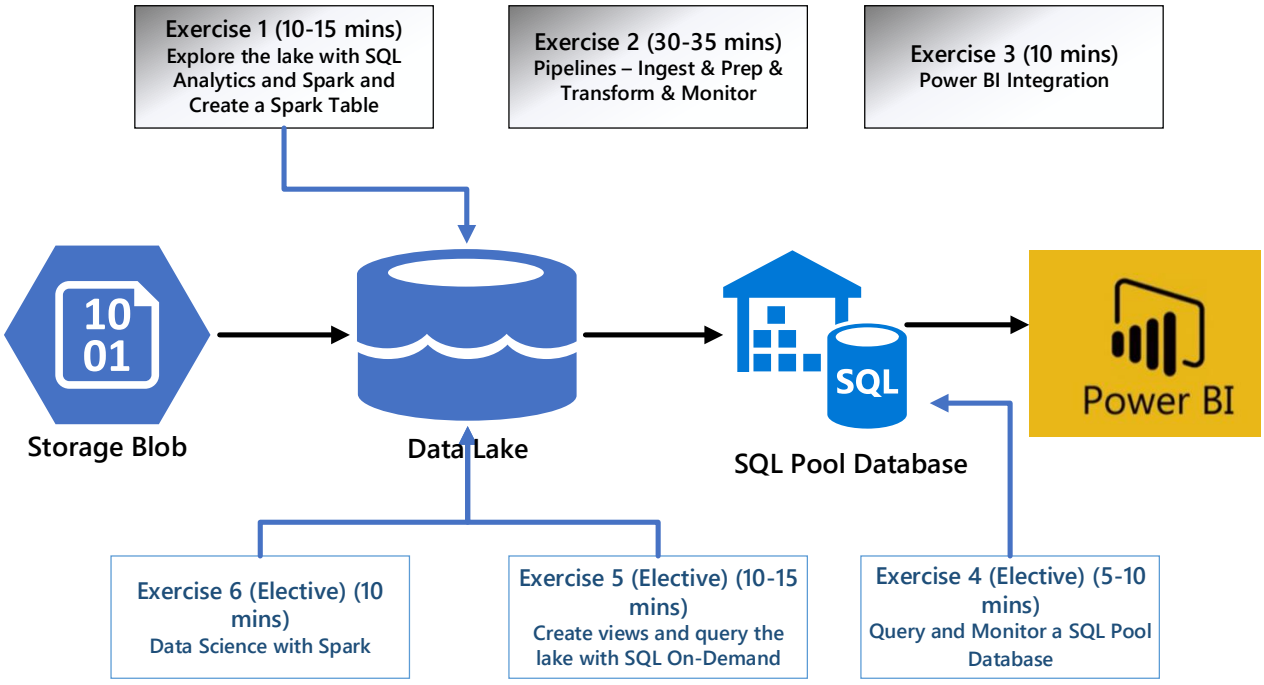
Azure Synapse Analytics Lab

Abstract and learning objectives

- "Driving Company" provides payment processing services for businesses. It is designing and implementing a Proof of Concept (PoC) for a unified data analytics platform. Their soft goals is to bring siloed teams to work together on a single platform.
- In this lab, you will play the role of various persona: a data engineer, a business analyst and a data scientist. The workspace is already setup so you can focus on some of the core development capabilities of Azure Synapse Analytics.
- By the end of this workshop, you will have performed a non-exhaustive list of operations that combine the strength of Big Data and SQL analytics into a single platform.
- To show the breadth and flexibility of tasks you can accomplish on Arcadia, we have two sections to achieve: Core + Elective. We do expect you to finish the core but hope you can accomplish at least one elective of your choice.

Solution architecture

You will probably not be able to finish all exercises, so we make exercise 3, 5 and 6 electives.



In exercise 1, you will explore the data lake with the data explorer and its tight integration to Spark and SQL for retrieving information about the data. You will learn how to create a Spark Table and have it accessible through other engines. This is what we call engine interoperability!

In exercise 2, you will build a pipeline with 4 nested pipelines underneath. Each pipeline will serve its own purpose:

- ✓ Copy data from a blob storage account to the Data Lake
- ✓ Prepare & Transform the Green Cab data using Data Flow (a code-free big data transformation tool), copy the data into a SQL Pool staging table and transform that data into a destination table for querying (through stored procedure)
- ✓ Prepare & Transform the Yellow Cab data using a Scala Notebook with AAD passthrough between Spark and ADLSg2, seamless integration between Spark and SQL to copy the processed data into SQL.
- ✓ Simple copy activity from the lake to the SQL Pool triggering a stored procedure transformation in the SQL Pool.

By the end of exercise 2, you will have built and run a modern data warehouse pipeline that will serve future scenarios.

What to consider if you cannot move forward:

- ✓ **You can run a backup pipeline, called "EXE2 LabIgnitePipelineBackup"**
- ✓ **In case we experience technical issues to load the data into the SQL Pool, you can finish the service with the pipeline called "SQLStagingToDBObackup"**

In exercise 3, you will see how easy it is to integrate a Power BI workspace in Azure Synapse Analytics.

In exercise 4, we run a query that will aggregate 3 tables (one with 600m rows) and join them based on the dates. You will see how performant the query is and how to monitor SQL requests through DMVs.

In exercise 5, we are going back to SQL On-Demand, a serverless engine to query the lake with the full T-SQL support! You will learn how to create views over files and folders with SQL and query that views.

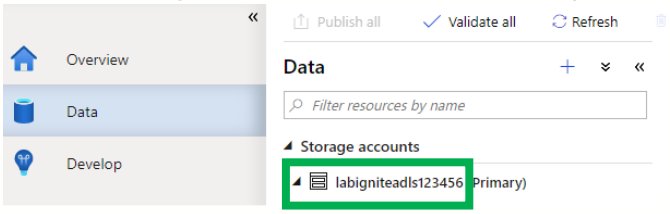
In exercise 6, you will be able to run and understand a sophisticated analysis that a Data Scientist could make with Spark in Synapse.

Exercises

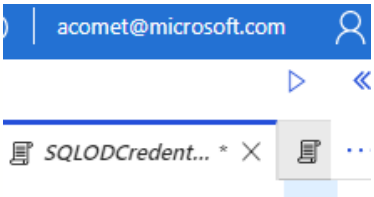
In the exercises and artifacts uploaded, you will have to replace those generic names with the names specific to your workspace.

DO NOT CLOSE THE WEB WINDOW WITH THAT INFORMATION

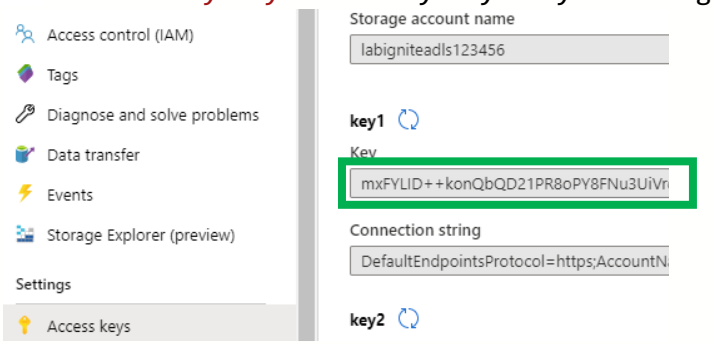
- ✓ `<ADLSg2 Account Name>`: name of your ADLSg2 account



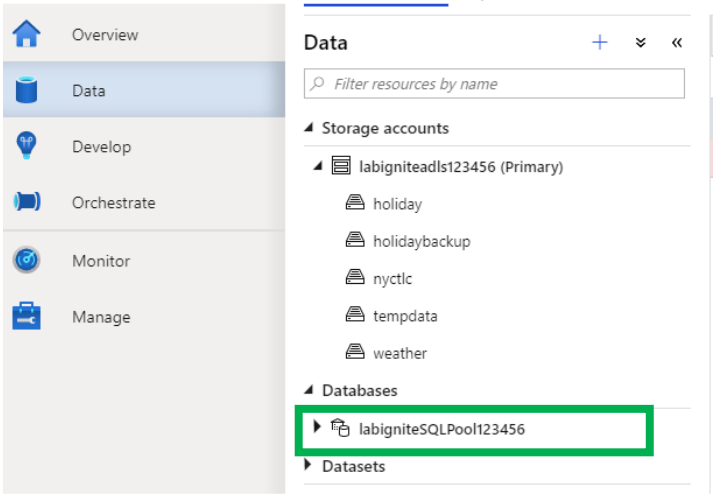
- ✓ `<AAD User ID>`: name of your ADLSg2 account



✓ <Primary Key>: Primary key of your ADLSg2 account



✓ <SQL Pool Name>: Name of your SQL Pool



Exercise 1 – Explore the lake with SQL On-Demand and Spark

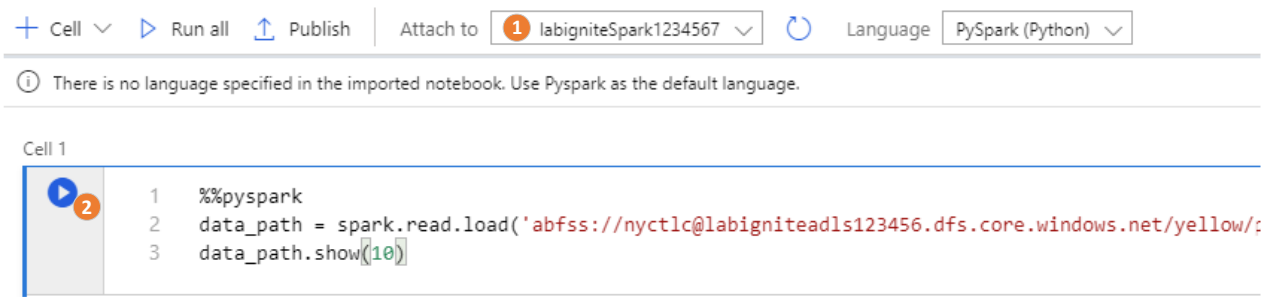
This section will highlight how you can explore data using the engine of your choice.

Task 1 – Explore the lake with SQL On-Demand

In this task, you will browse your data lake. Go into the YellowCab folder, select the year and month folders of your choice, then select a file, right click and select “New SQL script”. A script is automatically generated. Run this script to see how SQL on demand queries the file and returns the first 100 rows of that file with the header allowing you to easily explore data in the file.

Task 2 – Explore the lake and create a table with Spark

Similarly, go to the folder holidaybackup/processed. Select the file part-00000-tid-5126373485025311044-8121f1d6-4c9d-4e4c-8e78-fbe2733fc3a5-649-c000.snappy.parquet and “New Notebook”. This will generate a notebook with PySpark code to load the data in a dataframe and display 10 rows with the header. Attach the notebook to a Spark pool and run the command.



Bonus Challenge: Now, take time to explore on your own the functionalities of the data explorer, the SQL Script and Notebook.

- ✓ Try to find the export/import features of the SQL Script and Notebooks
- ✓ Try to add **text cells** (no code) into a notebook
- ✓ Try to find which languages are supported on a notebook

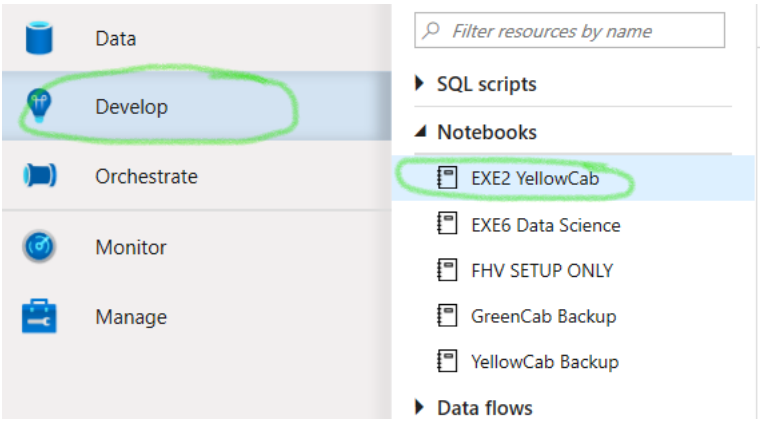
Exercise 2 – Build Modern Data Warehouse pipelines

This section is very important to create a pipeline with parallel activities to bring data into the lake, transform it and load it into the SQL Pool.

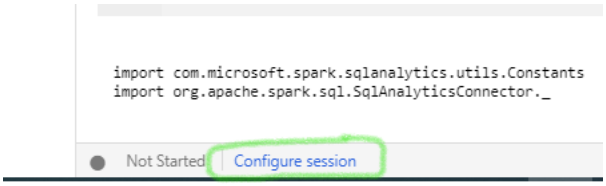
Task 1 – Create and run a Notebook for YellowCab

In this notebook you will see the power of the AAD passthrough between compute and storage whether it's a data lake or a database. You will see how simple it is to write into a SQL Pool table with Spark thanks to the connector. No need to create password, identity, external table, format sources. It's all managed by the connector!

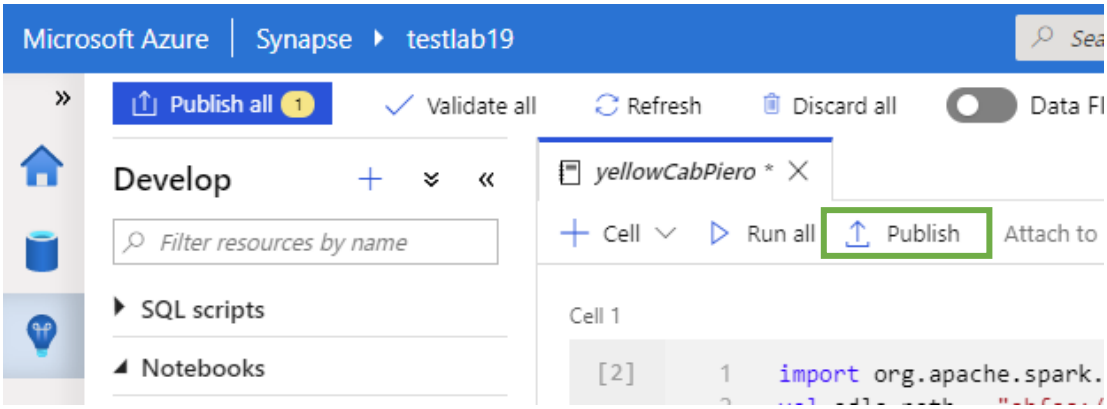
1. Go into the Develop section
2. Select the notebook section and click on **EXE2 YellowCab**



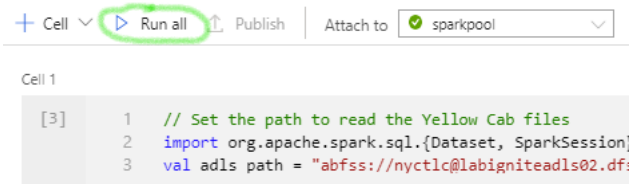
3. Configure and author your notebook:
    a. Attach your Spark Compute
    b. Select Spark as a language
    c. Define the configuration of the session. Defining the configuration of a session enables you to increase the resources of running a notebook. Use 4 executors of medium size for that notebook. You want to run it fast!



4. Click on "*Publish*"



5. Click on *Run All*



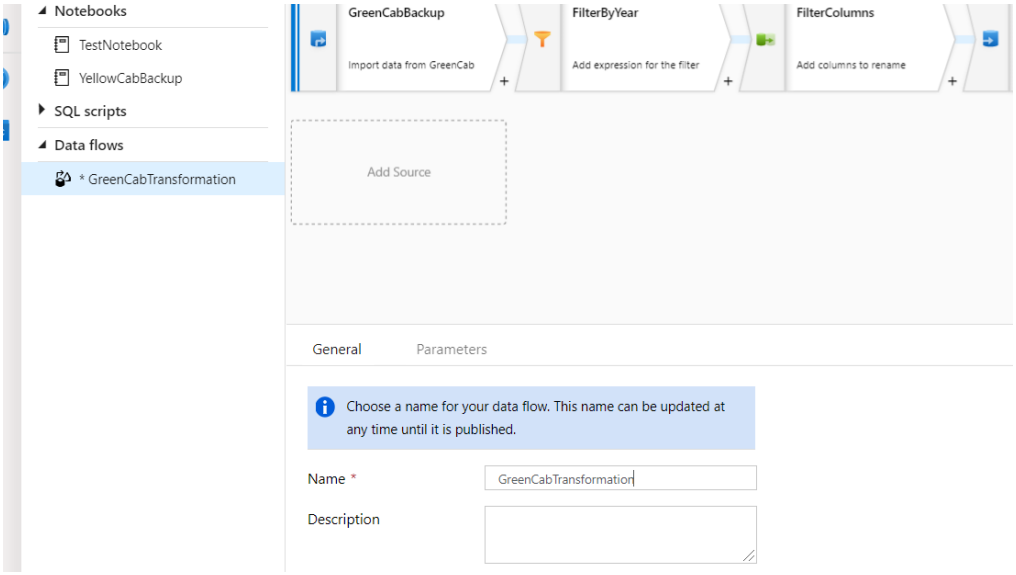
6. Click on *Run All*

Bonus Challenge: Open and check the notebook called **EXE2 Bonus Notebook with C#**. See how simple C# for Spark is!!

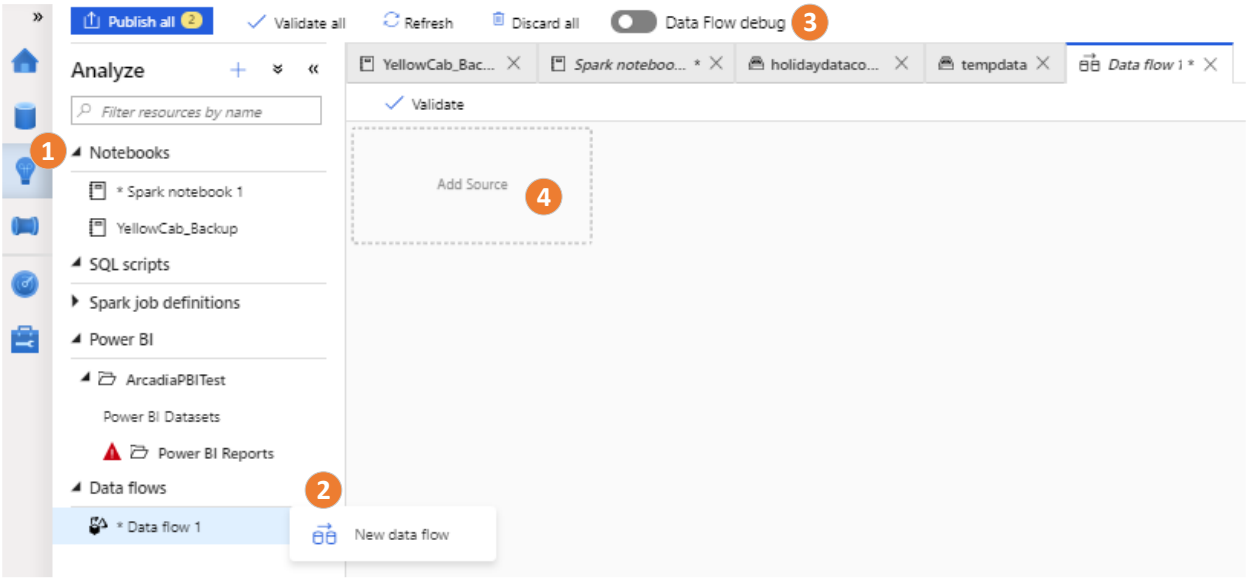
Task 2 – Create and run a Dataflow with GreenCab dataset

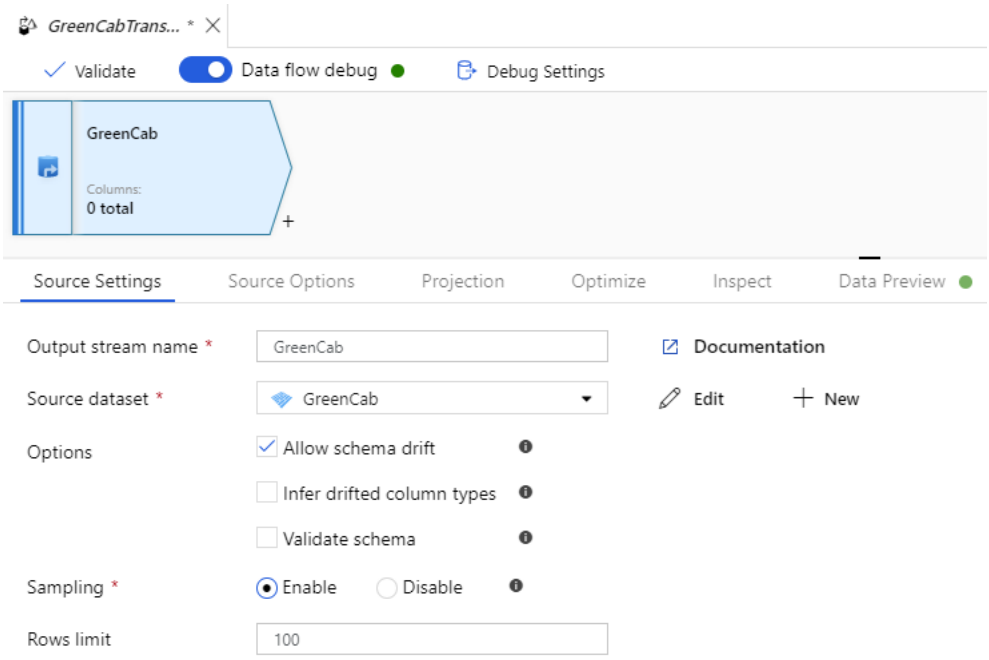
In this task, we will be doing a similar as in task 1 but in a code-free environment on the green cab data. You will also learn the important concept of dataset by creating a new dataset.

1. Click on Develop
2. In Data Flow, create a "*New data flow*"
3. Name the Data Flow activity **GreenCabTransformation**

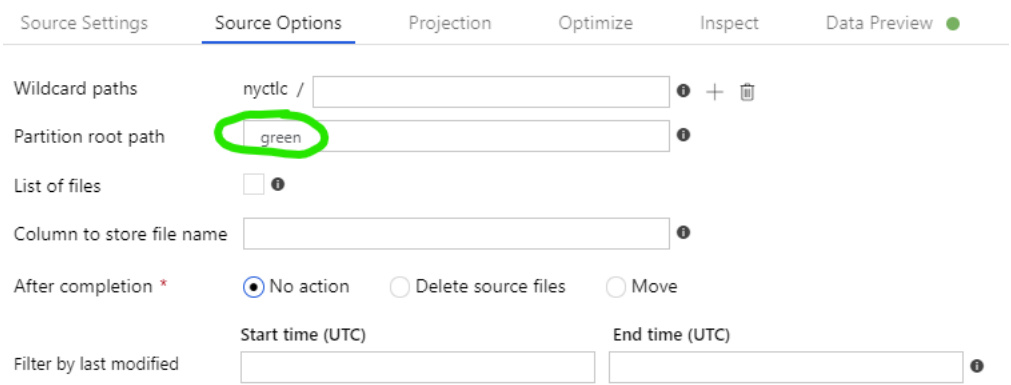


4. Click on *Data Flow Debug*, select *"AutoResolveIntegrationRuntime"and click "Ok"*
5. Click on Add Source

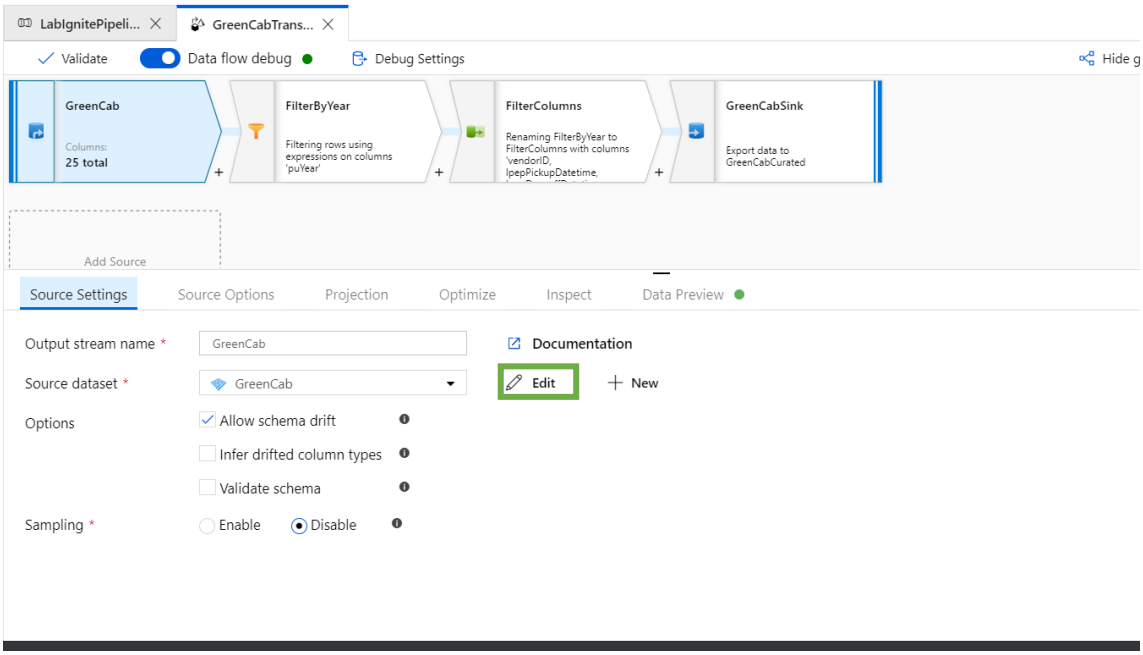




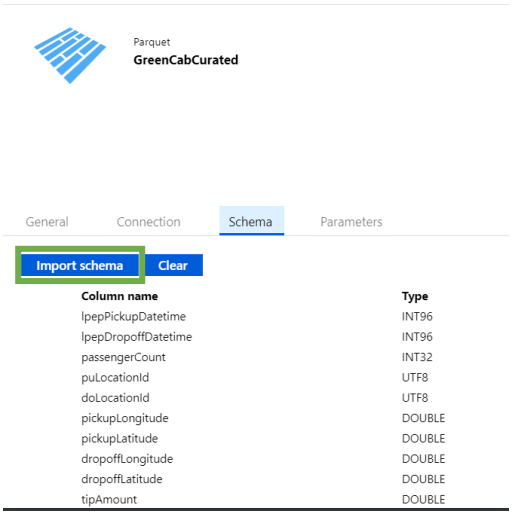
6. Select the dataset **GreenCab**
7. Call the output stream name **GreenCab**
8. Enable sampling
9. In the Source Options:
    a. Partition root path: **green**



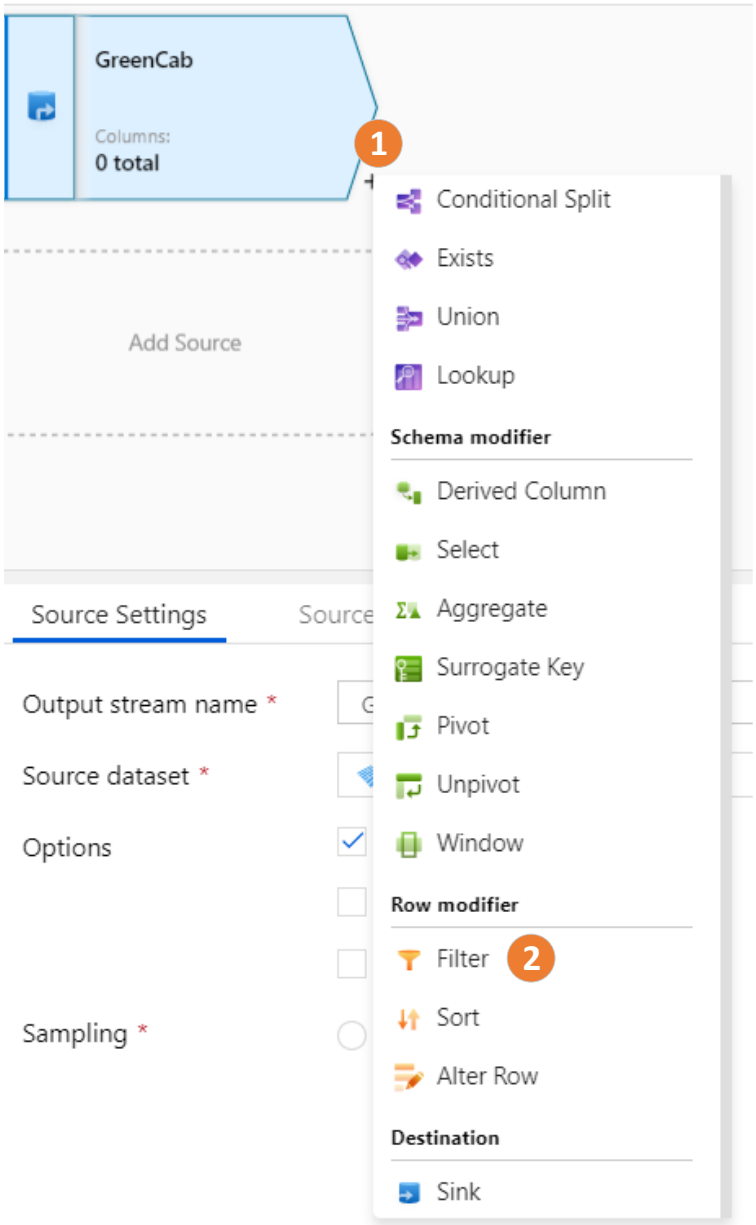
10. On Source Settings *Edit* the **GreenCab** Dataset



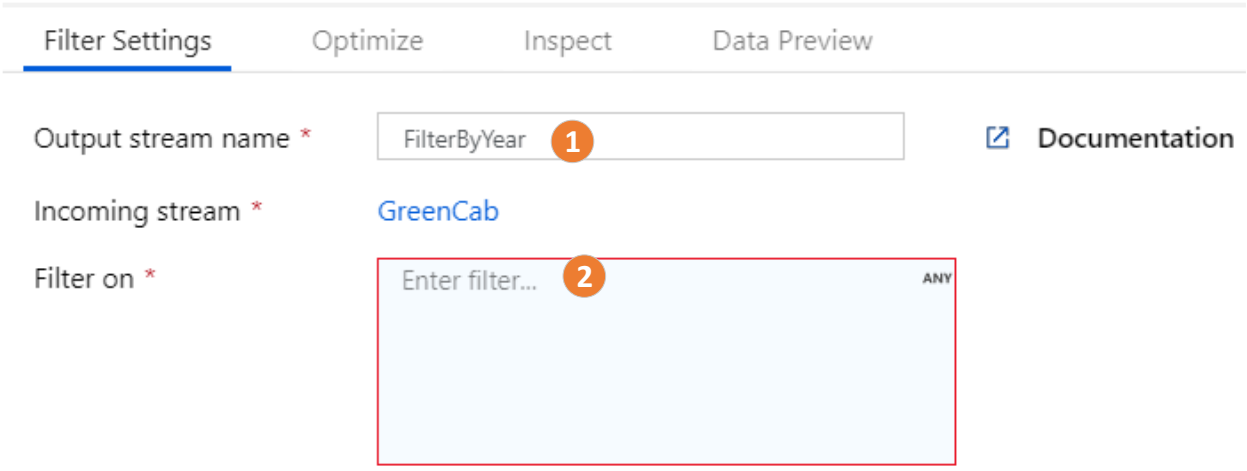
11. Import schema **From connection/store**



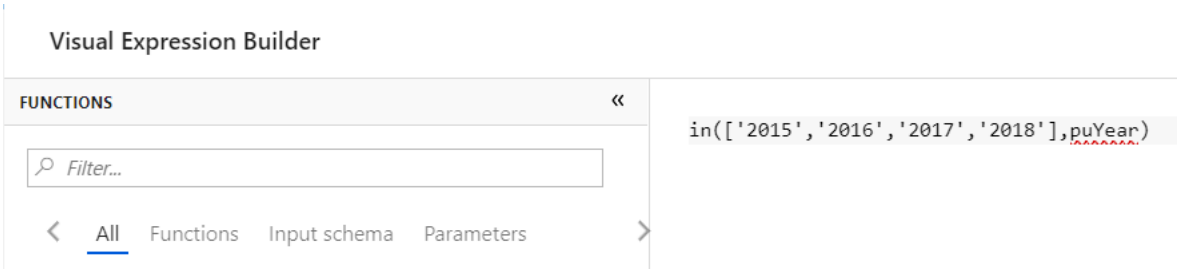
12. Click on **+** next to the GreenCab activity



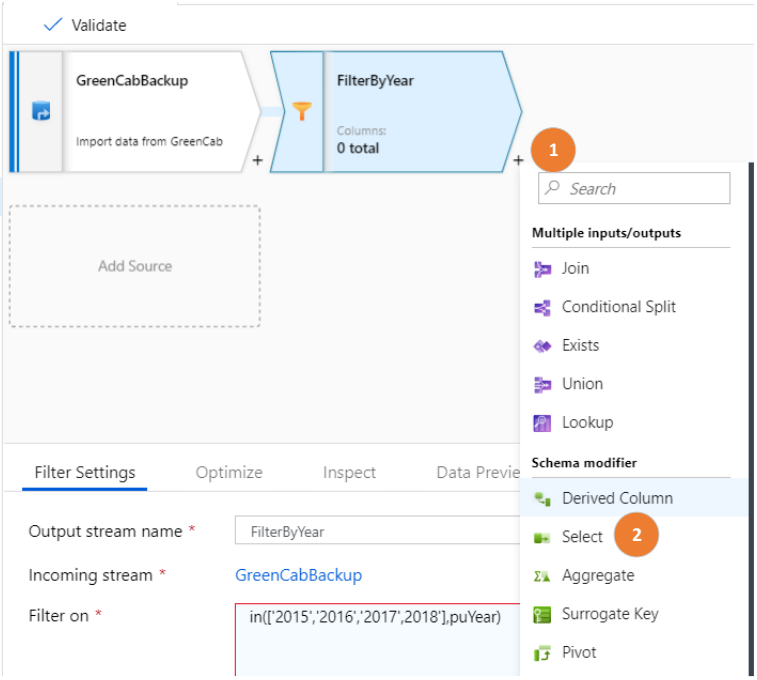
13. Select the **row modifier** filter
14. Enter the following for **Output stream name**: **FilterByYear**
15. Click on **Filter on**



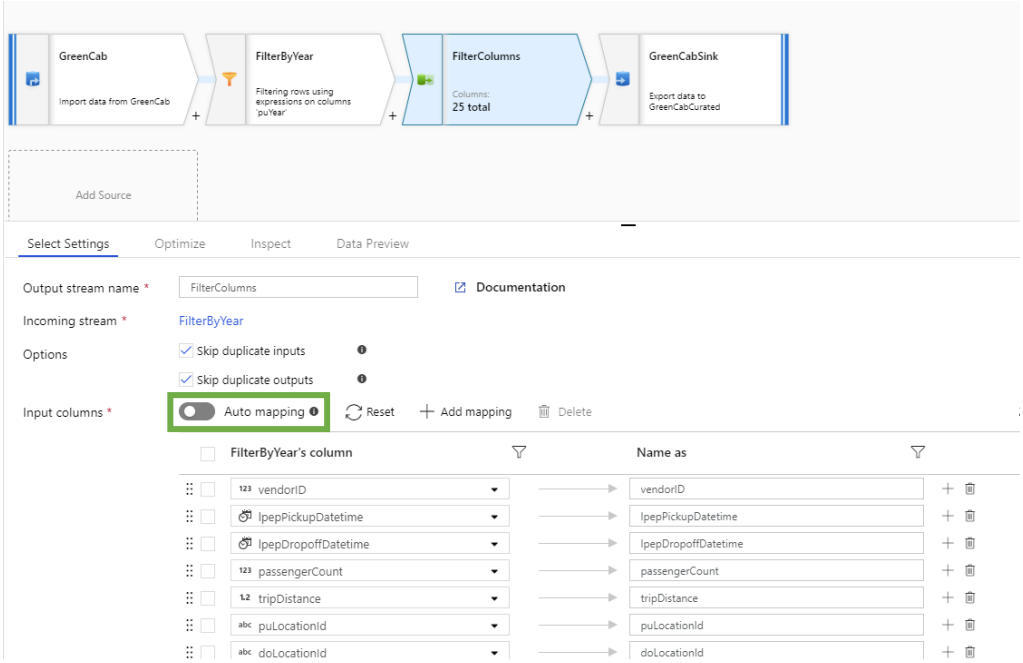
16. Write the following in the box: in(['2015', '2016', '2017', '2018'],puYear)



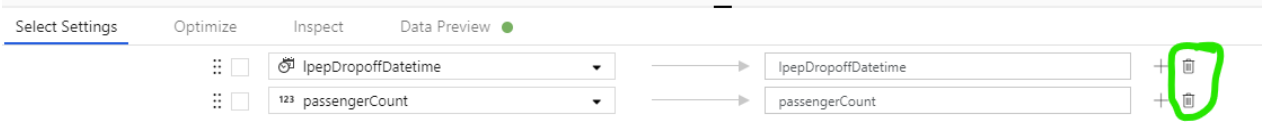
17. Click on **Save and Finish**
18. Click on + next to the Filter activity
19. Select the **Select** Activity



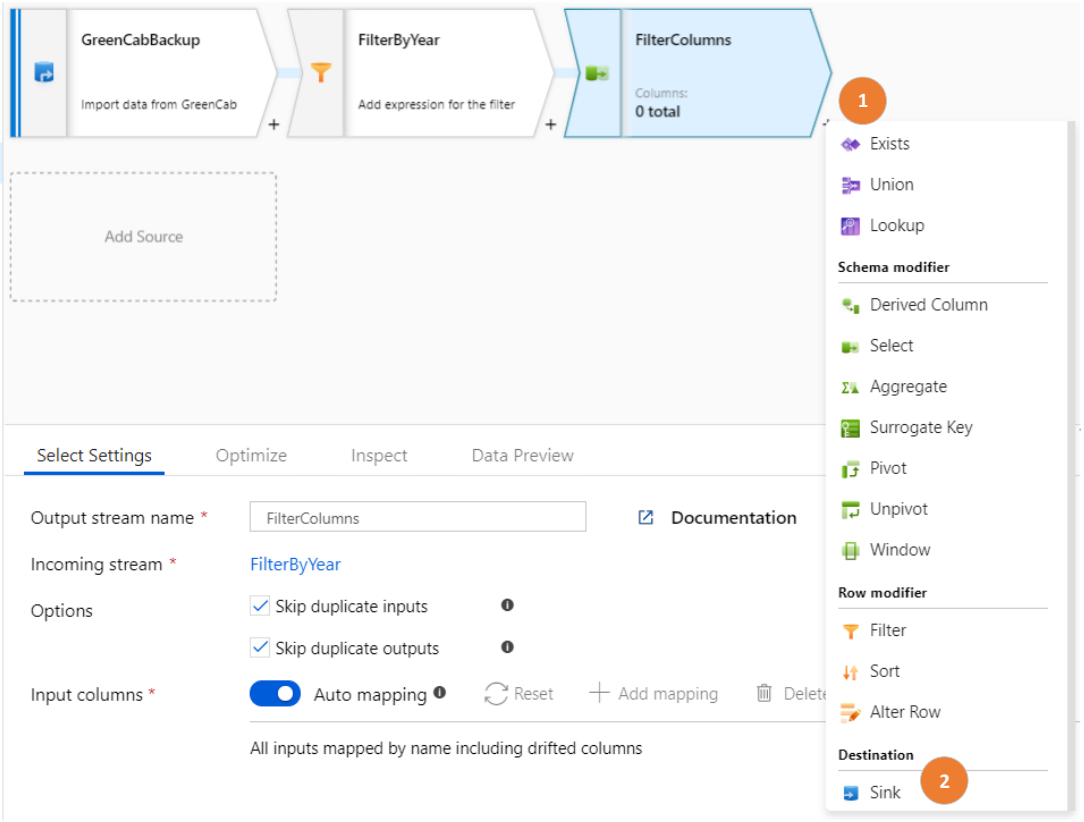
20. Enter the following for **Output stream name**: **FilterColumns**
21. Click on **Filter on**
22. **Select Automapping**



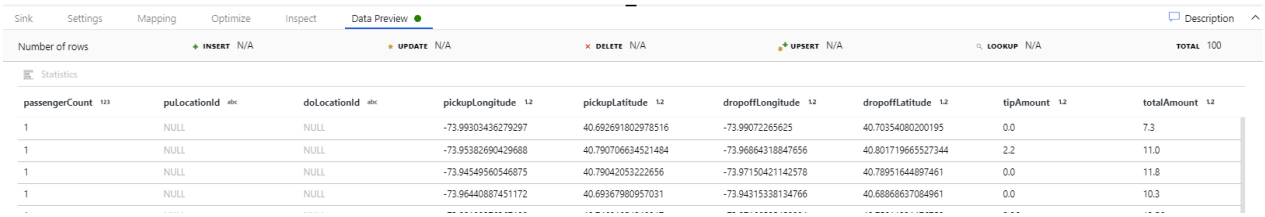
23. Remove the columns: vendorID, lpepPickupDateTime, lpepDropoffDateTime, tripDistance, rateCodeId, storeAndFwdFlag, paymentType, fareAmount, extra, mtaTax, improvementSurcharge, tollsAmount, ehailFee, tripType.



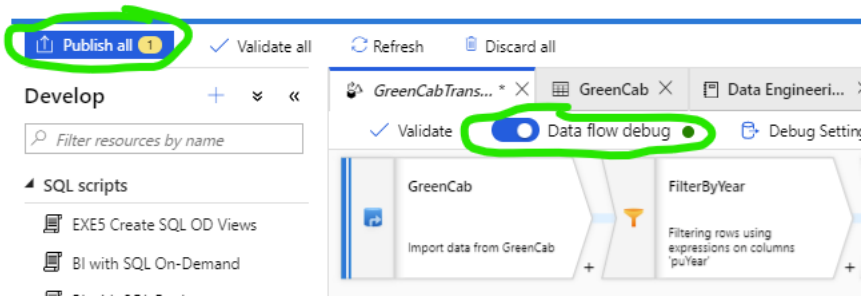
24. Click on + next to the Filter activity
25. Select the **Sink** destination



26. Call the *output stream name* **GreenCabSink**
27. Select *+New* as dataset, Select *+ADLSg2* as a storage layer
28. Select *Parquet*
29. Name it as **DataflowSink**
30. Select the linked service **CoreDataLakeStorageBackup**
31. For the File system in the file path, write **tempdata**
32. Click on **Finish**
33. If you preview the data of GreenCabSink, you should see the following:

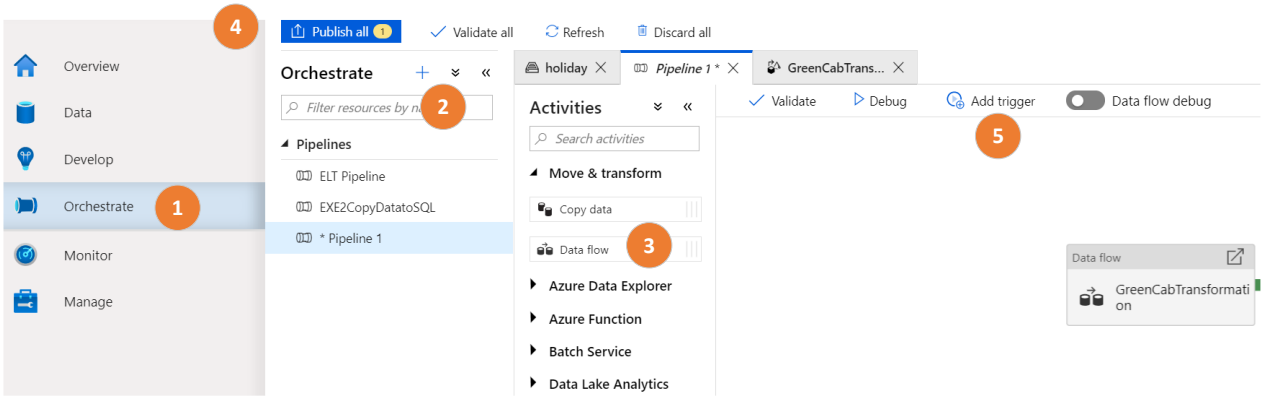


34. Turn of the **Debug Session** and click on **Publish all**



Bonus Challenge: Try to run the data flow in a pipeline by following the flow below (you will monitor that pipeline later on in task 5)

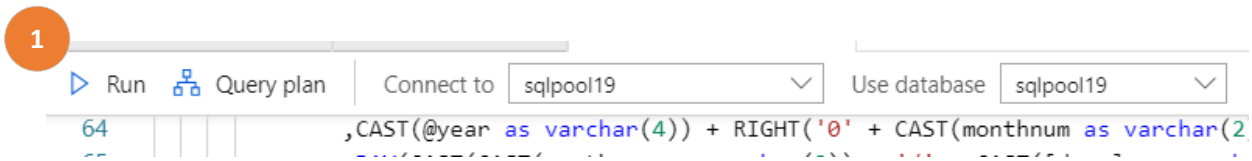
1. Click on *Orchestrate*
2. Add a new pipeline by clicking *"+"*
3. Drag and drop a data flow activity in the pipeline. Make sure that you select **GreenCabTransformation**.
4. Publish the pipeline
5. Click on Add and trigger and select "*Trigger Now*" **(Do not wait for the result to finish (it will take 7 minutes – you will see the result in task 5))**



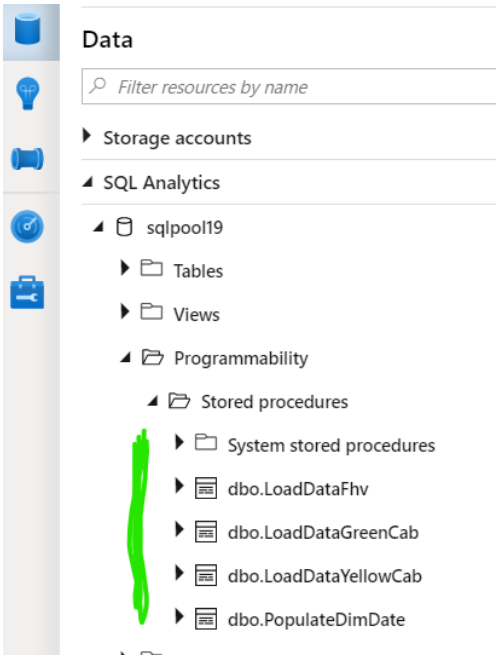
Task 3 – Create Stored Procedures

In this section, we will create stored procedures in the SQL Pool that will be triggered in the pipeline once the curated data is loaded in the SQL Pool.

1. Open the SQL script (in the "Develop section") called **EXE 2 StoredProceduresCabs**. This script will create four stored procedures that you will later integrate in your pipeline once the load operation in the SQL Pool happens.
2. Click on Run and it will run the entire SQL script



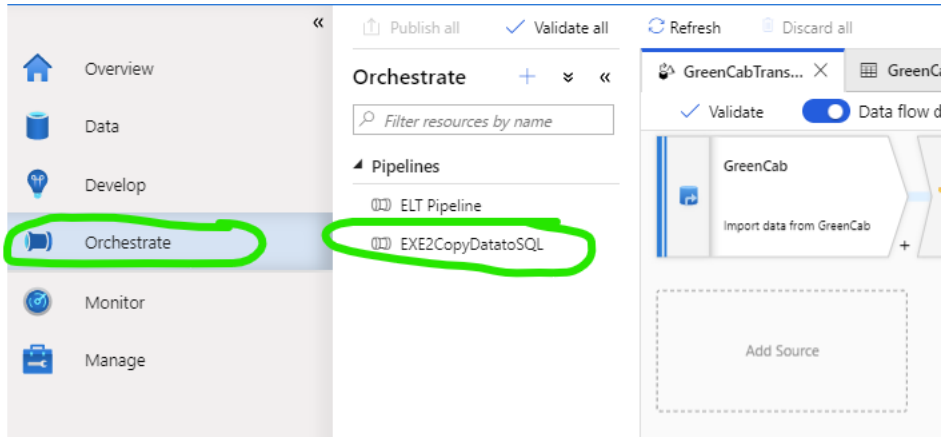
3. Check that you can see the four stored procedures by browsing your SQL Pool:



Task 4 – Understand a pipeline that contains all activities

This pipeline had been run before the lab. This pipeline copied curated data from the lake into a SQL Pool as staging table. A staging table is a table optimized for loading/writing data. Once the copy activity was complete, it triggered a stored procedure that transformed the staging table into a destination table whose goal is to provide strong read performance. Exercise 4 will highlight the speed of reading data!

1. Click into the *Orchestrate* section
2. Click into **…** in the *Pipelines*



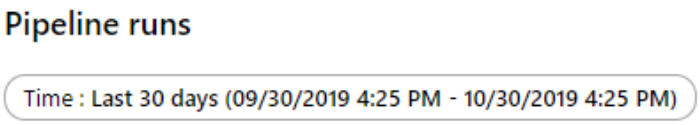
3. Click on **EXE2CopyDatatoSQL**
4. Check the various activities run in the pipeline

Bonus Challenge: Try to find the activity to orchestrate a Synapse notebook.

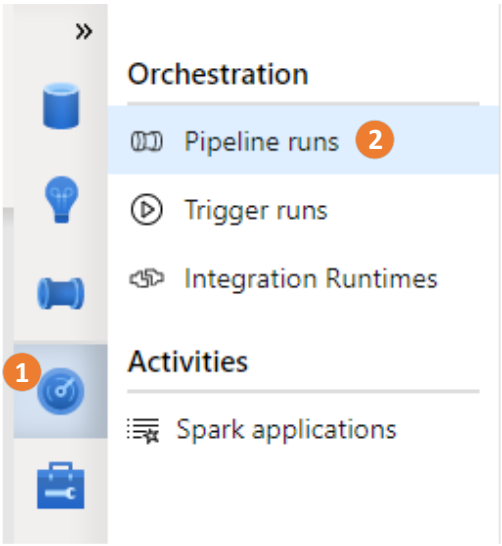
Task 5 – Monitor the pipeline

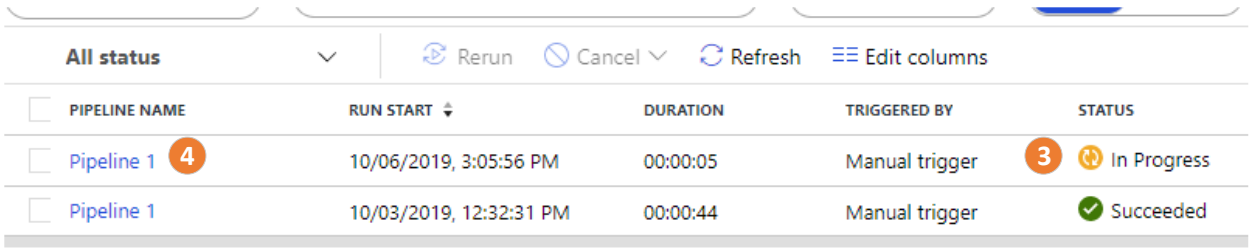
Synapse will offer a single pane of glass to monitor activities. We will first look at the pipeline that you have triggered.

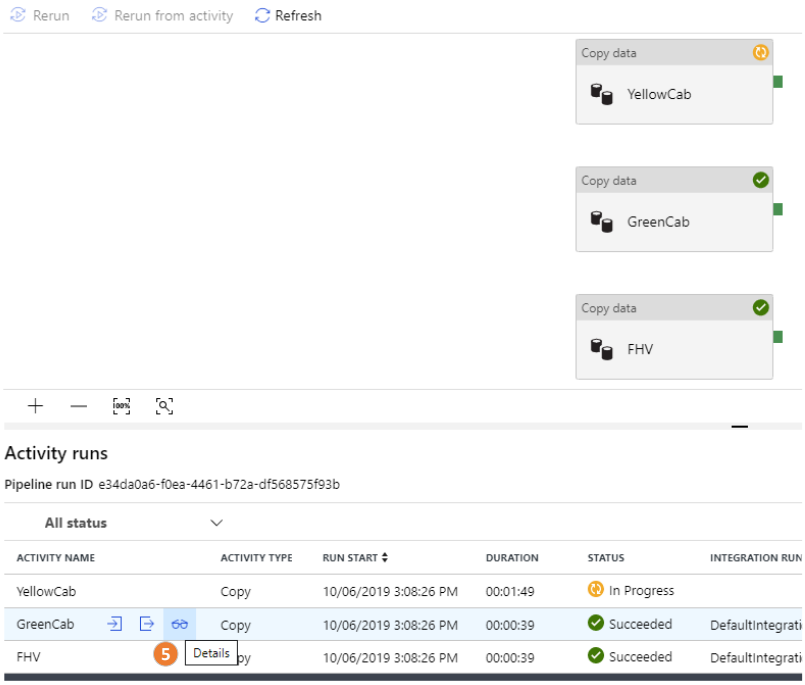
1. Click at the section Monitoring
2. Click into *Pipeline runs*, make sure that the time of filtering is set to *Last 30 days*



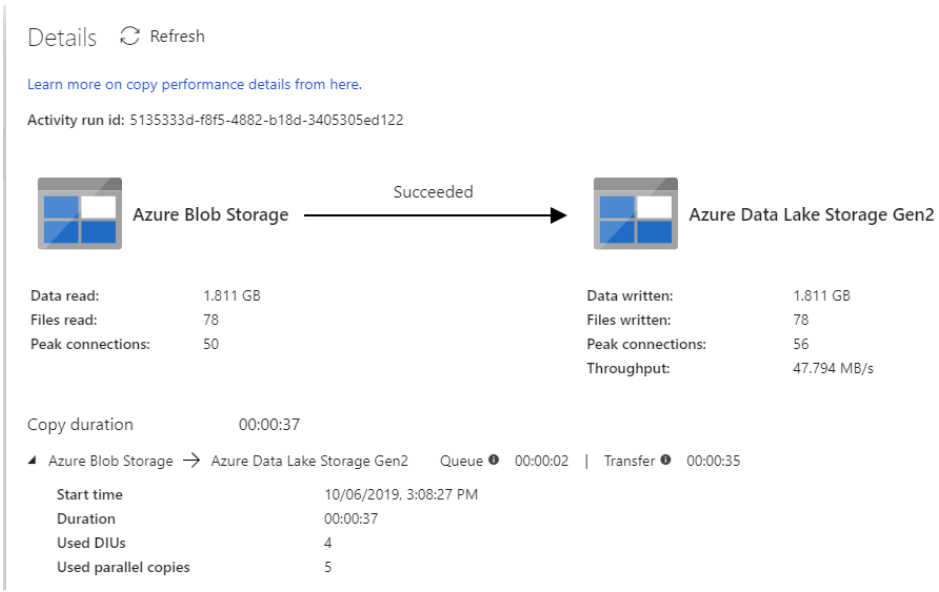
3. Check the status of your pipeline. It should be in-progress
4. To understand at a more granular level, the status of the activities in the pipeline, click on the pipeline







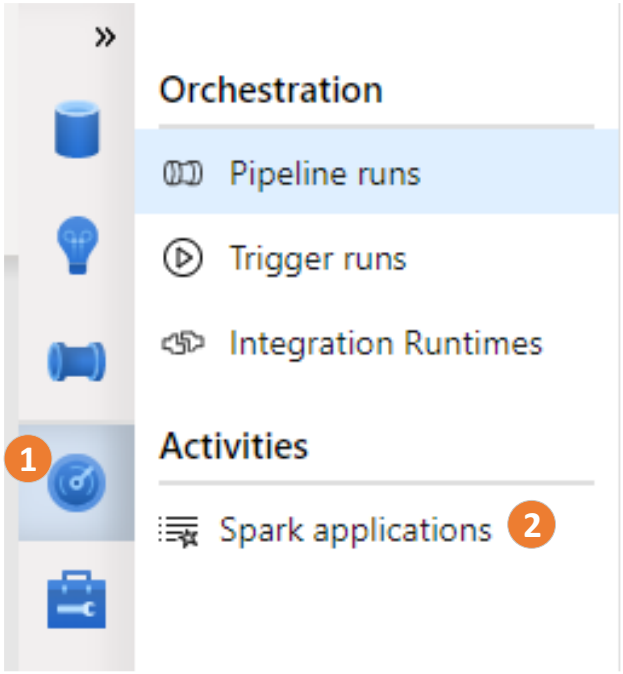
5. You can get the detail of an activity by clicking to the binocular icon. In this case, you can see the performance of a copy command activity. Check the various details based on the different activities you ran. (Data Flow, notebook, SQL Script)



Task 6 – Monitor Spark applications

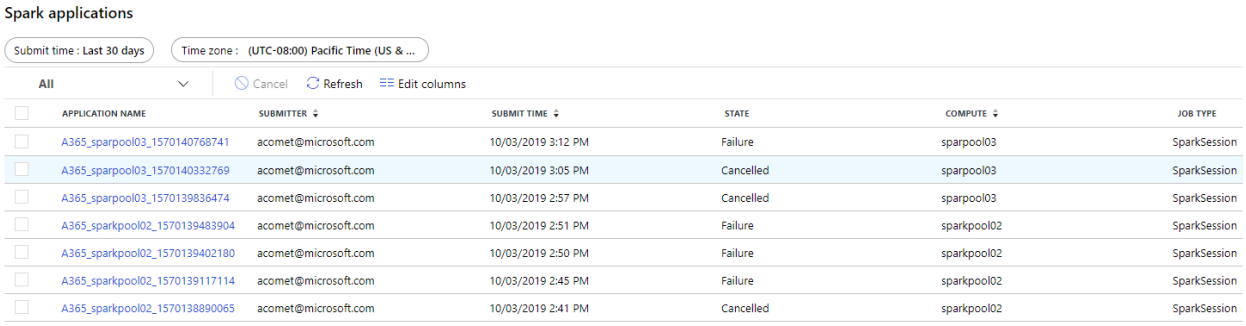
A Spark application consists of many activities that have run into a single session. A session is displayed as in-progress, failed or cancelled. When a user is done with her job, the application session ends as "cancelled". It is by no mean a negative impact.

1. Click at the section Monitoring
2. Click into Spark Applications

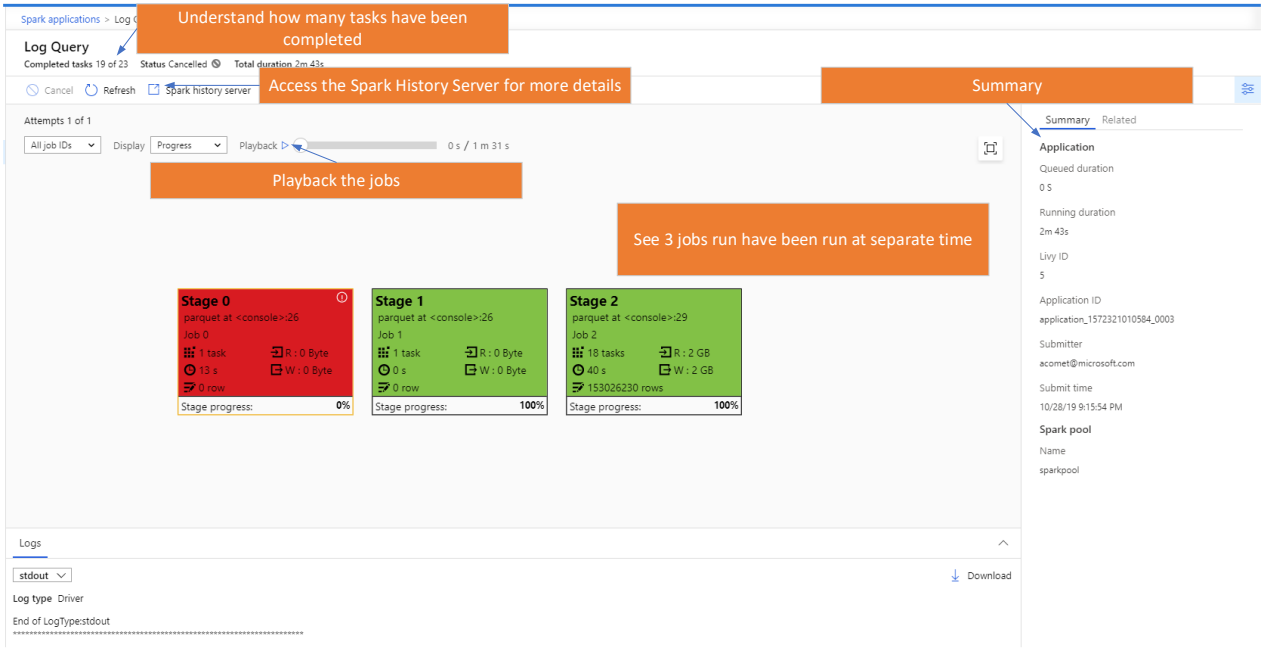


1

3. You can see the application name (and click into it for more details), see who submitted this job, the time, the stats, which pool template was leveraged and the job type.



4. Click into one application and check the picture below highlighting some of the monitoring functionalities



Exercise 3 – Power BI Integration

In this exercise you will be able to create a Power BI Report and build a visualization within Synapse Analytics leveraging previously created datasets.

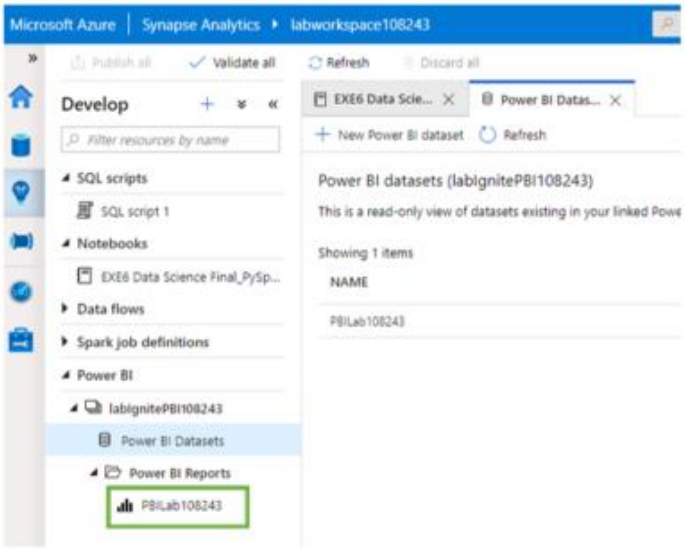
Pre-requisites: The following set-up steps have been completed for you:

1. A Power BI workspace under you tenant
2. A Synapse Linked Service has been created using “Connect to Power BI” and the workspace created at #1
3. Create a Power BI data set using Synapse

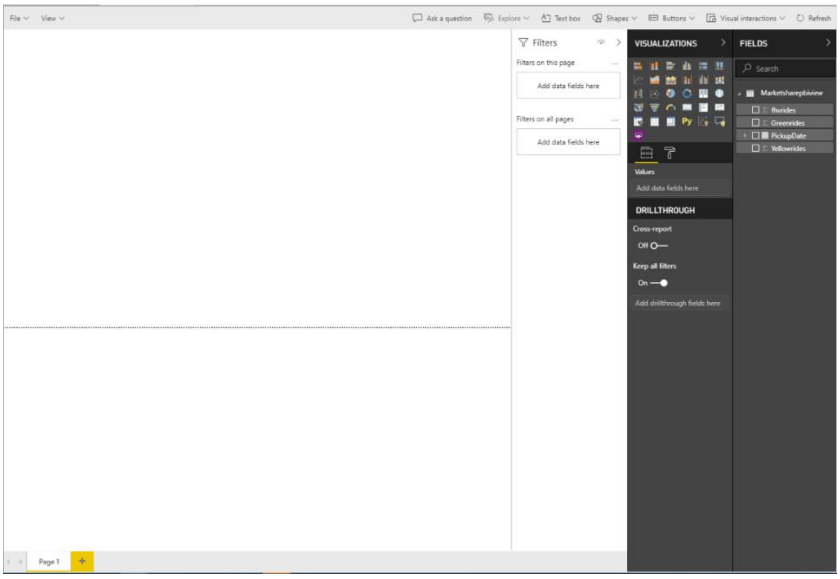
You will create a Power BI report

Create the Power BI Report in Synapse

Click on the Power BI Report with your Lab code

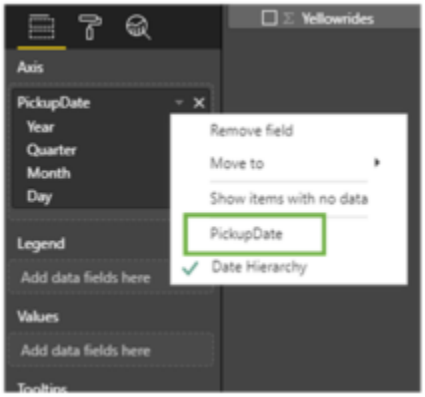


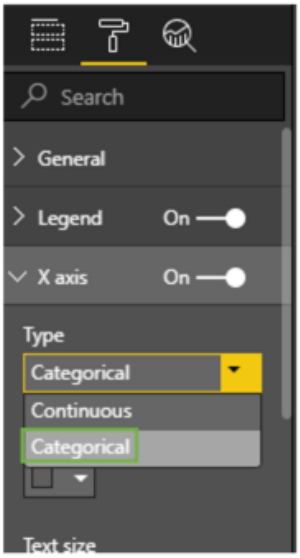
The Power BI Report Builder will appear



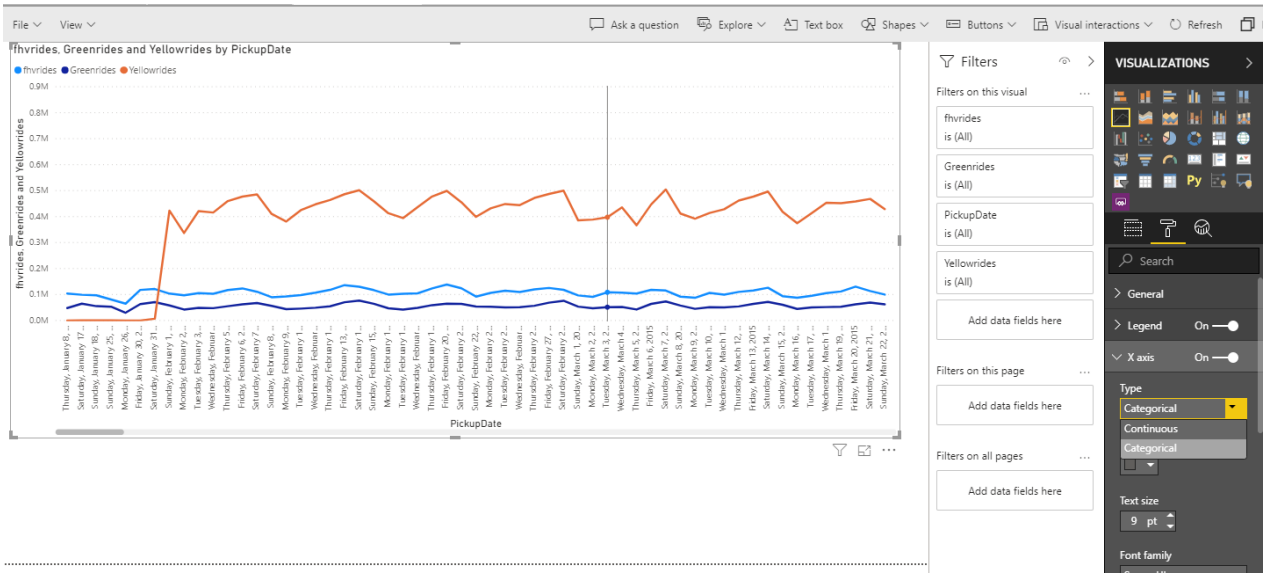
You can now build your report based on the Imported SQL Dataset

Select from the Visualization the Line Chart and drag and drop the “PickupDate” into the Axis and the other fields into the Values and then adjust the report:

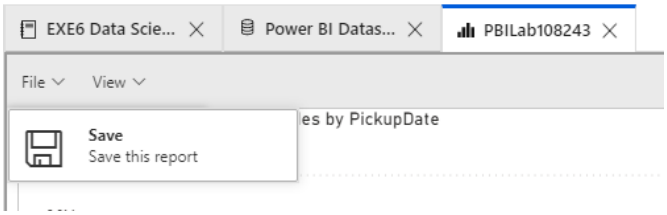




You will get your report



You can Save the Report to the Workspace:



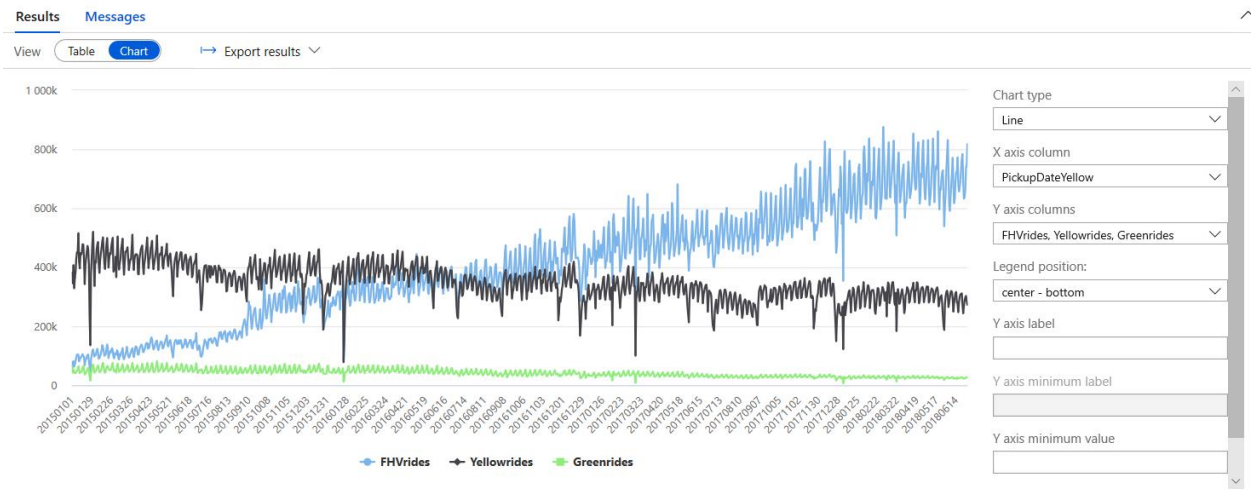
You can view the report in both Synapse Analytics and PowerBI

Exercise 4 (Elective) – High Performance Analysis with SQL Analytics Pool

Task 1 – SQL Pool query to understand market shares between cab companies

This query will be a simple exercise to understand the evolution of over the time of the amounts of daily rides that the yellow cabs, green cabs and for hire vehicle (includes companies like Uber and Lyft) served in New York.

Bonus Challenge: Can you run a query that will aggregate the count of rides per day for each view and join these three views together per day? Try to display the results in a chart similar to below:



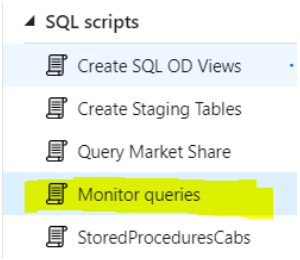
Solution:

1. Select the SQL Script called **Query Market Share** and run the script against the SQL Pool database. ▷ Run
2. Select Chart
3. From the Y axis columns unselect **PickupDate**
4. Select **PickupDate** for the X-Axis column

Task 2 – Monitor the queries through the DMV

Monitoring the queries that run in SQL Analytics Pool is very simple. You can look at the queries that have run in your SQL Pool

1. Select **Monitor queries** SQL Script in the *Analyze* section



2. Run the script against the SQL Pool database. Run

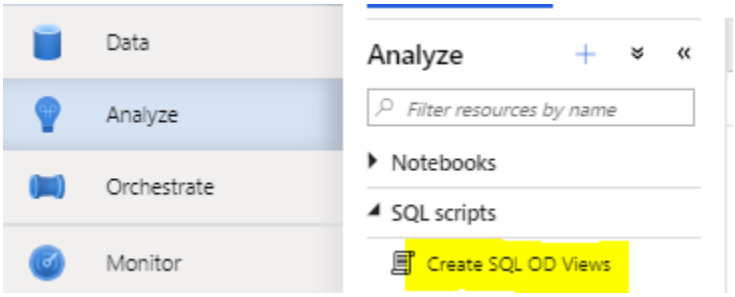
Bonus Challenge: Try to find the query that you ran in task 1

Exercise 5 (Elective) – Create views with SQL Analytics On-Demand

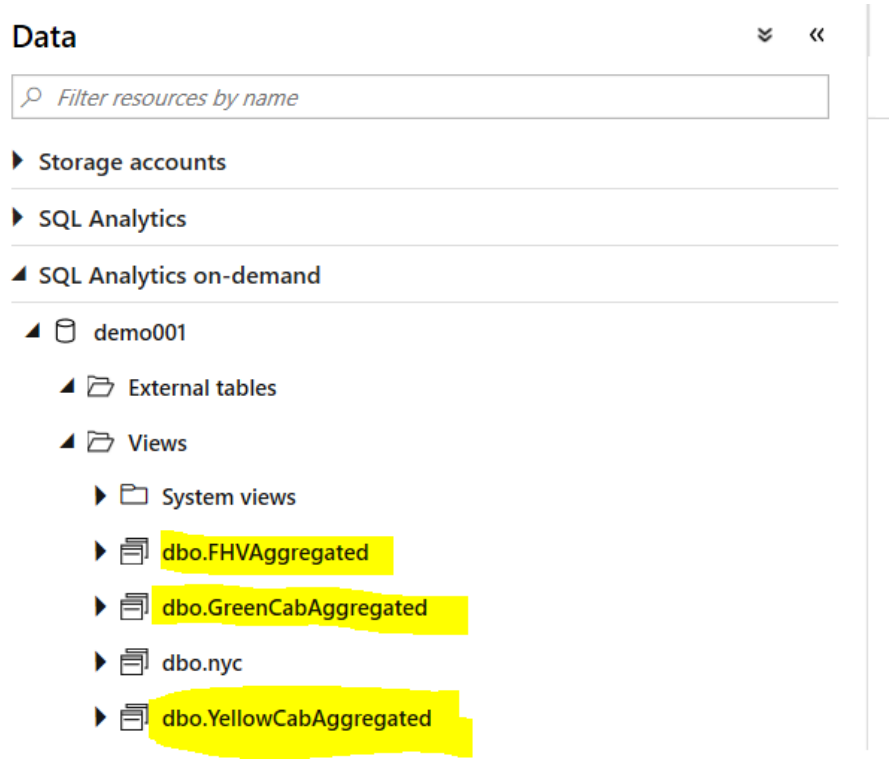
In this section, we will look at the same query you ran in a SQL Pool (Exercise 4) but over the Data Lake. If you have not done exercise 4, then this simple exercise is to understand the evolution of over the time of the amounts of daily rides that yellow cabs, green cabs and for hire vehicles (includes companies like Uber and Lyft) served in New York. Performance to query the lake will not be as strong as the query performance in a SQL Pool but SQL Analytics On-Demand is a powerful and flexible capability for data exploration and low cost BI with infrequent access to the lake. No data movement is required.

We will first ask you to create the three views over the data lake:

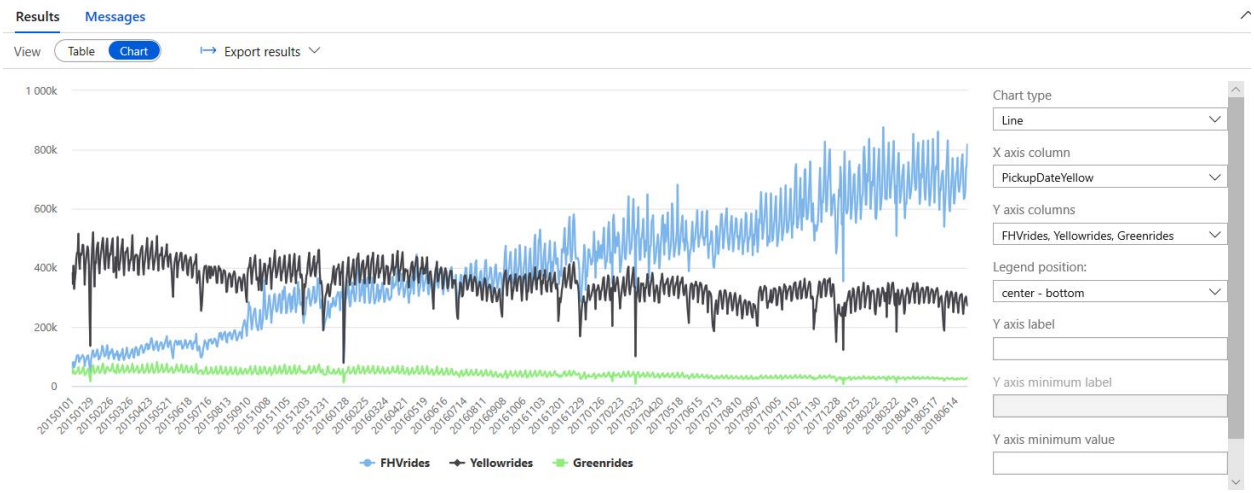
1. Select the SQL Script called **Create SQL OD Views**



2. Make sure that script is connect to the *SQL on-demand* and the master database
3. Run the script Run
4. Check that the three views are available (refresh if needed the SQL Analytics On-Demand)



Bonus Challenge: Can you run a query that will aggregate the count of rides per day for each view and join these three views together per day? Try to display the results in a chart similar to below:



Solution:

3. Run the following SQL Script against SQL On-Demand and the database (not master):

```
Select PickupDateYellow,FHVrides,Yellowrides, Greenrides from
dbo.YellowCabAggregated
        INNER JOIN dbo.FHVAggregated ON dbo.FHVAggregated.PickupDateFHV =
dbo.YellowCabAggregated.PickupDateYellow
        INNER JOIN dbo.GreenCabAggregated ON
dbo.GreenCabAggregated.PickupDateGreen = dbo.YellowCabAggregated.PickupDateYellow
```

```
ORDER BY PickupDateFHV ASC
```

4. Select Chart
5. From the Y axis columns unselect **PickupDateYellow**
6. Select **PickupDateYellow** for the X-Axis column

Exercise 6 (Elective) – Data Science with Spark

In this exercise you will play the role of a Data Scientist that based on the NYC Yellow Cab Dataset (that tracks trips and various attributes) using Synapse Notebook creates a model to predict for a given trip whether there will be a tip or not.

Create a new Notebook (for details check Exercise 3).

1. Configure and author your notebook:
    a. Attach your Spark Compute
    b. Select Spark as a language: "Pyspark"
    c. Click on "Add text" or " {} Add code" for each cell below:

For text cell:

For code cell:

Cell 1 – "Text cell"

Predict NYC Taxi Tips using Spark ML and Azure Open Datasets

The notebook ingests, visualizes, prepares and then trains a model based on an Open Dataset that tracks NYC Yellow Taxi trips and various attributes around them.

The goal is to predict for a given trip whether there will be a tip or not.

The Notebook "EXE6 Data Science Final_PySpark" is uploaded for you in the "Develop" section and rich text is provided to explain every single step. Note that you will need to replace the ADLS G2 Storage account `"<YourADLSAccount>"` with your Account name.

Here's a summary of the steps you will be performing:

Ingest Data

Get the data from the Open Datasets store and then down sample using filtering and sampling to generate a smaller set of data to make it faster/easier to evaluate different approaches to prep for the modelling phase later in the notebook.

Exploratory Data Analysis

Look at the data and evaluate its suitability for use in a model, do this via some basic charts focused on tip values and relationships.

Data Prep and Featurization

It's clear from the visualizations above that there are a bunch of outliers in the data. These will need to be filtered out in addition there are extra variables that are not going to be useful in the model we build at the end.

Finally there is a need to create some new (derived) variables that will work better with the model.

Data Prep and Featurization Part 2

Having created new variables its now possible to drop the columns they were derived from so that the dataframe that goes into the model is the smallest in terms of number of variables, that is required.

Also create some more features based on new columns from the first round.

Encoding

Different ML alogirthms support different type sof input, for this example Logistic Regression is being used for Binry Classification. This means that any Categorical (string) variables must be converted to numbers.

The process is not as simple as a "map" style function as the relationship between the numbers can introduce a bias in the resulting model, the approach is to index the variable and then encode using a std approach called One Hot Encoding.

This approach requires the encoder to "learn"/fit a model over the data in the Spark instance and then transform based on what was learnt.

Generation of Testing and Training Data Sets

Simple split, 70% for training and 30% for testing the model. Playing with this ratio may result in different models.

Train the Model

Train the Logistic Regression model and then evaluate it using Area under ROC as the metric.

The ROC is a graphical plot that illustrates the diagnostic ability.

For our Model the “Area under ROC = 0.989821882951654” and this is considered excellent

Evaluate and Visualize

Plot the actual curve to develop a better understanding of the model.

See the Area under the ROC model