Processing Big Data with Azure Data Lake

Lab 5 – Final Challenge

Overview

This final lab is designed to test your understanding of the concepts and techniques discussed in the course by presenting you with some real challenges to solve for yourself. You will be given some source data and some high-level instructions, but it's up to you to figure out how to meet the required goals.

What You'll Need

To complete this lab, you will need the following:

- A web browser
- A Microsoft account
- A Microsoft Azure subscription
- A Windows, Linux, or Mac OS X computer
- The lab files for this course

Note: To set up the required environment for the lab, follow the instructions in the Setup document for this course. Then if you have not done so already, provision an Azure Data Lake Analytics resource in your Azure subscription.

Challenge 1: Explore Stock Items Data

1. Examine **stock.txt** in the **retail** folder where you extracted the lab files for this course. This is a tab-delimited text file containing a list of stock items sold by a retailer. Each item has a numeric identifier and a name.
2. Upload **stock.txt** to your Azure Data Lake store.
3. Create a database named **retail** in your Azure Data Lake Analytics instance, and then in the database create a schema named **sales** containing a table named **stock** in the **retail** database. The **stock** table should contain the following columns, and an appropriate index:
   - **id** (int)
   - **item** (string)
4. Run a U-SQL job to load the data in **stock.txt** into the **sales.stock** table in the **retail** database.
5. Run a U-SQL job to query the **sales.stock** table, and generate a comma-delimited text file that contains the **id** and **item** values of products where **item** contains the text "chocolate" (in any case-format – for example, the results should include "Chocolate", "CHOCOLATE", and "chocolate").

Hint: Review the documentation for the **IndexOf**, **ToLower**, and **ToUpper** methods of the C# **String** class at https://msdn.microsoft.com/en-us/library/system.string_methods(v=vs.110).aspx. These may be useful when filtering the data.

6. When the job has finished, note the number of rows that were output to the results file.

Challenge 2: Explore Orders Data

1. Examine the **orders-XXXX.txt** files in the **retail\orders** folder where you extracted the lab files for this course. These tab-delimited text files contain details of orders for the years 2013 to 2016. Each row represents an order line item, and includes a **StockItemID** field that references the id field in the **stock.txt** file you explored previously. A single order is identified by an **OrderID** field, so for orders that include multiple items, the same **OrderID** is referenced in multiple rows of the orders data file.
2. Upload the **orders-XXXX.txt** files to a folder in your Azure Data Lake Store.
3. Create a table named **orders** in the **sales** schema of the **retail** database, using the following schema with an index on the **orderid** field, and load the data from the orders-XXXX.txt files into the table:
    - **orderid** (int)
    - **orderdate** (string)
    - **customername** (string)
    - **phonenumber** (string)
    - **deliveryaddressline1** (string)
    - **deliveryaddressline2** (string)
    - **cityname** (string)
    - **stockitemid** (int)
    - **quantity** (int?)
    - **unitprice** (decimal?)

    Hint: The orders files are tab-delimited, and the first row contains the column names. Use the appropriate built-in extractor with a suitable value for the **skipFirstNRows** parameter to extract the data.

4. Write a query to return the year and the number of orders placed for each year. Output the results to a comma-delimited file in ascending order of year.

    Hint: The **orderdate** field is a string in the format YYYY-MM-DD, so you need to extract the first four characters of this to determine the year – take a look at the **Substring** methods of the C# **String** class at https://msdn.microsoft.com/en-us/library/system.string_methods(v=vs.110).aspx.

    The challenge is to count the number of orders, not the number of order lines – so your query will need to count the number of DISTINCT **orderid** values for each year.

Challenge 3: Query Stock and Orders

1. Write a query to return the **item** name from the **stock** table and the sum of **quantity** from the corresponding orders in the **orders** table for the items that contain the word “chocolate” as identified in the previous exercise. Output the results to a comma-delimited file with the rows in

descending order of total quantity sold, and then view the output file to identify the chocolate-based item for which most units have been sold.

2. Write a query to return the **customername** name and the sum of **quantity** from the **orders** table for the items that contain the word “chocolate” as identified in the previous exercise. Output the results to a comma-delimited file with the rows in descending order of total quantity sold, and then view the output file to identify the customer who has purchased the most chocolate-based items.

Note: You have now completed the labs in this course. Unless you want to experiment further, you can delete the resource group containing your Azure Data Lake Store and Azure Data Lake Analytics resources.