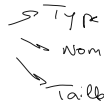


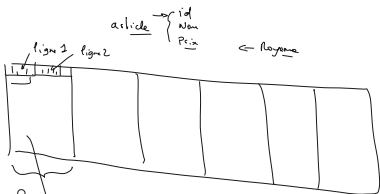
ligne



Colonnes



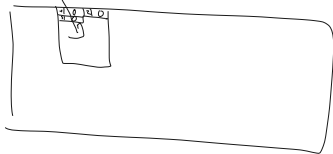
ligne vers



Page bloc $\approx 8ko$

→ Row Store → Pas adapté pour l'OLAP (Moy. Sum, Count)

Mémoire (RAM) (cache)



Column Store

→ accès
↳ Colonnes

Schema Relationnel ⇒
(Table, jointure, index)

OLAP (BI)

Schema flexion
-table

(Table, join)

→ faits
dimensions
des
mesures

→ Ecriture
Unique

(Batch d'insertion) Hierarchies

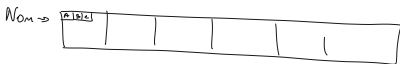
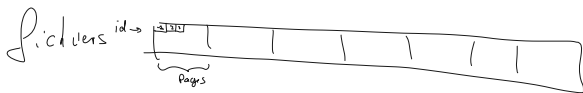
→ lectures multiples

(Requêtes analytiques)

ligne

→ Prix en dollars

id	Nom	Prix
----	-----	------



Théorème de CAP

→ Cohérence

→ Dispon.b.ité' (A) variableté)

→ Partitionnement (découpage, étalement de charge)

$$\text{Haute disponib.ité'} = A + P -$$

Aller sur <https://community.cloud.databricks.com>

Créer un compte

Créer un cluster

Créer un workspace

Importez un notebook, l'url :

<https://www.databricks.com/notebooks/gcp-qs-notebook.html>

exécutez le ligne pas ligne.

Connection dans la VM de stage sous Azure :

Dans un navigateur :

<http://hadoop.francecentral.cloudapp.azure.com/guacamole/#/>

login / mot de passe : student / Pa\$\$w0rd (avec un zéro)

Choisissez votre user : (mot de passe : password)

Fabricio : st01

Cédric : st02

Hervé : st03

Arnaud : st04

Estelle : st05

Sarah : st06

Machine Bastion Host :

<https://bst-84688f3a-b0a0-4bc4-8852-0e89cbbd1dd9.bastion.azure.com/api/shareable-url/7f6c3535-5a9f-4ffe-987f-b7d4b594a2dd>

Prenez RDP, votre compte (stxx / mot de passe : password)

Référence Apache Hive :

<https://wiki.apache.org/confluence/display/Hive/LanguageManual+Types>

Fonctions Hive :

<https://wiki.apache.org/confluence/display/Hive/LanguageManual+UDF>

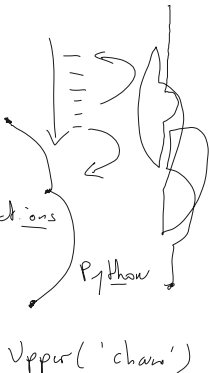
L1G → assembleur

L2G → basic, script, cobol
R ↙ ↘
bash - groovy

L3G → Prog structurée
→ pascal, C, VB
→ structures, fonctions

L4G → Prog. orienté objet
→ .net
→ Java

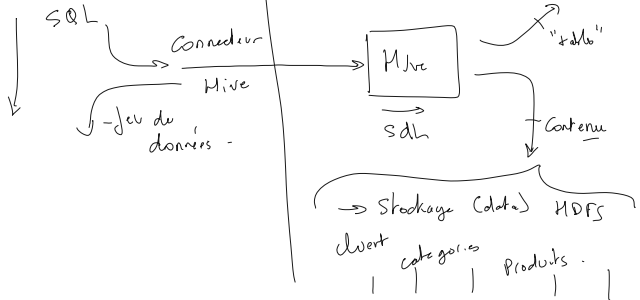
fonction (paramètres)
} décrit une fonction
opération (fonction)



SAS

Hadoop

metadatas



client
fonction predicat
Tri

→ par tranches d'age
si on a salaire

Select

where sort(fa)

Predicat

Scala

.where

row => row.age < 80 &&
row. —

fonction

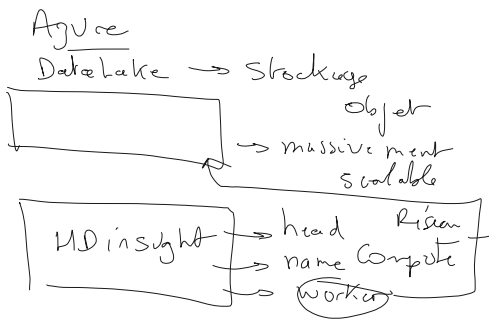
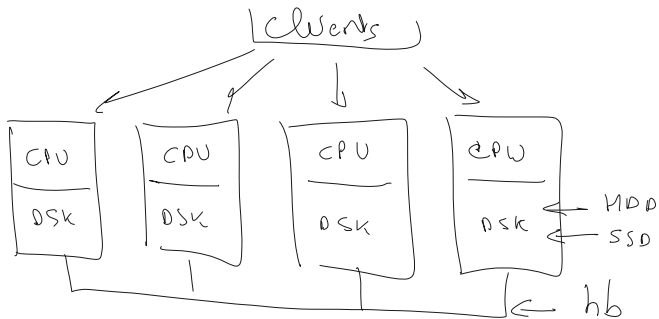
Data
Hadoop

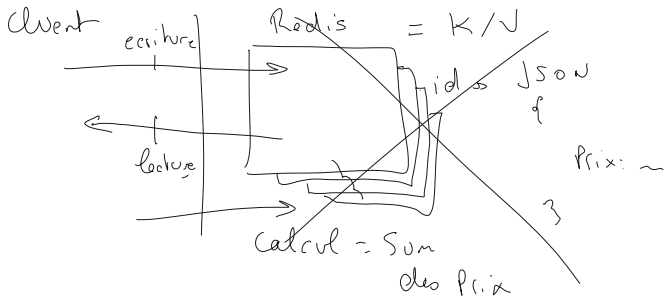
FS to data

.jar



code



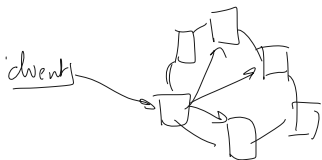


Cassandra → Columnstore

→ CREATE Table

Insert, Update, delete

RVCC



→ Répliques +
→ Partitions

libs → javascript

Documents → JSON

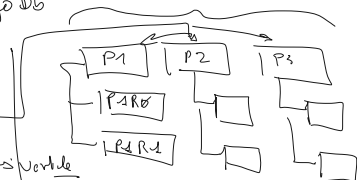
Sentilles par Clé

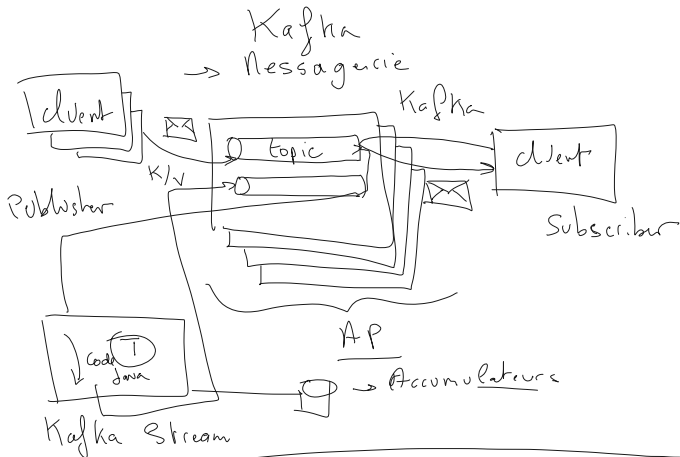
{ id: " " } → hash → verticle

Tratament puissant
map Reduce

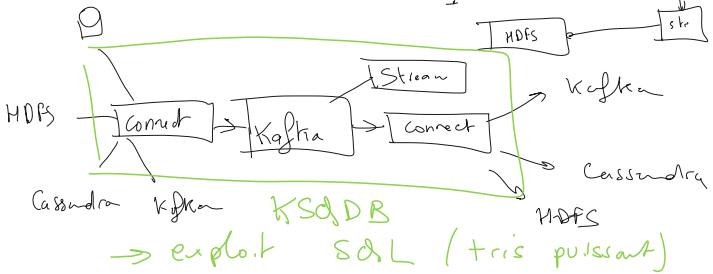
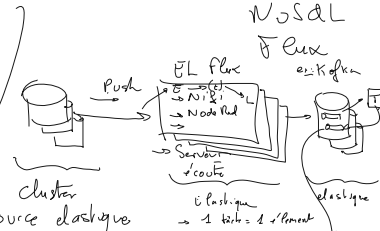
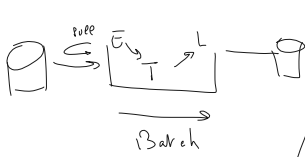
MongoDB

Partitions





ETL → ELT → "Moderne"



Ressources Hadoop

<https://we.tl/t-w0yq5vleEI>

Ambari :

<https://agirc.azurehdinsight.net/#/main/dashboard/metrics>

admin / Pa\$\$w0rd2023

Portail Hive via Hue :

https://agirc.azurehdinsight.net/#/main/view/HIVE/auto_hive20_instance

Reference Hive :

<https://wiki.apache.org/confluence/display/hive/languagemanual>

VM :

student / Pa\$\$w0rd2023

rdp :

sas :

<https://sashadoop.francecentral.cloudapp.azure.com>