Big Data - MongoDB Assignment

Student: Chien-Wei WENG
Program: Data Sciences and Business Analytics M2
Date: November 2025
Data Source: Open Food Facts sample dataset (https://world.openfoodfacts.org/)

1.1 Exercise: Understanding Data

Methodology

I explored the collection by:

1. Examined sample documents using `db.products.limit(10)`
2. Checked field presence with `{$exists: True}` query
3. Reviewed the official documentation (data-field.txt)
4. Calculated field presence rates for key attributes

Most Common Document Structure

- Definition of common attributes (>50% presence rates)
- Analyzed 27 keys fields from metadata

General Information Fields

`code` / `_id` : product barcode or identifier
`creator` : contributor who first added the product
`creator_t` : date that the product was added
`last_modified_t` : date that the product was last modified
`product_name` : name of the product

Tags Fields

`packaging` : material or type of packaging
`brands` : brand of the product
`countries` : country of the product

Ingredients Fields

`ingredients_text` : full ingredients list

`traces` : traces ingredients (e.g. eggs, milk)

Field Presence Analysis:

```
// example: product_name = 94.94%
var total = db.products.count()
var product_name = db.products.find({"product_name": {$exists: true}}).count()
print(product_name/total*100 + "%")
```

Nested Documents

- `nutriments` : object containing nutritional values (energy_100g, proteins_100g, etc.)
- Fields ending with `_tags` are typically arrays (not nested documents), e.g., `brands_tags` , `categories_tags`

Data Observations

- Many optional fields contain `null` values
- Many fields contain empty string or empty arrays (not null)
- `$exists: true` only checks if field name exists, not if it has meaningful data

1.2 Exercise: Querying Data

Query 1: Number of products

Query

```
db.products.count()
```

Result

```
4802
```

Sample

```
db.products.findOne({}, {product_name: 1})
```

```
{
    "_id" : "00000000000",
    "product_name" : "Mediterranean Veggie Sandwich"
}
```


Explanation: Count the number of documents in the collection.

Query 2: Products contains specific product name

Query

```
db.products.find({product_name: "Sharon's, sorbet, dutch chocolate"}, {product_name:1})
```

Result/Sample


```
{
    "_id" : "0009073102079",
    "product_name" : "Sharon's, sorbet, dutch chocolate"
}
```


Explanation: Find the specific product name.

Query 3: How many times a product has been modified

Query

```
db.products.find({_id: "0009073102079"}, {rev:1, product_name:1})
```

Result

```
2
```

Sample

```
{
    "_id" : "0009073102079",
    "rev" : NumberInt(2),
    "product_name" : "Sharon's, sorbet, dutch chocolate"
}
```

Explanation: The `rev` field indicates the number of revisions. This product has been modified 2 times.

Query 4: Products have sodium in the nutriments list

Query

```
db.products.find({"nutriments.sodium": {$exists: true}}).count()
```

Result

```
3425
```

Sample

```
db.products.findOne({"nutriments.sodium": {$exists: true}}, {"nutriments.sodium":1})
```

```
{
    "_id" : "00000000000",
    "nutriments" : {
        "sodium" : 1.26
    }
}
```

Explanation: Count the products that have sodium in embedded documents.

Query 5: Products with nutriscore_grade equal to 'c'

Query

```
db.products.find({nutriscore_grade: "c"}).count()
```

Result

```
287
```

Sample

```
db.products.findOne({nutriscore_grade: "c"}, {nutriscore_grade:1})
```

```
{
    "_id" : "00",
    "nutriscore_grade" : "c"
}
```

Explanation: 287 products with `nutriscore_grade` equal to `c` .

Query 6: Number of different creators

Query

```
db.products.distinct("creator").length
```

Result

```
131
```

Sample

```
db.products.findOne({creator: {$exists: true}}, {creator:1})
```

```
{
    "_id" : "00000000000",
    "creator" : "omnomnotes-app"
}
```

Explanation: 131 creators have contributed to the product creation.

Query 7: Creators have created more than one product

Query

```
db.products.aggregate([
    {$group: {_id: "$creator", count: {$sum: 1}}},
    {$match: {count: {$gt: 1}}},
    {$count: "total"}
])
```

Result

```
51
```

Sample

```
db.products.aggregate([
    {$group: {_id: "$creator", count: {$sum:1}}},
    {$match: {count: {$gt:1}}},
    {$sort: {count:-1}},
    {$limit:1}
])
```


```
{
    "_id" : {
        "creator" : "kiliweb"
    },
    "count" : 2156.0
}
```


Explanation: Aggregation groups by creator, filters for count > 1, then counts the results. Creator "kiliweb" has the most products with 2,156.

Query 8: The most recently modified product

Query

```
db.products.find({last_modified_t: {$exists: true}}, {last_modified_t:1})
           .sort({last_modified_t:-1})
           .limit(1)
```

Result/Sample


```
{
    "_id" : "000123456789",
    "last_modified_t" : NumberInt(1636317917)
}
```


Query 9: Products have exactly one ingredient

Query

```
db.products.find({ingredients_tags: {$size:1}}, {ingredients_tags:1}).count()
```

Result

```
106
```

Sample

```
db.products.findOne({ingredients_tags: {$size:1}}, {ingredients_tags:1})
```


```
{
    "_id" : "000",
    "ingredients_tags" : [
        "it:semi-di-canapa-sgusciati-bio-dall-europa"
    ]
}
```


Explanation: 106 products have exactly one ingredient.

Query 10: Products that have 20 or more ingredients

Query

```
db.products.find(
    {
     ingredients_tags: {$exists: true},
     $expr: {$gte: [{$size: "$ingredients_tags"}, 20]}
     },
    {ingredients_tags:1}
).count()
```

Result

```
1118
```

Sample

```
db.products.findOne(
    {
     ingredients_tags: {$exists: true},
     $expr: {$gte: [{$size: "$ingredients_tags"}, 20]}
     },
    {ingredients_tags:1}
)
```


```
{
    "_id" : "00000000000000000000",
    "ingredients_tags" : [
        "en:rapeseed-oil",
        "en:oil-and-fat",
        "en:vegetable-oil-and-fat",
        "en:water",
        "en:milk-powder",
        "en:dairy",
        "en:sugar",
        .........
}
```


Explanation: 1118 products have 20 or more ingredients

Query 11: Number of products that are characterized as desserts

Query

```
db.products.find({_keywords: {$in: ["dessert"]}}, {_keywords:1}).count()
```

Result

```
47
```

Sample

```
db.products.findOne({_keywords: {$in: ["dessert"]}}, {_keywords:1})
```


```
{
    "_id" : "0000000001663",
    "_keywords" : [
        "fremondiere",
        "dessert",
        "ferme",
        "chocolat",
        "la",
        "creme",
        "de"
    ]
}
```


Explanation: 47 products are characterized as desserts in `_keywords`.

Query 12: Number of products that are characterized as chocolate

Query

```
db.products.find({_keywords: {$in: ["chocolate"]}}, {_keywords:1}).count()
```

Result

```
536
```

Sample

```
db.products.findOne({_keywords: {$in: ["chocolate"]}}, {_keywords:1})
```


```
{
    "_id" : "000000000114",
    "_keywords" : [
        "de",
        "jeff",
        "chocolate",
        "bruge",
        "made-in-france"
    ]
}
```


Explanation: 536 products are characterized as chocolate in `_keywords` .

Query 13: Number of products that are characterized as chocolate and dessert

Query

```
db.products.find({_keywords: {$all: ["chocolate", "dessert"]}}, {_keywords:1}).count()
```

Result

```
3
```

Sample

```
db.products.findOne({_keywords: {$all: ["chocolate", "dessert"]}}, {_keywords:1})
```

```
{
    "_id" : "00033817",
    "_keywords" : [
        "crou",
        "languedoc",
        "roussillon",
        "dessert",
        "chocolate",
        "chocolat",
        "mousse",
        "au",
        "resto"
    ]
}
```


Explanation: 3 products are characterized as chocolate **and** dessert in `_keywords` .

Query 14: Number of products that are characterized as chocolate or dessert

Query

```
db.products.find({_keywords: {$in: ["chocolate", "dessert"]}}, {_keywords:1}).count()
```

Result

```
580
```

Sample

```
db.products.findOne({_keywords: {$in: ["chocolate", "dessert"]}}, {_keywords:1})
```

```
{
    "_id" : "000000000114",
    "_keywords" : [
        "de",
        "jeff",
        "chocolate",
        "bruge",
        "made-in-france"
    ]
}
```


Explanation: 580 products are characterized as chocolate **or** dessert in `_keywords` .

Query 15: Convert string type to array and create a new field

Query

```
db.products.updateMany(
       {categories: {$type: "string"}},
       [{
           $set: {
               new_att_category: {$split: ["$categories", ","]}
           }
       }]
)
```

Result


```
{
    "acknowledged" : true,
    "insertedId" : null,
    "matchedCount" : 2340.0,
    "modifiedCount" : 0.0,
    "upsertedCount" : 0.0
}
```


Sample

```
db.products.findOne({new_att_category: {$exists: true}}, {new_att_category:1})
```

```
{
    "_id" : "0",
    "new_att_category" : [
        "en:beverages"
    ]
}
```


Explanation: Converted the `categories` string field to an array using `$split` by comma delimiter, creating a new field `new_att_category`. Operation matched 2,340 documents with string-type categories.

Query 16: Products classed as F in `nutriscore_grade` and have palm-oil in ingredients

Query

```
db.products.distinct("nutriscore_grade")
```

Result


```
[
    "a",
    "b",
    "c",
    "d",
    "e"
]
```


Sample

```
db.products.find({nutriscore_grade: "f"}, {nutriscore_grade:1}).count()
db.products.find({ingredients_tags: "en:palm-oil"}, {ingredients_tags:1}).count()
```

```
0
217
```

Explanation: There are no 'F' score in `nutriscore_grade`.