LAB 03 – Extracción, transformación y carga de datos mediante Interactive Query

En este laboratorio, usará un archivo CSV de datos de vuelo sin procesar, disponible públicamente, lo importará en un almacenamiento de clúster de HDInsight y, después, transformará los datos mediante Interactive Query en Azure HDInsight. Una vez que los datos se han transformado, se cargan en una base de datos de Azure SQL mediante Apache Sqoop.

Se describen las tareas siguientes:

Descarga de datos de vuelos de ejemplo
Carga de datos en un clúster de HDInsight
Transformación de los datos mediante Interactive Query
Creación de una tabla en una base de datos de Azure SQL
Uso de Sqoop para exportar datos a la base de datos de Azure SQL.

Paso 1: Pre-Requisitos

1. Inicie sesión en Azure Portal.
2. En Azure Portal, Provisione un clúster de Interactive Query en HDInsight.
3. Una base de datos de Azure SQL. Use una base de datos de Azure SQL como almacén de datos de destino.
4. Un cliente SSH

Paso 2: Descarga de los datos de vuelo

1. Diríjase a Research and Innovative Technology Administration, Bureau of Transportation Statistics.
2. En la página, desactive todos los campos y, a continuación, seleccione los valores siguientes:

NOMBRE	Valor
Filter Year	2019
Filter Period	January
Fields	Year, FlightDate, Reporting_Airline, DOT_ID_Reporting_Airline,

	Flight_Number_Reporting_Airline, OriginAirportID, Origin, OriginCityName, OriginState, DestAirportID, Dest, DestCityName, DestState, DepDelayMinutes, ArrDelay, ArrDelayMinutes, CarrierDelay, WeatherDelay, NASDelay, SecurityDelay, LateAircraftDelay.

3. Seleccione **Descargar**. Obtenga un archivo .zip con los campos de datos que ha seleccionado.

Paso 3: Carga de datos en un clúster de HDInsight

Hay muchas maneras de cargar datos en el almacenamiento asociado a un clúster de HDInsight. En esta sección, se usa **scp** para cargar datos. Para obtener información acerca de otras formas de cargar los datos

1. Cargue el archivo .zip en el nodo principal del clúster de HDInsight. Modifique el comando siguiente: reemplace FILENAME por el nombre del archivo .zip y CLUSTERNAME por el nombre del clúster de HDInsight. Luego, abra un símbolo del sistema, establezca el directorio de trabajo en la ubicación del archivo y, después, escriba el comando.

scp FILENAME.zip sshuser@CLUSTERNAME-ssh.azurehdinsight.net:FILENAME.zip

Si se le pide que escriba yes o no para continuar, escriba yes en el símbolo del sistema y presione Entrar. El texto no es visible en la ventana mientras lo escribe.

2. Cuando la carga haya finalizado, conéctese al clúster mediante SSH. Modifique el comando siguiente: reemplace CLUSTERNAME por el nombre del clúster de HDInsight. Después, escriba el comando siguiente:

ssh sshuser@CLUSTERNAME-ssh.azurehdinsight.net

3. Una vez establecida la conexión SSH, configure la variable de entorno. Reemplace FILE_NAME, SQL_SERVERNAME, SQL_DATABASE, SQL_USER y SQL_PASWORD por los valores adecuados. Luego, escriba el comando:

```
export FILENAME=FILE_NAME
export SQLSERVERNAME=SQL_SERVERNAME
export DATABASE=SQL_DATABASE
export SQLUSER=SQL_USER
export SQLPASWORD='SQL_PASWORD'
```

4. Descomprima el archivo .zip mediante el comando siguiente:

unzip $FILENAME.zip

5. Cree un directorio en el almacenamiento de HDInsight y, luego, copie en él el archivo .csv mediante el siguiente comando:

hdfs dfs -mkdir -p /tutorials/flightdelays/data

hdfs dfs -put $FILENAME.csv /tutorials/flightdelays/data/

Paso 4: Transformación de datos mediante una consulta de Hive

Hay muchas formas de ejecutar un trabajo de Hive en un clúster de HDInsight. En esta sección se usa **Beeline** para ejecutar un trabajo de Hive.

Como parte del trabajo de Hive importe los datos del archivo .csv en una tabla de Hive denominada **Delays**.

1. En el símbolo del sistema de SSH que ya tiene para el clúster de HDInsight, use el siguiente comando para crear y editar un nuevo archivo denominado **flightdelays.hql**:

nano flightdelays.hql

2. Use el texto siguiente como contenido de este archivo:

```
DROP TABLE delays_raw;
-- Creates an external table over the csv file
CREATE EXTERNAL TABLE delays_raw (
    YEAR string,
    FL_DATE string,
    UNIQUE_CARRIER string,
```

```
    CARRIER string,
    FL_NUM string,
    ORIGIN_AIRPORT_ID string,
    ORIGIN string,
    ORIGIN_CITY_NAME string,
    ORIGIN_CITY_NAME_TEMP string,
    ORIGIN_STATE_ABR string,
    DEST_AIRPORT_ID string,
    DEST string,
    DEST_CITY_NAME string,
    DEST_CITY_NAME_TEMP string,
    DEST_STATE_ABR string,
    DEP_DELAY_NEW float,
    ARR_DELAY_NEW float,
    CARRIER_DELAY float,
    WEATHER_DELAY float,
    NAS_DELAY float,
    SECURITY_DELAY float,
    LATE_AIRCRAFT_DELAY float)
-- The following lines describe the format and location of the file
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
STORED AS TEXTFILE
LOCATION '/tutorials/flightdelays/data';

-- Drop the delays table if it exists
DROP TABLE delays;
-- Create the delays table and populate it with data
-- pulled in from the CSV file (via the external table defined previously)
CREATE TABLE delays AS
SELECT YEAR AS year,
    FL_DATE AS flight_date,
    substring(UNIQUE_CARRIER, 2, length(UNIQUE_CARRIER) -1) AS unique_carrier,
    substring(CARRIER, 2, length(CARRIER) -1) AS carrier,
    substring(FL_NUM, 2, length(FL_NUM) -1) AS flight_num,
    ORIGIN_AIRPORT_ID AS origin_airport_id,
    substring(ORIGIN, 2, length(ORIGIN) -1) AS origin_airport_code,
    substring(ORIGIN_CITY_NAME, 2) AS origin_city_name,
    substring(ORIGIN_STATE_ABR, 2, length(ORIGIN_STATE_ABR) -1)  AS origin_state_abr,
    DEST_AIRPORT_ID AS dest_airport_id,
```

```
    substring(DEST, 2, length(DEST) -1) AS dest_airport_code,
    substring(DEST_CITY_NAME,2) AS dest_city_name,
    substring(DEST_STATE_ABR, 2, length(DEST_STATE_ABR) -1) AS dest_state_abr,
    DEP_DELAY_NEW AS dep_delay_new,
    ARR_DELAY_NEW AS arr_delay_new,
    CARRIER_DELAY AS carrier_delay,
    WEATHER_DELAY AS weather_delay,
    NAS_DELAY AS nas_delay,
    SECURITY_DELAY AS security_delay,
    LATE_AIRCRAFT_DELAY AS late_aircraft_delay
FROM delays_raw;
```

3. Para guardar el archivo, presione **Ctrl+x**, luego **y** y después Entrar.
4. Para iniciar Hive y ejecutar el archivo **flightdelays.hql**, use el siguiente comando:

```
beeline -u 'jdbc:hive2://localhost:10001/;transportMode=http' -f flightdelays.hql
```

5. Cuando finalice el script **flightdelays.hql**, use el siguiente comando para abrir una sesión interactiva de Beeline:

```
beeline -u 'jdbc:hive2://localhost:10001/;transportMode=http'
```

6. Cuando aparezca jdbc:hive2://localhost:10001/> en el símbolo del sistema, utilice la consulta siguiente para recuperar los datos importados sobre los retrasos de vuelos:

```
INSERT OVERWRITE DIRECTORY '/tutorials/flightdelays/output'
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
SELECT regexp_replace(origin_city_name, "", "),
    avg(weather_delay)
FROM delays
WHERE weather_delay IS NOT NULL
GROUP BY origin_city_name;
```

Esta consulta recupera una lista de ciudades que experimentaron demoras por inclemencias del tiempo, junto con el tiempo medio de retraso, y la guarda en /tutorials/flightdelays/output. Más adelante, Sqoop

leerá los datos desde esta ubicación y los exportará a Azure SQL Database.

7. Para salir de Beeline, escriba **!quit** en el símbolo del sistema.

Paso 5: Creación de una tabla de SQL Database

Hay muchas maneras de conectarse a SQL Database y crear una tabla. En los siguientes pasos se utiliza **FreeTDS** desde el clúster de HDInsight.

1. Para instalar FreeTDS, use el siguiente comando desde una conexión SSH abierta al clúster:

```
sudo apt-get --assume-yes install freetds-dev freetds-bin
```

2. Cuando finalice la instalación, use el comando siguiente para conectarse al servidor de SQL Database.

```
TDSVER=8.0 tsql -H $SQLSERVERNAME.database.windows.net -U $SQLUSER -p 1433 -D $DATABASE -P $SQLPASWORD
```

Recibirá una salida similar al texto siguiente:

```
locale is "en_US.UTF-8"

locale charset is "UTF-8"

using default charset "UTF-8"

Default database being set to <yourdatabase>

1>
```

3. En el símbolo del sistema **1>** , introduzca las líneas siguientes:

```
CREATE TABLE [dbo].[delays](

[origin_city_name] [nvarchar](50) NOT NULL,
```

```
[weather_delay] float,

CONSTRAINT [PK_delays] PRIMARY KEY CLUSTERED

([origin_city_name] ASC))

GO
```

Cuando se haya especificado la instrucción **GO**, se evaluarán las instrucciones anteriores. Esta instrucción crea una tabla denominada **delays** con un índice agrupado.

Use la siguiente consulta para comprobar que se ha creado la tabla:

```
SELECT * FROM information_schema.tables

GO
```

La salida será similar al siguiente texto:

```
TABLE_CATALOG   TABLE_SCHEMA    TABLE_NAME      TABLE_TYPE

databaseName        dbo             delays          BASE TABLE
```

4. Entrar **exit** at the **1>** .

Paso 6: Exportación de datos a una base de datos SQL mediante Apache Sqoop

En las secciones anteriores, copió los datos transformados en **/tutorials/flightdelays/output**. En esta sección, va a usar Sqoop para exportar los datos de **/tutorials/flightdelays/output** a la tabla que creó en la base de datos de Azure SQL.

1. Compruebe que Sqoop puede ver la base de datos SQL con el siguiente comando:

```
sqoop list-databases --connect
jdbc:sqlserver://$SQLSERVERNAME.database.windows.net:1433 --username $SQLUSER
--password $SQLPASWORD
```

Este comando devuelve una lista de bases de datos, incluida la base de datos en la que creó anteriormente la tabla **delays**.

2. Exporte los datos de **/tutorials/flightdelays/output** a la tabla **delays** con el siguiente comando:

```
sqoop export --connect
"jdbc:sqlserver://$SQLSERVERNAME.database.windows.net:1433;database=$DATABAS
E" --username $SQLUSER --password $SQLPASWORD --table 'delays' --export-dir
'/tutorials/flightdelays/output' --fields-terminated-by '\t' -m 1
```

Sqoop se conecta a la base de datos que contiene la tabla **delays** y exporta los datos del directorio **/tutorials/flightdelays/output** a dicha tabla.

3. Cuando finalice el comando sqoop, use la utilidad tsql para conectarse a la base de datos mediante el siguiente comando

```
TDSVER=8.0 tsql -H $SQLSERVERNAME.database.windows.net -U $SQLUSER -p 1433 -D
$DATABASE -P $SQLPASWORD
```

Use las instrucciones siguientes para comprobar que los datos se exportaron a la tabla de retrasos:

```
SELECT * FROM delays
GO
```

Debería ver una lista de los datos de la tabla. La tabla incluye el nombre de la ciudad y el tiempo medio de retraso de los vuelos promedio a la ciudad.

Escriba **exit** para salir de la utilidad de tsql.