



Unidad Nacional Experimental de Guayana.
Vicerrectorado Académico.
Coordinación General de Pregrado.
Proyecto de carrera: Ingeniería en Informática.
Asignatura: Sistemas de bases de datos II – Sección: 1.

Bases de Datos Documentales con MongoDB

Profesor:
Clinia Cordero

Integrantes:
V-25.933.680 Alburquerque Sheen
V-31.818.222 Valencia Haddan
V-31.445.710 Longart Daniela

Ciudad Guayana, 05 Diciembre del 2025

Índice.

Introducción	3
Objetivos.....	4
Descripción del problema	5
Requerimientos del sistema.....	6
Modelado y Validación.....	7
Desarrollo	9
1. Configuración del entorno en MongoDB Atlas	9
2. Conexión del cliente MongoDB Compass	11
3.Implementación del Schema Validation.....	13
4. Aplicar validación en sales_transformed	15
Datos De Prueba Para Validación	16
5. Aplicación del Pipeline de Transformación en Sales	19
6. Pipeline 1: Reporte De Ventas Por Categoría Y Mes	20
7. Pipeline 2: completo y auto-contenido (genera rating aleatorio y filtra)....	22
8. Pipeline 3: Bucket por rangos de precios unitarios	23
ÍNDICE 1: Compuesto (categoría + fecha)	25
ÍNDICE 2: Simple (product_id)	26
ÍNDICE 3: Compuesto (sales + quantity).....	26
Observación	28
Explain plan	31
Visualización	34
1. Ventas por país (gráfico de barras – arriba izquierda).....	34
2. Cantidad por ganancias (gráfico de rosquilla – arriba centro).....	34
3. Productos vendidos (gráfico horizontal – arriba derecha).....	35
4. Ventas Totales (parte inferior izquierda).....	35
5. Productos por fecha — Productos más pedidos por año (abajo centro).....	35
6. Producto por país – Mapa mundial (abajo derecha)	36
Anexo	36
Recomendaciones generales	37
Conclusión.....	38

Introducción

GlobalMarket, es una startup de comercio electrónico en expansión. Su base de datos relacional actual está fallando bajo la carga de consultas complejas y la variabilidad de los atributos de sus productos. Nuestra función es migrar y optimizar el catálogo de productos y el registro de ventas a MongoDB Atlas, implementando un motor de búsqueda eficiente y un dashboard de análisis de ventas en tiempo real.

Este proyecto consiste en la creación, gestión y consulta de una base de datos NoSQL utilizando MongoDB Atlas y el shell de comandos. Se trabajará con colecciones, documentos, consultas simples y avanzadas, operadores lógicos y actualizaciones, con el objetivo de comprender la estructura flexible de datos de MongoDB.

Objetivos

Objetivo General

Crear y gestionar una base de datos NoSQL mediante comandos en MongoDB Shell utilizando MongoDB Atlas como servidor remoto.

Objetivos Específicos

- Conectar un cliente local a una base de datos en la nube.
- Crear una base de datos y colecciones.
- Insertar documentos utilizando formato JSON.
- Aplicar consultas simples y avanzadas con operadores.
- Actualizar, eliminar y filtrar documentos.
- Evaluar el funcionamiento del CRUD completo.

Descripción del problema

GlobalMarket, una empresa dedicada a la comercialización de productos de diferentes categorías, gestiona información clave relacionada con inventarios, precios, disponibilidad de productos y características comerciales. Hasta el momento, estos datos han sido almacenados en un sistema tradicional que presenta limitaciones al momento de realizar consultas avanzadas, aplicar cambios estructurales o integrar nueva información de forma dinámica.

Debido al crecimiento de la empresa y a la necesidad de realizar análisis más completos y deducciones automáticas sobre los datos almacenados, se plantea una migración de la base de datos actual hacia un sistema NoSQL basado en MongoDB.

MongoDB permite trabajar con una base de datos deductiva, capaz de almacenar y procesar datos de forma flexible mediante documentos no estructurados, lo cual facilita la interpretación de la información, el establecimiento de relaciones implícitas y la adaptación a nuevos requerimientos sin afectar la continuidad del sistema. Asimismo, su implementación en la nube mediante MongoDB Atlas garantiza accesibilidad remota, disponibilidad continua y una gestión centralizada más eficiente.

La falta de una solución moderna que soporte consultas más inteligentes y flexibles limita actualmente el aprovechamiento de los datos empresariales. Por ello, la migración hacia MongoDB busca resolver estas dificultades y mejorar la toma de decisiones dentro de GlobalMarket mediante un sistema de datos más avanzado y orientado al análisis.

Requerimientos del sistema

Software

- MongoDB Atlas
- MongoDB Compass
- Navegador web
- Editor de texto (opcional)

Hardware

- Computador con Windows
- Conexión a Internet

Modelado y Validación

Diagrama relacional:

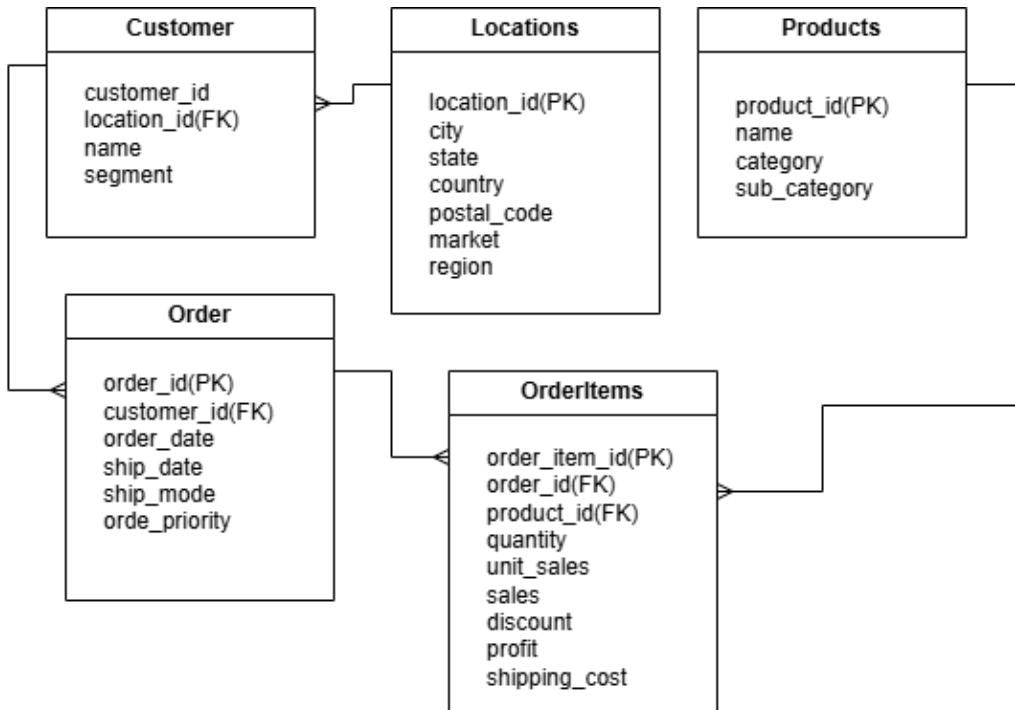
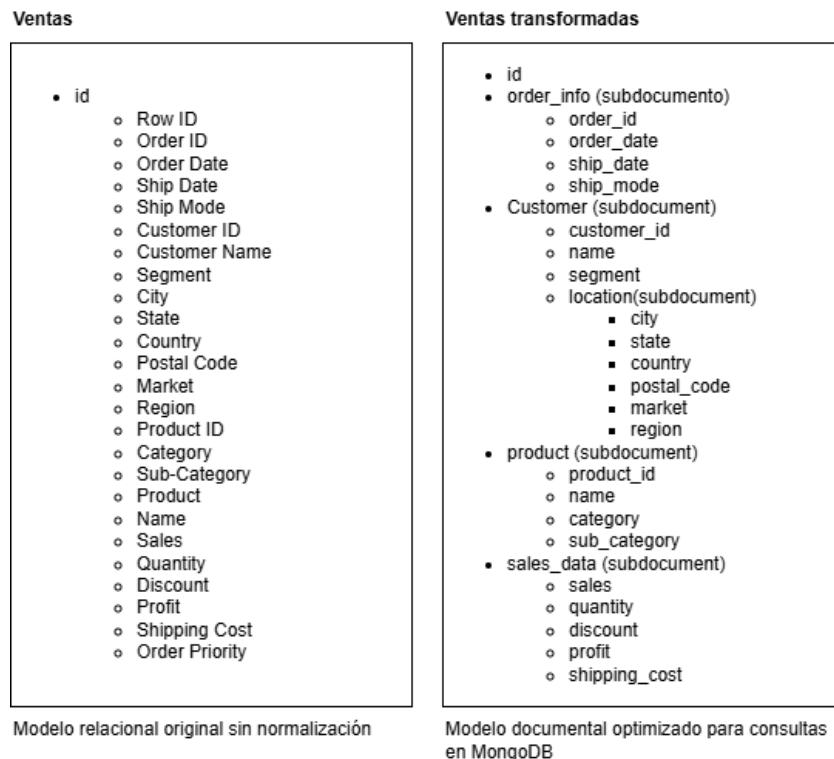


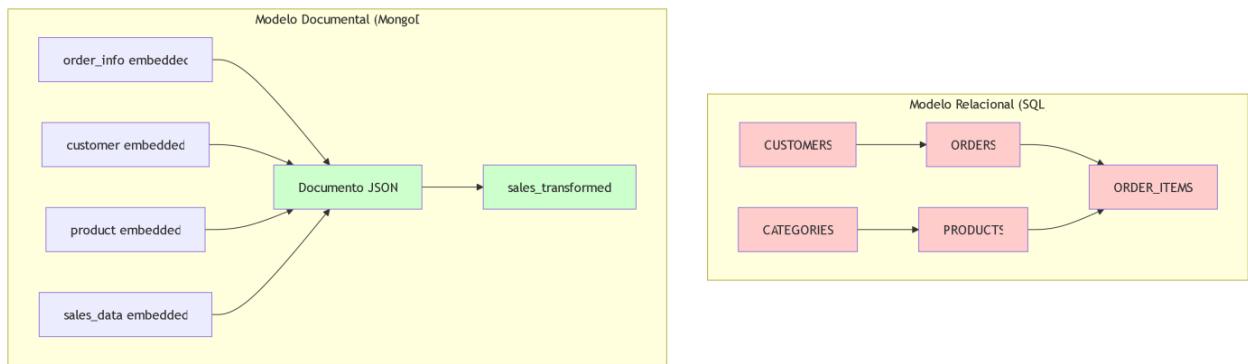
Diagrama documental:



La base de datos relacional garantiza integridad y normalización, pero implica un mayor costo en consultas analíticas debido a la fragmentación de la información en múltiples tablas.

La base de datos documental desarrollada en MongoDB permite un acceso más ágil a información consolidada por pedido, cliente y producto, lo cual beneficia directamente los análisis operativos y estratégicos del negocio GlobalMarket.

Por esa razón, la migración a un modelo NoSQL documental representa una mejora clara en eficiencia, escalabilidad y velocidad para el manejo de grandes volúmenes de datos comerciales.



Desarrollo

1. Configuración del entorno en MongoDB Atlas

a. Crear el GlobalMarket-Cluster en Mongo DB atlas

The screenshot shows the first step of a three-step wizard titled "Connect to GlobalMarket-Cluster". Step 1, "Set up connection security", is completed with a green checkmark. Step 2, "Choose a connection method", is the current step, indicated by a green outline around its number. Step 3, "Connect", is shown with a grey outline. A note at the top says: "You need to secure your MongoDB Atlas cluster before you can use it. Set which users and IP addresses can access your cluster now. [Read more](#)". Below the note, two sections are listed:

- Add a connection IP address**: A note states: "Your current IP address [45.186.208.72] has been added to enable local connectivity. Only an IP address you add to your Access List will be able to connect to your project's clusters. Add more later in [Network Access](#)".
- Create a database user**: A note states: "A database user has been added to this project. Create another user later in [Database Access](#). You'll need your database user's credentials in the next step."

Connect to your application

Access your data through tools

Close **Choose a connection method** **x**

b. Copiar el String

Connect to GlobalMarket-Cluster

Setup connection security Choose a connection method Connect

Connecting with MongoDB Compass

I don't have MongoDB Compass installed I have MongoDB Compass installed

1. Select your operating system and download MongoDB Compass

macOS x64 (Intel) (11+)

Download Compass (1.48.2) or Copy download URL

Compass is an interactive tool for querying, optimizing, and analyzing your MongoDB data.

2. Copy the connection string, then open MongoDB Compass

Show Password ⓘ

Use this connection string in your application

`mongodb+srv://sheen1386_db_user:6RMo9H0GRJKetqNX@globalmarket-cluster.3rebz5q.mongodb.net/`

The password for `sheen1386_db_user` is included in the connection string for your first time setup. This password will not be available again after exiting this connect flow.

RESOURCES

Connect with Compass ⓘ Import and Export Data ⓘ
Access your Database Users ⓘ Troubleshoot Connections ⓘ

Go Back Done



c. Pegar la URL en el MongoDB Compass

New Connection

Manage your connection settings

URI ⓘ `mongodb+srv://sheen1386_db_user:*****@globalmarket-cluster.3rebz5q.mongodb.net/` Edit Connection String ⓘ

Name: globalmarket-cluster.3rebz5q.mongodb.net Color: Purple

Favorite this connection

Favoriting a connection will pin it to the top of your list of connections

Advanced Connection Options

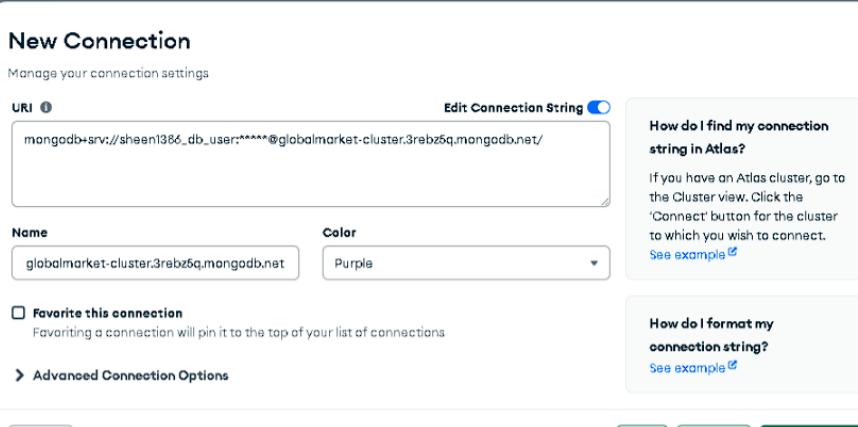
How do I find my connection string in Atlas?

If you have an Atlas cluster, go to the Cluster view. Click the 'Connect' button for the cluster to which you wish to connect. See example ⓘ

How do I format my connection string?

See example ⓘ

Cancel Save Connect Save & Connect



2. Conexión del cliente MongoDB Compass

a. Descarga del archivo CSV

Global Super Store Dataset

Data Card Code (19) Discussion (4) Suggestions (0)

Global_Superstore2.csv (12.09 MB)

Detail Compact Column

Row ID Order ID Order Date Ship Date Ship Mode Customer

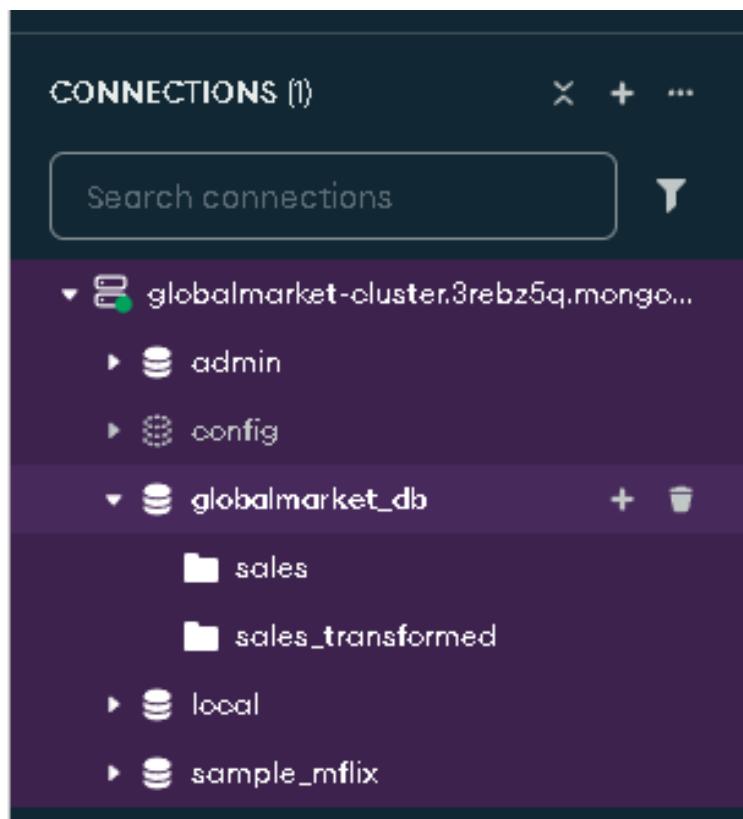
Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer
1	CA-2014-135999	14-10-2014	21-10-2014	Standard Class	JW-15228
48155	CA-2012-116638	28-01-2012	31-01-2012	Second Class	JH-15985
48936	CA-2011-182988	05-04-2011	09-04-2011	Second Class	GM-14695
34577	ID-2012-28482	19-04-2012	22-04-2012	First Class	AJ-18788
28879	SA-2011-1838	27-12-2011	29-12-2011	Second Class	MM-7268
45794	MX-2012-138815	13-11-2012	13-11-2012	Same Day	WF-21715
4132	IN-2013-73951	06-06-2013	08-06-2013	Second Class	PF-19128
27784					

Version 2 (20.13 MB)

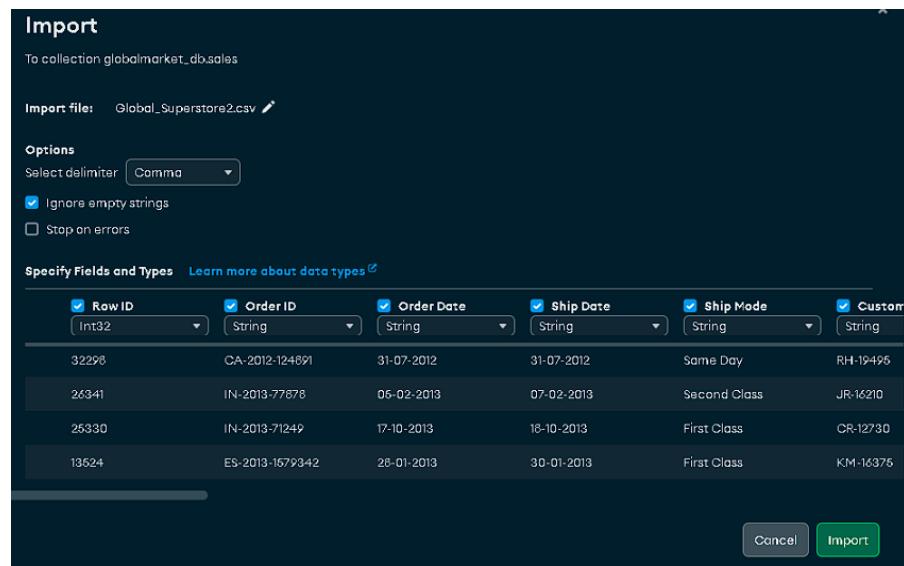
Global_Superstore2.csv
Global_Superstore2.xlsx

Summary 2 files 48 columns

b. Creación de la base de datos globalmarket_dv en el Compass y el nombre de la colección Sales



c. Importamos el archivo (csv)



d. Data cargada

The screenshot shows the MongoDB Compass interface. The left sidebar shows 'Connections' and the current connection is 'globalmarket-cluster.3rebz5q.mongodb.net'. The main area shows the 'sales' collection under 'globalmarket_db'. The document count is 51K. A specific document is selected, showing its fields and values:

```
_id: ObjectId('692d125d209add783aae2dca')
Row ID : 32298
Order ID : "CA-2012-124891"
Order Date : "31-07-2012"
Ship Date : "31-07-2012"
Ship Mode : "Same Day"
Customer ID : "RH-19495"
Customer Name : "Rick Hansen"
Segment : "Consumer"
City : "New York City"
State : "New York"
Country : "United States"
Postal Code : 10024
Market : "US"
Region : "East"
Product ID : "TEC-AC-10000033"
Category : "Technology"
Sub-Category : "Accessories"
Product Name : "Plantronics CS510 - Over-the-Head monaural Wireless Headset System"
Sales : 2389.65
Quantity : 7
Discount : 0
Profit : 762.1845
Shipping Cost : 933.57
Order Priority : "Critical"
```

Below this, another document is partially visible:

```
_id: ObjectId('692d125d209add783aae2dc1')
Row ID : 26341
```

3.Implementación del Schema Validation

a. Validación

```
{  
  "$jsonSchema": {  
    "bsonType": "object",  
    "required": ["order_info", "customer", "product", "sales_data"],  
    "properties": {  
      "sales_data.sales": {  
        "bsonType": "double",  
        "minimum": 0,  
        "description": "Sales must be a positive number"  
      },  
      "sales_data.profit": {  
        "bsonType": "double",  
        "description": "Profit must be a number (can be negative)"  
      },  
      "sales_data.quantity": {  
        "bsonType": "int",  
        "minimum": 1,  
        "description": "Quantity must be positive integer"  
      },  
      "sales_data.discount": {  
        "bsonType": "double",  
        "minimum": 0,  
        "maximum": 1,  
        "description": "Discount must be between 0 and 1"  
      },  
      "customer.customer_id": {  
        "bsonType": "string",  
        "description": "Customer ID is required"  
      },  
      "product.category": {  
        "bsonType": "string",  
        "enum": ["Technology", "Furniture", "Office Supplies"],  
        "description": "Category must be one of the allowed values"  
      }  
    }  
  }  
}
```

The screenshot shows the Compass MongoDB interface. On the left, the 'Connections' sidebar lists several databases, with 'globalmarket_db' selected. In the main area, the 'sales' collection is selected under the 'globalmarket_db' database. The 'Validation' tab is active. A code editor displays the schema validation rule for the 'sales' collection:

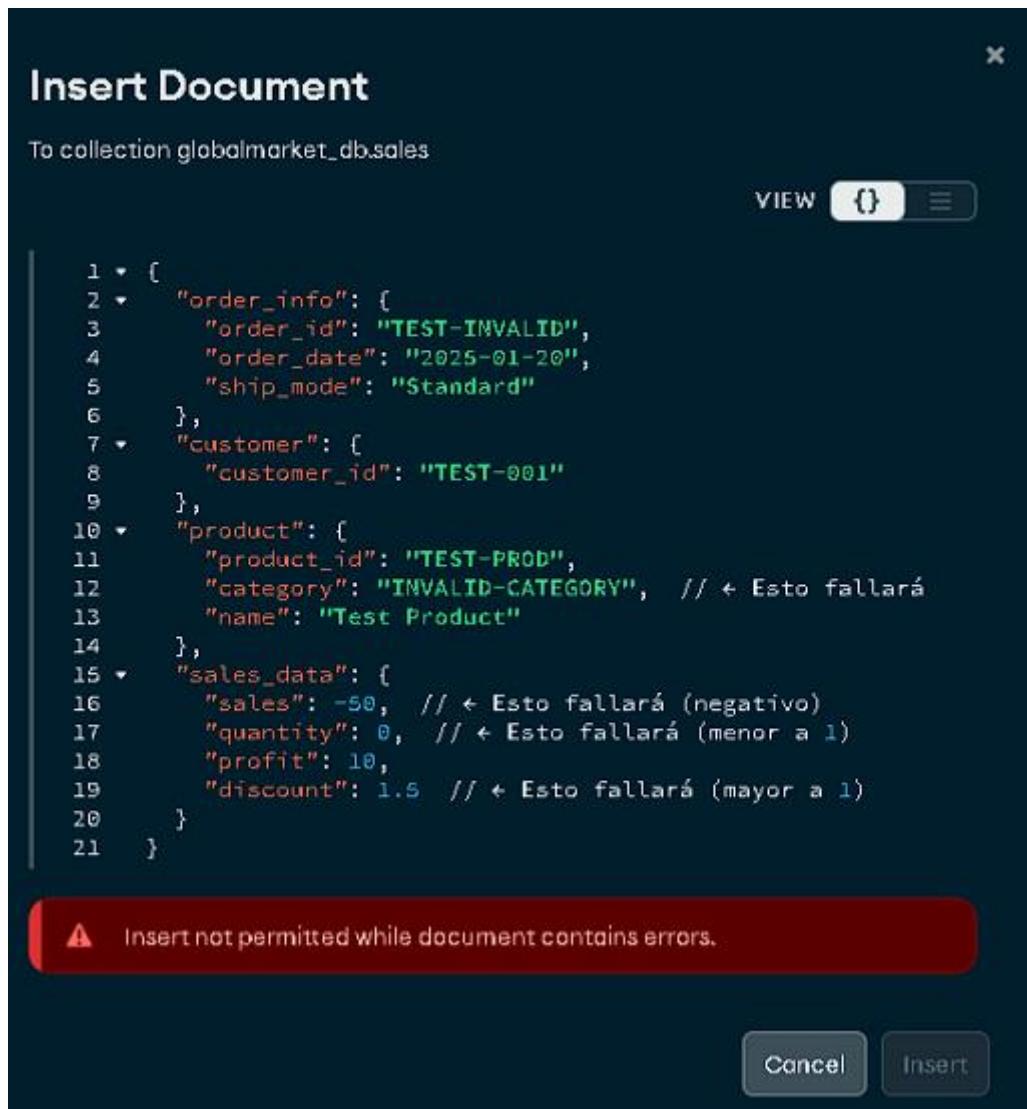
```

1 * /**
2  * This is a starter template for a schema validation rule for a collection.
3  * More information on schema validation rules can be found at:
4  * https://www.mongodb.com/docs/manual/core/schema-validation/
5 */
6 {
7   $jsonSchema: {
8     title: "Library.books",
9     bsonType: "object",
10    required: ["fieldname1", "fieldname2"],
11   properties: {
12     fieldname1: {
13       bsonType: "string",
14       description: "Fieldname1 must be a string",
15     },
16     fieldname2: {
17       bsonType: "int",
18       description: "Fieldname2 must be an integer",
19     },
20     arrayFieldName: {
21       bsonType: "array",
22       items: {
23         bsonType: "string"
24       },
25       description: "arrayFieldName must be an array of strings"
26     },
27   }
28 }
29 
```

Below the code editor, a note says: 'Rules are modified & not applied. Please review before applying.' At the bottom right are 'Cancel' and 'Apply' buttons.

b. Prueba de validación

```
{
  "order_info": {
    "order_id": "TEST-INVALID",
    "order_date": "2025-01-20",
    "ship_mode": "Standard"
  },
  "customer": {
    "customer_id": "TEST-001"
  },
  "product": {
    "product_id": "TEST-PROD",
    "category": "INVALID-CATEGORY", // ← Esto fallará
    "name": "Test Product"
  },
  "sales_data": {
    "sales": -50, // ← Esto fallará (negativo)
    "quantity": 0, // ← Esto fallará (menor a 1)
    "profit": 10,
    "discount": 1.5 // ← Esto fallará (mayor a 1)
  }
}
```



4. Aplicar validación en sales_transformed

a. Validation en sale_trans

```
{
  "$jsonSchema": {
    "bsonType": "object",
    "required": ["order_info", "customer", "product", "sales_data"],
    "properties": {
      "sales_data.sales": {
        "bsonType": "double",
        "minimum": 0,
        "description": "Sales must be a positive number"
      },
      "sales_data.profit": {
        "bsonType": "double",

```

```

    "description": "Profit must be a number (can be negative)"
},
"sales_data.quantity": {
    "bsonType": "int",
    "minimum": 1,
    "description": "Quantity must be positive integer"
},
"sales_data.discount": {
    "bsonType": "double",
    "minimum": 0,
    "maximum": 1,
    "description": "Discount must be between 0 and 1"
},
"customer.customer_id": {
    "bsonType": "string",
    "description": "Customer ID is required"
},
"product.category": {
    "bsonType": "string",
    "enum": ["Technology", "Furniture", "Office Supplies"],
    "description": "Category must be one of the allowed values"
}
}
}
}

```

Datos De Prueba Para Validación

Documento INVÁLIDO (debe fallar):

```

{
    "order_info": {
        "order_id": "TEST-VALIDACION-1",
        "order_date": "20-01-2025"
    },
    "customer": {
        "customer_id": "TEST-CLIENTE"
    },
    "product": {
        "product_id": "TEST-PROD",
        "category": "CATEGORIA-INVALIDA", // ← Error: no está en enum
        "name": "Producto Test"
    },
    "sales_data": {
        "sales": -100, // ← Error: negativo
        "quantity": 0, // ← Error: menor a 1
        "discount": 1.5, // ← Error: mayor a 1
        "profit": 50
    }
}

```

Insert Document

To collection globalmarket_db.sales_transformed

VIEW { } ⋮

```
1 * [
2 *   "order_info": {
3     "order_id": "TEST-VALIDACION-1",
4     "order_date": "20-01-2025"
5   },
6 *   "customer": {
7     "customer_id": "TEST-CLIENTE"
8   },
9 *   "product": {
10    "product_id": "TEST-PROD",
11    "category": "CATEGORIA-INVALIDA", // ← Error: no está
12    "name": "Producto Test"
13  },
14 *   "sales_data": {
15    "sales": -100, // ← Error: negativo
16    "quantity": 0, // ← Error: menor a 1
17    "discount": 1.5, // ← Error: mayor a 1
18    "profit": 50
19  }
20 ]
```

⚠ Insert not permitted while document contains errors.

Cancel **Insert**

Documento VÁLIDO (debe pasar):

```
{
  "order_info": {
    "order_id": "TEST-VALIDO-001",
    "order_date": "20-01-2025",
    "ship_mode": "Standard",
    "order_priority": "High"
  },
  "customer": {
    "customer_id": "CLIENTE-OK",
    "name": "Cliente Prueba",
    "segment": "Consumer",
    "location": {
      "city": "Test City",
      "country": "Test Country"
    }
  }
}
```

```

        }
    },
"product": {
    "product_id": "PROD-VALIDO",
    "category": "Technology",
    "sub_category": "Accessories",
    "name": "Producto Válido"
},
"sales_data": {
    "sales": 250.75,
    "quantity": 3,
    "discount": 0.1,
    "profit": 50.25,
    "shipping_cost": 15.50
}
}
}

```

Insert Document

To collection globalmarket_db.sales_transformed

```

3      "order_id": "TEST-VALIDO-001",
4      "order_date": "20-01-2025",
5      "ship_mode": "Standard",
6      "order_priority": "High"
7      },
8      "customer": {
9          "customer_id": "CLIENTE-OK",
10         "name": "Cliente Prueba",
11         "segment": "Consumer",
12         "location": [
13             "city": "Test City",
14             "country": "Test Country"
15         ]
16     },
17     "product": {
18         "product_id": "PROD-VALIDO",
19         "category": "Technology",
20         "sub_category": "Accessories",
21         "name": "Producto Válido"
22     },
23     "sales_data": [
24         "sales": 250.75,
25         "quantity": 3,
26         "discount": 0.1,
27         "profit": 50.25,
28         "shipping_cost": 15.50
29     ]
30 }

```

Cancel

Insert

5. Aplicación del Pipeline de Transformación en Sales

a. Pipeline en text

```
{  
  "$project": {  
    "order_info": {  
      "order_id": "$Order ID",  
      "order_date": "$Order Date",  
      "ship_date": "$Ship Date",  
      "ship_mode": "$Ship Mode",  
      "order_priority": "$Order Priority"  
    },  
    "customer": {  
      "customer_id": "$Customer ID",  
      "name": "$Customer Name",  
      "segment": "$Segment",  
      "location": {  
        "city": "$City",  
        "state": "$State",  
        "country": "$Country",  
        "postal_code": "$Postal Code",  
        "market": "$Market",  
        "region": "$Region"  
      }  
    },  
    "product": {  
      "product_id": "$Product ID",  
      "name": "$Product Name",  
      "category": "$Category",  
      "sub_category": "$Sub-Category"  
    },  
    "sales_data": {  
      "sales": "$Sales",  
      "quantity": "$Quantity",  
      "discount": "$Discount",  
      "profit": "$Profit",  
      "shipping_cost": "$Shipping Cost"  
    }  
  }  
}
```

```

[{"$match": {"sales_data.sales": { $gt: 0 }, "product.category": { $exists: true }}, {"$addFields": {"order_month": { $dateToString: { format: "%Y-%m", date: { $dateFromString: { dateString: "$order_info.order_date", format: "%d-%m-%Y" } } } }}, {"$group": {"_id": { category: "$product.category", month: "$order_month" }, total_sales: { $sum: "$sales_data.sales" }, total_orders: { $sum: 1 }, avg_profit: { $avg: "$sales_data.profit" } }}, {"$project": {"customer": { _id: "$customer._id", name: "$customer.name", segment: "$customer.segment", location: { city: "$customer.location.city", state: "$customer.location.state", country: "$customer.location.country", postal_code: "$customer.location.postal_code", market: "$customer.location.market", region: "$customer.location.region" } }, product: { _id: "$product._id", name: "$product.name", category: "$product.category", sub_category: "$product.subCategory" }, sales_data: { sales: "$sales_data.sales", quantity: "$sales_data.quantity", discount: "$sales_data.discount", profit: "$sales_data.profit", shipping_cost: "$sales_data.shippingCost" } }}, {"$output": {"stages": [{"$sample": 10}], "options": {"outputFormat": "json"}}}]

```

6. Pipeline 1: Reporte De Ventas Por Categoría Y Mes

```

[{"$match": {"sales_data.sales": { $gt: 0 }, "product.category": { $exists: true }}, {"$addFields": {"order_month": { $dateToString: { format: "%Y-%m", date: { $dateFromString: { dateString: "$order_info.order_date", format: "%d-%m-%Y" } } } }}, {"$group": {"_id": { category: "$product.category", month: "$order_month" }, total_sales: { $sum: "$sales_data.sales" }, total_orders: { $sum: 1 }, avg_profit: { $avg: "$sales_data.profit" } }}, {"$project": {"customer": { _id: "$customer._id", name: "$customer.name", segment: "$customer.segment", location: { city: "$customer.location.city", state: "$customer.location.state", country: "$customer.location.country", postal_code: "$customer.location.postal_code", market: "$customer.location.market", region: "$customer.location.region" } }, product: { _id: "$product._id", name: "$product.name", category: "$product.category", sub_category: "$product.subCategory" }, sales_data: { sales: "$sales_data.sales", quantity: "$sales_data.quantity", discount: "$sales_data.discount", profit: "$sales_data.profit", shipping_cost: "$sales_data.shippingCost" } }}, {"$output": {"stages": [{"$sample": 10}], "options": {"outputFormat": "json"}}}]

```

```

        }
    },
{
  $project: {
    _id: 0,
    category: "$_id.category",
    month: "$_id.month",
    total_sales: 1,
    total_orders: 1,
    avg_profit: 1,
    sales_per_order: { $divide: ["$total_sales", "$total_orders"] }
  }
},
{
  $sort: { month: 1, total_sales: -1 }
}
]

```

The screenshot shows the MongoDB Compass interface with the following details:

- Database:** globalmarket-cluster.3rebz5q.mongodb.net > globalmarket_db > sales_transformed
- Collection:** sales_transformed (5K documents)
- Aggregations Tab:** The current tab.
- Results:**
 - Furniture:** total_sales: 34463.753, total_orders: 80, avg_profit: 8.648011249999997, category: "Furniture", month: "2011-01", sales_per_order: 438.7969124999996
 - Office Supplies:** total_sales: 33526.7275, total_orders: 286, avg_profit: 15.992091258741258, category: "Office Supplies", month: "2011-01", sales_per_order: 117.22631993006993
 - Technology:** total_sales: 20908.00936, total_orders: 67, avg_profit: 45.62480537313433, category: "Technology", month: "2011-01", sales_per_order: 461.31355761194827
- Buttons:** \$match, \$addFields, \$group, \$project, \$sort, Edit, Explain, Export, Run, Options
- Information:** Showing 1 - 20 count results

The screenshot shows the MongoDB Compass interface with the following details:

- My Queries:** A list of saved queries.
- Query Preview:**
 - NAME:** reporte_ventas_catego...
 - Database:** globalmarket_db
 - Collection:** sales_transformed
 - Last modified:** 2 minutes ago

7. Pipeline 2: completo y auto-contenido (genera rating aleatorio y filtra)

Como el dataset de ventas no incluye ratings, simulamos un campo rating aleatorio entre 1 y 5 para demostrar el uso del aggregation pipeline con \$match, \$group y \$project, cumpliendo la estructura solicitada.

top_productos_mejor_calificacion

```
[  
  {  
    $addFields: {  
      rating: {  
        $add: [  
          { $floor: { $multiply: [ { $rand: {} }, 5 ] } },  
          1  
        ]  
      },  
      factor_resenas: 10  
    }  
  },  
  {  
    $group: {  
      _id: "$product.product_id",  
      nombre: { $first: "$product.name" },  
      categoria: { $first: "$product.category" },  
      total_resenas: { $sum: "$factor_resenas" },  
      calificacion_promedio: { $avg: "$rating" }  
    }  
  },  
  {  
    $match: {  
      total_resenas: { $gt: 50 }  
    }  
  },  
  {  
    $sort: {  
      calificacion_promedio: -1  
    }  
  },  
  {  
    $project: {  
      _id: 0,  
      producto_id: "$_id",  
      nombre_producto: "$nombre",  
      categoria: "$categoria",  
      total_resenas: 1,  
      calificacion_promedio: { $round: ["$calificacion_promedio", 2] }  
    }  
  },  
]
```

```
{
  $limit: 10
}
]
```

Untitled - modified SAVE + CREATE NEW EXPORT TO LANGUAGE

PREVIEW STAGES TEXT WIZARD

Pipeline Output Preview
Sample of 10 documents

OUTPUT OPTIONS

```

4   rating: [
5     $add: [
6       { $floor: { $Multiply: [ { $rand: {} }, 5 ] } },
7       1
8     ]
9   },
10  factor_resenas: 10
11 },
12 },
13 {
14   $group: {
15     _id: "$product.product_id",
16     nombre: { $first: "$product.name" },
17     categoria: { $first: "$product.category" },
18     total_resenas: { $sum: "$factor_resenas" },
19     calificacion_promedio: { $avg: "$rating" }
20   }
21 },
22 {
23   $match: {
24     total_resenas: { $gt: 50 }
25   },
26   {
27     $sort: { calificacion_promedio: -1 }
28   },
29   {
30     $project: {
31       _id: 0,
32       producto_id: "$_id",
33       nombre_producto: "$nombre",
34       categoria: "$categoria",
35       total_resenas: 1,
36       calificacion_promedio: { $round: ["$calificacion_promedio", 2] }
37     }
38   }
39 }
```

total_resenas	producto_id	nombre_producto	categoria	calificacion_promedio
60	OFF-PA-10000523	Southworth Parchment Paper & Envelopes	Office Supplies	4.92
80	OFF-FA-10000554	DIC Bulk Pack Metal Binder Clips	Office Supplies	4.75
60	OFF-STA-10000558	Startech Card Printer, Red	Technology	4.67
60	TEC-PH-10000177	Samsung Speaker Phone, with Caller ID	Technology	4.67
60	TEC-PH-10000177	Samsung Speaker Phone, with Caller ID	Technology	4.67
60	TEC-PH-10000177	Samsung Speaker Phone, with Caller ID	Technology	4.67
60	TEC-PH-10000177	Samsung Speaker Phone, with Caller ID	Technology	4.67
60	TEC-PH-10000177	Samsung Speaker Phone, with Caller ID	Technology	4.67
60	TEC-PH-10000177	Samsung Speaker Phone, with Caller ID	Technology	4.67
60	TEC-PH-10000177	Samsung Speaker Phone, with Caller ID	Technology	4.67

My Queries +

Search All databases A

AGGREGATE

reporte_ventas_categ...

globalmarket_db sales_transformed

Last modified: 36 minutes ago

AGGREGATE

top_productos_mejor...

globalmarket_db sales_transformed

Last modified: 2 minutes ago

8. Pipeline 3: Bucket por rangos de precios unitarios

Objetivo:

Agrupar productos en 3 rangos de precio unitario: **Bajo, Medio, Alto.**

```
[
  // Paso 1: Calcular precio unitario (sales / quantity)
  {
    $addFields: {
```

```

precio_unitario: {
    $cond: {
        if: { $gt: ["$sales_data.quantity", 0] },
        then: { $divide: ["$sales_data.sales", "$sales_data.quantity"] },
        else: 0
    }
}
},
// Paso 2: Agrupar en rangos usando $bucket
{
    $bucket: {
        groupBy: "$precio_unitario",
        boundaries: [0, 50, 200, 1000], // Rangos: Bajo (0-50), Medio (50-200), Alto (200-1000)
        default: "Fuera de rango",
        output: {
            count: { $sum: 1 },
            productos: { $push: "$product.name" },
            precio_promedio: { $avg: "$precio_unitario" }
        }
    }
},
// Paso 3: Renombrar rangos para claridad
{
    $project: {
        _id: 0,
        rango: {
            $switch: {
                branches: [
                    { case: { $eq: ["$_id", "Fuera de rango"] }, then: "Fuera de rango" },
                    { case: { $lt: ["$_id", 50] }, then: "Bajo" },
                    { case: { $lt: ["$_id", 200] }, then: "Medio" },
                    { case: { $gte: ["$_id", 200] }, then: "Alto" }
                ],
                default: "No clasificado"
            }
        },
        cantidad_productos: "$count",
        precio_promedio: { $round: ["$precio_promedio", 2] }
    }
},
// Paso 4: Ordenar por rango
{
    $sort: {
        rango: 1
    }
}
]

```

Untitled - modified SAVE + CREATE NEW EXPORT TO LANGUAGE

```

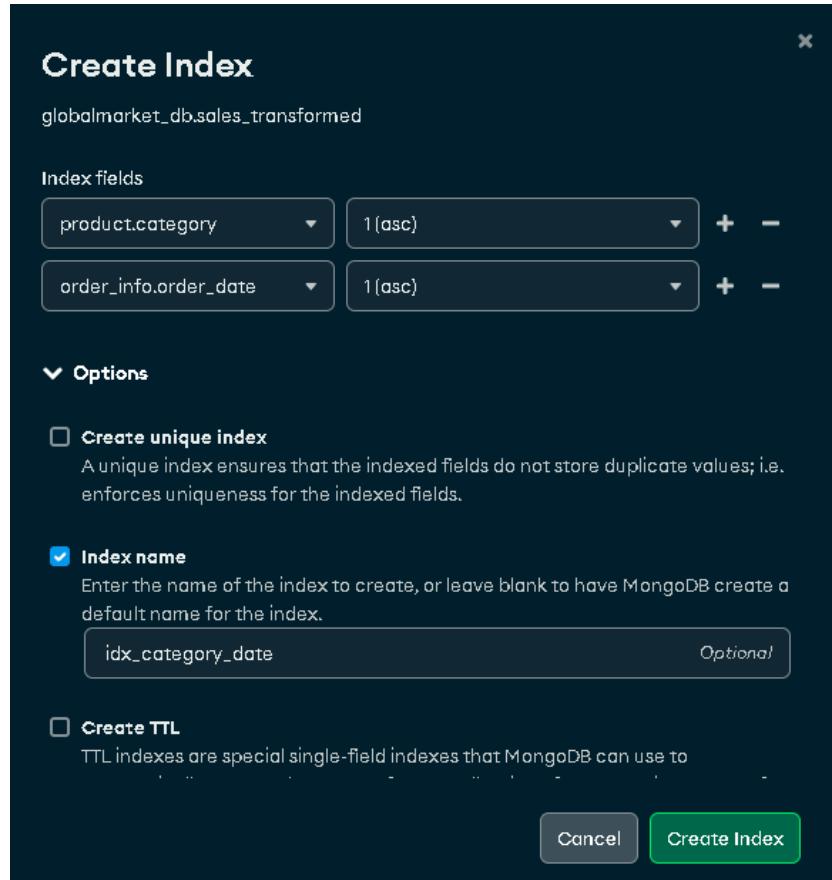
1  [
2    // Paso 1: Calcular precio unitario (sales / quantity)
3    [
4      $addFields: [
5        precio_unitario: [
6          $cond: [
7            if: { $gt: ["$sales_data.quantity", 0] },
8            then: { $divide: ["$sales_data.sales", "$sales_data.quantity"] },
9            else: 0
10           }
11         ]
12       ]
13     ],
14    // Paso 2: Agrupar en rangos usando $bucket
15    [
16      $bucket: [
17        groupBy: "$precio_unitario",
18        boundaries: [0, 50, 200, 1000], // Rangos: Bajo (0-50), Medio (50-200), Alto (200+)
19        default: "Fuera de rango",
20        output: [
21          count: { $sum: 1 },
22          productos: { $push: "$product.name" },
23          precio_promedio: { $avg: "$precio_unitario" }
24        ]
25      ]
26    ],
27    // Paso 3: Renombrar rangos para claridad
28    [
29      $project: {
30        _id: 0,
31        rango: [
32          $switch: [
33            branches: [
34              { case: { $eq: ["$_id", "Fuera de rango"] }, then: "Fuera de rango" },
35              { case: { $lt: ["$_id", 50] }, then: "Bajo" },
36              { case: { $lt: ["$_id", 200] }, then: "Medio" },
37              { case: { $gte: ["$_id", 200] }, then: "Alto" }
38            ],
39            default: "No clasificado"
40          }
41        ],
42        cantidad_productos: "$count",
43        precio_promedio: "$precio_promedio"
44      }
45    ]
46  ]
47]

```

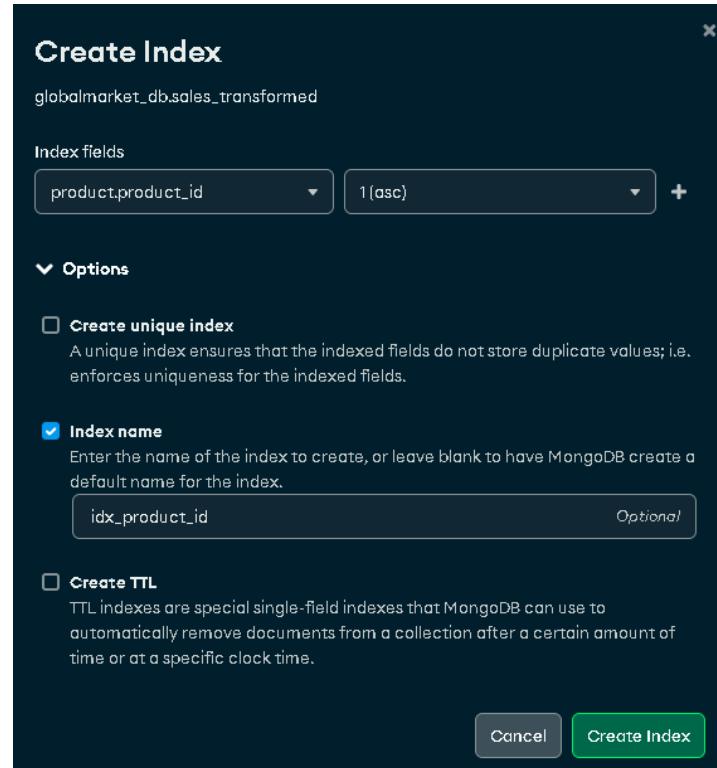
Pipeline Output Preview
Sample of 4 documents

- rango : "Alto"
cantidad_productos : 4524
precio_promedio : 345.46
- rango : "Bajo"
cantidad_productos : 23199
precio_promedio : 19.18
- rango : "Fuera de rango"
cantidad_productos : 27
precio_promedio : 1662.59
- rango : "Medio"
cantidad_productos : 13530
precio_promedio : 184.52

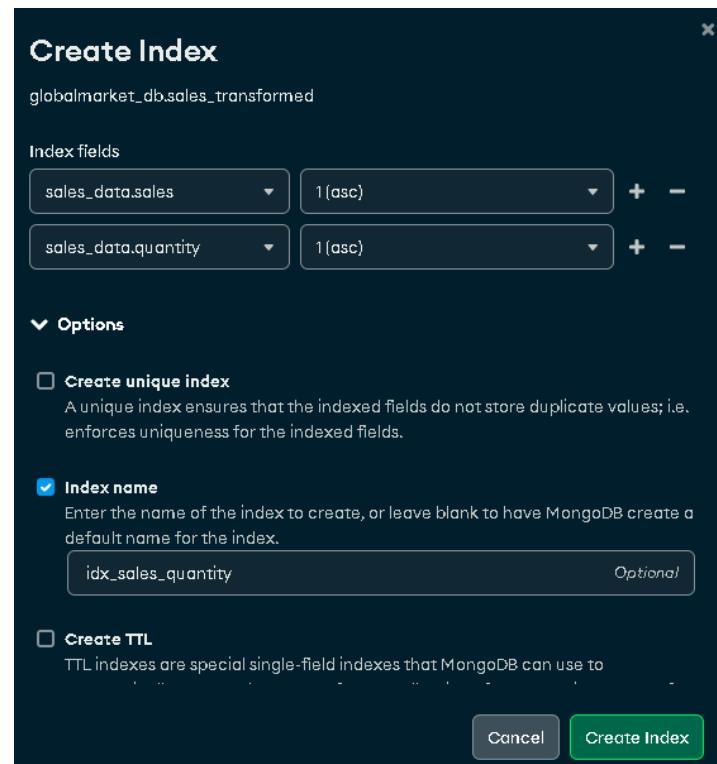
ÍNDICE 1: Compuesto (categoría + fecha)



ÍNDICE 2: Simple (product_id)



ÍNDICE 3: Compuesto (sales + quantity)



globalmarket-cluster.3rebz5q.mongodb.net > globalmarket_db > sales_transformed

[Open MongoDB shell](#)

Documents	51K	Aggregations	Schema	Indexes (5)	Validation
				Create Refresh	VIEWING INDEXES SEARCH INDEXES
Name & Definition	Type	Size	Usage	Properties	Status
id	REGULAR i	2.0 MB	47 (since Mon Dec 01 2025)	UNIQUE i	READY
text_product_name	TEXT i	2.2 MB	10 (since Thu Dec 04 2025)		READY
idx_category_date	REGULAR i	643.1 kB	0 (since Thu Dec 04 2025)	COMPOUND i	READY
idx_product_id	REGULAR i	622.6 kB	0 (since Thu Dec 04 2025)		READY
idx_sales_quantity	REGULAR i	720.9 kB	0 (since Thu Dec 04 2025)	COMPOUND i	READY

Observación

En la configuración de la búsqueda con Atlas Search, debíamos poder buscar productos por nombre con tolerancia a errores tipográficos, es decir, con índice Fuzzy_product_search en product name. Debido a las limitaciones del tier M0 gratuito, las consultas Search no retornan resultados.

```
{  
  "mappings": {  
    "dynamic": false,  
    "fields": {  
      "product.name": {  
        "analyzer": "lucene.standard",  
        "searchAnalyzer": "lucene.standard",  
        "type": "string"  
      }  
    }  
  }  
}
```

fuzzy_product_search READY QUERYABLE

This search index parses the data in **globalmarket_db.sales_transformed** and has the following configurations.

[View Atlas Search Docs](#)

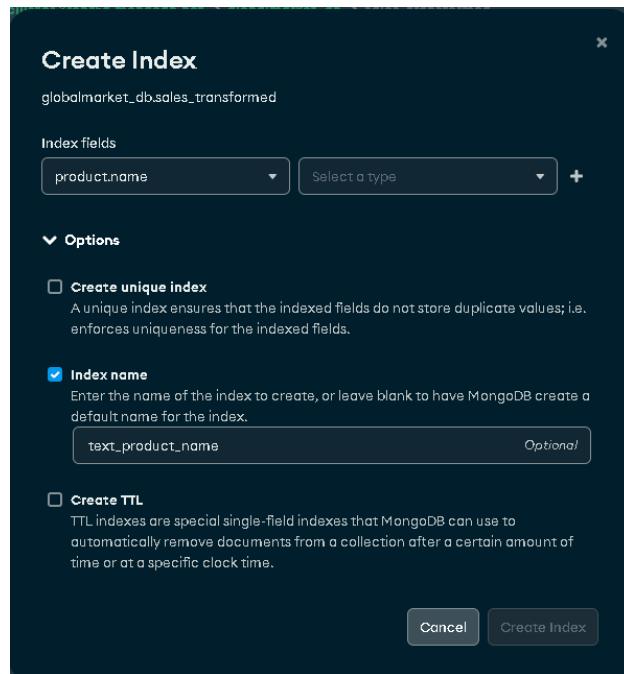
```
1  {  
2    "mappings": {  
3      "dynamic": false,  
4      "fields": {  
5        "product.name": {  
6          "analyzer": "lucene.standard",  
7          "searchAnalyzer": "lucene.standard",  
8          "type": "string"  
9        }  
10      }  
11    }  
12  }
```

Database	Collection	Index Name	Status	Queryable	Type	Index Fields	Documents (Estimated)	Size	Actions
globalmarket_db	sales_transformed	fuzzy_product_search	READY	✓	search	"product.name"	51,290 (100%) indexed of 51,290	783.2KB	QUERY ...

[Learn more about Atlas Search](#) [Learn more about Atlas Vector Search](#)

Como alternativa funcional, implementamos un índice de texto tradicional con text y simulamos la búsqueda difusa

Creamos el index



```
[  
{  
  $match: {  
    $text: { $search: "Headset" }  
  },  
  {  
    $project: {  
      "product.name": 1,  
      "product.category": 1,  
      score: { $meta: "textScore" }  
    },  
  },  
  {  
    $sort: {  
      score: { $meta: "textScore" }  
    },  
  },  
  {  
    $limit: 5  
  }  
]
```

Pipeline Output Preview

Sample of 5 documents

OUTPUT OPTIONS ▾

```

1  [ 
2   {
3     $match: {
4       $text: { $search: "Headset" }
5     }
6   },
7   [
8     {
9       $project: {
10        "product.name": 1,
11        "product.category": 1,
12        score: { $meta: "textScore" }
13      }
14    },
15    {
16      $sort: {
17        score: { $meta: "textScore" }
18      }
19    },
20    {
21      $limit: 5
22    }
23  ]

```

_id: ObjectId('692d1362209add793aae8464')
 product : Object
 name : "Samsung Headset, VoIP"
 category : "Technology"
 score : 0.6666666666666666

_id: ObjectId('692d1363209add783aae96fb')
 product : Object
 name : "Nokia Headset, Cordless"
 category : "Technology"
 score : 0.6666666666666666

_id: ObjectId('692d1361209add793aae7496')
 product : Object
 score : 0.6666666666666666

_id: ObjectId('692d1361209add793aae76df')
 product : Object
 score : 0.6666666666666666

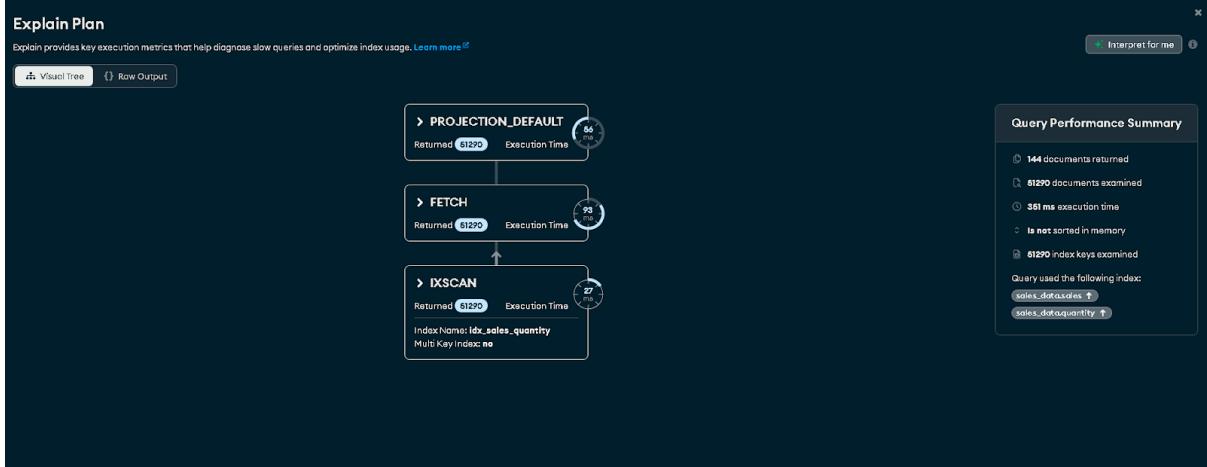


Explain plan

Paso 1: Cargar el pipeline

Selecciona **reporte_ventas_categoria_mes**

Paso 2: Ejecutar Explain



Del archivo JSON

Índice usado. **Muestra que se usó un índice** (no COLLSCAN).

```
    "inputStage": {  
        "stage": "IXSCAN",  
        "keyPattern": {  
            "sales_data.sales": 1,  
            "sales_data.quantity": 1  
        },  
        "indexName": "idx_sales_quantity",  
        "isMultiKey": false,  
        "multiKeyPaths": {  
            "sales_data.sales": [],  
            "sales_data.quantity": []  
        }  
    }
```

Estadísticas de ejecución

- **Tiempo total:** 351 ms.
- **Documentos examinados:** 51,290 (todos los documentos, porque el filtro sales > 0 aplica a casi todos).

```

    "executionSuccess": true,
    "nReturned": 51290,
    "executionTimeMillis": 351,
    "totalKeysExamined": 51290,
    "totalDocsExamined": 51290,
    ...

```

Plan rechazado. Muestra que MongoDB consideró otro índice (`idx_category_date`) pero eligió `idx_sales_quantity` porque estimó que era más eficiente.

```

{
  "rejectedPlans": [
    {
      "isCached": false,
      "stage": "PROJECTION_DEFAULT",
      "transformBy": [
        "order_info.order_date": 1,
        "product.category": 1,
        "sales_data.profit": 1,
        "sales_data.sales": 1,
        "_id": 0
      ],
      "inputStage": {
        "stage": "FETCH",
        "filter": {
          "$and": [
            {
              "product.category": {
                "$exists": true
              }
            },
            {
              "sales_data.sales": {
                "$gt": 0
              }
            }
          ]
        },
        "inputStage": {
          "stage": "IXSCAN",
          "keyPattern": {
            "product.category": 1,
            "order_info.order_date": 1
          },
          "indexName": "idx_category_date",
          "isMultiKey": false,
          "multiKeyPaths": [
            "product.category": [],
            "order_info.order_date": []
          ]
        }
      }
    }
  ]
}

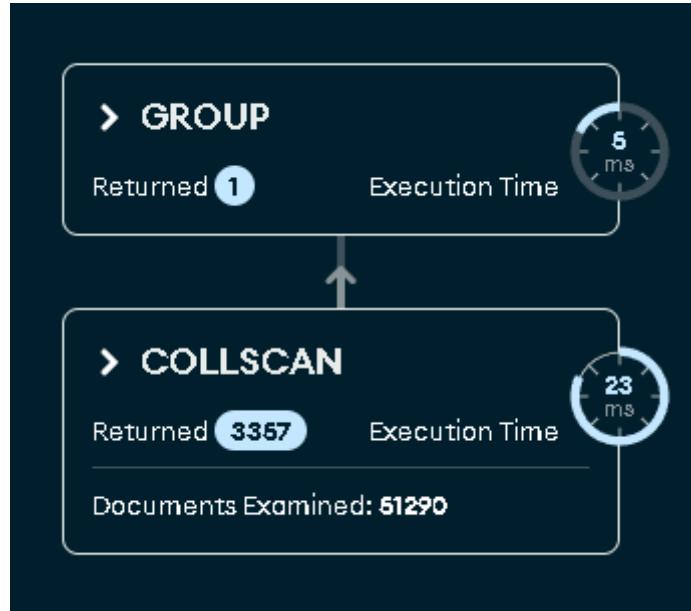
```

El Explain Plan muestra que la consulta utilizó el índice `idx_sales_quantity` (IXSCAN) en lugar de un escaneo completo (COLLSCAN), examinando 51,290 claves y tomando 351 ms. MongoDB también consideró el índice `idx_category_date` pero lo rechazó basado en el optimizador de consultas

Una demostración directa sin índice (COLLSCAN):

```
[
  { $match: { "product.sub_category": "Phones" } }, // Campo sin índice
  { $count: "total" }
```

]

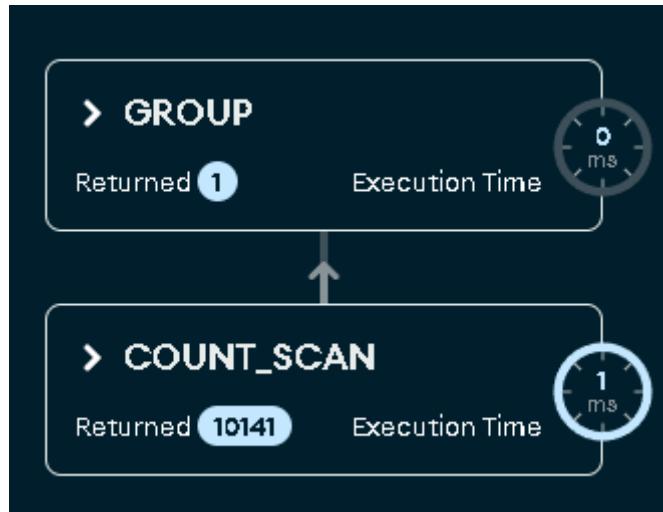


Con índice (IXSCAN):

[

```
{ $match: { "product.category": "Technology" } }, // Usa idx_category_date
{ $count: "total" }
```

]



Visualización



<https://charts.mongodb.com/charts-project-0-xueadb/public/dashboards/9a480942-1fd4-454e-9ae7-1e0342ac5e5c>

Dashboard

1. Ventas por país (gráfico de barras – arriba izquierda)

Es un ranking de ventas totales por país.

El eje Y muestra la suma de ventas.

El eje X muestra los países.

Estados Unidos aparece como el país con mayores ventas (más de 2.400.000).

Luego vienen Indonesia, México, Honduras, Bangladesh, etc.

La forma de la curva indica que pocos países concentran la mayoría de las ventas, típico comportamiento long tail.

2. Cantidad por ganancias (gráfico de rosquilla – arriba centro)

Muestra la distribución de la ganancia agrupada por rangos.

Cada color representa un rango de ganancia (ej: 500–550, 600–650, etc.).

La gráfica permite ver cuáles rangos de ganancia son más frecuentes.

Parece que los rangos medios (alrededor de 300–450) concentran más cantidad.

3. Productos vendidos (gráfico horizontal – arriba derecha)

Compara dos métricas:

Cantidad de ventas (verde)

Cantidad de productos distintos vendidos (azul)

Las barras están normalizadas (0–100%).

Se observan productos específicos como:

Staples

Eidon Tray

Novimex Stool

Brother Fax

SanDisk Memo Slips

etc.

Permite identificar qué productos tienen más movimiento o más variedad.

4. Ventas Totales (parte inferior izquierda)

Es un KPI destacado que muestra el total de ventas acumuladas.

Valor total de ventas: 12,642,501.9099

Representa todo el ingreso comercial sumado en la base de datos.

5. Productos por fecha — Productos más pedidos por año (abajo centro)

Gráfico lineal de número de pedidos por fecha.

En el eje X aparecen fechas ordenadas.

En el eje Y se muestra el número de pedidos.

La curva es descendente: algunas fechas o años tienen muchos pedidos y luego va cayendo en orden ranking.

Funciona como un top de fechas más demandadas.

6. Producto por país – Mapa mundial (abajo derecha)

Mapa de calor que muestra cuántos productos se venden por país.

Tonos más oscuros = mayor cantidad.

Estados Unidos aparece con el mayor conteo (al nivel de 9000 productos).

Otros países tienen valores más bajos.

Permite ver dónde hay mayor actividad comercial.

Anexo

Repositorio de Github <https://github.com/haddan17/GlobalMarket---MongoDB>

Recomendaciones generales

- Fortalecer la validación de datos configurando reglas estrictas para tipos, rangos y valores permitidos, de forma que la base de datos rechace automáticamente registros inconsistentes y mantenga la calidad de la información.
- Incorporar reglas de negocio adicionales en el esquema (por ejemplo, coherencia entre ventas, cantidad, descuentos y precios unitarios) para asegurar que los datos utilizados en los reportes reflejen correctamente la realidad comercial.
- Optimizar los pipelines de agregación colocando los filtros y proyecciones en las primeras etapas y verificando que los campos usados en filtros, agrupaciones y ordenamientos estén debidamente indexados, con el fin de reducir documentos examinados y mejorar los tiempos de respuesta.
- Evaluar el uso de un plan de servicio superior en la nube cuando se requiera búsqueda difusa real (fuzzy search) sobre nombres de productos, aprovechando índices especializados de búsqueda en lugar de depender únicamente de índices de texto tradicionales.
- Enriquecer el dashboard analítico incorporando indicadores accionables (productos de alta y baja rotación, clientes de alto valor, fechas pico de demanda, anomalías en descuentos), de manera que las visualizaciones apoyen decisiones concretas de ventas, marketing e inventario.

Conclusión

El modelo documental aplicado a través de MongoDB consolida en un solo documento la información de pedido, cliente, producto y datos de venta, lo que reduce la fragmentación típica del modelo relacional y facilita consultas analíticas y operativas sobre ventas, categorías y clientes. Esta estructura, junto con los pipelines de agregación implementados (reportes por categoría y mes, bucket de rangos de precios, ranking de productos, etc.), muestra un uso correcto del framework de agregación para análisis de negocio y soporta un dashboard claro en Charts para monitorear ventas, ganancias y distribución geográfica. Además, el uso de validación con `jsonSchema` y de índices compuestos mejora la calidad de los datos y demuestra preocupación por el rendimiento de las consultas al evitar escaneos completos de colección mediante IXSCAN, lo que es consistente con las buenas prácticas de MongoDB para validación e indexación.