

# Mini-Projet

## Création d'un Framework BI pour l'analyse BIG Data



Encadré par

Mr. FENNAN Abdelhadi

Réalisé par

GHARBI Hafsa

CHERKAOUI Rabab

## Table des matières

<b>Introduction .....</b>	4
<b>Objectif.....</b>	5
<b>Collection de données.....</b>	6
1. Installation des outils .....	6
○ Scrapy.....	6
○ Splash & Twisted .....	6
○ Docker .....	7
○ Création du projet Scrapy .....	8
2. Ressources de données choisies.....	9
○ ACM.....	9
○ IEEE xplore.....	9
○ Science direct .....	9
3. Extraction des données .....	9
○ Le fichier items.py.....	9
○ Génération des spiders.....	10
○ Développement du contenu des spiders .....	10
<b>Stockage de données .....</b>	18
1. Installation des outils .....	18
○ MongoDB : .....	18
2. Enregistrement des données .....	18
○ Fichier settings.py .....	19
○ Fichier pipelines.py .....	19
<b>Analyse &amp; Visualisation Spark.....</b>	21
1. Installation des outils .....	21
○ Spark .....	21
○ Pyspark .....	22
2. Traitement et analyse des données .....	23

○ Connexion à la base de données .....	23
○ Traitement et analyse.....	24
<b>Analyse et visualisation BI.....</b>	<b>31</b>
1. Installation des outils .....	31
○ PDI.....	31
○ Power BI .....	32
2. Schéma en étoile.....	33
○ Exportation des données sous format CSV :.....	34
○ Génération des dimensions (exemple titre) : .....	34
○ Création des composants .....	35
○ Chargement dans la base de données .....	37
○ Exécution des transformations.....	38
○ Table de fait.....	39
○ Exécution de la transformation de la table de fait.....	40
○ Résultats dans MySql .....	41
○ Schéma en étoile .....	41
3. Visualisation et Reporting par Power BI.....	42
<b>Versions des outils utilisés .....</b>	<b>50</b>
<b>Problèmes rencontrés .....</b>	<b>51</b>
<b>Conclusion.....</b>	<b>52</b>
<b>Références.....</b>	<b>53</b>

## Introduction

Les responsables d'entreprises sont aujourd'hui confrontés à la nécessité de prendre des décisions toujours plus rapides, toujours plus précises et toujours plus efficaces.

Avec des stratégies adaptées et des logiciels performants, nous pouvons dès aujourd'hui prendre des décisions pilotées par les données, plus rapidement et avec une plus grande précision. On parle alors de Business Intelligence.

Ce que nous connaissons aujourd'hui sous le nom de business intelligence a commencé à être développé dans les années 1980, lorsque l'avènement de l'utilisation généralisée de l'ordinateur a rendu possible la collecte et l'analyse de données pour les entreprises. Au fil des ans, les processus de BI se sont élargis et améliorés pour inclure une exploration de données approfondie, des outils de visualisation de données et diverses méthodes d'analyse de données pour fournir aux décideurs commerciaux des informations importantes.

Les principales avancées en matière de business intelligence incluent la capacité de collecter et de gérer des ensembles de données extrêmement volumineux, la capacité de combiner des données externes et internes, un partage accru des données et la création de tableaux de bord de business intelligence.

## Objectif

L'objectif de projet est la création d'un petit Framework BI pour l'analyse du BIG Data.

Elle fait intervenir les technologies et les compétences suivantes :

- La collection de données avec Scrapy.
- Le Stockage de données volumineuses avec MongoDB.
- L'analyse des Données avec Apache Spark.
- Mise en place d'un datawarehouse et analyse par la plateforme BI : Pentaho Data Integration.
- Visualisation et générations des rapports avec Microsoft Power BI.

## Collection de données

### 1. Installation des outils

#### ○ Scrapy

Scrapy est un framework d'exploration Web gratuit et open-source écrit en python. Il a été conçu à l'origine pour effectuer du web scraping, mais peut également être utilisé pour extraire des données à l'aide d'API. Il est maintenu par Scrapinghub ltd.

Scrapy est un package complet pour le téléchargement des pages Web, le traitement et le stockage des données sur les bases de données.

#### [Installation de scrapy](#)

```
> pip install scrapy
```



# Scrapy

#### ○ Splash & Twisted

Splash est une solution interne pour le rendu JavaScript, implémentée en Python à l'aide de Twisted et QT . Splash est un navigateur Web, open-source, léger capable de traiter plusieurs pages en parallèle, d'exécuter du JavaScript personnalisé dans le contexte de la page, et bien plus encore.

#### [Installation de splash](#)

```
PS C:\Users\user\Documents\LSI-S5\Business Intelligence & Big Data\Projet\Projet BI> pip install scrapy-splash
Collecting scrapy-splash
  Downloading scrapy_splash-0.8.0-py2.py3-none-any.whl (27 kB)
Installing collected packages: scrapy-splash
Successfully installed scrapy-splash-0.8.0
WARNING: You are using pip version 21.1.1; however, version 21.3.1 is available.
You should consider upgrading via the 'c:\users\user\appdata\local\programs\python\python38\python.exe -m pip install --upgrade pip' command.
```

#### [Installation de twisted](#)

	Twisted-20.3.0-cp37-cp37m-win32.whl	21/12/2021 23:05	Fichier WHL	3 017 Ko
--	-------------------------------------	------------------	-------------	----------

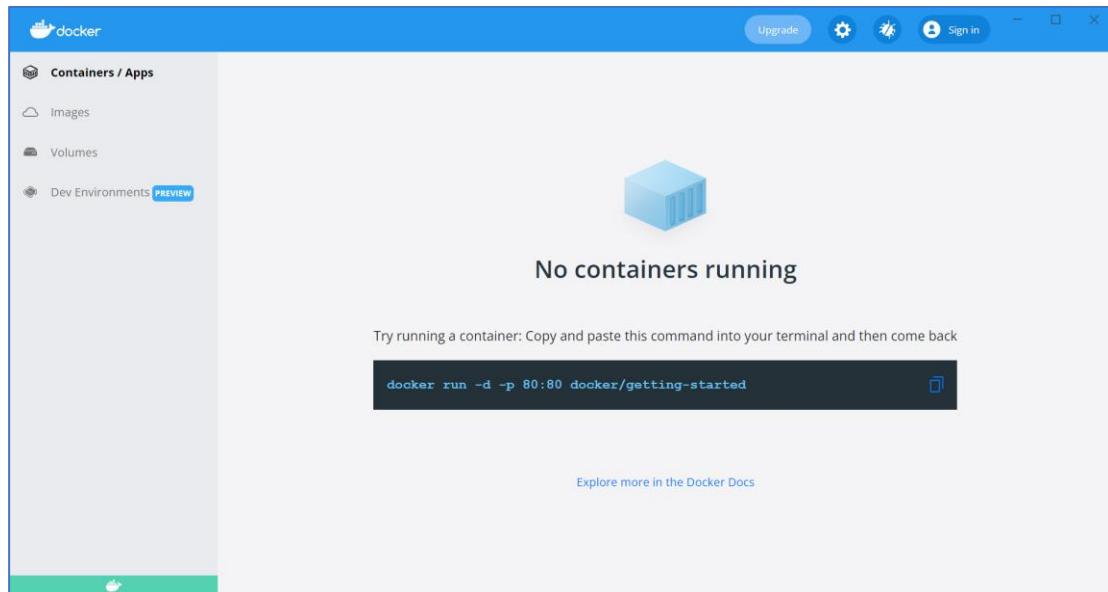
```
C:\Users\user\Documents\LSI-S5\Business Intelligence & Big Data\Projet>pip install Twisted-20.3.0-cp37-cp37m-win32.whl
Processing c:\users\user\documents\lsi-s5\business intelligence & big data\projet\twisted-20.3.0-cp37-cp37m-win32.whl
Collecting hyperlink>=17.1.1 (from Twisted==20.3.0)
  Using cached https://files.pythonhosted.org/packages/6e/aa/8caf6a0a3e62863cbb9dab27135660acba46903b703e224f14f447e5793
4/hyperlink-21.0.0-py2.py3-none-any.whl
Collecting zope.interface>=4.4.2 (from Twisted==20.3.0)
  Downloading https://files.pythonhosted.org/packages/0f/26/8d5d40ee551a84a099bcc35006de6f47dd13f26d2f8a27ef0b0879588d5
/zope.interface-5.4.0-cp37-cp37m-win32.whl (208kB)
    [██████████] 215kB 819kB/s
Collecting constantly>=15.1 (from Twisted==20.3.0)
  Using cached https://files.pythonhosted.org/packages/b9/65/48c1909d0c0aeae6c10213340ce682db01b48ea900a7d9fce7a7910ff31
8/constantly-15.1.0-py2.py3-none-any.whl
Collecting incremental>=16.10.1 (from Twisted==20.3.0)
  Using cached https://files.pythonhosted.org/packages/99/3b/4f80dd10cb716f3a9e22ae88f026d25c47cc3fdf82c2747f3d59c98e4ff
1/incremental-21.3.0-py2.py3-none-any.whl
Collecting Automat>=0.3.0 (from Twisted==20.3.0)
  Using cached https://files.pythonhosted.org/packages/dd/83/5f6f3c1a562674d65fc320257bcd0873ec53147835aeeff7762fe758527
3/Automat-20.2.0-py2.py3-none-any.whl
Collecting PyHamcrest!=1.10.0,>=1.9.0 (from Twisted==20.3.0)
```

## ○ Docker

Docker permet d'embarquer une application dans un ou plusieurs containers logiciels qui pourra s'exécuter sur n'importe quel serveur machine, qu'il soit physique ou virtuel. Docker fonctionne sous Linux comme Windows Server. C'est une technologie qui a pour but de faciliter les déploiements d'application, et la gestion du dimensionnement de l'infrastructure sous-jacente.



### Installation de docker :



## ○ Création du projet Scrapy

On crée un projet Scrapy en exécutant la commande suivante :

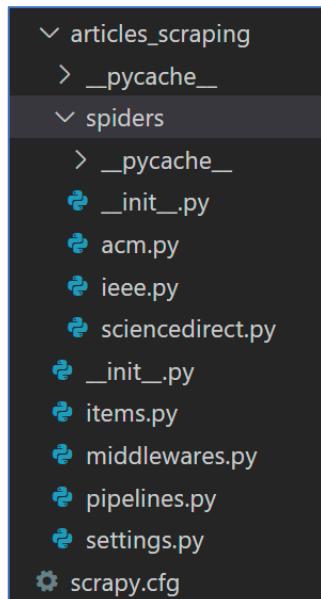
```
C:\Windows\System32\cmd.exe
Microsoft Windows [version 10.0.19043.1415]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\LENOVO\Desktop\CI LSI\S5\BI & Big Data\Projet BI & Big Data>scrapy startproject articles_scraping
New Scrapy project 'articles_scraping', using template directory 'c:\users\lenovo\appdata\local\programs\python\python38\lib\site-packages\scrapy\templates\project', created in:
  C:\Users\LENOVO\Desktop\CI LSI\S5\BI & Big Data\Projet BI & Big Data\Projet\articles_scraping

You can start your first spider with:
  cd articles_scraping
  scrapy genspider example example.com

C:\Users\LENOVO\Desktop\CI LSI\S5\BI & Big Data\Projet BI & Big Data\Projet>
```

L'arborescence du projet :



## 2. Ressources de données choisies

### ○ ACM

ACM est la revue scientifique majeure de l' Association for Computing Machinery. C'est une revue à comité de lecture qui couvre l'informatique en général, et plus particulièrement les aspects théoriques.

### ○ IEEE xplore

La bibliothèque numérique IEEE Xplore est une base de données de recherche pour la découverte et l'accès aux articles de revues, aux actes de conférence, aux normes techniques et aux documents connexes sur l'informatique, l'électrotechnique et l'électronique, et les domaines connexes.

### ○ Science direct

ScienceDirect est un site web géré par l'éditeur Elsevier. Lancée en mars 1997, la plateforme permet d'accéder à plus de 3 800 revues académiques qui forment plus de 14 millions de publications scientifiques revues par des pairs.

## 3. Extraction des données

### ○ Le fichier items.py

Ce fichier contient les éléments qu'on va extraire pour chaque article :

- Les informations liées aux auteurs de l'article :
  - Noms
  - Universités
  - Pays
- Les informations liées à l'article :
  - Titre
  - Sujet
  - Doi
  - Date de publication
  - Abstract
  - Références
  - Téléchargements
  - Citations
- Les informations liées au journal :
  - Nom

- Issn
- Impact factor
- Indexation

```
class ArticlesScrapingItem(scrapy.Item):
    # Authors
    authors_name = scrapy.Field()
    authors_university = scrapy.Field()
    authors_country = scrapy.Field()
    latitude = scrapy.Field()
    longitude = scrapy.Field()

    # Article
    title = scrapy.Field()
    topic = scrapy.Field()
    doi = scrapy.Field()
    date_publication = scrapy.Field()
    keywords = scrapy.Field()
    abstract_ = scrapy.Field()
    references = scrapy.Field()
    downloads = scrapy.Field()
    citations = scrapy.Field()

    # Journal
    journal_name = scrapy.Field()
    issn = scrapy.Field()
    impact_factor = scrapy.Field()
    indexation = scrapy.Field()

    pass
```

## ○ Génération des spiders

Génération du spider pour le journal ACM :

```
scrapy genspider acm dl.acm.org
```

Génération du spider pour le journal IEEE xplore :

```
scrapy genspider ieee ieeexplore.ieee.org
```

Génération du spider pour le journal Science direct :

```
scrapy genspider sciencedirect www.sciencedirect.com
```

## ○ Développement du contenu des spiders

Par la suite, on modifie les fonctions générées pour prendre en considération la structure de chaque site web et les champs qu'on souhaite extraire :

## ACM

Les initialisations des variables, la boucle **for** pour la pagination :

```
class AcmSpider(scrapy.Spider):
    name = 'acm'
    topic = "None"
    allowed_domains = ['dl.acm.org']
    start_urls = []

    handle_httpstatus_list = [302]
    handle_httpstatus_list = [301]

    def __init__(self, keyword=None, topic=None, *args, **kwargs):
        super(AcmSpider, self).__init__(*args, **kwargs)
        self.start_urls = ['https://dl.acm.org/action/doSearch?AllField=' + topic]
        for i in range(100):
            self.start_urls = ['https://dl.acm.org/action/doSearch?AllField=' + topic + '&startPage=' + str(i) + '&pageSize=' + str(40)]

        self.topic = topic
```

La fonction parse qui boucle sur les articles et fait appel par la suite à la fonction **parse\_article** qui va extraire les éléments de chaque article :

```
def parse(self, response):
    for article in response.css("a::attr(href)"):
        if '/doi/' in article.extract():
            yield SplashRequest('https://dl.acm.org' + article.extract(), self.parse_article, args={'wait': 3})
            time.sleep(5)
```

La fonction parse\_article :

```
def parse_article(self, response):
    item = ArticleScrapingItem()

    # ----- Déclaration -----
    # Article
    title = response.css('.citation_title::text').extract_first()
    topic = self.topic
    doi = response.css('.issue-item_doi::text').extract()
    date_publication = response.css('.CitationCoverDate::text').extract()
    abstract_ = response.css('.hlFld-Abstract').css('p::text').extract()
    references = response.css('.references_note::text').extract()
    downloads = ';' .join(response.css('.tooltip_metric').css('span::text').extract())
    citations = ';' .join(response.css('.tooltip_citation').css('span::text').extract())

    # Authors
    authors_name = response.css('.author-data').css('span::text').extract()
    authors_infos = response.css('.author-info_body').css('p::text').extract()
    authors_university = []
    authors_country = []
```

```
if len(authors_infos) != 0:
    for auth_info in authors_infos:
        if auth_info != "":
            s = auth_info.split(',')
            authors_university.append(s[0])

            if len(s) > 1:
                authors_country.append(s[len(s) - 1])
            else:
                authors_country = ""

        else:
            authors_university = ""
            authors_country = ""

# Journal
journal_name = "ACM"
issn = "00045411; 1557735X"
impact_factor = 0
indexation = "Oui"

# ----- Affectation -----
# Article Affectation
item['title'] = title
item['topic'] = topic
item['doi'] = ''.join(doi)

try:
    item['date_publication'] = int(''.join(date_publication).split(' ')[-1])
except:
    item['date_publication'] = 0

item['abstract_'] = ''.join(''.join(abstract_))
item['references'] = ''.join(''.join(references))

try:
    item['downloads'] = int(''.join(downloads).split(';')[0].replace(',', ''))
except:
    item['downloads'] = 0

try:
    item['citations'] = int(''.join(citations).split(';')[0].replace(',', ''))
except:
    item['citations'] = 0
```

```

# Authors Affectation
item['authors_name'] = ';' .join(authors_name)
item['authors_university'] = ';' .join(authors_university)
item['authors_country'] = ';' .join(authors_country)
item['latitude'] = 0
item['longitude'] = 0

# Journal Affectation
item['journal_name'] = journal_name
item['issn'] = issn

def impact_factor_of_year(year):
    switcher = {
        2013: 5.88,
        2014: 3.00,
        2015: 3.84,
        2016: 3.10,
        2017: 6.02,
        2018: 5.85,
        2019: 5.26,
        2020: 5.83
    }
    return switcher.get(year, 0)

item['impact_factor'] = impact_factor_of_year(item['date_publication'])
item['indexation'] = indexation

yield item

```

## IEEE xplore

Les initialisations :

```

import requests
import scrapy
from articles_scraping.items import ArticlesScrapingItem

class IeeeSpider(scrapy.Spider):
    name = 'ieee'
    topic = None
    start_urls = None
    page_no = 1
    r = None
    headers = {
        "Accept": "application/json, text/plain, */*",
        "Origin": "https://ieeexplore.ieee.org",
        "Content-Type": "application/json",
    }

```

```
payload = {  
    "newsearch": True,  
    "queryText": topic,  
    "highlight": False,  
    "returnFacets": ["ALL"],  
    "returnType": "SEARCH",  
    "pageNumber": page_no,  
}
```

```
def __init__(self, keyword=None, topic=None, *args, **kwargs):  
    super(IeeeSpider, self).__init__(*args, **kwargs)  
    self.start_urls = ['https://ieeexplore.ieee.org/rest/search']  
    self.topic = topic  
    self.payload['queryText'] = topic  
    self.r = requests.post(  
        "https://ieeexplore.ieee.org/rest/search",  
        headers=self.headers,  
        json=self.payload  
)
```

La fonction parse :

```
def parse(self, response):  
    item = ArticlesScrapingItem()  
  
    page_data = self.r.json()  
    for record in page_data["records"]:  
  
        print(record)  
  
        # ----- Déclaration ----- #  
        # Article  
        title = record["articleTitle"]  
        topic = self.topic  
        try:  
            doi = record["doi"]  
        except:  
            doi = ""  
        date_publication = record["publicationYear"]  
        abstract = record["abstract"]  
        references = ""  
        downloads = record["downloadCount"]  
        citations = record["citationCount"]
```

```
# Authors
authors_infos = record["authors"]
authors_name = []
authors_university = ""
authors_country = ""

l = len(authors_infos)
for i in range(l):
    auth = authors_infos[i]
    authors_name.append(auth['preferredName'])

# Journal
journal_name = "IEEE"
issn = "21682372"
impact_factor = 3.825
indexation = "Oui"

# ----- Affectation -----
# Article Affectation
item['title'] = title
item['topic'] = topic
item['doi'] = doi
try:
    item['date_publication'] = date_publication
except:
    item['date_publication'] = 0
item['abstract'] = abstract
item['references'] = references
item['downloads'] = downloads
item['citations'] = citations

# Authors Affectation
item['authors_name'] = authors_name
item['authors_university'] = authors_university
item['authors_country'] = authors_country

# Journal Affectation
item['journal_name'] = journal_name
item['issn'] = issn
item['impact_factor'] = impact_factor
item['indexation'] = indexation

yield item
```

## Science direct

Les initialisations :

```
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By
from selenium.common.exceptions import TimeoutException
from selenium.webdriver.support.wait import WebDriverWait
import os
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
import scrapy
from articles_scraping.items import ArticlesScrapingItem
from selenium.webdriver import ActionChains

class SciedirectSpider(scrapy.Spider):
    name = 'sciedirect'
    start_urls = None
    chrome_options = Options()
    driver = webdriver.Chrome(executable_path=os.path.abspath("C:/Users/LENOVO/Desktop/chromedriver_win32/chromedriver/chromedriver.exe"))
    options=chrome_options

    def __init__(self, topic=None, *args, **kwargs):
        super(SciedirectSpider, self).__init__(*args, **kwargs)
        self.start_urls = ['https://www.sciencedirect.com/search']
        self.driver.get("https://www.sciencedirect.com/search?qs=" + topic)
```

La fonction parse :

```
def parse(self, response):
    delay = 10 # (seconds)
    try:
        myElem = WebDriverWait(self.driver, delay).until(EC.presence_of_element_located((By.ID, 'srp-results-list')))
        elements = self.driver.find_elements_by_css_selector("div.result-item-content h2 span a")
        for element in elements:
            article = element.get_attribute("href")
            driver1 = webdriver.Chrome(executable_path=os.path \
                .abspath("C:/Users/LENOVO/Desktop/chromedriver_win32/chromedriver/chromedriver.exe"),options=self.chrome_options)
            driver1.get(str(article))

            # ---- Déclaration ---- #
            # Authors
            authors_name = []
            authors_infos = []
            authors_university = []
            authors_country = []

            # Article
            title = ""
            topic = ""
            doi = ""
            date_publication = 0
            abstract = ""
            references = []
            citations = 0
            download = 0
```

```
try:
    elem = WebDriverWait(driver1, delay).until(EC.presence_of_element_located((By.ID, 'abstracts')))
    elements1 = driver1.find_elements_by_css_selector("span.title-text")
    for element1 in elements1:
        title = element1.text
    elements1 = driver1.find_elements_by_css_selector(
        "div#author-group.author-group a.author.size-m.workspace-trigger span.content")
    for element1 in elements1:
        authors_name.append(element1.text)

    # authors_infos
    elems = driver1.find_elements_by_css_selector("button#show-more-btn")
    ActionChains(driver1).click(elems[0]).perform()
    elements1 = driver1.find_elements_by_css_selector("dl.affiliation dd")
    for element1 in elements1:
        authors_infos.append(element1.text)

    elements1 = driver1.find_elements_by_css_selector("a.doi")
    for element1 in elements1:
        doi = element1.text
    elements1 = driver1.find_elements_by_css_selector("div#publication.Publication div.publication-volume div.text-xs")
    for element1 in elements1:
        date_publication = element1.text
    elements1 = driver1.find_elements_by_css_selector("div#abSTRACTS.Abstracts.u-font-serif")
    for element1 in elements1:
        abstract = element1.text

elements1 = driver1.find_elements_by_css_selector(
    "ul li.bib-reference.u-margin-s-bottom")
for element1 in elements1:
    references.append(element1.text)

except TimeoutException:
    print("Loading took too much time!")
driver1.quit()

item = ArticlesScrapingItem()
item['title'] = title
item['abstract'] = abstract
item['authors_name'] = authors_name
item['topic'] = topic
item['doi'] = doi
item['date_publication'] = date_publication
item['journal_name'] = "ScienceDirect"
item['downloads'] = download
item['references'] = references

yield item
except TimeoutException:
    print("Loading took too much time!")
self.driver.quit()
```

## Stockage de données

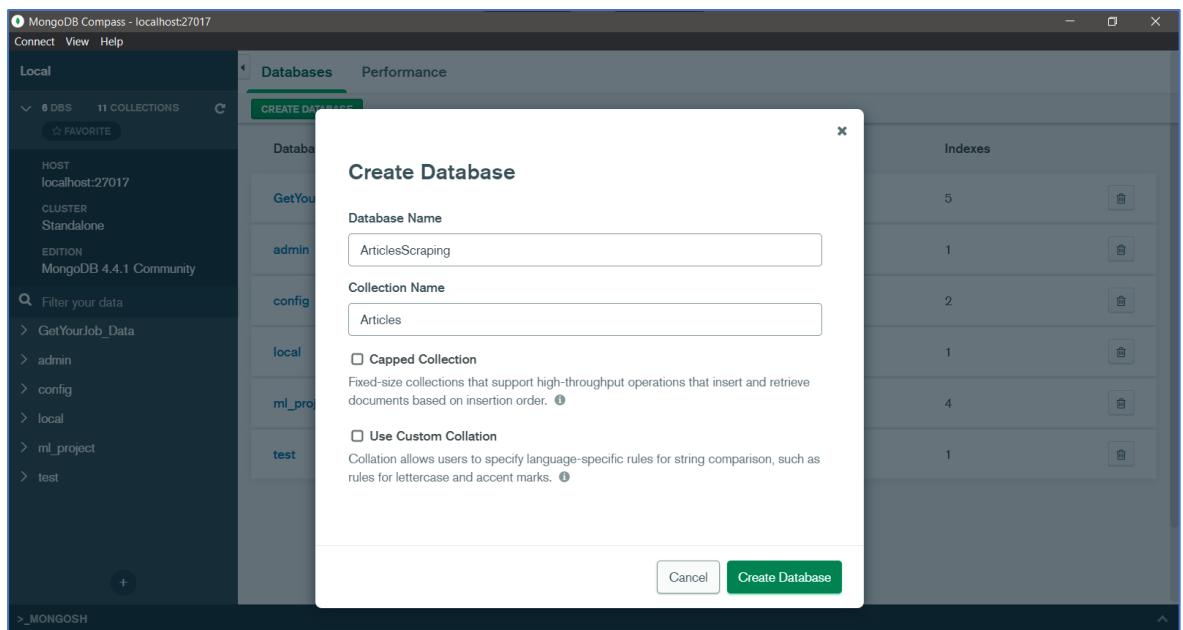
### 1. Installation des outils

#### ○ MongoDB

MongoDB est une base de données NoSQL orientée document utilisée pour stocker de gros volumes de données. Au lieu d'utiliser des tables et des lignes comme dans les bases de données relationnelles traditionnelles, MongoDB utilise des collections et des documents.



- Crédation d'une base de données avec une collection pour stocker les articles :



### 2. Enregistrement des données

Pour stocker les données dans la base de données, on doit réaliser une connexion avec MongoDB

## ○ Fichier settings.py

Dans ce fichier, on détermine les éléments de connexion :

```
ITEM_PIPELINES = {
    'articles_scraping.pipelines.MongoDBPipeline': 300,
}

MONGODB_SERVER = "localhost"
MONGODB_PORT = 27017
MONGODB_DB = "ArticlesScraping"
MONGODB_COLLECTION = "Articles"
MONGODB_URI = "mongodb://localhost:27017"
```

## ○ Fichier pipelines.py

Ce fichier contient une classe contenant les fonctions d'enregistrement après extraction :

```
import pymongo

class MongoDBPipeline(object):
    def __init__(self, mongo_uri, mongo_db, mongo_collection):
        self.mongo_uri = mongo_uri
        self.mongo_db = mongo_db
        self.mongo_collection = mongo_collection

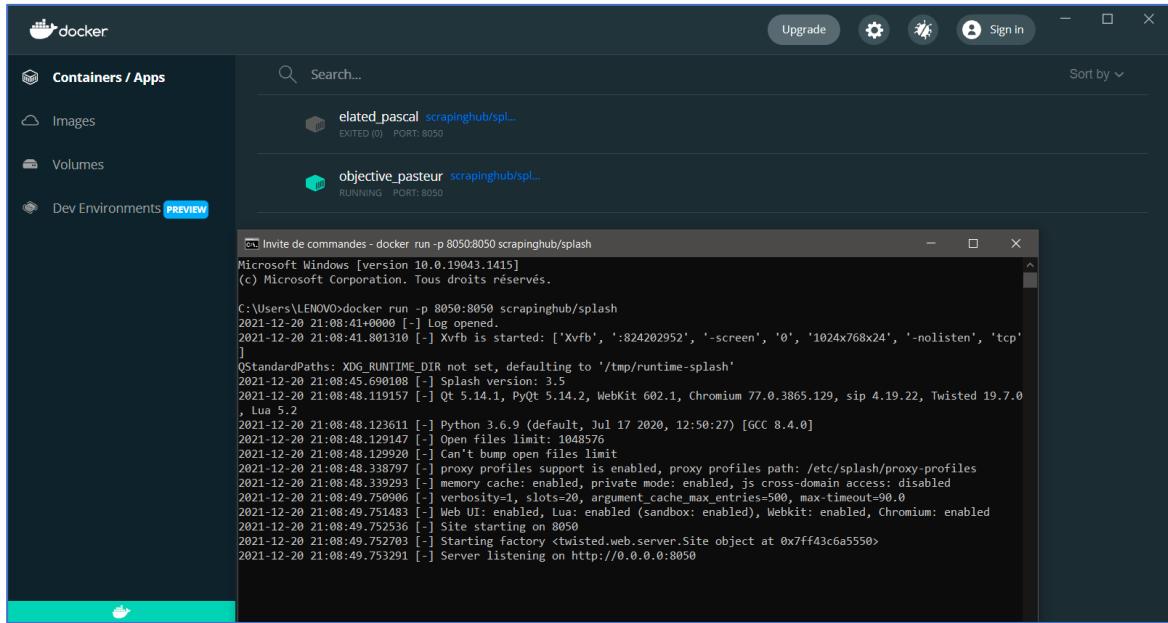
    @classmethod
    def from_crawler(cls, crawler):
        return cls(
            mongo_uri = crawler.settings.get('MONGODB_URI'),
            mongo_db = crawler.settings.get('MONGODB_DB', 'items'),
            mongo_collection = crawler.settings.get('MONGODB_COLLECTION', 'items')
        )

    def open_spider(self, spider):
        self.client = pymongo.MongoClient(self.mongo_uri)
        self.db = self.client[self.mongo_db]

    def close_spider(self, spider):
        self.client.close()

    def process_item(self, item, spider):
        self.db[self.mongo_collection].insert_one(dict(item))
        return item
```

Avant l'extraction des données, on lance docker avec la commande suivante :



## Exécution

Pour lancer le scrapping, on exécute la commande suivante (exemple de ACM avec le Machine Learning comme sujet) :

```
> scrapy crawl acm -a topic="Machine learning"
```

## Résultats

```
_id:ObjectId("61df1f47b601ac135cbd3b12")
title:"Artificial intelligence in CS1"
topic:"Artificial Intelligence"
doi:""
date_publication:2005
abstract:"The GuessingGame program from introductory artificial intelligence ser..." 
references:"gamenmaster.com Homepage, retrieved January 20, 2005."
downloads:207
citations:1
authors_name:"Brian C. Ladd"
authors_university:"St. Lawrence University"
authors_country:"Ny"
latitude:0
longitude:0
journal_name:"ACM"
issn:"00045411; 1557735X"
impact_factor:0
indexation:"Oui"

_id:ObjectId("61df1ff52b601ac135cbd3b13")
title:"Artificial Intelligence Computation"
```

# Analyse & Visualisation Spark

## 1. Installation des outils

### ○ Spark

Apache Spark est un système de traitement open source distribué, couramment utilisé pour les charges de travail de Big Data. Apache Spark utilise une mise en mémoire cache et une exécution optimisée pour offrir des performances élevées, et prend en charge le traitement par lot général, les analyses en continu, et les bases de données orientées graphes.

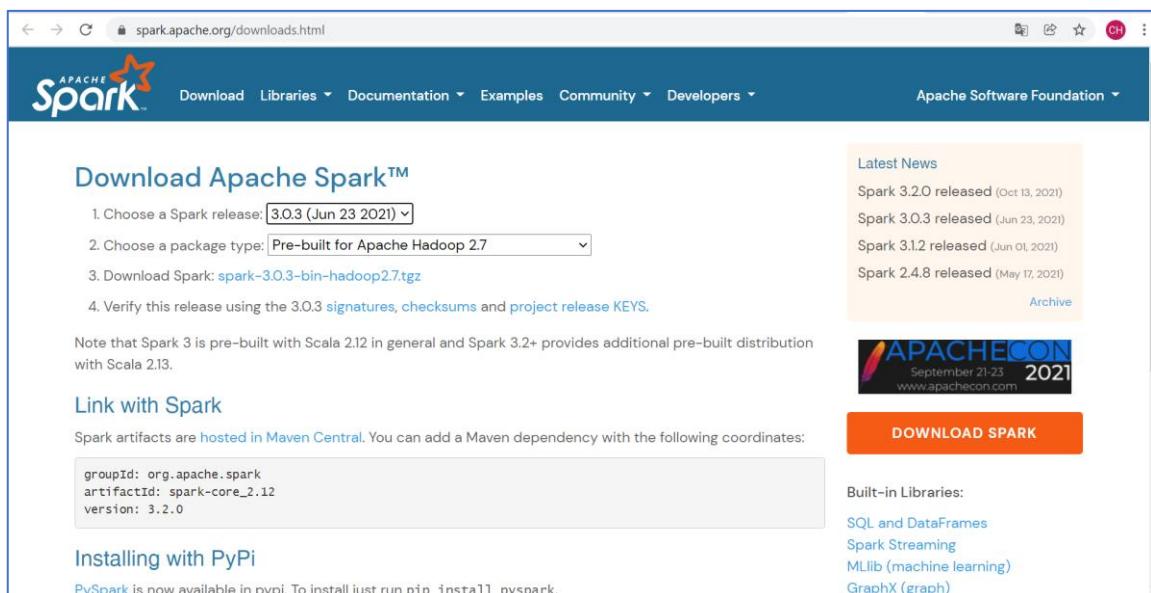


Spark permet de manipuler d'importants volumes de données en utilisant une API de bas niveau. Pour simplifier l'exploration des données, Spark SQL offre une API de plus haut niveau avec une syntaxe SQL. Spark SQL permet ainsi de réaliser, très rapidement, de nombreuses opérations sans écrire de code.

### Installation de spark

On installe Apache Spark depuis le lien suivant :

<https://spark.apache.org/downloads.html>



The screenshot shows the Apache Spark downloads page. At the top, there's a navigation bar with links for Download, Libraries, Documentation, Examples, Community, Developers, and the Apache Software Foundation. The main content area has a heading "Download Apache Spark™". It includes a step-by-step guide:

- Choose a Spark release: 3.0.3 (Jun 23 2021)
- Choose a package type: Pre-built for Apache Hadoop 2.7
- Download Spark: spark-3.0.3-bin-hadoop2.7.tgz
- Verify this release using the 3.0.3 signatures, checksums and project release KEYS.

Note that Spark 3 is pre-built with Scala 2.12 in general and Spark 3.2+ provides additional pre-built distribution with Scala 2.13.

**Link with Spark**

Spark artifacts are hosted in Maven Central. You can add a Maven dependency with the following coordinates:

```
groupId: org.apache.spark  
artifactId: spark-core_2.12  
version: 3.2.0
```

**Installing with PyPi**

PySpark is now available in pypi. To install just run pip install pyspark.

**Latest News**

- Spark 3.2.0 released (Oct 13, 2021)
- Spark 3.0.3 released (Jun 23, 2021)
- Spark 3.1.2 released (Jun 01, 2021)
- Spark 2.4.8 released (May 17, 2021)

**Archive**

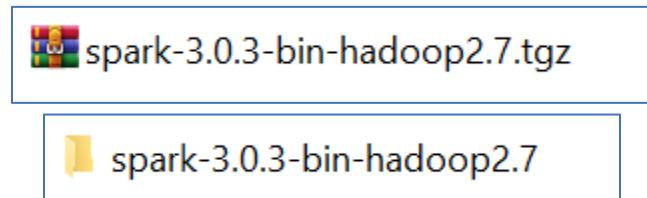
**APACHECON 2021**

**DOWNLOAD SPARK**

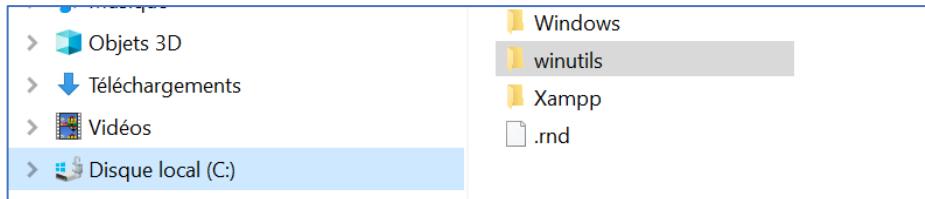
**Built-in Libraries:**

- SQL and DataFrames
- Spark Streaming
- MLlib (machine learning)
- GraphX (graph)

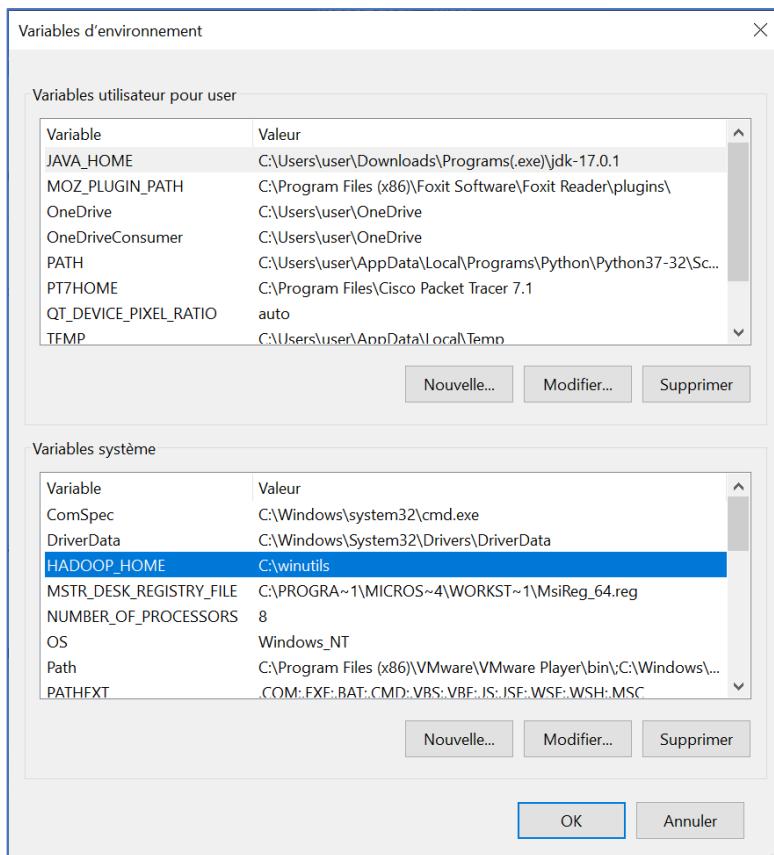
Le fichier téléchargé :



On doit ajouter les **Winutils** dans le disque local (C) pour le fonctionnement de Spark :



Création et ajout à la variable d'environnement HADOOP\_HOME :



## ○ Pyspark

PySpark est une bibliothèque Spark écrite en Python pour exécuter une application Python à l'aide des capacités d'Apache Spark. En utilisant PySpark,

nous pouvons exécuter des applications en parallèle sur le cluster distribué (nœuds multiples).

### Installation de pySpark

```
> pip install pyspark
Collecting pyspark
  Downloading pyspark-3.2.0.tar.gz (281.3 MB)
    |██████████| 281.3 MB 55 kB/s
  Preparing metadata (setup.py) ... done
Collecting py4j==0.10.9.2
  Downloading py4j-0.10.9.2-py2.py3-none-any.whl (198 kB)
    |██████████| 198 kB 384 kB/s
Using legacy 'setup.py install' for pyspark, since package 'wheel' is not installed.
Installing collected packages: py4j, pyspark
  Running setup.py install for pyspark ... done
Successfully installed py4j-0.10.9.2 pyspark-3.2.0
```

## 2. Traitement et analyse des données

### ○ Connexion à la base de données

Installation du connecteur MongoDB et Spark au lancement de Pyspark :

Le shell de Pyspark :

Ouverture d'une session Spark et connexion à MongoDB :

```
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("SparkApp") \
    .master("local") \
    .config("spark.mongodb.input.uri", "mongodb://localhost:27017/ArticlesScraping.Articles") \
    .config("spark.mongodb.output.uri", "mongodb://localhost:27017/ScrapingScraping.Articles") \
    .getOrCreate()
```

Chargement des données et affichage du schéma de la collection :

```
df = spark.read.format("com.mongodb.spark.sql.DefaultSource") \
    .option("uri", "mongodb://localhost:27017/ArticlesScraping.Articles").load()
df.printSchema()
```

## Schéma

```
root
 |-- _id: struct (nullable = true)
 |   |-- oid: string (nullable = true)
 |-- abstract_: string (nullable = true)
 |-- authors_country: string (nullable = true)
 |-- authors_name: string (nullable = true)
 |-- authors_university: string (nullable = true)
 |-- citations: integer (nullable = true)
 |-- date_publication: integer (nullable = true)
 |-- doi: string (nullable = true)
 |-- downloads: integer (nullable = true)
 |-- impact_factor: double (nullable = true)
 |-- indexation: string (nullable = true)
 |-- issn: string (nullable = true)
 |-- journal_name: string (nullable = true)
 |-- latitude: integer (nullable = true)
 |-- longitude: integer (nullable = true)
 |-- references: string (nullable = true)
 |-- title: string (nullable = true)
 |-- topic: string (nullable = true)
```

## ○ Traitement et analyse

### 1. Nombre d'articles par année

Filtrage et regroupement des données récupérées :

```

import matplotlib.pyplot as plt
import numpy as np

df = df[df['title'] != ""] # Eliminer les articles nulles
df = df[df['date_publication'] != 0] # Eliminer les articles ayant une date nulle

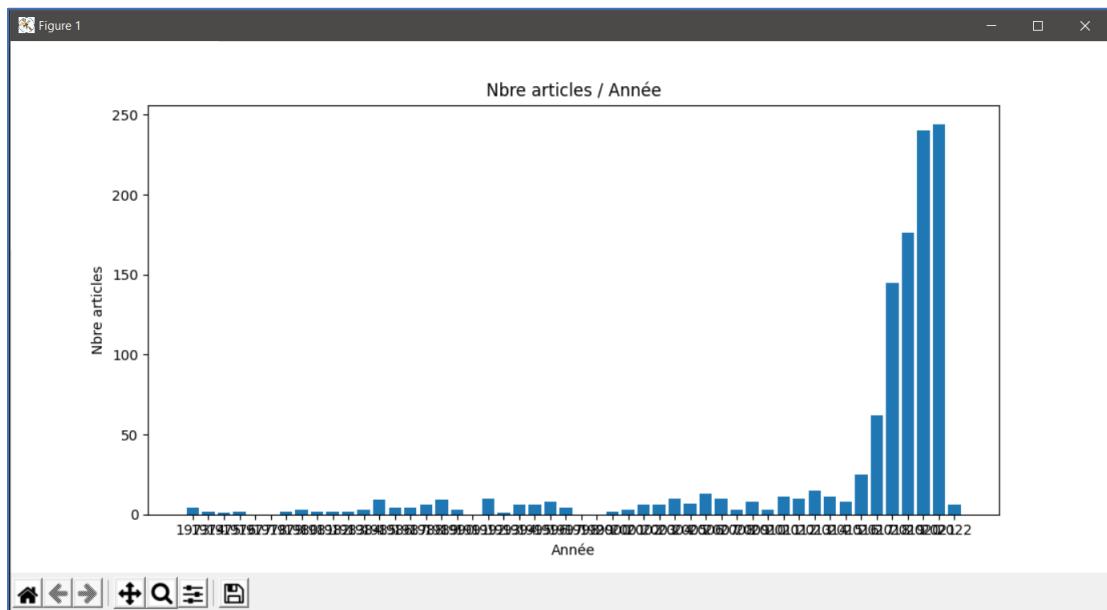
gr = df.groupBy("date_publication").count().sort("count", ascending=True) # Count et Group by
gr.show()
gr = gr.toPandas()

y = gr['count'].values.tolist()
x = gr['date_publication'].values.tolist()

plt.bar(x, y)
plt.xticks(np.arange(min(x), max(x)+1, 1.0)) # Ajuster l'échelle entre les années
plt.title("Nbre articles / Année")
plt.xlabel("Année")
plt.ylabel("Nbre articles")
plt.show()

```

### Résultat :



### Interprétation :

On observe que le nombre d'articles publiées dans les domaines de l'Intelligence Artificielle, le Machine Learning et le Blockchain était en évolution lente avant, mais dernièrement cette évolution est de plus en plus croissante et rapide.

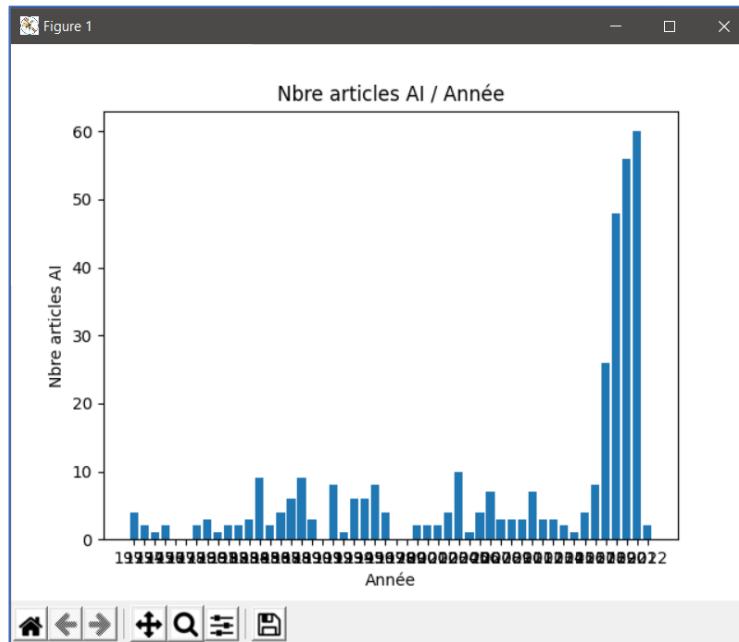
## 2. Nombre d'articles d'un sujet par année

Pour ce faire, on filtre les données par sujet :

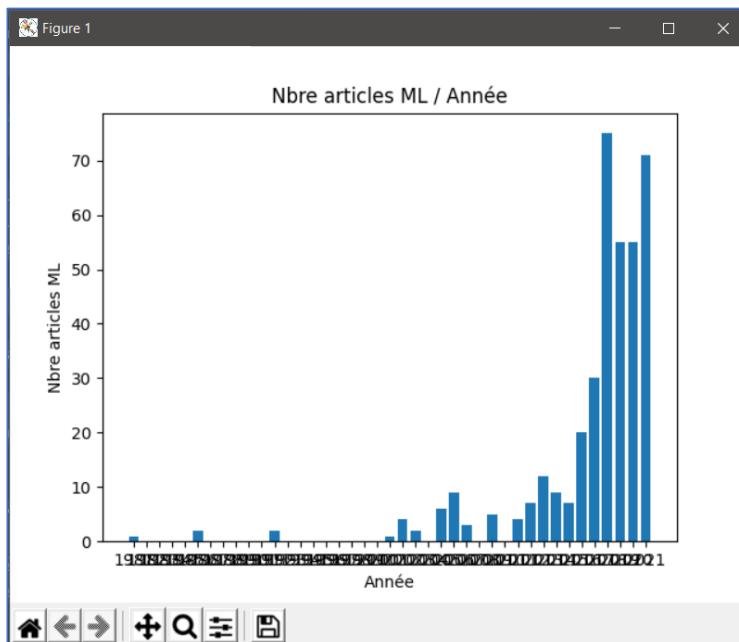
```
df = df[df['topic'] == "Artificial Intelligence"]
```

### Résultats :

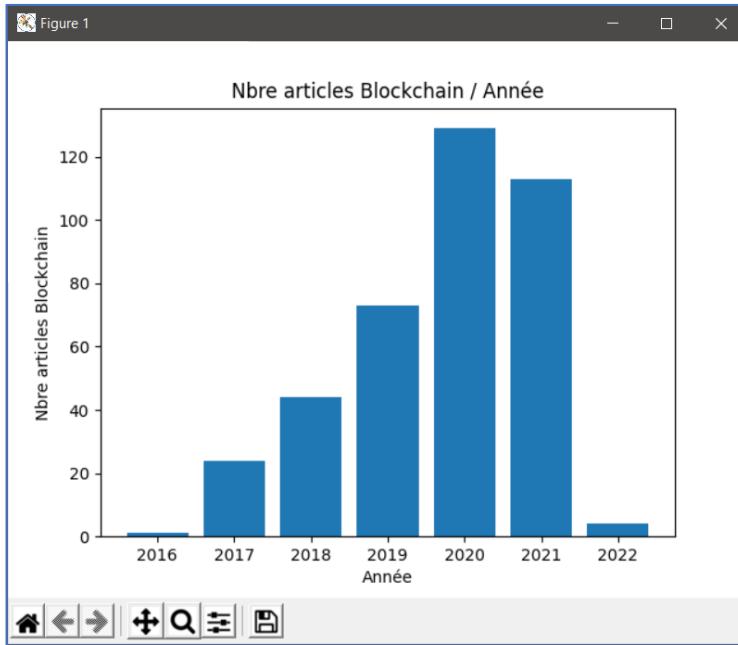
- Sujet « Intelligence Artificielle » :



- Sujet « Machine Learning » :



- Sujet « Blockchain » :



### Interprétation :

On observe que le Blockchain est un terme plus nouveau par rapport au IA et au ML, donc les années de publication sont récentes et le nombre d'articles est modeste par rapport aux autres, mais il est en évolution rapide aussi.

### 3. Nombre de téléchargement des articles d'un sujet par année

On utilise la fonction **sum()** de Python :

```
import matplotlib.pyplot as plt
import numpy as np

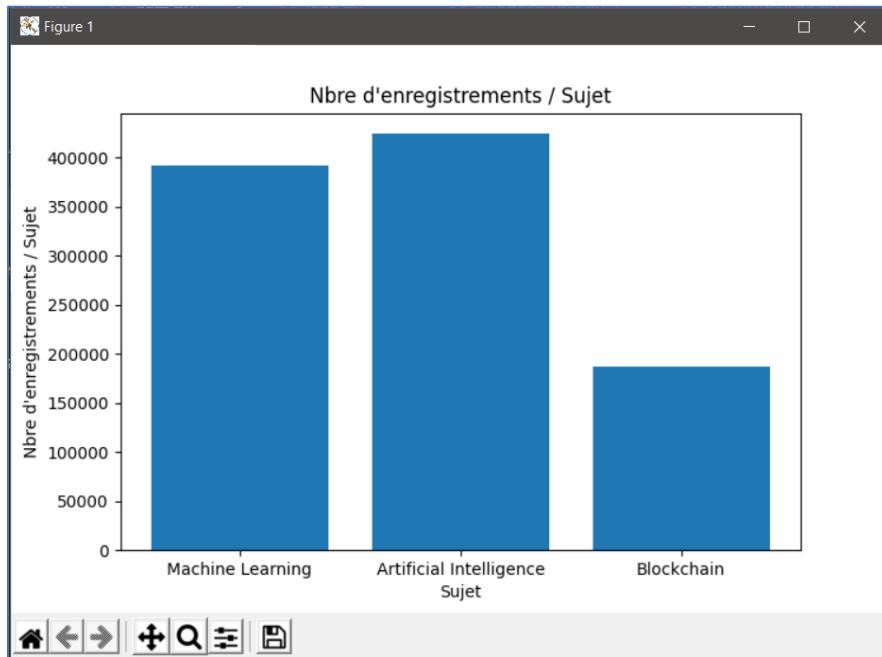
df = df[df['title'] != ""]
df = df[df['date_publication'] != 0]

g1 = df.groupby("topic").sum()
g1.show()
g1 = g1.toPandas()
g1

y = g1['sum(downloads)'].values.tolist()
x = g1['topic'].values.tolist()

plt.bar(x, y)
plt.title("Nbre d'enregistrements / Sujet")
plt.xlabel("Sujet")
plt.ylabel("Nbre d'enregistrements / Sujet")
plt.show()
```

## Résultat



## Interprétation :

On constate que les articles dans les sujets de l'Intelligence Artificielle et le Machine Learning sont plus téléchargés, ce qui est bien évidemment normal puisque le Blockchain est un nouveau concept dans le domaine de la recherche scientifique.

### 4. Nombre d'articles d'un sujet par année

Le traitement à effectuer :

```
import pandas as pd
import pycountry as pycountry
import plotly.express as px

df = df.toPandas()
df = df[df['title'] != ""] # Eliminer les articles nulles
df = df[df['authors_country'] != ""] # Eliminer les articles sans pays

x = df[['authors_country']]
y = pd.DataFrame({'authors_country' : x['authors_country']} \
    .apply(lambda x : pd.Series(list(set(x.split(';'))))).stack().tolist())
y['count'] = 0

countries_number = y.groupby("authors_country").count() \
    .sort_values(by=["count"], ascending = True).reset_index()
print(countries_number)
```

```
l = countries_number['authors_country'].values.tolist()

# Supprimer les espaces au début des pays récupérés
for i, c in enumerate(l):
    l[i] = c.strip()

for i, c in enumerate(l):
    countries_number['authors_country'][i] = l[i]

print(countries_number)

countries_number_dataframe = countries_number.values.tolist()
countries_number_df = pd.DataFrame(countries_number_dataframe)

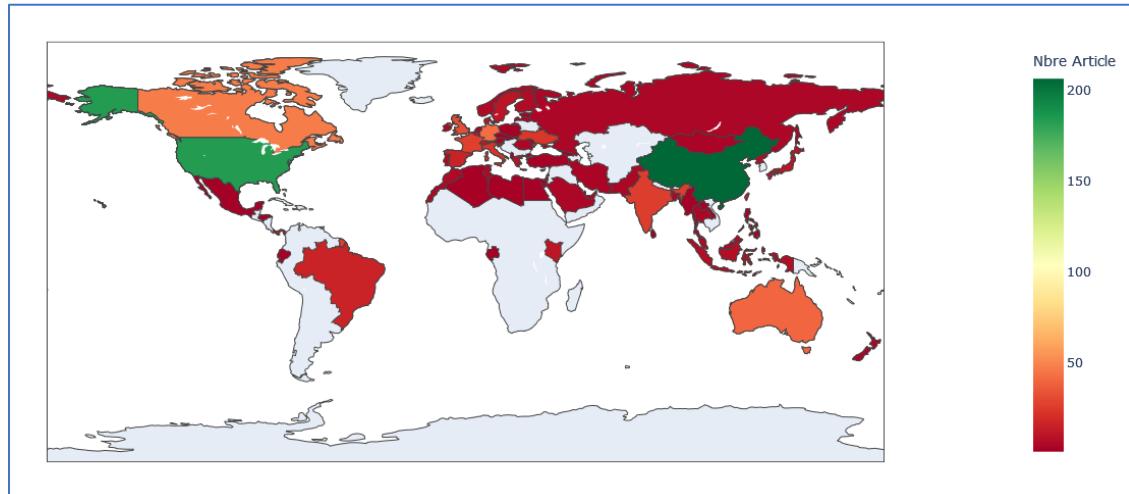
countries_number_df.columns = ['Country', 'Nbre Article']
print(countries_number_df)

countries_list = countries_number_df['Country'].unique().tolist()
print(countries_list)

countries_codes = {}
for country in countries_list:
    try :
        country_info = pycountry.countries.search_fuzzy(country)
        country_code = country_info[0].alpha_3
        countries_codes.update({country : country_code})
    except :
        print("Error: can't add this country's code => ", country)
        countries_codes.update({country : ' '})

for k, v in countries_codes.items():
    countries_number_df.loc[(countries_number_df.Country == k), 'iso_alpha'] = v

fig = px.choropleth(data_frame = countries_number_df,
                     locations = 'iso_alpha',
                     color = "Nbre Article",
                     hover_name = "Country",
                     color_continuous_scale = 'RdYlGn',
                     )
fig.show()
```

Résultat :Interprétation :

On observe que les pays publiant le plus d'articles scientifiques dans les 3 domaines cités précédemment par rapport aux autres sont la Chine, l'USA et Canada.

## Analyse et visualisation BI

### 1. Installation des outils

#### ○ PDI

PDI (Pentaho Data Integration), qui était auparavant connu sous le nom de Kettle, est un logiciel d'ETL (Extract, Transform, Load) Open Source qui permet la conception ainsi que l'exécution des opérations de manipulation et de transformation de données très complexes.



Son principal intérêt est de récupérer diverses sources dans divers formats, les traiter, les transformer, et former un résultat puis finalement exporter dans le format souhaité vers une destination souhaitée.

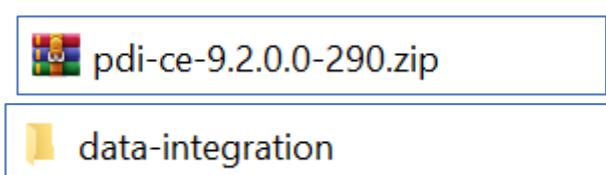
#### Installation de PDI

Téléchargement de Pentaho à partir du site web suivant :

<https://sourceforge.net/projects/pentaho/>

A screenshot of a web browser showing the SourceForge project page for "Pentaho from Hitachi Vantara". The page has a dark header with the SourceForge logo and navigation links for Help, Create, Join, and Login. A search bar is at the top right. The main content area features a banner for Microsoft Azure and an advertisement for a woman working on a laptop. Below this, the project title "Pentaho from Hitachi Vantara" is displayed with a red square logo containing the Hitachi Vantara branding. A brief description follows: "End to end data integration and analytics platform" and "Brought to you by: larrygrill, lcheng-pentaho, pedrovtelheira, pmgalves, and 2 others". There are reviews (5 stars, 67 reviews), download statistics ("Downloads: 6,213 This Week"), and a last update date ("Last Update: 2021-10-20"). At the bottom, there are download links for Windows, Mac, and Linux, along with "Get Updates" and "Share This" buttons. A sidebar on the right promotes "Get latest updates about Open Source Projects, Conferences and News."

Le fichier téléchargé :



On lance Pentaho via la commande « Spoon.bat » (dans le répertoire de Pentaho)

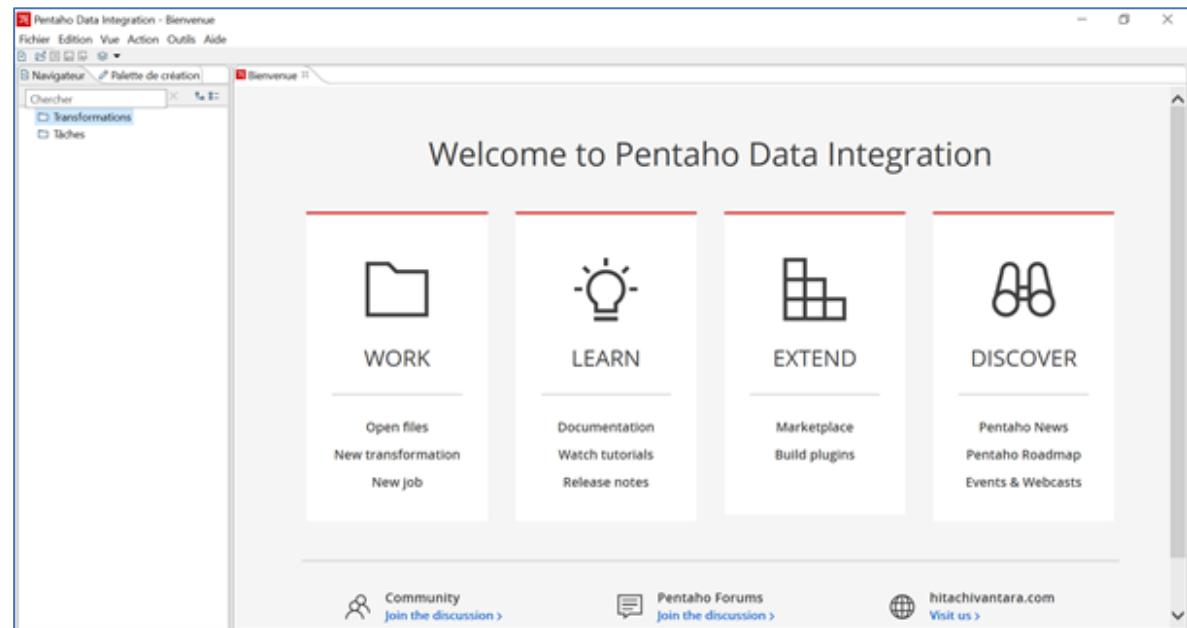


```
Sélection Administrateur : C:\Windows\System32\cmd.exe
Microsoft Windows [version 10.0.19043.1415]
(c) Microsoft Corporation. Tous droits réservés.

C:\data-integration>Spoon.bat
DEBUG: Using JAVA_HOME
DEBUG: _PENTAHO_JAVA_HOME=C:\Program Files\Java\jdk1.8.0_311
DEBUG: _PENTAHO_JAVA=C:\Program Files\Java\jdk1.8.0_311\bin\javaw.exe

C:\data-integration>start "Spoon" "C:\Program Files\Java\jdk1.8.0_311\bin\javaw.exe" -Xms1024m -Dhttps.protocols=TLSv1.1,TLSv1.2 "-Djava.library.path=libswt\win64;C:\winutils\bin" "-Djava.endorsed.dirs=C:\Program Files\Java\jdk1.8.0_311\jre\lib\endorsed;C:\Program Files\Java\jdk1.8.0_311\lib\endorsed;C:\data-integration\system\karaf\lib\endorsed" "-DKETTLE_HOME=" "-DKETTLE_REPOSITORY=" "-DKETTLE_USER=" "-DKETTLE_PASSWORD=" "-DKETTLE_PLUGIN_PACKAGES=" "-DKETTLE_LOG_SIZE_LIMIT" "-DKETTLE_JNDI_ROOT=" -jar launcher.launcher.jar -lib ..\libswt\win64

C:\data-integration>
```



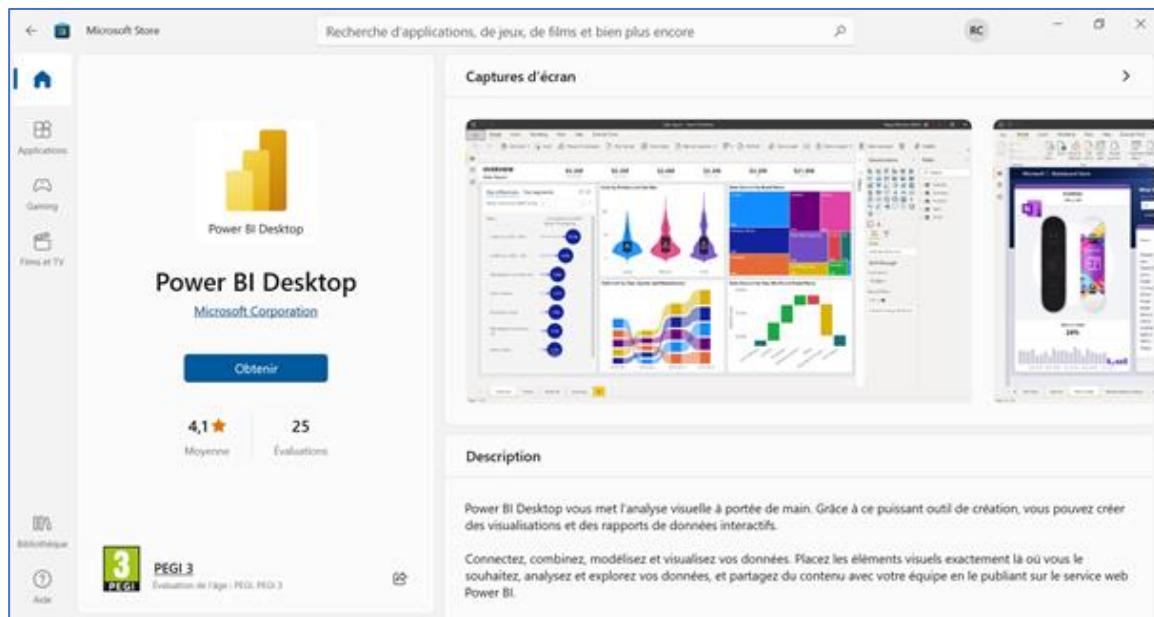
## ○ Power BI

Microsoft Power BI est un ensemble d'outils et d'applications, qui permettent de traiter et d'explorer un important volume de données. La solution, leader du marché, donne la possibilité de visualiser des insights de manière dynamique et d'interagir entre les graphiques. L'objectif : obtenir des analyses avancées pour nous aider à prendre des décisions stratégiques pour au sein d'une entreprise.

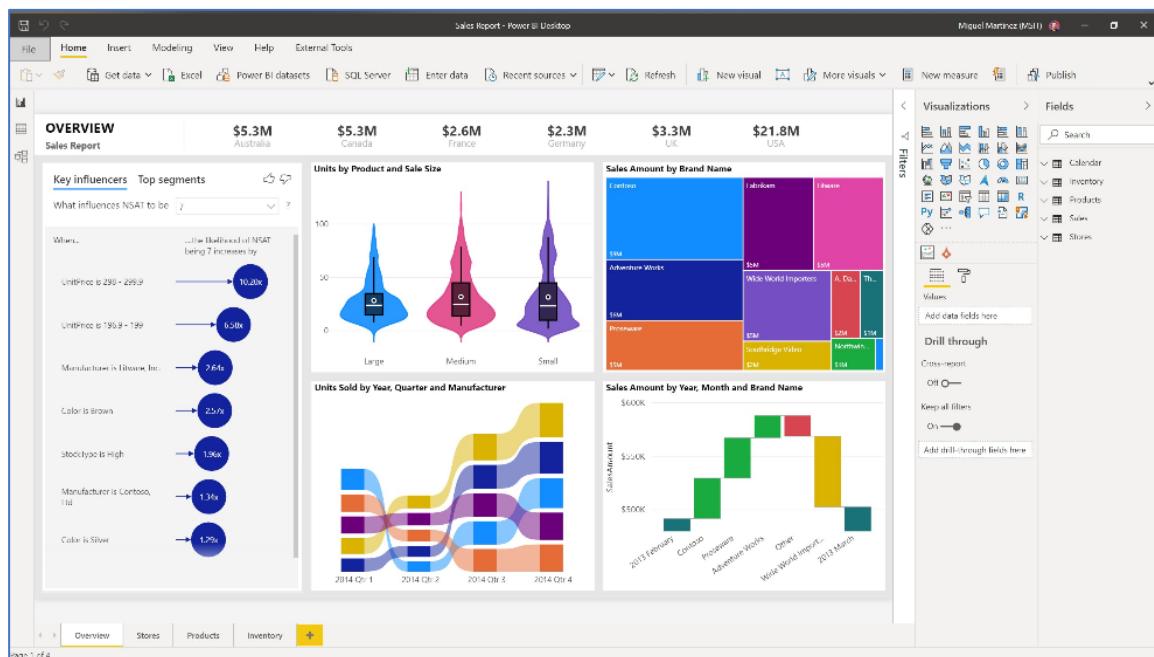


### Installation de Power BI :

On télécharge Power BI Desktop depuis le Microsoft Store :



Puis on le lance :

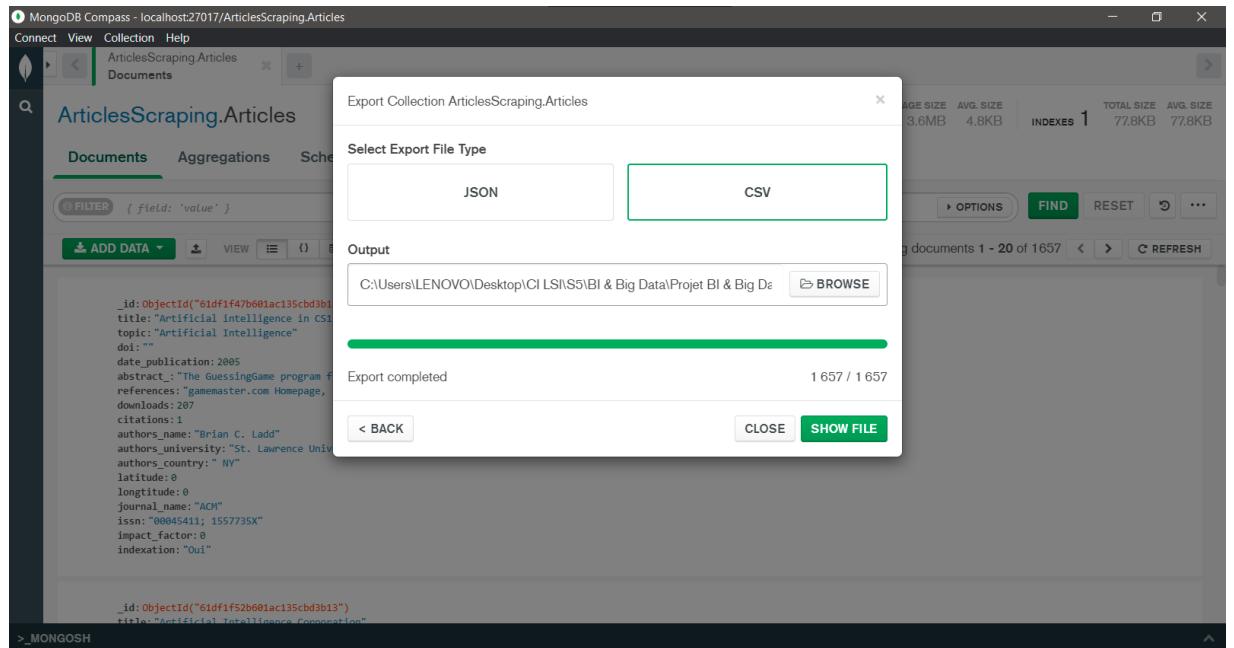


## 2. Schéma en étoile

Le schéma en étoile est le paradigme de modélisation simple et commun dans lequel l'entrepôt de données comprend une table de faits avec une seule table pour chaque dimension. Le schéma imite une étoile, avec une table de dimension présentée dans un motif étalé entourant la table de faits centrale. La table des

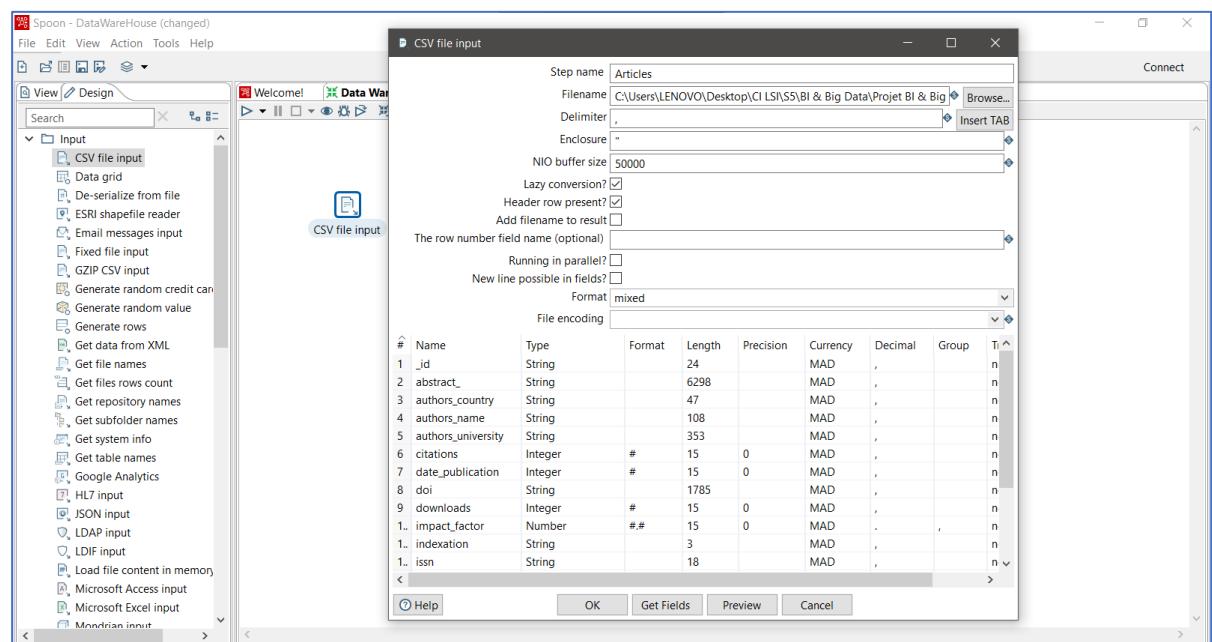
dimensions en fait est connectée à la table de dimension via la clé primaire et la clé étrangère.

## ○ Exportation des données sous format CSV :



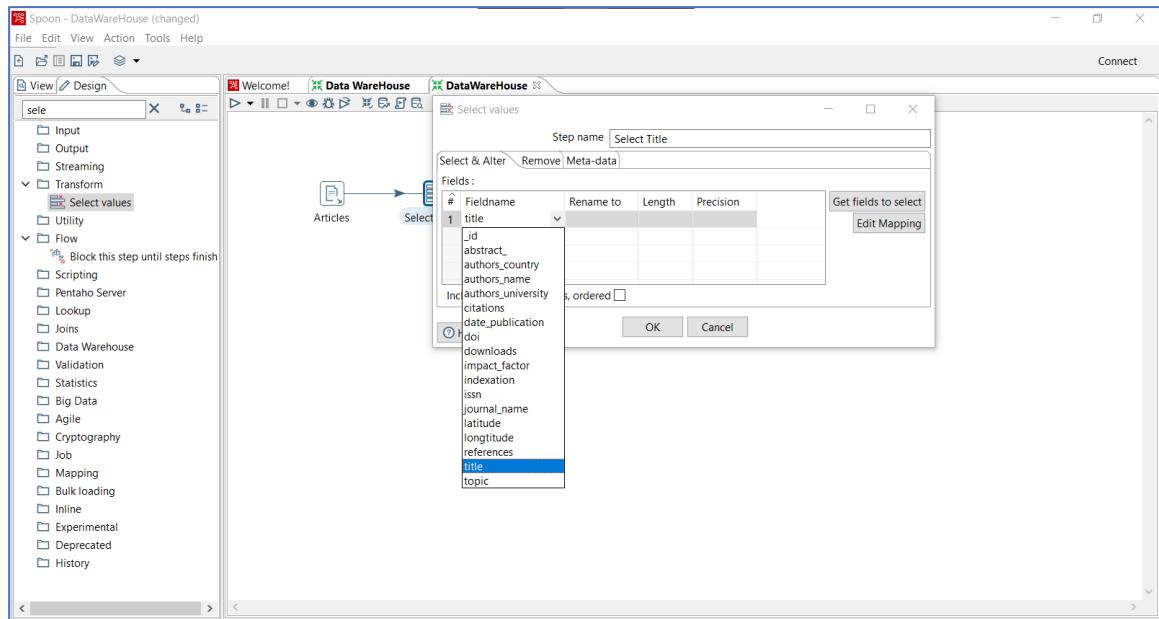
## ○ Génération des dimensions (exemple titre) :

Importation de la base de données depuis le fichier CSV

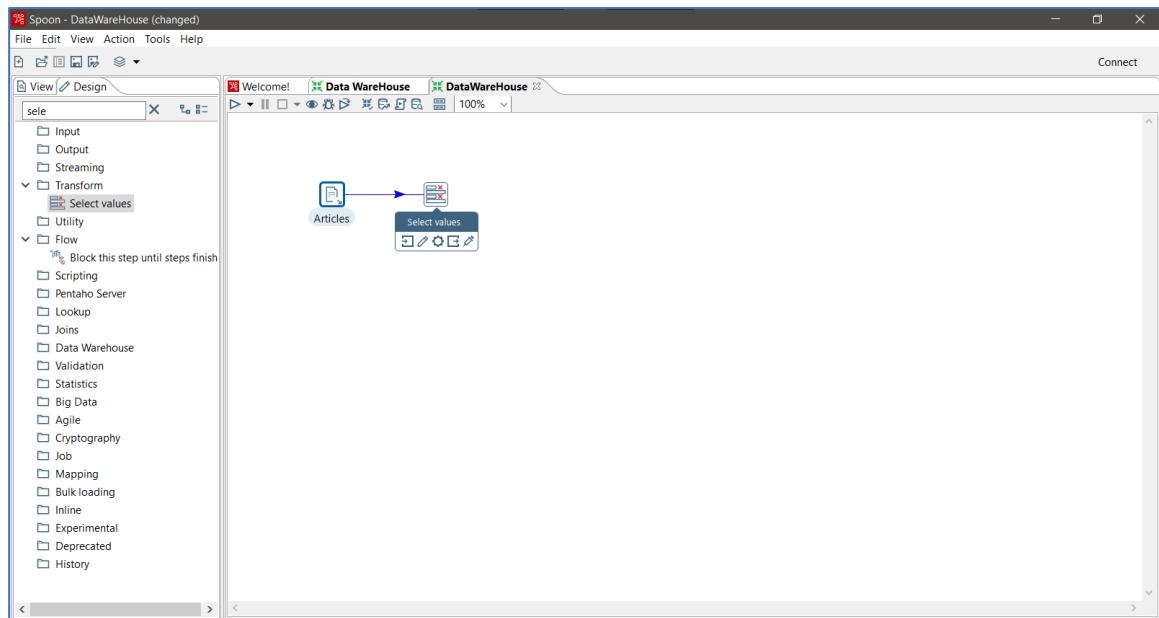


## O Création des composants

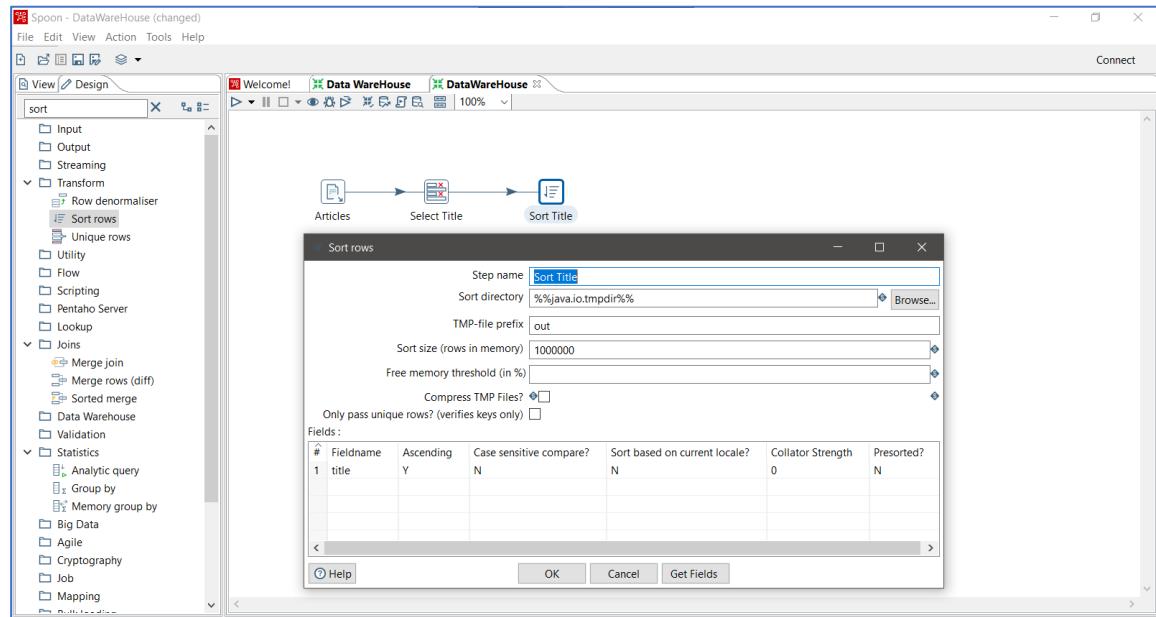
### Select Values :



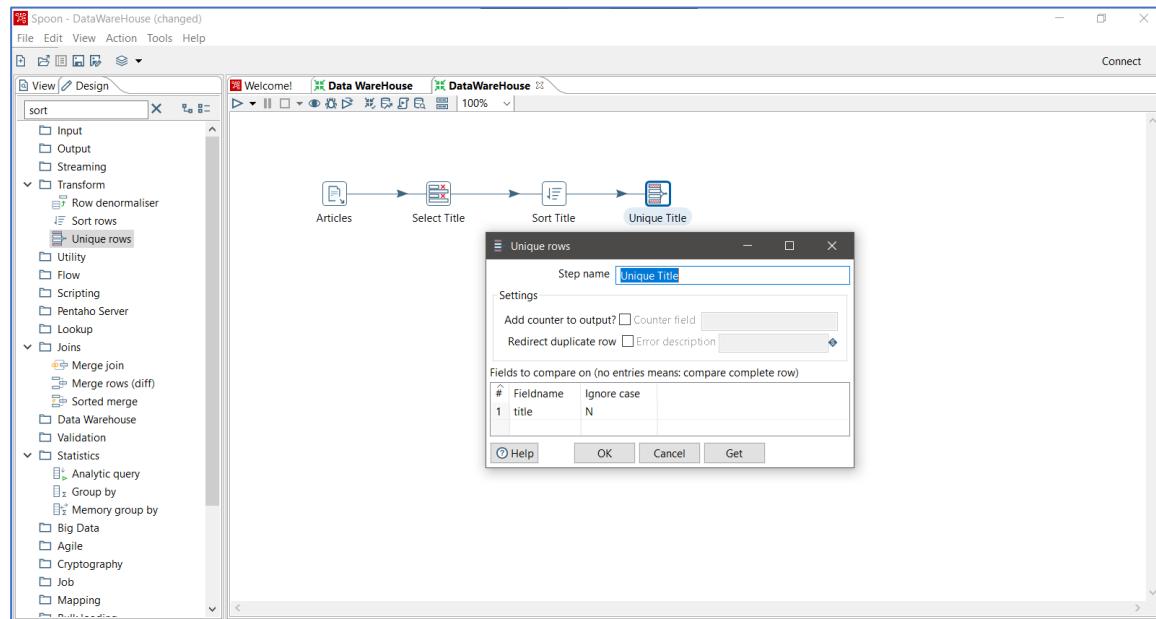
Liaison entre l'input (données) et l'output



## Sort Rows :

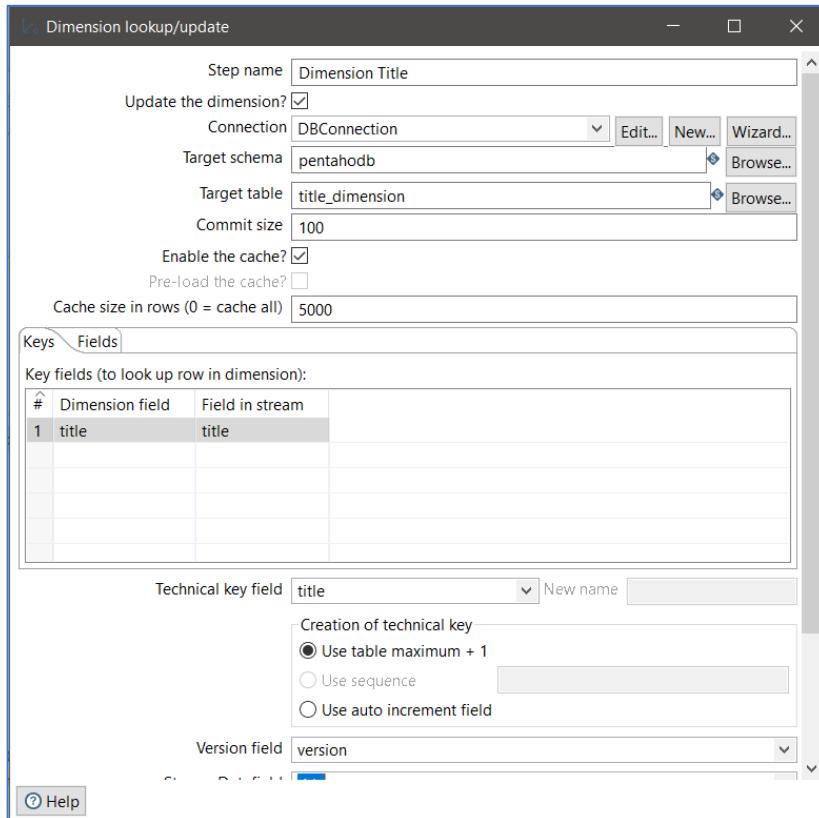


## Unique Rows :

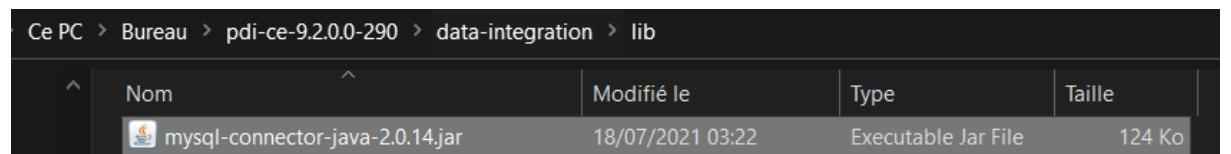


## ○ Chargement dans la base de données

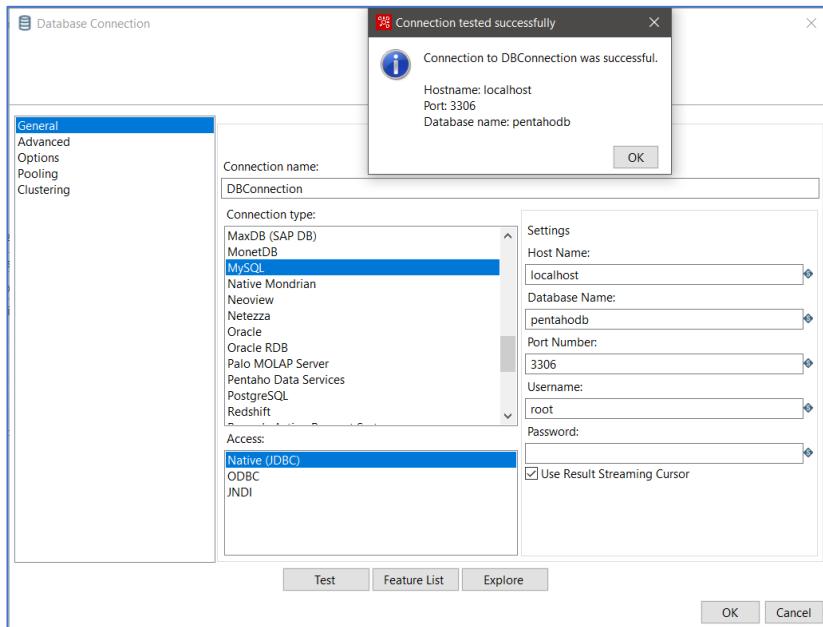
On ajoute la composante « Dimension Lookup/update » pour effectuer ce chargement.



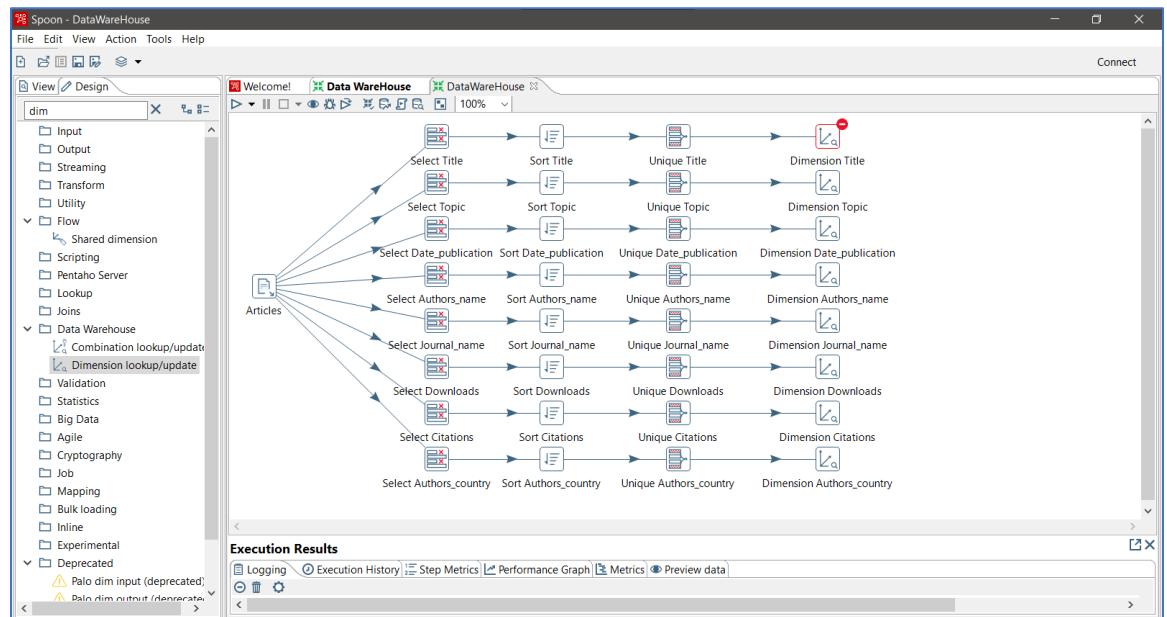
Création d'une connexion à la base de données MySQL : cela nécessite avoir le connecteur mysql dans le chemin suivant :



On indique les paramètres de connexion :



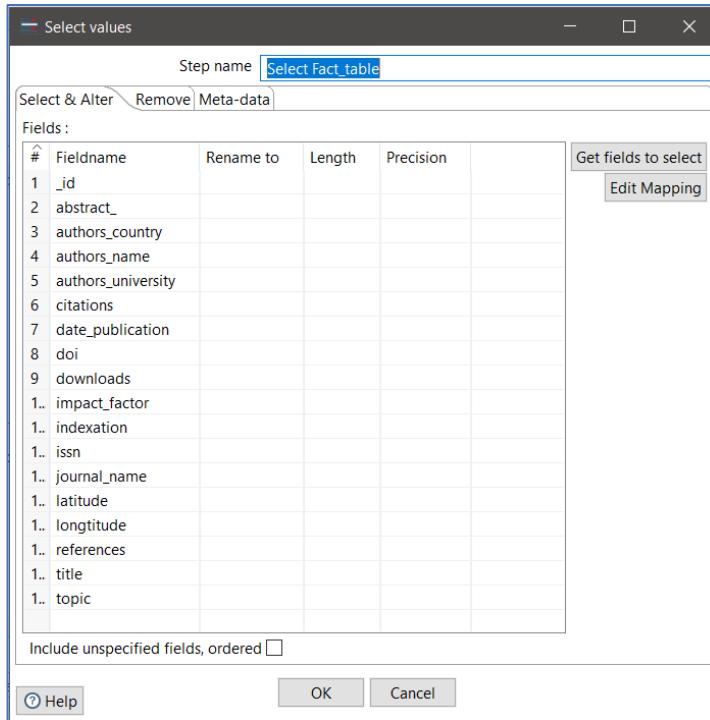
## ○ Exécution des transformations



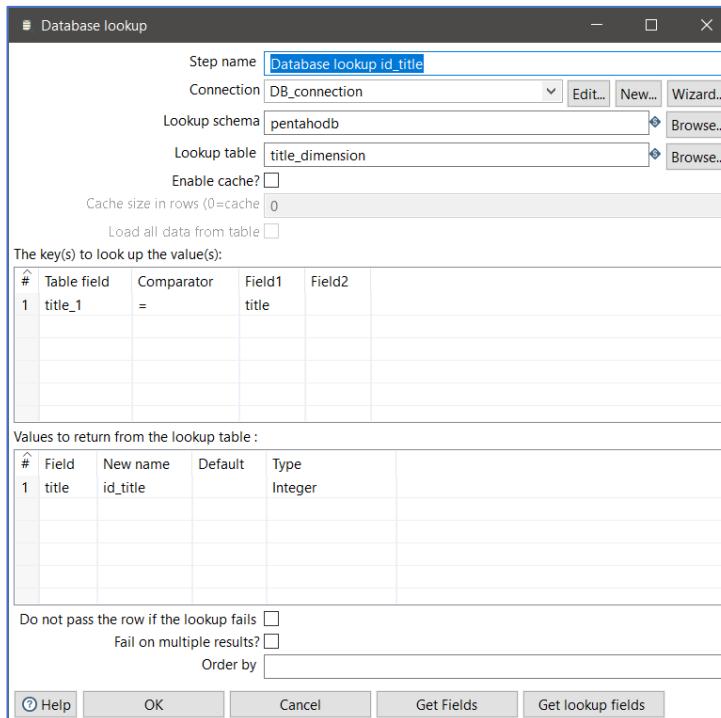
## ○ Table de fait

### Select values

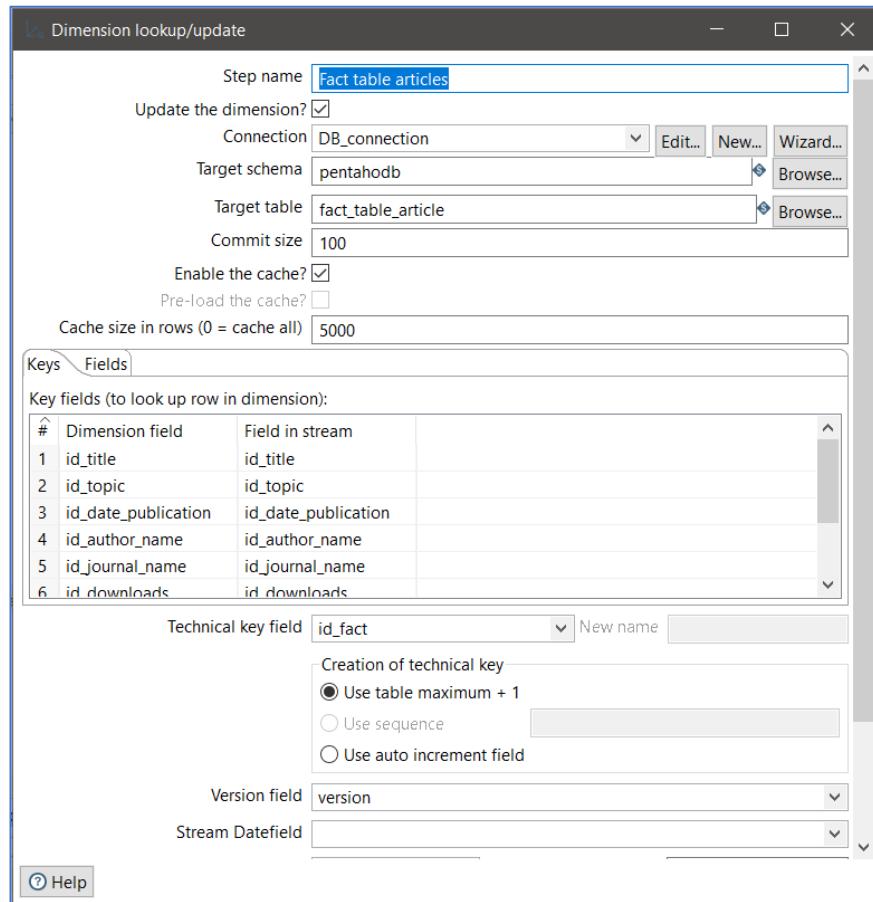
On va sélectionner toutes les valeurs :



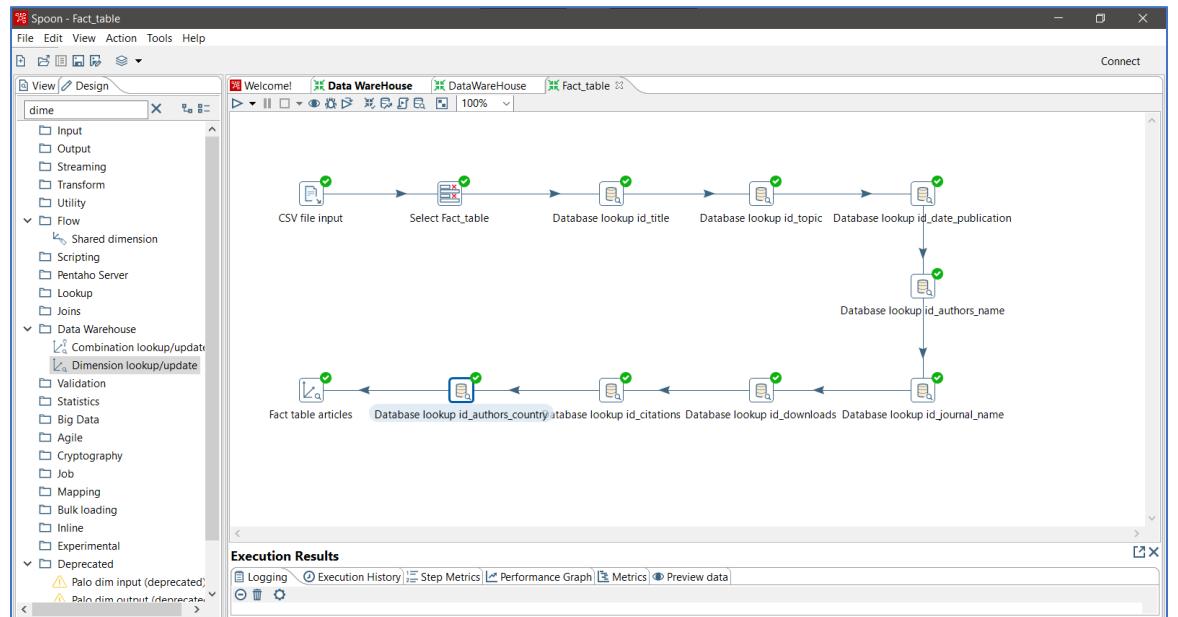
### Recherche id (exemple de l'attribut titre)



## Dimension lookup/update



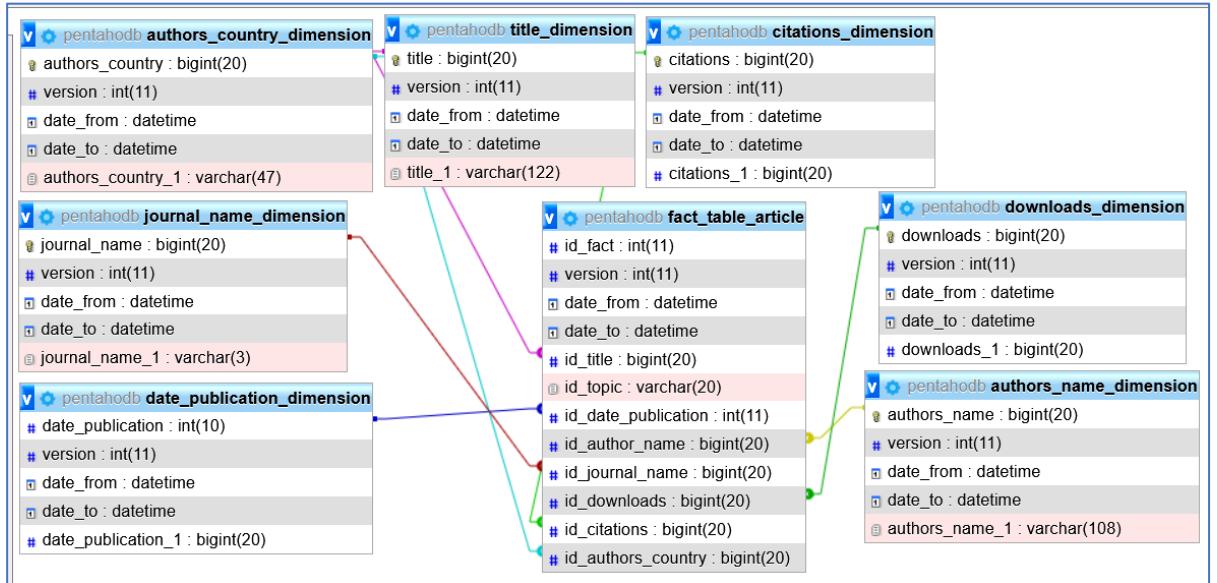
## ○ Exécution de la transformation de la table de fait



## ○ Résultats dans MySql

Table	Action	Lignes	Type	Interclassement	Taille
authors_country_dimension	Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_general_ci	48,0
authors_name_dimension	Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_general_ci	48,0
citations_dimension	Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_general_ci	48,0
date_publication_dimension	Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_general_ci	48,0
downloads_dimension	Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_general_ci	48,0
fact_table_article	Parcourir Structure Rechercher Insérer Vider Supprimer	1 658	InnoDB	utf8mb4_general_ci	208,0
journal_name_dimension	Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_general_ci	48,0
title_dimension	Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_general_ci	48,0
topic_dimension	Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_general_ci	48,0
9 tables	Somme	1 658	InnoDB	utf8mb4_general_ci	592,0

## ○ Schéma en étoile



### 3. Visualisation et Reporting par Power BI

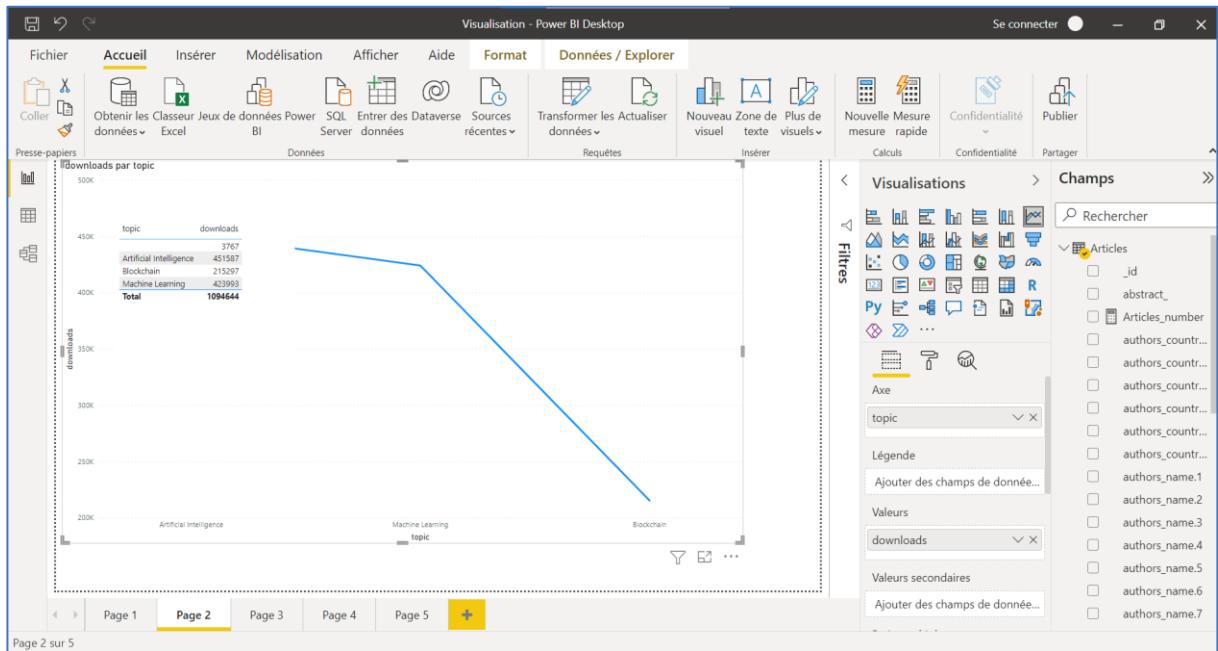
On exporte les données sous forme de fichier .csv, puis on l'importe dans Power BI :

The screenshots illustrate the process of importing a CSV file into Power BI. In the first screenshot, the 'Accueil' tab is active, and the 'Sources de données communes' pane is open, showing various data source options. A tooltip for 'Importez des données à partir d'un fichier texte ou CSV.' is visible. In the second screenshot, the 'Accueil' tab is active, and an 'Ouvrir' (Open) dialog box is displayed, showing a file named 'Articles.csv' selected from a folder path.

Après l'importation des données, on clique sur « Transformer les données » :

Puis on sélectionne la colonne pour la fractionner, afin d'enlever les « ; » entre les attributs de la colonne :

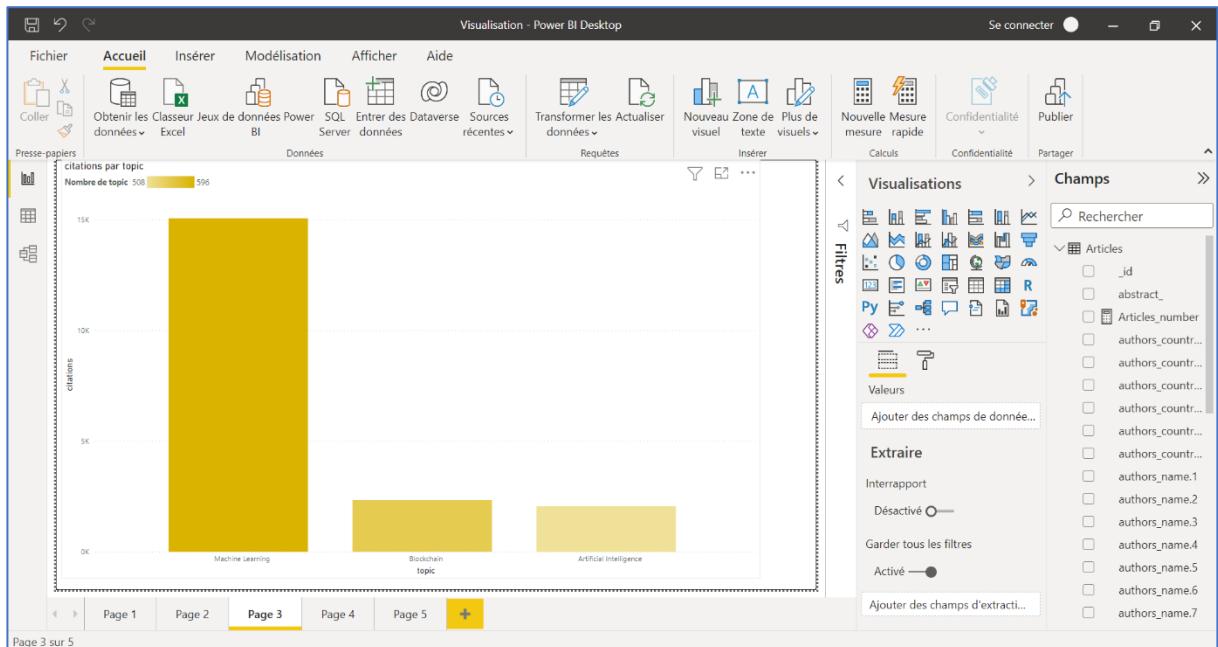
## 1. Nombre de téléchargements par sujet :



D'après le graphique, les articles de l'intelligence artificielle sont les plus téléchargés.

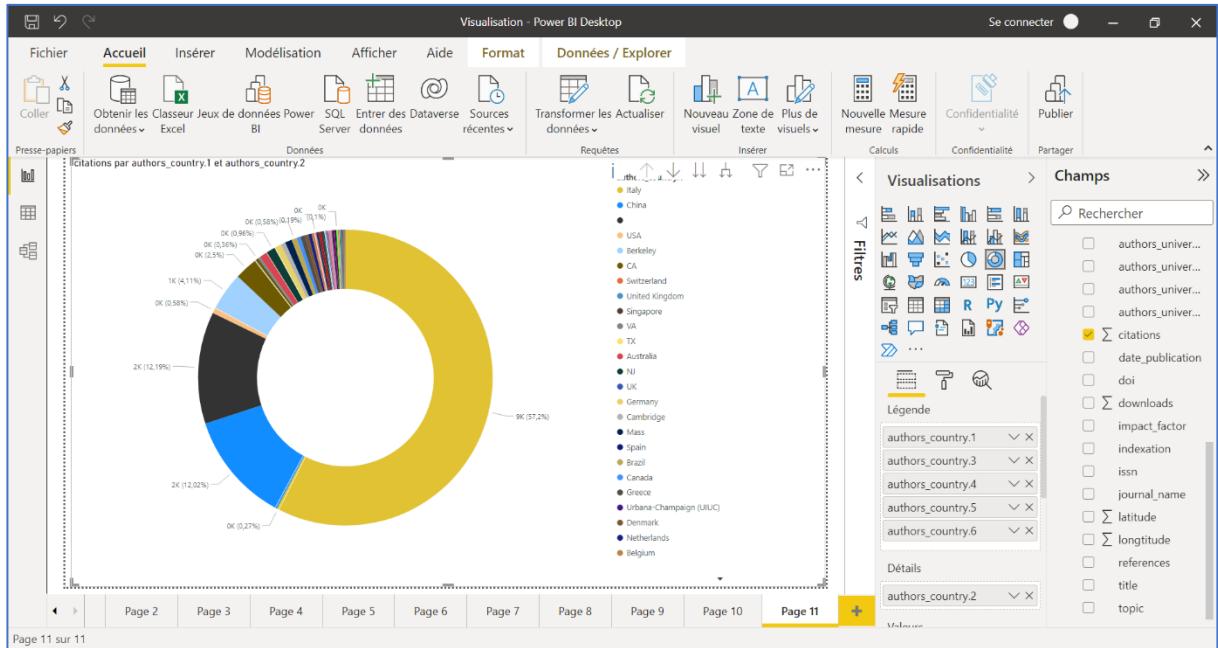
## 2. Nombre de citations par sujet :

Une citation est la reproduction d'un court extrait d'un propos ou d'un écrit antérieur dans la rédaction d'un texte ou dans une forme d'expression orale. Elle peut s'inscrire dans une référence.



Dans ce graphe, on constate que les articles les plus cités sont ceux de la machine learning, car l'intelligence artificielle et blockchain ont connu une explosion via le machine learning.

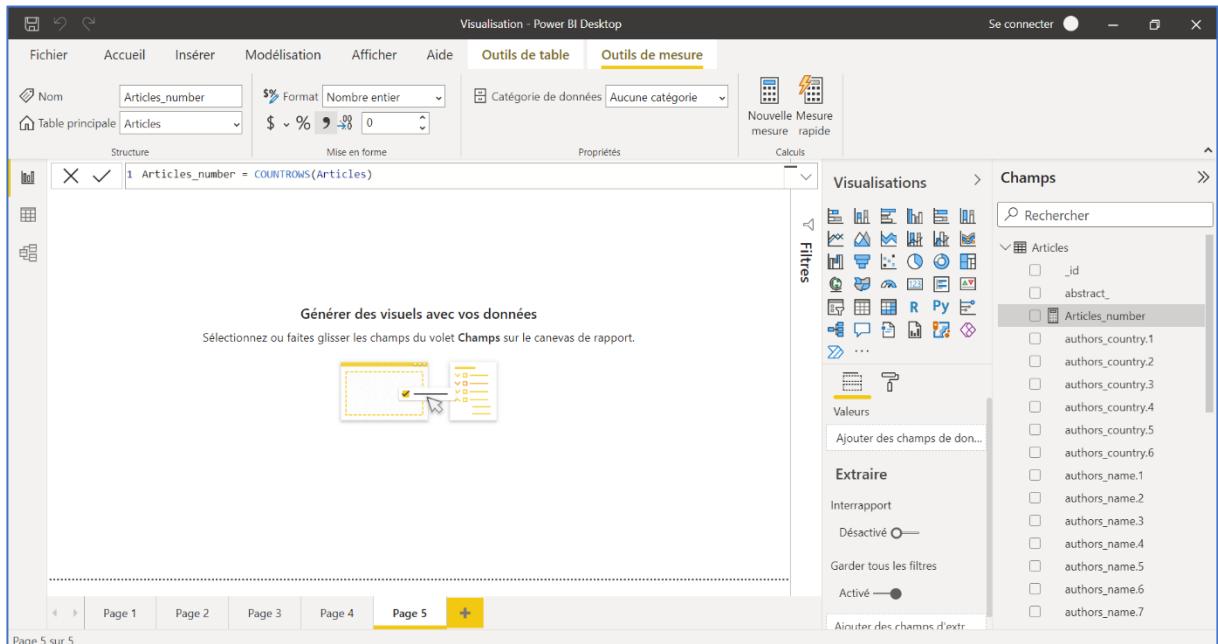
### 3. Nombre de citations par pays :



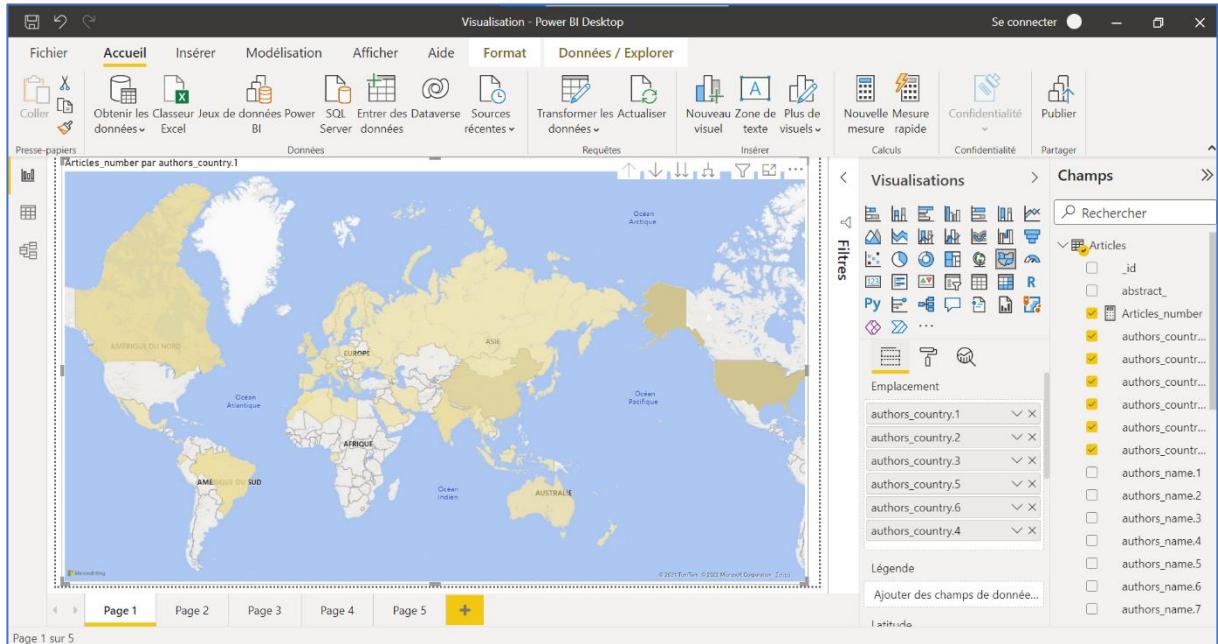
Les articles de l'Italie sont plus cités par rapport aux autres pays.

### 4. Nombre d'articles par pays :

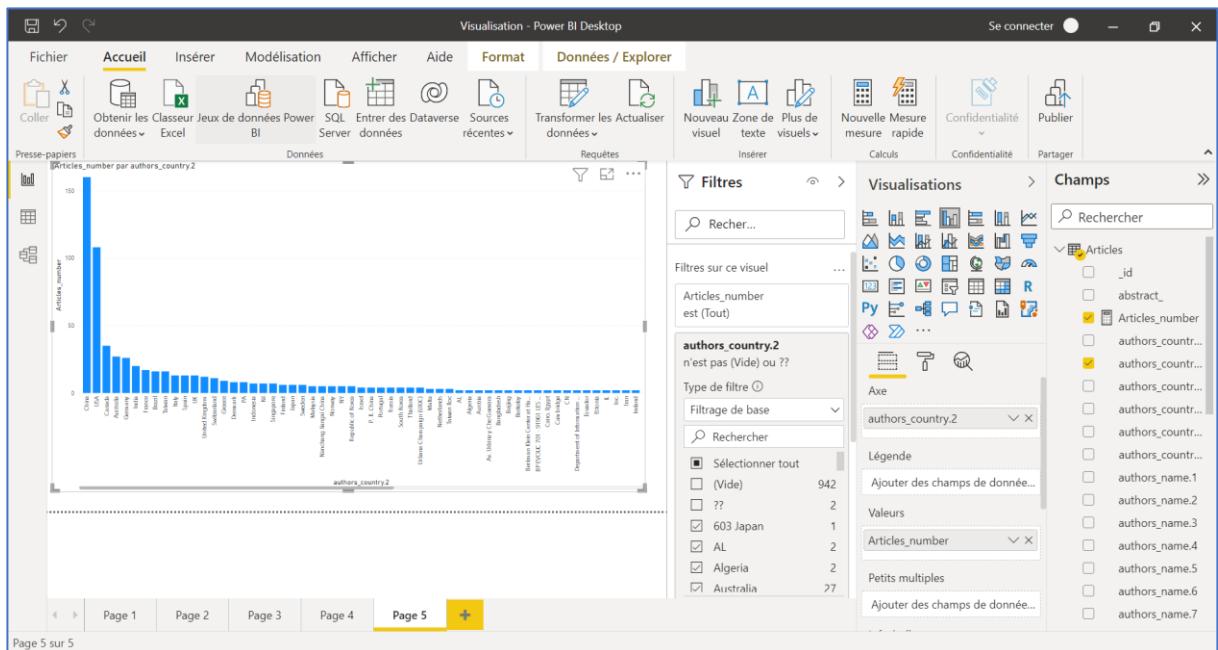
Pour cela, on a créé une mesure qui compte le nombre d'articles par pays :



On ajoute une carte choroplèthe pour la visualisation du nombre d'articles publiés par les pays qu'on a dans la base de données :



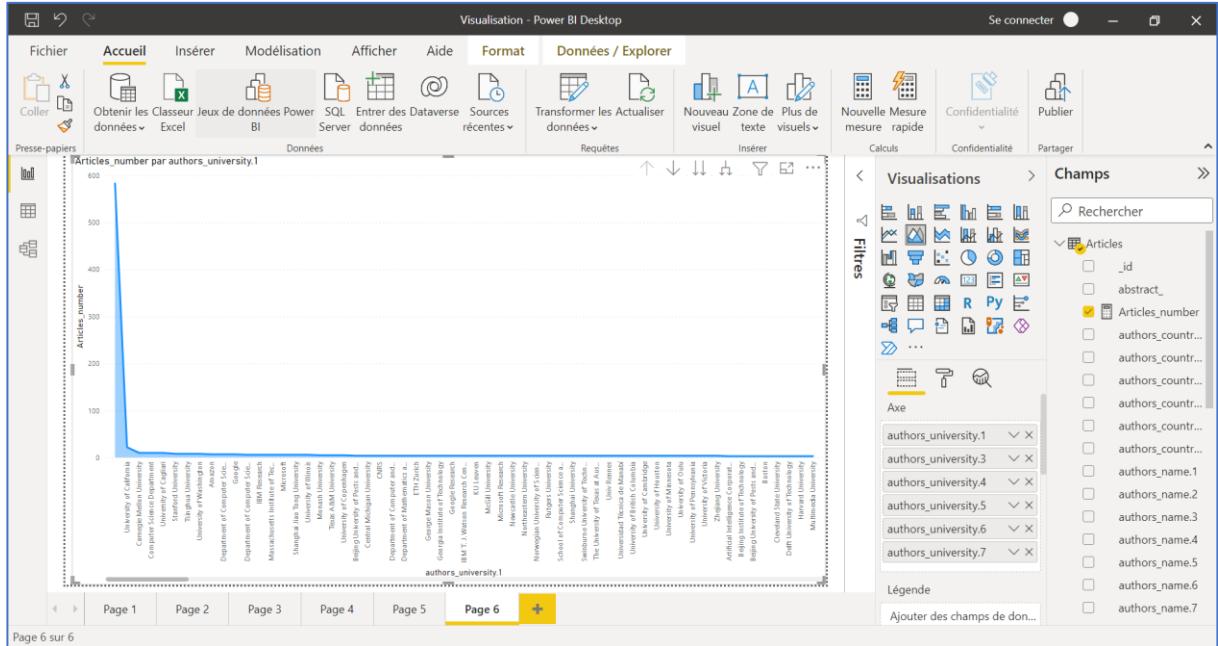
La représentation par des histogrammes :



On constate que les pays qui publient un grand nombre d'articles scientifiques sont : China, USA et Canada.

En effet, La Chine est sur le point de devenir une superpuissance scientifique grâce au financement public massif de la recherche et développement (R & D) et au très grand nombre de ses chercheurs en science et technologie.

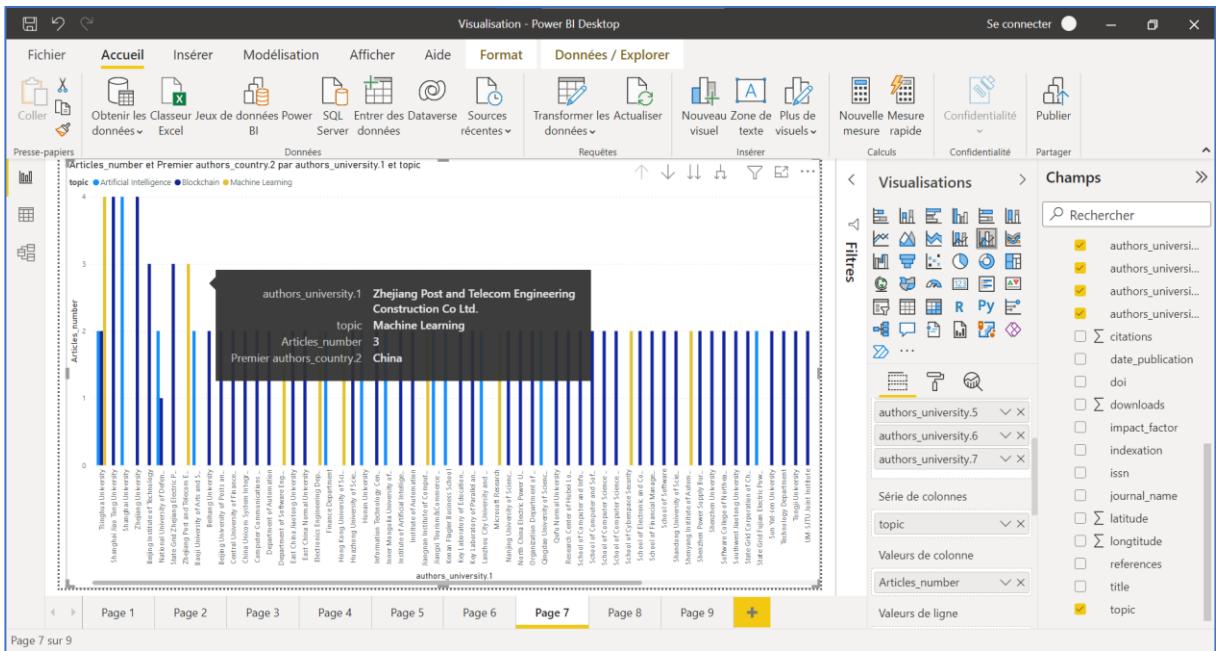
## 5. Nombre d'articles par université :



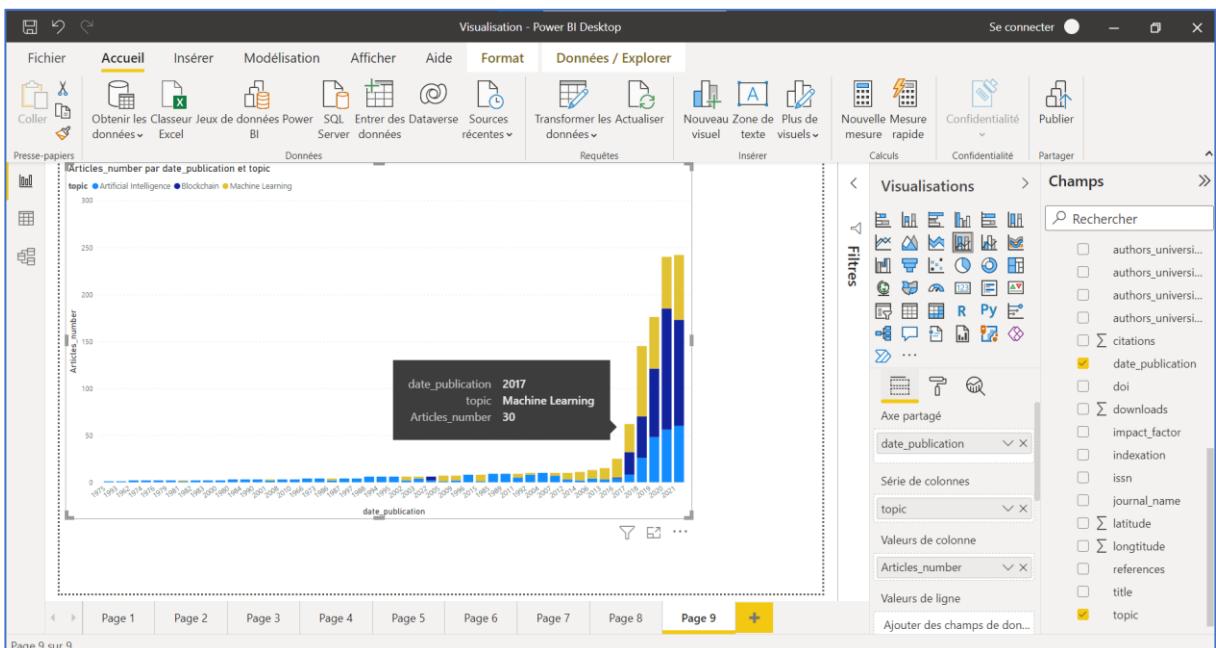
D'après le graphique, la plupart des auteurs sont de l'université « University of California ».

## 6. Nombre d'article par sujet dans les universités de la chine :

Le graphe ci-dessous, montre le nombre d'articles publiés dans les 3 sujets : « Artificial Intelligence », « Blockchain » et « Machine Learning » dans chaque université dans la chine.



## 7. Nombre d'articles dans chaque sujet par année :



Ce graphe montre le nombre d'articles publiés dans chaque année par tous les sujets.

On remarque que le sujet de blockchain a connu une croissance remarquable à partir de l'année 2017 car La blockchain est actuellement l'une des technologies dont on parle le plus dans le monde des affaires. La technologie Blockchain a le potentiel de provoquer des changements majeurs et de créer de nouvelles

opportunités dans tous les secteurs : de la banque et de la cybersécurité à la propriété intellectuelle et aux soins de santé.

## Versions des outils utilisés

### ○ Version de MongoDB

Version 1.30.1

### ○ Version de Spark

Spark version 3.0.3 avec Hadoop version 2.7

Cette version de Spark est compatible avec jdk version 13

### ○ Version du connecteur MongoDB et Spark

Version 2.12-2.4.2

[https://mvnrepository.com/artifact/org.mongodb.spark/mongo-spark-connector\\_2.12/2.4.2](https://mvnrepository.com/artifact/org.mongodb.spark/mongo-spark-connector_2.12/2.4.2)

### ○ Version de PDI

Version 9.2.0 compatible avec la version 1.8 du jdk

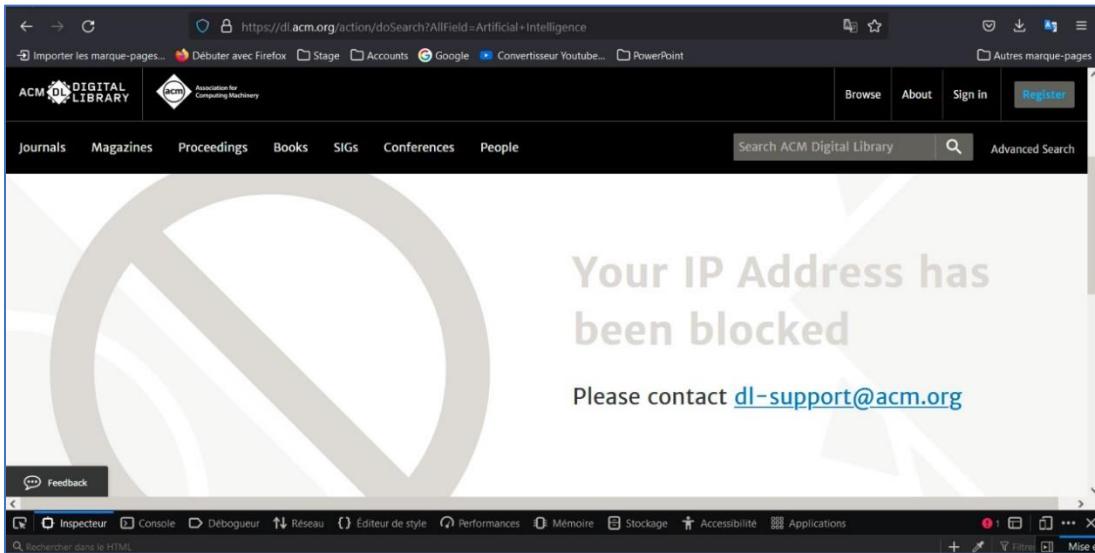
### ○ Version du connecteur MySql

Version 8.0.27

<https://jar-download.com/artifacts/mysql/mysql-connector-java>

## Problèmes rencontrés

On a eu des problème au mement du scrapping car notre adresse ip est blockée.



Spark n'a fonctionné qu'avec la version 13 du JDK.

L'extraction des données depuis IEEE nécessite l'utilisation de l'api.

## Conclusion

Avec sa diffusion, la Business Intelligence est devenue un mécanisme clé du fonctionnement de l'entreprise. Elle combine l'analyse commerciale, l'exploration de données, la visualisation de données, les outils et l'infrastructure de données et les meilleures pratiques pour aider les organisations à prendre des décisions davantage basées sur les données. Ce qui nous permet d'assurer une veille économique moderne lorsque nous avons une vue complète des données de l'organisation pour conduire le changement, éliminer les inefficacités et de s'adapter rapidement aux changements du marché ou de l'approvisionnement.

## Références

- <https://aws.amazon.com/fr/elasticmapreduce/details/spark/>
- <https://blog.ippon.fr/2015/01/13/introduction-a-spark-sql/>
- <https://sparkbyexamples.com/pyspark-tutorial/>
- <https://www.journaldunet.fr/web-tech/guide-de-l-entreprise-digitale/1146290-docker-definition-docker-compose-docker-hub-docker-swarm-160919/>
- <https://www.oracle.com/fr/database/comment-creer-base-donnees-mysql.html>
- <https://ichi.pro/fr/tutoriel-scrapy-comment-faire-un-web-crawler-a-l-aide-de-scrapy-94026790754555>
- <https://corporatefinanceinstitute.com/resources/knowledge/other/business-intelligence/>
- <https://www.talend.com/fr/resources/guide-business-intelligence/>
- <https://www.zyte.com/blog/handling-javascript-in-scrapy-with-splash/>
- <https://www.blogdumoderateur.com/tools/microsoft-power-bi/>
- [https://fr.wikipedia.org/wiki/Citation\\_\(litt%C3%A9rature\)#:~:text=Une%20citation%20est%20la%20reproduction,s'inscrire%20dans%20une%20r%C3%A9f%C3%A9rence.](https://fr.wikipedia.org/wiki/Citation_(litt%C3%A9rature)#:~:text=Une%20citation%20est%20la%20reproduction,s'inscrire%20dans%20une%20r%C3%A9f%C3%A9rence.)
- <https://waytolearnx.com/2018/08/difference-entre-le-schema-en-etoile-et-en-flocon.html>