**Ministry of Higher Education and Scientific Research**
**Higher School of Computer Science, May 8, 1945, Sidi Bel Abbes**
**Academic year: 2024-2025** **Semester 2**
**2 SC: IASD** **Module: Big Data Technologies**

# Lab 6: Spark GraphX and GraphFrames

We will use a dataset consisting of two files:

- The first file, **"station_data.csv"**, represents stations.
- The second file, **"trip_data.csv"**, represents trips made using bikes.

1) Create a DataFrame for each file.

## Graph Construction

The first step is to build the graph. To do this, we need to define the vertices and edges, which are DataFrames with specifically named columns.

In our case, we are creating a directed graph. This graph will point from the source to the destination.

In the context of the bike trip data, this will indicate the starting location of a trip to the ending location. To define the graph, we use the column naming conventions introduced in the GraphFrames library. In the vertex table, we define our identifier as **id** (in our case, this is the **name** column). In the edge table, we rename the source vertex ID (the **Start Station** column) of each edge to **src** and the destination vertex ID (the **End Station** column) to **dst**.

2) In the vertex table (the DataFrame representing the file **station_data.csv**), rename the column **name** to **id**.
3) In the edge table (the DataFrame representing the file **trip_data.csv**), rename the column **Start Station** to **src** and the column **End Station** to **dst**.
4) Create a GraphFrame that represents our graph.

## Querying the Graph

Answer the following queries:

5) Return the number of trips made between each source and destination, sorted in descending order (based on the number of trips).
6) Return the number of trips that start or end at the station "Townsend at 7th", sorted in descending order (based on the number of trips).
7) Return the vertices that have never been a destination for a trip starting from "Spear at Folsom".
8) Return the station with the maximum number of incoming trips.
9) Return the trip with the longest duration.

## Subgraphs

10) Create a subgraph that contains only the trips that start or end at "Townsend at 7th".

## Motif Finding

11) Return all paths that form a "triangle" pattern between three stations.
12) Return all paths that pass through three vertices and start from "Townsend at 7th".