

17/05/2023

BIG DATA

Livrable 2 : Modèle physique et optimisation

MEMBRES DU GROUPE :

GUELLATI Mohamed (Chef De Groupe)

ALI ZOUAOUI Zakaria

ZAMOUCHE Nadir

Table des matières :

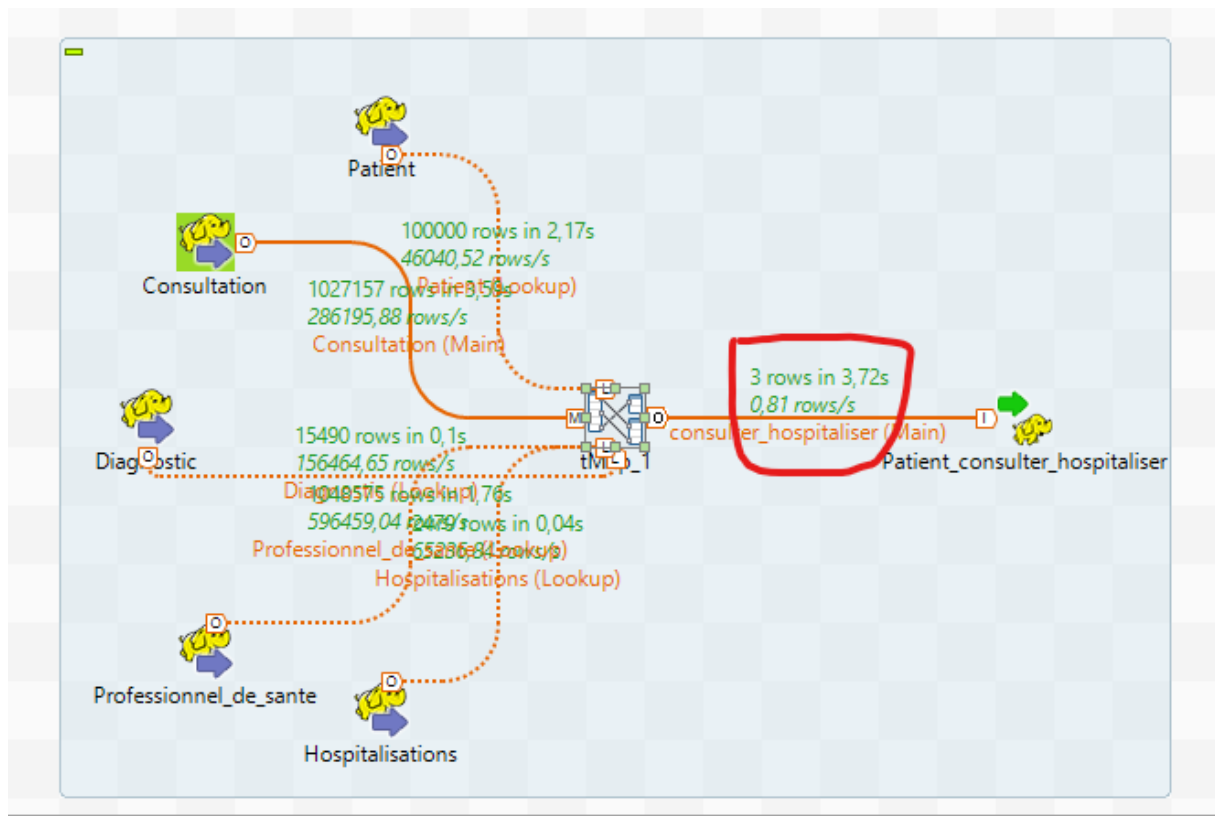
Introduction :	3
Changements :	3
A prendre en considération :	7
Objectifs :	7
I. Script pour la création et le chargement de données dans les tables :	8
II. Vérification des données présentes et accès aux données à travers les tables :	10
III. Script montrant le peuplement des tables :	13
IV. Script montrant le partitionnement et les buckets :	13
V. Graphes montrant les temps de réponses pour évaluer la performance d'accès à l'entrepôt de données :	21
VI. Requêtes faisant foi pour l'évaluation de la performance:	21
Conclusion :	23

Introduction :

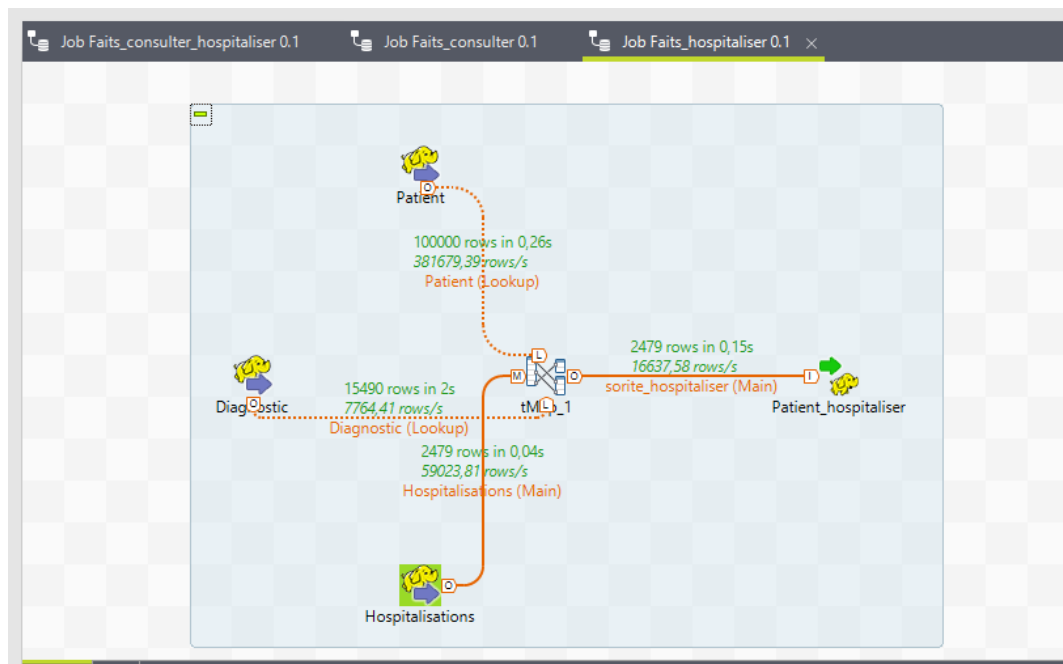
Dans ce livrable du projet Big Data, nous pouvons retrouver un rendu orienté au niveau des requêtes réalisées et des tables constituant notre modèle de liaison de données. De plus, la création des tables sur hive avec leur chargement, leur peuplement ainsi que les scripts dédiés au partitionnement et au buckets. En d'autres termes, le livrable 2 est centré autour de l'évaluation de performance par rapport au temps de réponse des requêtes.

Changements :

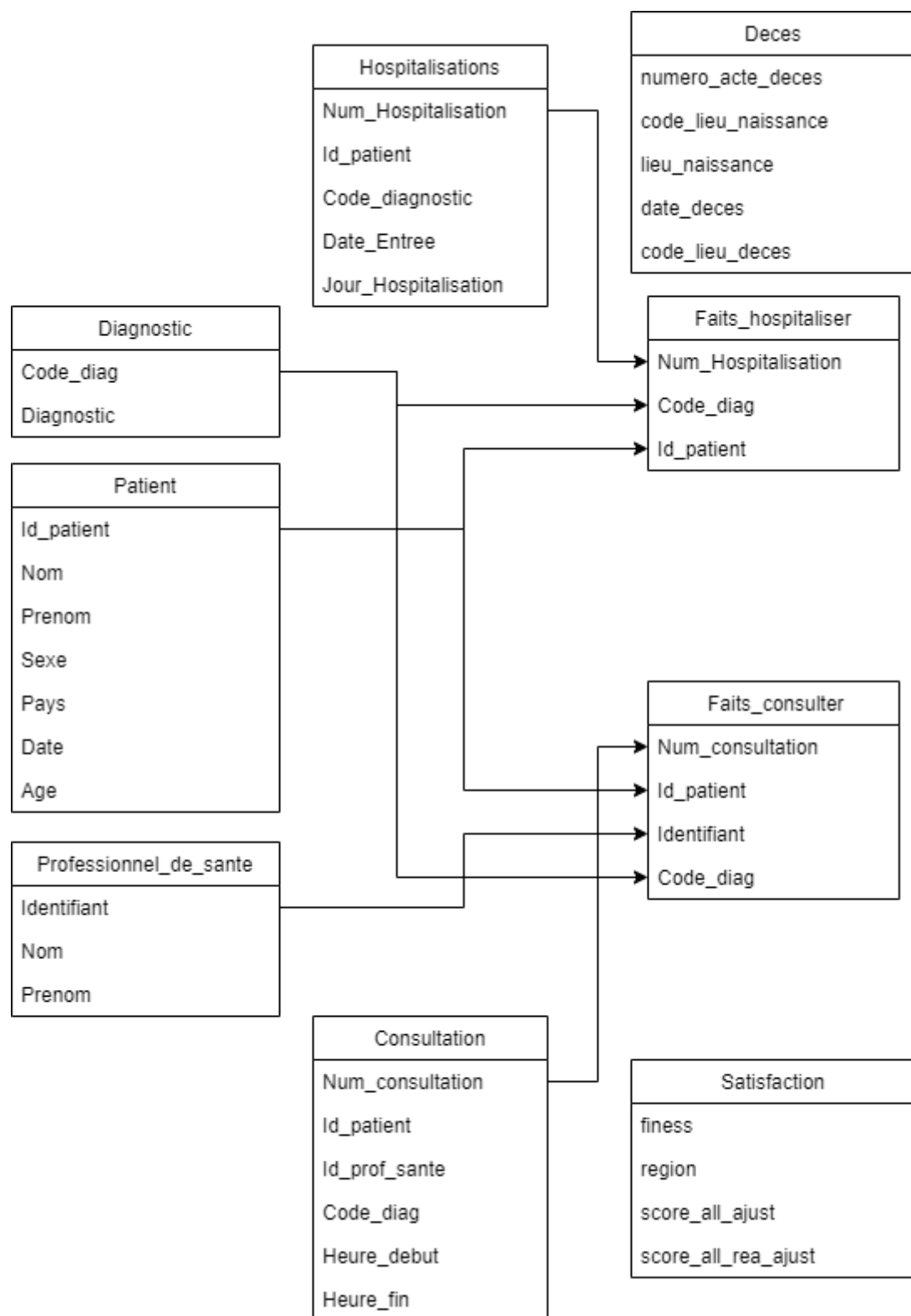
On a changé le job des Faits, au lieu d'avoir un seul job des Faits, on a créé deux : Faits_consulter et Faits_hospitaliser parce que si on les regroupe dans le même job et avec les jointures sur tMap on aura que 3 lignes comme résultats !



Les nouveaux jobs Faits_consulter et Faits_hospitaliser respectivement :



Et donc voici le nouveau modèle en constellation :



A prendre en considération :

- **Cluster** : Un cluster Hadoop est un type spécial de cluster de calcul (grappe de serveurs) conçu spécifiquement pour stocker et analyser de grandes quantités de données non structurées au sein d'un environnement de calcul distribué. Le travail d'analyse de données est réparti entre les différents nœuds du cluster (serveurs). Ces clusters permettent de lancer le logiciel de traitement distribué open source Hadoop sur des ordinateurs low-cost.
- **Partition** : est une division logique d'une table stockée en plusieurs parties indépendantes. Il peut être horizontal (les lignes d'une même table sont divisées en plusieurs partitions) ou vertical (les colonnes d'une même table sont divisées en plusieurs lots). Le partitionnement de tables est généralement effectué pour améliorer la gestion, la performance ou la disponibilité. Chaque partition peut être placée sur des serveurs ou des disques différents. Cela permet également d'obtenir une capacité de base de données supérieure à la taille maximum des disques durs ou d'effectuer des requêtes en parallèle sur plusieurs partitions.
- **Bucket** : Pour pallier le problème de sur-partitionnement, Hive a introduit le concept de Bucketing. Il s'agit d'une technique d'organisation des données en parties plus petites appelées "buckets". Dans ce concept, la table possède une structure spécifique basée sur une fonction de hachage qui s'applique sur une colonne donnée. Ceci impose aussi la façon avec laquelle les fichiers sous-jacents sont stockés. En effet, les données ayant la même bucket-colonne seront toujours dans le même bucket.

Objectifs :

- Script pour la création et le chargement de données dans les tables
- Vérification des données présentes et accès aux données à travers les tables
- Script montrant le peuplement des tables
- Script pour le partitionnement et les buckets
- Graphes montrant les temps de réponses pour évaluer la performance d'accès à l'entrepôt de données
- Requetes faisant foi pour l'évaluation de la performance

I. Script pour la création et le chargement de données dans les tables :

- consultation :

```
1 CREATE EXTERNAL TABLE consultation(num_consultation BIGINT, id_patient BIGINT,  
2 id_prof_sante BIGINT, code_diag STRING, heure_debut STRING, heure_fin STRING)  
3 ROW FORMAT DELIMITED FIELDS TERMINATED BY ';' ;  
4 STORED AS TEXTFILE  
5 LOCATION '/user/hive/data';  
6  
7 LOAD INPATH '/user/cloudera/student/Consultation.txt';
```

- deces :

```
1 CREATE EXTERNAL TABLE deces(numero_acte_deces STRING, lieu_naissance STRING,  
2 code_lieu_naissance BIGINT, date_deces STRING, code_lieu_deces BIGINT)  
3 ROW FORMAT DELIMITED FIELDS TERMINATED BY ';' ;  
4 STORED AS TEXTFILE  
5 LOCATION '/user/hive/data';  
6  
7 LOAD INPATH '/user/cloudera/student/Deces.txt';
```

- diagnostic :

```
1 CREATE EXTERNAL TABLE diagnostic(code_diag STRING, diagnostic STRING)  
2 ROW FORMAT DELIMITED FIELDS TERMINATED BY ';' ;  
3 STORED AS TEXTFILE  
4 LOCATION '/user/hive/data';  
5  
6 LOAD INPATH '/user/cloudera/student/Diagnostic.txt';
```

- hospitalisations :

```
1 CREATE EXTERNAL TABLE hospitalisations(num_hospitalisation BIGINT, id_patient BIGINT,  
2 date_entree STRING, jour_hospitalisation BIGINT, code_diagnostic STRING)  
3 ROW FORMAT DELIMITED FIELDS TERMINATED BY ';' ;  
4 STORED AS TEXTFILE  
5 LOCATION '/user/hive/data';  
6  
7 LOAD INPATH '/user/cloudera/student/Hospitalisations.txt';
```


- patient :

```
1 CREATE EXTERNAL TABLE patient(id_patient BIGINT, nom STRING,  
2 prenom STRING, sexe STRING, pays STRING, date STRING, age BIGINT)  
3 ROW FORMAT DELIMITED FIELDS TERMINATED BY ';' ;  
4 STORED AS TEXTFILE  
5 LOCATION '/user/hive/data';  
6  
7 LOAD INPATH '/user/cloudera/student/Patient.txt';|
```

professionnel de sante :

```
1 CREATE EXTERNAL TABLE professionnel_de_sante(identifiant BIGINT, nom STRING,  
2 prenom STRING)  
3 ROW FORMAT DELIMITED FIELDS TERMINATED BY ';' ;  
4 STORED AS TEXTFILE  
5 LOCATION '/user/hive/data';  
6  
7 LOAD INPATH '/user/cloudera/student/Professionnel_de_sante.txt';|
```

- satisfaction :

```
1 CREATE EXTERNAL TABLE satisfaction(id BIGINT, region STRING, score1 STRING,  
2 score2 STRING)  
3 ROW FORMAT DELIMITED FIELDS TERMINATED BY ';' ;  
4 STORED AS TEXTFILE  
5 LOCATION '/user/hive/data';  
6  
7 LOAD INPATH '/user/cloudera/student/Satisfaction.txt';|
```

- faits_consulter :

```
1 CREATE EXTERNAL TABLE faits_consulter(num_consultation BIGINT, id_patient BIGINT,  
2 identifiant BIGINT, code_diag STRING)  
3 ROW FORMAT DELIMITED FIELDS TERMINATED BY ';' ;  
4 STORED AS TEXTFILE  
5 LOCATION '/user/hive/data';  
6  
7 LOAD INPATH '/user/cloudera/student/Faits_consulter.txt';|
```

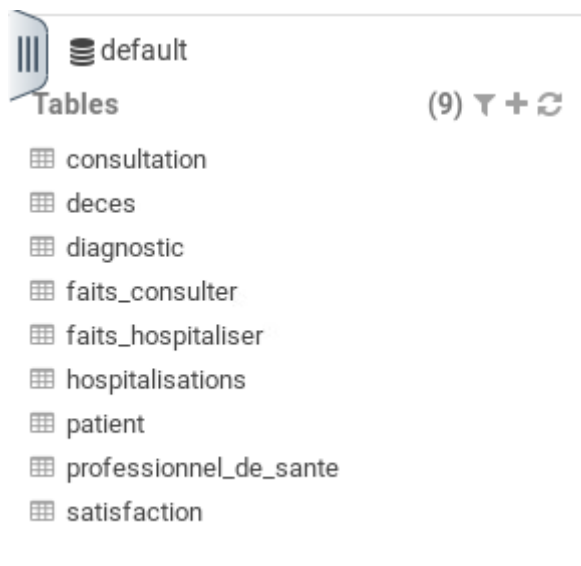
- faits_hospitaliser :

```

1 CREATE EXTERNAL TABLE faits_hospitaliser(num_hospitalisation BIGINT,
2 code_diag STRING, id_patient BIGINT)
3 ROW FORMAT DELIMITED FIELDS TERMINATED BY ';'
4 STORED AS TEXTFILE
5 LOCATION '/user/hive/data';
6
7 LOAD INPATH '/user/cloudera/student/Faits_hospitaliser.txt';

```

II. Vérification des données présentes et accès aux données à travers les tables :



- consultation :

	num_consultation	id_patient	id_prof_sante	code_diag	heure_debut	heure_fin
1	1059023406	272	10004985825	P041	01:00:00	06:00:00
2	1059023407	1071	10005233548	N744	01:00:00	06:00:00
3	1059023408	1285	10101362548	S92700	08:00:00	12:00:00
4	1059023409	3723	10003697488	M2532	01:00:00	06:00:00
5	1059023410	3841	10000436864	M512	01:00:00	06:00:00
6	1059023411	4092	10002478203	J069	01:00:00	06:00:00
7	1059023412	4164	10003737219	L752	01:00:00	06:00:00
8	1059023413	4477	10003330551	M0227	08:00:00	12:00:00
9	1059023414	4709	10100154573	M4140	08:00:00	12:00:00

- deces :

	numero_acte_deces	lieu_naissance	code_lieu_naissance	date_deces	code_lieu_deces
1	369	HOMBLIERES	2383	1983-04-11	2691
2	219	HOMBLIERES	2383	1983-02-28	2691
3	33	HOMBLIERES	2383	1983-10-30	2504
4	6	HOURY	2384	1983-04-13	2668
5	4-97	HOURY	2384	1983-02-14	44109
6	1075	HOURY	2384	1983-11-19	2691
7	36	HOUSSET	2385	1983-03-09	2361
8	105	HOUSSET	2385	1983-02-04	60159
9	599	HOUSSET	2385	1983-12-01	2408

- diagnostic :

	code_diag	diagnostic
1	A066	Abces amibien du cerveau
2	A064	Abces amibien du foie
3	A065	Abces amibien du poumon
4	K610	Abces anal
5	K612	Abces anorectal
6	L028	Abces cutane, furoncle et anthrax d'autres localisations
7	L020	Abces cutane, furoncle et anthrax de la face
8	L023	Abces cutane, furoncle et anthrax de la fesse
9	L021	Abces cutane, furoncle et anthrax du cou

- hospitalisations :

	num_hospitalisation	id_patient	date_entree	jour_hospitalisation	code_diagnostic
1	10546	29620	27/09/2017	17	S02800
2	52968	21126	16/08/2020	27	Q902
3	46019	4572	16/01/2016	1	R192
4	59194	60237	24/03/2018	5	M1125
5	75758	53831	16/04/2020	6	Y648
6	70998	42859	14/04/2021	4	C754
7	10728	34974	05/08/2018	28	S62471
8	63219	41726	31/12/2015	28	S92901
9	59938	83931	27/07/2017	24	M4850

- patient :

	id_patient	nom	prenom	sexe	pays	date	age
1	1	Christabel	Tougas	female	FR	4/6/1980	41
2	2	Lorraine	Lebel	female	FR	7/25/2013	7
3	3	Jolie	Majory	female	FR	8/8/2009	11
4	4	Agate	Chalut	female	FR	10/18/1957	63
5	5	D'Arcy	Casgrain	male	FR	10/4/1965	55
6	6	Montague	Laforest	male	FR	12/8/1939	81
7	7	Gérard	Bonnet	male	FR	7/29/1950	70
8	8	Fifine	Mainville	female	FR	8/26/1945	75
9	9	Turner	Mazuret	male	FR	1/17/1925	96

- professionnel_de_sante :

	identifiant	nom	prenom
1	826354284	DECLERCQ	Sophie
2	826354318	PRESANI	Christelle
3	826354342	RAFFIN	Lyonnelle
4	826354359	SOTELO FERNANDEZ	Iria
5	826354367	HACQUES	Nathalie
6	826354375	BORIES	Manon
7	826354409	PLANTADE	Corinne
8	826354482	DELLA DONNA	Fanny
9	826354524	CORNELISSEN	Pauline

- satisfaction :

	id	region	score1	score2
1	10780054	Auvergne-Rhône-Alpes	73,94	76,56
2	10780062	Auvergne-Rhône-Alpes	70,93	79,83
3	10008407	Auvergne-Rhône-Alpes	75,05	82,8
4	10780195	Auvergne-Rhône-Alpes	73,16	75,59
5	10780203	Auvergne-Rhône-Alpes	71,54	78,65
6	20000063	Hauts de France	69,43	76,63
7	20000360	Hauts de France	77,69	77,17
8	20000253	Hauts de France	71,39	76,22
9	20000261	Hauts de France	71,07	79,83

- faits_consulter :

	num_consultation	id_patient	identifiant	code_diag
1	1059023406	272	10004985825	P041
2	1059023407	1071	10005233548	N744
3	1059023408	1285	10101362548	S92700
4	1059023409	3723	10003697488	M2532
5	1059023410	3841	10000436864	M512
6	1059023411	4092	10002478203	J069
7	1059023412	4164	10003737219	L752
8	1059023413	4477	10003330551	M0227
9	1059023414	4709	10100154573	M4140
10	1059023415	4789	10001057420	Y613

- faits_hospitaliser :

	num_hospitalisation	code_diag	id_patient
1	10546	S02800	29620
2	52968	Q902	21126
3	46019	R192	4572
4	59194	M1125	60237
5	75758	Y648	53831
6	70998	C754	42859
7	10728	S62471	34974
8	63219	S92901	41726
9	59938	M4850	83931

III. Script montrant le peuplement des tables :

Voici un exemple d'un script montrant le peuplement de la table « consultation » :

```
1 SELECT * FROM consultation
```

IV. Script montrant le partitionnement et les buckets :

- patient_age :

Nous avons créé une table partitionnée patient par age avec quatre buckets trie par l'identifiant du patient .

```

1 CREATE TABLE patient_age
2 (id_patient BIGINT, nom STRING, prenom STRING, sexe STRING, pays STRING, date STRING)
3 PARTITIONED BY (age string)
4 CLUSTERED BY (id_patient) into 4 BUCKETS
5 ROW FORMAT DELIMITED
6 FIELDS TERMINATED BY "\;"
7 STORED AS TEXTFILE ;|

















```

```





1 INSERT OVERWRITE TABLE patient_age
2 PARTITION (age)
3 SELECT id_patient, nom, prenom, sexe, pays, date
4 FROM patient ;|

```

[Home](#) / [user](#) / [hive](#) / [warehouse](#) / [projet.db](#) / **patient_age1**

<input type="checkbox"/>	Name	Size	User
<input type="checkbox"/>	 ↑		clouder
<input type="checkbox"/>	 .		clouder
<input type="checkbox"/>	 age=0		clouder
<input type="checkbox"/>	 age=1		clouder
<input type="checkbox"/>	 age=10		clouder
<input type="checkbox"/>	 age=100		clouder
<input type="checkbox"/>	 age=11		clouder
<input type="checkbox"/>	 age=12		clouder
<input type="checkbox"/>	 age=13		clouder
<input type="checkbox"/>	 age=14		clouder
<input type="checkbox"/>	 age=15		clouder
<input type="checkbox"/>	 age=16		clouder
<input type="checkbox"/>	 age=17		clouder
<input type="checkbox"/>	 age=18		clouder
<input type="checkbox"/>	 age=19		clouder
<input type="checkbox"/>	 age=2		clouder

[Home](#) / [user](#) / [hive](#) / [warehouse](#) / [projet.db](#) / [patient_age2](#) / **age=10**

<input type="checkbox"/>	Name	Size	User
<input type="checkbox"/>	 ↑		cloudera
<input type="checkbox"/>	 .		cloudera
<input type="checkbox"/>	 000000_0	9.9 KB	cloudera
<input type="checkbox"/>	 000001_0	9.9 KB	cloudera
<input type="checkbox"/>	 000002_0	9.8 KB	cloudera
<input type="checkbox"/>	 000003_0	10.5 KB	cloudera

Show of 4 items

- patient_sexe :

Nous avons créé une table partitionnée patient par sexe afin de créer deux partitions mâle et femelle avec quatre buckets triés par l'identifiant du patient.

```
1 CREATE TABLE patient_sexe
2 (id_patient BIGINT, nom STRING, prenom STRING, pays STRING, date STRING, age BIGINT)
3 PARTITIONED BY (sexe string)
4 CLUSTERED BY (id_patient) INTO 4 BUCKETS
5 ROW FORMAT DELIMITED
6 FIELDS TERMINATED BY "\;"
7 STORED AS TEXTFILE ;|
```

```
1 INSERT OVERWRITE TABLE patient_sexe
2 PARTITION (sexe)
3 SELECT id_patient, nom, prenom, pays, date, age
4 FROM patient ;|
```

[Home](#) / [user](#) / [hive](#) / [warehouse](#) / [projet.db](#) / [patient_sexe1](#)

<input type="checkbox"/>	Name	Size	Use
<input type="checkbox"/>	↑		clou
<input type="checkbox"/>	.		clou
<input type="checkbox"/>	sexe=female		clou
<input type="checkbox"/>	sexe=male		clou
Show <input type="text" value="200"/> of 2 items			

- consultation_periode :

Nous avons créé une table partitionnée consultation par période avec quatre buckets trie par numéro de consultation.

```
1 CREATE TABLE consultation_periode
2 (num_consultation BIGINT, id_patient BIGINT, id_prof_sante BIGINT,
3 code_diag STRING, heure_debut STRING, heure_fin STRING)
4 PARTITIONED BY (periode_de_temps string)
5 CLUSTERED BY (num_consultation) INTO 4 BUCKETS
6 ROW FORMAT DELIMITED
7 FIELDS TERMINATED BY "\;"
8 STORED AS TEXTFILE ;
```

```
1 INSERT OVERWRITE TABLE consultation_periode
2 PARTITION (periode_de_temps)
3 SELECT num_consultation, id_patient, id_prof_sante, code_diag, heure_debut,
4 heure_fin, periode_de_temps
5 FROM consultation ;
```


[Home](#) / [user](#) / [hive](#) / [warehouse](#) / [projet.db](#) / **consultation_periode1**

<input type="checkbox"/>	Name	Size	User
<input type="checkbox"/>	↑		cloudera
<input type="checkbox"/>	.		cloudera
<input type="checkbox"/>	periode_de_temps=01%3A00%3A00-06%3A00%3A00		cloudera
<input type="checkbox"/>	periode_de_temps=08%3A00%3A00-02%3A00%3A00		cloudera
<input type="checkbox"/>	periode_de_temps=08%3A00%3A00-11%3A00%3A00		cloudera
<input type="checkbox"/>	periode_de_temps=08%3A00%3A00-12%3A00%3A00		cloudera

Show of 4 items

- hospitalisations_periode :

Nous avons créé une table partitionnée hospitalisations par période avec quatre buckets trie par numéro d'hospitalisation.

```

1 CREATE TABLE hospitalisations_periode
2 (num_Hospitalisation BIGINT, id_patient BIGINT, code_diagnostic STRING,
3 Date_Entree STRING)
4 PARTITIONED BY (jour_hospitalisation string)
5 CLUSTERED BY (num_Hospitalisation) into 4 BUCKETS
6 ROW FORMAT DELIMITED
7 FIELDS TERMINATED BY "\;"
8 STORED AS TEXTFILE ;

```

```

1 INSERT OVERWRITE TABLE hospitalisations_periode
2 PARTITION (jour_hospitalisation)
3 SELECT num_Hospitalisation, id_patient, code_diag, date_entree, jour_hospitalisation
4 FROM hospitalisations ;|

```

[Home](#) / [user](#) / [hive](#) / [warehouse](#) / [projet.db](#) / **hospitalisations_periode2**

<input type="checkbox"/>	Name	Size	User
<input type="checkbox"/>	↑		cloudera
<input type="checkbox"/>	.		cloudera
<input type="checkbox"/>	jour_hospitalisation=1		cloudera
<input type="checkbox"/>	jour_hospitalisation=10		cloudera
<input type="checkbox"/>	jour_hospitalisation=11		cloudera
<input type="checkbox"/>	jour_hospitalisation=12		cloudera
<input type="checkbox"/>	jour_hospitalisation=13		cloudera
<input type="checkbox"/>	jour_hospitalisation=14		cloudera
<input type="checkbox"/>	jour_hospitalisation=15		cloudera
<input type="checkbox"/>	jour_hospitalisation=16		cloudera
<input type="checkbox"/>	jour_hospitalisation=17		cloudera
<input type="checkbox"/>	jour_hospitalisation=18		cloudera
<input type="checkbox"/>	jour_hospitalisation=19		cloudera
<input type="checkbox"/>	jour_hospitalisation=2		cloudera

- satisfaction_region :

Nous avons créé une table satisfaction par région avec quatre buckets trie par finess.

```

1 CREATE TABLE satisfaction_region
2 (fitness BIGINT, score DOUBLE)
3 PARTITIONED BY (region string)
4 CLUSTERED BY (fitness) into 4 BUCKETS
5 ROW FORMAT DELIMITED
6 FIELDS TERMINATED BY ";"
7 STORED AS TEXTFILE ;|

```

```

1 INSERT OVERWRITE TABLE satisfaction_region
2 PARTITION (region)
3 SELECT fitness, score, region
4 FROM satisfaction ;|

```

[Home](#) / [user](#) / [hive](#) / [warehouse](#) / [projet.db](#) / [satisfaction_reg1](#)

<input type="checkbox"/>	Name	Size	User
<input type="checkbox"/>	↑		cloudera
<input type="checkbox"/>	.		cloudera
<input type="checkbox"/>	region=Auvergne-Rhône-Alpes		cloudera
<input type="checkbox"/>	region=Bourgogne-Franche-Comté		cloudera
<input type="checkbox"/>	region=Bretagne		cloudera
<input type="checkbox"/>	region=Centre-Val de Loire		cloudera
<input type="checkbox"/>	region=Corse		cloudera
<input type="checkbox"/>	region=Grand Est		cloudera
<input type="checkbox"/>	region=Guadeloupe		cloudera
<input type="checkbox"/>	region=Guyane		cloudera
<input type="checkbox"/>	region=Hauts de France		cloudera
<input type="checkbox"/>	region=Ile de France		cloudera
<input type="checkbox"/>	region=Martinique		cloudera
<input type="checkbox"/>	region=Normandie		cloudera

- faits_consulter_buckets :

Nous avons devisé la table des faits consulter par quatre buckets trie par numéro de consultation sans partitions.

```

1 CREATE TABLE faits_consulter_buckets
2 (num_consultation BIGINT, id_patient BIGINT,identifiant BIGINT,code_diag STRING)
3 CLUSTERED BY (num_consultation) INTO 4 BUCKETS
4 ROW FORMAT DELIMITED
5 FIELDS TERMINATED BY ";"
6 STORED AS TEXTFILE ;

```

```

1 SELECT num_consultation , id_patient , identifiant ,code_diag
2 FROM satisfaction ;
3 INSERT OVERWRITE TABLE faits_consulter2
4 FROM faits_consulter;|

```

[Home](#) / [user](#) / [hive](#) / [warehouse](#) / [projet.db](#) / **faits_consulter2**

<input type="checkbox"/>	Name	Size	User
<input type="checkbox"/>	 ↑		cloud
<input type="checkbox"/>	 .		cloud
<input type="checkbox"/>	 000000_0	8.4 MB	cloud
<input type="checkbox"/>	 000001_0	8.4 MB	cloud
<input type="checkbox"/>	 000002_0	8.4 MB	cloud
<input type="checkbox"/>	 000003_0	8.4 MB	cloud

Show of 4 items

V. Graphes montrant les temps de réponses pour évaluer la performance d'accès à l'entrepôt de données :

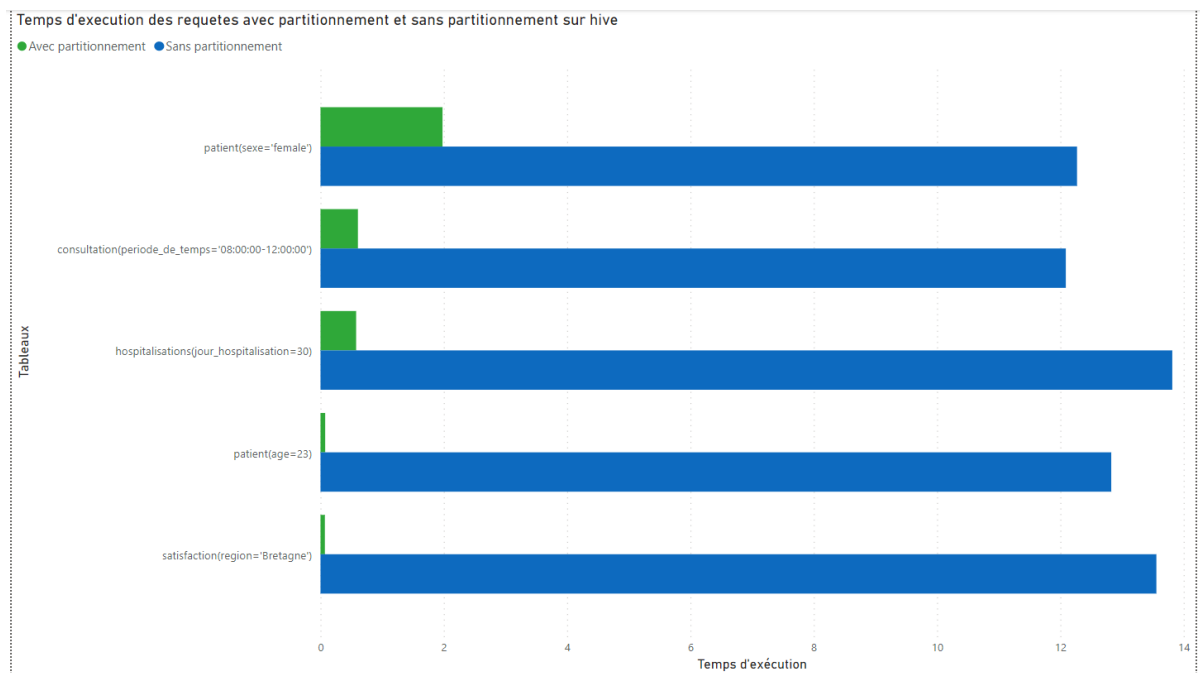


tableau	avec_part	sans_part
consultation(periode_de_temps='08:00:00-12:00:00')	0,607	12,088
hospitalisations(jour_hospitalisation=30)	0,578	13,815
satisfaction(region='Bretagne')	0,071	13,557
patient(sexe='female')	1,979	12,27
patient(age=23)	0,076	12,825

VI. Requêtes faisant foi pour l'évaluation de la performance:

Nous avons testé quelques requêtes pour tester les performances et voici les résultats obtenus :

- Patient « femme » :
 - Sans partitions : 12,27s

```
1|select from projet.patient where sexe="female" limit 500;|
```

- Avec partitions : 1,979

```
1|select from projet.patient_sexe where sexe="female" limit 500;|
```

- Patient « 23 ans » :
 - Sans partitions : 12,825s

```
1|select from projet.patient where age=23 limit 500;|
```

- Avec partitions : 0,076s

```
1|select from projet.patient_age where age=23 limit 500;|
```

- Satisfaction « Bretagne » :
 - Sans partitions : 13,557s

```
1|select from projet.satisfaction where region="Bretagne" limit 500;|
```

- Avec partitions : 0,071s

```
1|select from projet.satisfaction_region where region="Bretagne" limit 500;|
```

- Hospitalisation « jour hospitalisation=30 » :
 - Sans partitions : 13,815s

```
1|select from projet.hospitalisations where jour_hospitalisation=30 limit 500;|
```

- Avec partitions : 0,578s

```
1|select from projet.hospitalisations_periode where jour_hospitalisation=30 limit 500;|
```

Conclusion :

Pendant cette phase de travail, nous avons étudié le fonctionnement du partitionnement et du clustering, ainsi que leurs bénéfices. Comme le démontre notre graphique, l'utilisation du partitionnement et du clustering a considérablement amélioré les temps d'exécution des requêtes.