----------Scripts de construction de la Datalake----------

Ensemble d'instructions pour charger des données, effectuer des transformations et des analyses à l'aide d'outils comme HDFS, MongoDB, Hive et HBase.

1.1 Chargement des fichiers Excel « CO2.csv » et « Catalogue.csv » dans HDFS :

- Vous démarrez HDFS et YARN :

```
[vagrant@oracle-21c-vagrant] start-dfs.sh
[vagrant@oracle-21c-vagrant] start-yarn.sh
```

- Vous listez le contenu du système de fichiers HDFS : hdfs dfs -ls
- Vous créez un répertoire nommé "MBDS_Projet" dans HDFS : hdfs dfs -mkdir /MBDS_Projet
- Vous copiez plusieurs fichiers CSV dans ce repertoire :

```
hadoop fs -put /vagrant/Groupe_TPT_8/CO2.csv  /MBDS_Projet

hadoop fs -put /vagrant/Groupe_TPT_8/Catalogue.csv  /MBDS_Projet
```

- Vous listez le contenu du répertoire "MBDS_Projet" pour vérification :

```
hadoop fs -ls /MBDS_Projet
```

1.2 Chargement des fichiers Excel « Clients_12.csv » et « Clients_7.csv » dans MongoDB :

- Vous importez des fichiers CSV dans une base de données MongoDB nommée "Clients" :

[vagrant@oracle-21c-vagrant ~]$ mongoimport -d Clients -c Clients_7 --type csv --file /vagrant/Groupe_TPT_8/Clients_7.csv –headerline

[vagrant@oracle-21c-vagrant ~]$ mongoimport -d Clients -c Clients_12 --type csv –file /vagrant/Groupe_TPT_8/Clients_12.csv –headerline

- Vous effectuez des opérations de renommage sur certaines colonnes des collections importées :

[vagrant@oracle-21c-vagrant ~]$ mongo

```
> use Clients
switched to db Clients
> db.Clients_7.find()
> db.Clients_12.find()
```

1.3 Chargement du fichier Excel « Immatriculations.csv » dans HBase :

- Vous démarrez HBase :

[vagrant@oracle-21c-vagrant] start-hbase.sh

- Vous créez une table HBase nommée "Immatriculations" avec une colonne de famille "cf" :

```
hbase:006:0> Create 'Immatriculations', 'cf'
```

- Vous importez un fichier CSV dans cette table HBase :

```
vagrant@oracle-21c-vagrant ~]$ hbase
org.apache.hadoop.hbase.mapreduce.ImportTsv -
Dimporttsv.separator=',' -
Dimporttsv.columns=HBASE_ROW_KEY,cf:marque,cf:nom,cf:puissance,cf:l
ongueur,cf:nbPlaces,cf:nbPortes,cf:couleur,cf:occasion,cf:prix
Immatriculations /vagrant/Groupe_TPT_8/Immatriculations.csv
```

- Vous vérifiez le contenu de la table HBase :

```
hbase:006:0>scan 'Immatriculations'
hbase:006:0>count 'Immatriculations'
1996633
```

Après plusieurs tentatives, Il semble que 1 996 633 enregistrements aient été chargés au lieu des 2 millions existants dans la table Excel, ce qui représente un écart d'environ 3400 enregistrements. Cet écart peut être considéré comme négligeable.

1.4 Chargement du fichier Excel « Marketing.csv » dans la base de donnée MySQL :

- Connectez-vous à la base de données MySQL en tant que superutilisateur (root) :

```
[vagrant@oracle-21c-vagrant ~]$  sudo mysql
```

- Création de la base de données et de la table MySQL :

```
mysql > CREATE DATABASE  Marketing_MBDS ;

mysql> USE Marketing_MBDS;
```

```
mysql> CREATE TABLE Marketing_Db ( age INT,  sexe VARCHAR(1),  taux INT,  situationFamiliale VARCHAR(20),  nbEnfantsAcharge INT,  deuxieme_voiture VARCHAR(5) );
```

- Chargement des données dans la table MySQL :

```
mysql> Load data local infile '/vagrant/Groupe_TPT_8/Marketing.csv' Into table Marketing_Db fields terminated by ','lines terminated by '\n' ignore 1 rows;

ERROR 3948 (42000): Loading local data is disabled; this must be enabled on both the client and server sides
```

- Activation de la variable global **'local_infile'** pour permettre le chargement des données à partir d'un fichier Excel sur MySQL :

```
mysql> SHOW GLOBAL VARIABLES LIKE 'local_infile';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| local_infile  | OFF   |
+---------------+-------+
1 row in set (0.00 sec)

mysql> SET GLOBAL local_infile='ON';
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW GLOBAL VARIABLES LIKE 'local_infile';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| local_infile  | ON    |
+---------------+-------+
1 row in set (0.01 sec)
```

- Chargement des données dans MySQL :

```
mysql> Load data local infile '/vagrant/Groupe_TPT_8/Marketing.csv' Into table Marketing_Db fields terminated by ','lines terminated by '\n' ignore 1 rows;
```

Query OK, 20 rows affected, 15 warnings (0.01 sec)

Records: 20 Deleted: 0 Skipped: 0 Warnings: 15

mysql> SELECT * FROM Marketing_Db \G; | sed 's/\r//';

```
*************************** 3. row ***************************
            age: 48
           sexe: M
           taux: 401
situationFamiliale: Célibataire
 nbEnfantsAcharge: 0
 deuxieme_voiture: false
*************************** 4. row ***************************
            age: 26
           sexe: F
           taux: 420
situationFamiliale: En Couple
 nbEnfantsAcharge: 3
 deuxieme_voiture: true
*************************** 5. row ***************************
            age: 80
           sexe: M
           taux: 530
situationFamiliale: En Couple
 nbEnfantsAcharge: 3
 deuxieme_voiture: false
*************************** 6. row ***************************
            age: 27
           sexe: F
           taux: 153
situationFamiliale: En Couple
 nbEnfantsAcharge: 2
 deuxieme_voiture: false
*************************** 7. row ***************************
            age: 59
           sexe: F
           taux: 572
situationFamiliale: En Couple
 nbEnfantsAcharge: 2
 deuxieme_voiture: false
*************************** 8. row ***************************
            age: 43
           sexe: F
           taux: 431
situationFamiliale: Célibataire
 nbEnfantsAcharge: 0
 deuxieme_voiture: false
*************************** 9. row ***************************
            age: 64
           sexe: M
           taux: 559
situationFamiliale: Célibataire
 nbEnfantsAcharge: 0
 deuxieme_voiture: false
*************************** 10. row ***************************
            age: 22
           sexe: M
           taux: 154
situationFamiliale: En Couple
```

```
mysql> SELECT COUNT(*) FROM Marketing_Db;
+----------+
| COUNT(*) |
+----------+
|       20 |
+----------+
1 row in set (0.40 sec)

mysql>
```

2.1 Extraction des données d'immatriculations depuis HBase vers Hive :

- Vous créez une table externe dans Hive pour mapper les données depuis HBase et vous spécifiez les correspondances entre les colonnes HBase et les colonnes Hive :

```
[vagrant@oracle-21c-vagrant] nohup hive --service metastore >
hive_metastore.log 2>&1 &
[vagrant@oracle-21c-vagrant] nohup hiveserver2 > hive_server.log 2>&1 &
[vagrant@oracle-21c-vagrant] beeline -u jdbc:hive2://localhost:10000 vagrant

0: jdbc:hive2://localhost:10000> CREATE DATABASE IF NOT EXISTS
MBDS_Projet;

0: jdbc:hive2://localhost:10000> USE MBDS_Projet;

0: jdbc:hive2://localhost:10000> CREATE EXTERNAL TABLE
table_ext_immatriculations (
    immatriculation STRING,
    marque STRING,
    nom STRING,
    puissance INT,
    longueur STRING,
    nbPlaces INT,
    nbPortes INT,
    couleur STRING,
    occasion BOOLEAN,
    prix INT
)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
WITH SERDEPROPERTIES (
   "hbase.columns.mapping" =
":key,cf:marque,cf:nom,cf:puissance,cf:longueur,cf:nbPlaces,cf:nbPortes,
cf:couleur,cf:occasion,cf:prix"
)
TBLPROPERTIES("hbase.table.name" = "Immatriculations");
```

- Vérification du chargement des données :

```
0: jdbc:hive2://localhost:10000> Select * from
table_ext_immatriculations  LIMIT 40;
0: jdbc:hive2://localhost:10000> Select COUNT(*) from
table_ext_immatriculations; ==> 1996633
```

2.2 Extraction des données d'immatriculations depuis MongoDB vers Hive :

- Vous créez des tables externes dans Hive pour mapper les données depuis MongoDB.
- Vous spécifiez les correspondances entre les colonnes MongoDB et les colonnes Hive.

```
0: jdbc:hive2://localhost:10000> CREATE EXTERNAL TABLE client_7_ext (
    age INT,
    sexe STRING,
    taux INT,
    situationFamiliale STRING,
    nbEnfantsAcharge INT,
    deuxiemevoiture STRING,
    immatriculation STRING
)
STORED BY 'com.mongodb.hadoop.hive.MongoStorageHandler'
WITH SERDEPROPERTIES('mongo.columns.mapping'='{
    "age":"age",
    "sexe":"sexe",
    "taux":"taux",
    "situationFamiliale":"situationFamiliale",
    "nbEnfantsAcharge":"nbEnfantsAcharge",
    "deuxiemevoiture":"2eme voiture",
    "immatriculation":"immatriculation"
}')
TBLPROPERTIES('mongo.uri'='mongodb://localhost:27017/Clients.Clients
_7');


0: jdbc:hive2://localhost:10000> SELECT * FROM client_7_ext LIMIT 10;


0: jdbc:hive2://localhost:10000> SELECT COUNT(*) FROM client_7_ext;
➔ 100 000

0: jdbc:hive2://localhost:10000>CREATE EXTERNAL TABLE client_12_ext (
    age INT,
    sexe STRING,
```

```
    taux INT,
    situationFamiliale STRING,
    nbEnfantsAcharge INT,
    deuxiemeVoiture STRING,
    immatriculation STRING
)
STORED BY 'com.mongodb.hadoop.hive.MongoStorageHandler'
WITH SERDEPROPERTIES('mongo.columns.mapping'='{
    "age":"age",
    "sexe":"sexe",
    "taux":"taux",
    "situationFamiliale":"situationFamiliale",
    "nbEnfantsAcharge":"nbEnfantsAcharge",
    "deuxiemevoiture":"2eme voiture",
    "immatriculation":"immatriculation"
}')
TBLPROPERTIES('mongo.uri'='mongodb://localhost:27017/Clients.Clients
_12');

0: jdbc:hive2://localhost:10000> SELECT * FROM client_12_ext LIMIT 10;
0: jdbc:hive2://localhost:10000> SELECT COUNT(*) FROM client_12_ext;
➔100 000
```

2.3 Transfert des résultats de la tâche Map-Reduce vers Hive :

- Vous configurez une table dans Hive pour accueillir les résultats du traitement Map-Reduce nommé "resultat_Catalogue_CO2.csv" effectué sur les fichiers Excel "Catalogue.csv" et "CO2.csv", préalablement chargés dans HDFS.

```
0: jdbc:hive2://localhost:10000> CREATE TABLE IF NOT EXISTS
resultat_catalogue_co2 (
    marque STRING,
    nom STRING,
    puissance INT,
    longueur STRING,
    nbPlaces INT,
    nbPortes INT,
    couleur STRING,
    occasion BOOLEAN,
    prix INT,
```

```
    moyenne_bonus_malus DOUBLE,
    moyenne_rejets_co2 DOUBLE,
    cout_energie_moyen DOUBLE
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
TBLPROPERTIES ("skip.header.line.count"="1");
```

- Vous importez les données du fichier résultat "resultat_Catalogue_CO2.csv", qui ont été préalablement chargées dans HDFS, dans cette table :

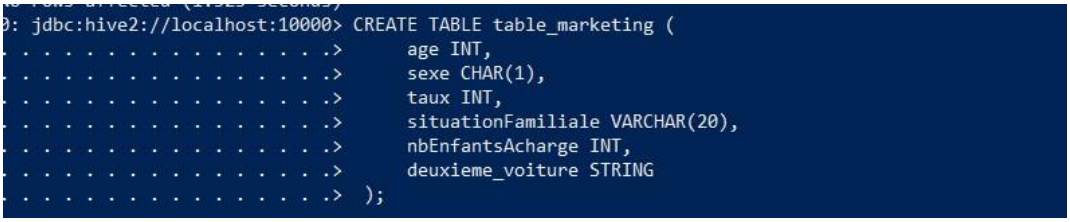
```
0: jdbc:hive2://localhost:10000> LOAD DATA INPATH
'/MBDS_Projet/resultat_Catalogue_CO2.csv' OVERWRITE INTO TABLE
resultat_catalogue_co2;
```

• Vous effectuez une vérification pour vous assurer que les données ont été correctement chargées et vous comptez le nombre de lignes dans la table :

```
0: jdbc:hive2://localhost:10000> Select * from resultat_catalogue_co2;

0: jdbc:hive2://localhost:10000> Select COUNT(*) from
resultat_catalogue_co2; ==> 270
```

2.4 Extraction des données Marketing depuis MySQL vers Hive :

- Création d'une table Hive destinée à contenir les données marketing :



- Utilisation de l'ELT **Sqoop** pour copier les données depuis la table MySQL «Marketing_Db» vers la table Hive «table_marketing» :

```
[vagrant@oracle-21c-vagrant ~]$
[vagrant@oracle-21c-vagrant ~]$ sqoop import -D org.apache.sqoop.splitter.allow_text_splitter=true \
> --connect jdbc:mysql://localhost:3306/Marketing_MBDS?characterEncoding=latin1 \
> --driver com.mysql.cj.jdbc.Driver --target-dir /user/vagrant/Marketing_MBDS.Marketing_Db  \
> --username root \
> --table Marketing_Db \
> --fields-terminated-by ',' \
> --lines-terminated-by '\n' \
> --hive-import \
> --hive-table MBDS_Projet.table_marketing \
> -m 1
```

- Vérification du chargement des données dans la table Hive :

```
0: jdbc:hive2://localhost:10000> SELECT * from table_marketing;
```

table_marketing.age	table_marketing.sexe	table_marketing.taux	table_marketing.situationfamiliale	table_marketing.nbenfantsacharge	table_marketing.deuxieme_voiture
21	F	1396	Célibataire	0	false
35	M	223	Célibataire	0	false
48	M	401	Célibataire	0	false
26	F	420	En Couple	3	true
80	M	530	En Couple	3	false
27	F	153	En Couple	2	false
59	F	572	En Couple	2	false
43	F	431	Célibataire	0	false
64	M	559	Célibataire	0	false
22	M	154	En Couple	1	false
79	F	981	En Couple	2	false
55	M	588	Célibataire	0	false
19	F	212	Célibataire	0	false
34	F	1112	En Couple	0	false
60	M	524	En Couple	0	true
22	M	411	En Couple	3	true
58	M	1192	En Couple	0	false
54	F	452	En Couple	3	true
35	M	589	Célibataire	0	false
59	M	748	En Couple	0	true

```
0: jdbc:hive2://localhost:10000> SELECT COUNT(*) from table_marketing;
```

_c0
20

3.1 Nettoyage et transformation des données de la table « table_immatriculations_ext » :

```
0: jdbc:hive2://localhost:10000> DESC table_ext_immatriculations;
```

col_name	data_type	comment
immatriculation	string	
marque	string	
nom	string	
puissance	int	
longueur	string	
nbplaces	int	
nbportes	int	
couleur	string	
occasion	boolean	
prix	int	

0: jdbc:hive2://localhost:10000> SELECT COUNT(*) from table_ext_immatriculations;

_c0
1996633

----------Traitement des valeurs de la colonne « marque »----------

0: jdbc:hive2://localhost:10000> SELECT DISTINCT marque from table_ext_immatriculations;

marque
Audi
BMW
Dacia
Daihatsu
Fiat
Ford
Jaguar
Kia
Lancia
Mercedes
Mini
Nissan
Peugeot
Renault
Saab
Seat
Skoda
Volkswagen
Volvo
marque

```
0: jdbc:hive2://localhost:10000> SELECT COUNT(*) from
table_ext_immatriculations where marque="marque";
```

```
+------+
| _c0  |
+------+
| 1    |
+------+
```

Comme vous pouvez le remarquer, la colonne "marque" contient la valeur "marque" qui n'existe pas le dictionnaire de donnée fournit. Vérifions l'enregistrement en entier correspondant à cette valeur sur HBase :

```
hbase:036:0> scan 'Immatriculations', {FILTER => "SingleColumnValueFilter('cf', 'marque', =, 'binary:marque')", LIMIT =>1}
ROW                           COLUMN+CELL

 immatriculation              column=cf:couleur, timestamp=2024-04-01T18:30:59.881, value=couleur
 immatriculation              column=cf:longueur, timestamp=2024-04-01T18:30:59.881, value=longueur
 immatriculation              column=cf:marque, timestamp=2024-04-01T18:30:59.881, value=marque
 immatriculation              column=cf:nbPlaces, timestamp=2024-04-01T18:30:59.881, value=nbPlaces
 immatriculation              column=cf:nbPortes, timestamp=2024-04-01T18:30:59.881, value=nbPortes
 immatriculation              column=cf:nom, timestamp=2024-04-01T18:30:59.881, value=nom
 immatriculation              column=cf:occasion, timestamp=2024-04-01T18:30:59.881, value=occasion
 immatriculation              column=cf:prix, timestamp=2024-04-01T18:30:59.881, value=prix
 immatriculation              column=cf:puissance, timestamp=2024-04-01T18:30:59.881, value=puissance
1 row(s)
Took 18.8859 seconds
```

La présence de cet enregistrement peut s'expliquer par l'importation involontaire de l'en-tête de notre fichier « Immatriculations.csv » lors de la migration des données vers HBase. Supprimons cet enregistrement :

```
hbase:043:0> deleteall 'Immatriculations','immatriculation'
Took 0.0156 seconds
hbase:044:0> get 'Immatriculations','immatriculation'
COLUMN                                                  CELL
0 row(s)
Took 0.0167 seconds
hbase:045:0>
```

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT marque from
table_ext_immatriculations;
```

marque
Audi
BMW
Dacia
Daihatsu
Fiat
Ford
Jaguar
Kia
Lancia
Mercedes
Mini
Nissan
Peugeot
Renault
Saab
Seat
Skoda
Volkswagen
Volvo

Maintenant la colonne « marque » correspondant au dictionnaire de donnée.

----------Explorations des valeurs de la colonne « nom »----------

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT nom from
table_ext_immatriculations;
```

nom
1007 1.4
120i
9.3 1.8T
A2 1.4
A200
A3 2.0 FSI
Almera 1.8
Copper 1.6 16V
Croma 2.2
Cuore 1.0
Golf 2.0 FSI
Laguna 2.0T
Logan 1.6 MPI
M5
Maxima 3.0 V6
Megane 2.0 16V
Mondeo 1.8
New Beatle 1.8
Picanto 1.1
Polo 1.2 6V
Primera 1.6
S500
S80 T6
Superb 2.8 V6
Toledo 1.6
Vel Satis 3.5 V6
X-Type 2.5 V6
Ypsilon 1.4 16V

----------Exploration des valeurs de la colonne « longueur »----------

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT longueur from
table_ext_immatriculations;
```

```
+--------------+
|   longueur   |
+--------------+
| courte       |
| longue       |
| moyenne      |
| tres longue  |
+--------------+
```

----------Exploration des valeurs de la colonne « couleur » ----------

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT couleur from
table_ext_immatriculations;
```

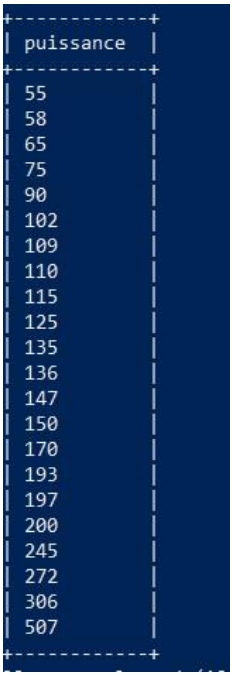
```
+-----------+
| couleur   |
+-----------+
| blanc     |
| bleu      |
| gris      |
| noir      |
| rouge     |
+-----------+
```

----------Exploration des valeurs de la colonne « occasion » ----------

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT occasion from
table_ext_immatriculations;

+-----------+
| occasion  |
+-----------+
| false     |
| true      |
+-----------+
```

----------Exploration des valeurs de la colonne « puissance »----------

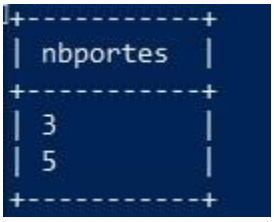
```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT puissance from
table_ext_immatriculations;
```



puissance
55
58
65
75
90
102
109
110
115
125
135
136
147
150
170
193
197
200
245
272
306
507

----------Exploration des valeurs de la colonne « nbPortes »----------

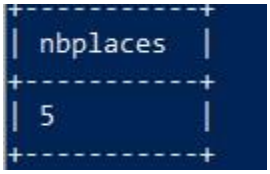
```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT nbPortes from
table_ext_immatriculations;
```



nbportes
3
5

----------Exploration des valeurs de la colonne « nbPlaces »----------

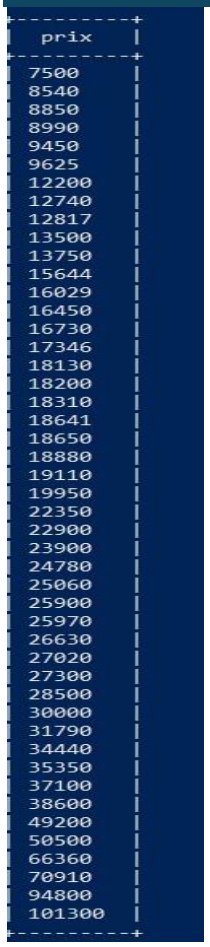
```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT nbPlaces from
table_ext_immatriculations;
```



nbplaces
5

----------Exploration des valeurs de la colonne « prix »----------

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT prix from
table_ext_immatriculations;
```

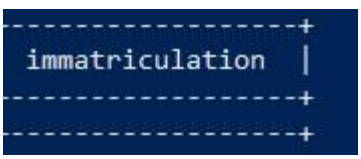


prix
7500
8540
8850
8990
9450
9625
12200
12740
12817
13500
13750
15644
16029
16450
16730
17346
18130
18200
18310
18641
18650
18880
19110
19950
22350
22900
23900
24780
25060
25900
25970
26630
27020
27300
28500
30000
31790
34440
35350
37100
38600
49200
50500
66360
70910
94800
101300

----------Exploration de la colonne « immatriculation »----------

Vérification du format de la colonne « immatriculation »

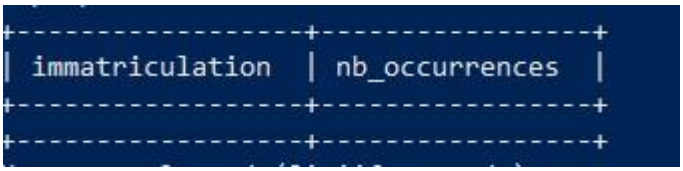
```
0: jdbc:hive2://localhost:10000>SELECT DISTINCT immatriculation

FROM table_ext_immatriculations

WHERE immatriculation NOT REGEXP '^[0-9]{1,4} [A-Z]{2} [0-9]{2}$';
```



immatriculation

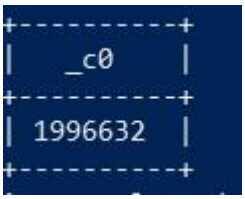
Vérification de l’unicité des valeurs dans la colonne « immatriculations »

```
0: jdbc:hive2://localhost:10000> SELECT immatriculation, COUNT(*) AS nb_occurrences

FROM table_ext_immatriculations

GROUP BY immatriculation

HAVING COUNT(*) > 1;
```



----------Dimension de la table finale----------

```
0: jdbc:hive2://localhost:10000>  SELECT Count(*) from table_ext_immatriculations;
```



Nous pouvons déduire qu'il y avait une seule ligne dans la table Hive "table_immatriculation_ext" contenant des valeurs aberrantes qui correspondaient à l'en-tête du fichier Excel source.

3.2 Exploration des données de la table « resultat_catalogue_co2 » :

```
0: jdbc:hive2://localhost:10000>DESC resultat_catalogue_co2 ;
```

col_name	data_type	comment
marque	string	
nom	string	
puissance	int	
longueur	string	
nbplaces	int	
nbportes	int	
couleur	string	
occasion	boolean	
prix	int	
moyenne_bonus_malus	double	
moyenne_rejets_co2	double	
cout_energie_moyen	double	

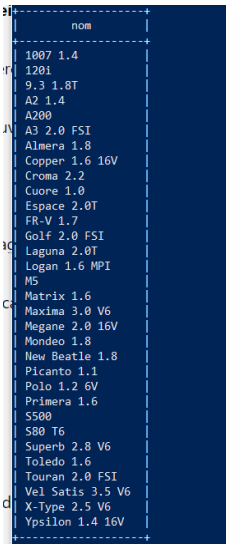
----------Exploration des valeurs de la colonne « marque » ----------

```
0: jdbc:hive2://localhost:10000>Select  DISTINCT marque from resultat_catalogue_co2;
```

marque
Audi
BMW
Dacia
Daihatsu
Fiat
Ford
Honda
Hyundaï
Jaguar
Kia
Lancia
Mercedes
Mini
Nissan
Peugeot
Renault
Saab
Seat
Skoda
Volkswagen
Volvo

----------Exploration des valeurs de la colonne « nom » ----------

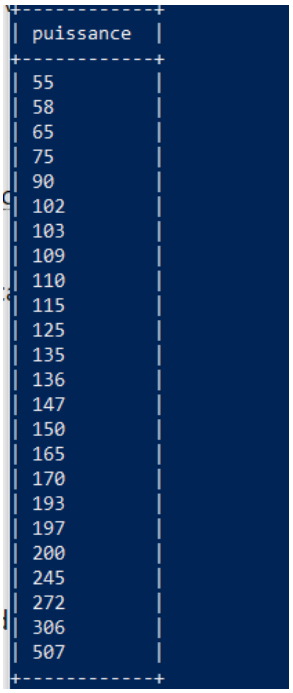
```
0: jdbc:hive2://localhost:10000>Select  DISTINCT nom from resultat_catalogue_co2;
```



nom
1007 1.4
1201
9.3 1.8T
A2 1.4
A200
A3 2.0 FSI
Almera 1.8
Copper 1.6 16V
Croma 2.2
Cuore 1.0
Espace 2.0T
FR-V 1.7
Golf 2.0 FSI
Laguna 2.0T
Logan 1.6 MPI
M5
Matrix 1.6
Maxima 3.0 V6
Megane 2.0 16V
Mondeo 1.8
New Beatle 1.8
Picanto 1.1
Polo 1.2 6V
Primera 1.6
S500
S80 T6
Superb 2.8 V6
Toledo 1.6
Touran 2.0 FSI
Vel Satis 3.5 V6
X-Type 2.5 V6
Ypsilon 1.4 16V

----------Exploration des valeurs de la colonne « puissance » ----------

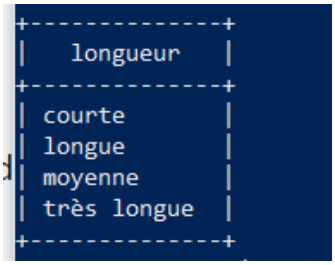
```
0: jdbc:hive2://localhost:10000>Select  DISTINCT puissance from resultat_catalogue_co2;
```



puissance
55
58
65
75
90
102
103
109
110
115
125
135
136
147
150
165
170
193
197
200
245
272
306
507

----------Exploration des valeurs de la colonne « longueur » ----------

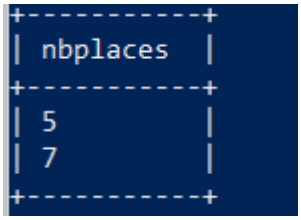
```
0: jdbc:hive2://localhost:10000>Select  DISTINCT longueur from resultat_catalogue_co2;
```



longueur
courte
longue
moyenne
très longue

----------Exploration des valeurs de la colonne « nbplaces » ----------

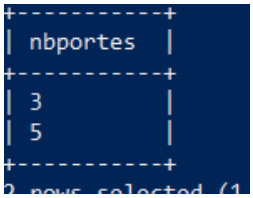
```
0: jdbc:hive2://localhost:10000>Select  DISTINCT nbplaces from resultat_catalogue_co2;
```



nbplaces
5
7

----------Exploration des valeurs de la colonne « nbportes » ----------

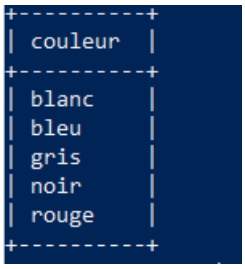
```
0: jdbc:hive2://localhost:10000>Select  DISTINCT nbportes from resultat_catalogue_co2;
```



nbportes
3
5

----------Exploration des valeurs de la colonne « couleur » ----------

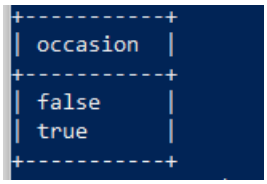
```
0: jdbc:hive2://localhost:10000>Select  DISTINCT couleur from resultat_catalogue_co2;
```



couleur
blanc
bleu
gris
noir
rouge

----------Exploration des valeurs de la colonne « occasion » ----------

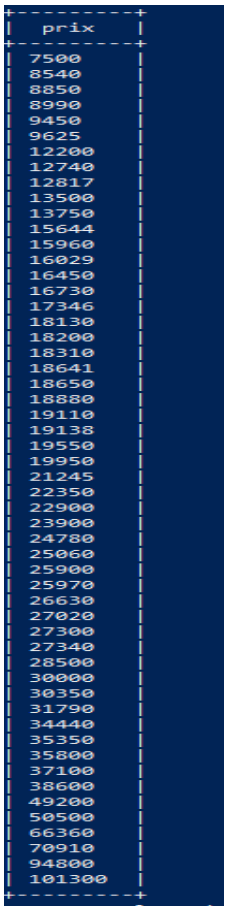
```
0: jdbc:hive2://localhost:10000>Select  DISTINCT occasion from resultat_catalogue_co2;
```



occasion
false
true

----------Exploration des valeurs de la colonne « prix » ----------

```
0: jdbc:hive2://localhost:10000>Select  DISTINCT prix from resultat_catalogue_co2;
```

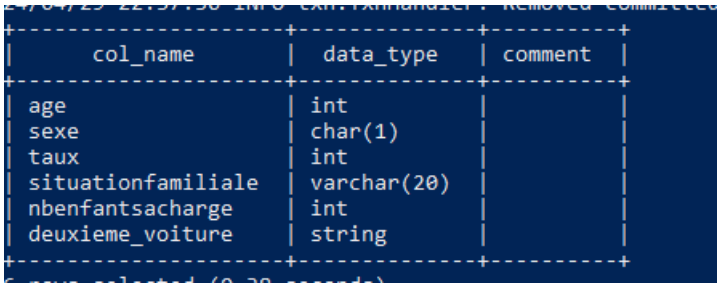


prix
7500
8540
8850
8990
9450
9625
12200
12740
12817
13500
13750
15644
15960
16029
16450
16730
17346
18130
18200
18310
18641
18650
18880
19110
19138
19550
19950
21245
22350
22900
23900
24780
25060
25900
25970
26630
27020
27300
27340
28500
30000
30350
31790
34440
35350
35800
37100
38600
49200
50500
66360
70910
94800
101300

Nous pouvons conclure que toutes les colonnes associées au catalogue respectent le dictionnaire des données.

3.3 Exploration des données de la table « table_marketing» :

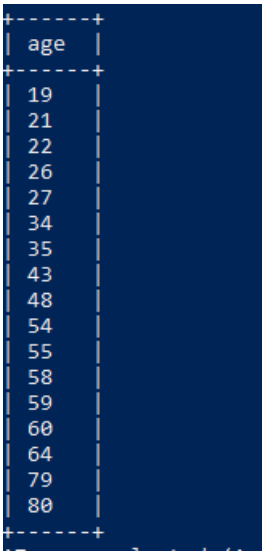
0: jdbc:hive2://localhost:10000>DESC table_marketing ;



col_name	data_type	comment
age	int	
sexe	char(1)	
taux	int	
situationfamiliale	varchar(20)	
nbenfantsacharge	int	
deuxieme_voiture	string	

----------Exploration des valeurs de la colonne « age » ----------

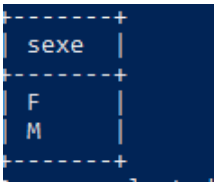
0: jdbc:hive2://localhost:10000>Select DISTINCT age from table_marketing;



age
19
21
22
26
27
34
35
43
48
54
55
58
59
60
64
79
80

----------Exploration des valeurs de la colonne « sexe » ----------

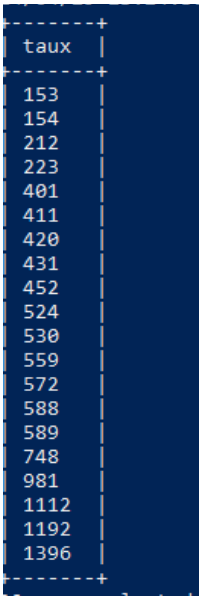
```
0: jdbc:hive2://localhost:10000>Select DISTINCT sexe from table_marketing;
```



sexe
F
M

----------Exploration des valeurs de la colonne « taux » ----------

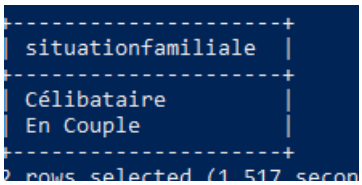
```
0: jdbc:hive2://localhost:10000>Select DISTINCT taux from table_marketing;
```



taux
153
154
212
223
401
411
420
431
452
524
530
559
572
588
589
748
981
1112
1192
1396

----------Exploration des valeurs de la colonne « situationfamiliale » ----------

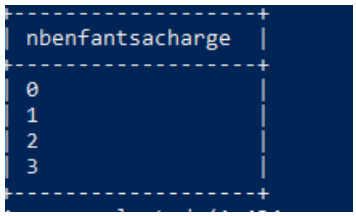
```
0: jdbc:hive2://localhost:10000>Select DISTINCT situationfamiliale from table_marketing;
```



```
+---------------------+
| situationfamiliale  |
+---------------------+
| Célibataire         |
| En Couple           |
+---------------------+
2 rows selected (1.517 secon
```

----------Exploration des valeurs de la colonne « nbenfantsacharge » ----------

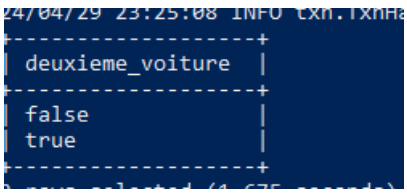
```
0: jdbc:hive2://localhost:10000>Select DISTINCT nbenfantsacharge from table_marketing;
```



```
+--------------------+
| nbenfantsacharge   |
+--------------------+
| 0                  |
| 1                  |
| 2                  |
| 3                  |
+--------------------+
```

----------Exploration des valeurs de la colonne « deuxieme_voiture » ----------

```
0: jdbc:hive2://localhost:10000>Select DISTINCT deuxieme_voiture from table_marketing;
```



```
24/04/29 23:25:08 INFO txn.TxnHa
+--------------------+
| deuxieme_voiture   |
+--------------------+
| false              |
| true               |
+--------------------+
```

Nous pouvons affirmer que toutes les colonnes relatives au marketing sont conformes au dictionnaire des données.

3.4 Transformation et nettoyage des données de la table «client_7_ext » :

3.4.1 Description de la table « client_7_ext » :

col_name	data_type	comment
age	int	from deserializer
sexe	string	from deserializer
taux	int	from deserializer
situationfamiliale	string	from deserializer
nbenfantsacharge	int	from deserializer
deuxiemevoiture	string	from deserializer
immatriculation	string	from deserializer

3.4.2 Comptage du nombre de valeurs indéfinis dans la table «client_7_ext »

```
0: jdbc:hive2://localhost:10000>SELECT DISTINCT age FROM client_7_ext;
```

age
NULL
-1
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32

```
0: jdbc:hive2://localhost:10000>SELECT COUNT(*) FROM client_7_ext WHERE age = -1 OR age IS NULL;
```

_c0
320

```
0: jdbc:hive2://localhost:10000>SELECT DISTINCT sexe FROM client_7_ext;
```

```
+-----------+
|   sexe    |
+-----------+
|           |
| ?         |
| F         |
| Femme     |
| Féminin   |
| Homme     |
| M         |
| Masculin  |
| N/D       |
+-----------+
```

```
0: jdbc:hive2://localhost:10000>SELECT
   SUM(CASE WHEN sexe = '' THEN 1 ELSE 0 END) AS missing_values,
   SUM(CASE WHEN sexe = '?' THEN 1 ELSE 0 END) AS question_marks,
   SUM(CASE WHEN sexe = 'N/D' THEN 1 ELSE 0 END) AS nd_values
FROM client_7_ext;
```

```
+-----------------+-----------------+------------+
| missing_values  | question_marks  | nd_values  |
+-----------------+-----------------+------------+
| 107             | 93              | 108        |
+-----------------+-----------------+------------+
```

```
0: jdbc:hive2://localhost:10000>SELECT DISTINCT taux FROM client_7_ext;
```

```
+--------+
|  taux  |
+--------+
| NULL   |
| -1     |
| 150    |
| 151    |
| 152    |
| 153    |
| 154    |
| 155    |
| 156    |
| 157    |
| 158    |
| 159    |
| 160    |
| 161    |
| 162    |
| 163    |
| 164    |
| 165    |
| 166    |
| 167    |
| 168    |
```

```
0: jdbc:hive2://localhost:10000>SELECT COUNT(*)
FROM client_7_ext
WHERE taux is NULL and taux= -1;
```

```
+------+
| _c0  |
+------+
| 319  |
+------+
```

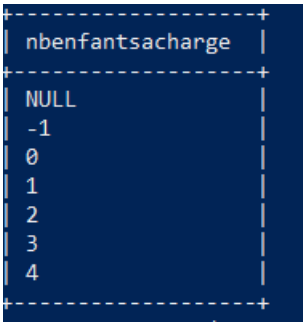
```
0: jdbc:hive2://localhost:10000>SELECT DISTINCT SituationFamiliale  FROM client_7_ext;
```

```
+---------------------+
| situationfamiliale  |
+---------------------+
|                     |
| ?                   |
| Célibataire         |
| Divorcée            |
| En Couple           |
| Marié(e)            |
| N/D                 |
| Seul                |
| Seule               |
+---------------------+
```

```
0: jdbc:hive2://localhost:10000>SELECT
    SUM(CASE WHEN SituationFamiliale = '' THEN 1 ELSE 0 END) AS missing_values,
    SUM(CASE WHEN SituationFamiliale = '?' THEN 1 ELSE 0 END) AS question_marks,
    SUM(CASE WHEN SituationFamiliale = 'N/D' THEN 1 ELSE 0 END) AS nd_values
FROM
    client_7_ext;
```

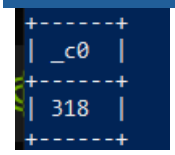
```
+-----------------+-----------------+------------+
| missing_values  | question_marks  | nd_values  |
+-----------------+-----------------+------------+
| 101             | 95              | 105        |
+-----------------+-----------------+------------+
```

```
0: jdbc:hive2://localhost:10000>SELECT DISTINCT NbEnfantsAcharge  FROM client_7_ext;
```



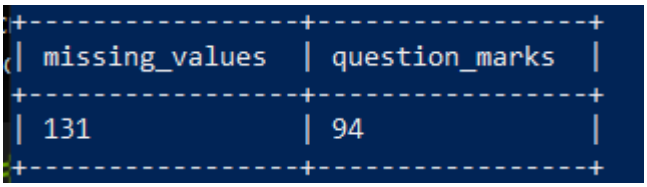
nbenfantsacharge
NULL
-1
0
1
2
3
4

```
0: jdbc:hive2://localhost:10000>SELECT COUNT(*)
FROM client_7_ext
WHERE NbEnfantsAcharge  = -1 OR NbEnfantsAcharge  IS NULL;
```



_c0
318

```
0: jdbc:hive2://localhost:10000>SELECT DISTINCT deuxiemevoiture  FROM client_7_ext;
0: jdbc:hive2://localhost:10000>SELECT
   SUM(CASE WHEN deuxiemevoiture = '' THEN 1 ELSE 0 END) AS missing_values,
   SUM(CASE WHEN deuxiemevoiture = '?' THEN 1 ELSE 0 END) AS question_marks
FROM
   client_7_ext;
```



missing_values	question_marks
131	94

Total des valeurs indéfini: 1791

Comme la table "clients_7_ext" est externe, nous utiliserons des requêtes MongoDB pour effectuer des modifications sur celle-ci.

3.4.3 Nettoyage des données relatives à la colonne « sexe »

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT sexe  FROM client_7_ext;
0: jdbc:hive2://localhost:10000> SELECT
   SUM(CASE WHEN sexe = ''  THEN 1 ELSE 0 END) AS missing_values,
```

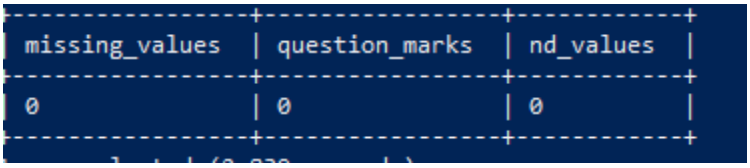
```
    SUM(CASE WHEN sexe = '?' THEN 1 ELSE 0 END) AS question_marks,
    SUM(CASE WHEN sexe = 'N/D' THEN 1 ELSE 0 END) AS nd_values
FROM
    client_7_ext;
```

```
+-----------------+-----------------+------------+
| missing_values  | question_marks  | nd_values  |
+-----------------+-----------------+------------+
| 107             | 93              | 108        |
+-----------------+-----------------+------------+
```

```
> db.Clients_7.aggregate([
...     {
...       $group: {
...         _id: null,
...         missing_values: { $sum: { $cond: [{ $eq: ["$sexe", ""] }, 1, 0] } },
...         question_marks: { $sum: { $cond: [{ $eq: ["$sexe", "?"] }, 1, 0] } },
...         nd_values: { $sum: { $cond: [{ $eq: ["$sexe", "N/D"] }, 1, 0] } }
...       }
...     }
...   ]);
{ "_id" : null, "missing_values" : 107, "question_marks" : 93, "nd_values" : 108 }
>
>
> db.Clients_7.distinct("sexe");
[ "F", "M", "", "Masculin", "Féminin", "?", "Homme", "Femme", "N/D" ]
>
```

```
> db.Clients_7.updateMany(
...     { $or: [ { sexe: "" }, { sexe: "?" }, { sexe: "N/D" } ] },
...     { $set: { sexe: null } }
... );
{ "acknowledged" : true, "matchedCount" : 308, "modifiedCount" : 308 }
>
```

```
0: jdbc:hive2://localhost:10000> SELECT
    SUM(CASE WHEN sexe = ''  THEN 1 ELSE 0 END) AS missing_values,
    SUM(CASE WHEN sexe = '?' THEN 1 ELSE 0 END) AS question_marks,
    SUM(CASE WHEN sexe = 'N/D' THEN 1 ELSE 0 END) AS nd_values
FROM
    client_7_ext;
```



missing_values	question_marks	nd_values
0	0	0

```
> db.Clients_7.distinct("sexe");

[ "F", "M", null, "Masculin", "Féminin", "Homme", "Femme" ]
```

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT sexe  FROM client_7_ext;
```

sexe
NULL
F
Femme
Féminin
Homme
M
Masculin

3.4.4 Transformation des valeurs de la colonne «sexe »

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT sexe  FROM client_7_ext;
```

sexe
NULL
F
Femme
Féminin
Homme
M
Masculin

```
> db.Clients_7.distinct("sexe");
[ "F", "M", null, "Masculin", "Féminin", "Homme", "Femme" ]
>
```

```
> db.Clients_7.updateMany(
...     { sexe: { $in: ["Femme", "Féminin"] } },
...     { $set: { sexe: "F" } }
...   );
{ "acknowledged" : true, "matchedCount" : 615, "modifiedCount" : 615 }
> db.Clients_7.updateMany(
...     { sexe: { $in: ["Homme", "Masculin"] } },
...     { $set: { sexe: "M" } }
...   );
{ "acknowledged" : true, "matchedCount" : 1352, "modifiedCount" : 1352 }
> db.Clients_7.distinct("sexe");
[ "F", "M", null ]
>
```

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT sexe  FROM client_7_ext;
```

```
+--------+
| sexe   |
+--------+
| NULL   |
| F      |
| M      |
+--------+
```

3.4.5 Nettoyage des données relatives à la colonne « situationfamiliale »

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT SituationFamiliale FROM client_7_ext;
```

```
+---------------------+
| situationfamiliale  |
+---------------------+
|                     |
| ?                   |
| Célibataire         |
| Divorcée            |
| En Couple           |
| Marié(e)            |
| N/D                 |
| Seul                |
| Seule               |
+---------------------+
```

```
0: jdbc:hive2://localhost:10000> SELECT
    SUM(CASE WHEN SituationFamiliale = '' THEN 1 ELSE 0 END) AS missing_values,
    SUM(CASE WHEN SituationFamiliale = '?' THEN 1 ELSE 0 END) AS question_marks,
```

```
    SUM(CASE WHEN SituationFamiliale = 'N/D' THEN 1 ELSE 0 END) AS nd_values
FROM
    client_7_ext;
```

```
+-----------------+-----------------+------------+
| missing_values  | question_marks  | nd_values  |
+-----------------+-----------------+------------+
| 101             | 95              | 105        |
+-----------------+-----------------+------------+
```

```
> db.Clients_7.updateMany(
...     { $or: [ { situationFamiliale: "" }, { situationFamiliale: "?" }, { situationFamiliale: "N/D" } ] },
...     { $set: { situationFamiliale: null } }
... );
{ "acknowledged" : true, "matchedCount" : 301, "modifiedCount" : 301 }
> db.Clients_7.distinct("situationFamiliale");
[
        "En Couple",
        "Marié(e)",
        "Célibataire",
        "Seule",
        null,
        "Seul",
        "Divorcée"
]
>
```

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT SituationFamiliale FROM client_7_ext;
```

```
+---------------------+
| situationfamiliale  |
+---------------------+
| NULL                |
| Célibataire         |
| Divorcée            |
| En Couple           |
| Marié(e)            |
| Seul                |
| Seule               |
+---------------------+
```

```
0: jdbc:hive2://localhost:10000> SELECT
    SUM(CASE WHEN SituationFamiliale = '' THEN 1 ELSE 0 END) AS missing_values,
    SUM(CASE WHEN SituationFamiliale = '?' THEN 1 ELSE 0 END) AS question_marks,
    SUM(CASE WHEN SituationFamiliale = 'N/D' THEN 1 ELSE 0 END) AS nd_values
FROM
    client_7_ext;
```

```
+-----------------+-----------------+------------+
| missing_values  | question_marks  | nd_values  |
+-----------------+-----------------+------------+
| 0               | 0               | 0          |
+-----------------+-----------------+------------+
```

3.4.6 Transformation des valeurs de la colonne « situationfamiliale »

- Vous transformez les valeurs "Seul", "Seule" et "Divorcée" en "Célibataire" et les valeurs "Marié(e)" en "En Couple".

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT SituationFamiliale FROM client_7_ext;
```

```
+---------------------+
| situationfamiliale  |
+---------------------+
| NULL                |
| Célibataire         |
| Divorcée            |
| En Couple           |
| Marié(e)            |
| Seul                |
| Seule               |
+---------------------+
```

```
> db.Clients_7.distinct("situationFamiliale");
[
        "En Couple",
        "Marié(e)",
        "Célibataire",
        "Seule",
        null,
        "Seul",
        "Divorcée"
]
> db.Clients_7.updateMany(
...       { situationFamiliale: { $in: ["Seul", "Seule", "Divorcée"] } },
...       { $set: { situationFamiliale: "Célibataire" } }
...   );
{ "acknowledged" : true, "matchedCount" : 5283, "modifiedCount" : 5283 }
> db.Clients_7.updateMany(
...       { situationFamiliale: "Marié(e)" },
...       { $set: { situationFamiliale: "En Couple" } }
...   );
{ "acknowledged" : true, "matchedCount" : 645, "modifiedCount" : 645 }
> db.Clients_7.distinct("situationFamiliale");
[ "En Couple", "Célibataire", null ]
>
```

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT SituationFamiliale FROM client_7_ext;
```

```
+---------------------+
| situationfamiliale  |
+---------------------+
| NULL                |
| Célibataire         |
| En Couple           |
+---------------------+
```

3.4.7 Nettoyage des données relatives à la colonne « nbenfantsacharge »

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT NbEnfantsAcharge  FROM
client_7_ext;
```

```
+-------------------+
| nbenfantsacharge  |
+-------------------+
| NULL              |
| -1                |
| 0                 |
| 1                 |
| 2                 |
| 3                 |
| 4                 |
+-------------------+
```

```
0: jdbc:hive2://localhost:10000> SELECT COUNT(*)

FROM client_7_ext

WHERE NbEnfantsAcharge= -1 OR NbEnfantsAcharge  IS NULL;
```

```
+------+
| _c0  |
+------+
| 318  |
+------+
```

```
> db.Clients_7.distinct("nbEnfantsAcharge");
[ 1, 3, 2, 4, 0, "", "?", -1 ]
> db.Clients_7.find({ nbEnfantsAcharge: { $in: ["", -1, "?"] } }).count();
318
> db.Clients_7.updateMany(
...    { nbEnfantsAcharge: { $in: ["", -1, "?"] } },
...    { $set: { nbEnfantsAcharge: null } }
... );
{ "acknowledged" : true, "matchedCount" : 318, "modifiedCount" : 318 }
> db.Clients_7.distinct("nbEnfantsAcharge");
[ 1, 3, 2, 4, 0, null ]
>
```

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT NbEnfantsAcharge  FROM
client_7_ext;
```

```
+-------------------+
| nbenfantsacharge  |
+-------------------+
| NULL              |
| 0                 |
| 1                 |
| 2                 |
| 3                 |
| 4                 |
+-------------------+
```

```
0: jdbc:hive2://localhost:10000> SELECT
   SUM(CASE WHEN NbEnfantsAcharge = '' THEN 1 ELSE 0 END) AS missing_values,
   SUM(CASE WHEN NbEnfantsAcharge = '?' THEN 1 ELSE 0 END) AS question_marks,
   SUM(CASE WHEN NbEnfantsAcharge= -1 THEN 1 ELSE 0 END) AS minus1_values
FROM
   client_7_ext;
```

```
+-----------------+-----------------+----------------+
| missing_values  | question_marks  | minus1_values  |
+-----------------+-----------------+----------------+
| 0               | 0               | 0              |
+-----------------+-----------------+----------------+
```

3.4.8 Nettoyage des données relatives à la colonne « deuxiemevoiture »

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT deuxiemevoiture  FROM client_7_ext;
```

```
+-------------------+
| deuxiemevoiture   |
+-------------------+
|                   |
| ?                 |
| false             |
| true              |
+-------------------+
```

```
0: jdbc:hive2://localhost:10000> SELECT
   SUM(CASE WHEN deuxiemevoiture = '' THEN 1 ELSE 0 END) AS missing_values,
   SUM(CASE WHEN deuxiemevoiture = '?' THEN 1 ELSE 0 END) AS question_marks
FROM
        client_7_ext;
```

```
+-----------------+-----------------+
| missing_values  | question_marks  |
+-----------------+-----------------+
| 131             | 94              |
+-----------------+-----------------+
```

```
> db.Clients_7.distinct("2eme voiture");
[ "false", "true", "?", "" ]
> db.Clients_7.find({ "2eme voiture": { $in: ["", "?"] } }).count();
225
> db.Clients_7.updateMany(
...     { "2eme voiture": { $in: ["","?"] } },
...     { $set: { "2eme voiture": null } }
...  );
{ "acknowledged" : true, "matchedCount" : 225, "modifiedCount" : 225 }
> db.Clients_7.distinct("2eme voiture");
[ "false", "true", null ]
>
```

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT deuxiemevoiture  FROM client_7_ext;
```

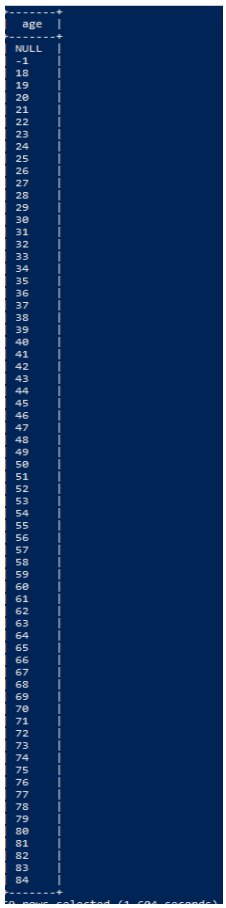
```
+------------------+
| deuxiemevoiture  |
+------------------+
| NULL             |
| false            |
| true             |
+------------------+
```

```
0: jdbc:hive2://localhost:10000> SELECT
   SUM(CASE WHEN deuxiemevoiture = '' THEN 1 ELSE 0 END) AS missing_values,
   SUM(CASE WHEN deuxiemevoiture = '?' THEN 1 ELSE 0 END) AS question_marks
FROM
   client_7_ext;
```

missing_values	question_marks
0	0

3.4.9 Nettoyage des données relatives à la colonne « age »

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT age  FROM client_7_ext;
```




```
0: jdbc:hive2://localhost:10000> SELECT COUNT(*)
FROM client_7_ext
WHERE age = -1 OR age IS NULL;
```

_c0
320

```
> db.Clients_7.distinct("age");
[
        57,
        24,
        73,
        52,
]       59,
>       63,
        79,
        44,
        81,
        ...
        null,
        70,
        84,
        82,
        76
]
```

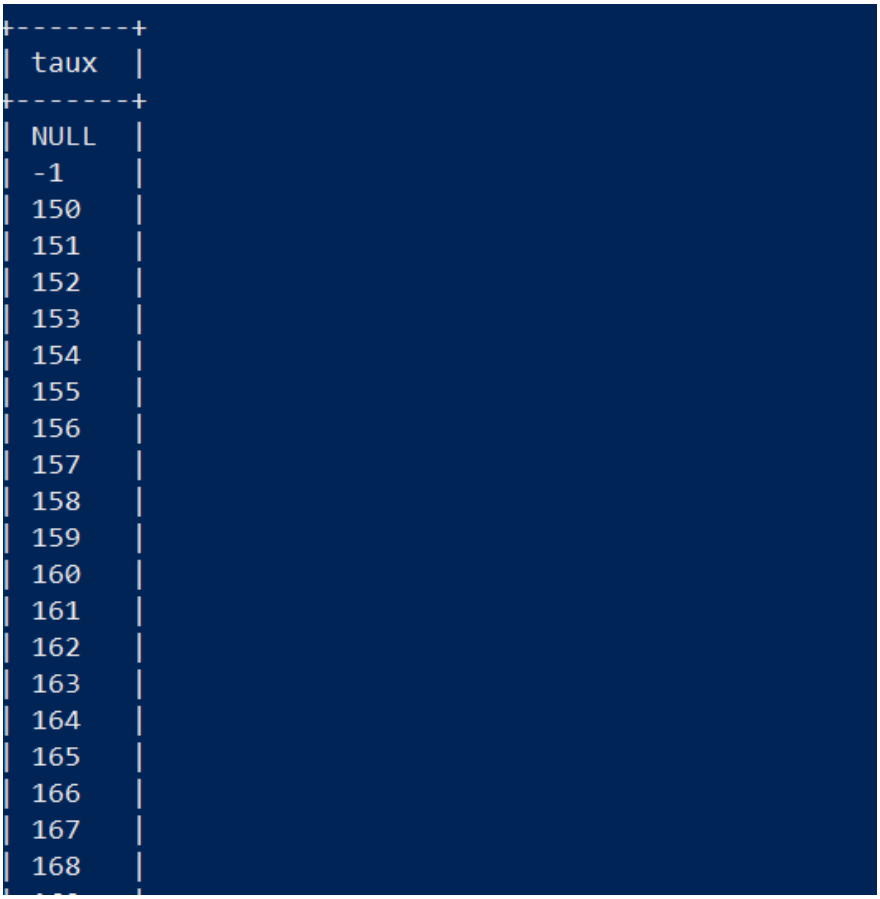
```
> db.Clients_7.find({ "age": { $in: ["", "?",-1] } }).count();
320
>
> db.Clients_7.updateMany(
...     { "age": { $in: ["", "?", -1] } },
...     { $set: { "age": null } }
... );
{ "acknowledged" : true, "matchedCount" : 320, "modifiedCount" : 320 }
>
```

```
0: jdbc:hive2://localhost:10000> SELECT
    SUM(CASE WHEN Age = '' THEN 1 ELSE 0 END) AS missing_values,
    SUM(CASE WHEN Age = '?' THEN 1 ELSE 0 END) AS question_marks,
    SUM(CASE WHEN Age = -1 THEN 1 ELSE 0 END) AS minus1_values
FROM
    client_7_ext;
```

missing_values	question_marks	minus1_values
0	0	0

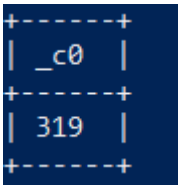
3.4.10 Nettoyage des données relatives à la colonne « taux »

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT taux  FROM client_7_ext;
```



```
+-------+
| taux  |
+-------+
| NULL  |
| -1    |
| 150   |
| 151   |
| 152   |
| 153   |
| 154   |
| 155   |
| 156   |
| 157   |
| 158   |
| 159   |
| 160   |
| 161   |
| 162   |
| 163   |
| 164   |
| 165   |
| 166   |
| 167   |
| 168   |
```

```
0: jdbc:hive2://localhost:10000> SELECT COUNT(*)

FROM client_7_ext

WHERE taux= -1 OR taux  IS NULL;
```



```
+------+
| _c0  |
+------+
| 319  |
+------+
```

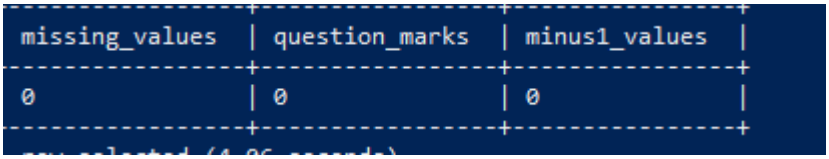
```
> db.Clients_7.distinct("taux");
```

```
> db.Clients_7.find({ taux: { $in: ["", -1, "?"] } }).count();
319
> db.Clients_7.updateMany(
...     { taux: { $in: ["", -1, "?"] } },
...     { $set: { taux: null } }
... );
{ "acknowledged" : true, "matchedCount" : 319, "modifiedCount" : 319 }
>
```

```
SELECT DISTINCT taux  FROM client_7_ext;
```

```
+--------+
| taux   |
+--------+
| NULL   |
| 150    |
| 151    |
| 152    |
| 153    |
| 154    |
| 155    |
| 156    |
| 157    |
| 158    |
| 159    |
| 160    |
| 161    |
| 162    |
| 163    |
| 164    |
| 165    |
| 166    |
| 167    |
| 168    |
| 169    |
| 170    |
| 171    |
| 172    |
| 173    |
| 174    |
```

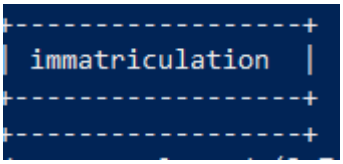
```
0: jdbc:hive2://localhost:10000> SELECT
    SUM(CASE WHEN taux = '' THEN 1 ELSE 0 END) AS missing_values,
    SUM(CASE WHEN taux = '?' THEN 1 ELSE 0 END) AS question_marks,
    SUM(CASE WHEN taux = -1 THEN 1 ELSE 0 END) AS minus1_values
FROM client_7_ext;
```



missing_values	question_marks	minus1_values
0	0	0

3.4.11 Vérification de la conformité du format des immatriculations

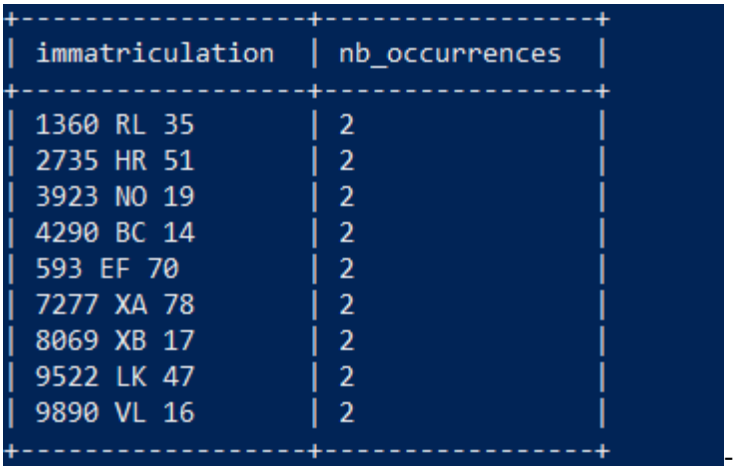
```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT immatriculation
FROM client_7_ext
WHERE immatriculation NOT REGEXP '^[0-9]{1,4} [A-Z]{2} [0-9]{2}$';
```



immatriculation

3.4.12 Gestion des valeurs récurrentes de la colonne « immatriculation »

```
0: jdbc:hive2://localhost:10000> SELECT immatriculation, COUNT(*) AS nb_occurrences
FROM client_7_ext
GROUP BY immatriculation
HAVING COUNT(*) > 1;
```



immatriculation	nb_occurrences
1360 RL 35	2
2735 HR 51	2
3923 NO 19	2
4290 BC 14	2
593 EF 70	2
7277 XA 78	2
8069 XB 17	2
9522 LK 47	2
9890 VL 16	2

```
0: jdbc:hive2://localhost:10000> SELECT * FROM client_7_ext WHERE immatriculation IN
   (SELECT immatriculation FROM client_7_ext
    GROUP BY immatriculation HAVING COUNT(*) > 1);
```

client_7_ext.age	client_7_ext.sexe	client_7_ext.taux	client_7_ext.situationfamiliale	client_7_ext.nbenfantsacharge	client_7_ext.deuxiemevoiture	client_7_ext.immatriculation
24	F	199	En Couple	0	false	1360 RL 35
18	F	234	Célibataire	0	false	1360 RL 35
66	M	543	Célibataire	0	false	2735 HR 51
36	M	911	En Couple	1	true	2735 HR 51
36	M	737	En Couple	1	false	3923 NO 19
39	M	1150	En Couple	1	false	3923 NO 19
48	F	1236	En Couple	3	true	4290 BC 14
53	F	907	Célibataire	0	false	4290 BC 14
45	M	447	Célibataire	0	false	593 EF 70
53	M	999	Célibataire	0	false	593 EF 70
50	M	463	En Couple	0	false	7277 XA 78
68	F	549	Célibataire	0	false	7277 XA 78
70	M	503	En Couple	4	true	8069 XB 17
50	M	575	Célibataire	0	false	8069 XB 17
27	M	424	En Couple	1	true	9522 LK 47
78	M	241	Célibataire	0	false	9522 LK 47
30	M	504	Célibataire	0	false	9890 VL 16
49	M	492	Célibataire	0	false	9890 VL 16

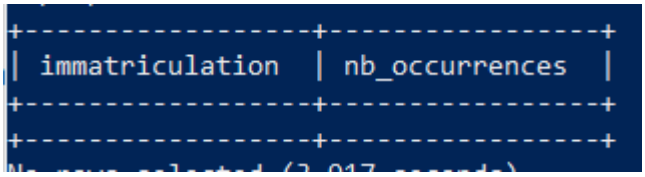
Nous constatons que les 9 paires d'immatriculations des voitures des clients sont principalement différentes entre elles, rendant ainsi difficile la mise en place d'une solution générale pour traiter cette ambiguïté d'unicité. Une solution possible et simple serait de supprimer une occurrence et de conserver une seule (étant donné que chaque immatriculation se répète deux fois).

```
0: jdbc:hive2://localhost:10000> SELECT COUNT(*) FROM Client_7_ext
```

_c0
100000

```
> db.Clients_7.aggregate([
...       { $group: { _id: "$immatriculation", count: { $sum: 1 } } },
...       { $match: { count: { $gt: 1 } } }
...   ]);
{ "_id" : "9522 LK 47", "count" : 2 }
{ "_id" : "3923 NO 19", "count" : 2 }
{ "_id" : "1360 RL 35", "count" : 2 }
{ "_id" : "593 EF 70", "count" : 2 }
{ "_id" : "2735 HR 51", "count" : 2 }
{ "_id" : "8069 XB 17", "count" : 2 }
{ "_id" : "4290 BC 14", "count" : 2 }
{ "_id" : "7277 XA 78", "count" : 2 }
{ "_id" : "9890 VL 16", "count" : 2 }
>
> var immatriculationsRepetees = db.Clients_7.aggregate([
...       { $group: { _id: "$immatriculation", count: { $sum: 1 } } },
...       { $match: { count: { $gt: 1 } } }
... ]).toArray();
> immatriculationsRepetees.forEach(function(doc) {
...       var immatriculation = doc._id;
...       // Supprimer une seule occurrence de l'immatriculation répétée
...       db.Clients_7.deleteOne({ immatriculation: immatriculation });
... });
> db.Clients_7.count();
99991
```

```
0: jdbc:hive2://localhost:10000> SELECT immatriculation, COUNT(*) AS nb_occurrences
FROM client_7_ext
GROUP BY immatriculation
HAVING COUNT(*) > 1;
```



3.4.13 Gestion des valeurs NULL de la table « client_7_ext »

```
0: jdbc:hive2://localhost:10000> SELECT *
FROM client_7_ext
WHERE immatriculation is NULL OR age is NULL  OR sexe is NULL OR taux is NULL OR situationFamiliale is NULL  OR nbEnfantsAcharge IS NULL OR deuxiemevoiture IS NULL;
0: jdbc:hive2://localhost:10000> SELECT t.*,
   (CASE WHEN immatriculation IS NULL THEN 1 ELSE 0 END) +
   (CASE WHEN age IS NULL THEN 1 ELSE 0 END) +
   (CASE WHEN sexe IS NULL THEN 1 ELSE 0 END) +
   (CASE WHEN taux IS NULL THEN 1 ELSE 0 END) +
   (CASE WHEN situationFamiliale IS NULL THEN 1 ELSE 0 END) +
   (CASE WHEN nbEnfantsAcharge IS NULL THEN 1 ELSE 0 END) +
   (CASE WHEN deuxiemevoiture IS NULL THEN 1 ELSE 0 END) AS nb_colonnes_null
FROM client_7_ext t
WHERE immatriculation IS NULL OR age IS NULL OR sexe IS NULL OR taux IS NULL OR situationFamiliale IS NULL OR nbEnfantsAcharge IS NULL OR deuxiemevoiture IS NULL;
```

35	F	589	En Couple	2	NULL	7698 BX 28	1
37	M	1389	Célibataire	NULL	false	6723 YX 21	1
26	M	408	Célibataire	0	NULL	7592 CQ 34	1
NULL	M	414	En Couple	0	false	9625 CC 23	1
28	M	NULL	En Couple	2	false	4660 OE 55	1
49	M	580	En Couple	2	NULL	2666 XD 12	1
24	M	1377	NULL	2	false	8931 TV 67	1
29	M	1136	NULL	3	false	7523 RP 20	1
46	M	NULL	En Couple	1	false	8728 ZG 85	1
44	M	191	NULL	0	false	9834 FF 74	1
67	F	175	NULL	0	false	9422 LR 48	1
30	M	543	Célibataire	NULL	false	4871 OR 32	1
NULL	M	208	Célibataire	0	false	4254 WH 70	1
61	M	NULL	Célibataire	0	false	3555 YR 75	1
20	M	550	Célibataire	3	NULL	8439 RH 84	1
74	M	NULL	Célibataire	2	false	9462 EP 38	1
51	F	583	Célibataire	NULL	false	4736 QQ 94	1
29	M	NULL	En Couple	3	false	8646 JU 95	1
27	F	NULL	En Couple	3	false	837 BY 12	1
56	F	NULL	En Couple	4	false	8506 YS 27	1
57	M	554	En Couple	3	NULL	2288 BR 54	1
68	M	238	En Couple	NULL	false	2762 IR 96	1
72	M	563	Célibataire	NULL	false	1077 XU 37	1
25	F	237	En Couple	2	NULL	3699 HF 30	1
45	F	NULL	En Couple	4	false	1873 TJ 24	1
59	F	512	Célibataire	NULL	false	3451 CE 73	1
NULL	F	996	Célibataire	0	false	3509 VG 66	1
27	M	NULL	En Couple	2	false	7644 SE 95	1
NULL	M	414	En Couple	4	false	3464 SB 42	1
31	M	576	En Couple	NULL	false	5212 OK 49	1
42	M	NULL	Célibataire	0	false	893 YN 50	1
34	M	NULL	Célibataire	0	false	5099 BC 54	1
53	M	1054	En Couple	NULL	false	8546 DA 59	1
29	F	422	En Couple	4	NULL	4678 RE 12	1
48	NULL	592	En Couple	1	false	8662 GB 84	1
25	NULL	459	En Couple	4	false	9625 IO 19	1
76	M	414	En Couple	NULL	true	79 YX 19	1
75	M	493	En Couple	1	NULL	6654 JV 36	1
NULL	M	544	Célibataire	0	false	2685 OA 33	1
76	M	963	Célibataire	NULL	false	3352 DW 30	1
48	M	734	NULL	0	false	4652 AQ 31	1
37	F	488	Célibataire	0	NULL	566 LU 47	1
60	M	522	En Couple	2	NULL	8351 TE 27	1
27	M	167	En Couple	NULL	false	280 FB 84	1
18	M	NULL	Célibataire	0	false	7254 EF 55	1
41	M	583	Célibataire	NULL	false	9638 UT 57	1
20	F	NULL	En Couple	2	true	7369 FF 88	1
38	M	156	NULL	4	false	1807 VH 18	1
26	M	468	En Couple	1	NULL	6329 MH 52	1
NULL	M	492	Célibataire	3	false	5375 LO 21	1
47	F	176	NULL	0	true	9983 RJ 78	1
NULL	M	153	Célibataire	1	false	4891 RD 57	1
54	M	866	Célibataire	0	NULL	2301 MB 19	1

Nous observons que la plupart des colonnes présentent une seule valeur NULL pour chaque ligne, cependant, cela nécessite une vérification.

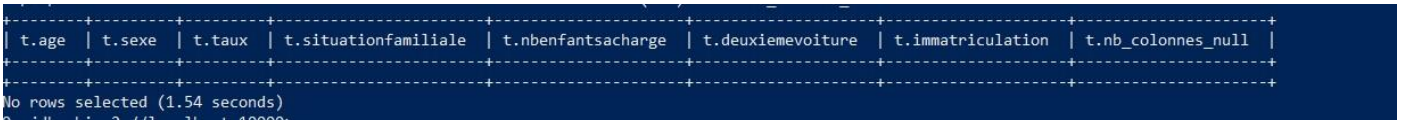
```
db.Clients_7.aggregate([
..  {
..    $addFields: {
..      nb_colonnes_null: {
..        $sum: [
..          { $cond: [{ $eq: ["$immatriculation", null] }, 1, 0] },
..          { $cond: [{ $eq: ["$age", null] }, 1, 0] },
..          { $cond: [{ $eq: ["$sexe", null] }, 1, 0] },
..          { $cond: [{ $eq: ["$taux", null] }, 1, 0] },
..          { $cond: [{ $eq: ["$situationFamiliale", null] }, 1, 0] },
..          { $cond: [{ $eq: ["$nbEnfantsAcharge", null] }, 1, 0] },
..          { $cond: [{ $eq: ["$2eme voiture", null] }, 1, 0] }
..        ]
..      }
..    }
..  },
..  {
..    $match: {
..      $and: [
..        { $or: [{ immatriculation: null }, { age: null }, { sexe: null }, { taux: null }, { situationFamiliale: null }, { nbEnfantsAcharge: null }, { "2eme voiture": null }] },
..        { nb_colonnes_null: { $gte: 2 } }
..      ]
..    }
..  }
.. ])
{ "_id" : ObjectId("6614b231afa50bb5d2927561"), "age" : 69, "sexe" : null, "taux" : 590, "situationFamiliale" : "En Couple", "nbEnfantsAcharge" : null, "2eme voiture" : "false", "immatriculation" : "7960 KS 89", "nb_colonnes_null" : 2 }
{ "_id" : ObjectId("6614b231afa50bb5d292814f"), "age" : 46, "sexe" : "F", "taux" : null, "situationFamiliale" : "En Couple", "nbEnfantsAcharge" : 1, "2eme voiture" : null, "immatriculation" : "1934 OO 43", "nb_colonnes_null" : 2 }
{ "_id" : ObjectId("6614b232afa50bb5d29284c0"), "age" : 42, "sexe" : null, "taux" : 969, "situationFamiliale" : "En Couple", "nbEnfantsAcharge" : null, "2eme voiture" : "false", "immatriculation" : "7973 MB 86", "nb_colonnes_null" : 2 }
{ "_id" : ObjectId("6614b232afa50bb5d2928b3e"), "age" : 82, "sexe" : null, "taux" : 1004, "situationFamiliale" : "En Couple", "nbEnfantsAcharge" : null, "2eme voiture" : "false", "immatriculation" : "5390 EU 67", "nb_colonnes_null" : 2 }
{ "_id" : ObjectId("6614b232afa50bb5d292b633"), "age" : null, "sexe" : "M", "taux" : 586, "situationFamiliale" : null, "nbEnfantsAcharge" : 2, "2eme voiture" : "true", "immatriculation" : "2797 GY 47", "nb_colonnes_null" : 2 }
{ "_id" : ObjectId("6614b232afa50bb5d292e538"), "age" : null, "sexe" : "M", "taux" : null, "situationFamiliale" : "Célibataire", "nbEnfantsAcharge" : 0, "2eme voiture" : "false", "immatriculation" : "8332 VJ 91", "nb_colonnes_null" : 2 }
{ "_id" : ObjectId("6614b232afa50bb5d2930215"), "age" : 52, "sexe" : null, "taux" : 798, "situationFamiliale" : "En Couple", "nbEnfantsAcharge" : 2, "2eme voiture" : null, "immatriculation" : "7291 PF 31", "nb_colonnes_null" : 2 }
{ "_id" : ObjectId("6614b232afa50bb5d29362f5"), "age" : 67, "sexe" : null, "taux" : 448, "situationFamiliale" : null, "nbEnfantsAcharge" : 0, "2eme voiture" : "false", "immatriculation" : "9676 MV 67", "nb_colonnes_null" : 2 }
{ "_id" : ObjectId("6614b232afa50bb5d2937579"), "age" : null, "sexe" : "M", "taux" : 573, "situationFamiliale" : "En Couple", "nbEnfantsAcharge" : 3, "2eme voiture" : null, "immatriculation" : "9398 DJ 37", "nb_colonnes_null" : 2 }
{ "_id" : ObjectId("6614b232afa50bb5d2938aa7"), "age" : null, "sexe" : "F", "taux" : 409, "situationFamiliale" : null, "nbEnfantsAcharge" : 3, "2eme voiture" : "true", "immatriculation" : "7809 EF 55", "nb_colonnes_null" : 2 }
{ "_id" : ObjectId("6614b233afa50bb5d293cb39"), "age" : null, "sexe" : "M", "taux" : 474, "situationFamiliale" : "En Couple", "nbEnfantsAcharge" : 0, "2eme voiture" : null, "immatriculation" : "3852 GZ 18", "nb_colonnes_null" : 2 }


db.Clients_7.aggregate([
```

Nous notons qu'il y a 11 lignes où le nombre de colonnes nulles est de 2, et également que 10 lignes ont au moins des valeurs NULL dans les colonnes sexe et âge, des données critiques et importantes pour notre analyse. Par conséquent, nous concluons qu'il est nécessaire de supprimer ces 11 lignes qui n'apportent pas une valeur significative comparativement aux lignes qui ne présentent qu'une seule colonne à valeur NULL.

```
> var documentsASupprimer = db.Clients_7.aggregate([
...   {
...     $addFields: {
...       nb_colonnes_null: {
...         $sum: [
...           { $cond: [{ $eq: ["$immatriculation", null] }, 1, 0] },
...           { $cond: [{ $eq: ["$age", null] }, 1, 0] },
...           { $cond: [{ $eq: ["$sexe", null] }, 1, 0] },
...           { $cond: [{ $eq: ["$taux", null] }, 1, 0] },
...           { $cond: [{ $eq: ["$situationFamiliale", null] }, 1, 0] },
...           { $cond: [{ $eq: ["$nbEnfantsAcharge", null] }, 1, 0] },
...           { $cond: [{ $eq: ["$2eme voiture", null] }, 1, 0] }
...         ]
...       }
...     }
...   },
...   {
...     $match: {
...       $and: [
...         { $or: [{ immatriculation: null }, { age: null }, { sexe: null }, { taux: null }, { situationFamiliale: null }, { nbEnfantsAcharge: null }, { "2eme voiture": null }] },
...         { nb_colonnes_null: { $gte: 2 } }
...       ]
...     }
...   }
... ]).toArray();
> documentsASupprimer.forEach(function(document) {
...   db.Clients_7.deleteOne({ _id: document._id });
... });
```

```
0: jdbc:hive2://localhost:10000> SELECT *
FROM (
   SELECT t.*,
      (CASE WHEN immatriculation IS NULL THEN 1 ELSE 0 END) +
      (CASE WHEN age IS NULL THEN 1 ELSE 0 END) +
      (CASE WHEN sexe IS NULL THEN 1 ELSE 0 END) +
      (CASE WHEN taux IS NULL THEN 1 ELSE 0 END) +
      (CASE WHEN situationFamiliale IS NULL THEN 1 ELSE 0 END) +
      (CASE WHEN nbEnfantsAcharge IS NULL THEN 1 ELSE 0 END) +
      (CASE WHEN deuxiemevoiture IS NULL THEN 1 ELSE 0 END) AS nb_colonnes_null
   FROM client_7_ext t
) t
```

```
WHERE (immatriculation IS NULL OR age IS NULL OR sexe IS NULL OR taux IS NULL OR
situationFamiliale IS NULL OR nbEnfantsAcharge IS NULL OR deuxiemevoiture IS NULL)
AND nb_colonnes_null >= 2;
```



3.4.14 Dimension final de la table « client_7_ext »

```
0: jdbc:hive2://localhost:10000> SELECT COUNT(*) from client_7_ext;

+--------+
| _c0    |
+--------+
| 99980  |
+--------+
```

3.5 Transformation et nettoyage des données de la table «client_12_ext » :

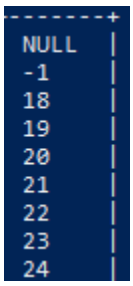
3.5.1 Description de la table « client_12_ext » :

```
0: jdbc:hive2://localhost:10000> DESC client_12_ext ;
```

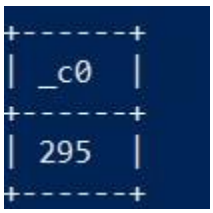
col_name	data_type	comment
age	int	from deserializer
sexe	string	from deserializer
taux	int	from deserializer
situationfamiliale	string	from deserializer
nbenfantsacharge	int	from deserializer
deuxiemevoiture	string	from deserializer
immatriculation	string	from deserializer

3.5.2 Comptage du nombre de valeurs indéfinis dans la table «client_12_ext »

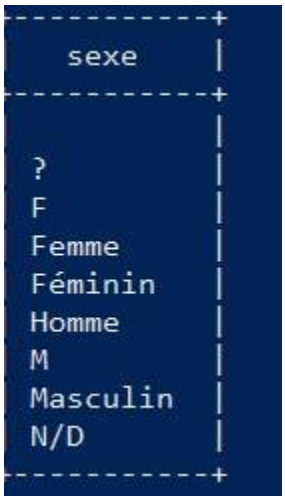
```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT age FROM client_12_ext;
```




```
0: jdbc:hive2://localhost:10000> SELECT COUNT(*)
FROM client_12_ext
WHERE age = -1 OR age IS NULL;
```




```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT sexe FROM client_12_ext;
```



```
0: jdbc:hive2://localhost:10000> SELECT
  SUM(CASE WHEN sexe = '' THEN 1 ELSE 0 END) AS missing_values,
  SUM(CASE WHEN sexe = '?' THEN 1 ELSE 0 END) AS question_marks,
  SUM(CASE WHEN sexe = 'N/D' THEN 1 ELSE 0 END) AS nd_values
FROM
  client_12_ext;
```

```
+-----------------+-----------------+------------+
| missing_values  | question_marks  | nd_values  |
+-----------------+-----------------+------------+
| 90              | 98              | 95         |
+-----------------+-----------------+------------+
1 row selected (2.811 seconds)
```

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT taux FROM client_12_ext;
```

```
+-------+
| taux  |
+-------+
| NULL  |
| -1    |
| 150   |
| 151   |
| 152   |
| 153   |
| 154   |
| 155   |
| 156   |
| 157   |
```

```
0: jdbc:hive2://localhost:10000> SELECT COUNT(*)
FROM client_12_ext
WHERE taux = -1 OR taux IS NULL;
```

```
+------+
| _c0  |
+------+
| 339  |
+------+
```

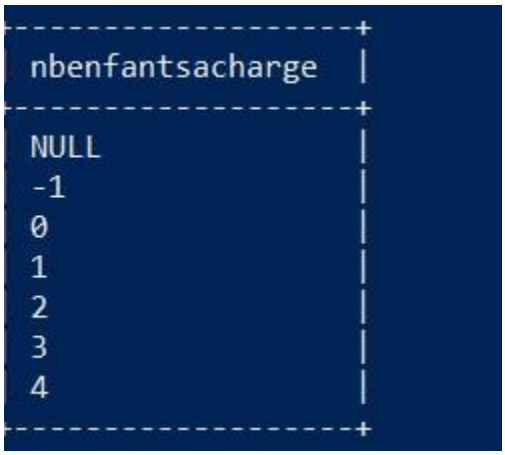
```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT SituationFamiliale  FROM
client_12_ext;
```

```
+----------------------+
| situationfamiliale   |
+----------------------+
|                      |
| ?                    |
| Célibataire          |
| Divorcée             |
| En Couple            |
| Marié(e)             |
| N/D                  |
| Seul                 |
| Seule                |
+----------------------+
```

```
0: jdbc:hive2://localhost:10000> SELECT
   SUM(CASE WHEN SituationFamiliale = '' THEN 1 ELSE 0 END) AS
missing_values,
   SUM(CASE WHEN SituationFamiliale = '?' THEN 1 ELSE 0 END) AS
question_marks,
   SUM(CASE WHEN SituationFamiliale = 'N/D' THEN 1 ELSE 0 END) AS
nd_values
FROM
   client_12_ext;
```

```
-----------------+-----------------+------------+
 missing_values  | question_marks  | nd_values  |
-----------------+-----------------+------------+
 104             | 100             | 92         |
-----------------+-----------------+------------+
```

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT NbEnfantsAcharge  FROM
client_12_ext;
```

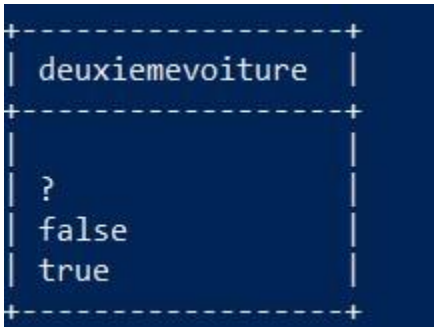


nbenfantsacharge
NULL
-1
0
1
2
3
4

```
0: jdbc:hive2://localhost:10000> SELECT COUNT(*)
FROM client_12_ext
WHERE NbEnfantsAcharge  = -1 OR NbEnfantsAcharge  IS NULL;
```

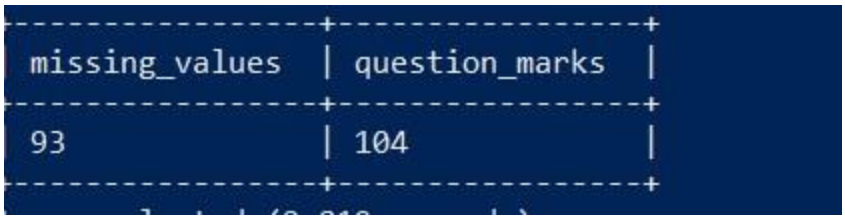
_c0
286

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT deuxiemevoiture  FROM
client_12_ext;
```



deuxiemevoiture
?
false
true

```
0: jdbc:hive2://localhost:10000> SELECT
   SUM(CASE WHEN deuxiemevoiture = '' THEN 1 ELSE 0 END) AS
missing_values,
   SUM(CASE WHEN deuxiemevoiture = '?' THEN 1 ELSE 0 END) AS
question_marks
FROM
   client_12_ext;
```



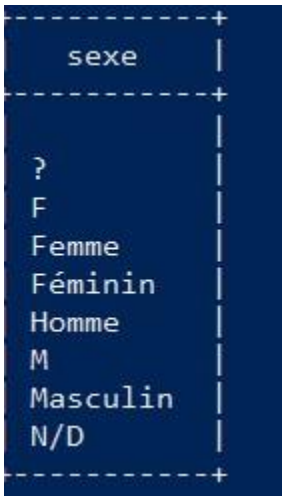
missing_values	question_marks
93	104

Total des valeurs indéfini: 1696

Comme la table "clients_12_ext" est externe, nous utiliserons des requêtes MongoDB pour effectuer des modifications sur celle-ci.

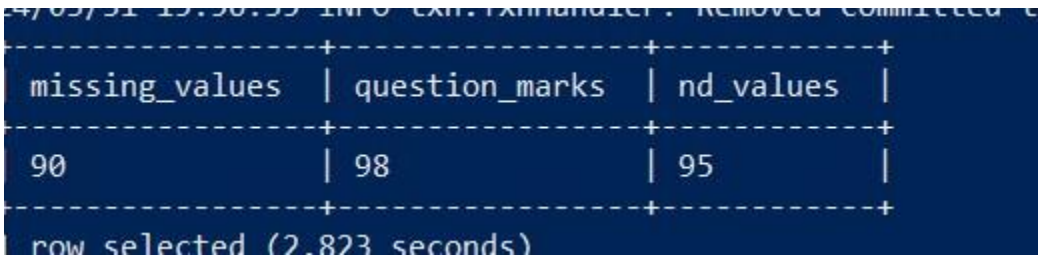
3.5.3 Nettoyage des données relative à la colonne « sexe »

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT sexe  FROM client_12_ext;
```



sexe
?
F
Femme
Féminin
Homme
M
Masculin
N/D

```
0: jdbc:hive2://localhost:10000> SELECT
   SUM(CASE WHEN sexe = '' THEN 1 ELSE 0 END) AS missing_values,
   SUM(CASE WHEN sexe = '?' THEN 1 ELSE 0 END) AS question_marks,
   SUM(CASE WHEN sexe = 'N/D' THEN 1 ELSE 0 END) AS nd_values
FROM
   client_12_ext;
0: jdbc:hive2://localhost:10000>
```



missing_values	question_marks	nd_values
90	98	95

row selected (2.823 seconds)

```
> db.Clients_12.distinct("sexe");
[ "M", "F", "Masculin", "Féminin", "Femme", "Homme", "N/D", "", "?" ]
>
```

```
> use Clients
switched to db Clients
> db.Clients_12.aggregate([
...     {
...       $group: {
...         _id: null,
...         missing_values: { $sum: { $cond: [{ $eq: ["$sexe", ""] }, 1, 0] } },
: [{        question_marks: { $sum: { $cond: [{ $eq: ["$sexe", "?"] }, 1, 0] } },
...         nd_values: { $sum: { $cond: [{ $eq: ["$sexe", "N/D"] }, 1, 0] } }
...       }
...     }
...   ]);
{ "_id" : null, "missing_values" : 90, "question_marks" : 98, "nd_values" : 95 }
```

```
> db.Clients_12.updateMany(
...     { $or: [ { sexe: "" }, { sexe: "?" }, { sexe: "N/D" } ] },
...     { $set: { sexe: null } }
...   );
{ "acknowledged" : true, "matchedCount" : 283, "modifiedCount" : 283 }
> db.Clients_12.distinct("sexe");
[ "M", "F", "Masculin", "Féminin", "Femme", "Homme", null ]
>
```

```
0: jdbc:hive2://localhost:10000> SELECT
   SUM(CASE WHEN sexe = '' THEN 1 ELSE 0 END) AS missing_values,
   SUM(CASE WHEN sexe = '?' THEN 1 ELSE 0 END) AS question_marks,
   SUM(CASE WHEN sexe = 'N/D' THEN 1 ELSE 0 END) AS nd_values
FROM
   client_12_ext;
```

missing_values	question_marks	nd_values
0	0	0

```
> db.Clients_12.distinct("sexe");
[ "M", "F", "Masculin", "Féminin", "Femme", "Homme" ]
0: jdbc:hive2://localhost:10000> SELECT DISTINCT sexe  FROM client_12_ext;
```

sexe
NULL
F
Femme
Féminin
Homme
M
Masculin

3.5.4 Transformation des données de la colonne « sexe »

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT sexe  FROM client_12_ext;
```

sexe
NULL
F
Femme
Féminin
Homme
M
Masculin

```
> db.Clients_12.updateMany(
...     { $or: [ { sexe: "" }, { sexe: "?" }, { sexe: "N/D" } ] },
...     { $set: { sexe: null } }
...  );
{ "acknowledged" : true, "matchedCount" : 283, "modifiedCount" : 283 }
> db.Clients_12.distinct("sexe");
[ "M", "F", "Masculin", "Féminin", "Femme", "Homme", null ]
>
>
>
>
> db.Clients_12.distinct("sexe");
[ "M", "F", "Masculin", "Féminin", "Femme", "Homme", null ]
> db.Clients_12.updateMany(
...    { sexe: { $in: ["Femme", "Féminin"] } },
...    { $set: { sexe: "F" } }
...  );
{ "acknowledged" : true, "matchedCount" : 646, "modifiedCount" : 646 }
> db.Clients_12.updateMany(
...    { sexe: { $in: ["Homme", "Masculin"] } },
...    { $set: { sexe: "M" } }
...  );
{ "acknowledged" : true, "matchedCount" : 1380, "modifiedCount" : 1380 }
>
> db.Clients_12.distinct("sexe");
[ "M", "F", null ]
>
```

```
+--------+
| sexe   |
+--------+
| NULL   |
| F      |
| M      |
+--------+
```

3.5.6 Nettoyage des données relative à la colonne « situationfamiliale »

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT SituationFamiliale FROM
client_12_ext;
```

```
+----------------------+
| situationfamiliale   |
+----------------------+
|                      |
| ?                    |
| Célibataire          |
| Divorcée             |
| En Couple            |
| Marié(e)             |
| N/D                  |
| Seul                 |
| Seule                |
+----------------------+
```

```
> db.Clients_12.distinct("situationFamiliale");
[
      "En Couple",
      "Célibataire",
      "Seule",
      "",
      "Marié(e)",
      "Seul",
      "?",
      "N/D",
      "Divorcée"
]
0: jdbc:hive2://localhost:10000> SELECT
   SUM(CASE WHEN  SituationFamiliale = '' THEN 1 ELSE 0 END) AS
missing_values,
```

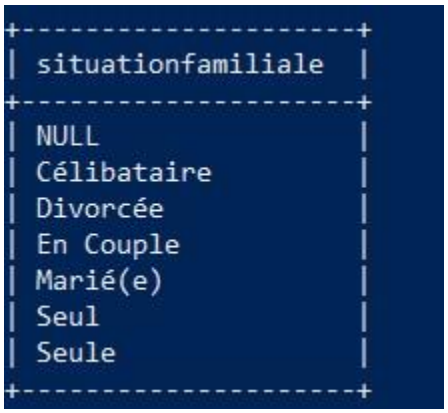
```
   SUM(CASE WHEN SituationFamiliale = '?' THEN 1 ELSE 0 END) AS
question_marks,
   SUM(CASE WHEN SituationFamiliale = 'N/D' THEN 1 ELSE 0 END) AS
nd_values
FROM
   client_12_ext;
```

```
+-----------------+-----------------+------------+
| missing_values  | question_marks  | nd_values  |
+-----------------+-----------------+------------+
| 104             | 100             | 92         |
+-----------------+-----------------+------------+
row selected (2.993 seconds)
```

```
> db.Clients_12.updateMany(
...     { $or: [ { situationFamiliale: "" }, { situationFamiliale: "?" }, {
situationFamiliale: "N/D" } ] },
...     { $set: { situationFamiliale: null } }
...   );
{ "acknowledged" : true, "matchedCount" : 296, "modifiedCount" : 296 }
```

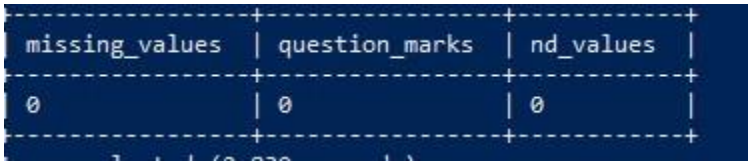
```
> db.Clients_12.distinct("situationFamiliale");
[
        "En Couple",
        "Célibataire",
        "Seule",
        "",
        "Marié(e)",
        "Seul",
        "?",
        "N/D",
        "Divorcée"
]
>
```

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT SituationFamiliale FROM
client_12_ext;
```



situationfamiliale
NULL
Célibataire
Divorcée
En Couple
Marié(e)
Seul
Seule

```
0: jdbc:hive2://localhost:10000> SELECT
   SUM(CASE WHEN SituationFamiliale = '' THEN 1 ELSE 0 END) AS
missing_values,
   SUM(CASE WHEN SituationFamiliale = '?' THEN 1 ELSE 0 END) AS
question_marks,
   SUM(CASE WHEN SituationFamiliale = 'N/D' THEN 1 ELSE 0 END) AS
nd_values
FROM
   client_12_ext;
```

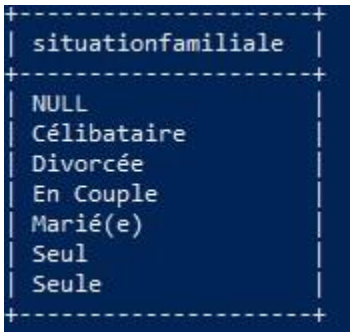


missing_values	question_marks	nd_values
0	0	0

3.5.7 Transformation des valeurs de la colonne "situationFamiliale" :

- Vous transformez les valeurs "Seul", "Seule" et "Divorcée" en "Célibataire" et les valeurs "Marié(e)" en "En Couple".

```
0: jdbc:hive2://localhost:10000>SELECT DISTINCT SituationFamiliale FROM
client_12_ext;
```



situationfamiliale
NULL
Célibataire
Divorcée
En Couple
Marié(e)
Seul
Seule

```
> db.Clients_12.distinct("situationFamiliale");
[
        "En Couple",
        "Célibataire",
        "Seule",
        null,
        "Marié(e)",
        "Seul",
        "Divorcée"
]
> db.Clients_12.updateMany(
...         { situationFamiliale: { $in: ["Seul", "Seule", "Divorcée"] } },
...         { $set: { situationFamiliale: "Célibataire" } }
...     );
{ "acknowledged" : true, "matchedCount" : 5316, "modifiedCount" : 5316 }
> db.Clients_12.updateMany(
...         { situationFamiliale: "Marié(e)" },
...         { $set: { situationFamiliale: "En Couple" } }
...     );
{ "acknowledged" : true, "matchedCount" : 647, "modifiedCount" : 647 }
> db.Clients_12.distinct("situationFamiliale");
[ "En Couple", "Célibataire", null ]
>
```

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT SituationFamiliale FROM client_12_ext;
```

```
+---------------------+
|  situationfamiliale  |
+---------------------+
| NULL                |
| Célibataire         |
| En Couple           |
+---------------------+
```

3.5.8 Nettoyage des données relative à la colonne « nbenfantsacharge »

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT NbEnfantsAcharge  FROM client_12_ext;
```

```
+--------------------+
| nbenfantsacharge   |
+--------------------+
| NULL               |
| -1                 |
| 0                  |
| 1                  |
| 2                  |
| 3                  |
| 4                  |
+--------------------+
```

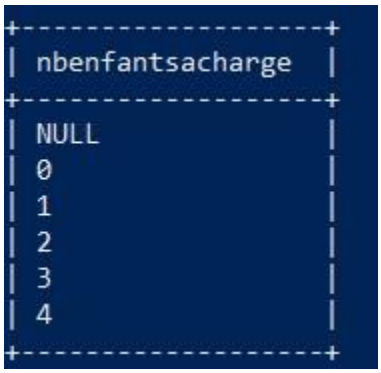
```
0: jdbc:hive2://localhost:10000> SELECT COUNT(*)
FROM client_12_ext
WHERE NbEnfantsAcharge= -1 OR NbEnfantsAcharge  IS NULL;
```

```
+------+
| _c0  |
+------+
| 286  |
+------+
```

```
> db.Clients_12.distinct("nbEnfantsAcharge");
[ 4, 0, 3, 1, 2, -1, "", "?" ]
```

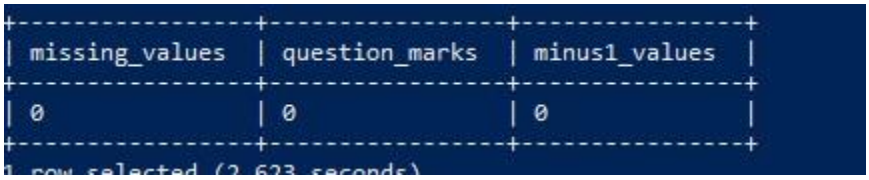
```
  db.Clients_12.updateMany(
...     { nbEnfantsAcharge: { $in: ["", -1, "?"] } },
...     { $set: { nbEnfantsAcharge: null } }
...   );
{ "acknowledged" : true, "matchedCount" : 286, "modifiedCount" : 286 }
> db.Clients_12.find({ nbEnfantsAcharge: { $in: ["", -1, "?"] } }).count();
0
> db.Clients_12.distinct("nbEnfantsAcharge");
[ 4, 0, 3, 1, 2, null ]
>
```

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT NbEnfantsAcharge  FROM
client_12_ext;
```



```
+--------------------+
| nbenfantsacharge   |
+--------------------+
| NULL               |
| 0                  |
| 1                  |
| 2                  |
| 3                  |
| 4                  |
+--------------------+
```

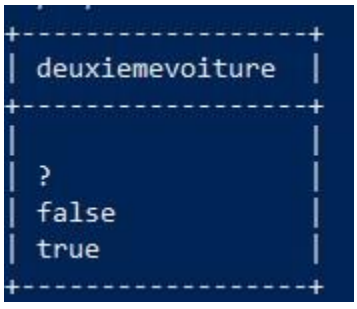
```
0: jdbc:hive2://localhost:10000> SELECT
   SUM(CASE WHEN NbEnfantsAcharge = '' THEN 1 ELSE 0 END) AS
missing_values,
   SUM(CASE WHEN NbEnfantsAcharge = '?' THEN 1 ELSE 0 END) AS
question_marks,
   SUM(CASE WHEN NbEnfantsAcharge = -1 THEN 1 ELSE 0 END) AS
minus1_values
FROM
   client_12_ext;
```



```
+-----------------+-----------------+----------------+
| missing_values  | question_marks  | minus1_values  |
+-----------------+-----------------+----------------+
| 0               | 0               | 0              |
+-----------------+-----------------+----------------+
1 row selected (2.623 seconds)
```

3.5.9 Nettoyage des données relative à la colonne « deuxiemevoiture »

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT deuxiemevoiture  FROM
client_12_ext;
```



```
+--------------------+
| deuxiemevoiture    |
+--------------------+
|                    |
| ?                  |
| false              |
| true               |
+--------------------+
```

```
0: jdbc:hive2://localhost:10000> SELECT
   SUM(CASE WHEN deuxiemevoiture = '' THEN 1 ELSE 0 END) AS
missing_values,
```

```
   SUM(CASE WHEN deuxiemevoiture = '?' THEN 1 ELSE 0 END) AS
question_marks
FROM
   client_12_ext;
```

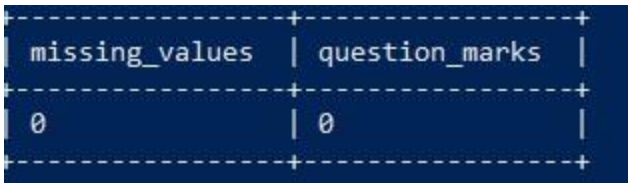
```
+-----------------+-----------------+
| missing_values  | question_marks  |
+-----------------+-----------------+
| 93              | 104             |
+-----------------+-----------------+
```

```
> db.Clients_12.distinct("2eme voiture");
[ "false", "true", "", "?" ]
> db.Clients_12.find({ "2eme voiture": { $in: ["", "?"] } }).count();
197
> db.Clients_12.updateMany(
...       { "2eme voiture": { $in: ["","?"] } },
...       { $set: { "2eme voiture": null } }
...    );
{ "acknowledged" : true, "matchedCount" : 197, "modifiedCount" : 197 }
> db.Clients_12.distinct("2eme voiture");
[ "false", "true", null ]
>
```

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT deuxiemevoiture  FROM
client_12_ext;
```

```
+------------------+
| deuxiemevoiture  |
+------------------+
| NULL             |
| false            |
| true             |
+------------------+
```

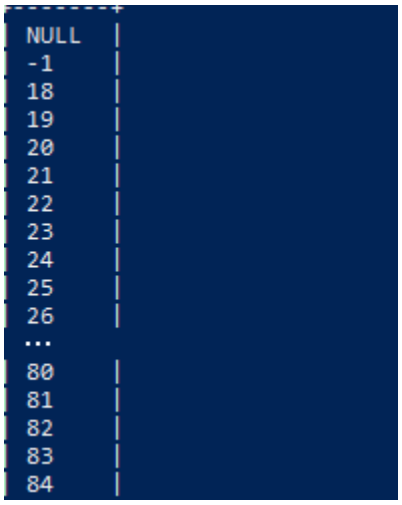
```
0: jdbc:hive2://localhost:10000> SELECT
   SUM(CASE WHEN deuxiemevoiture = '' THEN 1 ELSE 0 END) AS
missing_values,
   SUM(CASE WHEN deuxiemevoiture = '?' THEN 1 ELSE 0 END) AS
question_marks
FROM
   client_12_ext;
```



missing_values	question_marks
0	0

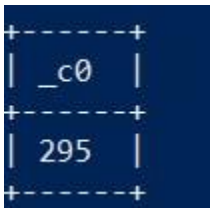
3.5.10 Nettoyage des données relative à la colonne « age »

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT age  FROM client_12_ext;
```



age
NULL
-1
18
19
20
21
22
23
24
25
26
...
80
81
82
83
84

```
0: jdbc:hive2://localhost:10000> SELECT COUNT(*)
FROM client_12_ext
WHERE age = -1 OR age IS NULL;
```



_c0
295

```
db.Clients_12.distinct("age");
      34,
      58,
      57,
      18,
      33,
      84,
      19,
      65,
      53,
      29,
      81,,
      45,
      20,
      71,
      64,
      82,
      73,
      61,
      "",
      "?",
      -1
```

```
> db.Clients_12.find({ "age": { $in: ["", "?",-1] } }).count();
295
> db.Clients_12.updateMany(
...     { "age": { $in: ["", "?", -1] } },
...     { $set: { "age": null } }
...  );
{ "acknowledged" : true, "matchedCount" : 295, "modifiedCount" : 295 }
```

```
> db.Clients_12.distinct("age");
[
        58,
        57,
        18,
        33,
        84,
        19,
        65,
        53,
        29,
        81,
        45,
        20,
        39,
        72,
        56,
        42,
        55,
        26,
        23,
        73,
        61,
        null
]
```

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT age  FROM client_12_ext;
```

```
|  age  |
+-------+
| NULL  |
| 18    |
| 19    |
| 20    |
| 21    |
| 22    |
| 23    |
| 24    |
| 79    |
| 80    |
| 81    |
| 82    |
| 83    |
| 84    |
```

```
0: jdbc:hive2://localhost:10000> SELECT
   SUM(CASE WHEN Age = '' THEN 1 ELSE 0 END) AS missing_values,
   SUM(CASE WHEN Age = '?' THEN 1 ELSE 0 END) AS question_marks,
   SUM(CASE WHEN Age = -1 THEN 1 ELSE 0 END) AS minus1_values
FROM
   client_12_ext;
```

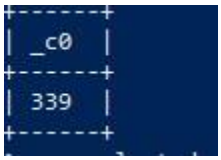
```
+-----------------+-----------------+----------------+
| missing_values  | question_marks  | minus1_values  |
+-----------------+-----------------+----------------+
| 0               | 0               | 0              |
+-----------------+-----------------+----------------+
1 row selected (2.623 seconds)
```

3.5.11 Nettoyage des données relative à la colonne « taux »

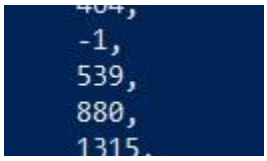
```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT taux  FROM client_12_ext;
```

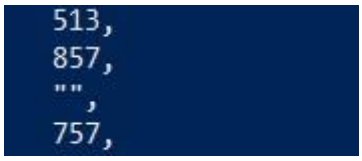
```
+-------+
| taux  |
+-------+
| NULL  |
| -1    |
| 150   |
| 151   |
| 152   |
| 153   |
| 154   |
| 155   |
| 156   |
```

```
0: jdbc:hive2://localhost:10000> SELECT COUNT(*)
FROM client_12_ext
WHERE taux= -1 OR taux  IS NULL; ==>
```




```
> db.Clients_12.distinct("taux");
```






```
> db.Clients_12.find({ taux: { $in: ["", -1, "?"] } }).count();
339
>
```

```
> db.Clients_12.updateMany(
...     { taux: { $in: ["", -1, "?"] } },
...     { $set: { taux: null } }
...  );
{ "acknowledged" : true, "matchedCount" : 339, "modifiedCount" : 339 }
>
```

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT taux  FROM client_12_ext;
```

```
+--------+
| taux   |
+--------+
| NULL   |
| 150    |
| 151    |
| 152    |
| 153    |
| 154    |
| 155    |
| 156    |
| 157    |
| 158    |
| 159    |
| 160    |
| 161    |
| 162    |
| 163    |
| 164    |
| 165    |
| 166    |
| 167    |
| 168    |
| 169    |
| 170    |
| 171    |
| 172    |
| 173    |
| 174    |
| 175    |
| 176    |
| 177    |
| 178    |
| 179    |
```

```
0: jdbc:hive2://localhost:10000> SELECT
   SUM(CASE WHEN taux = '' THEN 1 ELSE 0 END) AS missing_values,
   SUM(CASE WHEN taux = '?' THEN 1 ELSE 0 END) AS question_marks,
   SUM(CASE WHEN taux = -1 THEN 1 ELSE 0 END) AS minus1_values
FROM
   client_12_ext;
```

```
+-----------------+-----------------+----------------+
| missing_values  | question_marks  | minus1_values  |
+-----------------+-----------------+----------------+
| 0               | 0               | 0              |
+-----------------+-----------------+----------------+
1 row selected (1.762 seconds)
```

3.5.11 Vérification de la conformité du format des immatriculations

```
0: jdbc:hive2://localhost:10000>SELECT DISTINCT immatriculation
FROM client_12_ext
WHERE immatriculation NOT REGEXP '^[0-9]{1,4} [A-Z]{2} [0-9]{2}$';
```

```
+------------------+
| immatriculation  |
+------------------+
+------------------+
```

3.4.12 Gestion des valeurs récurrentes de la colonne « immatriculation »

```
0: jdbc:hive2://localhost:10000> SELECT immatriculation, COUNT(*) AS
nb_occurrences
FROM client_12_ext
GROUP BY immatriculation
HAVING COUNT(*) > 1;
```

```
+------------------+------------------+
| immatriculation  | nb_occurrences   |
+------------------+------------------+
| 105 WI 84        | 2                |
| 1608 CZ 60       | 2                |
| 203 TL 35        | 2                |
| 5507 KS 86       | 2                |
| 5546 NA 78       | 2                |
| 568 ZI 43        | 2                |
| 5907 OD 56       | 2                |
| 7002 DU 26       | 2                |
| 8264 EA 89       | 2                |
| 8521 EV 85       | 2                |
| 8985 RJ 57       | 2                |
| 9065 KM 42       | 2                |
+------------------+------------------+
```

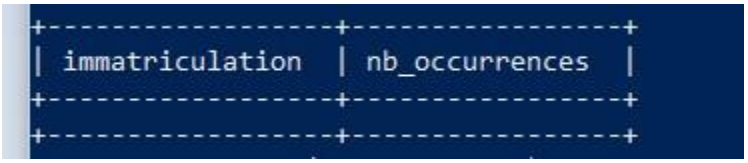
```
0: jdbc:hive2://localhost:10000> SELECT * FROM client_12_ext WHERE
immatriculation IN
   (SELECT immatriculation FROM client_12_ext
    GROUP BY immatriculation HAVING COUNT(*) > 1);
```

client_12_ext.age	client_12_ext.sexe	client_12_ext.taux	client_12_ext.situationfamiliale	client_12_ext.nbenfantsacharge	client_12_ext.deuxiemevoiture	client_12_ext.immatriculation
56	M	171	Célibataire	0	false	105 WI 84
63	M	837	Célibataire	0	false	105 WI 84
50	F	937	En Couple	3	false	1608 CZ 60
25	F	1039	En Couple	1	false	1608 CZ 60
28	M	753	En Couple	2	true	203 TL 35
27	F	599	En Couple	0	true	203 TL 35
24	M	893	En Couple	2	false	5507 KS 86
21	M	1310	En Couple	3	true	5507 KS 86
50	F	164	En Couple	2	false	5546 NA 78
73	F	1104	Célibataire	0	false	5546 NA 78
58	M	434	En Couple	4	false	568 ZI 43
34	M	1114	Célibataire	0	false	568 ZI 43
76	M	1037	Célibataire	0	false	5907 OD 56
27	M	797	Célibataire	0	false	5907 OD 56
35	M	1020	Célibataire	0	false	7002 DU 26
62	M	780	En Couple	1	false	7002 DU 26
82	M	559	Célibataire	0	false	8264 EA 89
38	F	553	Célibataire	0	false	8264 EA 89
68	F	1144	En Couple	3	true	8521 EV 85
49	M	1105	Célibataire	0	false	8521 EV 85
33	F	493	En Couple	1	false	8985 RJ 57
55	F	166	En Couple	1	false	8985 RJ 57
23	M	211	En Couple	2	false	9065 KM 42
81	F	573	En Couple	3	false	9065 KM 42

Nous constatons que 12 paires d'immatriculations des clients diffèrent entre elles, rendant ainsi difficile l'adoption d'une solution générale pour résoudre cette ambiguïté d'unicité. Une solution simple et possible serait de supprimer une occurrence et d'en conserver une seule, étant donné que chaque immatriculation se répète deux fois.

```
> db.Clients_12.aggregate([
...       { $group: { _id: "$immatriculation", count: { $sum: 1 } } },
...       { $match: { count: { $gt: 1 } } }
...   ]);
{ "_id" : "8521 EV 85", "count" : 2 }
{ "_id" : "1608 CZ 60", "count" : 2 }
{ "_id" : "9065 KM 42", "count" : 2 }
{ "_id" : "105 WI 84", "count" : 2 }
{ "_id" : "5507 KS 86", "count" : 2 }
{ "_id" : "8264 EA 89", "count" : 2 }
{ "_id" : "5907 OD 56", "count" : 2 }
{ "_id" : "203 TL 35", "count" : 2 }
{ "_id" : "7002 DU 26", "count" : 2 }
{ "_id" : "5546 NA 78", "count" : 2 }
{ "_id" : "568 ZI 43", "count" : 2 }
{ "_id" : "8985 RJ 57", "count" : 2 }
> var immatriculationsRepetees = db.Clients_12.aggregate([
...       { $group: { _id: "$immatriculation", count: { $sum: 1 } } },
...       { $match: { count: { $gt: 1 } } }
... ]).toArray();
> immatriculationsRepetees.forEach(function(doc) {
...       var immatriculation = doc._id
...       db.Clients_12.deleteOne({ immatriculation: immatriculation });
... });
> db.Clients_12.count();
99988
>
```

```
0: jdbc:hive2://localhost:10000> SELECT immatriculation, COUNT(*) AS
nb_occurrences
FROM client_12_ext
GROUP BY immatriculation
HAVING COUNT(*) > 1;
```



immatriculation	nb_occurrences

3.5.12 Gestion des valeurs NULL de la table « client_12_ext »

```
0: jdbc:hive2://localhost:10000> SELECT *
FROM client_12_ext
WHERE immatriculation is NULL OR age is NULL  OR sexe is NULL OR taux is
NULL OR situationFamiliale is NULL  OR nbEnfantsAcharge IS NULL OR
deuxiemevoiture IS NULL;
```

NULL	F	496	En Couple	1	true	282 HF 46
29	F	NULL	En Couple	1	false	9327 PG 50
65	M	NULL	En Couple	2	false	8630 XJ 90
NULL	M	206	En Couple	4	false	7083 SY 97
63	M	421	En Couple	NULL	false	2126 NC 80
55	M	1024	NULL	3	false	8417 KC 41
70	M	592	En Couple	4	NULL	737 CQ 65
24	M	NULL	En Couple	2	false	1240 PB 98
76	M	NULL	En Couple	1	false	3011 CO 62
NULL	M	569	Célibataire	0	false	8654 ET 43
33	F	568	NULL	0	false	299 TF 40
22	F	435	En Couple	NULL	false	5539 KF 91
30	F	NULL	En Couple	0	false	6168 BW 31
29	M	NULL	En Couple	3	false	485 BJ 83
29	M	194	NULL	2	false	1858 VK 65
51	M	NULL	Célibataire	2	false	4515 HE 59
53	M	420	NULL	1	false	9877 AC 60
29	NULL	1254	En Couple	2	false	9553 DZ 56
NULL	M	424	En Couple	2	false	2301 CB 76
35	NULL	517	En Couple	0	false	5286 FS 65
21	F	206	En Couple	NULL	true	9331 SQ 40
64	M	NULL	Célibataire	0	false	8150 EM 60
38	M	557	NULL	2	false	8211 OL 43
55	M	781	NULL	1	false	7437 MR 66
58	F	473	En Couple	3	NULL	5668 ND 10
70	M	568	NULL	1	true	439 NO 47
62	M	226	NULL	1	false	8421 SC 67
23	M	463	Célibataire	0	NULL	4173 LD 83
26	F	NULL	Célibataire	0	false	7855 KB 12
30	NULL	570	Célibataire	0	false	5851 CW 55
NULL	F	740	En Couple	0	false	631 PF 20
38	M	NULL	En Couple	1	false	961 XR 13
26	M	NULL	En Couple	3	true	767 DC 90
NULL	M	1169	Célibataire	0	false	9719 XF 31
NULL	M	417	Célibataire	0	false	5532 AH 80
78	M	NULL	Célibataire	0	false	6722 BB 71
47	M	200	En Couple	NULL	false	5657 SL 36
NULL	M	552	En Couple	4	false	2524 XW 73
23	M	NULL	En Couple	0	false	4387 QE 55
35	F	561	Célibataire	NULL	false	4330 LW 70
22	F	566	Célibataire	0	NULL	4880 HB 97
23	M	858	En Couple	1	NULL	8742 CM 18
71	NULL	835	En Couple	2	false	2983 SD 52
NULL	F	545	Célibataire	0	false	6901 UF 31
28	F	1180	NULL	0	false	5247 BU 47
27	M	NULL	En Couple	0	false	9742 YA 91
22	M	555	NULL	0	true	1547 TW 71
41	M	784	NULL	0	false	1873 BK 62
78	M	1223	En Couple	NULL	NULL	1433 HA 51
67	NULL	567	En Couple	0	true	5297 YB 21
29	M	227	Célibataire	NULL	false	5509 PY 22
30	M	231	NULL	0	false	7180 UV 44
26	NULL	933	En Couple	4	false	439 SW 96
28	M	1274	En Couple	NULL	false	7023 QY 10
20	M	161	En Couple	3	NULL	4526 VW 22
40	NULL	1113	Célibataire	1	false	3378 YG 75
20	M	581	Célibataire	0	NULL	3714 XU 54
NULL	M	925	Célibataire	2	false	1966 XL 73
20	M	932	En Couple	NULL	false	5165 ZG 62
18	M	590	NULL	0	true	7567 QU 59
26	M	1068	NULL	2	false	1269 NB 92
84	F	195	En Couple	NULL	true	1282 HO 95
65	F	511	En Couple	NULL	false	851 UK 37
48	M	1001	En Couple	2	NULL	2790 VO 80
NULL	M	1269	Célibataire	0	false	377 IO 10
44	M	181	Célibataire	0	NULL	607 XD 90

```
0: jdbc:hive2://localhost:10000>SELECT *
FROM (
   SELECT t.*,
      (CASE WHEN immatriculation IS NULL THEN 1 ELSE 0 END) +
      (CASE WHEN age IS NULL THEN 1 ELSE 0 END) +
      (CASE WHEN sexe IS NULL THEN 1 ELSE 0 END) +
      (CASE WHEN taux IS NULL THEN 1 ELSE 0 END) +
      (CASE WHEN situationFamiliale IS NULL THEN 1 ELSE 0 END) +
      (CASE WHEN nbEnfantsAcharge IS NULL THEN 1 ELSE 0 END) +
      (CASE WHEN deuxiemevoiture IS NULL THEN 1 ELSE 0 END) AS
nb_colonnes_null
   FROM client_12_ext t
) t
WHERE (immatriculation IS NULL OR age IS NULL OR sexe IS NULL OR taux IS
NULL OR situationFamiliale IS NULL OR nbEnfantsAcharge IS NULL OR
deuxiemevoiture IS NULL);
```

Nous remarquons que la grande majorité des colonnes contiennent une seule valeur NULL pour chaque ligne, toutefois, cela requiert une vérification.

```
0: jdbc:hive2://localhost:10000>SELECT *
FROM (
   SELECT t.*,
      (CASE WHEN immatriculation IS NULL THEN 1 ELSE 0 END) +
      (CASE WHEN age IS NULL THEN 1 ELSE 0 END) +
      (CASE WHEN sexe IS NULL THEN 1 ELSE 0 END) +
      (CASE WHEN taux IS NULL THEN 1 ELSE 0 END) +
      (CASE WHEN situationFamiliale IS NULL THEN 1 ELSE 0 END) +
      (CASE WHEN nbEnfantsAcharge IS NULL THEN 1 ELSE 0 END) +
      (CASE WHEN deuxiemevoiture IS NULL THEN 1 ELSE 0 END) AS
nb_colonnes_null
   FROM client_12_ext t
) t
WHERE (immatriculation IS NULL OR age IS NULL OR sexe IS NULL OR taux IS
NULL OR situationFamiliale IS NULL OR nbEnfantsAcharge IS NULL OR
deuxiemevoiture IS NULL) AND nb_colonnes_null >= 2;
```

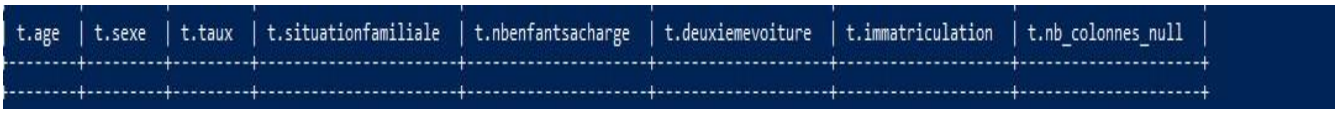
t.age	t.sexe	t.taux	t.situationfamiliale	t.nbenfantsacharge	t.deuxiemevoiture	t.immatriculation	t.nb_colonnes_null
27	NULL	NULL	Célibataire	0	false	2271 GG 52	2
NULL	M	NULL	En Couple	0	false	1561 TI 75	2
58	M	NULL	NULL	4	false	1081 RP 47	2
38	NULL	NULL	En Couple	2	false	627 AX 72	2
56	NULL	243	En Couple	NULL	false	5170 NO 84	2
79	M	473	NULL	NULL	true	3188 US 65	2
42	M	NULL	NULL	4	false	1664 KZ 82	2
NULL	M	NULL	En Couple	2	false	5091 LA 41	2
NULL	F	NULL	En Couple	1	false	3239 MU 15	2
55	NULL	545	Célibataire	NULL	false	9655 NC 34	2
78	M	1223	En Couple	NULL	NULL	1433 HA 51	2

```
> db.Clients_12.aggregate([
...   {
...     $addFields: {
...       nb_colonnes_null: {
...         $sum: [
...           { $cond: [{ $eq: ["$immatriculation", null] }, 1, 0] },
...           { $cond: [{ $eq: ["$age", null] }, 1, 0] },
...           { $cond: [{ $eq: ["$sexe", null] }, 1, 0] },
...           { $cond: [{ $eq: ["$taux", null] }, 1, 0] },
...           { $cond: [{ $eq: ["$situationFamiliale", null] }, 1, 0] },
...           { $cond: [{ $eq: ["$nbEnfantsAcharge", null] }, 1, 0] },
...           { $cond: [{ $eq: ["$2eme voiture", null] }, 1, 0] }
...         ]
...       }
...     }
...   },
...   {
...     $match: {
...       $and: [
...         { $or: [{ immatriculation: null }, { age: null }, { sexe: null }, { taux: null }, { situationFamiliale: null }, { nbEnfantsAcharge: null }, { "2eme voiture": null }] },
...         { nb_colonnes_null: { $gte: 2 } }
...       ]
...     }
...   }
... ])
{ "_id" : ObjectId("661581da4817545a7e046f36"), "age" : 27, "sexe" : null, "taux" : null, "situationFamiliale" : "Célibataire", "nbEnfantsAcharge" : 0, "2eme voiture" : "false", "immatriculation" : "2271 GG 52", "nb_colonnes_null" : 2 }
{ "_id" : ObjectId("661581db4817545a7e04bdfc"), "age" : null, "sexe" : "M", "taux" : null, "situationFamiliale" : "En Couple", "nbEnfantsAcharge" : 0, "2eme voiture" : "false", "immatriculation" : "1561 TI 75", "nb_colonnes_null" : 2 }
{ "_id" : ObjectId("661581db4817545a7e04e997"), "age" : 58, "sexe" : "M", "taux" : null, "situationFamiliale" : null, "nbEnfantsAcharge" : 4, "2eme voiture" : "false", "immatriculation" : "1081 RP 47", "nb_colonnes_null" : 2 }
{ "_id" : ObjectId("661581db4817545a7e04ece7"), "age" : 38, "sexe" : null, "taux" : null, "situationFamiliale" : "En Couple", "nbEnfantsAcharge" : 2, "2eme voiture" : "false", "immatriculation" : "627 AX 72", "nb_colonnes_null" : 2 }
{ "_id" : ObjectId("661581db4817545a7e04ed13"), "age" : 56, "sexe" : null, "taux" : 243, "situationFamiliale" : "En Couple", "nbEnfantsAcharge" : null, "2eme voiture" : "false", "immatriculation" : "5170 NO 84", "nb_colonnes_null" : 2 }
{ "_id" : ObjectId("661581db4817545a7e054583"), "age" : 79, "sexe" : "M", "taux" : 473, "situationFamiliale" : null, "nbEnfantsAcharge" : null, "2eme voiture" : "true", "immatriculation" : "3188 US 65", "nb_colonnes_null" : 2 }
{ "_id" : ObjectId("661581db4817545a7e055946"), "age" : 42, "sexe" : "M", "taux" : null, "situationFamiliale" : null, "nbEnfantsAcharge" : 4, "2eme voiture" : "false", "immatriculation" : "1664 KZ 82", "nb_colonnes_null" : 2 }
{ "_id" : ObjectId("661581db4817545a7e055eaa"), "age" : null, "sexe" : "M", "taux" : null, "situationFamiliale" : "En Couple", "nbEnfantsAcharge" : 2, "2eme voiture" : "false", "immatriculation" : "5091 LA 41", "nb_colonnes_null" : 2 }
{ "_id" : ObjectId("661581db4817545a7e059323"), "age" : null, "sexe" : "F", "taux" : null, "situationFamiliale" : "En Couple", "nbEnfantsAcharge" : 1, "2eme voiture" : "false", "immatriculation" : "3239 MU 15", "nb_colonnes_null" : 2 }
{ "_id" : ObjectId("661581dc4817545a7e05a4f8"), "age" : 55, "sexe" : null, "taux" : 545, "situationFamiliale" : "Célibataire", "nbEnfantsAcharge" : null, "2eme voiture" : "false", "immatriculation" : "9655 NC 34", "nb_colonnes_null" : 2
}
{ "_id" : ObjectId("661581dc4817545a7e05ed27"), "age" : 78, "sexe" : "M", "taux" : 1223, "situationFamiliale" : "En Couple", "nbEnfantsAcharge" : null, "2eme voiture" : null, "immatriculation" : "1433 HA 51", "nb_colonnes_null" : 2 }
```

Nous constatons qu'il y a 11 enregistrements pour lesquelles le nombre de colonnes contenant la valeur NULL est de 2, et que plus de la moitié d'entre elles ont au moins des valeurs NULL dans les colonnes sexe, âge et nbEnfantAcharge, des données critiques et importantes pour notre analyse. Par conséquent, nous concluons qu'il est nécessaire de supprimer ces 11 lignes, car elles ne présentent pas une grande valeur significative par rapport aux lignes qui n'ont qu'une seule colonne à valeur NULL.

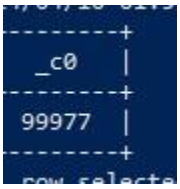
```
> var documentsASupprimer = db.Clients_12.aggregate([
...   {
...     $addFields: {
...       nb_colonnes_null: {
...         $sum: [
...           { $cond: [{ $eq: ["$immatriculation", null] }, 1, 0] },
...           { $cond: [{ $eq: ["$age", null] }, 1, 0] },
...           { $cond: [{ $eq: ["$sexe", null] }, 1, 0] },
...           { $cond: [{ $eq: ["$taux", null] }, 1, 0] },
...           { $cond: [{ $eq: ["$situationFamiliale", null] }, 1, 0] },
...           { $cond: [{ $eq: ["$nbEnfantsAcharge", null] }, 1, 0] },
...           { $cond: [{ $eq: ["$2eme voiture", null] }, 1, 0] }
...         ]
...       }
...     }
...   },
...   {
...     $match: {
...       $and: [
...         { $or: [{ immatriculation: null }, { age: null }, { sexe: null },
...         { taux: null }, { situationFamiliale: null }, { nbEnfantsAcharge: null }, { "2eme voiture": null }] },
...         { nb_colonnes_null: { $gte: 2 } }
...       ]
...     }
...   }
... ]).toArray();
> documentsASupprimer.forEach(function(document) {
...   db.Clients_12.deleteOne({ _id: document._id });
... });
```

```
0: jdbc:hive2://localhost:10000> SELECT *
FROM (
   SELECT t.*,
      (CASE WHEN immatriculation IS NULL THEN 1 ELSE 0 END) +
      (CASE WHEN age IS NULL THEN 1 ELSE 0 END) +
      (CASE WHEN sexe IS NULL THEN 1 ELSE 0 END) +
      (CASE WHEN taux IS NULL THEN 1 ELSE 0 END) +
      (CASE WHEN situationFamiliale IS NULL THEN 1 ELSE 0 END) +
      (CASE WHEN nbEnfantsAcharge IS NULL THEN 1 ELSE 0 END) +
      (CASE WHEN deuxiemevoiture IS NULL THEN 1 ELSE 0 END) AS
nb_colonnes_null
   FROM client_12_ext t
) t
WHERE (immatriculation IS NULL OR age IS NULL OR sexe IS NULL OR taux IS
NULL OR situationFamiliale IS NULL OR nbEnfantsAcharge IS NULL OR
deuxiemevoiture IS NULL) AND nb_colonnes_null >= 2;
```



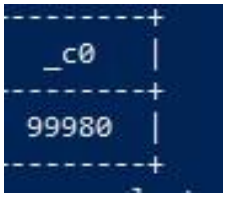
3.5.14 Dimension final de la table « client_12_ext »

```
0: jdbc:hive2://localhost:10000> SELECT COUNT(*) from client_12_ext;
```

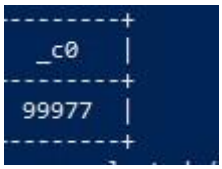


3.6 Construction du model d'analyse : Jointure entre les données Clients et Immatriculation

```
0: jdbc:hive2://localhost:10000>SELECT COUNT(*) from client_7_ext;
```



0: jdbc:hive2://localhost:10000>SELECT COUNT(*) from client_12_ext;



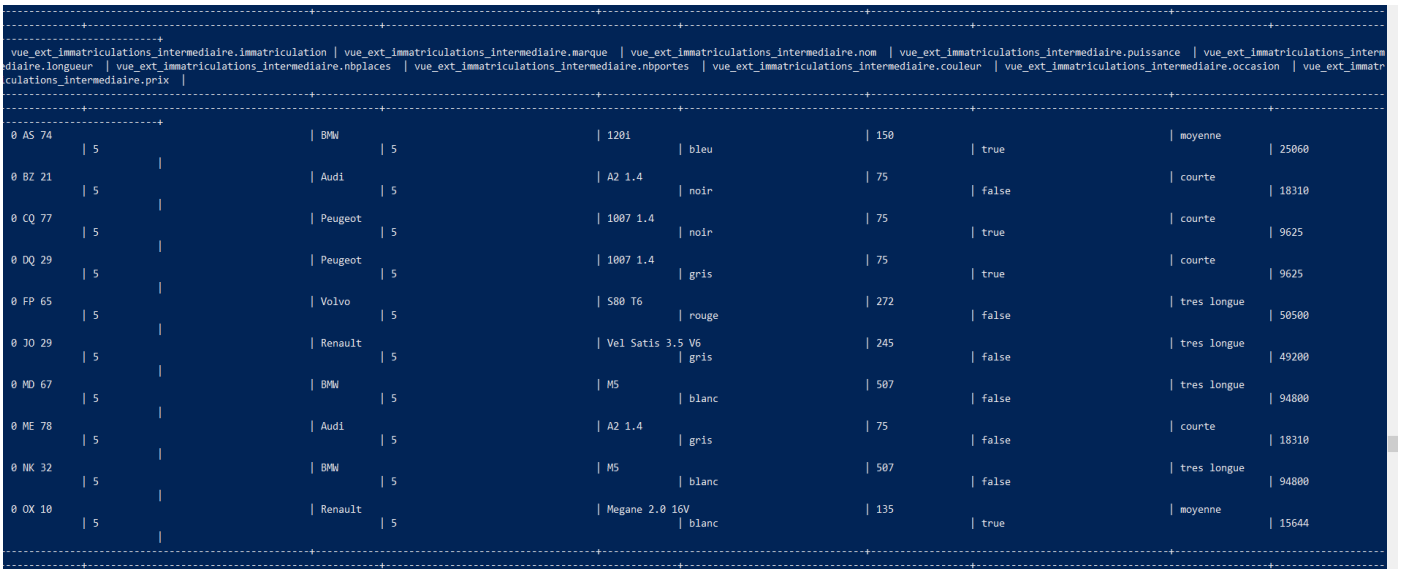
```
+---------+
|   _c0   |
+---------+
| 99977   |
+---------+
```

• En admettant une fusion des deux tables et si on suppose que les immatriculations des clients soient uniques entre elles, le nombre d'immatriculations attendu serait de 199957.

Mise en place d'une vue matérialisée intermédiaire incluant les données d'immatriculation des clients :

```
0: jdbc:hive2://localhost:10000> CREATE MATERIALIZED VIEW vue_ext_immatriculations_intermediaire AS
. . . . . .> SELECT *
. . . . . .> FROM table_ext_immatriculations
. . . . . .> WHERE immatriculation IN (
. . . . . .>     SELECT immatriculation FROM client_7_ext
. . . . . .>     UNION
. . . . . .>     SELECT immatriculation FROM client_12_ext
. . . . . .> );
```

0: jdbc:hive2://localhost:10000> SELECT * from vue_ext_immatriculations_intermediaire LIMIT 10;



vue_ext_immatriculations_intermediaire.immatriculation	vue_ext_immatriculations_intermediaire.marque	vue_ext_immatriculations_intermediaire.nom	vue_ext_immatriculations_intermediaire.puissance	vue_ext_immatriculations_intermediaire.longueur	vue_ext_immatriculations_intermediaire.nbplaces	vue_ext_immatriculations_intermediaire.nbportes	vue_ext_immatriculations_intermediaire.couleur	vue_ext_immatriculations_intermediaire.occasion	vue_ext_immatriculations_intermediaire.prix
0 AS 74	BMW	120i	150	moyenne	5	5	bleu	true	25060
0 BZ 21	Audi	A2 1.4	75	courte	5	5	noir	false	18310
0 CQ 77	Peugeot	1007 1.4	75	courte	5	5	noir	true	9625
0 DQ 29	Peugeot	1007 1.4	75	courte	5	5	gris	true	9625
0 FP 65	Volvo	S80 T6	272	tres longue	5	5	rouge	false	50500
0 JO 29	Renault	Vel Satis 3.5 V6	245	tres longue	5	5	gris	false	49200
0 MD 67	BMW	M5	507	tres longue	5	5	blanc	false	94800
0 ME 78	Audi	A2 1.4	75	courte	5	5	gris	false	18310
0 NK 32	BMW	M5	507	tres longue	5	5	blanc	false	94800
0 OX 10	Renault	Megane 2.0 16V	135	moyenne	5	5	blanc	true	15644

```
0: jdbc:hive2://localhost:10000> SELECT COUNT(*) from
vue_ext_immatriculations_intermediaire;
```

```
+---------+
|   _c0   |
+---------+
| 199943  |
+---------+
```

Lors de l'insertion des immatriculations de client_7_ext et client_12_ext dans une vue matérialisée vue_ext_immatriculations_intermediaire, qui englobe l'ensemble des données d'immatriculation des clients, nous remarquons une différence de nombre. Nous pouvons émettre une hypothèse qu'il existe des immatriculations communes entre les deux tables client_7_ext et client_12_ext, et nous devons vérifier cette hypothèse.

```
0: jdbc:hive2://localhost:10000> SELECT immatriculation, COUNT(*) AS
nb_occurrences
FROM (
   SELECT immatriculation FROM client_7_ext
   UNION ALL
   SELECT immatriculation FROM client_12_ext
) AS combined_clients
GROUP BY immatriculation
HAVING COUNT(*) > 1;
```

immatriculation	nb_occurrences
186 MN 34	2
2298 CY 44	2
3014 OR 44	2
3786 ZM 27	2
3910 NF 86	2
3989 YB 64	2
5617 WE 28	2
5886 NW 27	2
7145 CH 89	2
7528 CX 55	2
8521 UD 59	2
8960 GM 31	2
9105 PG 41	2
9505 VT 49	2

Effectivement, nous avons identifié 14 immatriculations qui se répètent entre les tables client_7_ext et client_12_ext, ce qui explique la disparité entre les nombres observés précédemment. Par conséquent, il est impératif de prendre cet élément en compte lors de la jointure entre les données des clients et celles des immatriculations.

Insertion des données de Client_7 de la table client_7_ext en les fusionnant avec les données d'immatriculation de la vue matérialisée vue_ext_immatriculations_intermediaire créée précédemment, et stocker les données dans une nouvelle table « model_immatriculations_clients » qui sera le model d'analyse de la datalake Hive :

```
: jdbc:hive2://localhost:10000> CREATE TABLE model_immatriculations_clients AS
. . . . . .> SELECT i.*, c7.age, c7.sexe, c7.taux, c7.situationFamiliale, c7.nbEnfantsAcharge, c7.deuxiemevoiture
. . . . . .> FROM vue_ext_immatriculations_intermediaire i
. . . . . .> JOIN client_7_ext c7 ON i.immatriculation = c7.immatriculation;
```

0: jdbc:hive2://localhost:10000> SELECT COUNT(*) from model_immatriculations_clients;

```
+--------+
|  _c0   |
+--------+
| 99980  |
+--------+
1 row selecte
```

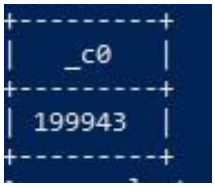
0: jdbc:hive2://localhost:10000> SELECT * from model_immatriculations_clients LIMIT 10;

jointure_immatriculations_clients.immatriculation	jointure_immatriculations_clients.marque	jointure_immatriculations_clients.nom	jointure_immatriculations_clients.puissance	jointure_immatriculations_clients.longueur	jointure_immatriculations_clients.nbplaces	jointure_immatriculations_clients.nbportes	jointure_immatriculations_clients.couleur	jointure_immatriculations_clients.occasion	jointure_immatriculations_clients.prix	jointure_immatriculations_clients.age	jointure_immatriculations_clients.sexe	jointure_immatriculations_clients.taux	jointure_immatriculations_clients.situationfamiliale	jointure_immatriculations_clients.nbenfantsacharge	jointure_immatriculations_clients.deuxiemevoiture
0 BZ 21	Audi	A2 1.4	75	courte	5	5	noir	false	18310	56	M	1382	Célibataire	0	false
0 CQ 77	Peugeot	1007 1.4	75	courte	5	5	noir	true	9625	27	F	239	Célibataire	0	false
0 DQ 29	Peugeot	1007 1.4	75	courte	5	5	gris	true	9625	51	M	234	Célibataire	0	false
0 JO 29	Renault	Vel Satis 3.5 V6	245	tres longue	5	5	gris	false	49200	51	M	552	En Couple	1	false
0 NK 32	BMW	M5	507	tres longue	5	5	blanc	false	94800	39	M	963	En Couple	3	false
0 OX 10	Renault	Megane 2.0 16V	135	moyenne	5	5	blanc	true	15644	75	M	174	Célibataire	0	false
0 RH 46	BMW	M5	507	tres longue	5	5	gris	false	94800	41	F	823	En Couple	0	false
0 UL 83	Volkswagen	Polo 1.2 6V	55	courte	5	3	gris	false	12200	31	M	481	Célibataire	0	false
0 VG 85	Volvo	S80 T6	272	tres longue	5	5	rouge	false	50500	50	M	528	En Couple	3	false
0 WB 68	Jaguar	X-Type 2.5 V6	197	longue	5	5	noir	true	25970	38	F	513	En Couple	0	false

Insertion des données du client_12 en fusionnant les données client avec leurs immatriculations, tout en vérifiant l'absence de ses immatriculations dans les données de la vue matérialisé vue_ext_immatriculations_intermediaire, qui contient les données des clients et leurs immatriculations de Client_7 :

```
INSERT INTO TABLE model_immatriculations_clients
SELECT i.*, c12.age, c12.sexe, c12.taux, c12.situationFamiliale, c12.nbEnfantsAcharge, c12.deuxiemevoiture
FROM table_ext_immatriculations_intermediaire i
JOIN client_12_ext c12 ON i.immatriculation = c12.immatriculation
WHERE i.immatriculation NOT IN (
    SELECT immatriculation FROM model_immatriculations_clients
);
```

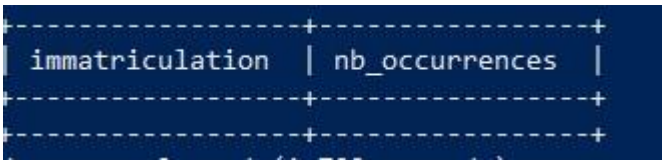
```
0: jdbc:hive2://localhost:10000> SELECT COUNT(*) from
model_immatriculations_clients;
```



Comme on peut le constater, le nombre correspond au nombre initial lors de la création de la vue matérialisée vue_ext_immatriculations_intermediaire, qui englobe les données d'immatriculation de tous les clients.

Validation de l'unicité des immatriculations :

```
0: jdbc:hive2://localhost:10000> SELECT immatriculation, COUNT(*) AS
nb_occurrences
FROM model_immatriculations_clients
GROUP BY immatriculation
HAVING COUNT(*) > 1;
```



Nous pouvons en déduire qu'il n'y a aucune occurrence d'immatriculation.

```
0: jdbc:hive2://localhost:10000>DROP MATERIALIZED VIEW
vue_ext_immatriculations_intermediaire;
```