# Analytics Vidhya –JobaThon – September 2022

## Problem statement

GoShop is an online e-commerce company that caters to different products to users across the globe. The company is growing at an exponential rate and has seen tremendous growth in the last 2 years.

The users can place their orders through the company website. The company tracks each and every activity of the user on the website. This information collected from the user activity is known as Clickstream data.

Clickstream data is the information that is collected about a user while they are browsing through a website. This includes a wide range of information such as -

* Which browser did the user use to visit the website?
* What page did the user visit on the website?
* Was the user logged in while visiting the website?
* Did the user click on a certain element of the webpage?

Now, the company would like to use Clickstream data to understand the user behavior on the website. In this hackathon, as a Data Engineer of the company, you will need to build an ETL process with the Clickstream data.

## Objective

You will be given the Clickstream data of the company. Your task at hand is to implement an ETL process and generate a table that will help the company to know more about its user base. This table can be used by the company in answering the following questions -

* The users who visited on a particular date.
* How many registered and non-registered users visited the website on a particular date?
* Which was the first URL the user visited on the website on a particular date?
* Determining the no. of Clickstream events occurring on the website.

## Environment setup

You are allowed to use only PySpark to develop the solution.

About the Dataset  
You are provided with a sample dataset containing 2 files - Clickstream and Login.

1. Clickstream (20% of the total records will be shared with the participants in JSON format)
2. Login (All the records will be shared with the participants in CSV format)

## Data Dictionary

### Clickstream

This table contains information about the clicks occurred on the website from 1st August, 2022 to 10th August, 2022.

|  |  |  |
| --- | --- | --- |
| **Variable** | **Datatype** | **Description** |
| browser\_id | string | Id of browser from which user is accessing the website. |
| session\_id | string | Id of the session created for the visiting user. |
| client\_side\_data | string | Embedded JSON element containing - current\_page\_url, time\_elapsed  **current\_page\_url** - The URL of the page the user has visited with the click.  **time\_elapsed** - The time spent by the user on the particular page. |
| event\_date\_time | string | Date and time when the event occurred on the website. |
| event\_type | string | Whether the click was a simple **click** event or a **pageload**event. A new page is loaded on a **pageload** event. |

### Login

This table contains information about when the user logged in on the website from 1st August, 2022 to 10th August, 2022.

|  |  |  |
| --- | --- | --- |
| **Variable** | **Datatype** | **Description** |
| login\_date\_time | string | Date and time at which the user logged in to the website. |
| session\_id | string | Id of the session created for the visiting user. |
| user\_id | string | Unique id of registered user on the website. |

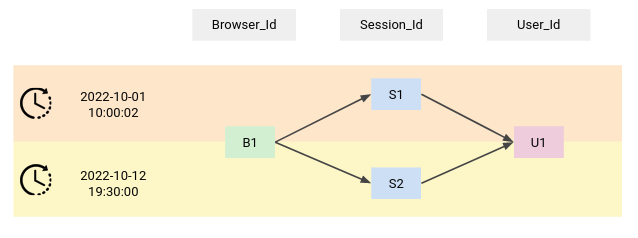
## Submission File Format

You need to submit the .py file that creates the output file similar to that of sample submission file. The format of the sample submission file is given below:

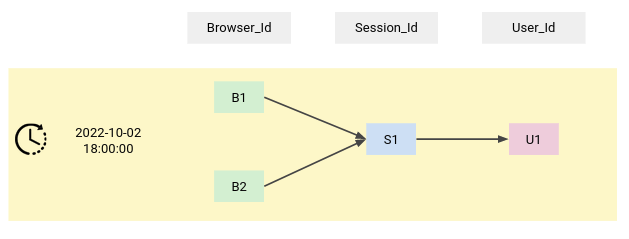
|  |  |  |
| --- | --- | --- |
| **Variable** | **Datatype** | **Description** |
| current\_date | string | The date of the click. |
| browser\_id | string | Id of browser from which user is accessing the website. |
| user\_id | string | Unique id of registered user on the website. |
| logged\_in | string | 0/1 flag determining whether the user is logged in or not. |
| first\_url | string | Url of the page on which the user first interacted with on the particular date. |
| number\_of\_clicks | string | Number of **click** events for the given user on the given date. |
| number\_of\_pageloads | string | Number of **pageload** events for the given user on the given date. |

## Points to Remember

* The dataset contains the following scenarios
* Scenario 1 -
  + It is given that at any given point of time, a single **user\_id** maps to a single **session\_id**.
  + **session\_id**for a **user\_id** can change for various reasons like the user logged out of the website, or the session for a user became stale after being inactive for some time, etc.



* Scenario 2 -
  + A session\_id can map to multiple browser\_ids at any given time. This can happen when a user has logged in from two different browsers.



Solution:

# Import libraries

import pyspark.sql.functions as F

from pyspark.sql.window import Window

# Function input - spark object, click data path, resolved data path

# Function output - final spark dataframe

def sample\_function(spark, s3\_clickstream\_path, s3\_login\_path):

df\_clickstream = spark.read.format("json").load(s3\_clickstream\_path)

user\_mapping = spark.read.format("csv").option("header",True).load(s3\_login\_path)

# Join Clickstream with user mapping to Identify the User already registered

df = df\_clickstream.join(

user\_mapping,

'session\_id',

'left\_outer'

)

# Sine the client\_side\_data is STRUCT, using getField to extract URL and timespend

df = df.withColumn('client\_page\_url', F.col('client\_side\_data').getField("current\_page\_url"))\

.withColumn('current\_date',F.split('event\_date\_time',' ')[0])\

.drop('client\_side\_data')

#Agg the clickstream for user on a particular date

df\_grp = df.groupBy('user\_id','current\_date','browser\_id')\

.pivot('event\_type').agg({'event\_type':'count'})

df = df.join(

df\_grp,

['user\_id','current\_date','browser\_id'],

'left\_outer'

)

#Window Spec

window\_spec = Window\

.partitionBy('user\_id','current\_date','browser\_id')\

.orderBy(F.col('event\_date\_time').asc())

df = df.withColumn('logged\_in',F.when(F.col('pageload')>0,F.lit(1))\

.otherwise(F.lit('0')))\

.withColumn('row\_number',F.row\_number().over(window\_spec))\

.filter('row\_number == 1')\

.withColumn('pageload',F.coalesce(F.col('pageload'),F.lit(0))) \

.withColumn('click',F.coalesce(F.col('click'),F.lit(0))) \

.select('current\_date','browser\_id','user\_id','logged\_in',F.col('client\_page\_url').alias('first\_url'),\

F.col('click').alias('number\_of\_clicks'), F.col('pageload').alias('number\_of\_pageloads'))

return df