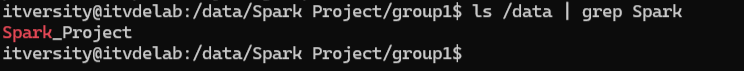
**Business Assumptions**

* We will assume that data (group of files) arrives to Local File System inside a folder named whatever, which means we will have a new folder every hour.

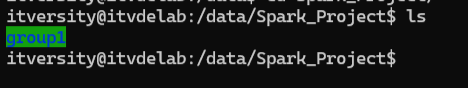
A screenshot of a computer

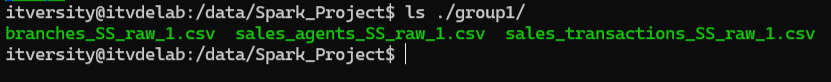
Description automatically generatedThis is how the folder will look like after 6 hours

* We will assume source data is stored locally on a folder named “Spark\_Project” which itself is inside a Folder named “/data”.



* “Spark Project” Folder contains a Single Folder for each group of files (Batch), initially it contains a single folder.



* Each folder contains the 3 files coming from source systems.
* After each group (Folder) is uploaded into the Data Lake it will be archived:
  + We will keep archived data in a folder named “Archived”.
  + We will use a specific data retention policy for this folder, let’s say we will keep data for 1 week for disaster recovery.
* Data comes to Local File System in a non-idempotent manner (Same group/batch can be pushed twice into LFS).

**Load Data Into HDFS**

* Source system pushes 3 files every hour which means we have various portioning options for storing Row Data in the Data Lake.
* We will divide the data into 3 Folders (one for each file).
* Each Folder will store the historical data for each file.
* We will store the Row Data inside HDFS in a folder named “/Spark\_Project/data/Q\_company/”
* The structure of “HDFS:/ Spark\_Project/data/Q\_company/”will go something like this:

/Spark Project/data/Q company/  
├── branches/  
│ ├── 2024-06-09-00  
│ ├── 2024-06-09-01  
│ ├── ...

│ ├── 2024-06-09-23  
├── sales\_agents/  
│ ├── 2024-06-09-00  
│ ├── 2024-06-09-01  
│ ├── ...

│ ├── 2024-06-09-23/  
├── sales\_transactions/

│ ├── 2024-06-09-00

│ ├── 2024-06-09-01

│ ├── ...

│ ├── 2024-06-09-23

* This structure will make it easier for us to read the data for each file and clean/process it.

**Loading Script:**

TODO:

* Automate putting data to Loca File System every 1 hour (crontab)