ODC 2025

Omnis Developer Conference

Vereinfachen. Virtualisieren. Entwickeln.

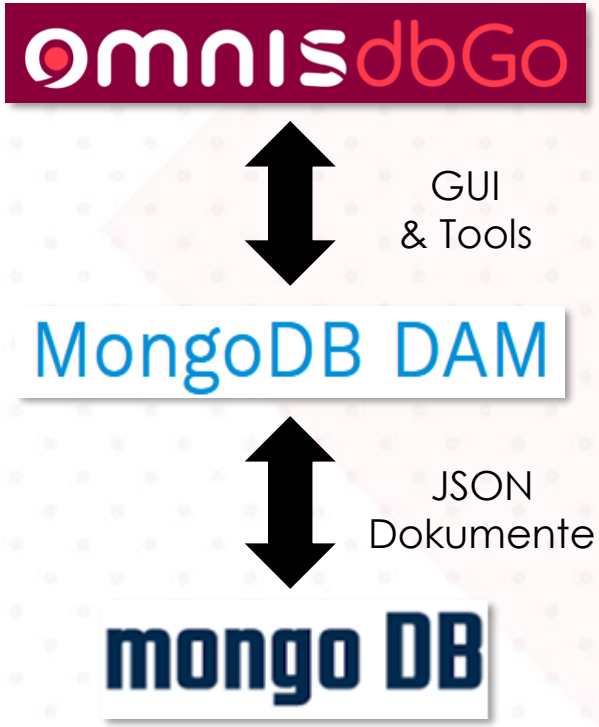
Einführung von Omnis dbGo - die Überholspur zur MongoDB Entwicklung

Götz Krija

OMNIS

Omnis dbGo

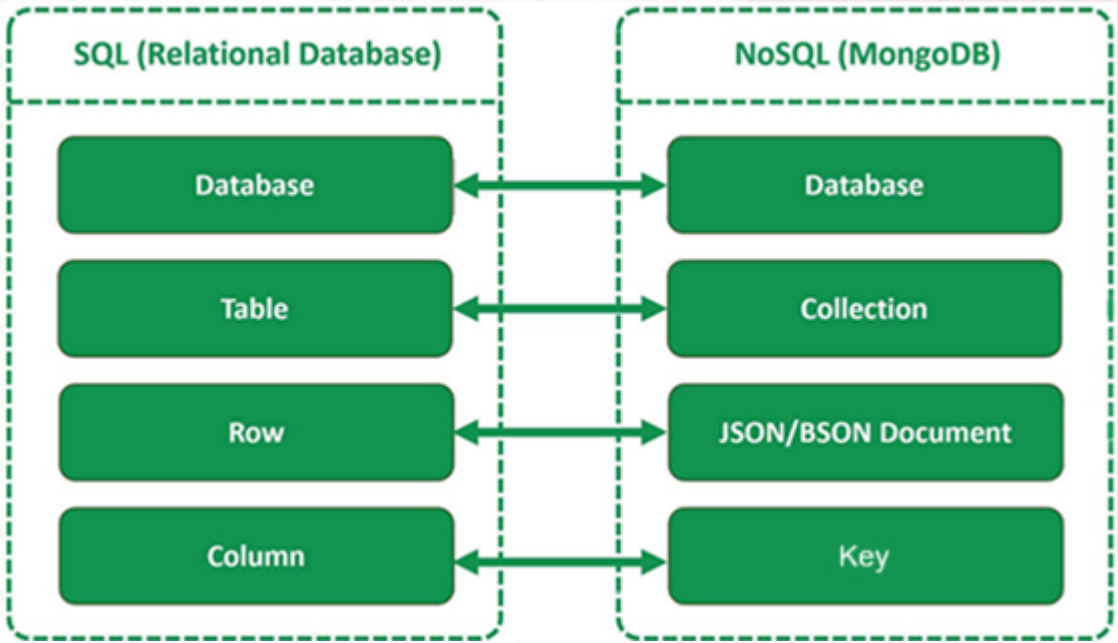
Was ist dbGo?



- Leistungsstarkes Tool auf Basis des **MongoDB DAMs** (NoSQL)
- Bietet zusätzliche Funktionen und **GUI mit visuellen Tools**, die es ermöglichen, MongoDB Datenbanken zu verwalten und Abfragen zu erstellen
- Ermöglicht schnell und einfach **Web- und Desktop Applikationen zu erstellen**, die auf den MongoDB Daten und Abfragen basieren

RDBMS vs MongoDB (NoSQL)

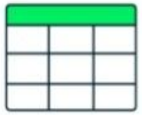
Datenmodell und Terminologie



- Relationale DBs speichern Daten in Tabellen (mehrere Zeilen)
- MongoDB speichert Daten in Collections (mehrere JSON-Dokumente)
- Zeile in einer Tabelle entspricht einem JSON-Dokument in einer MongoDB Collection
- Eine Spalte in einer Tabelle ist vergleichbar mit dem Schlüssel eines Schlüssel-Werte-Paares in einem JSON-Dokument

RDBMS vs MongoDB (NoSQL)

Tabellen vs Collections



Relational Database

MongoDB

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

User Collection (JSON Document)

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

- Relationale DBs speichern Daten in Tabellen mit Spalten und Zeilen
- MongoDB speichert Daten als Schlüssel/Wert-Paare in einem JSON-Dokument
- Keine Notwendigkeit für Daten-Normalisierung
- Keine Notwendigkeit für Joins

RDBMS vs MongoDB (NoSQL)

SQL vs MongoDB Query Language(MQL)

SQL	MongoDB
`SELECT * FROM users WHERE age > 30;`	`db.users.find({ age: { $gt: 30 } })`
`INSERT INTO users (name, age) VALUES ('Alice', 30);`	`db.users.insertOne({ name: "Alice", age: 30 })`
`UPDATE users SET age = 31 WHERE name = 'Alice';`	`db.users.updateOne({ name: "Alice" }, { $set: { age: 31 } })`
`DELETE FROM users WHERE age < 20;`	`db.users.deleteMany({ age: { $lt: 20 } })`

Verwendung einer eigenen proprietären Abfragesprache

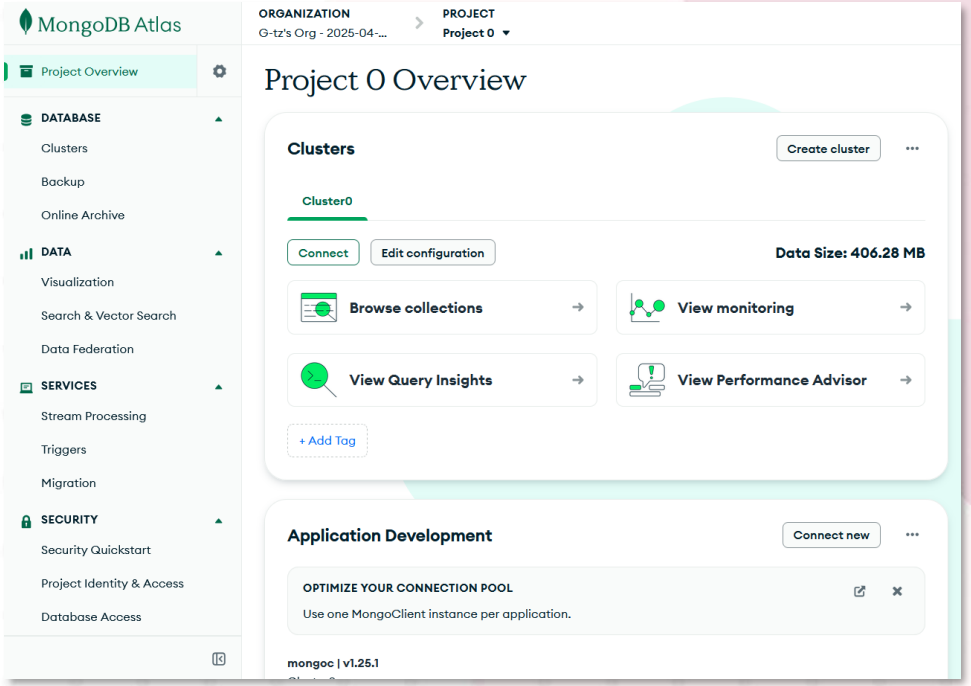
Typische Abfrage besteht aus drei Hauptbestandteilen:

1. Collection: *db.users*
2. Methode: *find()*
3. Bedingung mit Operatoren im JSON-Format:

({ age: { $gt: 30 } })

Omnis dbGo

MongoDB Atlas



Schneller Einstieg mit MongoDB Atlas (Cloud Lösung):

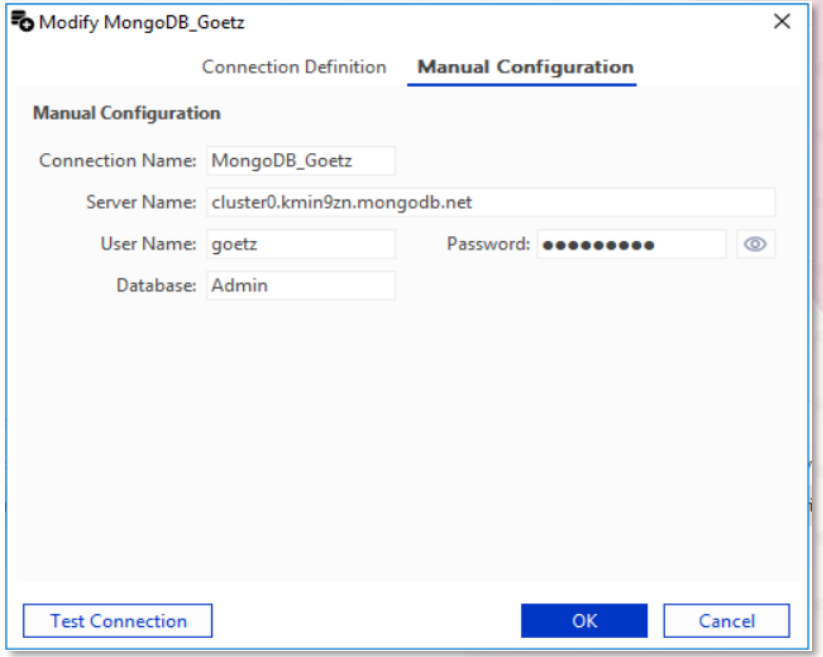
https://www.mongodb.com/products/platform/cloud

- Einfache Anmeldung
- Bietet kostenlose Stufe mit Beispiel-Collections und Daten
- Eigene Host-URI, Benutzernamen und Passwort

Omnis dbGo Features

Omnis dbGo Features

Connection Manager

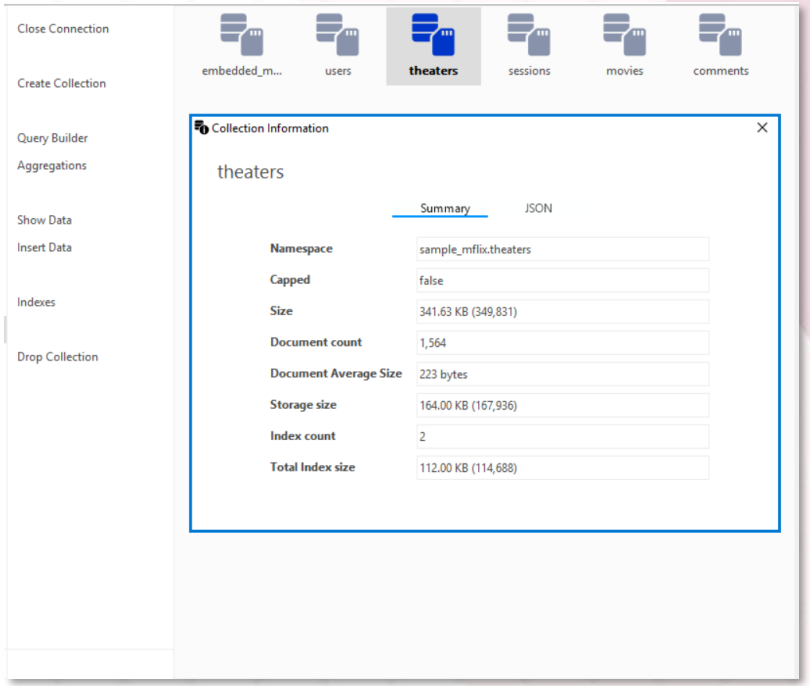


Datenbankverbindungen verwalten

- Neue MongoDB-Verbindungen einrichten und öffnen
- Zuvor angelegte Verbindungen öffnen

Omnis dbGo Features

dbGo Browser

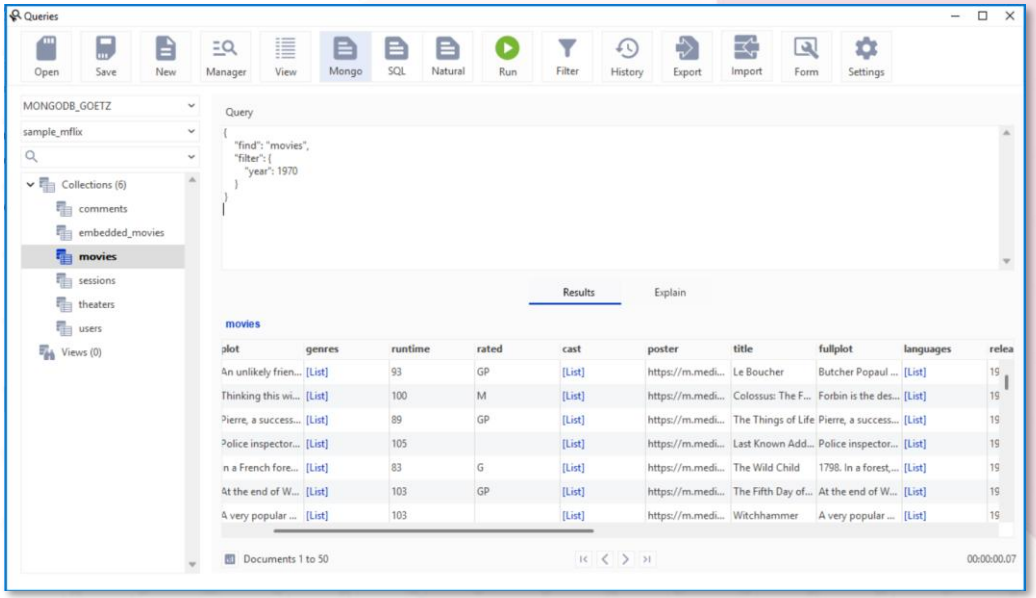


Datenbanken, Collections und Dokumente verwalten

- Datenbanken, Collections und Dokumente anzeigen
- Collections erstellen & löschen
- Neue JSON-Dokumente einfügen oder importieren
- Daten in Dokumenten anzeigen und aktualisieren
- Indizes hinzufügen, bearbeiten und löschen

Omnis dbGo Features

Query Builder



Daten abfragen

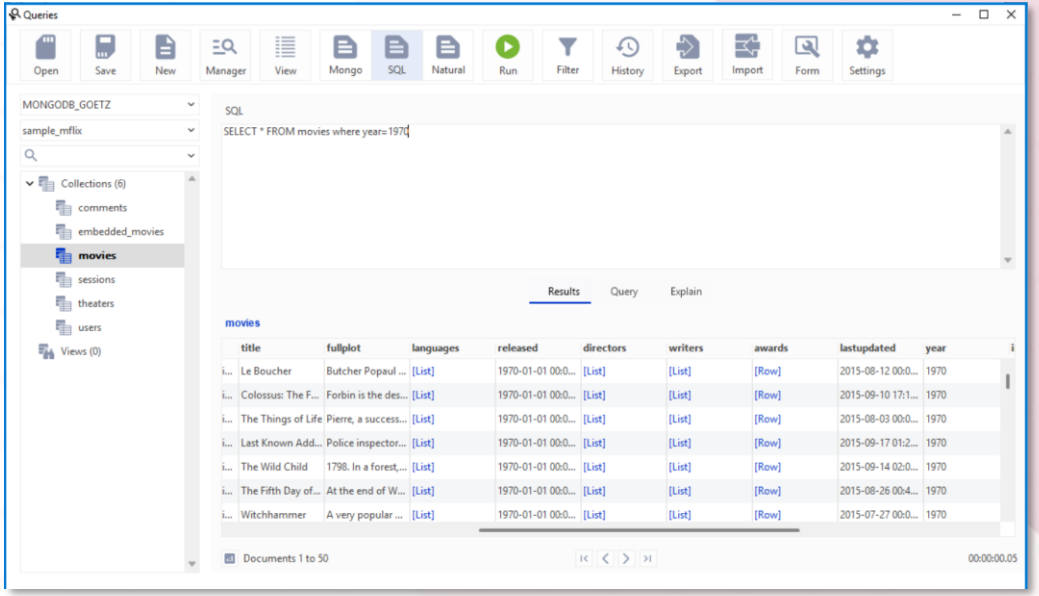
Erstellung und Ausführung nativer MongoDB-Abfragen im JSON-Format, z.B.:

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

Ergebnisdaten werden im Fenster angezeigt.

Omnis dbGo Features

Query Builder



Daten abfragen

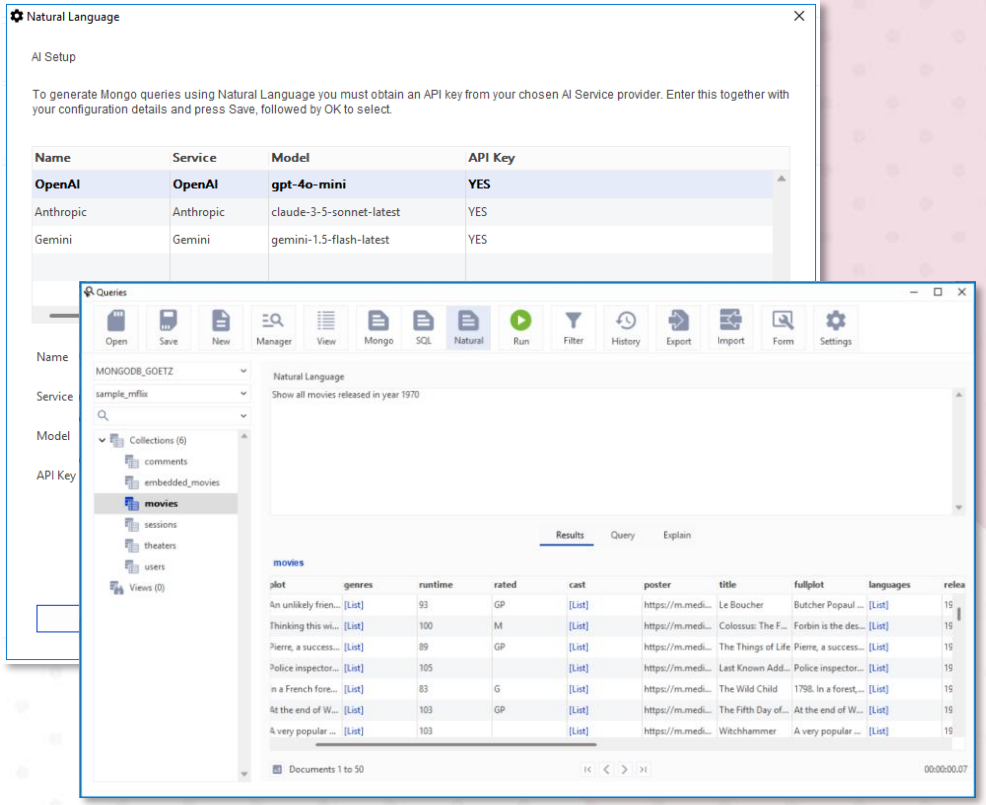
Erstellung von Standard-SQL Abfragen:

SELECT * FROM movies where year=1970

Omnis dbGo “übersetzt” in eine native MongoDB Abfrage

Omnis dbGo Features

Query Builder



Daten abfragen

Erstellung von Abfragen in natürlicher Sprache (KI-gestützt):

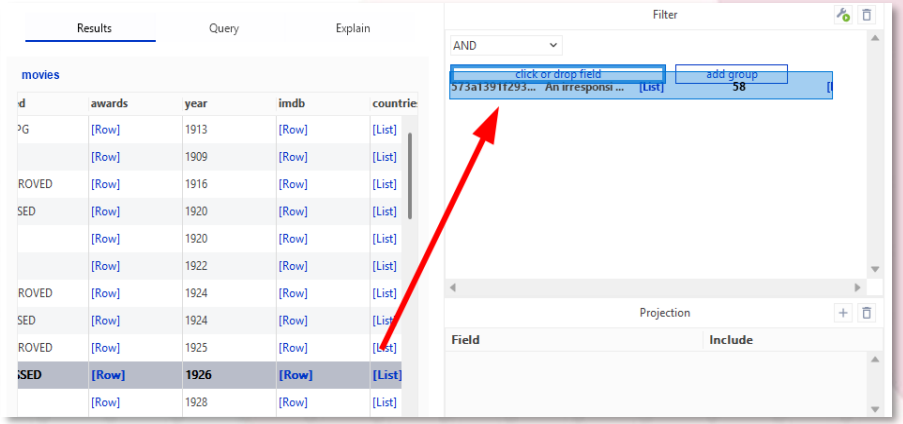
“Show all movies released in year 1970”

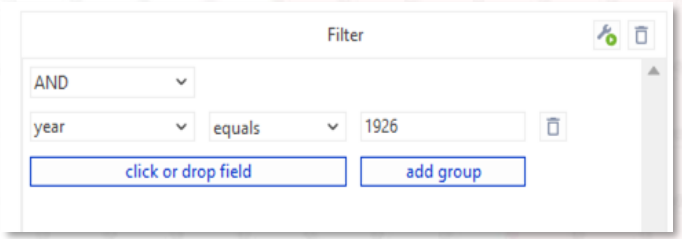
KI-Abieter auswählen und API-Key hinzufügen:

- OpenAI
- Anthrophic
- Gemini

Omnis dbGo Features

Query Builder





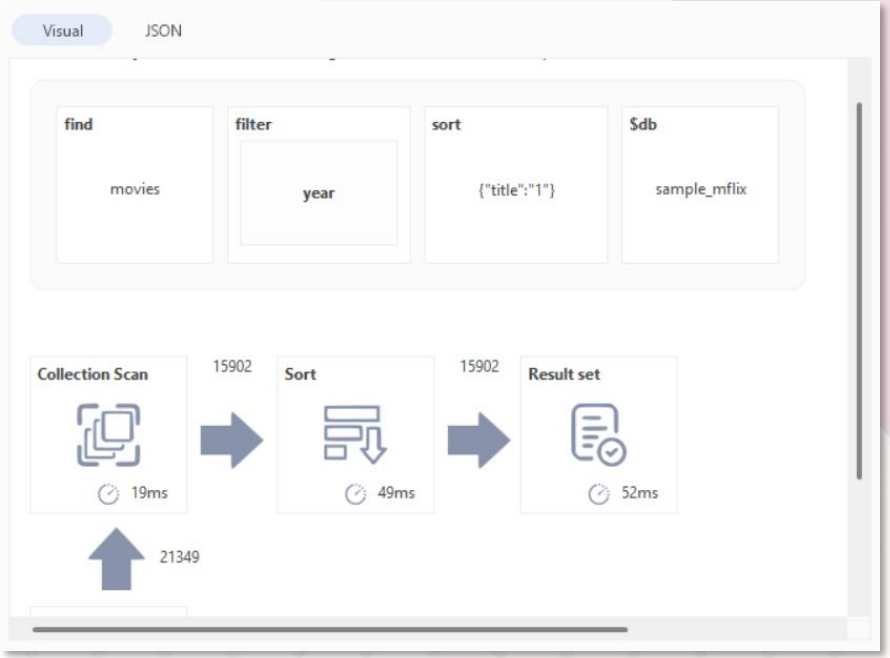
Daten abfragen

Verwendung eines **Filter-Panels** zur Erstellung von Abfragen:

- Hinzufügen einer oder mehrerer Bedingungen per Drag&drop
- Felder aus den Ergebnissen in den Filter-Panel ziehen und den Operator auswählen
- Felder im Ergebis ein- oder ausschließen
- Ergebnis sortieren

Omnis dbGo Features

Query Builder



Abfragen visualisieren

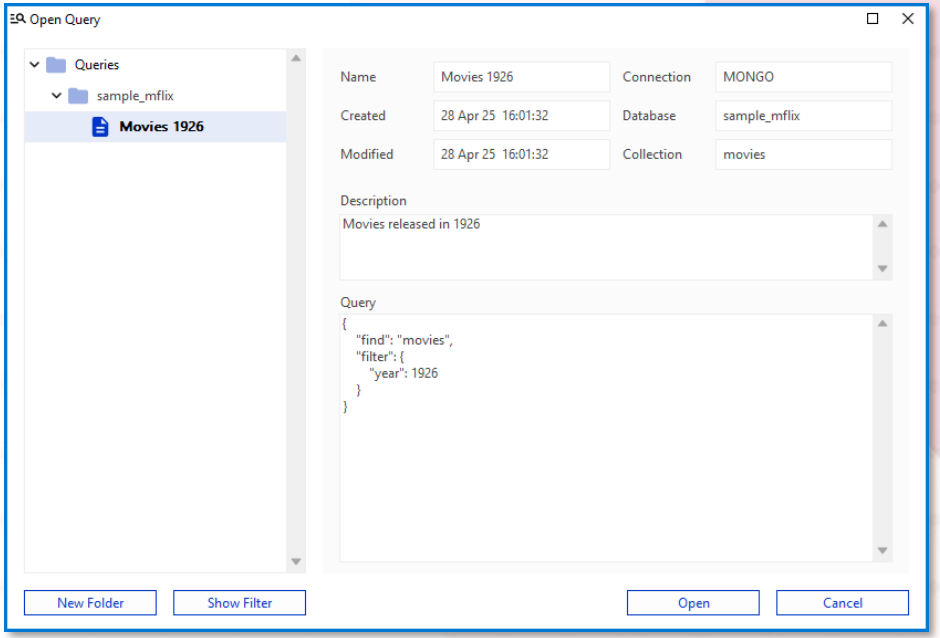
Visuelles **Explain-Tool**, das information zu Abfragen als Bild anzeigt:

- Ausführungsplan (wie MongoDB die Abfrage auflöst)
- Geschwindigkeit
- Anzahl gescannter und zurückgegebener Dokumente
- Verwendeter Index

Hilft dabei, Abfragen zu optimieren und die Performance zu verbessern.

Omnis dbGo Features

Query Manager

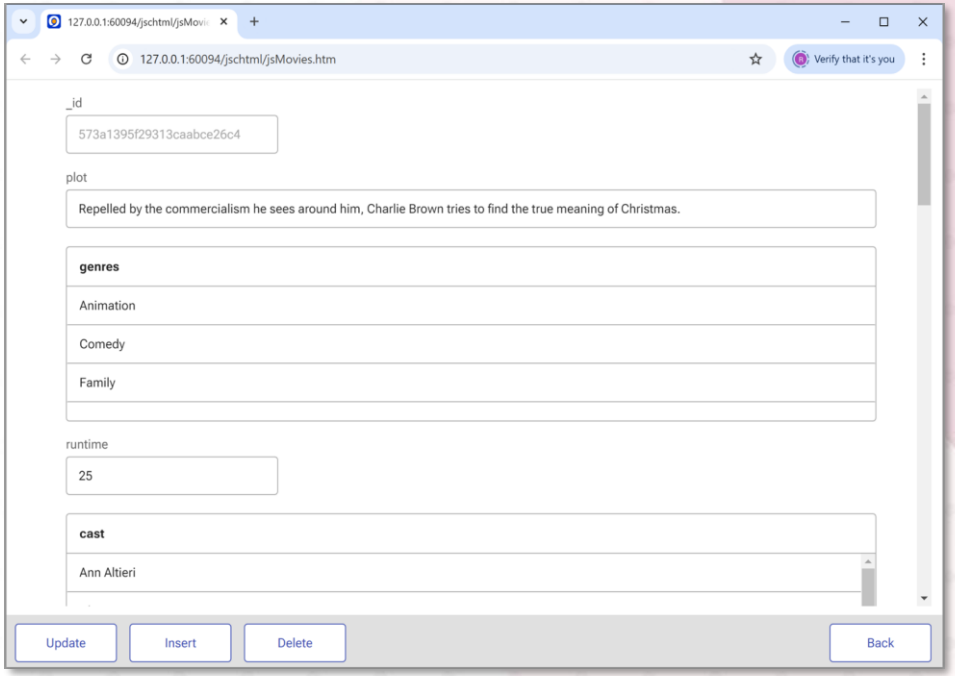


Verwaltung von Abfragen

- Abfragen speichern
- Zuvor gespeicherte Abfragen öffnen
- Gespeicherte Abfragen nach Stichworten durchsuchen
- Abfragen im JSON-Format exportieren

Omnis dbGo Features

Form Builder



Formulare erzeugen

- Erstellt Web- oder Desktop Applikation mit den Ergebnisdaten der Abfragen
- Erzeugt Library mit Window oder RemoteForm
- Generiert den Code
- Kann als Grundlage für weitere Entwicklung mit Omnis Studio genutzt werden

Omnis dbGo

Versionen

dbGo Standard

- Standalone Version
- Für Kunden ohne Omnis Studio 11.2 Entwicklerlizenz
- Enthält alle Features, aber keine Möglichkeit zur Editierung oder Weiterentwicklung der erzeugten Form

dbGo Pro

- Add-on für Omnis Studio
- Für Kunden mit Omnis Studio 11.2 Entwicklerlizenz

Kostenlose, zeitlich begrenzte Demoversion verfügbar

omnisdbGo

Demo

Many thanks