

# Εισαγωγή στην mongoDB®

Προχωρημένα Θέματα Βάσεων Δεδομένων  
τΗΜΜΥ Πανεπιστημίου Δυτικής Μακεδονίας

# Database-Engines Ranking - May 2023

Rank	DBMS	Database Model
1	Oracle	Relational, Multi-model
2	MySQL	Relational, Multi-model
3	Microsoft SQL Server	Relational, Multi-model
4	PostgreSQL	Relational, Multi-model
5	MongoDB	Document, Multi-model

**Πηγή Στατιστικών:**

- <https://db-engines.com/en/ranking>
- <https://statisticsanddata.org/data/the-most-popular-databases-2006-2021/>

# Εισαγωγή στην MongoDB

- ▶ Η MongoDB είναι μια βάση δεδομένων εγγράφων (Document) και ανήκει στην κατηγορία των NoSQL με χαρακτηριστικά της γνωρίσματα την επεκτασιμότητα και την ευελιξία για ερωτήματα (querying) και την δημιουργία ευρετηρίων (indexing).
- ▶ Ιστοσελίδα: <https://www.mongodb.com/>
- ▶ Manual: <https://www.mongodb.com/docs/manual/>

# Παράδειγμα κώδικα Python

```
JS Python Java C++ C#

# Connect to MongoDB instance running on localhost
client = pymongo.MongoClient()

# Access the 'restaurants' collection
# in the 'test' database
collection = client.test.restaurants
```

**Connect**  
Here we are connecting to a locally hosted MongoDB database called test with a collection named restaurants.

Σύνδεση:

*Σε αυτό το σημείο γίνεται η σύνδεση της εφαρμογής μας σε Python με την τοπική ΒΔ MongoDB με όνομα test δίνοντας πρόσβαση στην συλλογή με τίτλο Restaurants*

# Παράδειγμα κώδικα Python

```
JS Python Java C++ C#
```

```
new_documents = [  
  {  
    "name": "Sun Bakery Trattoria",  
    "stars": 4,  
    "categories": ["Pizza","Pasta","Italian","Coffee","Sandwiches"]  
  }, {  
    "name": "Blue Bagels Grill",  
    "stars": 3,  
    "categories": ["Bagels","Cookies","Sandwiches"]  
  }  
]
```

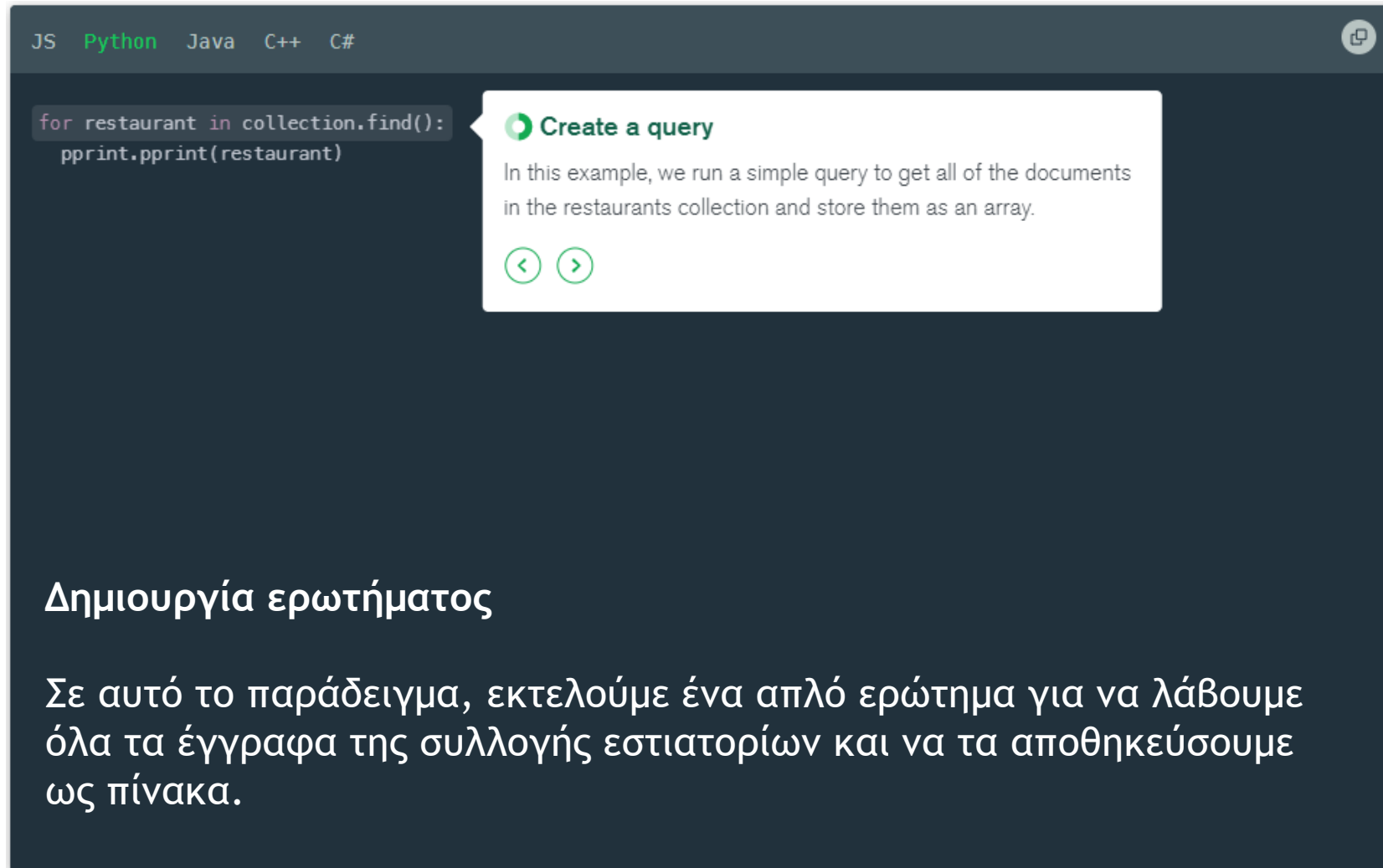
```
collection.insert_many(new_documents)
```

**Insert a document**  
2 documents are being inserted into the restaurants collection.  
Each document represents a restaurant with a name, star rating,  
and categories (stored as an array).

## Εισαγωγή εγγράφου/εγγραφής:

Σε αυτό το στιγμιότυπο εισάγουμε 2 έγγραφα (ή εγγραφές όπως θα λέγαμε στην MySQL) στην συλλογή εστιατόριο. Κάθε έγγραφο αντιπροσωπεύει ένα εστιατόριο με όνομα, βαθμολογία αστεριών και κατηγορίες (αποθηκευμένες ως πίνακας).

# Παράδειγμα κώδικα Python

A screenshot of a code editor interface. At the top, there are tabs for 'JS', 'Python', 'Java', 'C++', and 'C#', with 'Python' selected. The code editor contains the following Python code:

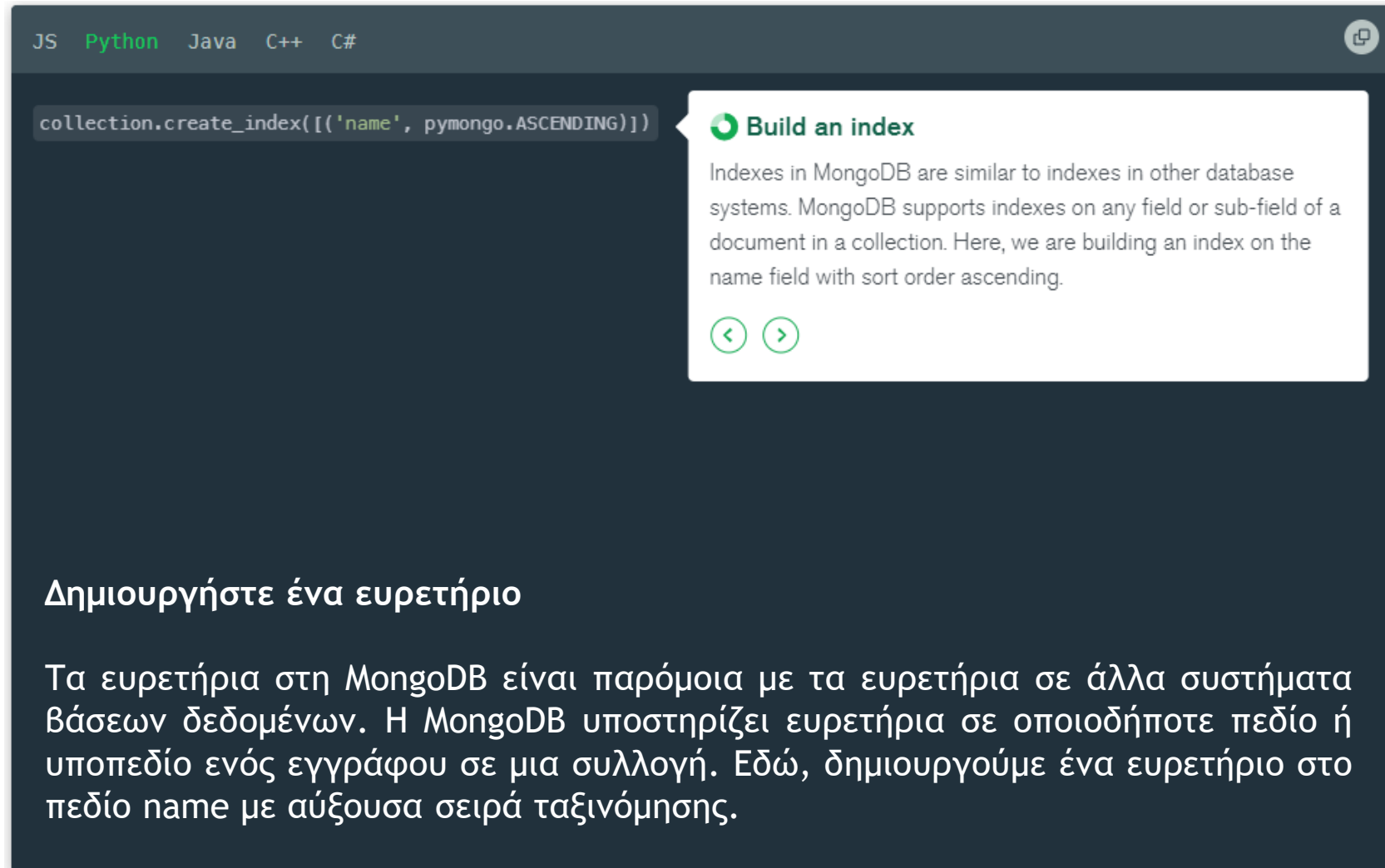
```
for restaurant in collection.find():  
    pprint.pprint(restaurant)
```

A tooltip is displayed over the code, titled 'Create a query'. The tooltip text reads: 'In this example, we run a simple query to get all of the documents in the restaurants collection and store them as an array.' Below the text are two circular navigation arrows, one pointing left and one pointing right. In the top right corner of the editor, there is a copy icon.

## Δημιουργία ερωτήματος

Σε αυτό το παράδειγμα, εκτελούμε ένα απλό ερώτημα για να λάβουμε όλα τα έγγραφα της συλλογής εστιατορίων και να τα αποθηκεύσουμε ως πίνακα.

# Παράδειγμα κώδικα Python



```
JS Python Java C++ C#  
collection.create_index([{'name', pymongo.ASCENDING}])
```

**Build an index**  
Indexes in MongoDB are similar to indexes in other database systems. MongoDB supports indexes on any field or sub-field of a document in a collection. Here, we are building an index on the name field with sort order ascending.

**Δημιουργήστε ένα ευρετήριο**

Τα ευρετήρια στη MongoDB είναι παρόμοια με τα ευρετήρια σε άλλα συστήματα βάσεων δεδομένων. Η MongoDB υποστηρίζει ευρετήρια σε οποιοδήποτε πεδίο ή υποπεδίο ενός εγγράφου σε μια συλλογή. Εδώ, δημιουργούμε ένα ευρετήριο στο πεδίο name με αύξουσα σειρά ταξινόμησης.

# Παράδειγμα κώδικα Python

- Χρησιμοποιώντας ένα aggregation pipeline μπορείτε να φιλτράρετε και να αναλύετε δεδομένα με βάση ένα συγκεκριμένο σύνολο κριτηρίων.
- Σε αυτό το παράδειγμα, τραβάμε όλα τα έγγραφα στη συλλογή εστιατορίων που έχουν την κατηγορία Bakery χρησιμοποιώντας τον τελεστή \$match και στη συνέχεια τα ομαδοποιούμε με βάση τη βαθμολογία των αστεριών τους χρησιμοποιώντας τον τελεστή \$group.
- Χρησιμοποιώντας τον τελεστή συσσώρευσης, \$sum, μπορούμε να δούμε πόσα αρτοποιεία στη συλλογή μας έχουν κάθε βαθμολογία αστέρων.

JS Python Java C++ C#

```
pipeline = [  
  {"$match": {"categories": "Bakery"}},  
  {"$group": {"_id": "$stars", "count": {"$sum": 1}}}  
]
```

```
pprint.pprint(list(collection.aggregate(pipeline)))
```

## Aggregate

Using MongoDB's aggregation pipeline, you can filter and analyze data based on a given set of criteria. In this example, we pull all the documents in the restaurants collection that have a category of Bakery using the \$match operator and then group them by their star rating using the \$group operator. Using the accumulator operator, \$sum, we can see how many bakeries in our collection have each star rating.



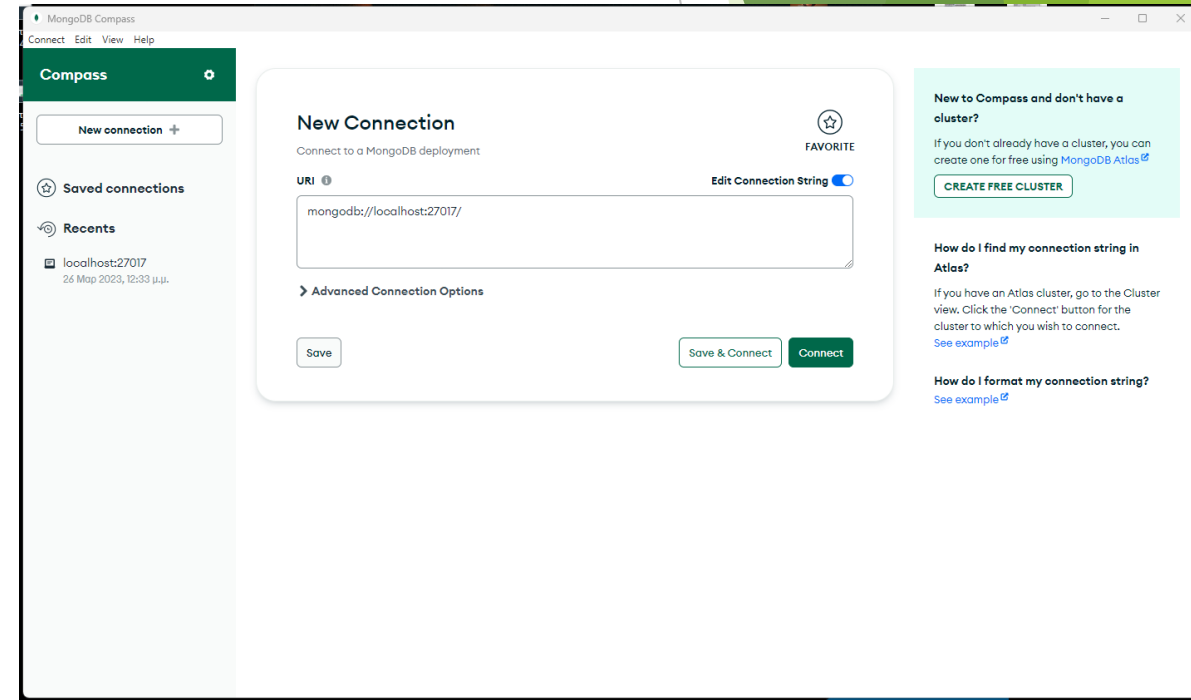
[Start Free with MongoDB Atlas](#)



# Εργαλείο Ανάπτυξης - Compass

Εγκατάσταση MongoDB και Compass:

- <https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-windows/>
- <https://www.youtube.com/watch?v=gDOKSgqM-bQ&list=PL4cUxeGkcC9h77dJ-QJlwGlZlTd4ecZOA&index=2>

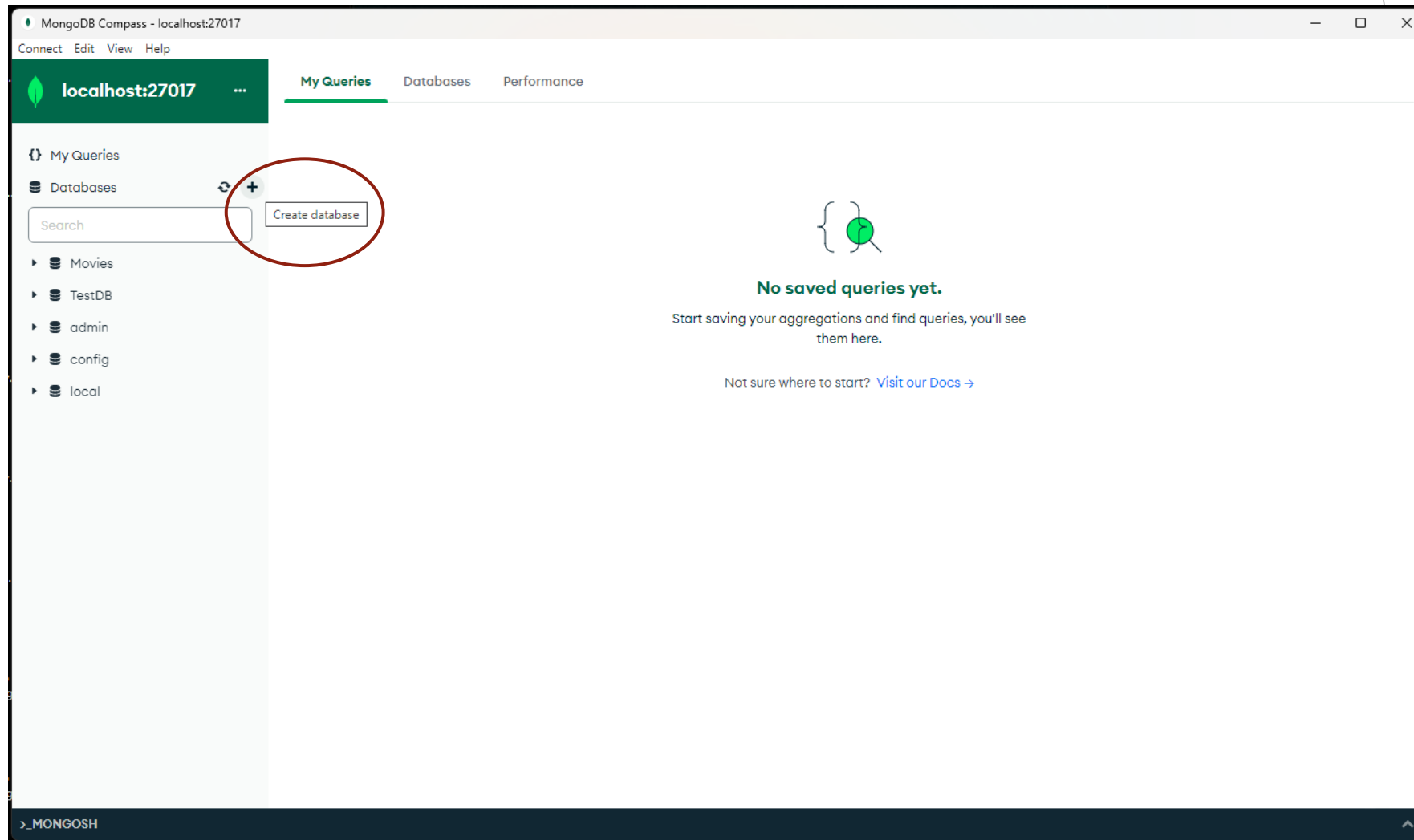


# Περιβάλλον Compass

The screenshot shows the MongoDB Compass interface. On the left, there is a sidebar with a 'Compass' header and a 'New connection +' button. Below this, there are sections for 'Saved connections' and 'Recents', each listing a connection to 'localhost:27017'. The main area is titled 'New Connection' and contains a text input field with the URI 'mongodb://localhost:27017/'. To the right of the input field is a toggle for 'Edit Connection String'. Below the input field is a section for 'Advanced Connection Options'. At the bottom of the dialog, there are two buttons: 'Save' and 'Connect'. The 'Connect' button is highlighted with a red circle. On the right side of the dialog, there are three informational panels: 'New to Compass and don't have a cluster?' with a 'CREATE FREE CLUSTER' button, 'How do I find my connection string in Atlas?', and 'How do I format my connection string?'.

Σύνδεση με την Mongo DB

# Περιβάλλον Compass



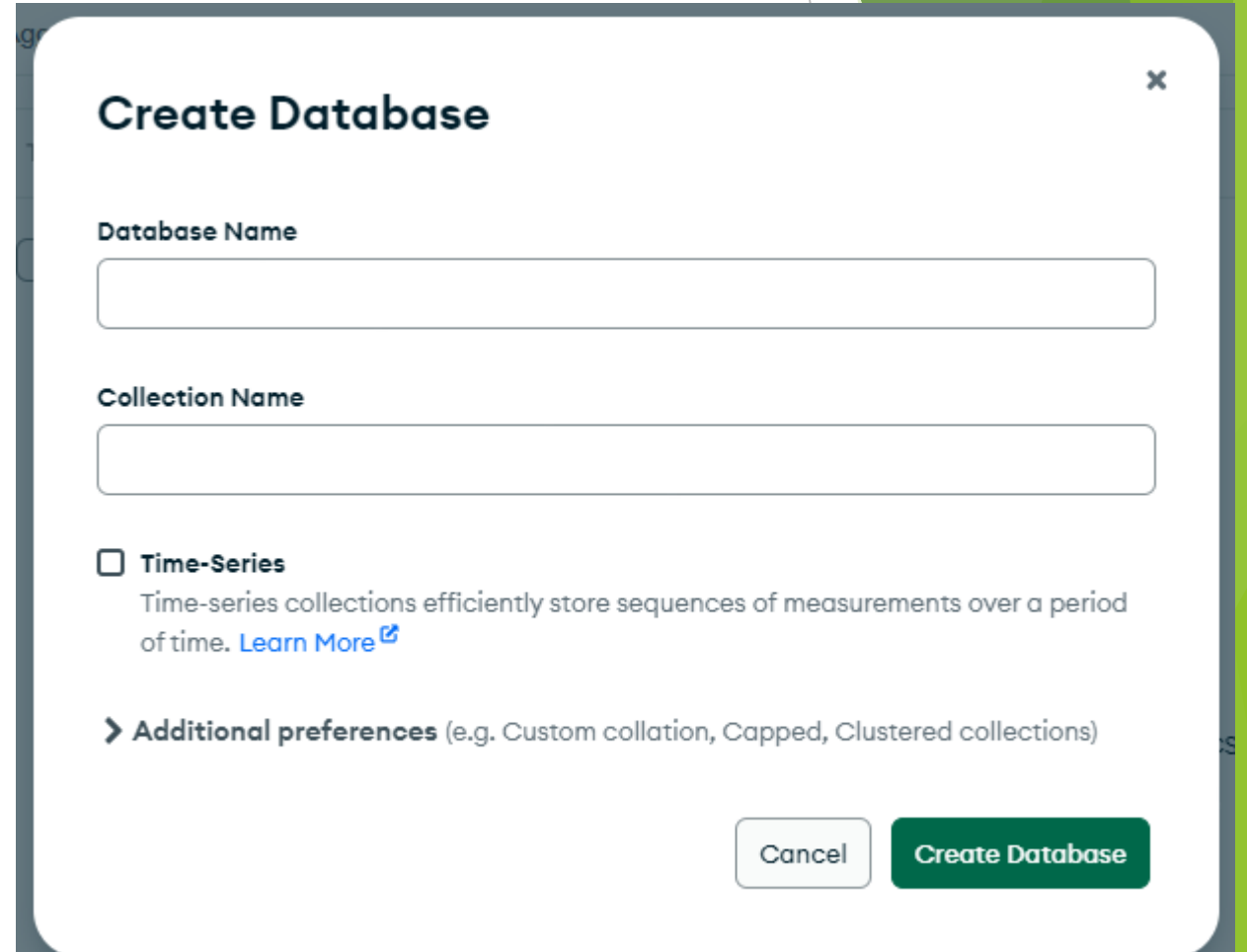
Δημιουργία Βάσης Δεδομένων

# Δημιουργία Βάσης Δεδομένων

Για το σημερινό μάθημα θα φτιάξουμε μια ΒΔ

**Database Name:** bookstore

**Collection Name (αντίστοιχο του πίνακα):** books



The screenshot shows a 'Create Database' dialog box with the following elements:

- Title:** Create Database (with a close button 'x' in the top right corner)
- Database Name:** A text input field.
- Collection Name:** A text input field.
- Time-Series:** A checkbox that is currently unchecked. Below it is the text: "Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)".
- Additional preferences:** A section header with a right-pointing chevron, followed by the text "(e.g. Custom collation, Capped, Clustered collections)".
- Buttons:** A 'Cancel' button and a 'Create Database' button (highlighted in dark green).

# Εισαγωγή Δεδομένων

```
INSERT INTO books (title, author, pages, genre)
VALUES ('Introduction to MySQL', 'Christos Hitiris', 500, 'informatics');
```

## Insert Document

To collection TestDB.TestDB

VIEW  

```
1  /**
2  * Paste one or more documents here
3  */
4  {
5    "title": "Introduction to MongoDB",
6    "author": "Christos Hitiris",
7    "pages": 500,
8    "genres": [
9      "informatics"
10 ]
11 }
```

Cancel

Insert

# Εισαγωγή περισσότερων δεδομένων

Στον παρακάτω σύνδεσμο θα βρείτε το JSON File που περιέχει 10 εγγραφές για τα books:  
<https://gist.github.com/Tilemaxos46/79b7c46316f21609016f3c1d52fb9508>

```
[
  {
    "title": "Ο Ξένος",
    "author": "Albert Camus",
    "pages": 123,
    "genre": "Μυθιστόρημα"
  },
  {
    "title": "Η Αλχημία",
    "author": "Paulo Coelho",
    "pages": 208,
    "genre": "Μυθιστόρημα"
  },
  {
    "title": "Ο Αρχοντας των Δαχτυλιδιών",
    "author": "J.R.R. Tolkien",
    "pages": 1178,
    "genre": "Φαντασίας"
  },
  {
    "title": "Η Κυρία Νταλογκ",
    "author": "Gabriel Garcia Marquez",
    "pages": 368,
    "genre": "Μαγικός Ρεαλισμός"
  },
  {
    "title": "Το Πείραμα",
    "author": "John Grisham",
    "pages": 432,
    "genre": "Θρiller"
  }
]
```

# Καρτέλα Documents

MongoDB Compass - localhost:27017/TestDB.books

localhost:27017 Documents TestDB.books

My Queries Databases Search

TestDB books

TestDB.books 15 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Filter Type a query: { field: 'value' } Reset Find More Options

ADD DATA EXPORT DATA

1 - 15 of 15

<pre>{ "_id": ObjectId('6475edb75cc9f0ba3fa0eae8'), "title": "Ο Ξένος", "author": "Albert Camus", "pages": 123, "genre": "Μυθιστόρημα" }</pre>
<pre>{ "_id": ObjectId('6475edb75cc9f0ba3fa0eae9'), "title": "Η Αλχημία", "author": "Paulo Coelho", "pages": 268, "genre": "Μυθιστόρημα" }</pre>
<pre>{ "_id": ObjectId('6475edb75cc9f0ba3fa0eaea'), "title": "Ο Αρχοντας των Δαχτυλιδιών", "author": "J.R.R. Tolkien", "pages": 1178, "genre": "Φαντασίας" }</pre>
<pre>{ "_id": ObjectId('6475edb75cc9f0ba3fa0eaeab'), "title": "Η Κυρία Νταλογκ", "author": "Gabriel Garcia Marquez", "pages": 368, "genre": "Μαγικός Ρεαλισμός" }</pre>
<pre>{ "_id": ObjectId('6475edb75cc9f0ba3fa0eaeac'), "title": "Το Πείραμα", "author": "John Grisham", "pages": 432, "genre": "Θρίλερ" }</pre>
<pre>{ "_id": ObjectId('6475f1425cc9f0ba3fa0eaeaf'), "title": "", "author": "", "pages": 0, "genre": "" }</pre>

> \_MONGOSH

Παρουσίαση Εγγραφών/Αρχείων του Πίνακα

# Εύρεση Εγγραφών

The screenshot shows the MongoDB Compass interface. The left sidebar displays the database structure with 'TestDB.books' selected. The main area shows the 'TestDB.books' collection with a filter of `{author:"J.R.R. Tolkien"}`. Two documents are displayed:

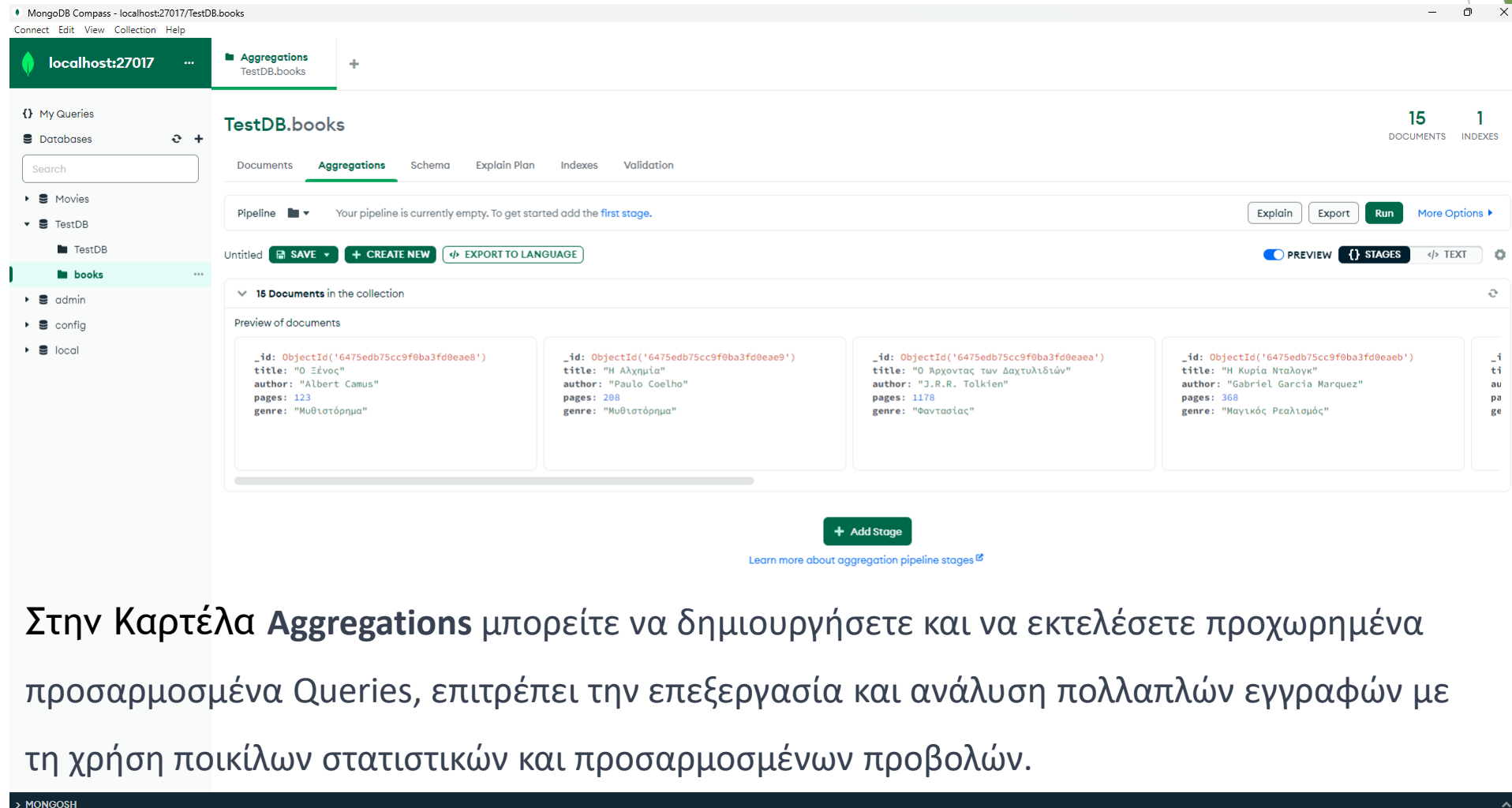
```
{ "_id": ObjectId('6475edb75cc9f0ba3fd0eaea'), "title": "Ο Άρχοντας των Δαχτυλιδών", "author": "J.R.R. Tolkien", "pages": 1178, "genre": "Θανταστικός" }
```

```
{ "_id": ObjectId('6475f1425cc9f0ba3fd0eaf7'), "title": "The Lord of the Rings", "author": "J.R.R. Tolkien", "pages": 1178, "genre": "Fantasy" }
```

`{author:"J.R.R. Tolkien"}`



# Καρτέλα Aggregations



The screenshot shows the MongoDB Compass interface for the 'TestDB.books' collection. The 'Aggregations' tab is active, displaying an empty pipeline with the message 'Your pipeline is currently empty. To get started add the first stage.' Below the pipeline, there are buttons for 'Explain', 'Export', 'Run', and 'More Options'. A 'PREVIEW' button is also visible. The document preview section shows 15 documents in the collection, with a preview of four documents displayed. Each document is a JSON object with fields for '\_id', 'title', 'author', 'pages', and 'genre'. The documents are:

- `{ "_id": "ObjectId('6475edb75cc9f0ba3fd0eae8')", "title": "Ο Ξένος", "author": "Albert Camus", "pages": 123, "genre": "Μυθιστόρημα" }`
- `{ "_id": "ObjectId('6475edb75cc9f0ba3fd0eae9')", "title": "Η Αλλημιά", "author": "Paulo Coelho", "pages": 208, "genre": "Μυθιστόρημα" }`
- `{ "_id": "ObjectId('6475edb75cc9f0ba3fd0eaea')", "title": "Ο Άρχοντας των Δαχτυλιδιών", "author": "J.R.R. Tolkien", "pages": 1178, "genre": "Φαντασίας" }`
- `{ "_id": "ObjectId('6475edb75cc9f0ba3fd0eaeab')", "title": "Η Κυρία Νταλογκ", "author": "Gabriel Garcia Marquez", "pages": 368, "genre": "Μαγικός Ρεαλισμός" }`

At the bottom of the interface, there is a '+ Add Stage' button and a link to 'Learn more about aggregation pipeline stages'.

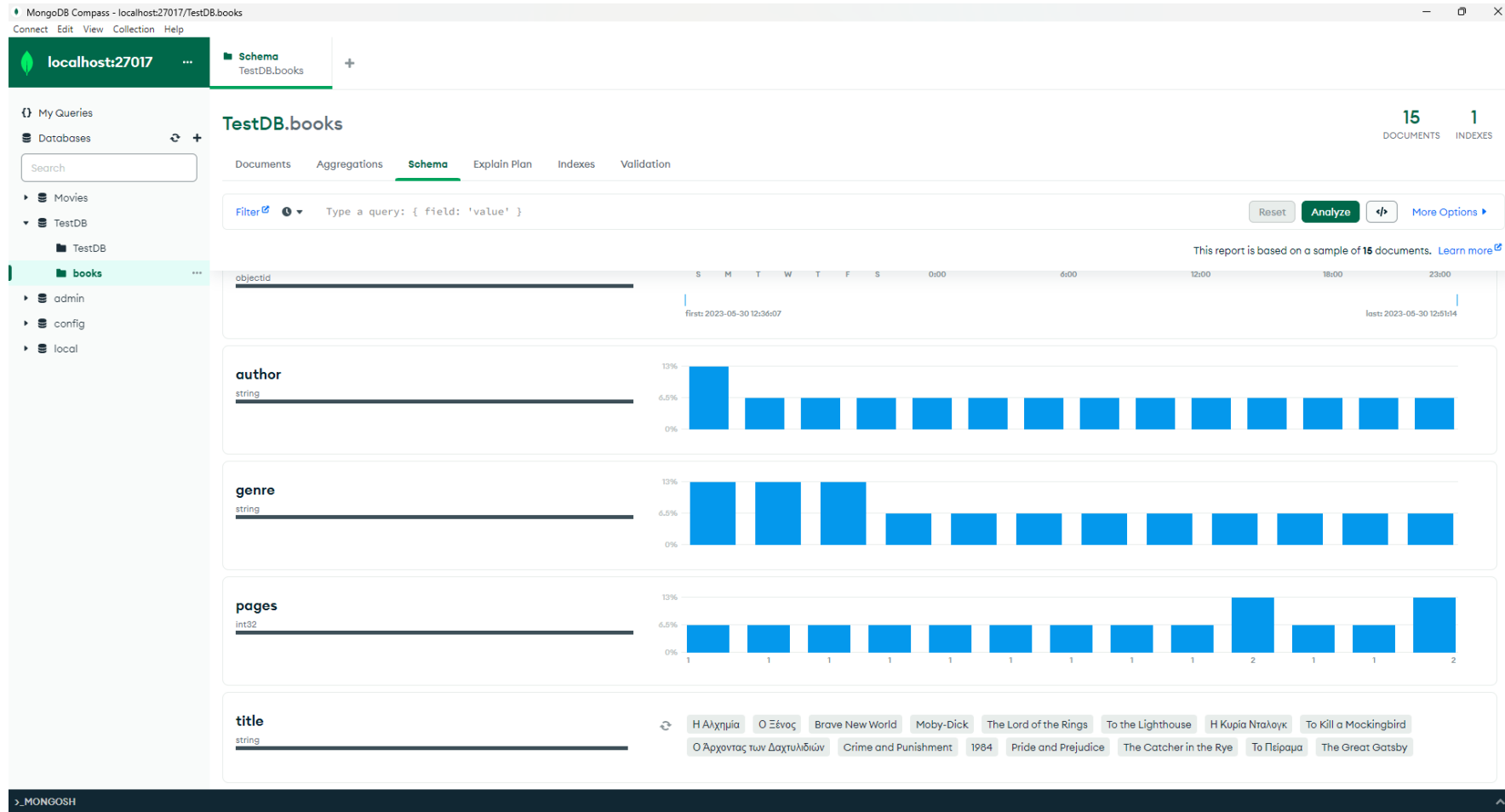
Στην Καρτέλα **Aggregations** μπορείτε να δημιουργήσετε και να εκτελέσετε προχωρημένα προσαρμοσμένα Queries, επιτρέπει την επεξεργασία και ανάλυση πολλαπλών εγγραφών με τη χρήση ποικίλων στατιστικών και προσαρμοσμένων προβολών.

# Καρτέλα Aggregations

Χρησιμοποιώντας την καρτέλα "Aggregations", μπορείτε να δημιουργήσετε πολύπλοκες προσαρμοσμένες ερωτήσεις που περιλαμβάνουν πολλά στάδια επεξεργασίας και ανάλυσης δεδομένων. Μερικά από τα στάδια που μπορείτε να χρησιμοποιήσετε περιλαμβάνουν τα εξής:

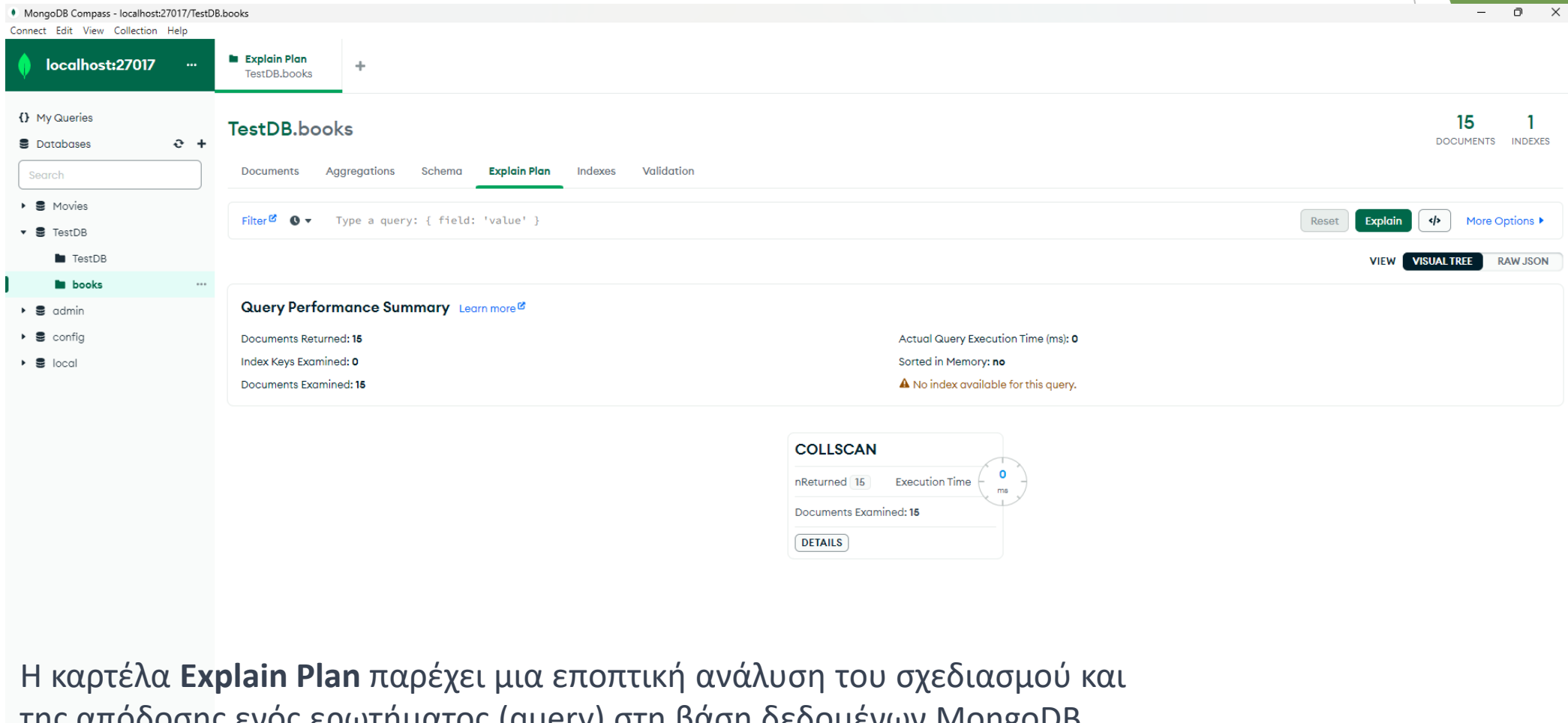
- **Match:** Φιλτράρει τις εγγραφές που πληρούν συγκεκριμένες συνθήκες.
- **Group:** Ομαδοποιεί τις εγγραφές με βάση ένα ή περισσότερα πεδία και εκτελεί συναρτήσεις συγκεντρώσεων, όπως sum, average, max, min κ.λπ.
- **Project:** Επιλέγει ποια πεδία θα περιληφθούν στην έξοδο και προβάλλει νέα υπολογισμένα πεδία.
- **Sort:** Ταξινομεί τα αποτελέσματα βάσει ενός ή περισσότερων πεδίων.
- **Limit:** Περιορίζει τον αριθμό των εγγραφών που επιστρέφονται.
- **Skip:** Παραλείπει έναν αριθμό εγγραφών από την αρχή των αποτελεσμάτων.

# Καρτέλα Schema



Γραφική αναπαράσταση των δεδομένων

# Καρτέλα Explain Plan



The screenshot shows the MongoDB Compass interface for the 'TestDB.books' collection. The 'Explain Plan' tab is active, displaying a query performance summary and a visual tree of the execution plan. The query is `{ field: 'value' }`. The performance summary indicates that 15 documents were returned, 0 index keys were examined, and 15 documents were examined. The actual query execution time is 0 ms, and the data is sorted in memory. A warning message states: 'No index available for this query.' The visual tree shows a 'COLLSCAN' operation with a circular gauge for execution time (0 ms) and a 'DETAILS' button.

localhost:27017 ... Explain Plan TestDB.books

TestDB.books

Documents Aggregations Schema **Explain Plan** Indexes Validation

Filter ⓘ ⓘ Type a query: { field: 'value' } Reset Explain ↩ More Options ▶

VIEW VISUAL TREE RAW JSON

**Query Performance Summary** [Learn more](#) ⓘ

Documents Returned: 15 Actual Query Execution Time (ms): 0  
Index Keys Examined: 0 Sorted in Memory: no  
Documents Examined: 15 ⚠ No index available for this query.

**COLLSCAN**

nReturned 15 Execution Time 0 ms  
Documents Examined: 15

DETAILS

Η καρτέλα **Explain Plan** παρέχει μια εποπτική ανάλυση του σχεδιασμού και της απόδοσης ενός ερωτήματος (query) στη βάση δεδομένων MongoDB. Αναλύει τον τρόπο με τον οποίο η MongoDB θα εκτελέσει το ερώτημα και παρέχει στατιστικές πληροφορίες σχετικά με την απόδοση της εκτέλεσης.

# Καρτέλα Explain Plan

Οι πληροφορίες που παρέχονται από την καρτέλα "Explain Plan" περιλαμβάνουν:

- **Στατιστικά στοιχεία ερωτήματος:** Εμφανίζει πληροφορίες για τον αριθμό των επιστρεφόμενων εγγραφών, τον αριθμό των επηρεαζόμενων εγγραφών, τον χρόνο εκτέλεσης, την αποδοτικότητα και άλλα στατιστικά στοιχεία που σχετίζονται με το ερώτημα.
- **Πλάνο εκτέλεσης:** Εμφανίζει τη σειρά εκτέλεσης των βημάτων που ακολουθεί ο MongoDB για να εκτελέσει το ερώτημα. Περιλαμβάνει πληροφορίες για τον τρόπο αξιοποίησης των διαθέσιμων δεικτών, τον τρόπο φιλτραρίσματος των εγγραφών και τη σειρά εφαρμογής των διαφόρων πράξεων στην εκτέλεση του ερωτήματος.
- **Αναλυτικές πληροφορίες:** Παρέχει αναλυτικές πληροφορίες σχετικά με τις στατιστικές επιδόσεις των διάφορων βημάτων εκτέλεσης, όπως ο αριθμός των εγγραφών που αποκτήθηκαν και ο χρόνος εκτέλεσης κάθε βήματος.

# Σύνδεση Python και MongoDB



mongoDB®

# Python Example 1

```
from pymongo import MongoClient

# Σύνδεση στη βάση δεδομένων MongoDB
client = MongoClient("mongodb://localhost:27017/")
db = client["bookstore"] # Όνομα Βάσης
collection = db["books"] #Όνομα Collection

# Ανάκτηση όλων των εγγραφών από τη συλλογή "books"
books = collection.find()

# Εμφάνιση του περιεχομένου των βιβλίων
for book in books:
    print("Title:", book["title"])
    print("Author:", book["author"])
    print("Pages:", book["pages"])
    print("Genre:", book["genre"])
    print("-----")

# Κλείσιμο της σύνδεσης με τη βάση δεδομένων
client.close()
```

# Python Example 2

```
from pymongo import MongoClient

# Σύνδεση στη βάση δεδομένων MongoDB
client = MongoClient("mongodb://localhost:27017/")
db = client["bookstore"]
collection = db["books"]

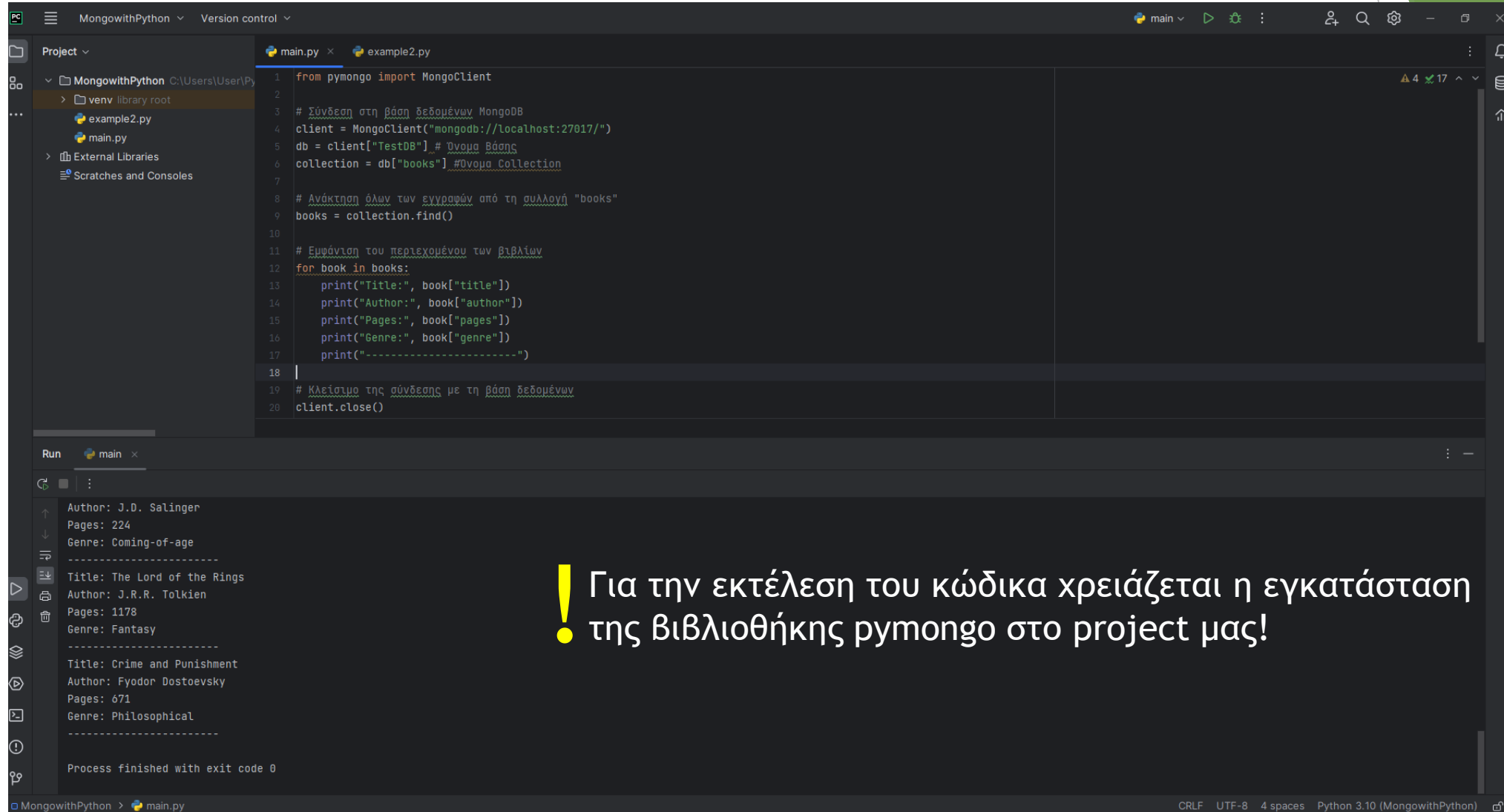
# Ανάκτηση εγγραφών με τον συγγραφέα "J.R.R. Tolkien"
query = {"author": "J.R.R. Tolkien"}
books = collection.find(query)

# Εμφάνιση του περιεχομένου των βιβλίων
for book in books:
    print("Title:", book["title"])
    print("Author:", book["author"])
    print("Pages:", book["pages"])
    print("Genre:", book["genre"])
    print("-----")

# Κλείσιμο της σύνδεσης με τη βάση δεδομένων
client.close()
```



# PyCharm



```
1 from pymongo import MongoClient
2
3 # Σύνδεση στη βάση δεδομένων MongoDB
4 client = MongoClient("mongodb://localhost:27017/")
5 db = client["TestDB"] # Όνομα βάσης
6 collection = db["books"] #Όνομα Collection
7
8 # Ανάκτηση όλων των εγγραφών από τη συλλογή "books"
9 books = collection.find()
10
11 # Εμφάνιση του περιεχομένου των βιβλίων
12 for book in books:
13     print("Title:", book["title"])
14     print("Author:", book["author"])
15     print("Pages:", book["pages"])
16     print("Genre:", book["genre"])
17     print("-----")
18
19 # Κλείσιμο της σύνδεσης με τη βάση δεδομένων
20 client.close()
```

Run main ×

```
Author: J.D. Salinger
Pages: 224
Genre: Coming-of-age
-----
Title: The Lord of the Rings
Author: J.R.R. Tolkien
Pages: 1178
Genre: Fantasy
-----
Title: Crime and Punishment
Author: Fyodor Dostoevsky
Pages: 671
Genre: Philosophical
-----
Process finished with exit code 0
```

MongowithPython > main.py CRLF UTF-8 4 spaces Python 3.10 (MongowithPython)

! Για την εκτέλεση του κώδικα χρειάζεται η εγκατάσταση της βιβλιοθήκης pymongo στο project μας!

Τέλος

 mongoDB Compass

