# Challenge 02 – Ingest and Prepare Data

## Summary

In this challenge, you will use your Azure Databricks cluster to ingest the raw data, perform data engineering steps, and prepare the data for use with the Power BI and SQL DW you will build in a later challenge.

You will start with source data files, take steps to conform all source data to a consistent structure (schema), enrich the conformed source data, then merge it all together into one data set for later consumption.

To perform this challenge, some experience working in Spark and Jupyter notebooks will be helpful. If needed, Azure Databricks documentation can be found here: <https://docs.azuredatabricks.net/>

## Technologies

You will be working on the Databricks premium cluster you created earlier. You will need to access data files in Azure blob storage, write data engineering code, and export a final dataset for further use.

## Task 1 – Copy source data files to your Azure storage location

You will use the public NYC Taxi data set. We have provided the data in an Azure storage account. You can connect to it in Azure Storage Explorer using the following connection string:

BlobEndpoint=https://dtafy19datahack.blob.core.windows.net/;SharedAccessSignature=sv=2017-11-09&ss=bfqt&srt=sco&sp=rwdlacup&se=2018-09-22T04:13:25Z&st=2018-09-21T20:13:25Z&spr=https&sig=joExg0fzhgqZnW51qX3uGzsYXc%2FnSFMVJm1Wk%2BWSER8%3D

We have provided three folders:

1. reference-data
2. transactional-data-small
3. transactional-data-big

You will need folder 1 (reference data).

Please choose ***either*** folder 2 (transactional data small) or folder 3 (transactional-data-big). ***Do not use both.*** What’s the difference?

* transactional-data-big is the full NYC Taxi data set, in original format.
* transactional-data-small is a sample of the full data set. We randomly sampled from the full-size data set to accomplish two goals:
  + provide a much smaller (~10-15%) data set so that shorter hack events are still realistic; the full-size data set is so large that working with it in Spark can take hours
  + equalize the relative sizes of Yellow and Green; in the source, Yellow is far bigger than Green, so we sampled more heavily from Green than Yellow

Use the big data set when you have time. Use the small data set when you’re at a quick event, or to explore and try quickly.

Copy the appropriate folders into the storage account you created for this event.

## Task 2 – Connect Databricks cluster to your Azure storage location

Connect to the Azure storage location where you uploaded the source data files and perform a directory listing.

This task is complete when: in a Databricks notebook, you can list files in storage and see the source data files you uploaded.

## Task 3 – Create a Hive Database

You will need a Hive database on the Databricks cluster. Create one and note its name. you can use language of your choice.

This task is complete when you can list all the tables in the database (at this point, there should not be any tables yet).

## Task 4 – Create Hive Tables for Reference Data

At the root of the source data, there is a reference data folder.

Create a Hive table on top of each of the 6 reference data files.

This task is complete when you can run typical SQL queries against each of the Hive tables, and see the data that is in the source files.

## Task 5 – Create Hive Tables for Transaction Data

At the root of the source data, there is a transaction data folder. It contains taxi ride transaction data for both Yellow and Green taxis. 2010 to 2013 contains only Yellow taxi data and 2014 to 2018 contains both yellow and green taxi data.

Your task is to create one Hive table for each taxi type (i.e. yellow and green) and to load all transaction data into the appropriate table.

Be careful! The source data spans multiple years. Even within one taxi company (e.g. Yellow)… could there have been changes to the source files over the years? Are there outliers in the data? Check your assumptions with some exploratory data analysis.

This task is complete when:

* You have two new Hive tables: one for Yellow, and one for Green taxi transaction data
* All the transactional data is accessible through these tables with appropriate SQL statements
  + In particular, you should be able to query either table’s data with a predicate on the taxi trip year or month and prove that data from the entire range of source files is queryable

## Task 6 – Conform and Merge Data Sets

Now you need to provide a merged view of the two transactional data sets (Yellow and Green) you’ve been working with so far, so that there is one main transactional data set.

Write code to merge the two data sets from task 6 into one merged Hive table.

This task is complete when you can query this single new table and see data from both Yellow and Green, correctly identified as such.

## Bonus Task – Connect your Databricks Notebook(s) to github

Databricks notebooks can be connected to a github repo so that notebook history can be tracked as a series of commits there. As a bonus challenge, connect your notebook(s) to a github repo and determine how you can

This task is complete when you can navigate to [https://github.com/[YourAccount]/[YourRepo](https://github.com/%5bYourAccount%5d/%5bYourRepo)] and see your notebook(s) checked in there, including a history of committed changes.

## Notes and Resources

To successfully complete this challenge, the following resources may be helpful.

Spark API documentation (select the language you’re using): <https://spark.apache.org/docs/latest/api/>. In particular, the dataframe and dataset API docs will come in very handy…

Databricks documentation: <https://docs.azuredatabricks.net/>