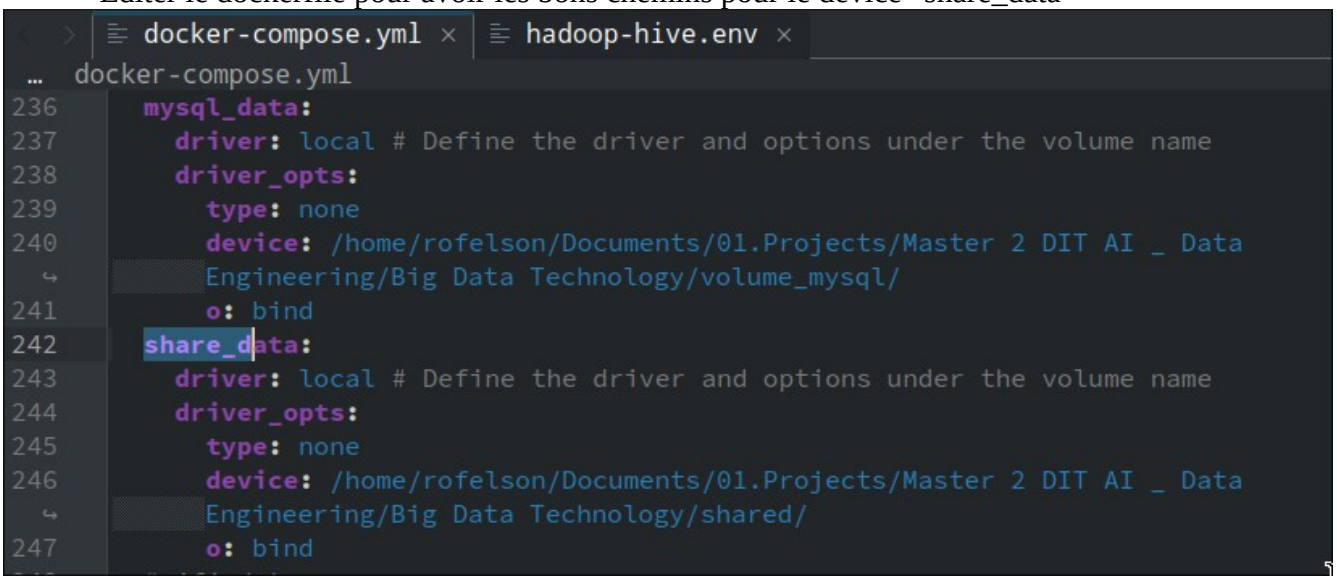
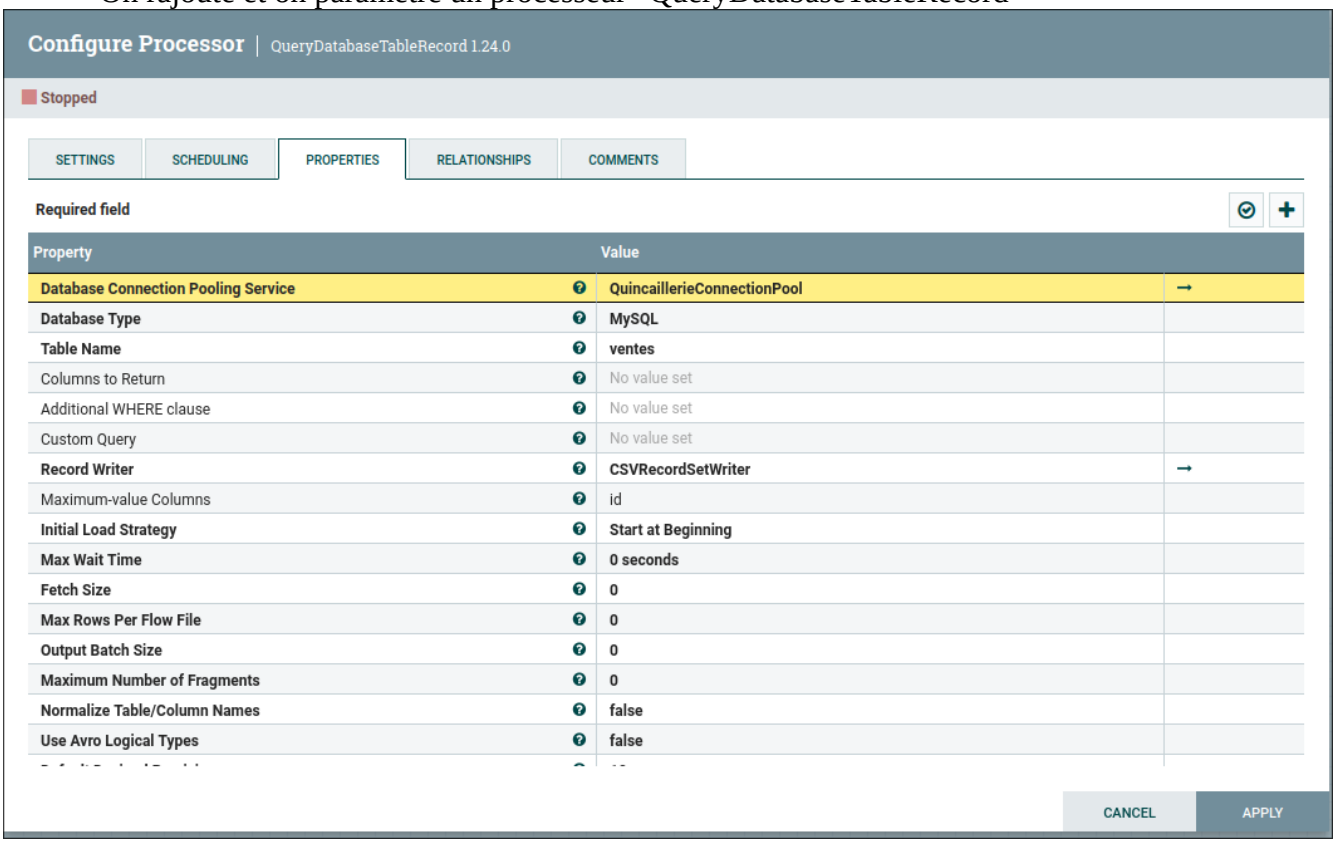
1. Nifi

On commence le travail sur Nifi.

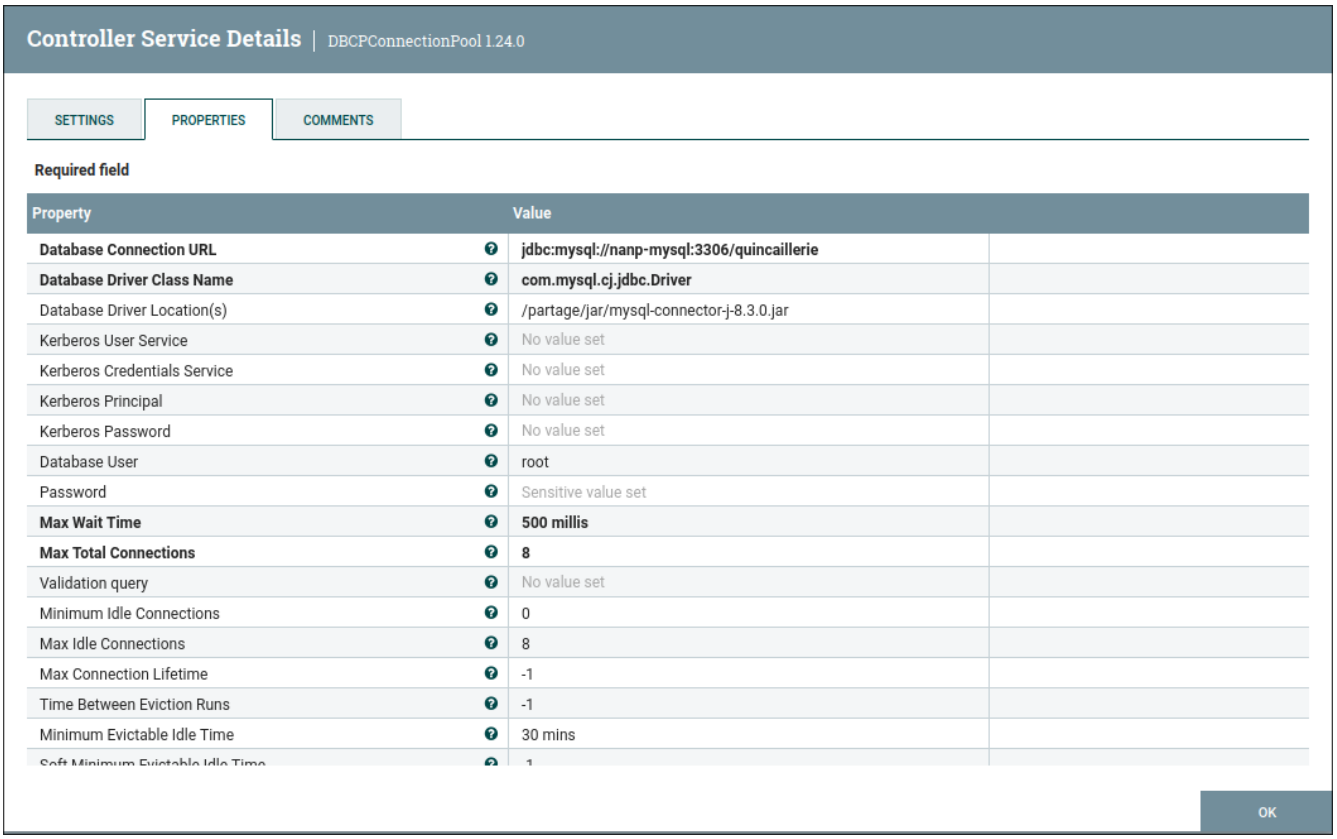
- Editer le dockerfile pour avoir les bons chemins pour le device “share_data”



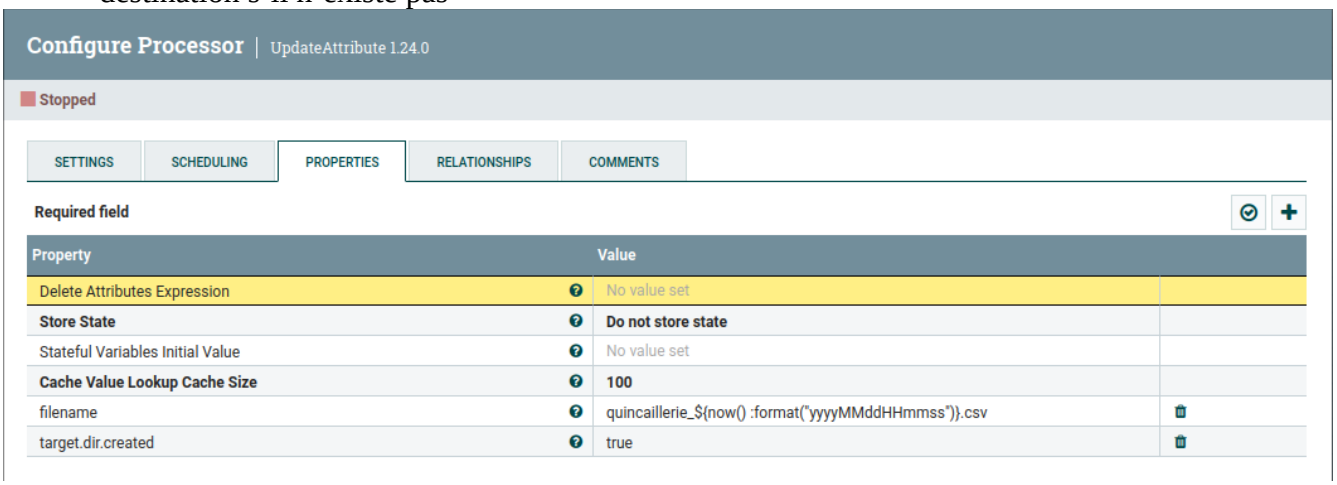
- Copier les fichiers “*-site.xml” vers <shared_data_path>/xml et le connecteur vers <shared_data_path>/jar sur l’hôte
- démarrer les containers avec “podman compose up”
- on se connecte à nifi avec les identifiants qui sont dans le dockerfile
- on va travailler dans le groupe qu’on aura créé “nifi-mysql-connector”
- On rajoute et on paramètre un processeur “QueryDatabaseTableRecord”



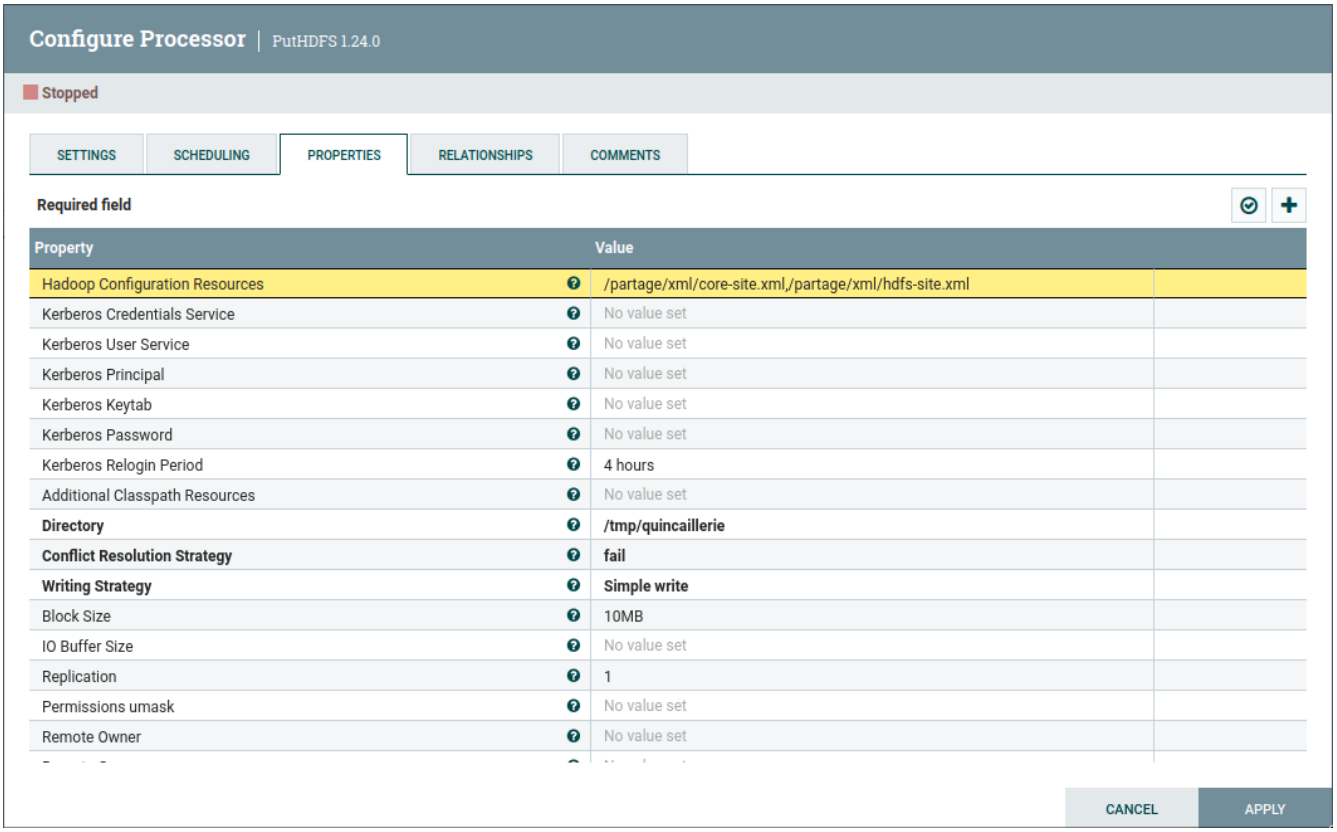
- ON rajoute et on configure la connexion à la base de donnée qu'on a créée au préalable avec Dbeaver



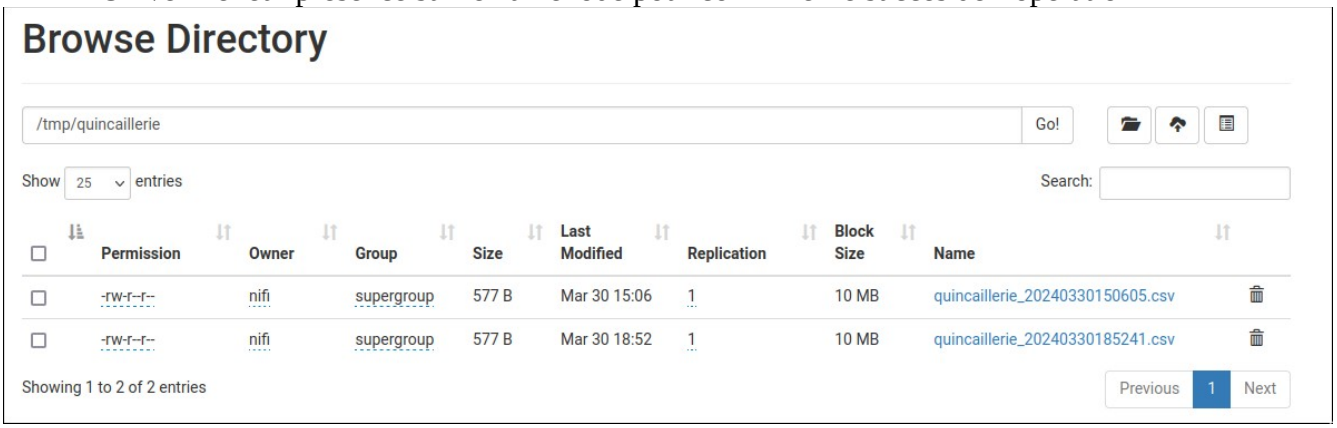
- On rajoute un updateAttribute pour rajouter un nom de fichier et demander à créer le dossier de destination s'il n'existe pas



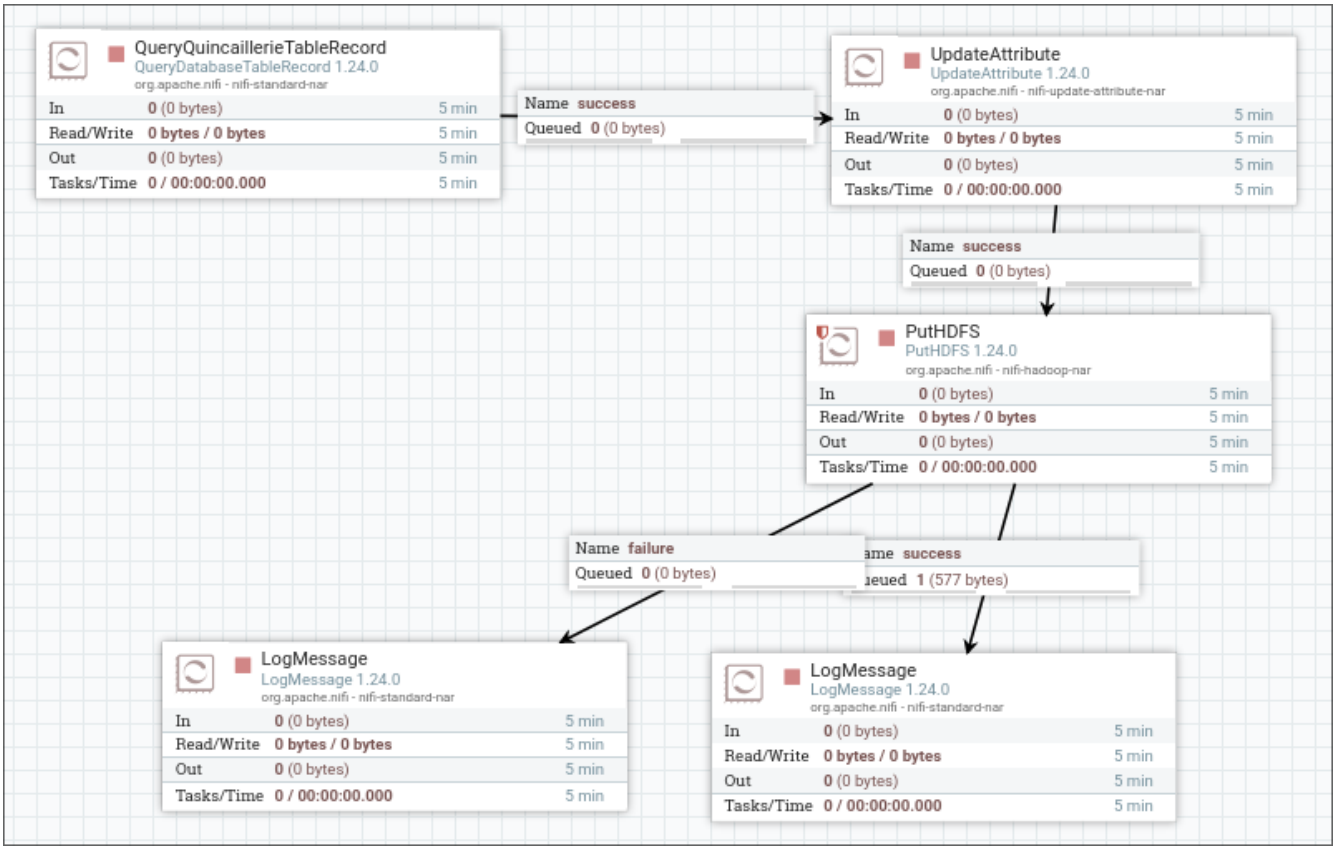
- On rajoute un processeur "putHDFS" pour mettre les infos sur le serveur



- On vérifie leur présence sur le namenode pour confirmer le succès de l'opération



- Le template enrigstré sera joint au mail



2. Hive

On va se connecter au container hive:
“podman exec -it hive-server bash”

- On se connecte à hive avec “jdbc:hive2://hive-server:10000/”
- On crée la base de donnée et la table (la capture provient de mes notes)

- Create appropriate db

```sql
CREATE DATABASE IF NOT EXISTS quincaillerie;
USE quincaillerie;
```

- Create external table quincaillerie

```sql
DROP TABLE quincaillerie;

CREATE EXTERNAL TABLE IF NOT EXISTS quincaillerie(
  id int,
  code_client string,
  code_produit string,
  qte int,
  date_ops bigint,
  pu int,
  montant int
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
COLLECTION ITEMS TERMINATED BY '\n'
LOCATION '/tmp/quincaillerie'
tblproperties ("skip.header.line.count"="1");
```

- On peut observer les données chargées select * from quincaillerie;

```
0: jdbc:hive2://localhost:10000/quincaillerie> select * from quincaillerie;
```

quincaillerie.id	quincaillerie.code_client	quincaillerie.code_produit	quincaillerie.qte	quincaillerie.date_ops	quincaillerie.pu	quincaillerie.montant
1	CLI1	PROD1	3	1710806400000	40	120
2	CLI2	PROD1	5	1710806400000	25	125
3	CLI1	PROD2	2	1710806400000	500	1000
4	CLI3	PROD3	6	1710806400000	100	600
5	CLI3	PROD3	7	1710806400000	100	700
6	CLI4	PROD1	1	1710806400000	120	120
7	CLI1	PROD2	3	1710806400000	500	1500
8	CLI3	PROD1	2	1710806400000	300	600
9	CLI5	PROD2	3	1710806400000	400	1200
10	CLI3	PROD1	7	1710633600000	1000	7000
11	CLI3	PROD3	2	1710806400000	400	800
12	CLI4	PROD2	10	1710806400000	20	200
13	CLI3	PROD2	3	1710806400000	200	600
14	CLI4	PROD4	6	1711238400000	20	300

- On crée une vue pour avoir le revenu total de chaque produit par ordre décroissant

- product income view

```sql
CREATE OR REPLACE VIEW product_income AS
SELECT code_produit, sum(qte) as qte_vendue, sum(montant) as montant_vendu
FROM quincaillerie
GROUP BY code_produit
ORDER BY montant_vendu DESC;
```

```
0: jdbc:hive2://localhost:10000/quincaillerie> select * from product_income;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider
 using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
+-------------------------------+-------------------------------+---------------------------------+
| product_income.code_produit   | product_income.qte_vendue     | product_income.montant_vendu    |
+-------------------------------+-------------------------------+---------------------------------+
| PROD1                         | 36                            | 15930                           |
| PROD2                         | 42                            | 9000                            |
| PROD3                         | 30                            | 4200                            |
| PROD4                         | 12                            | 600                             |
+-------------------------------+-------------------------------+---------------------------------+
```

- On peut rajouter une vue pour où on remet la date dans un format lisible par l'humain

• datetime of date_ops

sql

```sql
CREATE OR REPLACE VIEW original_vendu AS
select id,
  code_client,
  code_produit,
  qte,
  to_date(from_utc_timestamp(date_ops, 'UTC')) as date_ops,
  pu,
  montant
FROM quincaillerie;
```

```
0: jdbc:hive2://localhost:10000/quincaillerie> select * from original_vendu
. . . . . .> ;
+--------------------+-------------------------+--------------------------+---------------------+-------------------------+----------------------+-------------------------+
| original_vendu.id  | original_vendu.code_client | original_vendu.code_produit | original_vendu.qte | original_vendu.date_ops | original_vendu.pu | original_vendu.montant |
+--------------------+-------------------------+--------------------------+---------------------+-------------------------+----------------------+-------------------------+
| 1                  | CLI1                    | PROD1                    | 3                   | 2024-03-19              | 40                   | 120                     |
| 2                  | CLI2                    | PROD1                    | 5                   | 2024-03-19              | 25                   | 125                     |
| 3                  | CLI1                    | PROD2                    | 2                   | 2024-03-19              | 500                  | 1000                    |
| 4                  | CLI3                    | PROD3                    | 6                   | 2024-03-19              | 100                  | 600                     |
| 5                  | CLI3                    | PROD3                    | 7                   | 2024-03-19              | 100                  | 700                     |
| 6                  | CLI4                    | PROD1                    | 1                   | 2024-03-19              | 120                  | 120                     |
| 7                  | CLI1                    | PROD2                    | 3                   | 2024-03-19              | 500                  | 1500                    |
| 8                  | CLI3                    | PROD1                    | 2                   | 2024-03-19              | 300                  | 600                     |
| 9                  | CLI5                    | PROD2                    | 3                   | 2024-03-19              | 400                  | 1200                    |
| 10                 | CLI3                    | PROD1                    | 7                   | 2024-03-17              | 1000                 | 7000                    |
| 11                 | CLI3                    | PROD3                    | 2                   | 2024-03-19              | 400                  | 800                     |
| 12                 | CLI4                    | PROD2                    | 10                  | 2024-03-19              | 20                   | 200                     |
| 13                 | CLI3                    | PROD2                    | 3                   | 2024-03-19              | 200                  | 600                     |
| 14                 | CLI4                    | PROD4                    | 6                   | 2024-03-24              | 20                   | 300                     |
```

- ON peut rajouter une vue pour voir le montant dépensé par chaque client

• best spending client view

sql

```sql
CREATE OR REPLACE VIEW client_view AS
SELECT code_client, count(code_client) AS nombre_achats, sum(montant) AS montant_vendu
FROM quincaillerie
GROUP BY code_client
ORDER BY montant_vendu DESC;
```

```
0: jdbc:hive2://localhost:10000/quincaillerie> select * from client_view;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider
 using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
+--------------------------+----------------------------+----------------------------+
| client_view.code_client  | client_view.nombre_achats  | client_view.montant_vendu  |
+--------------------------+----------------------------+----------------------------+
| CLI3                     | 12                         | 20600                      |
| CLI1                     | 6                          | 5240                       |
| CLI5                     | 2                          | 2400                       |
| CLI4                     | 6                          | 1240                       |
| CLI2                     | 2                          | 250                        |
+--------------------------+----------------------------+----------------------------+
```