambios a sessions.pig y ejecute el script. La salida debe ser similar a lo siguiente:

```
(01e5259c-c5a6-45b0-8d04-1be86182d12e,0.16571666666666665)
(164be386-1df2-40dd-9331-563e1b8a7275,4.030883333333334)
(16ab9225-28d3-45f6-9d07-f065223046bb,38.809916666666666)
(18362695-d032-424a-a983-33ab45638700,0.0)
(2699ef77-bd37-4611-a239-ddbd80066043,10.398116666666665)
(3077f9d1-a5d5-4bf9-8212-87ae848b4ed8,3.44485)
(3e732d19-e3ed-4cc4-810f-f05c8534fb28,1.1402833333333333)
(455183ea-c3bb-43fe-9f07-63e0c0199008,14.648516666666666)
(5a65d8dc-1a4e-4355-b86a-f1efc519b084,63.620149999999995)
(5ef45fc4-01df-40d8-805f-a61c60fc421e,0.03173333333333333)
(61e14bcf-1fb4-4f7e-a3b4-2b67b8840756,1.0819833333333333)
(63b53f03-31e9-4a01-8029-6334020080e4,4.48765)
(66f58bc2-7aeb-487d-a28e-21090578cfe2,22.9298)
```

g) ¿Cuánto tiempo duró la sesión mas larga?

Actividad 6: Calcule la duración promedio de las sesiones

a) En el script, invalide la línea DUMP session_times con –
b) Defina una relación con nombre sessiontimes_all que corresponda a la agrupación de todos los session_times
c) Defina la relación sessiontimes_avg usando la siguiente sentencia:

```
sessiontimes_avg = FOREACH sessiontimes_all
GENERATE AVG(session_times.session_length);
```

d) Agregue DUMP sessiontimes_avg
e) Guarde los cambios a sessions.pig y ejecute el script de nuevo.
f) Verifique la salida, que deberá ser un solo valor correspondiente al tiempo promedio de sesión.
g) ¿Cuál fue el valor?

Actividad 7: Calcule la mediana de la longitud de las sesiones

a) Usando la relación sessiontimes_avg como ejemplo, calcule la mediana. Usted necesitara

definir un función Median a partir de la librería DataFu de nombre datafu.pig.stats.Median()

b) ¿Cuál fue el valor de la mediana?

Tarea 3: Analizar Datos de la bolsa utilizando Cuantiles

Datos:
Ubicación de archivos: /root/devph/labs/lab6.5
Salida esperada: Se calculará la longitud de cada sesión junto con sus valores promedio y mediana.

Actividad 1: Revise los datos de la bolsa

a) Ubíquese en el directorio /root/devph/labs/lab6.5
b) Observe con tail el contenido del archivo stocks.csv, que contiene los precios históricos para las acciones de la casa de bolsa de New York, que empiezan con la letra "Y". La primer columna contiene siempre el valor NYSE, la segunda columna es el símbolo del stock, la tercer columna es la fecha en que los precios ocurrieron. Las siguientes columnas son las características de los precios de las operaciones: abre, alto, bajo, cierre y volumen.
c) Ponga el archivo stocks.csv en el folder HDFS /user/root

Actividad 2: Defina la función Quantile

a) Usando gedit, cree un nuevo archivo de texto en el folder /root/devph/labs/lab6.5 con nombre quantile.pig
b) En la primer línea del archivo, registre el JAR datafu
c) Defina la función datafu.pig.stats.Quantile como un cuantil y pase los valores para cálculo de cuartiles de un conjunto de números:

```
define Quantile datafu.pig.stats.Quantile(
'0.0','0.25','0.50','0.75','1.0');
```

Actividad 3: Cargue los valores

a) Codifique el siguiente comando LOAD para cargar los primeros cinco valores de cada registro:

```
stocks = LOAD 'stocks.csv' USING PigStorage(',') AS
   (nyse:chararray,
   symbol:chararray,
   closingdate:chararray,
   openprice:double,
   highprice:double,
   lowprice:double);
```

Actividad 4: Filtre los valores nulos

a) La función Quantile falla si cualquiera de los valores otorgados es nulo. Defina una relación de nombre stocks_filter que filtre la relación stocks donde el highprice no sea nulo.

Actividad 5: Agrupe los valores

a) Dado que se desea calcular los cuantiles para cada stock individual (al contrario de todos los precios que empiezan con Y), defina una relación stocks_group que agrupe la relación stock_filter por symbol.

Actividad 6: Calcule los cuantiles

a) Defina la siguiente relación que invoca el método Quantile sobre los valores de highprice.

```
quantiles = FOREACH stocks_group {
   sorted = ORDER stocks_filter BY highprice;
   GENERATE group AS symbol,
   Quantile(sorted.highprice) AS quant;
   }
```

b) ¿Cuántas veces la función quantile será invocada en la sentencia anidada FOREACH del inciso b?

c) Agregue la sentencia DUMP para ver la relación quantiles

Actividad 7: Ejecute el script

a) Guarde los cambios en quantile.pig

b) Ejecute el script

c) Como hay información de stocks en los datos de entrada, la salida serán los cuales del precio más alto de estos 5 stocks.

```
(YGE,(3.22,10.97,14.79,19.6,41.5))
(YPF,(9.0,23.62,31.94,41.47,69.98))
(YSI,(1.56,8.04,16.435000000000002,19.93,23.61))
(YUM,(21.9,32.08,37.85,48.91,73.87))
(YZC,(4.41,14.4,20.795,47.13,116.73))
```

Actividad 8: Calcule la mediana

a) Ahora que tiene un script de Pig para el cálculo de cuantiles de los precios altos de stocks, verifique si puede modificar el script (son pocos cambios) para calcular la mediana en un script de nombre median.pig

Objetivo: Que el alumno comprenda como se almacena una tabla Hive en HDFS, así como particionamiento y sesgos de la información en Hive.

ENTREGABLES: Entregar vía correo electrónico, el día estipulado como fecha de entrega, lo siguiente:

1. Archivo que contenga las capturas de pantalla mostrando la evidencia de realización de todas y cada una de las actividades, así como de sus correspondientes resultados.

NOTA: Los archivos deben estar adjuntos al correo electrónico, NO deben ser parte del texto del mensaje.
Datos:
Ubicación de archivos: /root/devph/labs/7.1
Requerimientos: Poner datos en el folder wh_visits, es decir, realizar laboratorio 4 de Pig.

Tarea 1: Comprensión de tablas Hive

Actividad 1: Familiarizarse con los datos del folder wh_visits

a) Desde la línea de comandos, ubíquese en el directorio `/apps/hive/warehouse/wh_visits/` y observe el contenido, deberían existir dos archivos part-m
b) Recuerde que la proyección para crear estos archivos tuvo el siguiente esquema:

```
project_potus = FOREACH potus GENERATE
$0 AS lname:chararray,
$1 AS fname:chararray,
$6 AS time_of_arrival:chararray,
$11 AS appt_scheduled_time:chararray,
$21 AS location:chararray,
$25 AS comment:chararray ;
```

En este laboratorio, usted definirá una tabla Hive que corresponda con estos registros y contenga los datos exportados del script Pig.

Actividad 2: Defina el script Hive

a) En el directorio /root/devph/labs/7.1 existe un archivo de texto con nombre wh_visits.hive. Observe su contenido con more. Note que define una tabla Hive de nombre wh_visits con el siguiente esquema que corresponde con los datos en el directorio project_potus. Nota: no puede usar 'comment' o 'location' como nombres de columnas porque son palabras reservadas, así que se cambiaron ligeramente.
b) Ejecute el script con el comando: `hive –f wh_visits.hive`
c) Si se ejecuta de manera satisfactoria, deberá ver "OK" en la salida.

Actividad 3: Verifique la creación de la tabla.

a) Arranque el Hive Shell con el binario hive.
b) Desde el prompt de hive ejecute el comando `show tables`
c) Use el comando describe para observar el detalle de wh_visits
d) Trate de ejecutar una consulta que despliegue hasta 20 registros de wh_visits a pesar de que la

tabla este vacía. ¿Cómo es que la tabla Hive ya está poblada con registros? _____

e) ¿Por qué la consulta anterior requiere de Tez o un job MapReduce para ejecutarse? __

Actividad 4: Cuente el número de registros

a) Cuente el número de registros de wh_visits con select count(*)
b) ¿Cuántos registros contiene? __

Actividad 5: Seleccionando el nombre de archivo de entrada

a) Hive tiene dos columnas virtuales que se crean automáticamente para todas las tablas: INPUT__FILE y BLOCK__OFFSET__INSIDE__FILE. Note que entre cada palabra existen dos guiones bajos. Ejecute la siguiente consulta:

```
hive> select INPUT__FILE__NAME, lname, fname FROM wh_visits
WHERE lname LIKE
'Y%';
```

b) El resultado de esta consulta es aquellos visitantes a la Casa Blanca cuyos apellidos empiezan con "Y". Note que la salida también contiene el archivo particular de que el registro fue encontrado en:

```
hdfs://sandbox.hortonworks.com:8020/apps/hive/warehouse/wh_visits/
part-m00000
YOUNG MICHELLE
hdfs://sandbox.hortonworks.com:8020/apps/hive/warehouse/wh_visits/
part-m00001
YOUNG LEDISI
```

Actividad 6: Borre la tabla

a) Veremos que sucede cuando una tabla es eliminada. Empiece definiendo una tabla llamada names usando el Shell de Hive:

```
hive> create table names (id int, name string)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';
```

b) Use el comando Hive dfs para poner Labs7.1/names.txt en el directorio de la tabla warehouse.

```
hive> dfs -put /root/devph/labs/Lab7.1/names.txt
/apps/hive/warehouse/names/;
```

c) Observe el contenido del folder de la table warehouse
d) Desde el Shell de hive, ejecute la siguiente consulta: hive> select * from names;
e) Ahora borre la tabla names.
f) Observe el contenido del folder de la tabla warehouse de nuevo. Note que el folder names ya no está.

Actividad 7: Defina una tabla externa

a) En este paso observará como trabaja una tabla externa en Hive. Empiece poniendo el archivo names.txt en HDFS con dfs -put
b) Cree un folder en HDFS para ahí sea almacenada la tabla externa con dfs –mkdir
c) Defina la tabla names como externa con create external
d) Carga los datos a la tabla con load

e) Verifique que la carga funcionó con un select *
f) Note que el archivo names.txt ha sido movido a /user/root/hivedemo con dfs –ls
g) De manera similar, verifique que el archivo names.txt ya no está en el folder /user/root en HDFS. ¿Por qué ya no está? ___
h) Use el comando ls para verificar que el directorio /apps/hive/warehouse no contiene un subdirectorio para la tabla names con dfs –ls
i) Ahora borre la tabla names con drop
j) Observe el contenido de /user/root/hivedemo. Note que names.txt aún sigue ahí.

Tarea 2: Introducción a Particiones y Sesgos

Actividad 1: Observe los datos

a) Ubíquese en el directorio /root/devph/labs/demos y observe los archivos hivedata_<<state>>.txt

Actividad 2: Defina la tabla en Hive

a) Observe la sentencia de creación de tabla en partitiondemo.sql con more
b) Ejecute la consulta para definir la tabla names
c) Muestre las particiones con show partitions names (no habrá ninguna aún)

Actividad 3: Cargue datos a la tabla

a) Cuando se cargan datos a la tabla, se le debe especificar a qué partición. Por ejemplo:

```
hive> load data local inpath
'/root/devph/labs/demos/hivedata_ca.txt'
into table names partition (state = 'CA');
```

b) Cargue los archivos CO y SD también.
c) Verifique que todos los datos ya conforman la tabla names con select *

Actividad 4: Observe la estructura del directorio

a) Observe el contenido de /apps/hive/warehouse/names con dfs –ls
b) Note que cada partición tiene su propio subdirectorio para almacenar su contenido.

Actividad 5: Realice una consulta

a) Cuando se especifica una cláusula where que incluye una partición Hive sólo accesa los archivos en esa partición. Por ejemplo:

```
hive> select * from names where state = 'CA';
```

b) Note que no se ejecutó un job MapReduce, ¿por qué? ___
c) Se puede seleccionar el campo partition, aunque no esté realmente en el archivo de datos, Hive usa el nombre del directorio para obtener el valor:

```
hive>select name, state from names where state = 'CA'
```

d) Se pueden ejecutar consultas a lo largo del archivo completo. Por ejemplo, la siguiente consulta se ejecuta a lo largo de múltiples particiones. Cuando termine use el comando exit para salir del Shell de Hive.

```
hive> select name, state from names where state = 'CA' or state
= 'SD';
```

Actividad 6: Creación de una tabla sesgada

a) Verifique la existencia del archivo salaries.txt en el directorio /root/devph/labs/demos/ y póngalo en el directorio /user/root/sdalarydata/ en HDFS
b) Observe el contenido de demos/skewdemo.hive, el cual define una tabla sesgada con nombre skew_demo usando los datos de salaries.txt
c) ¿Cuáles son los valores que están sesgando a la tabla? ____
   Ejecute el script skewdemo.hive con hive –f
d) Observe el contenido del directorio warehouse de Hive.
e) Seleccione algunos registros para asegurarse de que la tabla tenga datos con hive –f

Práctica 2 Hive

Objetivo: Que el alumno explore técnicas de análisis de Big Data.

ENTREGABLES: Entregar vía correo electrónico, el día estipulado como fecha de entrega, lo siguiente:

1. Archivo que contenga las capturas de pantalla mostrando la evidencia de realización de todas y cada una de las actividades, así como de sus correspondientes resultados.

NOTA: Los archivos deben estar adjuntos al correo electrónico, NO deben ser parte del texto del mensaje.
Datos:
Ubicación de archivos: /root/devph/labs/7.2

Actividad 1: Encontrar la primera visita

a) Use gedit para crear un nuevo archivo de texto con nombre whitehouse.hive y guárdelo en el folder /root/devph/labs/Lab7.2
b) En este paso, el script deberá encontrar el primer visitante de la Casa Blanca con base en nuestro archivo. Será una consulta grande, empiece obteniendo todas las columnas donde el time_of_arrival no esté vacio: select * from wh_visits where time_of_arrival != ”” no ponga punto y coma al final de este paso.
c) Para encontrar la primera visita, necesitamos clasificar el resultado, esto requiere convertir la cadena time_of_arrival a un timestamp. Use la función unix_timestamp para lograrlo, también ordene por este campo convertido, no use punto y coma en este paso
d) Dado que sólo se está buscando un solo resultado, no necesitamos regresar todos los registros. Limite el resultado a 10 registros, para ver los primeros 10 visitantes, ahora si agregue el punto y coma, para terminar toda la consulta.
e) Guarde el resultado en el archivo whitehouse.hive
f) Ejecute el script whitehouse.hive y espere a que los resultados sean desplegados.
g) El primer visitante debería ser en 2009, dado que es cuando el archivo empieza a contener los datos. El primer visitante es Charles Kahn y Carol Keehan el 3/5/2009.

Actividad 2: Encuentre la última visita

a) Tome el script de la actividad anterior y modifíquelo para encontrar la última visita.
b) Ejecute la consulta de nuevo y el ultimo visitante debe ser Jackie Walker el 3/18/2011

Actividad 3: Encuentre el comentario más común

a) En este paso, se explorará el campo infor_comment para determinar el comentario más común. Utilizará algunas funciones agregadas de Hive para lograrlo. Empiece con gedit para crear un nuevo archivo de texto con nombre comments.hive y guárdelo en /root/devph/labs/lab7.2
b) Genere una consulta que despliegue los 10 comentarios más frecuentes:

```
from wh_visits
select count(*) as comment_count, info_comment
```

c) Agrupe el resultado de la consulta por la columna info_comment
d) Ordene el resultado por el campo comment_count, porque solo estamos interesados en los comentarios que aparecen más frecuentemente.

e) Como estamos interesados en los más frecuentes limite la salida a 10
f) Guarde los cambios en el archivo comments.hive y ejecute el script. Espere el job MapReduce a que se ejecute.
g) Muestre la salida de su consulta.
h) Parece que el comentario más frecuente esta en blanco, seguido de Holiday Ball.
i) Modifique la consulta para que ignore comentarios vacíos.

Actividad 4: Encuentre los comentarios menos frecuentes.

a) Ejecute la consulta anterior, pero esta vez, encuentre los 10 menos frecuentes. El primer comentario debería ser CONGRESSIONAL BALL/

Actividad 5: Analice inconsistencia de datos.

a) Analizando los resultados de los comentarios más y menos frecuentes, parece que existen variaciones de GENERAL RECPETION. En este paso, se tratará de determinar el número de visitas a POTUS que corresponden a una recepción general, tratando de limpiar algunas inconsistencias en los datos.
b) Modifique la consulta en el archivo comments.hive. En lugar de buscar comentarios vacíos, busque los comentarios que contienen variaciones de la cadena "GEN RECEP."
c) Cambie el límite de 10 a 30.
d) Ejecute la consulta de nuevo
e) Note que hay varias entradas de GENERAL RECEPTION que solo difieren por un numero al final de la abreviación GEN RECEP con `*GEN.*\\s+RECEP.*'`
f) Modifique la consulta en comments.hive para incluir %GEN% con like "%RECEP%" y "%GEN%"
g) Ponga el limite a 30, guarde los cambios y ejecute la consulta de nuevo.
h) La salida esta vez, muestra todas las variaciones de GEN y RECEP. Ahora sume el total de ocurrencias de estas variaciones, guarde los cambios y ejecute la consulta.
i) Muestre la salida, deberán existir 2697 visitas a POTUS con GEN RECP que corresponde al 12% de las 21819 visitas.

Actividad 6: Verifique el resultado

a) Tenemos 12% de visitantes a POTUS yendo de una recepción general, pero existieron muchas sentencias en los comentarios que contenían WHO y EOP. Modifique la consulta a partir del último paso y despliegue los 30 comentarios que contienen WHO y EOP.
   El resultado debe ser como sigue:

```
894 WHO EOP RECEP 2
700 WHO EOP 1 RECEPTION/
43 WHO EOP RECEP/
20 WHO EOP HOLIDAY RECEP/
13 WHO/EOP #2/
8 WHO EOP RECEPTION
7 WHO EOP RECEP
1 WHO EOP/
1 WHO EOP RECLEAR
```

b) Modifique el script de nuevo, esta vez ejecute una consulta que cuente el número de registros con WHO y EOP en los comentarios. Usted deberá obtener 1687 visitas o 7.7% de los visitantes

a POTUS, así que GENERAL RECEPTION sigue apareciendo como el comentario más frecuente.

Actividad 7: Encuentre los que más visitan.

a) Escriba un script que encuentre a las 20 personas que más visitaron a POTUS. TIP agrupe por fname y lname
b) El siguiente script ayudara al objetivo del punto anterior.

```
from wh_visits
select count(*) as most_visit, fname, lname
group by fname, lname
order by most_visit DESC
limit 20;
```

Para verificar que tu script funcionó aquí se muestra la salida correspondiente:

```
16 ALAN PRATHER
15 CHRISTOPHER FRANKE
15 ANNAMARIA MOTTOLA
14 ROBERT BOGUSLAW
14 CHARLES POWERS
12 SARAH HART
12 JACKIE WALKER
12 JASON FETTIG
12 SHENGTSUNG WANG
12 FERN SATO
12 DIANA FISH
11 JANET BAILEY
11 PETER WILSON
11 GLENN DEWEY
11 MARCIO BOTELHO
11 DONNA WILLINGHAM
10 DAVID AXELROD
10 CLAUDIA CHUDACOFF
10 VALERIE JARRETT
10 MICHAEL COLBURN
```

Práctica 3 Hive

Objetivo: Que el alumno calcule n-gramas con Hive.

ENTREGABLES: Entregar vía correo electrónico, el día estipulado como fecha de entrega, lo siguiente:

1. Archivo que contenga las capturas de pantalla mostrando la evidencia de realización de todas y cada una de las actividades, así como de sus correspondientes resultados.

NOTA: Los archivos deben estar adjuntos al correo electrónico, NO deben ser parte del texto del mensaje.
Datos:
Ubicación de archivos: /root/devph/labs/demos

Actividad 1: Crear una tabla Hive para datos.

a) Se calculan ngramas sobre la Constitución de Estados Unidos, en un archivo de texto que se encuentre en /root/devph/labs/demos/ como constitution.txt

Inicie sesión en el shell de Hive y defina la siguiente tabla:

```
create table constitution (
line string )
ROW FORMAT DELIMITED;
```

Actividad 2: Cargue la tabla

a) Cargue el archivo constitution.txt a la tabla constitution
b) Verifique que se cargaron los datos con select *

Actividad 3: Calcule un Bigrama

a) Ejecute el siguiente comando de Hive, el cual calcula un bigrama para la tabla constitution y muestra los primeros 15 resultados:

```
select explode(ngrams(sentences(line),2,15)) as x from constitution;
```

El resultado deberá ser...

```
{"ngram":["of","the"],"estfrequency":194.0}
{"ngram":["shall","be"],"estfrequency":100.0}
{"ngram":["the","United"],"estfrequency":76.0}
{"ngram":["United","States"],"estfrequency":76.0}
{"ngram":["to","the"],"estfrequency":57.0}
{"ngram":["shall","have"],"estfrequency":44.0}
{"ngram":["the","President"],"estfrequency":30.0}
{"ngram":["shall","not"],"estfrequency":29.0}
{"ngram":["in","the"],"estfrequency":28.0}
{"ngram":["by","the"],"estfrequency":25.0}
{"ngram":["the","Congress"],"estfrequency":22.0}
{"ngram":["and","the"],"estfrequency":21.0}
{"ngram":["for","the"],"estfrequency":21.0}…
…
```

Actividad 4: Calcule un Trigrama.

a) Ejecute la consulta anterior, pero esta vez, calcule un trigrama

El resultado deberá ser..


```
{"ngram":["the","United","States"],"estfrequency":68.0}
{"ngram":["of","the","United"],"estfrequency":51.0}
{"ngram":["shall","not","be"],"estfrequency":16.0}
{"ngram":["of","the","Senate"],"estfrequency":14.0}
{"ngram":["States","shall","be"],"estfrequency":13.0}
{"ngram":["House","of","Representatives"],"estfrequency":13.0}
{"ngram":["United","States","shall"],"estfrequency":13.0}
{"ngram":["shall","have","been"],"estfrequency":12.0}
{"ngram":["the","several","States"],"estfrequency":12.0}….
```


Actividad 5: Calcule un n-grama contextual.

a) Encuentre las 20 palabras más frecuentes después de la palabra “the” con context ngram

b) El resultado deberá ser


```
{"ngram":["United"],"estfrequency":76.0}
{"ngram":["President"],"estfrequency":30.0}
{"ngram":["Congress"],"estfrequency":22.0}
{"ngram":["Senate"],"estfrequency":21.0}
{"ngram":["several"],"estfrequency":15.0}
{"ngram":["Vice"],"estfrequency":12.0}
{"ngram":["State"],"estfrequency":11.0}
{"ngram":["same"],"estfrequency":10.0}….
```

Práctica 4 Hive

Objetivo: Ejecutando JOINS con archivos en HIVE y ngramas de correos electrónicos en formato Avro.

ENTREGABLES: Entregar vía correo electrónico, el día estipulado como fecha de entrega, lo siguiente:

1. Archivo que contenga las capturas de pantalla mostrando la evidencia de realización de todas y cada una de las actividades, así como de sus correspondientes resultados.

NOTA: Los archivos deben estar adjuntos al correo electrónico, NO deben ser parte del texto del mensaje.

Tarea 1: Joins de dos archivos en Hive
Datos:
Ubicación de archivos: /root/devph/labs/7.4

Actividad 1: Cargue los datos en Hive

a) Observe el contenido del archivo setup.hive en el folder /root/devph/labs/Lab7.4
b) Note que este script crea tres tablas en Hive, nyse_data que está llena de precios diarios de acciones que empiezan con la letra K, la tabla dividends que contiene los dividendos trimestrales de esas acciones. La tabla stock_aggregates será usada para juntas estas dos fuentes de datos y contener el precio y dividendo en la fecha en que el dividendo fue pagado.
c) Ejecute el script setup.hive
d) Para verificar que el script funcionó, entre al Shell de hive y ejecute las siguientes consultas:

```
hive> select * from nyse_data limit 20;
hive> select * from dividends limit 20;
```

Se deberían observar precios diarios y los dividendos de esos stocks que empiezan con la letra K.

e) La tabla stock_aggregates debería estar vacía, pero podremos ver su esquema con DESCRIBE para verificar que fue creada satisfactoriamente.

Actividad 2: Junte las fuentes de datos

a) La sentencia del join será ejecutada durante algún tiempo, Use gedit para crear un nuevo archivo de texto en /root/devph/labs/Lab7.4 como join.hive
b) La sentencia de join se dividirá en dos secciones, primero el resultado del join se guardará en la tabla stock:_aggregates, que requiere de un insert:

```
insert overwrite table stock_aggregates
```

El overwrite causa que cualquier dato existente en stock_aggregates sea eliminado

c) Los datos que serán insertados serán el resultado de una consulta que contiene varios indicadores significativos acerca del stock. El resultado contendrá el símbolo, fecha de intercambio, el máximo del mas álto, el mínimo del más bajo, promedio y la suma de los dividentos `select a.symbol, year(a.trade_date), max(a.high), min(a.low), avg(a.close), sum(b.dividend)`
d) A partir de la tabla nyse_data

e) El join será un left outer join de los dividendos de la tabla:
```
left outer join dividends b
```
f) El join será con los campos symbol y trade_date
g) Agrupe el resultado por los campos symbol y trade_date
h) Guarde los cambios al archivo join.hive

Actividad 3: Ejecute la consulta

a) Ejecute la consulta y espere que los Jobs MapReduce terminen de ejecutarse.
b) ¿Cuántos Jobs de MapReduce tuvieron que ejecutarse? _____

Actividad 4: Verifique el resultado

a) En la sesión de Hive ejecute una consulta a stock_aggregates con select *
La salida debería ser..
```
KYO 2004 90.9 66.25 75.79952 0.544
KYO 2005 78.45 62.58 72.042656 0.91999996
KYO 2006 98.01 71.73 85.80327 0.851
KYO 2007 110.01 81.0 93.737686 NULL
KYO 2008 100.78 45.41 79.6098 NULL
KYO 2009 93.2 52.98 77.04389 NULL
KYO 2010 93.83 85.94 90.71 NULL
stock_symbol NULL NULL NULL NULL NULL
```

b) Liste el contenido del directorio stock_aggregates en HDFS. El archivo 000000_0 fue creado como resultado del join:
c) Observe el contenido de la tabla stock_aggregates usando el comando dfs –cat /apps/hive/warehouse/stock_aggregates/000000_0;

Tarea 2: Calcule ngramas de correos electrónicos en formato Avro
Datos:
Ubicación de archivos: /root/devph/labs/7.5

Actividad 1: Observe un esquema Avro

a) Cambie al directorio /root/devph/labs/Lab7.5. Note que este folder contiene un archivo sample.avro.
b) Ejecute el siguiente comando para ver el esquema del contenido de sample.avro
```
# avro cat --print-schema sample.avro
```
c) ¿Cuántos campos tienen los registros en sample.avro?
d) Cree un archivo con esquema para sample.avro
```
# avro cat --print-schema sample.avro > sample.avsc
```
e) Ponga el archivo en HDFS con hadoop fs –put

Actividad 2: Cree una tabla Hive a partir de un esquema Avro

a) Observe el contenido del CREATE TABLE definido en create_sample_table.hive en el directorio de este laboratorio con more.
b) Asegúrese que el archivo de propiedades avro.schema.file apunte al esquema que se creó en el paso anterior.

```
WITH SERDEPROPERTIES (
'avro.schema.url'='hdfs:///user/root/sample.avsc')
```

c) Ejecuta el script create_sample_table.hive

Actividad 3: Verifique la tabla

a) Entre a sesión Shell de Hive
b) Ejecute el comando show tables y verifique que tenga la tabla sample_table
c) Ejecute un Describe sobre sample_table. Note que el esquema para sample_table corresponde con el esquema Avro de sample.avsc
d) Asocie algunos datos con sample_table. Copie sample.avro al directorio de warehouse de Hive ejecutando el siguiente comando (todo en una sola línea)

```
hive> dfs -put /root/devph/labs/Lab7.5/sample.avro
   /apps/hive/warehouse/sample_table;
```

e) Observe el contenido de sample_table, posteriormente salga de sesión con quit. Note que existe solo un registro en sample.avro.

Actividad 4: Crear una tabla de correos electrónicos de usuario.

a) Existe un archivo `/root/devph/labs/Lab7.5/mbox7.avro` que representa correos electrónicos en formato Avro a partir de una lista de correos Hive para el mes de Julio. Use la opción –print-schema de Avro para observar el esquema de este archivo
b) ¿Cuántos campos tienen los registros en mbox7.havro?
c) Ejecute el comando –print-schema de nuevo, pero esta vez la salida del esquema direcciónelo al archivo mbox.avsc `# avro cat --print-schema mbox7.avro > mbox.avsc`
d) Ponga el esquema Avro en HDFS en /user/root
e) Use el comando more para ver el contenido del script create_email_table.hive en el directorio `/root/devph/labs/Lab7.5`. Verifique que la propiedad avro.schema.url este correcta.
f) Ejecute el script para crear la tabla hive_user_mail
g) Copie mbox7.avro al directorio de warehouse
h) Entre en sesión Shell de Hive y verifique que la tabla contenga datos con select *

Actividad 5: Calcule un bigrama

a) Use la función ngrams de Hive para crear un bigrama de las palabras en mbox7.avro

```
hive> select
ngrams(sentences(content),2 ,10)
from hive_user_email;
```

La salida debe ser..

```
[{"ngram":["2013","at"],"estfrequency":802.0},{"ngram":["of","the"],"est
frequ
ency":391.0},{"ngram":["I","am"],"estfrequency":368.0},{"ngram":["I","ha
ve"],
"estfrequency":340.0},{"ngram":["J","E9r"],"estfrequency":306.0},{"ngram
":["f
or","the"],"estfrequency":291.0},{"ngram":["you","are"],"estfrequency":2
89.0}….
```

b) Para dejar el resultado más legible, use la función de Hive explode

```
hive> select
explode(ngrams(sentences(content),2 ,10))
from hive_user_email;
```

c) Ejecute la misma consulta de nuevo, pero esta vez ejecútela con Tez

```
hive> set hive.execution.engine=tez;
```

El tiempo de ejecución debió ser significativamente menor.

d) Tipicamente cuando se trabaja con comparaciones de palabras, se ignoran si son mayúsculas o minúsculas. Ejecute la consulta de nuevo, pero esta vez agregue la función lower y calcule 20 bigramas. La salida debe ser …

```
{"ngram":["2013","at"],"estfrequency":802.0}
{"ngram":["i","have"],"estfrequency":409.0}
{"ngram":["of","the"],"estfrequency":391.0}
{"ngram":["i","am"],"estfrequency":372.0}
{"ngram":["if","you"],"estfrequency":347.0}
{"ngram":["in","hive"],"estfrequency":337.0}
{"ngram":["for","the"],"estfrequency":309.0}
{"ngram":["j","e9r"],"estfrequency":306.0}….
```

Actividad 6: Calcule un ngrama contextual

a) Ejecute la siguiente consulta en hive

```
hive> select
explode(context_ngrams(sentences(lower(content)),
array("error", null) ,20))
from hive_user_email;
```

La salida debe ser…

```
{"ngram":["in"],"estfrequency":102.0}
{"ngram":["return"],"estfrequency":97.0}
{"ngram":["org.apache.hadoop.hive.ql.exec.udfargumenttypeexception"],
"estfreq
uency":49.0}
{"ngram":["failed"],"estfrequency":49.0}…
```

b) ¿Cuál es la palabra que se repite más veces seguida de la palabra error en estos correos?

c) Ejecute una consulta Hive que encuentre los 20 primeros resultados para palabras en mbox7.avro que sigan a la frase “error in”