

Nombre: Juan Esteban Valdés Rodríguez

Numero de documento: 1006351693

Correo: juanv04122002@gmail.com

## PostgreSQL

### Query para crear la tabla productos

```
Query Query History
1  -- Tabla: catalogo.productos
2
3  ✓ COMMENT ON TABLE catalogo.productos
4    IS 'Almacena información de productos';
5
6  -- Crear esquema si no existe
7  ✓ CREATE SCHEMA IF NOT EXISTS catalogo
8    AUTHORIZATION postgres; -- Ajusta el usuario según la configuración
9
10 COMMENT ON SCHEMA catalogo IS 'Esquema para catálogo de productos';
11
12 /* ***** Tabla: productos ***** */
13 -- Eliminar tabla si existe (para reiniciar)
14 DROP TABLE IF EXISTS catalogo.productos CASCADE;
15
16 -- Crear tabla
17 ✓ CREATE TABLE catalogo.productos (
18   id_producto SERIAL PRIMARY KEY, -- SERIAL esta obsoleto pero para mantener la simplicidad lo use para este ejercicio
19   nombre VARCHAR(100) NOT NULL, -- Límite razonable para cadenas de texto en 'nombre'
20   categoria VARCHAR(30) NOT NULL, -- Límite razonable para cadenas de texto en 'categorias'
21   precio NUMERIC(10) NOT NULL CHECK (precio >= 0) -- Para evitar numeros negativos
22 );
23
24 -- Comentarios descriptivos
25 COMMENT ON TABLE catalogo.productos IS 'Almacena información de los productos';
26 COMMENT ON COLUMN catalogo.productos.id_producto IS 'Identificador único autoincremental';
27 COMMENT ON COLUMN catalogo.productos.nombre IS 'Nombre del producto (máx. 100 caracteres)';
28 COMMENT ON COLUMN catalogo.productos.categoria IS 'Categoría del producto (ej: Electrónica, Ropa)';
29 COMMENT ON COLUMN catalogo.productos.precio IS 'Precio en formato numérico ';
```

### inserción de datos

```
34 /* ***** Registros de ejemplo ***** */
35 ✓ INSERT INTO catalogo.productos(nombre, categoria, precio)
36 VALUES
37   ('Laptop Gamer ASUS', 'Tecnología', 1800000),
38   ('Libro Python Para Data Science', 'Libros', 100000),
39   ('Memoria USB 64GB', 'Tecnología', 72000),
40   ('Peluche Tux Linux', 'Juguetes', 30000),
41   ('Libro SQL Avanzado', 'Libros', 60000 );
42
43 SELECT * FROM catalogo.productos;
```

## Consulta en la base de datos

Query Query History

```
1 SELECT * FROM catalogo.productos
2 WHERE categoria='Tecnología' :
```

Data Output Messages Notifications

Showing rows: 1 to 2 Page No: 1 of 1

	id_producto [PK] integer	nombre character varying (100)	categoria character varying (30)	precio numeric (10)
1	1	Laptop Gamer ASUS	Tecnología	1800000
2	3	Memoria USB 64GB	Tecnología	72000

## MongoDB

### Configuración MongoDB Compass, MongoDB Server y MongoDBShell en Windows

```
Directory: C:\Program Files\MongoDB\Server\8.0\bin

Mode                LastWriteTime         Length Name
----                -
-a-----         11/27/2024   9:28 PM           1558 InstallCompass.ps1
-a-----         11/27/2024   9:28 PM            496 mongod.cfg
-a-----         11/27/2024  10:54 PM       75501568 mongod.exe
-a-----         11/27/2024  10:54 PM     1231753216 mongod.pdb
-a-----         11/27/2024  10:53 PM       48079872 mongos.exe
-a-----         11/27/2024  10:53 PM     802304000 mongos.pdb

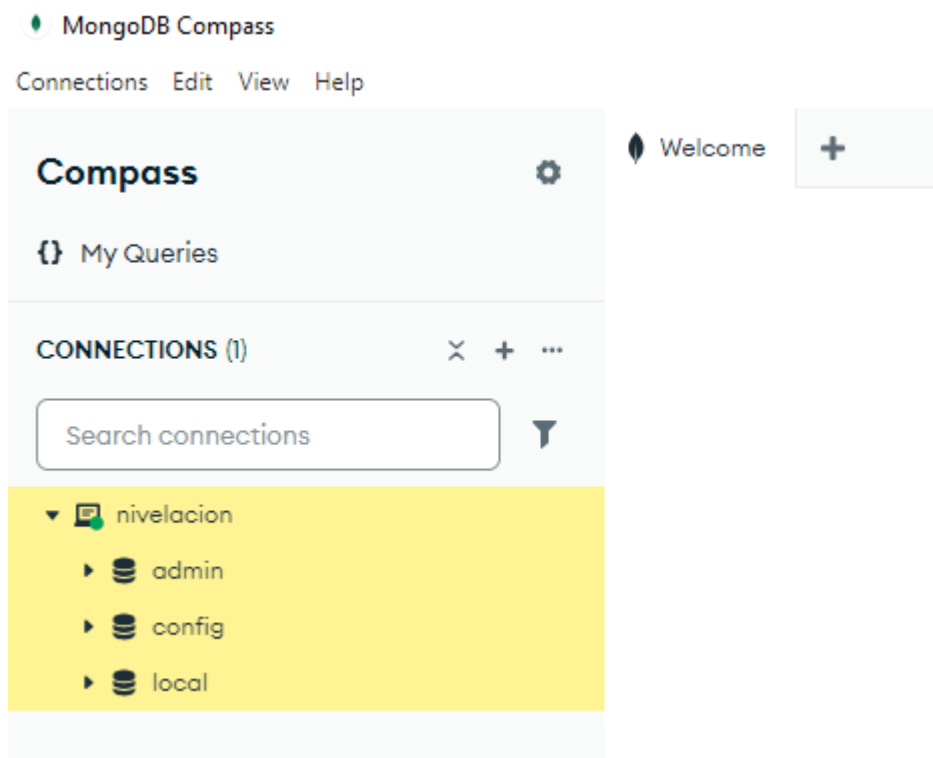
PS C:\Program Files\MongoDB\Server\8.0\bin> ./mongod.exe
{"t":{"$date":"2025-02-02T15:54:05.613-05:00"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"thread1","msg":"Automatically
rce-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2025-02-02T15:54:05.618-05:00"},"s":"I", "c":"CONTROL", "id":5945603, "ctx":"thread1","msg":"Multi threadin
{"t":{"$date":"2025-02-02T15:54:05.618-05:00"},"s":"I", "c":"NETWORK", "id":4648601, "ctx":"thread1","msg":"Implicit TCP F
TCP FastOpen is required, set at least one of the related parameters","attr":{"relatedParameters":["tcpFastOpenServer","tcpF
penQueueSize"]}}
{"t":{"$date":"2025-02-02T15:54:05.619-05:00"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"thread1","msg":"Initialized wi
:{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":25},"incomingInternalClient":{"minWireVersion":0,"ma
ing":{"minWireVersion":6,"maxWireVersion":25},"isInternalClient":true}}}}
{"t":{"$date":"2025-02-02T15:54:05.623-05:00"},"s":"I", "c":"TENANT_M", "id":7091600, "ctx":"thread1","msg":"Starting Tenar
egistry"}
{"t":{"$date":"2025-02-02T15:54:05.623-05:00"},"s":"I", "c":"CONTROL", "id":4615611, "ctx":"","msg":"MongoDB
3952,"port":27017,"dbPath":"/data/db","architecture":"64-bit","host":"DESKTOP-2DNJULT"}}
```

```
Windows PowerShell  mongosh mongodb://127.0.0.1
PS C:\Program Files\MongoDB\mongosh-2.3.8-win32-x64\bin> ./mongosh.exe
Current Mongosh Log ID: 679fe159274c1c8f2ecb0ce1
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.8
Using MongoDB:      8.0.4
Using Mongosh:      2.3.8

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

-----
The server generated these startup warnings when booting
2025-02-02T15:54:05.744-05:00: Access control is not enabled for the database. Read and write access to data and configura
2025-02-02T15:54:05.745-05:00: This server is bound to localhost. Remote systems will be unable to connect to this server.
to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behav
nd_ip 127.0.0.1 to disable this warning
-----

test> show dbs
admin    40.00 KiB
config  108.00 KiB
local   40.00 KiB
test> |
```



### Colección 'ventas' y los 3 registros

```
test> use db_nivelacion
switched to db db_nivelacion
db_nivelacion> show collections
ventas
db_nivelacion> db.ventas.insertOne({ id_venta: 1, fecha: new Date("2024-02-02"), id_producto: "ABC123", cantidad: 3 });
{
  acknowledged: true,
  insertedId: ObjectId('679ff36b274c1c8f2ecb0ce5')
}
db_nivelacion> db.ventas.insertOne({ id_venta: 2, fecha: new Date("2024-02-02"), id_producto: "DEF456", cantidad: 5 });
{
  acknowledged: true,
  insertedId: ObjectId('679ff377274c1c8f2ecb0ce6')
}
db_nivelacion> db.ventas.insertOne({ id_venta: 3, fecha: new Date("2024-02-03"), id_producto: "ABC123", cantidad: 1 });
{
  acknowledged: true,
  insertedId: ObjectId('679ff391274c1c8f2ecb0ce7')
}
```

```
db_nivelacion> db.ventas.find()
[
  {
    _id: ObjectId('679ff36b274c1c8f2ecb0ce5'),
    id_venta: 1,
    fecha: ISODate('2024-02-02T00:00:00.000Z'),
    id_producto: 'ABC123',
    cantidad: 3
  },
  {
    _id: ObjectId('679ff377274c1c8f2ecb0ce6'),
    id_venta: 2,
    fecha: ISODate('2024-02-02T00:00:00.000Z'),
    id_producto: 'DEF456',
    cantidad: 5
  },
  {
    _id: ObjectId('679ff391274c1c8f2ecb0ce7'),
    id_venta: 3,
    fecha: ISODate('2024-02-03T00:00:00.000Z'),
    id_producto: 'ABC123',
    cantidad: 1
  }
]
db_nivelacion> |
```

Consulta para calcular el total de productos vendidos en una fecha especifica

```
Windows PowerShell x mongosh mongodb://127.0.0.1 x + v - □ x
db_nivelacion> db.ventas.aggregate([ { $match: { fecha: { $gte: ISODate("2024-02-02"), $lt: ISODate("2024-02-03") } } }, { $group: { _id: null, total_vtotal_vendido: { $sum: "$cantidad" } } } ] );
[ { _id: null, total_vendido: 8 } ]
db_nivelacion> |
```

Explicación

- 1. **\$match**: Filtra las ventas por fechas especificas:
  - **\$gte: ISODate("2024-02-02")**: Fecha mayor o igual al inicio del día (00:00:00).
  - **\$lt: ISODate("2024-02-03")**: Fecha menor al inicio del día siguiente (para cubrir hasta 23:59:59).
- 2. **\$group**: Agrupa todos los documentos filtrados (**\_id: null**) se suman las cantidades y retorna el total.

+ ▾

🔗 ▾

✎

🗑

25 ▾ 1 - 3 of 3 ↺ < > ▾ ⋮ {} 🏠

🏠 ventas

	_id ObjectId	id_venta Int32	fecha Date	id_prodi	
1	ObjectId('679ff36b274c1c...	1	2024-02-02T00:00:00.000+...	"ABC123"	✎ 🗑 📄 🗑
2	ObjectId('679ff377274c1c...	2	2024-02-02T00:00:00.000+...	"DEF456"	✎ 🗑 📄 🗑
3	ObjectId('679ff391274c1c...	3	2024-02-03T00:00:00.000+...	"ABC123"	✎ 🗑 📄 🗑