Projet Data Integration

CORMEAUX Anthony, NANJI ENGA Gedeon Freddy, IBOBI Osée

Compréhension des données

Les données avec lesquelles nous allons travailler viennent du National Student Loan Data System

« The National Student Loan Data System (NSLDS) is the national database of information about loans and grants awarded to students under Title IV of the Higher Education Act (HEA) of 1965. NSLDS provides a centralized, integrated view of Title IV loans and grants during their complete life cycle, from aid approval through disbursement, repayment, deferment, delinquency, and closure."

https://catalog.data.gov/dataset/national-student-loan-data-system-722b0

FL_DASHBOARD

Les données qui viendront de HDFS sont les FL Dashboard, la description de chaque champ nous est donnée

Field Name	Definition
OPE ID	An 8-digit code identifying the school at its main branch
School	The name of the school associated with the OPE ID
State	The state in which the main campus is located
Zip Code	The zip code of the main campus
School Type	Indicates the control or ownership of the school.
Recipients	The number of loan recipients for the loan type during the award year for the time period reported on the spreadsheet. For Subsidized, Unsubsidized, and Graduate PLUS loans, this is a count of student borrowers. For Parent PLUS loans, this is a count of the students on whose behalf the loan was taken. Since students can have multiple loan types in the same award year, you cannot sum the recipient counts from the four categories to obtain an accurate count of total recipients for the loan program during that award year.
# of Loans Originated	The number of loans initiated for the loan type during the award year for the time period reported on the spreadsheet.
$ of Loans Originated	The dollar amount of the loans initiated for the loan type during the award year for the time period reported on the spreadsheet. This is the expected total loan amount if the loan is fully disbursed.
# of Loans Disbursed	The number of disbursements made for the loan type during the award year and quarter reported on the spreadsheet.
$ of Loans Disbursed	The dollar amount of disbursements made for the loan type during the award year for the time period reported on the spreadsheet.

Aperçu des données brutes

2009-2010 Award Year FFEL Volume by School

Award Year Quarterly Activity (07/01/2009-09/30/2009)

Data Run: 4/5/2012

					FFEL SUBSIDIZED					FFEL UNSUBSIDIZED				
OPE ID	School	State	Zip Code	School Type	Recipients	# of Loans Originated	$ of Loans Originated	# of Disburse ments	$ of Disbursements	Recipients	# of Loans Originated	$ of Loans Originated	# of Disbursements	$ of Disbursements
00106100	ALASKA PACIFIC UNIVERSITY	AK	995084672	PRIVATE	291	291	$ 1 546 994,00	292	$ 830 513,00	267	271	$ 1 674 604,00	271	$ 896 614,00
00106300	UNIVERSITY OF ALASKA FAIRBANKS	AK	997757500	PUBLIC	1 413	1 434	$ 6 394 735,00	1 455	$ 3 290 699,00	1 472	1 521	$ 8 806 921,00	1 542	$ 4 472 224,00
00106500	UNIVERSITY OF ALASKA SOUTHEAST	AK	998018680	PUBLIC	406	409	$ 1 866 473,00	439	$ 1 044 946,00	437	448	$ 2 925 528,00	478	$ 1 558 767,00
01146200	UNIVERSITY OF ALASKA ANCHORAGE	AK	995088050	PUBLIC	2 998	3 042	$ 12 780 036,00	3 045	$ 6 440 086,00	3 343	3 432	$ 20 274 742,00	3 435	$ 10 219 443,00
02541000	ALASKA CAREER COLLEGE	AK	995071033	PROPRIETARY	38	38	$ 103 869,00	38	$ 52 178,00	37	37	$ 134 553,00	37	$ 67 331,00
02576900	CHARTER COLLEGE	AK	995084103	PROPRIETARY	192	192	$ 516 838,00	193	$ 255 514,00	206	322	$ 653 151,00	328	$ 298 655,00

Données en streaming

Les données qui viendront de Kafka nous sont données dans un fichier Excel.

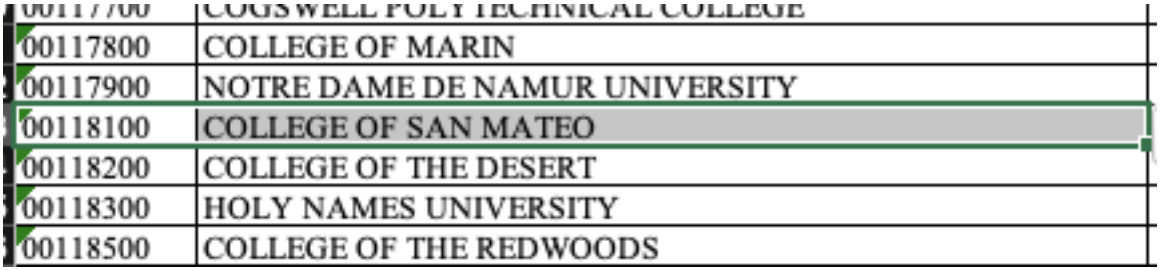
Ce dataset comprend des détails tels que le code de l'école, le nom, l'adresse, la ville, l'état, le code postal, la province, le pays et le code postal.

SchoolCode	SchoolName	Address	City	StateCode	ZipCode	Province	Country	PostalCode
B04724	WIDENER UNIV SCHOOL OF LAW - DE	4601 CONCORD PIKE/PO BOX 7474	WILMINGTON	DE	19803			
B06171	CENTER FOR ADVANCED STUDIES OF PUER	BOX S-4467	SAN JUAN	PR	00902			
B06511	PENTECOSTAL THEOLOGICAL SEMINARY	PO BOX 3330	CLEVELAND	TN	37320			
B07022	THE CHICAGO SCHOOL OF PROF PSYCHOLOGY	325 NORTH WELLS STREET	CHICAGO	IL	60610			
B07624	NATIONAL COLLEGE OF NATURAL MEDICINE	049 SW PORTER	PORTLAND	OR	97201			
B07625	OREGON COL OF ORIENTAL MEDICINE	10525 SE CHERRY BLOSSOM DR	PORTLAND	OR	97216			
B08041	ALFRED ADLER GRADUATE SCHOOL	1001 WEST HIGHWAY 7 SUITE 344	HOPKINS	MN	55305			
B08083	UNIV OF THE DIST OF COLU -SCHOOL OF LAW	4200 CONNECTICUT AVENUE NW	WASHINGTON	DC	20008			
B42154	GRACE SCHOOL OF THEOLOGY	3705 COLLEGE PARK DR	CONROE	TX	77384			
E00014	AMRCN REPERTORY THTR INST ADV THTR	64 BRATTLE STREET	CAMBRIDGE	MA	02138			
E00058	BROWN UNIVERSITY - GRADUATE SCHOOL	8 FONES ALLEY	PROVIDENCE	RI	02912			

Lien entre les données

Les données en streaming qui arriveront en streaming contiennent des informations sur les écoles qui ne font pas parti des fichiers FL_DASHBOARD et qui serviront donc à enrichir nos données.

Le lien entre les données se fera grâce à la colonne OPE ID des FL_DASHBOARD et la colonne SchoolCode des données en streaming. Le SchoolCode correspond aux 6 premiers caractères de l'OP ID.



00117700	COGSWELL POLYTECHNICAL COLLEGE
00117800	COLLEGE OF MARIN
00117900	NOTRE DAME DE NAMUR UNIVERSITY
00118100	COLLEGE OF SAN MATEO
00118200	COLLEGE OF THE DESERT
00118300	HOLY NAMES UNIVERSITY
00118500	COLLEGE OF THE REDWOODS

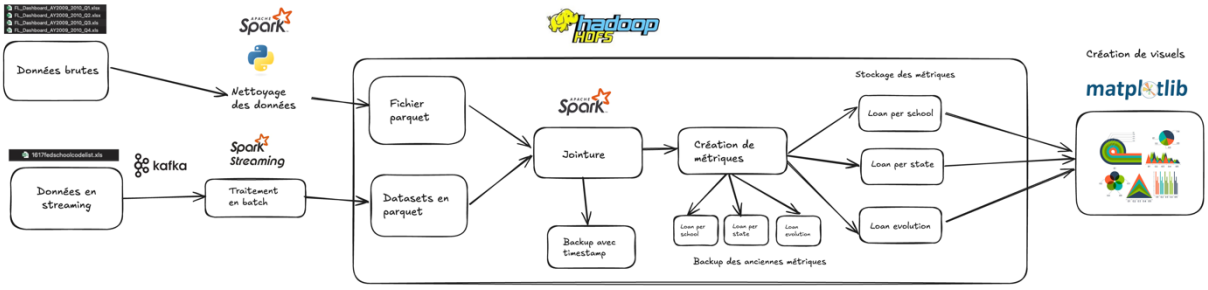
001178	COLLEGE OF MARIN	835 COLLEGE AVE	KENTFIELD	CA	94904
001179	NOTRE DAME DE NAMUR UNIVERSITY	1500 RALSTON AVE	BELMONT	CA	94002
001181	COLLEGE OF SAN MATEO	W HILLSDALE BLVD	SAN MATEO	CA	94402
001182	COLLEGE OF THE DESERT	43-500 MONTEREY AVENUE	PALM DESERT	CA	92260
001183	HOLY NAMES UNIVERSITY	3500 MOUNTAIN BLVD	OAKLAND	CA	94619
001185	COLLEGE OF THE REDWOODS	7351 TOMPKINS HILL ROAD	EUREKA	CA	95501

Création du processus de Data Integration

Nous allons suivre les étapes suivantes :

- Nettoyer les données brutes des FL_DASHBOARD
- Les combiner et créer un fichier parquet puis le stocker sur HDFS
- Envoyer les données streaming sur un topic Kafka
- Les consommer avec Spark Sctuctured Streaming par batch de 100 toutes les 10 secondes et écrire un nouveau Dataset à chaque Batch
- Effectuer une jointure entre le fichier parquet et les dataset pour écrire un nouveau fichier en sauvegardant l'ancien dans un dossier backup
- Créer des métriques à partir de ce nouveau fichier et stocker les métriques dans des tables
- Utiliser ces métriques dans des visualisations

Schéma du projet



Technologies utilisées :

- Spark
- Kafka
- Spark Structured Streaming

Nettoyage des données FL_DASHBOARD

Le script Python utilise PySpark, il est conçu pour automatiser le nettoyage et la préparation des données provenant des fichiers des tableaux de bords.

Résumé des actions effectuées par le code :

Extraction des informations clés :

- Pour chaque fichier de données FFEL, le script extrait les dates de début et de fin du trimestre financier concerné. Ces dates sont essentielles pour contextualiser les données et permettre une analyse temporelle précise.

Enrichissement des données :

- Les dates de trimestre extraites sont ajoutées aux enregistrements de données correspondants, ce qui enrichit les données avec une dimension temporelle supplémentaire.

Combinaison et consolidation des données :

- Les doublons éventuels sont supprimés pour garantir l'unicité et l'intégrité des informations.

Standardisation des noms de colonnes :

- Les noms de colonnes sont uniformisés selon une convention définie. Cette standardisation facilite l'accès aux données et réduit les risques d'erreurs lors des analyses ultérieures.

Conversion des types de données :

- Les montants financiers et autres valeurs numériques sont convertis en types numériques appropriés. Cela assure la précision des calculs financiers et statistiques futurs.

Nettoyage des données :

- Le script traite les valeurs manquantes et corrige les incohérences éventuelles dans les données.
- Il supprime les caractères inutiles, tels que les symboles monétaires ou les séparateurs de milliers, pour normaliser les formats.

Sauvegarde des données nettoyées :

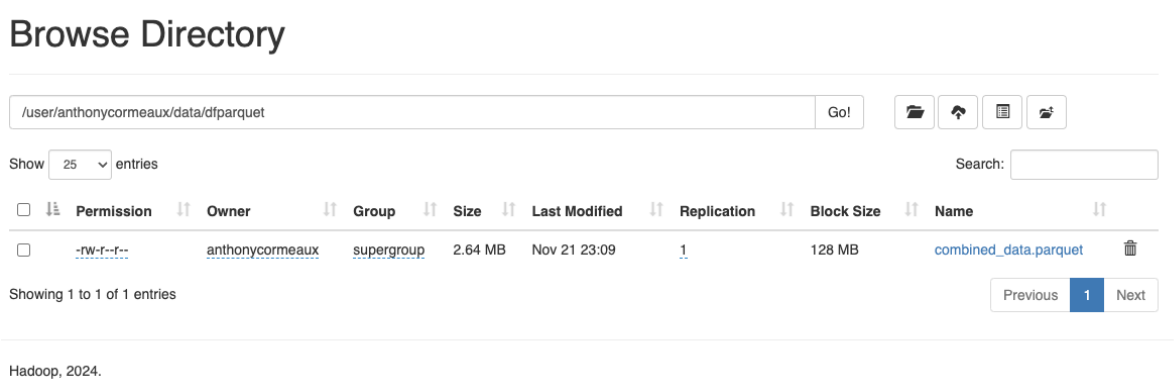
- Les données traitées sont enregistrées dans de nouveaux fichiers Excel, stockés dans un répertoire dédié aux données nettoyées.
- Ce stockage organisé facilite la gestion des versions et l'accès pour les analyses futures.

Traitement en lots des fichiers :

- Le script est conçu pour parcourir automatiquement tous les fichiers de données présents dans le répertoire source.
- Cela permet de traiter un grand nombre de fichiers sans intervention manuelle, augmentant ainsi l'efficacité du processus.

Une fois les données nettoyées, on les charge sur HDFS dans un répertoire.

Résultat dans HDFS :



Envoi et traitement des données en streaming

Nous avons utilisé le module Kafka avec python pour envoyer les données du fichier Excel sur un topic Kafka nommé « excel_data » par batch de 100 avec une pause toutes les 10 secondes. Le timestamp de l'envoi du message est inclus dans le message.

```
Message envoyé pour E00471 (Batch 1)
Message envoyé pour E00472 (Batch 1)
Message envoyé pour E00473 (Batch 1)
Message envoyé pour E00474 (Batch 1)
Message envoyé pour E00475 (Batch 1)
Message envoyé pour E00476 (Batch 1)
Message envoyé pour E00479 (Batch 1)
Message envoyé pour E00480 (Batch 1)
Message envoyé pour E00505 (Batch 1)
Message envoyé pour E00506 (Batch 1)
Message envoyé pour E00507 (Batch 1)
Message envoyé pour E00508 (Batch 1)
Message envoyé pour E00512 (Batch 1)
Message envoyé pour E00514 (Batch 1)
Message envoyé pour E00515 (Batch 1)
Message envoyé pour E00516 (Batch 1)
Message envoyé pour E00518 (Batch 1)
Message envoyé pour E00519 (Batch 1)
Message envoyé pour E00520 (Batch 1)
Message envoyé pour E00522 (Batch 1)
Message envoyé pour E00524 (Batch 1)
Message envoyé pour E00531 (Batch 1)
Message envoyé pour E00532 (Batch 1)
Message envoyé pour E00533 (Batch 1)
Message envoyé pour E00540 (Batch 1)
Message envoyé pour E00546 (Batch 1)
Message envoyé pour E00554 (Batch 1)
Message envoyé pour E00555 (Batch 1)
Message envoyé pour E00556 (Batch 1)
Message envoyé pour E00557 (Batch 1)
Message envoyé pour E00562 (Batch 1)
Message envoyé pour E00567 (Batch 1)
Message envoyé pour E00568 (Batch 1)
Message envoyé pour E00570 (Batch 1)
Message envoyé pour E00573 (Batch 1)
Message envoyé pour E00584 (Batch 1)
Message envoyé pour E00587 (Batch 1)
Message envoyé pour E00588 (Batch 1)
Batch 1 envoyé. Pause de 10 secondes.
```

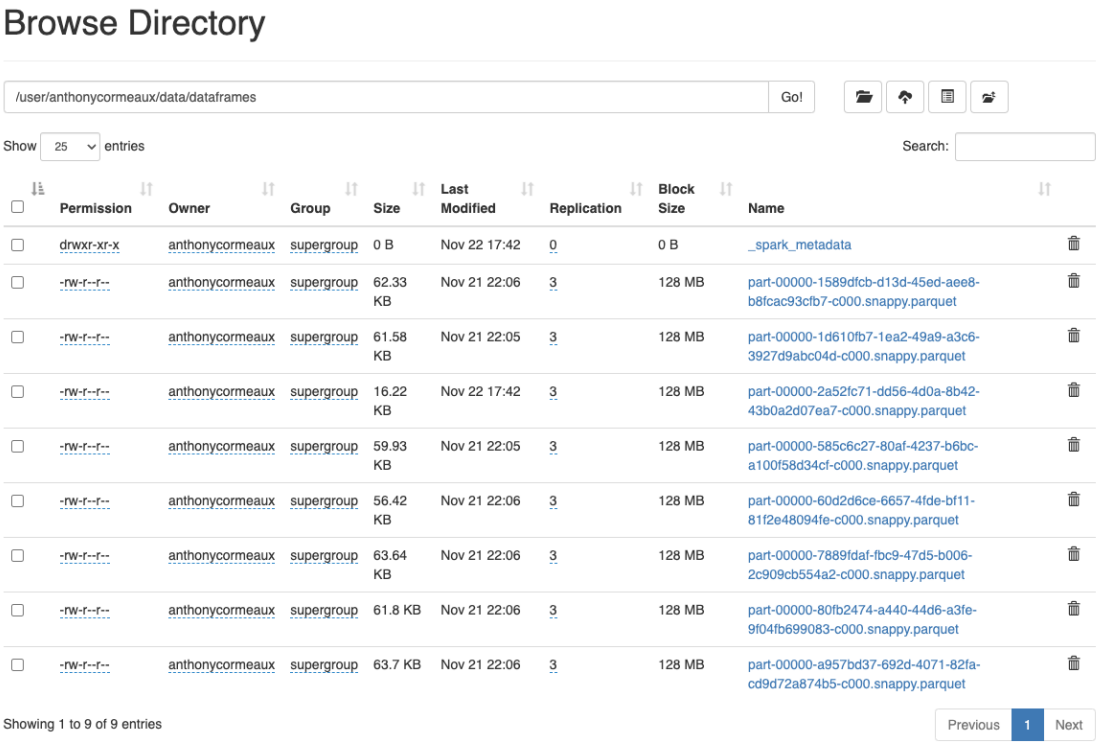
Une application Spark Streaming traite les données par batch et écris un nouveau Dataframe

```
Spark Structured Streaming est en cours d'exécution. Appuyez sur Ctrl+C pour arrêter.
24/11/22 17:42:53 WARN AdminClientConfig: These configurations '[key.deserializer, valu
```

```python
kafka_df = spark.readStream \
    .format("kafka") \
    .option("kafka.bootstrap.servers", "localhost:9092") \
    .option("subscribe", "excel_data") \
    .option("startingOffsets", "earliest") \
    .load()
```

```python
query = processed_df.writeStream \
    .format("parquet") \
    .option("path", hdfs_output_path) \
    .option("checkpointLocation", "hdfs://localhost:9080/user/anthonycormeaux/data/process") \
    .outputMode("append") \
    .trigger(processingTime='10 seconds') \
    .start()
```

Ici un test a été effectué par batch de 1000 pour plus de rapidité, les 7 Dataframe ont bien été crées.



Jointure des Dataframes

Le script qui effectue la jointure fait les actions suivantes :

Sauvegarde du fichier principal : Le script commence par créer une copie de sauvegarde du fichier de données principal. Cette sauvegarde est horodatée, ce qui permet de conserver un historique des versions pour une éventuelle restauration future.

Lecture des données : Il lit le fichier principal de données ainsi que les dataframes issus des données en streaming stockées dans le HDFS.

Préparation des données : Le script extrait les informations pertinentes des données du HDFS (adresse, ville, statecode et schoolcode) et traite les colonnes qui vont être utilisées pour la jointure comme chaine de caractères.

Fusion des données : La fusion est effectuée avec la colonne d'id trouvée entre les deux structures de données. Cette fusion permet d'enrichir le fichier principal avec des informations supplémentaires provenant du second ensemble de données.

Sauvegarde des données fusionnées : Après la fusion, le script crée une sauvegarde du fichier de la précédente jointure et écrit le nouveau dans le répertoire. Cette étape assure la traçabilité et permet de conserver une version sécurisée des données mises à jour.

Création des métriques

A partir du fichier parquet final, nous avons créé des métriques.

Calcul du total des prêts par école :

- Le script calcule le montant total de différents types de prêts émis pour chaque établissement scolaire. Les types de prêts incluent les prêts subventionnés, non subventionnés, Stafford et PLUS.
- Il additionne ces montants pour obtenir le total des prêts par école.
- **Objectif** : Fournir une vue détaillée du volume et de la distribution des prêts associés à chaque établissement, ce qui peut aider à identifier les écoles avec les plus hauts niveaux de financement par prêts.

Calcul du total des prêts par État :

- De manière similaire, le script agrège les montants totaux des prêts pour chaque État.
- En regroupant les données de toutes les écoles au sein d'un État, il calcule le total des prêts émis dans chaque catégorie de prêts.

Analyse de l'évolution des prêts dans le temps :

- Le script analyse comment les montants des prêts évoluent au fil du temps, en se basant sur les périodes trimestrielles.
- Il regroupe les données par début et fin de trimestre et calcule les totaux pour chaque type de prêt sur ces périodes.

Sauvegarde et préservation des données :

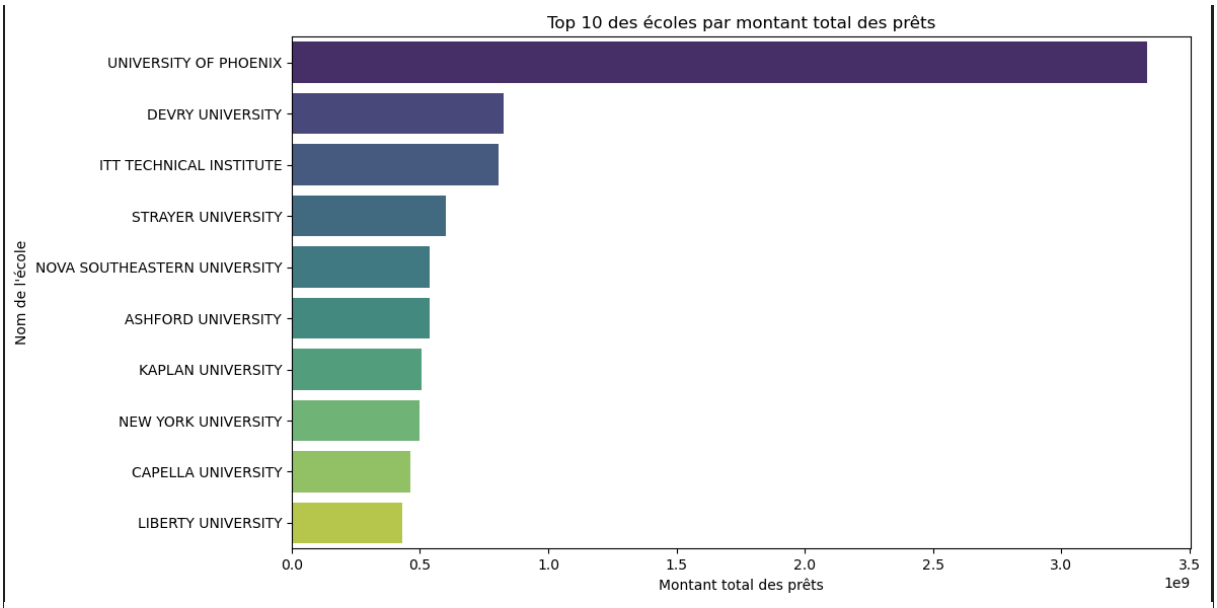
- Avant d'enregistrer les nouvelles métriques calculées, le script crée des copies de sauvegarde des données existantes. Ces sauvegardes sont horodatées pour assurer un historique précis.
- **Objectif** : Prévenir toute perte de données et maintenir un enregistrement des versions précédentes pour référence.

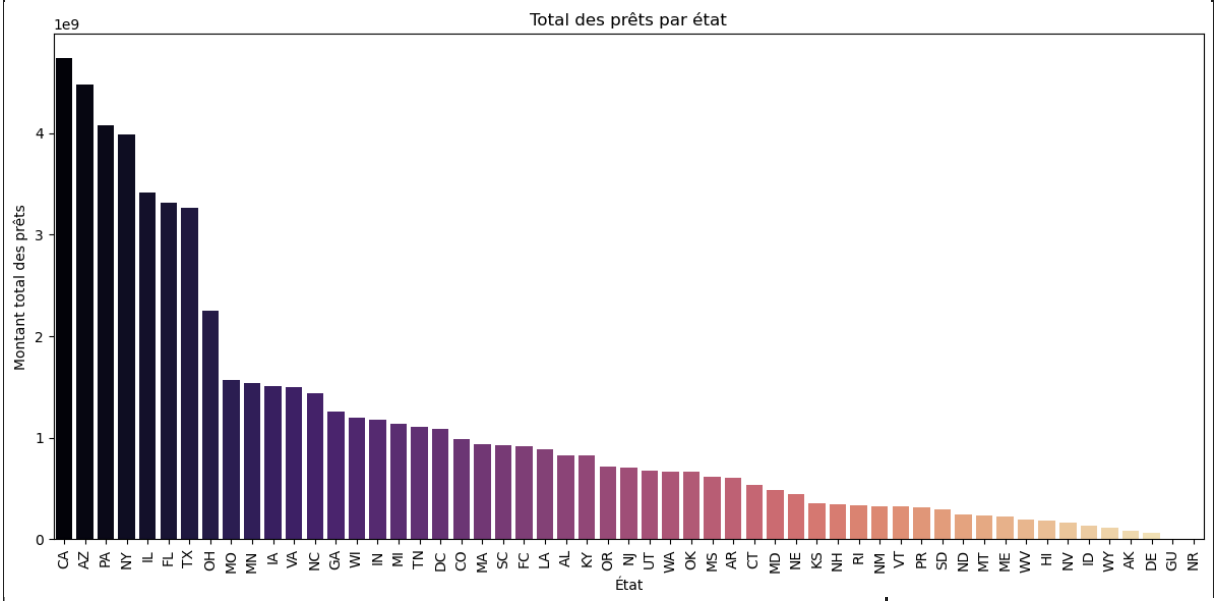
Enregistrement des métriques calculées :

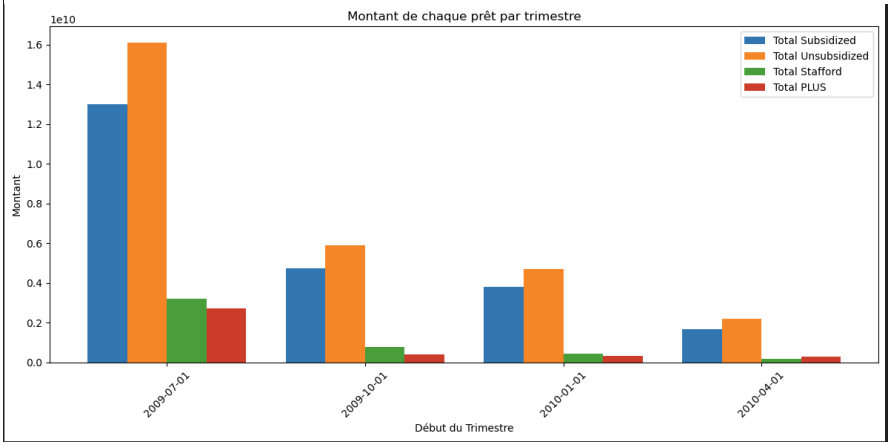
- Le script enregistre les nouvelles données agrégées dans le système de stockage, en remplaçant les anciennes données par les plus récentes.
- Les résultats sont stockés dans des emplacements spécifiques pour chaque type de métrique (par école, par État, évolution des prêts), facilitant ainsi l'accès et l'analyse.

Visualisation

A partir de ces tables de métriques, nous avons crée des visualisations. Étant donné que les tables dans le répertoire seront à jour après chaque nouvelle intégration, les visuels seront également à jour.



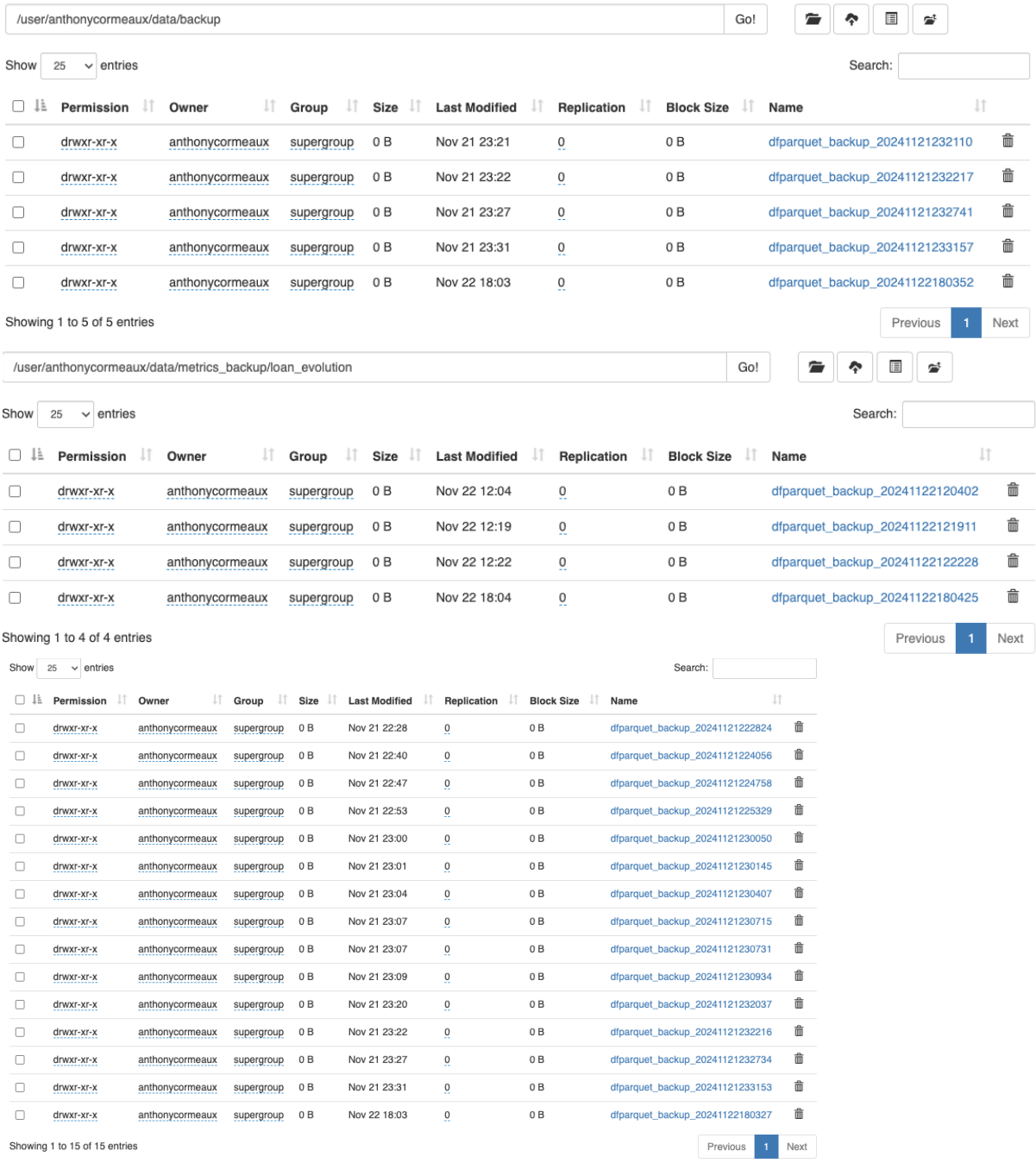




Procédure pour revenir à un précédent état

Des backups ont été mis en place pour le fichier principal, le fichier après jointure et les métriques.

Dans les répertoires de chaque backup, il y a des sous répertoires avec des timestamp, pour revenir à une version précédente il faut définir la date à laquelle revenir puis extraire et déplacer les données du répertoire avec le timestamp le plus proche de la date souhaitée.



/user/anthonycormeaux/data/backup Go!

Show 25 entries Search:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	anthonycormeaux	supergroup	0 B	Nov 21 23:21	0	0 B	dfparquet_backup_20241121232110
drwxr-xr-x	anthonycormeaux	supergroup	0 B	Nov 21 23:22	0	0 B	dfparquet_backup_20241121232217
drwxr-xr-x	anthonycormeaux	supergroup	0 B	Nov 21 23:27	0	0 B	dfparquet_backup_20241121232741
drwxr-xr-x	anthonycormeaux	supergroup	0 B	Nov 21 23:31	0	0 B	dfparquet_backup_20241121233157
drwxr-xr-x	anthonycormeaux	supergroup	0 B	Nov 22 18:03	0	0 B	dfparquet_backup_20241122180352

Showing 1 to 5 of 5 entries Previous 1 Next

/user/anthonycormeaux/data/metrics_backup/loan_evolution Go!

Show 25 entries Search:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	anthonycormeaux	supergroup	0 B	Nov 22 12:04	0	0 B	dfparquet_backup_20241122120402
drwxr-xr-x	anthonycormeaux	supergroup	0 B	Nov 22 12:19	0	0 B	dfparquet_backup_20241122121911
drwxr-xr-x	anthonycormeaux	supergroup	0 B	Nov 22 12:22	0	0 B	dfparquet_backup_20241122122228
drwxr-xr-x	anthonycormeaux	supergroup	0 B	Nov 22 18:04	0	0 B	dfparquet_backup_20241122180425

Showing 1 to 4 of 4 entries Previous 1 Next

Show 25 entries Search:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	anthonycormeaux	supergroup	0 B	Nov 21 22:28	0	0 B	dfparquet_backup_20241121222824
drwxr-xr-x	anthonycormeaux	supergroup	0 B	Nov 21 22:40	0	0 B	dfparquet_backup_20241121224056
drwxr-xr-x	anthonycormeaux	supergroup	0 B	Nov 21 22:47	0	0 B	dfparquet_backup_20241121224758
drwxr-xr-x	anthonycormeaux	supergroup	0 B	Nov 21 22:53	0	0 B	dfparquet_backup_20241121225329
drwxr-xr-x	anthonycormeaux	supergroup	0 B	Nov 21 23:00	0	0 B	dfparquet_backup_20241121230050
drwxr-xr-x	anthonycormeaux	supergroup	0 B	Nov 21 23:01	0	0 B	dfparquet_backup_20241121230145
drwxr-xr-x	anthonycormeaux	supergroup	0 B	Nov 21 23:04	0	0 B	dfparquet_backup_20241121230407
drwxr-xr-x	anthonycormeaux	supergroup	0 B	Nov 21 23:07	0	0 B	dfparquet_backup_20241121230715
drwxr-xr-x	anthonycormeaux	supergroup	0 B	Nov 21 23:07	0	0 B	dfparquet_backup_20241121230731
drwxr-xr-x	anthonycormeaux	supergroup	0 B	Nov 21 23:09	0	0 B	dfparquet_backup_20241121230934
drwxr-xr-x	anthonycormeaux	supergroup	0 B	Nov 21 23:20	0	0 B	dfparquet_backup_20241121232037
drwxr-xr-x	anthonycormeaux	supergroup	0 B	Nov 21 23:22	0	0 B	dfparquet_backup_20241121232216
drwxr-xr-x	anthonycormeaux	supergroup	0 B	Nov 21 23:27	0	0 B	dfparquet_backup_20241121232734
drwxr-xr-x	anthonycormeaux	supergroup	0 B	Nov 21 23:31	0	0 B	dfparquet_backup_20241121233153
drwxr-xr-x	anthonycormeaux	supergroup	0 B	Nov 22 18:03	0	0 B	dfparquet_backup_20241122180327

Showing 1 to 15 of 15 entries Previous 1 Next

Base de données à utiliser

Si nous devions utiliser une base de données pour ce projet, Apache Cassandra serait une bonne option.

Cette base de données est tolérante aux fautes, son efficacité est puissante en lecture et en écriture et surtout elle s'intègre bien avec Spark grâce au connecteur Spark Cassandra Connector.

Des autres options pourraient être Apache Hive ou Hbase pour leur intégration avec Hadoop.