

Parquet et DuckDB

23 mai 2024

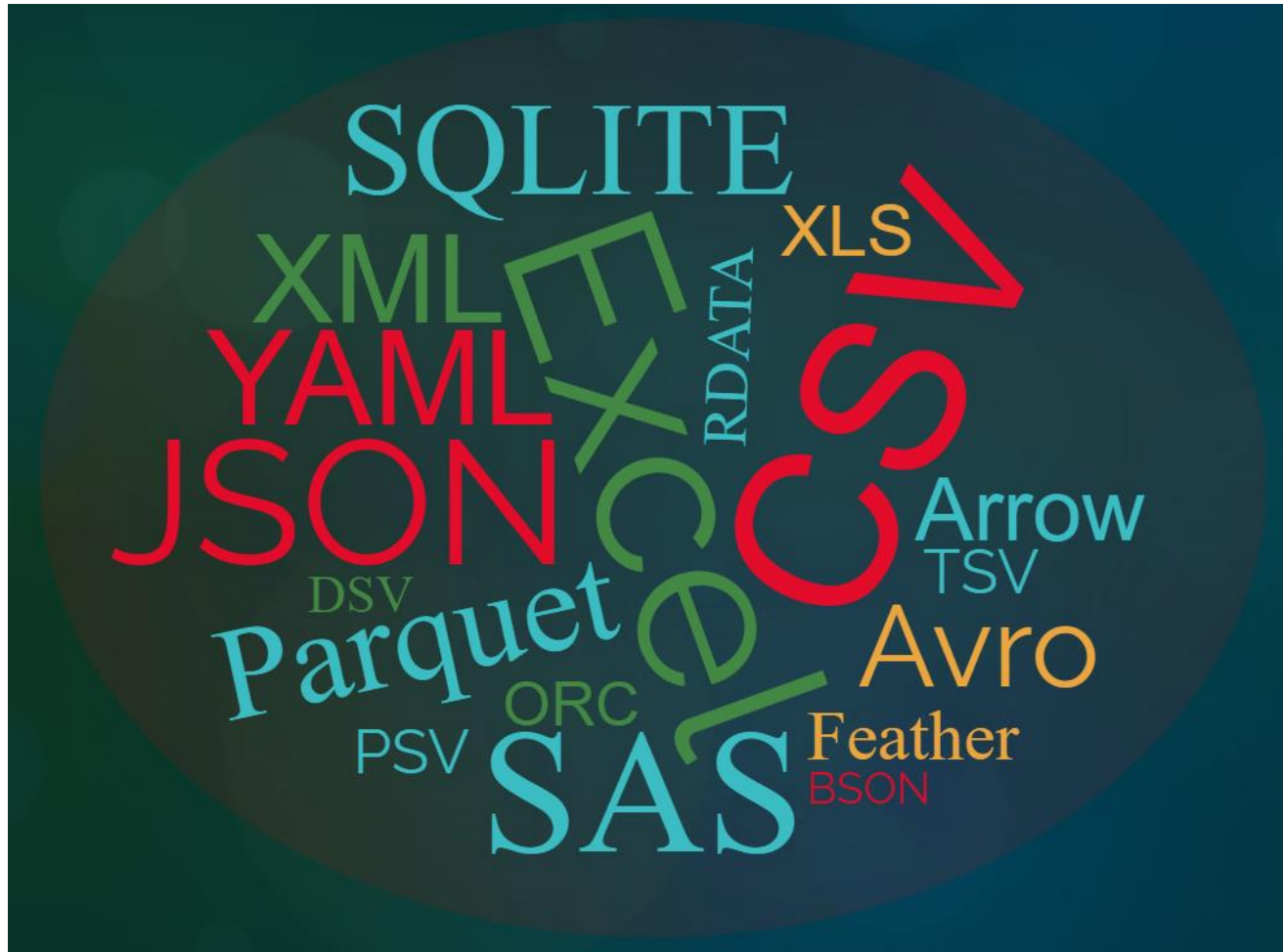
Pengfei Liu, Titouan Rigaud

Le CASD :

- Groupement d'intérêt public
- a but non lucratif
- fournir des accès sécurisé aux données, y compris de santé

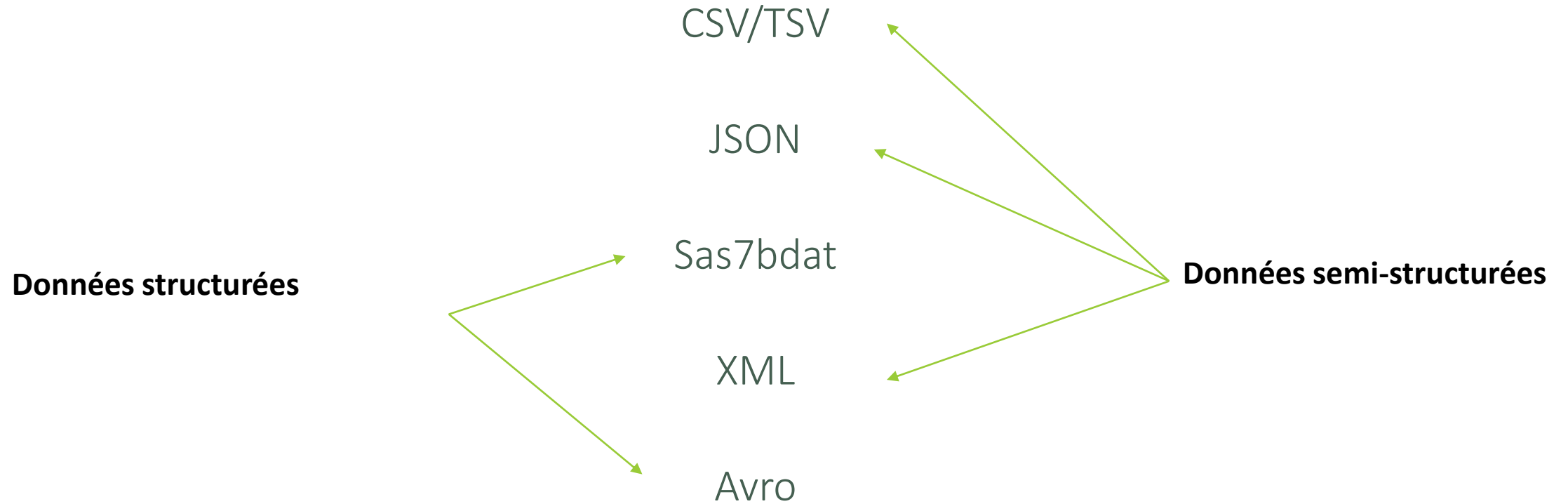
Introduction

Formats de données populaires



- Nombreux
- Une histoire ancienne
- Des formats pour chaque usage

Dans quelle catégorie sont ces formats ?



Le format Parquet

Format de fichiers parquet

- Lancé en 2012 - Cloudera et Twitter
- Pris en charge par la fondation Apache en 2014 (devenu projet de haut niveau en 2015)
- Stockage en colonnes sur disque :
 - Interoperable
 - Efficace au niveau de l'espace de stockage
 - Très efficace dans un objectif d'analyse

Dédié pour le WORM : **Write Once Read Many**

Stockage colonne ou ligne ?

Table

	Country	Product	Sales
Row 1	India	Chocolate	1000
Row 2	India	Ice-cream	2000
Row 3	Germany	Chocolate	4000
Row 4	US	Noodle	500

Row Store	
Row 1	India
	Chocolate
	1000
Row 2	India
	Ice-cream
	2000
Row 3	Germany
	Chocolate
	4000
Row 4	US
	Noodle
	500

Column Store	
Country	India
	India
	Germany
	US
Product	Chocolate
	Ice-cream
	Chocolate
	Noodle
Sales	1000
	2000
	4000
	500











- Pour lire la troisième ligne, mieux vaut le stockage ligne
- Pour calculer la moyenne des ventes, mieux vaut le stockage colonne !

Format parquet et compression

Différence CSV compressé / Parquet compressé ?

- En CSV, compression au niveau du fichier entier
- En parquet, compression colonne par colonne et les API en tiennent compte d'office (transparent pour l'utilisateur) (en snappy par défaut)

Format parquet et compression

	netflixraw.sas7bdat	28/11/2023 17:32	SAS Data Set	704 Ko
	netflixcompress.sas7bdat	28/11/2023 17:32	SAS Data Set	640 Ko
	estraddDuckDB.DDB.wal	29/11/2023 09:33	Fichier WAL	489 Ko
	netflix-csv.CSV	28/11/2023 17:25	Fichier CSV Micro...	425 Ko
	netflix-snappy.parquet	28/11/2023 16:57	Fichier PARQUET	91 Ko
	netflixFromArrow.parquet	29/11/2023 09:06	Fichier PARQUET	78 Ko
	netflixTize.parquet	29/11/2023 08:47	Fichier PARQUET	73 Ko
	netflix-csv.zip	28/11/2023 18:00	Dossier compressé	65 Ko
	netflix-ZSTD.parquet	28/11/2023 16:57	Fichier PARQUET	55 Ko
	netflix-ZSTD.zip	28/11/2023 18:01	Dossier compressé	51 Ko

Plusieurs algorithmes de compression sont disponibles

Comparaison

fichier	taille en ko	% par rapport au 1 ^{er}
netflix.sas7bdat	704	100%
netflix.sas7bdat compressé	640	91%
netflix.DDB	489	69%
netflix.CSV	425	60%
netflix.parquet snappy	91	13%
netflix.parquet arrow	78	11%
netflix.parquet Tize	73	10%
netflix.parquet ZSTD	55	8%

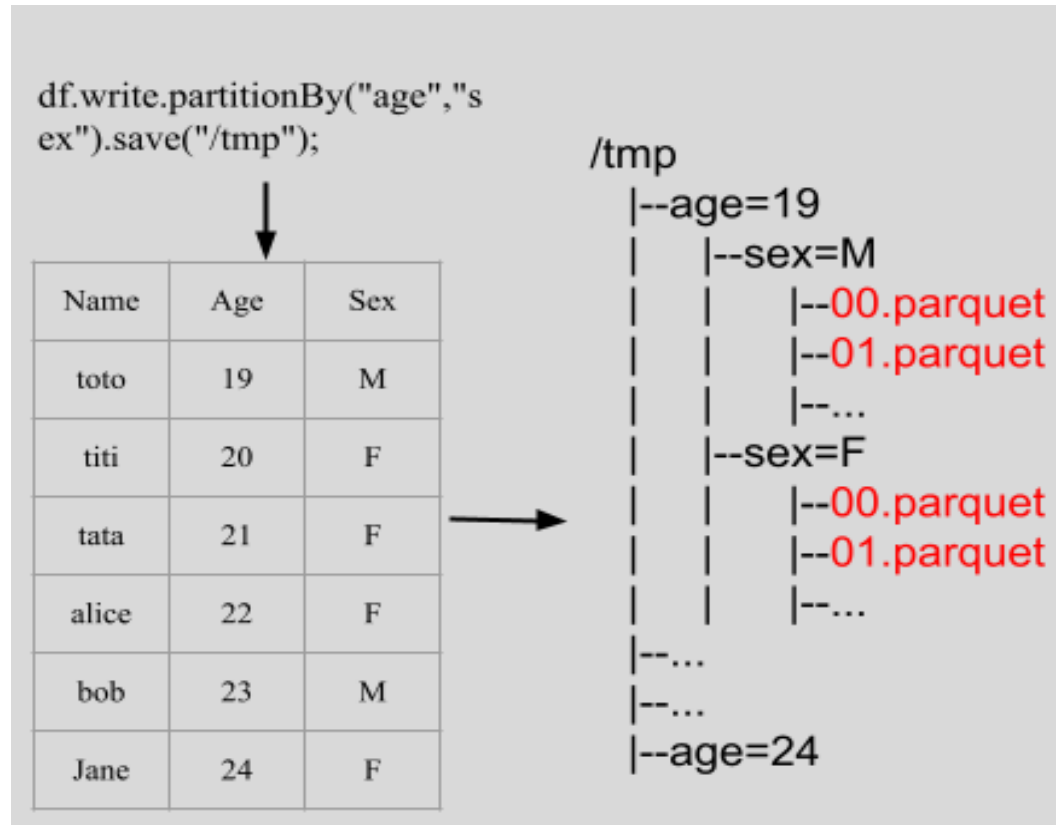
Exemple pour un fichier presta de données de 730 millions de lignes :

fichier	taille en Go	% par rapport au 1 ^{er}
presta.csv	417	100%
presta.sas7bdat compressé	311	75%
presta.parquet	24	6%

Partitionnement

- Création d'autant de fichiers que de partition dans un répertoire
- Transparent pour l'utilisateur
- Plutôt qu'un gros fichier : plusieurs petits fichiers (par exemple par mois, département...)
- Écueil à éviter : faire trop de partitions

Un exemple de partitionnement :





Partitions	2	5	1000
Colonne qui a servi au partitionnement	74,50%	46,30%	16,66%
Vers une autre colonne	89,51%	191,01%	556,99%
Select distinct(*)	136,79%	163,68%	1194,88%

Parquet est il adapté ?

Propriété	CSV	Json	Parquet	Avro	SAS7BDAT
Lisible humainement	Oui	Oui	Non	Non	Non
Compressé par défaut	Non	Non	Oui	Oui	Oui*
Divisible	Oui*	Oui*	Oui	Oui	Oui
Structures de données complexes	Non	Oui	Oui	Oui	Oui
Schema évolutif	Non	Non	Oui	Oui	Oui
En colonne	Non	Non	Oui	Non	Non
Supporté par les logiciels	Oui++	Oui++	Oui+	Oui+	Oui
Support des métadonnées	Non	Non	Oui++	Oui	Oui
Modifier une valeur	Oui	Oui	Non	Oui	Oui

DuckDB

DUCKDB

- DuckDB est un système de base de données (SGBD) in process
- pas de serveur dédié comme en PostgreSQL, MySQL, Oracle
 - L'exécution du code de DuckDB est réalisée par le programme hôte
 - R/Python
 - C++ ou C
- Le stockage de la base de données se fait
 - Par défaut en RAM (Non persistant) 
 - Il est recommandé de persister le fichier sur disque 



DUCKDB

- Un fichier DB peut contenir plusieurs tables
- DuckDB s'inspire de SQLite mais se spécialise sur l'analytics (optimisé pour cela) – dispose des dernières versions SQL (window functions...)



Performances de duckDB

- <https://duckdblabs.github.io/db-benchmark/>



groupby

join

groupby2014

















0.5 GB

5 GB

50 GB

basic questions

Input table: 1,000,000,000 rows x 9 columns (50 GB)

	duckdb-latest	0.9.1.1	2023-10-26	24s
	DuckDB	0.8.1.3	2023-10-26	25s
	Polars	0.19.8	2023-10-17	32s
	ClickHouse	23.9.1.1854	2023-10-24	39s
	DataFrames.jl	1.6.1	2023-10-17	84s
	data.table	1.14.9	2023-10-17	89s
	Datafusion	31.0.0	2023-10-24	133s
	InMemoryDataSets	0.7.1.8	2023-10-17	218s
	collapse	2.0.3	2023-10-26	233s
	spark	3.5.0	2023-10-24	297s
	Arrow	13.0.0.1	2023-10-17	511s
	pandas	2.1.1	2023-10-17	773s
	(py)datatable	1.1.0a0	2023-10-17	993s
	dplyr	1.1.3	2023-10-17	1022s
	dask	2023.10.0	2023-10-17	out of memory
	Modin		see README	pending

groupby

join

groupby2014
















0.5 GB

5 GB

50 GB

basic questions

Input table: 1,000,000,000 rows x 7 columns (55 GB)

	InMemoryDataSets	0.7.1.8	2023-11-06	252s
	DuckDB*	0.8.1.3	2023-10-25	428s
	duckdb-latest*	0.9.1.1	2023-10-25	459s
	collapse	2.0.3	2023-10-26	914s
	DataFrames.jl	1.6.1	2023-11-06	out of memory
	data.table	1.14.9	2023-11-08	timeout
	dplyr	1.1.3	2023-11-06	out of memory
	pandas	2.1.1	2023-11-06	out of memory
	spark*	3.5.0	2023-10-24	timeout
	dask	2023.10.1	2023-11-06	out of memory
	Polars	0.19.11	2023-11-06	out of memory
	Arrow	13.0.0.1	2023-11-06	out of memory
	Datafusion	31.0.0	2023-10-25	undefined exception
	ClickHouse	23.9.1.1854	2023-10-24	out of memory
	Modin		see README	pending

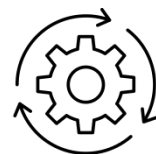
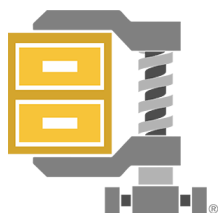
Accès à un fichier parquet



Fichier DuckDB (attention par défaut, création de la DB en RAM)

PRESENTATIONS

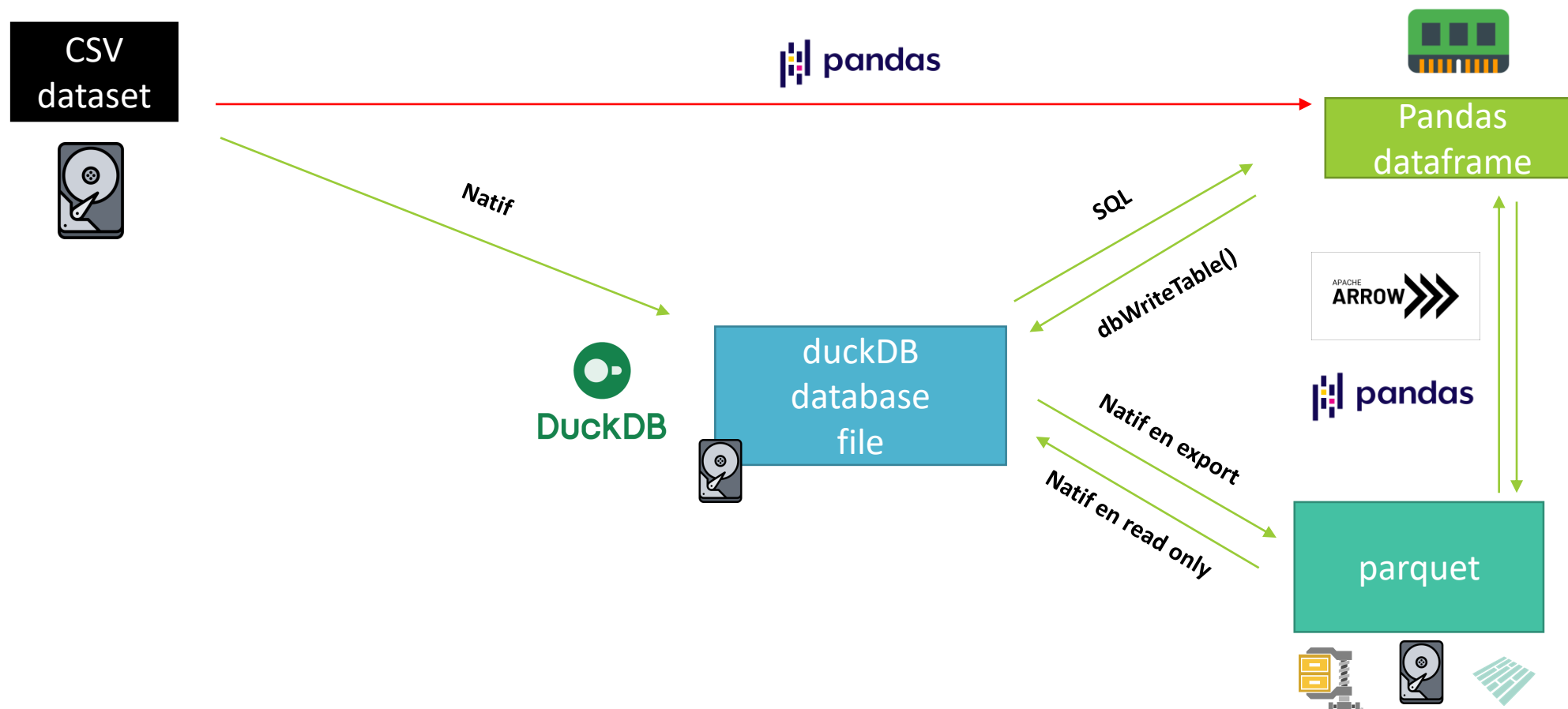
estraddDuckDB.DDB	29/11/2023 09:23	Fichier DDB	12 Ko
-------------------	------------------	-------------	-------



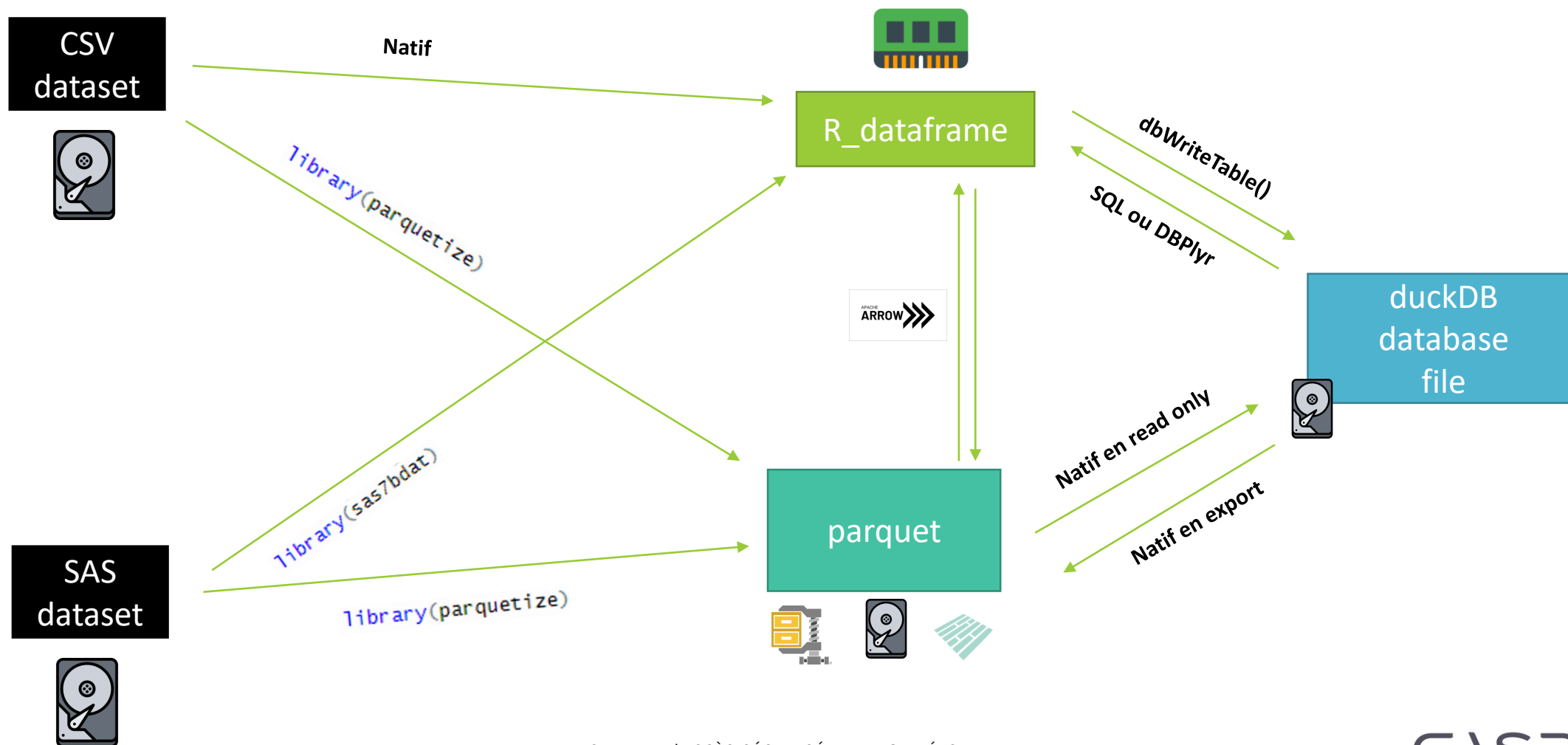
EXECUTION DE LA REQUETE DIRECTEMENT DANS LE
FICHIER PARQUET SUR DISQUE
netflixTize.parquet
(pas de duplication des données ni sur disque ni en RAM)

Convertir un fichier en Parquet

Exemples pour convertir en Parquet avec Python



Exemples pour convertir en Parquet avec R



Démonstration !

Conclusion

- Parquet est un format de stockage efficace mais WORM :
 - Efficace pour le stockage de données qui seront utilisées en lecture
 - A éviter si le fichier doit régulièrement être mis à jour : C'est le même principe que l'object-storage
- DuckDB permet d'adresser directement du Parquet en lecture
 - Mais pas seulement, il est particulièrement optimisé et performant pour l'analytics (requêtes complexes, grosses jointures)
 - Permet la modification via commande SQL ou dplyr sur les tables (pas pour les vues Parquet)
- Pour la modification de table, il faut préférer **SQL**, R, Python... et VTL qui permet de documenter et exécuter les transformations et validations sur les tables de données
- Data Build Tool pour faire un pipeline (ETL)

MERCI DE VOTRE ATTENTION

CASD