



Omnis Developer Conference

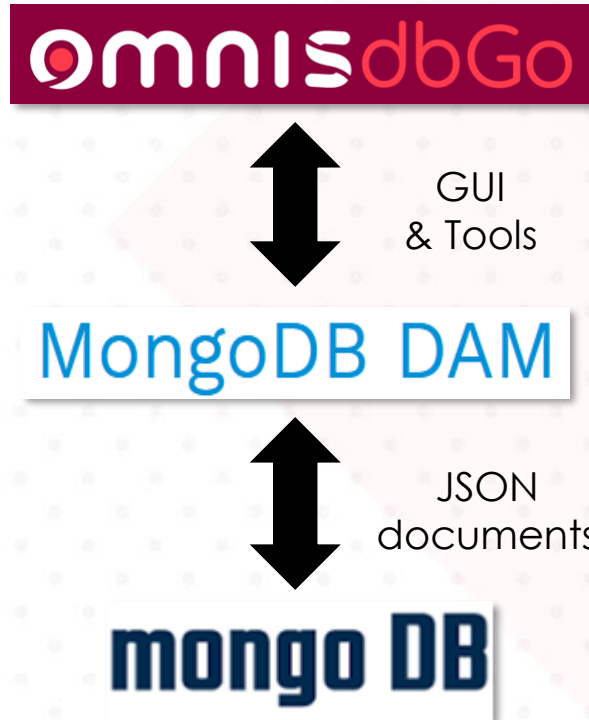
Simplify. Visualize. Build.

**Introduction of Omnis dbGo – the
fast lane to MongoDB development**

Götz Krija

Omnis dbGo

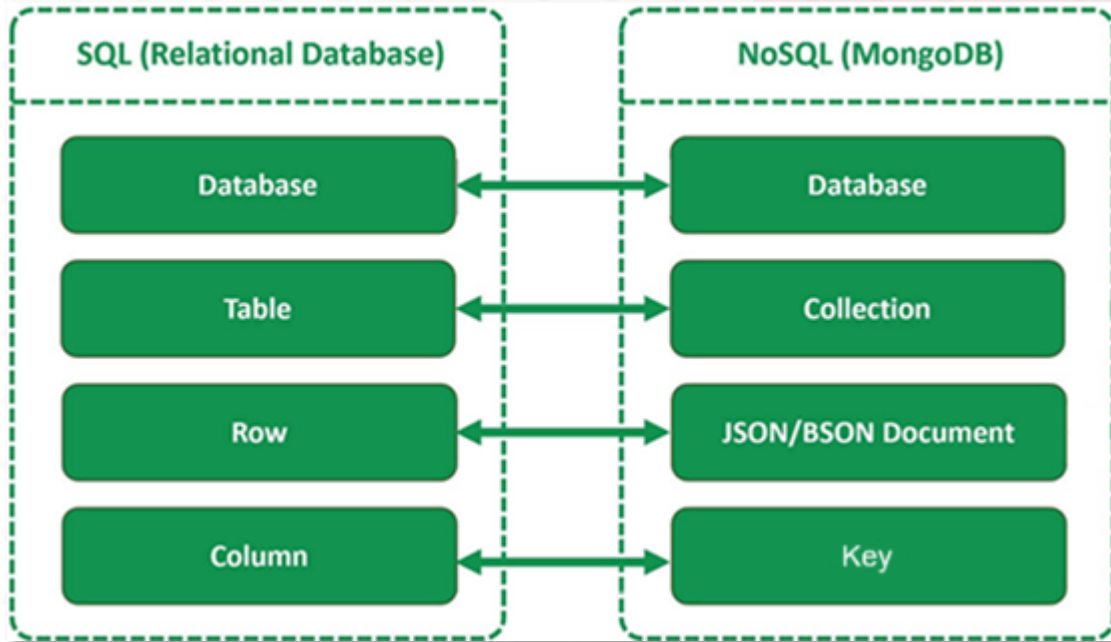
What is dbGo?



- Powerful tool built on top of the **MongoDB DAM** (NoSQL)
- Provides additional features and **GUI with visual tools** for managing MongoDB databases and for creating queries
- Allows you to quickly and easily **create web and desktop applications** based on your MongoDB data and queries

RDBMS vs MongoDB (NoSQL)

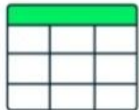
Data Model and Terminology



- Relational DBs store data in Tables (multiple rows)
- MongoDB stores data in Collections (multiple JSON documents)
- Row in a table is equivalent to a JSON document in a MongoDB collection
- A column in a table is comparable to the key of a key-value pair in a JSON document.

RDBMS vs MongoDB (NoSQL)

Tables vs Collections



Relational
Database

User table

ID	first_name	last_name	cell	city
1	Leslie	Yepp	8125552344	Pawnee

Hobbies table

ID	user_id	hobby
10	1	scrapbooking
11	1	eating waffles
12	1	working



MongoDB

User Collection (JSON Document)

```
{
  "_id": 1,
  "first_name": "Leslie",
  "last_name": "Yepp",
  "cell": "8125552344",
  "city": "Pawnee",
  "hobbies": ["scrapbooking", "eating waffles", "working"]
}
```

- Relational DBs store data in table columns and rows
- MongoDB stores data as key/value pairs in a JSON document
- No need for data normalization
- No need for joins

RDBMS vs MongoDB (NoSQL)

SQL vs MongoDB Query Language(MQL)

SQL

```
SELECT * FROM users WHERE age > 30;
```

```
INSERT INTO users (name, age) VALUES  
( 'Alice', 30 );
```

```
UPDATE users SET age = 31 WHERE name =  
'Alice';
```

```
DELETE FROM users WHERE age < 20;
```

MongoDB

```
db.users.find({ age: { $gt: 30 } })
```

```
db.users.insertOne({ name: "Alice", age: 30 })
```

```
db.users.updateOne({ name: "Alice" }, { $set: {  
age: 31 } })
```

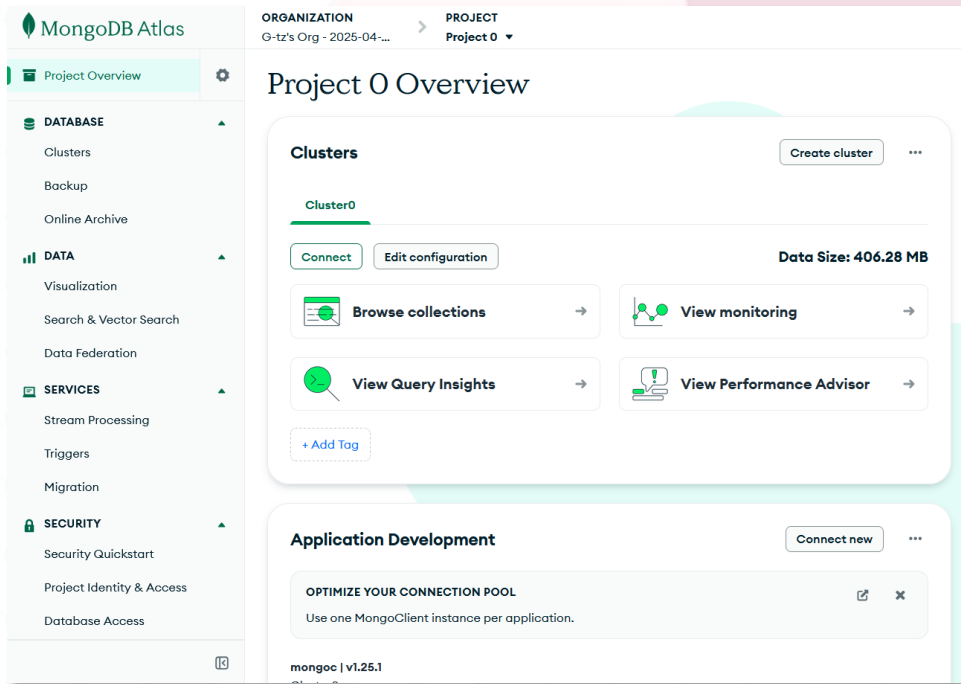
```
db.users.deleteMany({ age: { $lt: 20 } })
```

Using own proprietary query language

Typical query has three main parts:

1. Collection: `db.users`
2. Method: `find()`
3. Condition in JSON format using operators):
(`{ age: { $gt: 30 } }`)

MongoDB Atlas



For a quick start use MongoDB Atlas (cloud solution):

<https://www.mongodb.com/products/platform/cloud>

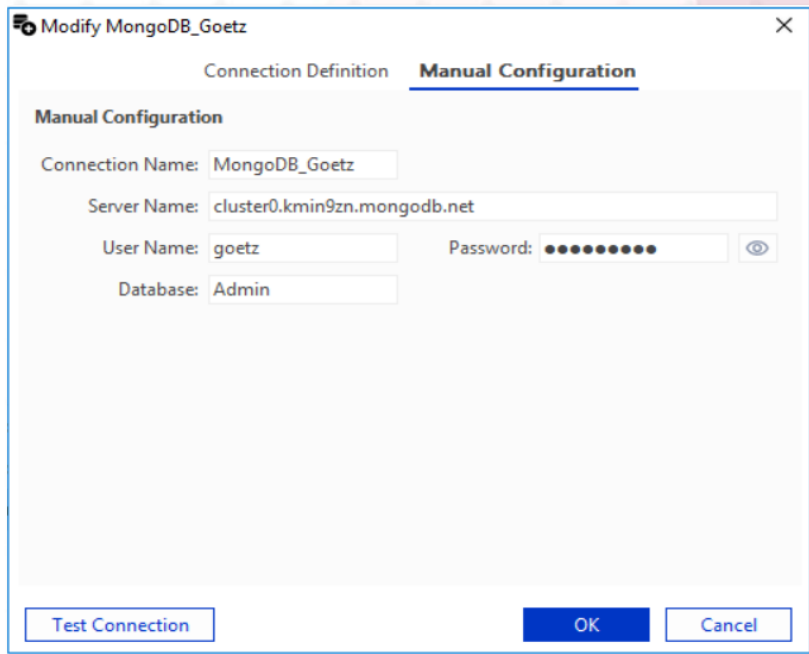
- Simple to sign-up
- Offers a free-tier with example collections and data
- You will have your own Host URI, username and password

Omnis dbGo

Features

Omni dbGo Features

Connection Manager



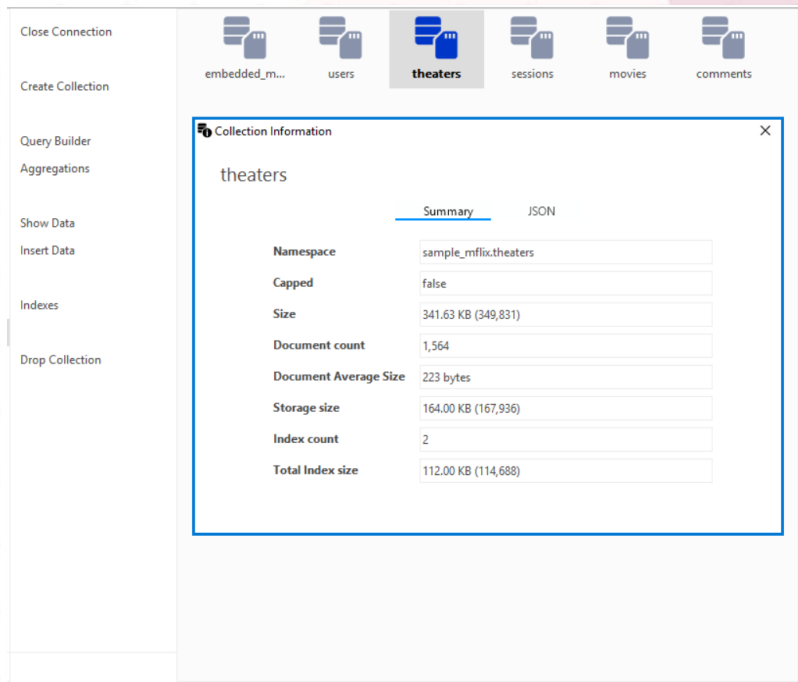
The screenshot shows a dialog box titled "Modify MongoDB_Goetz" with a close button (X) in the top right corner. The dialog has two tabs: "Connection Definition" and "Manual Configuration", with the latter being the active tab. Under the "Manual Configuration" tab, there are four input fields: "Connection Name" (containing "MongoDB_Goetz"), "Server Name" (containing "cluster0.kmin9zn.mongodb.net"), "User Name" (containing "goetz"), and "Password" (containing masked characters "••••••••"). To the right of the password field is an eye icon for toggling visibility. Below these fields is a "Database" field containing "Admin". At the bottom of the dialog, there are three buttons: "Test Connection", "OK", and "Cancel".

Manage database connections

- Setup and open new MongoDB connection
- Open previously defined connections

Omni dbGo Features

dbGo Browser

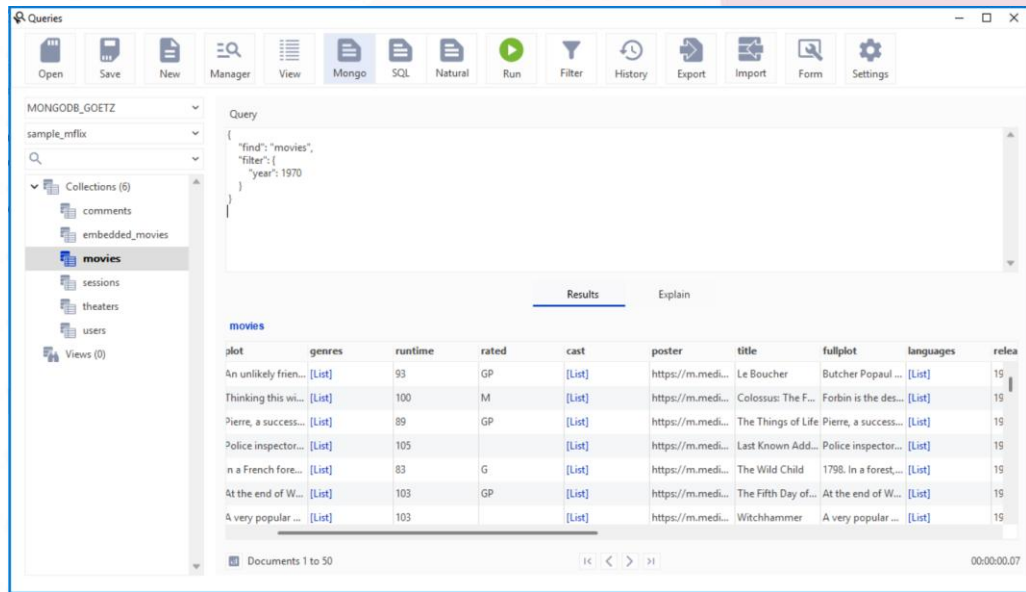


Manage databases, collections and documents

- View databases, collections and documents
- Create & drop Collections
- Insert or import new JSON documents
- Show & update data in documents
- Add, edit & drop indexes

Omni dbGo Features

Query Builder



Querying your data

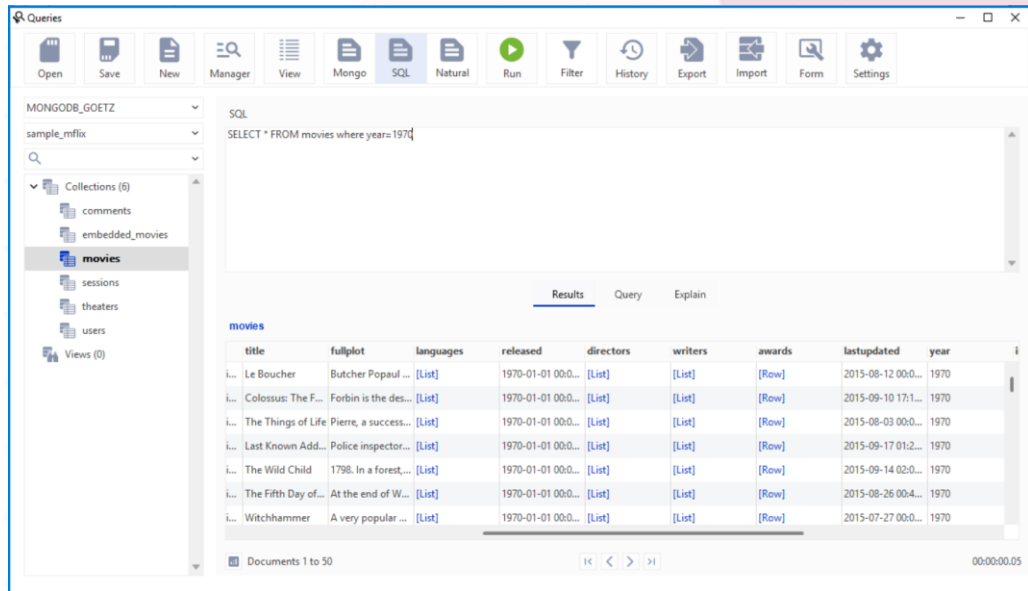
Create and run native MongoDB queries in JSON format, e.g.:

```
{
  "find": "movies",
  "filter": {
    "year": 1970
  }
}
```

Result data is displayed in the window.

Omnis dbGo Features

Query Builder



Querying your data

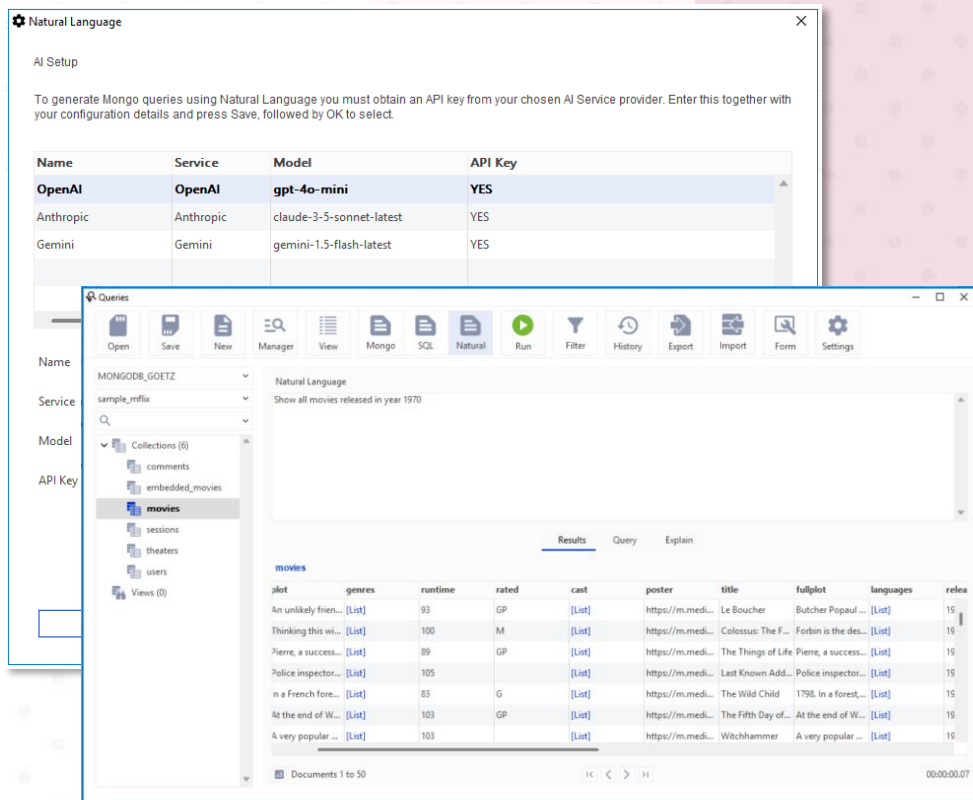
Create standard SQL queries:

`SELECT * FROM movies where year=1970`

Omnis dbGo will “translate” it into a native MongoDB query.

Omni dbGo Features

Query Builder



Querying your data

Create queries in natural language (AI enabled):

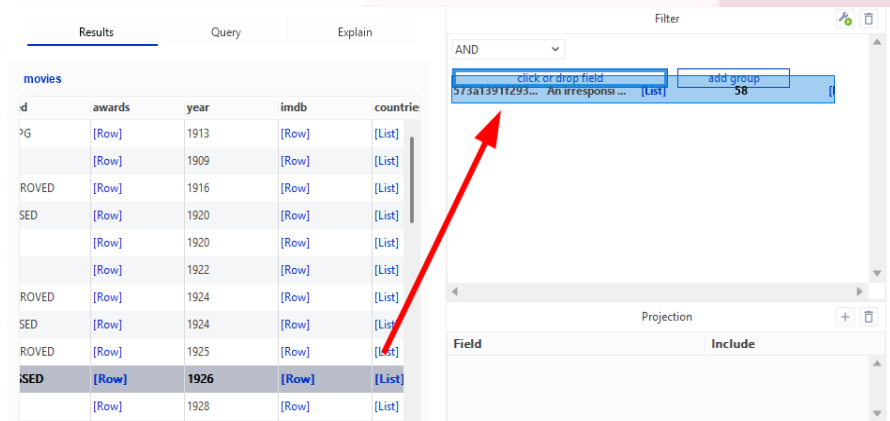
“Show all movies released in year 1970”

Choose your AI provider and add your API key:

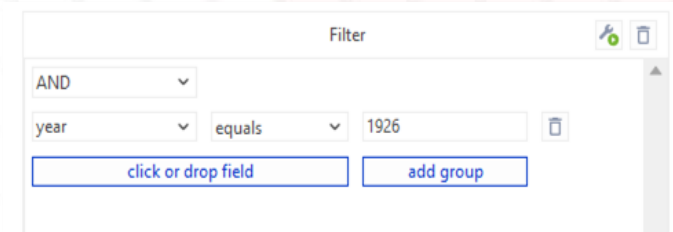
- OpenAI
- Anthropic
- Gemini

Omni dbGo Features

Query Builder



d	awards	year	imdb	country
PG	[Row]	1913	[Row]	[List]
	[Row]	1909	[Row]	[List]
ROVED	[Row]	1916	[Row]	[List]
SED	[Row]	1920	[Row]	[List]
	[Row]	1920	[Row]	[List]
	[Row]	1922	[Row]	[List]
ROVED	[Row]	1924	[Row]	[List]
SED	[Row]	1924	[Row]	[List]
ROVED	[Row]	1925	[Row]	[List]
SED	[Row]	1926	[Row]	[List]
	[Row]	1928	[Row]	[List]



Filter

AND

year equals 1926

click or drop field add group

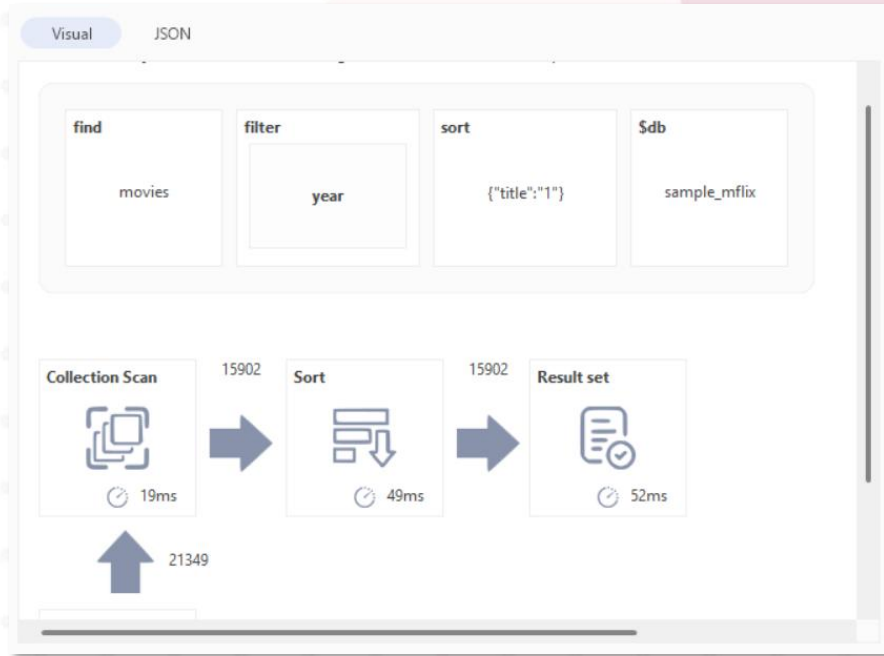
Querying your data

Use visual **Filter Panel** to create queries:

- Add one or more conditions via Drag&drop
- Drag fields from the Results to the Filter panel and choose the operator
- Exclude or include fields from results
- Sort results

Omni dbGo Features

Query Builder



Visualize Queries

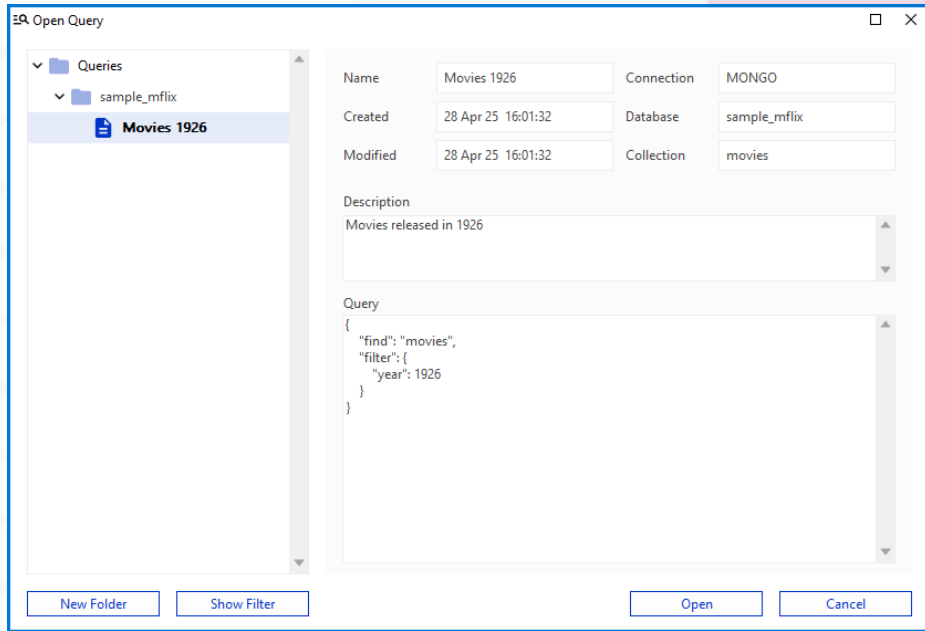
Visual **Explain tool** that shows information about your queries in an image:

- Execution plan (how MongoDB resolves query)
- Performance
- Number of documents scanned and returned
- Which index was used

Helps optimize and improve the performance of your queries

Omni dbGo Features

Query Manager

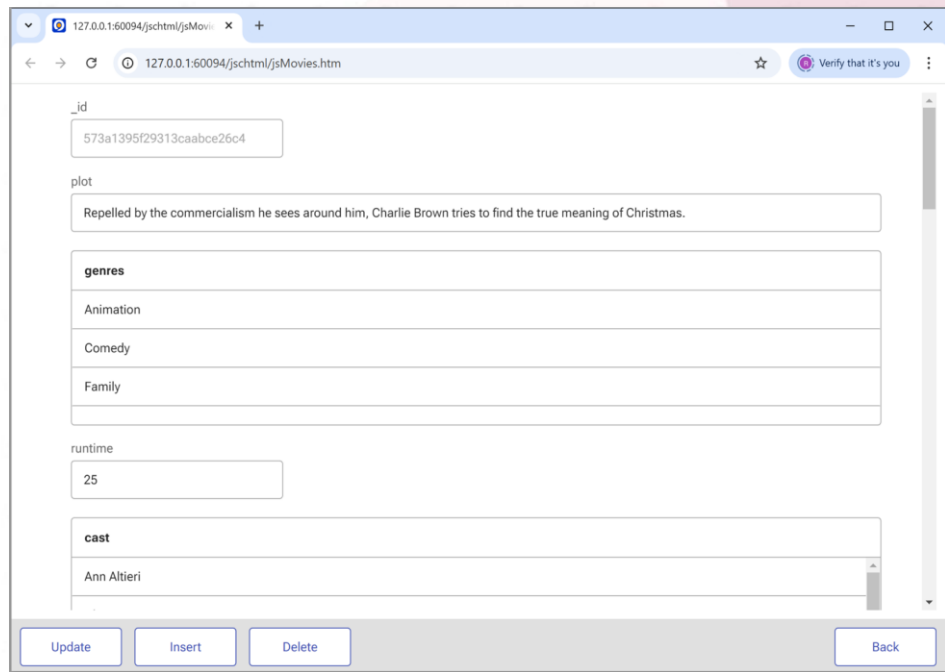


Query Management

- Save queries
- Open previously saved queries
- Search saved queries by keywords
- Export queries in JSON format

Omnis dbGo Features

Form Builder



The screenshot shows a web browser window displaying a form generated by Omnis dbGo. The form is for a movie record and includes the following fields:

- _id**: A text field containing the value "573a1395f29313caabce26c4".
- plot**: A text area containing the text "Repelled by the commercialism he sees around him, Charlie Brown tries to find the true meaning of Christmas."
- genres**: A list of text boxes containing the values "Animation", "Comedy", and "Family".
- runtime**: A text field containing the value "25".
- cast**: A text area containing the text "Ann Altieri".

At the bottom of the form, there are four buttons: "Update", "Insert", "Delete", and "Back".

Generate Forms

- Creates web or desktop application that displays the result data of your queries
- Generates Library with Window or RemoteForm
- Generates the code
- Can be used as basis for further development with Omnis Studio

Omnis dbGo

Versions

dbGo Standard

- Standalone version
- Available for customers without an Omnis Studio 11.2 developer license
- Contains all features, but generated forms cannot be edited or further developed

dbGo Pro

- Add-on for Omnis Studio
- Available to customers with an Omnis Studio 11.2 developer license

Free, time-limited demo version available

Demo



Many thanks