

# OPEN FOOD FACTS



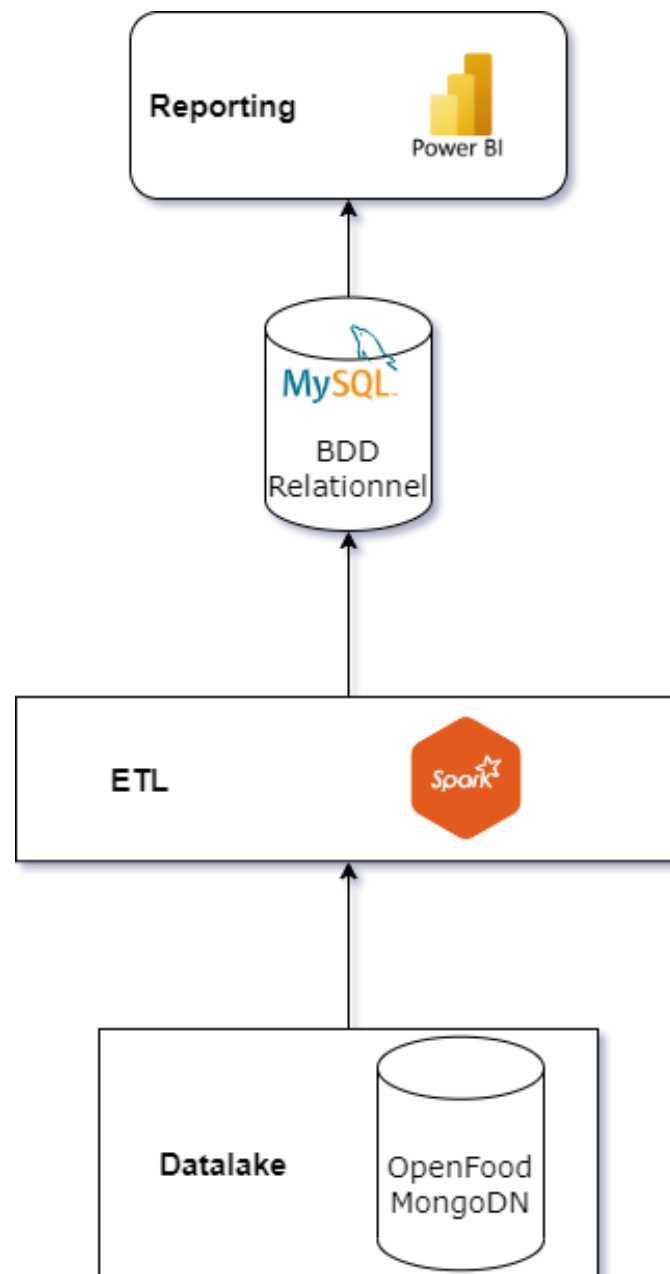
## Groupe :

- AZIZ Cherif
- Tourijan Omar
- Muller Jeremy

**GITHUB :** [https://github.com/AZIZcherif12/Openfood\\_Project](https://github.com/AZIZcherif12/Openfood_Project)

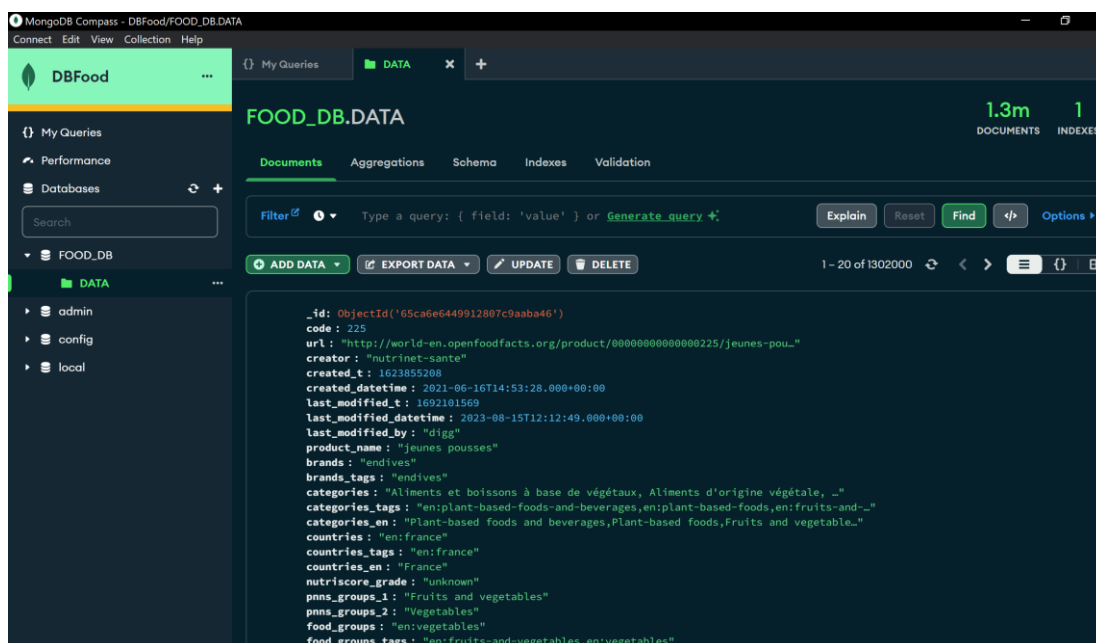
Schéma Project : .....	3
DATALAKE : .....	4
Création de BDD FOOD_DB .....	4
ETL : .....	5
Création nouveau projet : .....	5
Configuration Pom.xml : .....	6
Test affichage donnée.....	7
Filtrer et trie les données .....	7
Installer & configuration MySQL .....	10

## Schéma Project :



## DATALAKE :

L'installation de **MongoDB** sous Windows, suivie de sa configuration incluant la création de la base de données 'FOOD\_DB' et l'importation de la source de données 'openfood\_csv' dans le répertoire 'DATA', ont été réalisées avec succès.



## Création de BDD FOOD\_DB :

Importer le fichier csv et l'afficher :

```
openfood> use FOOD_DB
switched to db FOOD_DB
FOOD_DB> db.DATA.countDocuments()
1302000
FOOD_DB>
```

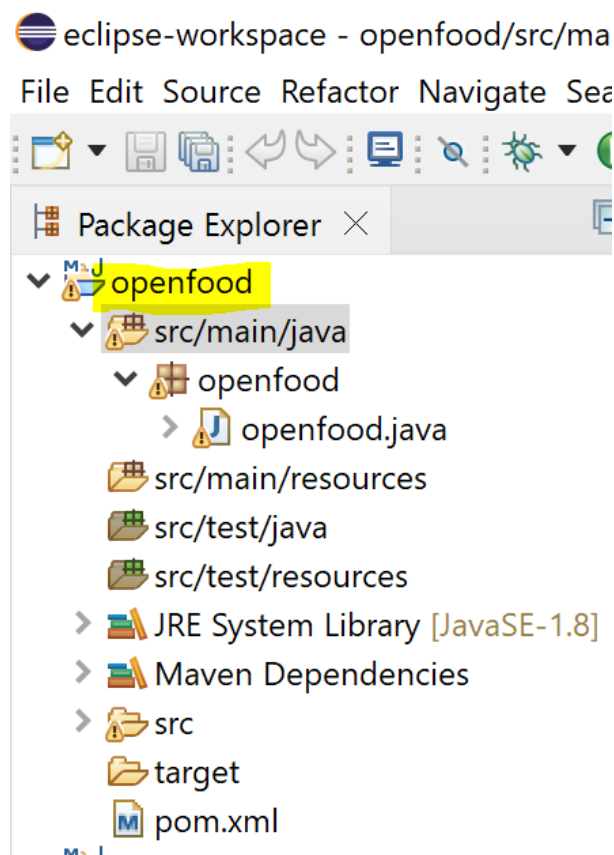
```
mongosh mongodb://127.0.0.1:27017/27017?directConnection=true
2 |
27017> show dbs
FOOD_DB 1.01 GiB
admin 40.00 KiB
config 108.00 KiB
local 40.00 KiB
27017> use FOOD_DB
switched to db FOOD_DB
FOOD_DB>
brands: 'endives',
brands_tags: 'endives',
categories: 'Aliments et boissons à base de végétaux, Aliments d\'origine végétale, ...',
categories_tags: 'en:plant-based-foods-and-beverages,en:plant-based-foods,en:fruits-and-vegetables',
categories_en: 'Plant-based foods and beverages,Plant-based foods,Fruits and vegetables',
countries: 'en:france',
countries_tags: 'en:france',
countries_en: 'France',
nutriscore_grade: 'unknown',
pnns_groups_1: 'Fruits and vegetables',
pnns_groups_2: 'Vegetables',
food_groups: 'en:vegetables',
food_groups_tags: 'en:fruits-and-vegetables,en:vegetables',
food_groups_en: 'Fruits and vegetables,Vegetables'
```

## ETL :

La configuration d'Apache Spark dans Eclipse a été achevée en ajoutant l'indépendance nécessaire pour les bases de données MongoDB et MySQL dans le fichier pom.xml. Nous avons ensuite procédé au tri des données conformément aux exigences du cahier des charges.

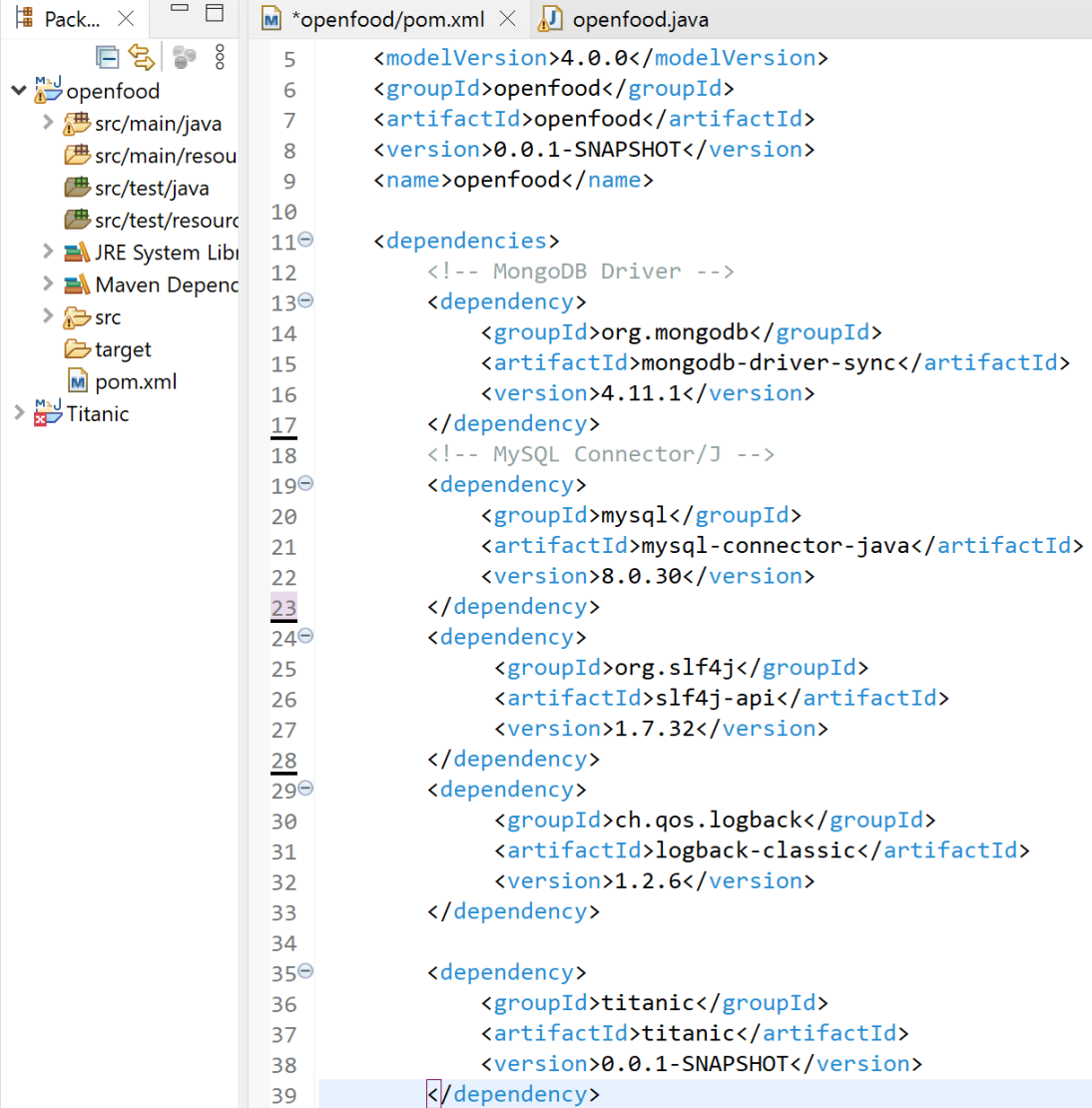
## Création nouveau projet :

Un nouveau projet nommé '**openfood.java**' a été créé dans Eclipse pour débiter le développement.



## Configuration Pom.xml :

Nous avons inclus les dépendances nécessaires pour MongoDB et MySQL dans le fichier pom.xml



The screenshot shows an IDE with a project named 'openfood'. The left sidebar displays the project structure, including 'src/main/java', 'src/main/resou', 'src/test/java', 'src/test/resourc', 'JRE System Libr', 'Maven Depend', 'src', 'target', 'pom.xml', and 'Titanic'. The main editor area shows the 'pom.xml' file with the following content:

```
5 <modelVersion>4.0.0</modelVersion>
6 <groupId>openfood</groupId>
7 <artifactId>openfood</artifactId>
8 <version>0.0.1-SNAPSHOT</version>
9 <name>openfood</name>
10
11 <dependencies>
12   <!-- MongoDB Driver -->
13   <dependency>
14     <groupId>org.mongodb</groupId>
15     <artifactId>mongodb-driver-sync</artifactId>
16     <version>4.11.1</version>
17   </dependency>
18   <!-- MySQL Connector/J -->
19   <dependency>
20     <groupId>mysql</groupId>
21     <artifactId>mysql-connector-java</artifactId>
22     <version>8.0.30</version>
23   </dependency>
24   <dependency>
25     <groupId>org.slf4j</groupId>
26     <artifactId>slf4j-api</artifactId>
27     <version>1.7.32</version>
28   </dependency>
29   <dependency>
30     <groupId>ch.qos.logback</groupId>
31     <artifactId>logback-classic</artifactId>
32     <version>1.2.6</version>
33   </dependency>
34
35   <dependency>
36     <groupId>titanic</groupId>
37     <artifactId>titanic</artifactId>
38     <version>0.0.1-SNAPSHOT</version>
39   </dependency>
```

## Test affichage donnée :

J'ai effectué des tests d'affichage des données via Apache Spark afin de vérifier la récupération des données dans MongoDB, assurant ainsi l'intégrité du processus.

The screenshot shows an IDE with a project named 'openfood'. The Package Explorer on the left shows the project structure. The main editor displays the 'openfood.java' file with the following code:

```

1 package openfood;
2
3 import com.mongodb.client.*;
4 import org.bson.Document;
5
6 public class openfood {
7
8     public static void main(String[] args) {
9         // URL de connexion à MongoDB
10        String uri = "mongodb://127.0.0.1/";
11
12        // Nom de la base de données et de la collection
13        String dbName = "FOOD_DB";
14        String collectionName = "DATA";
15
16        // Connexion à MongoDB
17        try (MongoClient mongoClient = MongoClient.create(uri)) {
18            // Récupération de la base de données et de la collection

```

The Console window at the bottom shows the output of the application, listing various product names:

```

<terminated> openfood [Java Application] C:\Users\Cherif\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.9.v20231028-0858\jre\bin\javaw.exe (1
févr. 13, 2024 8:25:34 PM com.mongodb.internal.diagnostics.logging.Loggers shouldUseSLF4J
AVERTISSEMENT: SLF4J not found on the classpath. Logging is disabled for the 'org.mongodb.driver' component
Product Name: jeunes pousses
Product Name: André
Product Name: L.casei
Product Name: Skyr
Product Name: Vitória crackers
Product Name: Cacao
Product Name: Filetes de pollo empanado
Product Name: Burger Meat Pollo
Product Name: Hamburguesas de ternera 100%
Product Name: Blueberry Cobbler Coffee
Product Name: hyde icon
Product Name: moutarde au mout de raisin
Product Name: Sauce Sweet chili 0%

```

## Filtrer et trie les données :

Le code se connecte à une base de données **MySQL** et à une base de données **MongoDB** localement à l'aide des URL et des informations d'identification fournies. Ensuite, il récupère les données de la collection "**DATA**" dans la base de données MongoDB "**FOOD\_DB**" et les insère dans une table appelée "**products**" dans la base de données MySQL "**openfood**".

Pour chaque **document récupéré** depuis la collection MongoDB, le code extrait les informations telles que **le nom du produit, la quantité, les nutriments, la valeur énergétique, les marques, les ingrédients et les valeurs nutritionnelles pour 100g**. Ces informations sont ensuite utilisées pour construire une requête SQL d'insertion qui est exécutée pour **insérer les données dans la table MySQL**.

Le code fournit également des méthodes utilitaires pour gérer les valeurs **nulles ou incorrectes** dans les documents MongoDB, telles que la conversion des valeurs de type objet en chaînes de caractères ou en nombres à virgule flottante, assurant ainsi l'intégrité des données insérées dans la base de données **MySQL**.

Le **code source** est disponible dans le référentiel Git suivant :

<https://github.com/AZIZcherif12/Openfood Project>

```

1 package openfood;
2
3 import java.sql.*;
4
5 public class openfood {
6
7     public static void main(String[] args) throws SQLException {
8         String mysqlUrl = "jdbc:mysql://localhost:3306/openfood";
9         String mysqlUser = "openfood";
10        String uri = "mongodb://127.0.0.1/";
11        try (Connection conn = DriverManager.getConnection(mysqlUrl, mysqlUser, "password")) {
12            try (MongoClient mongoClient = MongoClient.create(uri)) {
13                MongoDB database = mongoClient.getDatabase("FOOD_DB");
14                MongoCollection<Document> collection = database.getCollection("DATA");
15                try (Statement stmt = conn.createStatement()) {
16                    MongoCursor<Document> cursor = collection.find().iterator();
17                    while (cursor.hasNext()) {
18                        Document doc = cursor.next();
19                        Object productName = doc.get("product_name");
20                        String productNameString = null;
21                        if (productName != null) {
22                            productNameString = productName.toString();
23                        }
24                        Object quantity = doc.get("quantity");
25                        String quantityString = null;
26                        if (quantity != null) {
27                            quantityString = quantity.toString();
28                        }
29                        Document nutriments = (Document) doc.get("nutriments");
30                        if (nutriments != null) {
31                            String energyValueString = getStringValue(nutriments.get("energy_value"));
32                        }
33                        String sqlQuery = "INSERT INTO products (product_name, quantity) VALUES (?, ?)";
34                        try (PreparedStatement statement = conn.prepareStatement(sqlQuery)) {
35                            statement.setString(1, productNameString);
36                            //statement.setString(2, energyValueString);
37                            statement.setString(2, quantityString);
38                        }
39                    }
40                }
41            }
42        }
43    }
44 }

```

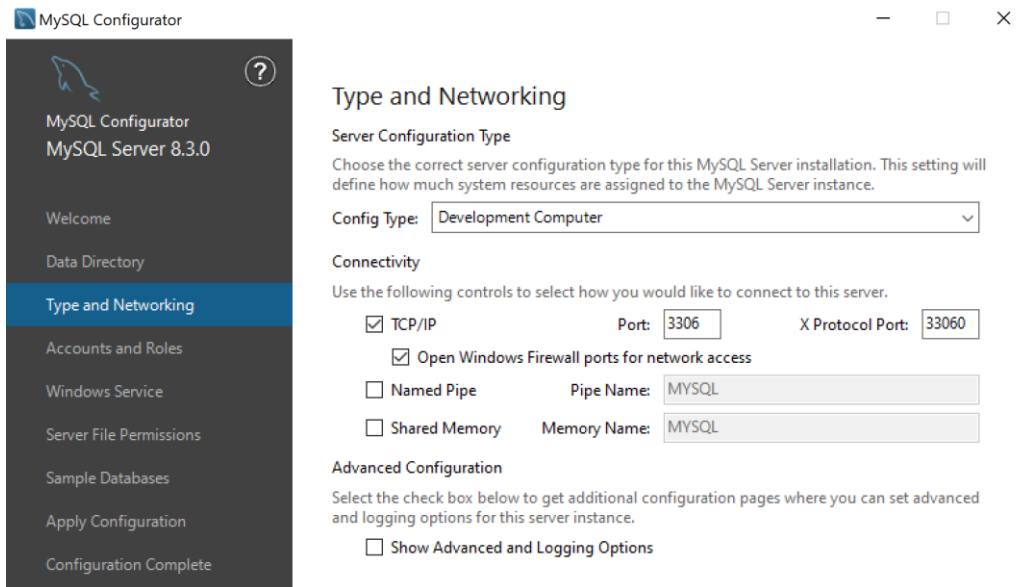


Exécution du code :

[illegible]

## Installer & configuration MySQL :

L'installation et la configuration de MySQL ont été effectuées avec succès, incluant la création d'une base de données nommée '**openfood**' et d'une table '**products**' destinée à recevoir les données provenant d'**Apache Spark**.



### Création DB : « openfood », table et l'utilisateur

```
mysql> CREATE USER 'openfood'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.05 sec)

mysql> GRANT ALL PRIVILEGES ON openfood.* TO 'openfood'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.06 sec)

mysql> SHOW GRANTS FOR 'openfood'@'localhost';
+-----+
| Grants for openfood@localhost |
+-----+
| GRANT USAGE ON *.* TO `openfood`@`localhost` |
| GRANT ALL PRIVILEGES ON `openfood`.* TO `openfood`@`localhost` |
+-----+
2 rows in set (0.04 sec)

mysql>
```

```
mysql> CREATE DATABASE openfood;
Query OK, 1 row affected (0.04 sec)

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| openfood |
| performance_schema |
| sys |
+-----+
5 rows in set (0.06 sec)

mysql>
```

Ajouter des colonnes dans la table :

```
mysql> DESCRIBE products;
+-----+-----+-----+-----+-----+-----+
| Field                | Type                | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| product_name         | varchar(255)        | YES  |     | NULL    |       |
| quantity             | varchar(255)        | YES  |     | NULL    |       |
| nutriments           | varchar(255)        | YES  |     | NULL    |       |
| countries_tags       | varchar(255)        | YES  |     | NULL    |       |
| carbohydrates_100g   | decimal(10,2)       | YES  |     | NULL    |       |
| sugars_100g          | decimal(10,2)       | YES  |     | NULL    |       |
| proteins_100g        | decimal(10,2)       | YES  |     | NULL    |       |
| salt_100g            | decimal(10,2)       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql>
```

Affichage donnée dans reçu dans la table :

```
mysql> select * from products;
+-----+-----+-----+-----+-----+
| product_name         | quantity            | nutriments | energyValue | energy_value |
+-----+-----+-----+-----+-----+
| jeunes pousses      | NULL                | NULL       | NULL        | NULL         |
| André               | NULL                | NULL       | NULL        | NULL         |
| L.casei              | NULL                | NULL       | NULL        | NULL         |
| Skyr                 | 480 g               | NULL       | NULL        | NULL         |
| Vitória crackers    | NULL                | NULL       | NULL        | NULL         |
| Cacao                | 130 g               | NULL       | NULL        | NULL         |
| Filetes de pollo empanado | NULL                | NULL       | NULL        | NULL         |
| Burger Meat Pollo   | NULL                | NULL       | NULL        | NULL         |
| Hamburguesas de ternera 100% | NULL                | NULL       | NULL        | NULL         |
| Blueberry Cobbler Coffee | NULL                | NULL       | NULL        | NULL         |
| hyde icon            | NULL                | NULL       | NULL        | NULL         |
| moutarde au moût de raisin | 100g                | NULL       | NULL        | NULL         |
| Sauce Sweet chili 0% | NULL                | NULL       | NULL        | NULL         |
| Mini coco            | NULL                | NULL       | NULL        | NULL         |
| Pistou d'ail des ours | NULL                | NULL       | NULL        | NULL         |
| Thé noir glacé pomme | NULL                | NULL       | NULL        | NULL         |
| Pain maïs            | NULL                | NULL       | NULL        | NULL         |
| Mendiants            | NULL                | NULL       | NULL        | NULL         |
| Limonade artisanale a la rose | NULL                | NULL       | NULL        | NULL         |
| Tablette Tanzanie    | NULL                | NULL       | NULL        | NULL         |
| Sachet calisson      | NULL                | NULL       | NULL        | NULL         |
| Pur jus de pomme     | NULL                | NULL       | NULL        | NULL         |
| Pate d'amande        | NULL                | NULL       | NULL        | NULL         |
| Chocolats            | NULL                | NULL       | NULL        | NULL         |
| Salade de carottes râpées | NULL                | NULL       | NULL        | NULL         |
| Fromage blanc aux myrtilles | NULL                | NULL       | NULL        | NULL         |
| NULL                 | NULL                | NULL       | NULL        | NULL         |
| Vainilla             | NULL                | NULL       | NULL        | NULL         |
| Iogurt de cabra      | NULL                | NULL       | NULL        | NULL         |
| Baguette parisien    | NULL                | NULL       | NULL        | NULL         |
| Chocolate n 3        | 80 g                | NULL       | NULL        | NULL         |
| "Baguette Lyonnais" | NULL                | NULL       | NULL        | NULL         |
| Solène céréales poulet | NULL                | NULL       | NULL        | NULL         |
| Tarte noix de coco   | NULL                | NULL       | NULL        | NULL         |
| Caramel au beurre salé | NULL                | NULL       | NULL        | NULL         |
| Salade de fruits exotiques | NULL                | NULL       | NULL        | NULL         |
| Chouquettes x 30     | NULL                | NULL       | NULL        | NULL         |
| Pain de maïs bio     | NULL                | NULL       | NULL        | NULL         |
| Tisane Cassis        | NULL                | NULL       | NULL        | NULL         |
| Fromage blanc pêche  | NULL                | NULL       | NULL        | NULL         |
| Crème dessert chocolat | NULL                | NULL       | NULL        | NULL         |
| Bouillie de sarrasin | NULL                | NULL       | NULL        | NULL         |
| Yaourt au chocolat   | NULL                | NULL       | NULL        | NULL         |
| Sablé aux pépites de chocolat | 125 g              | NULL       | NULL        | NULL         |
```