

CEDINSI
BOGOTÁ, JUNIO DE 2024
ENTREGA FINAL BASES DE DATOS
ALUMNO: JUAN MANUEL SILVA LÓPEZ
PROFESOR: CAMILO PINEDA

CASO: CERVECERIA - BASE DE DATOS NOSQL MONGO-DB

REGLAS DEL NEGOCIO:

1. Crear una base de datos nosql para el backoffice de una hipotética aplicación de una fábrica de cervezas..
2. Se debe generar una colección para mostrar los tipos de cerveza que será de 3 tipos:
Ale, Lager y Negra.
Cada cerveza Lager tiene un precio de venta de 13,4
Cada cerveza Ale tiene un precio de venta de 12.5
Cada cerveza Negra tiene un precio de venta de 15,0
3. Se debe generar una colección para registrar la materia prima que será de 4 ingredientes: Agua, Lúpulo, Cebada y Col. La cantidad inicial de inventario de materias primas será de 1000 unidades por cada uno de los cuatro tipos de materia prima.
4. Se debe generar una colección para registrar la fabricación de diferentes tipos de cervezas para lo cual se determina que para fabricar una unidad de cada tipo de cerveza se usará una unidad de cada materia prima.
5. Se debe generar la consulta de actualización que permita actualizar en la colección de materias primas los totales de unidades consumidas de materias primas, descontando de la colección de materias primas la cantidad pertinente de inventario para producir lotes de 100 unidades de cada tipo de cerveza.
6. Se debe generar la consulta de actualización que permita actualizar en la colección de tipos de cerveza los totales de unidades de cada tipo de cerveza producida.
7. Agregar una simulación de vender cervezas agregando 1 pedido de 50 cervezas de cada tipo disponible.

SOLUCIÓN:

1. Crear una base de datos

localhost:27017

My QueriesPerformanceDatabases

adminbookscerveceria_juanconfiglocalpruebouniversity

Create databaseRefresh

admin

Storage size: 20.48 kBCollections: 1Indexes: 1

books

Storage size: 81.92 kBCollections: 4Indexes: 4

cerveceria_juan

Storage size: 69.63 kBCollections: 4Indexes: 4

2. Crear colección tipos_de_cerveza e incluir los items: Ale, Lager y Negra

MongoDB Compass - localhost:27017/cerveceria_juan.tipos_de_cerveza

Connect Edit View Collection Help

localhost:27017

My QueriesPerformanceDatabases

adminbookscerveceria_juanmateria_primeproduccion

tipos_de_cervezaventas

configlocalpruebouniversity

localhost:27017 > cerveceria_juan > tipos_de_cerveza

Documents3AggregationsSchemaIndexes1Validation

Type a query: { field: 'value' } or Generate query

ADD DATAEXPORT DATAUPDATEDELETE

_id: ObjectId('666d0d992cd90ec0fd8e5604')

nombre: "Ale"

descripcion: "Cerveza de alta fermentación con sabor intenso y aroma frutal."

unidades_producidas: 0

precio_venta: 12.5

_id: ObjectId('666d0d992cd90ec0fd8e5605')

nombre: "Lager"

descripcion: "Cerveza de baja fermentación con sabor ligero y refrescante."

unidades_producidas: 0

precio_venta: 13.4

_id: ObjectId('666d0d992cd90ec0fd8e5606')

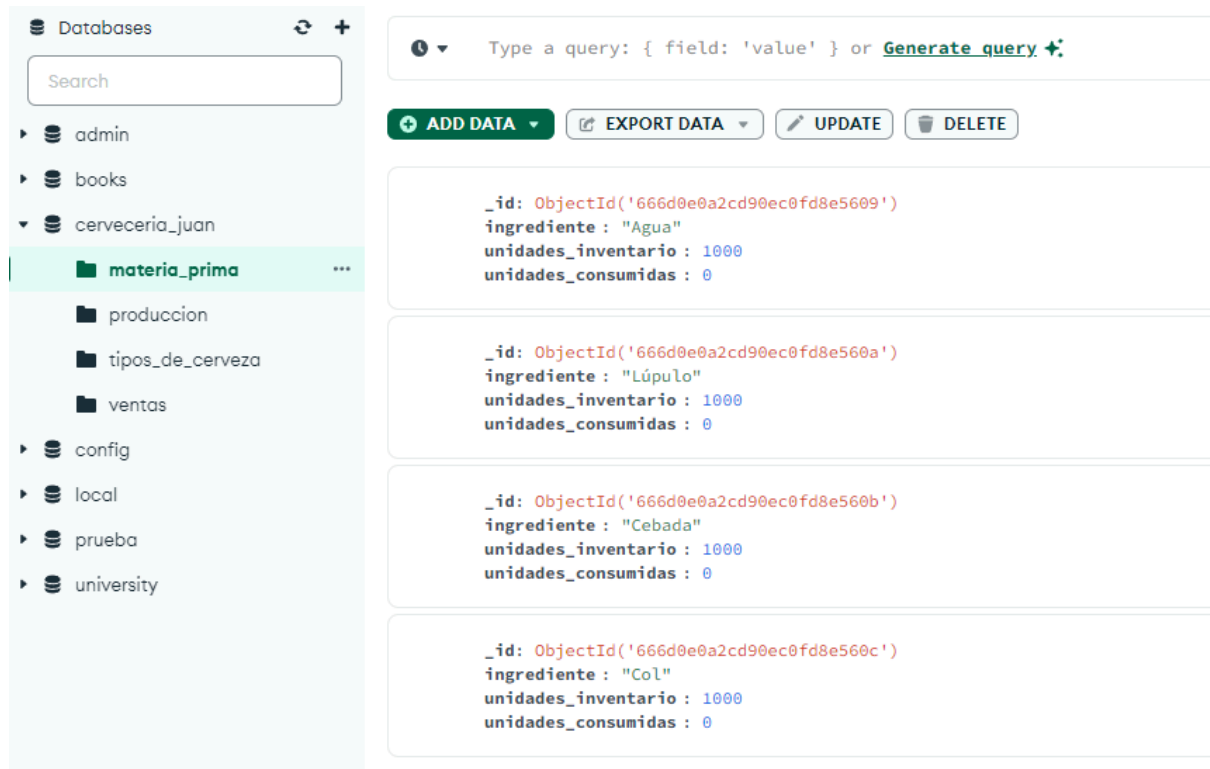
nombre: "Negra"

descripcion: "Cerveza oscura con sabor fuerte y aroma tostado."

unidades_producidas: 0

precio_venta: 15

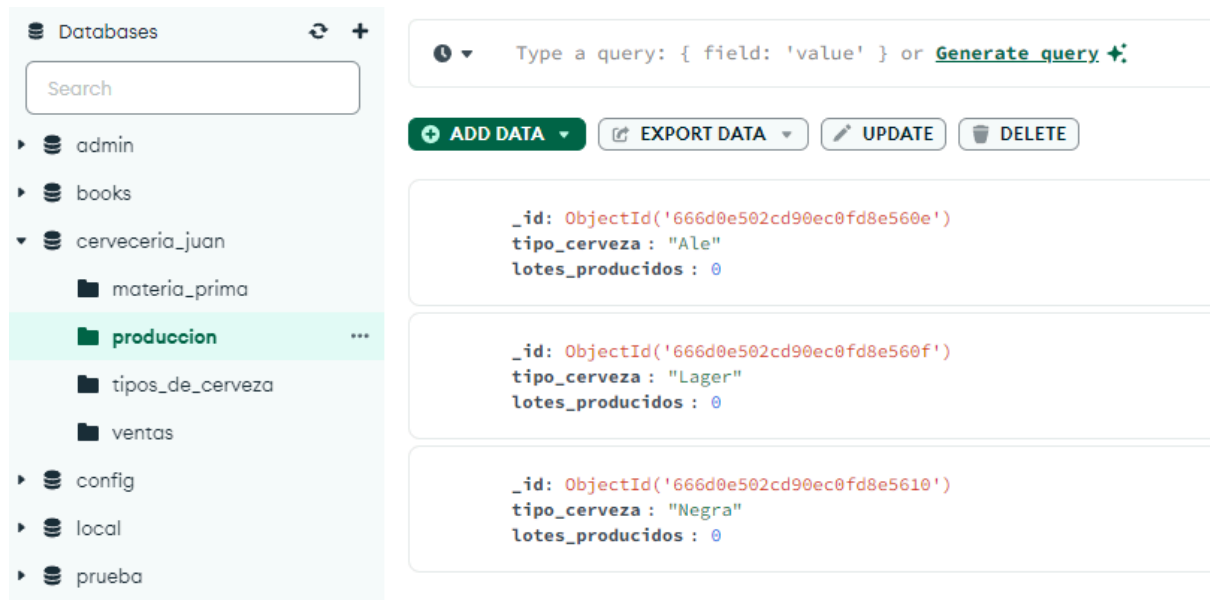
3. Generar una colección para registrar la materia prima que será de 4 ingredientes: Agua, Lúpulo, Cebada y Col. La cantidad inicial de inventario de materias primas será de 1000 unidades por cada uno de los cuatro tipos de materia prima.



The screenshot shows the MongoDB Compass interface. On the left, the 'Databases' sidebar lists 'admin', 'books', 'cerveceria_juan', and 'materia_prima' (highlighted). The main panel displays the 'materia_prima' collection with four documents. Each document has the following structure:

- `_id`: ObjectId (e.g., '666d0e0a2cd90ec0fd8e5609')
- `ingrediente`: String (e.g., 'Agua')
- `unidades_inventario`: Integer (1000)
- `unidades_consumidas`: Integer (0)

4. Generar colección para registrar la fabricación de diferentes tipos de cervezas. Se determina que para fabricar una unidad de cada tipo de cerveza se usará una unidad de cada materia prima.



The screenshot shows the MongoDB Compass interface. On the left, the 'Databases' sidebar lists 'admin', 'books', 'cerveceria_juan', 'materia_prima', and 'produccion' (highlighted). The main panel displays the 'produccion' collection with three documents. Each document has the following structure:

- `_id`: ObjectId (e.g., '666d0e502cd90ec0fd8e560e')
- `tipo_cerveza`: String (e.g., 'Ale')
- `lotes_producidos`: Integer (0)

5. Agregar una simulación de vender cervezas. Se crea otra colección Ventas, agregando 1 pedido de 50 cervezas de cada tipo disponible.

The screenshot shows the MongoDB Compass interface. On the left, the database 'cerveceria_juan' is expanded, showing collections 'materia_prima', 'produccion', 'tipos_de_cerveza', and 'ventas'. The 'ventas' collection is selected. On the right, a document is displayed with the following structure:

```
{
  "_id": ObjectId('666d0edd2cd90ec0fd8e5612'),
  "fecha_venta": "2024-06-14",
  "cervezas_vendidas": [
    {
      "tipo_cerveza": "Ale",
      "unidades": 50
    },
    {
      "tipo_cerveza": "Lager",
      "unidades": 50
    },
    {
      "tipo_cerveza": "Negra",
      "unidades": 50
    }
  ],
  "total_venta": 0
}
```

6. Generar la consulta de actualización de inventario de materias primas. Actualiza en la colección de materias primas los totales de unidades consumidas de materias primas, descontando de la colección de materias primas la cantidad pertinente de inventario para producir lotes de 100 unidades de cada tipo de cerveza.

```
cerveceria_juan.materiaprima.updateMany(
  {},
  [
    { $inc: { "unidades_consumidas": 50 } },
    { $inc: { "unidades_inventario": -50 } }
  ]
)
```

7. Generar la consulta de actualización de los totales de unidades de cada tipo de cerveza producida en la colección.

```
cerveceria_juan.produccion.updateMany( {}, [ { $inc: { "unidades_producidas": -50 } } ] )
```