

MongoDB

הכרות

סיכום טכנולוגיות Web

מסד נתונים מבוסס JSON

התקנות

יצירת מסד נתונים

מחיקת מסד נתונים

Collections

Documents

TomerBu

הכרות

- **שם**
- **מקום מגורים**
- **רקע בתחום**
- **למה הגעתינו בתחום**
- **כמה שעות אתם משקיעים ביום**
- **איך הולך לכם בקורס עד כה**
- **בקשות מהמכללה/מורים**
- **המלצות לשיפור/לשימור**

ניהול שיעור:

- **לשאול שאלות (לחכמת לסייעם הסביר - בהרמת מצבו - נוצר תור)**
- **אשמה לשימוש מכם - בשיעור עצמו**
- **מוזמנים לפנות בפרטיו**
- **עדיף לשאול שאלות בוואטסאפ הקבוצתי -**
- **יהונתו גם עונה וגם אני.**

יצירת קשר

tomer.bu@gmail.com

050-7123012

אשמה שתosisפו אותו לקבוצה ווואטסאפ
(ונתקשר שם).

הפסיקות:

- הפסקה ראשונה: 10:30 - 20 דקות**
- הפסקת צהרים: 12:30 - 45 דקות**
- הפסקת אתנהטה: 14:30 - 20 דקות**

מאומנים להזכיר לגבי הפסיקות.

תרגישו חופשי לפנות אליו
בכל נושא.

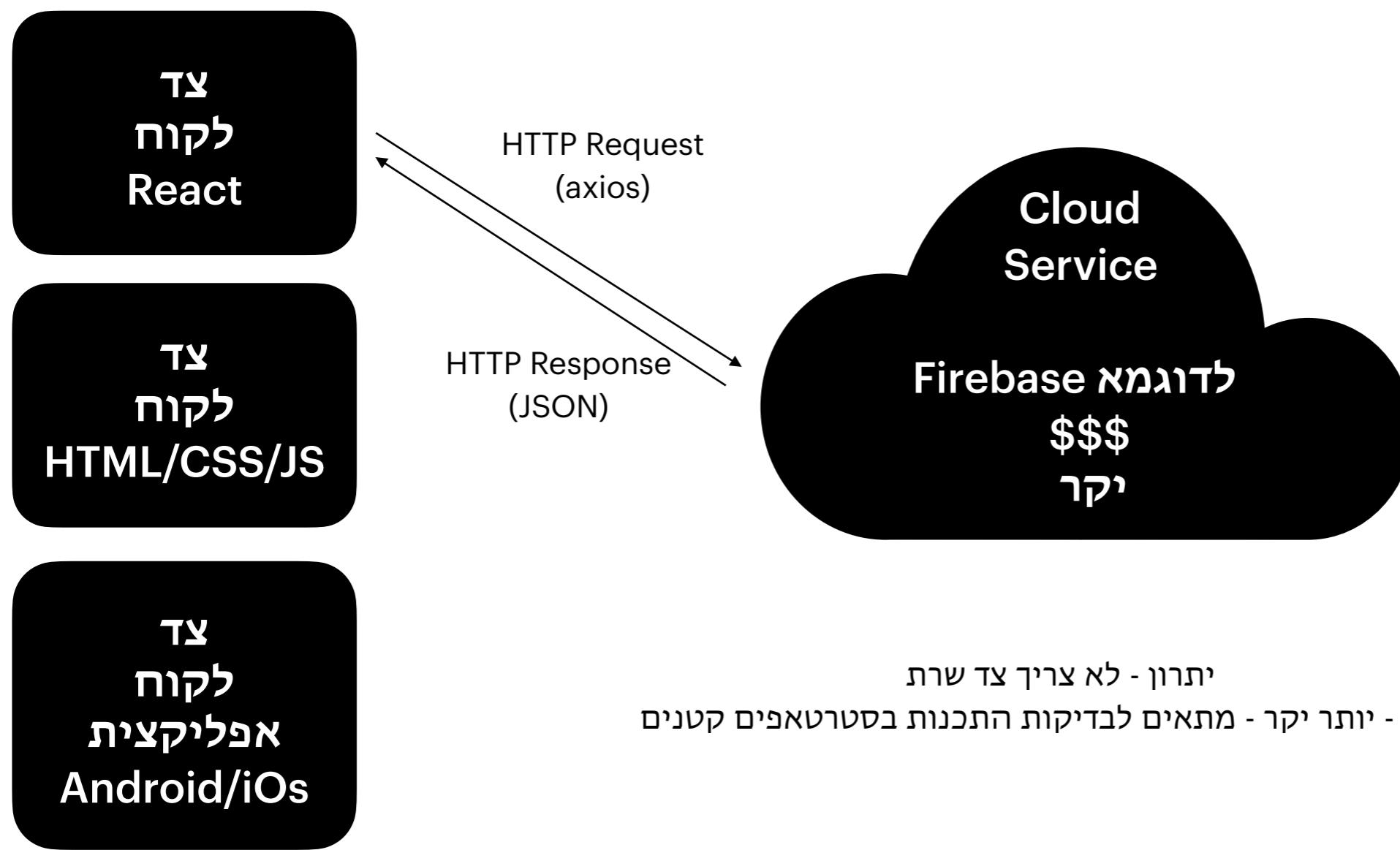
050-7123012

אשמה שתosisפו אותו לקבוצת וואטסאפ
(ונתקשר שם).

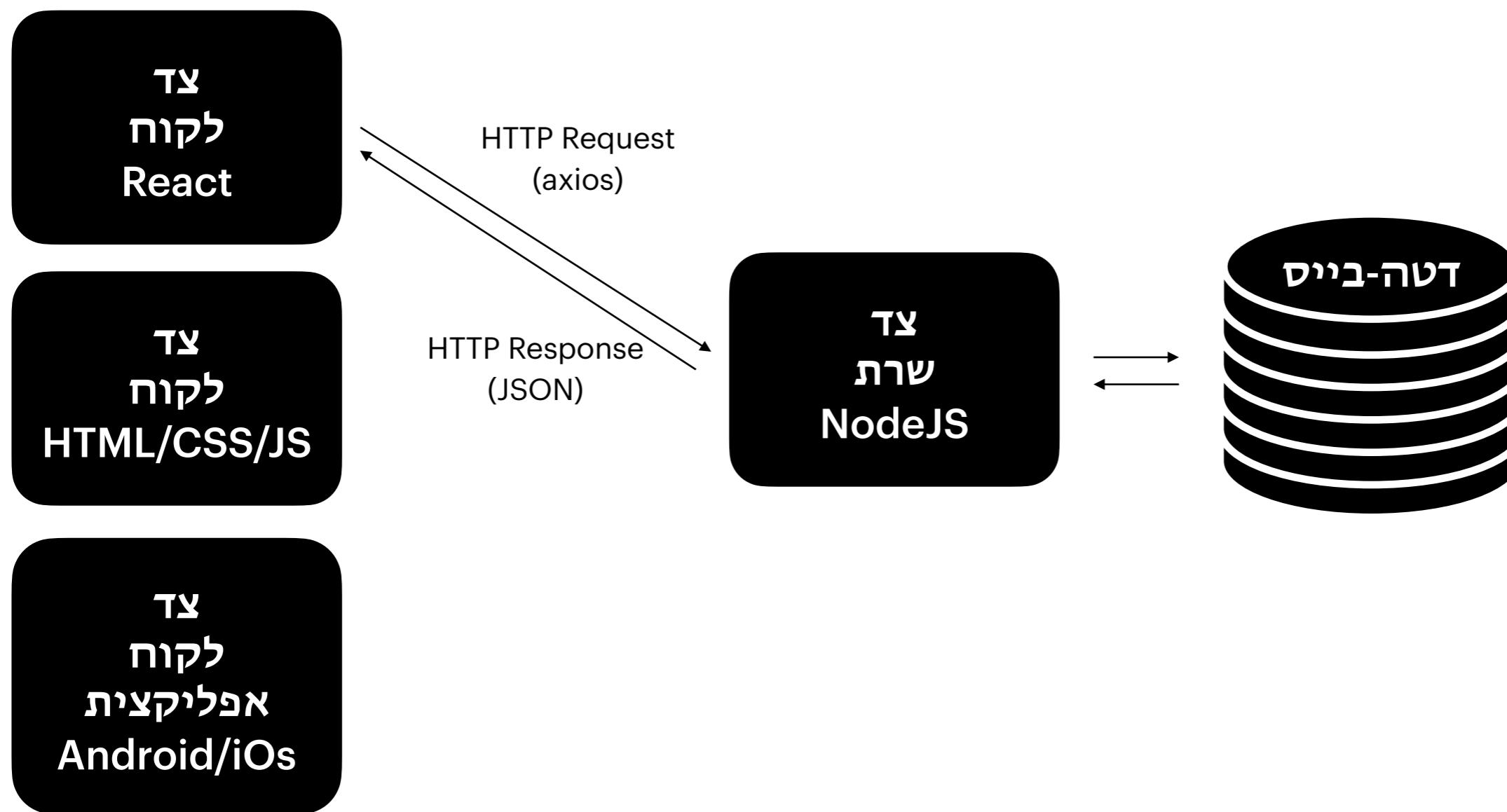
מבנה של הקורס:

- HTML/CSS/JS
- JS מתקדם טייפסקריפט
- צד לקוח - React
- SQL - דטה בייס
- MongoDB - דטה בייס
- NodeJS - צד שרת

מבנה של הקורס:



מבנה של הקורס:



MongoDB:

דטה-בייס מבוסס JSON:

העולם עבר עם SQL המון שנים

חלק מההנחות של SQL מבוססת עליהו התיישנו:
דיסק קשיח של 2TB עולה הרבה פחות מאשר בעבר.

נפח אחסון לא מהוות בעיה
בדטה-בייס שומרים טקסט

בSQL חיבים להגדר מראש את מבנה הטבלאות.
קודם כל CREATE TABLE עם ההגדרה של הסכמה.
ורק אז אפשר לשמור מידע.

כיום - לעיתים נרצה לשמור מידע ללא סכמה (מבנה הטבלאות):
כריית מידע באינטרנט

אם ידוע מראש המבנה של הדטה- אפשר להגדר במנגו את הסכמה.
אם לא ידוע מראש המבנה של הדטה- אפשר לשמור שירות את המידע. בלי להגדיר סכמה.

JSON מאד מוכר ופופולרי - פורמט נוח

MongoDB:

דטה-בייס מבוסס JSON:

העולם עבר עם SQL המון שנים

חלק מההנחות של SQL מבוססת עליהו התיישנו:
דיסק קשיח של 2TB עולה הרבה פחות מאשר בעבר.

נפח אחסון לא מהוות בעיה
בדטה-בייס שומרים טקסט

בSQL חיבים להגדר מראש את מבנה הטבלאות.
קודם כל CREATE TABLE עם ההגדרה של הסכמה.
ורק אז אפשר לשמור מידע.

כיום - לעיתים נרצה לשמור מידע ללא סכמה (מבנה הטבלאות):
כריית מידע באינטרנט

אם ידוע מראש המבנה של הדטה- אפשר להגדר במנגו את הסכמה.
אם לא ידוע מראש המבנה של הדטה- אפשר לשמור שירות את המידע. בלי להגדר סכמה.

JSON מאד מוכר ופופולרי - פורמט נוח

MongoDB:

```
{  
  "userId": 1,  
  "id": 1,  
  "title": "delectus aut autem",  
  "completed": false  
}
```



במקום לבצע שאילתת מסוג:

```
INSERT INTO People(userId, id, title, completed)  
VALUES(1, 1, 'delet...', false)
```

השאילתת במנגנון תקבל שירות את ה-JSON:

```
db.people.insertOne({userId: 1, id: 1, title: 'delet...', completed: false})
```

MongoDB:

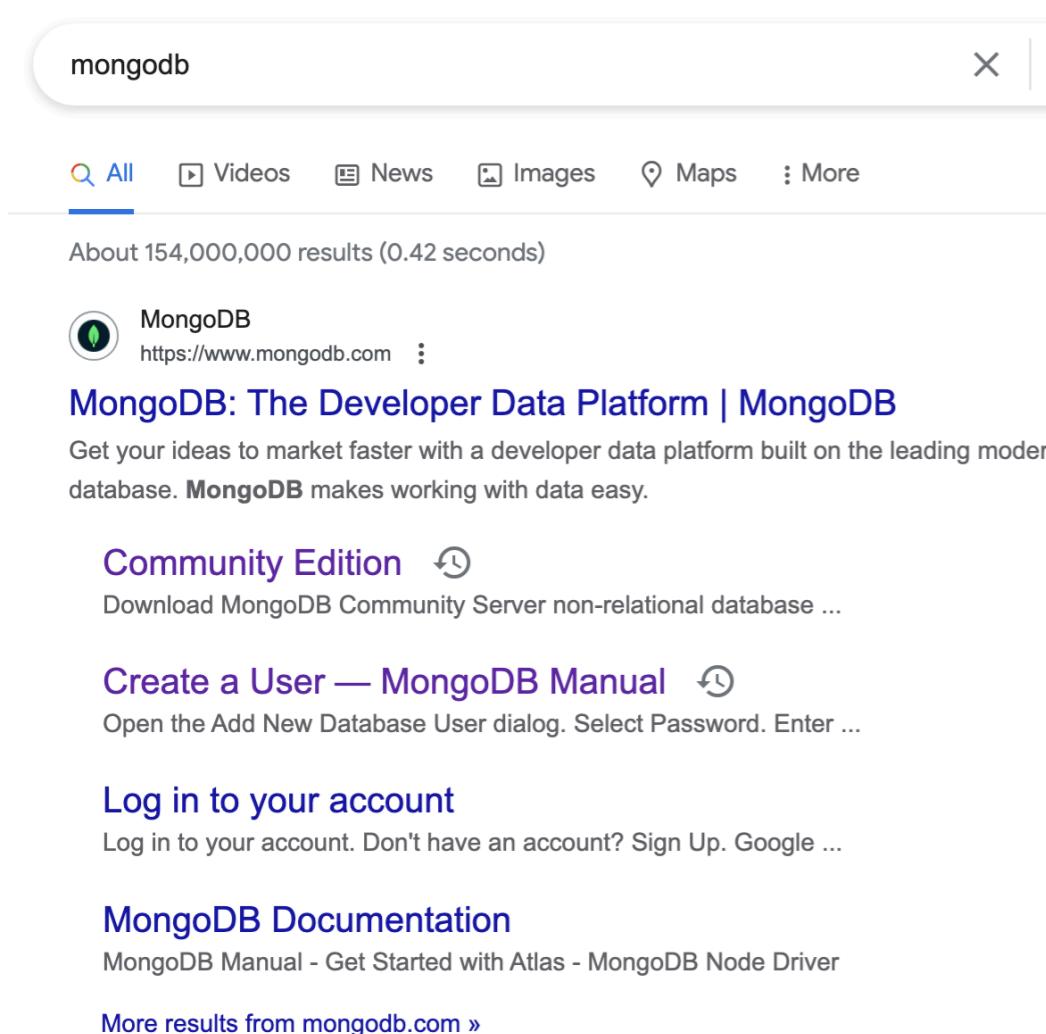
השם של מונגו מבוסס על המילה
humongous mongodb

מיועד להתמודד עם כמויות אדירות של מידע:

הרבה יותר מהיר

למה למדנו SQL
עדין בשימוש רחב
עבדה על פרויקטים ישנים יותר.
או פרויקטים שמשלבים גם SQL וגם MongoDB.
לדעת להשוות בין הטכנולוגיות.

התקנות: Mongodb



A screenshot of a search results page from a search engine. The search term "mongodb" is entered in the search bar. Below the search bar are filters: All (selected), Videos, News, Images, Maps, and More. A message indicates "About 154,000,000 results (0.42 seconds)". The top result is a link to the MongoDB website, featuring the MongoDB logo (a green leaf inside a circle) and the text "MongoDB: The Developer Data Platform | MongoDB". A brief description follows: "Get your ideas to market faster with a developer data platform built on the leading modern database. MongoDB makes working with data easy." Below this, there are several other links: "Community Edition" (with a download icon), "Create a User — MongoDB Manual" (with a help icon), "Log in to your account" (with a user icon), and "MongoDB Documentation" (with a book icon). Each link has a brief description underneath it.

mongodb

All Videos News Images Maps More

About 154,000,000 results (0.42 seconds)

 MongoDB
<https://www.mongodb.com> :

MongoDB: The Developer Data Platform | MongoDB

Get your ideas to market faster with a developer data platform built on the leading modern database. **MongoDB** makes working with data easy.

Community Edition ↗

Download MongoDB Community Server non-relational database ...

Create a User — MongoDB Manual ↗

Open the Add New Database User dialog. Select Password. Enter ...

Log in to your account

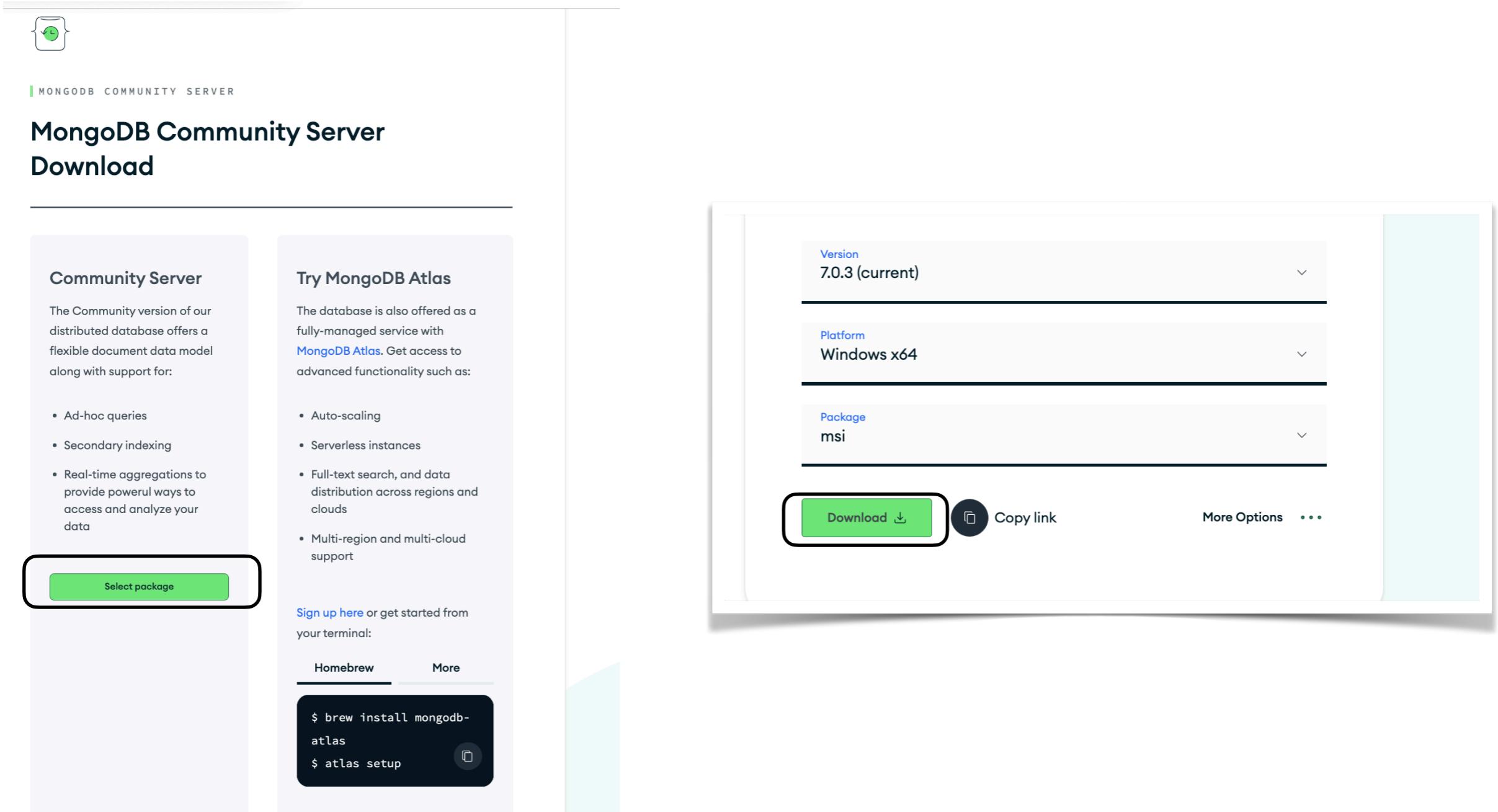
Log in to your account. Don't have an account? Sign Up. Google ...

MongoDB Documentation

MongoDB Manual - Get Started with Atlas - MongoDB Node Driver

[More results from mongodb.com »](#)

התקנות: חולנות:



The screenshot shows the MongoDB Community Server Download page. On the left, there's a sidebar with a logo and a 'MONGDB COMMUNITY SERVER' link. The main content area has two sections: 'Community Server' and 'Try MongoDB Atlas'. The 'Community Server' section describes the distributed database's features and lists support for Ad-hoc queries, Secondary indexing, and Real-time aggregations. It includes a 'Select package' button highlighted with a green border. The 'Try MongoDB Atlas' section promotes the managed service with features like Auto-scaling, Serverless instances, Full-text search, and Multi-region support. It also links to 'Sign up here' and terminal setup instructions for Homebrew and More. On the right, a detailed download interface is shown for version 7.0.3 (current) on Windows x64. It shows a 'Download' button, a 'Copy link' button, and a 'More Options' menu.

MONGDB COMMUNITY SERVER

MongoDB Community Server

Download

Community Server

The Community version of our distributed database offers a flexible document data model along with support for:

- Ad-hoc queries
- Secondary indexing
- Real-time aggregations to provide powerful ways to access and analyze your data

Select package

Try MongoDB Atlas

The database is also offered as a fully-managed service with [MongoDB Atlas](#). Get access to advanced functionality such as:

- Auto-scaling
- Serverless instances
- Full-text search, and data distribution across regions and clouds
- Multi-region and multi-cloud support

[Sign up here](#) or get started from your terminal:

[Homebrew](#) [More](#)

```
$ brew install mongodb-atlas  
$ atlas setup
```

Version
7.0.3 (current)

Platform
Windows x64

Package
msi

Download  **Copy link** **More Options** 

התקנות: בהתקנה - לבחור ב

complete version

בעיות שנטקלנו:
לא הותקן :compass

<https://www.mongodb.com/try/download/compass>

The screenshot shows the MongoDB Compass Download (GUI) page. It includes sections for Compass (full version), Readonly Edition (limited to read operations), and Isolated Edition (disabled all network connections except MongoDB instance). A 'Learn more' link is also present. At the bottom, there are dropdown menus for Version (set to 1.40.4 (Stable)), Platform, and Package, followed by a 'Download' button and a 'More Options' button.

התקנה של גירסה ישנה - אם יש באג בגרסה הנוכחית.

התקנות של Server Mac

<https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-os-x/>

```
xcode-select --install
```



```
brew tap mongodb/brew
```



```
brew update
```



```
brew install mongodb-community@7.0
```

```
brew services start mongodb-community@7.0
```



התקנות של MongoDB Compass Mac

<https://www.mongodb.com/try/download/compass>



TOOLS

MongoDB Compass Download (GUI)

Easily explore and manipulate your database with Compass, the GUI for MongoDB. Intuitive and flexible, Compass provides detailed schema visualizations, real-time performance metrics, sophisticated querying abilities, and much more.

Please note that MongoDB Compass comes in three versions: a full version with all features, a read-only version without write or delete capabilities, and an isolated edition, whose sole network connection is to the MongoDB instance.

For more information, see our [documentation pages](#).

Compass

The full version of MongoDB Compass, with all features and capabilities.

Readonly Edition

This version is limited strictly to read operations, with all write and delete capabilities removed.

Isolated Edition

This version disables all network connections except the connection to the MongoDB instance.

[Learn more](#)

Version
1.40.4 (Stable)

Platform

Package

Download

More Options ...

מה הותקו:

MongoDB
Server

שרת

כל הזמן רץ ברקע

MongoDB
Compass
GUI

תוכנה גרפית לעבודה עם השרת

התחברות - connect

NodeJS

Mongo
Shell
CMD
אפשר לעבוד עם

אחרי התחברות

The screenshot shows the MongoDB Compass application running on localhost:27017. The interface has a dark theme with a green header bar. The top navigation bar includes tabs for "My Queries" (which is active), "Databases", and "Performance". On the left sidebar, there are links for "My Queries", "Databases", and a search bar. Below the sidebar, a list of databases is shown: "admin", "biz_cards", and "config". The main content area contains Hebrew text: "חלון פקודה לכתיבה של אילות", "אפשר גם לכתוב ב-GUI", and "אבל כך לא נלמד את הסינטקס". To the right of the text is a green icon of a magnifying glass inside curly braces. Below the text, it says "No saved queries yet." and "Start saving your aggregations and find queries, you'll see them here." A downward arrow points from the Hebrew text towards the "No saved queries yet." message.

MongoDB Compass - localhost:27017

localhost:27017 ...

My Queries Databases Performance

{ My Queries

Databases

Search

admin

biz_cards

config

ןלון פקודה לכתיבה של אילות
אפשר גם לכתוב ב-GUI
אבל כך לא נלמד את הסינטקס

No saved queries yet.

Start saving your aggregations and find queries, you'll see them here.

>_MONGOSH

test>

פקודות ב-Shell

הצגת כל databases



```
>_MONGOSH
test> show databases
```

נקיוון של ה-
console (clear screen)



```
>_MONGOSH
test> cls
```

פקודות ב-Shell

בחירה בדטה-בייס שבו רוצים לעבוד:

(גם אם הוא לא קיים עדין)



```
>_MONGOSH
test> use lec1
```

ברגע שנשмарו מידע בפנים - יוצר הדטה-בייס

הכנסה של מידע לדטה-בייס:

```
>_MONGOSH
```

```
test> use lec1
```

```
lec1> db.mycollection.insertOne({firstname: 'Moe', isHappy: true})
```

↑
db מתייחס לlec1

שם האוסף
(בדומה לשם הטבלה בSQL)

פונקציה להכנסה של מידע

אובייקט שאותו רוצים להכניס

```
> use lec1
< switched to db lec1
> db.mycollection.insertOne({firstname: 'Moe', isHappy: true})
< {
    acknowledged: true,
    insertedId: ObjectId("655f28ad83d29d14e6eb2eb6")
}
```

מושגים בMongo

```
>_MONGOSH  
test> use lec1
```

Database מסד נתונים: → SQL Database

Collection אוסף: → SQL Table

Document מסמך: → SQL Row

```
lec1> db.mycollection.insertOne({firstname: 'Moe', isHappy: true})
```



הציגת מסמכים באוסף - GUI

The screenshot shows the MongoDB Compass interface. On the left, the sidebar lists databases: My Queries, Databases, Search, admin, config, lec1 (with movies and mycollection), local, and test. The 'mycollection' item under 'lec1' is selected and highlighted with a green border. The main panel shows the 'lec1.mycollection' collection. At the top, there are tabs for Documents, Aggregations, Schema, Indexes, and Validation. The 'Documents' tab is active. Below it, there's a search bar with 'Type a query: { field: 'value' }', and buttons for Explain, Reset, Find, and Options. A 'Filter' dropdown is also present. In the center, there's an 'ADD DATA' button with a plus sign and an 'EXPORT DATA' button with a download icon. The bottom section displays a single document with the following fields:

```
_id: ObjectId('655f28ad83d29d14e6eb2eb6')
firstname: "Moe"
isHappy: true
```

On the right side of the interface, there are two large numbers: '1 DOCUMENTS' and '1 INDEXES'. At the very bottom right, there is a vertical arrow pointing upwards with the Hebrew text 'אפשרויות תצוגה' (Display Options) next to it.

לכל רשומה יש `_id`

The screenshot shows the MongoDB Compass interface. On the left, the sidebar lists databases: My Queries, Databases, Search, admin, config, lec1 (with movies), mycollection (selected and highlighted in green), local, and test. The main area is titled 'lec1.mycollection' and shows the 'Documents' tab selected. It displays a single document with the following fields:

```
_id: ObjectId('655f28ad83d29d14e6eb2eb6')
firstname: "Moe"
isHappy: true
```

Below the document, there are buttons for 'ADD DATA' and 'EXPORT DATA'. At the bottom right, there are navigation buttons for '1 - 1 of 1' and other document controls. The top right corner shows '1 DOCUMENTS' and '1 INDEXES'.

הטיפוס
.ObjectId -
טיפוס מובנה בסongo

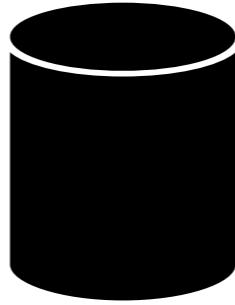
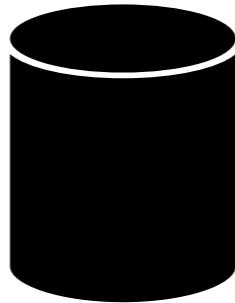
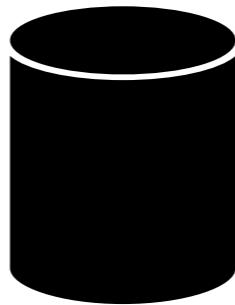
מתפקיד index

בדומה לPrimary Key בSQL

AUTO_INCREMENT

מתעדכו אוטומטית

לכל רשומה יש id



דאג שהID יהיה ייחודי

ולא נדרש לדאוג לסנכרון

העשרה - מהו מרכיב ObjectId

<https://www.mongodb.com/docs/manual/reference/method/ObjectId/>

Returns a new [ObjectId](#). The 12-byte [ObjectId](#) consists of:

- A 4-byte timestamp, representing the ObjectId's creation, measured in seconds since the Unix epoch.
- A 5-byte random value generated once per process. This random value is unique to the machine and process.
- A 3-byte incrementing counter, initialized to a random value.

For timestamp and counter values, the most significant bytes appear first in the byte sequence (big-endian). This is unlike other BSON values, where the least significant bytes appear first (little-endian).

If an integer value is used to create an ObjectId, the integer replaces the timestamp.

סיכון -
הIDI של המספרים מסודרים לפי הזמן
rndomilit - יוצר ייחודיות של המפתחות.

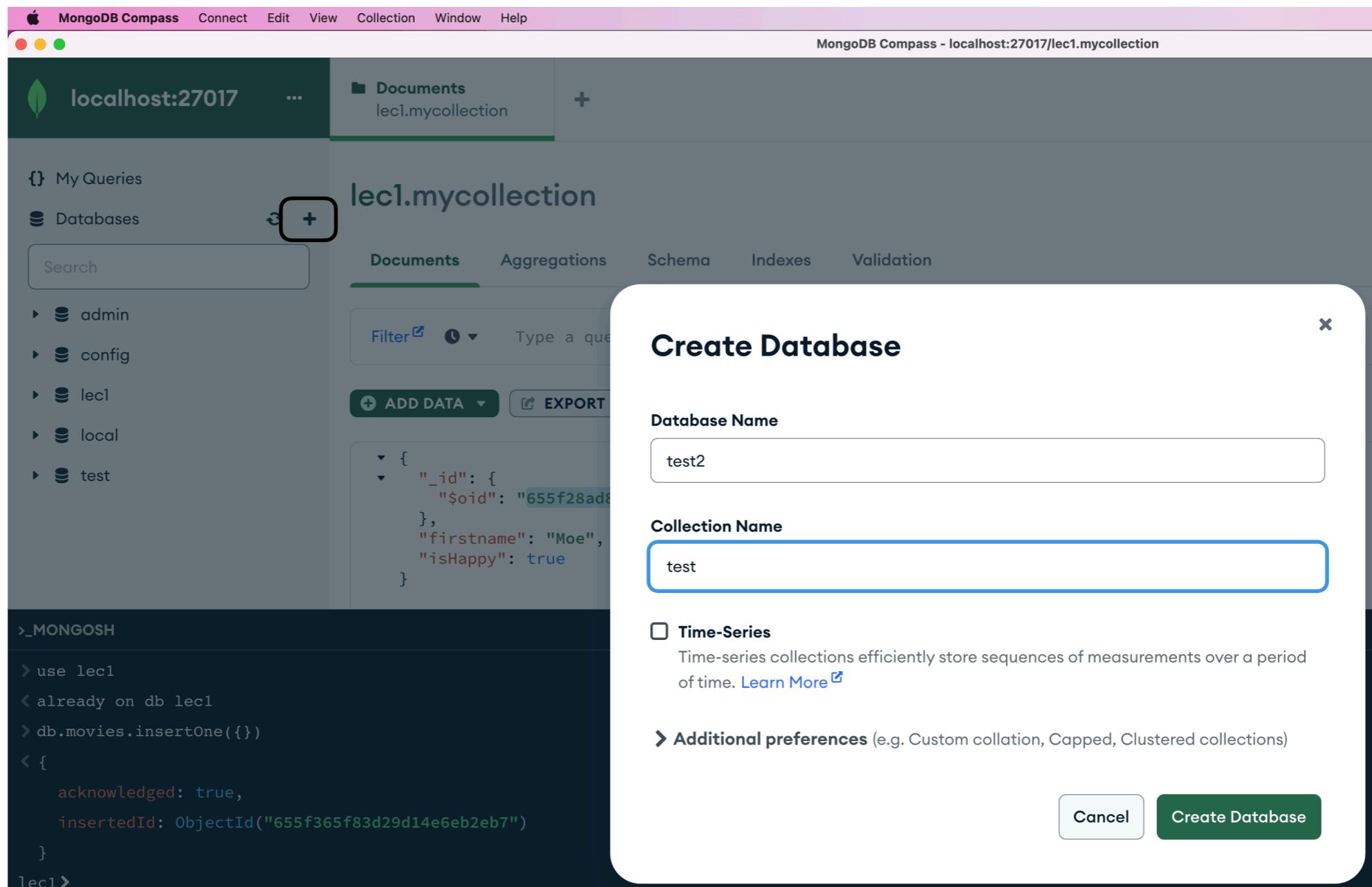
מורכב מ 12 bytes
(זה הגודל)

4 בתים ראשונים - הזמן שנוצרה הרשומה
(סידור אוטומטית לפי הזמן)

5 בתים רנדומליים - מונע חוזרתויות

3 בתים של counter עולה

יצירת דטה-בייס: GUI

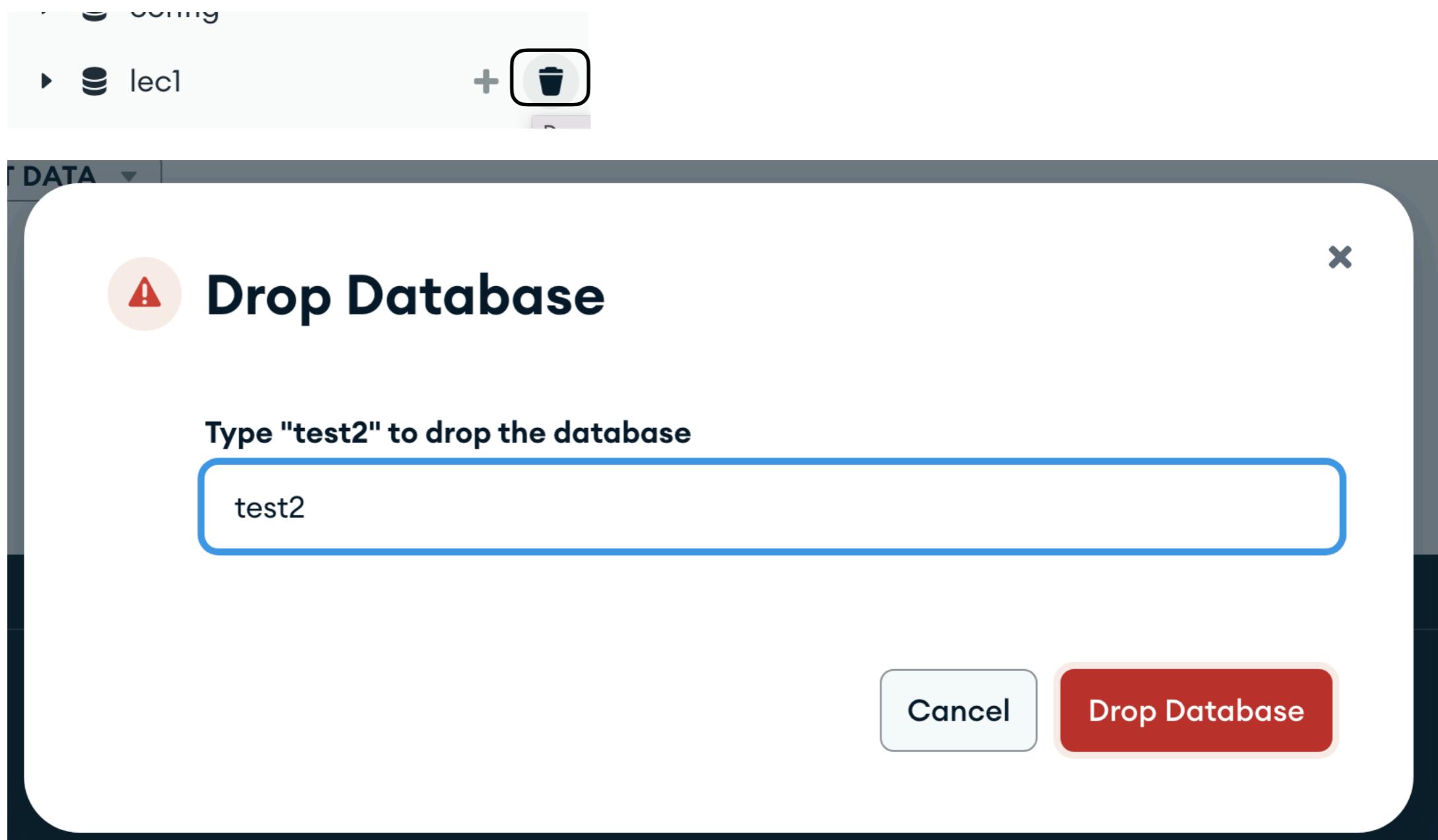


מסקנה:

כדי ליצור דטה-בייס - חיברים ליצור גם אוסף

פקודה use לא מספקת.

מחיקת DATABASE: GUI



יצירת דטה-בייס: SHELL

```
>_MONGOSH  
lec2> use lec2
```

```
lec2>|db.createCollection('test')
```

צילום מסך:
מак: cmd+ctrl+shift+4

חלונות:
win + shift + s

מחיקת דטה-בייס: SHELL

```
>_MONGOSH  
  
lec2> use lec2
```

```
lec2> db.dropDatabase()
```

drop database פונקציה

הצגת עזרה: help

```
>_MONGOSH

> help
< Shell Help

use                                Set current database
show                               'show databases'/'show dbs': Print a list of all available da
                                         current database. 'show profile': Prints system.profile infor
                                         Print a list of all roles for current database. 'show log <ty
                                         Print all logs.

exit                               Quit the MongoDB shell with exit/exit()/.exit
quit                               Quit the MongoDB shell with quit/quit()
Mongo                             Create a new connection and return the Mongo object. Usage: n
connect                           Create a new connection and return the Database object. Usage
                                         result of the last line evaluated; use to further iterate
it                                 Shell version
version                           Shell version
load                               Loads and runs a JavaScript file into the current shell envir
enableTelemetry                   Enables collection of anonymous usage data to improve the mon
disableTelemetry                  Disables collection of anonymous usage data to improve the mo
passwordPrompt                   Prompts the user for a password
sleep                             Sleep for the specified number of milliseconds
print                            Prints the contents of an object to the output
printjson                          Alias for print()
convertShardKeyToHashed           Returns the hashed value for the input using the same hashing
cls                                Clears the screen like console.clear()
isInteractive                    Returns whether the shell will enter or has entered interacti

For more information on usage: https://docs.mongodb.com/manual/reference/method
```

```
> version()
< 1.10.0
lec2> version()
```

הצגת עזרה:

עזרה לגבי פונקציה מסוימת:

```
lec2> db.mycollection.insertOne.help()
```

כדי לזכור —————

עזרה לגבי האובייקט db:

```
lec2> db.help()
```

עזרה לגבי האובייקט של האוסף:

```
lec2> db.mycollection.help()
```

תרגילים:

צרו דטה בייס חדש בשם testdb בעזרת הפקודה `use`
האם הדטה ביבס כבר נוצר?

```
use testdb
```

הוסיפו לדטה ביבס אוסף בשם `users`
לאוסף `users` נוסיף 2 מסמכים: לכל משתמש יש שם פרטי, שם משפחה וגיל

```
testdb > db.createCollection('users')
```

הציגו את המידע בcompass

מחקו את הדטה בייס שיצרתם (מחלון הפקודה).

איך אפשר להציג עזרה?

איך לנוקות את המסר?

שלחו פתרונות בצל'אט - יעזר לךולם למצוא פתרון
יעזר לי לקבל אינדיקציה שהבנתם

```
testdb > db.users.insertOne({firstName: 'אבי', lastName: 'green', age: 20})
```

פקודות לאוספים: collection methods: Create, Read, Update, Delete - CRUD

```
db.createCollection('ads')
```

יצירת אוסף

```
db.getCollectionNames()
```

שמות כל האוספים

```
db.ads.renameCollection('myads')
```

שינוי שם

```
testdb > db.myads.drop()
```

מחיקת אוסף

מינি תרגיל:

משימה 1

- צור בסיס נתונים חדש בשם mydb (דרך זה - cmd)
- בבסיס נתונים זה צור collection products בשם products ו כמו כן צור עוד collection categories בשם categories
- הציג את רשימת האוסףים (collection) שבבסיס הנתונים שיצרת

משימה 2

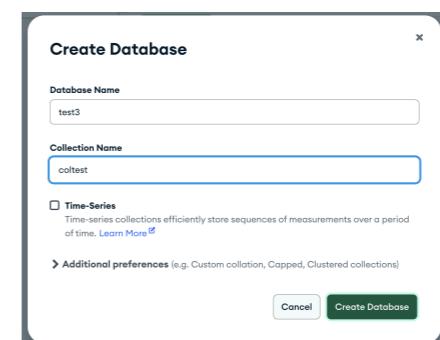
- מחק את בסיס הנתונים db mydb שיצרת בתרגיל הקודם וצור בסיס נתונים חדש בשם mydb2
- הוסיף לבסיס נתונים זה collection users בשם users
- שנה את שם ה - l collection users ל users
- בדוק שאכן השם השתנה ל users ואז מחק collection users

משימה 3

- הפעם בעזרת kompass צור בסיס נתונים חדש בשם test3 ובו>Create collection בשם coltest
- מחק את בסיס הנתונים וכן את collection בשם coltest

```
> use mydb
< switched to db mydb
> db.createCollection('products')
< { ok: 1 }
> db.createCollection('categories')
< { ok: 1 }
> db.getCollectionNames()
< [ 'categories', 'products' ]
```

```
> db.dropDatabase()
< { ok: 1, dropped: 'mydb' }
> use mydb2
< switched to db mydb2
> db.createCollection('myusers')
< { ok: 1 }
> db.myusers.renameCollection('users')
< { ok: 1 }
> db.getCollectionNames()
< [ 'users' ]
> db.users.drop()
< true
```



אינדקס:

- שיפור יעילות בחיפוש

- אפשר לנו להגדיר רשומה כעניקה

בכל הוספת מסמך:

המערכת מתחזק הקלעים - מוסיפה את העמודה הספציפית לעצם חיפוש
כדי למצוא את הערך בקלות

אינדקס:

כמו במלון שיש בסוף מפתח/index

קל להשתמש בו! מייעל את החיפוש.

ברגע שמחפשים לפי תכונה שיש לה אינדקס
המערכת תחפש באינדקס במקום לעבור על כל המסמכים.

בSQL כשהגדרנו עמודה כעיקרי Primary Key
ונוצר לנו אינדקס.

בSQL כשהגדרנו עמודה כUNIQUE
ונוצר לנו אינדקס.

אינדקס:

יצירת אינדקס: יותר יעיל לחפש לפי השדה זהה:

רואה - יעילות בחיפוש, הפסד: מאחסן גם באינדקס וגם באוסף עצמו.

```
db.users.createIndex({email: 1})
```



סדר עולה - A to Z

סדר הפוך
-1

```
db.users.createIndex({email: 1}, {unique: true})
```

יצירת אינדקס: שמוודא שכתובות האימייל - לא חוזרת על עצמה - UNIQUE

UNIQUE = ייחודי.

אינדקס:

The screenshot shows the MongoDB Compass interface. On the left, the sidebar lists databases: My Queries, Databases, Search, admin, config, lec1 (with movies and mycollection), users (selected and highlighted with a green border), local, mydb2, and test. The main area displays the 'lec1.users' collection. The top navigation bar includes tabs for Documents, Aggregations, Schema, Indexes (which is selected and highlighted with a green border), and Validation. Below the tabs are buttons for Create Index and Refresh. The main content area is a table titled 'Name and Definition' with columns for Type, Size, Usage, and Properties. It lists two indexes: '_id_' (REGULAR, 4.1 KB, 1 usage, UNIQUE) and 'email_1' (REGULAR, 4.1 KB, 0 usages, UNIQUE). Two vertical arrows point upwards from the bottom of the table towards the top of the page, likely indicating scroll position.

Name and Definition	Type	Size	Usage	Properties
id	REGULAR <small>i</small>	4.1 KB	1 (since Thu Nov 23 2023)	UNIQUE <small>i</small>
email_1	REGULAR <small>i</small>	4.1 KB	0 (since Thu Nov 23 2023)	UNIQUE <small>i</small>

**המשמעות של אינדקס UNIQUE:
למעט ייעילות בחיפוש:
"אי אפשר להכניס 2 משתמשים עם אותו email"**

```
> MONGOSH

> db.users.insertOne({email: 'tomer@gmail.com' })
< {
    acknowledged: true,
    insertedId: ObjectId("655f52dc83d29d14e6eb2eba")
}

> db.users.insertOne({email: 'tomer@gmail.com' })
✖ ▶ MongoServerError: E11000 duplicate key error collection:
lec1 >
```

הכנסת רשומות (ברבים)

```
lec1> db.movies.insertMany([]  
 {  
   "Title": "They Shall Not Grow Old",  
   "Year": 2018,  
   "Rated": "R",
```

תוספַּלכְרֹום:

<https://chromewebstore.google.com/detail/json-viewer/gbmdgpbipfallnflgajpalibnhdgobh?pli=1>

{≡} JSON Viewer

[Remove from Chrome](#)

Featured 4.5 ★ (1K ratings)

Extension

Developer Tools

1,000,000 users

```
1 // 20150625170715
2 // https://api.github.com/repos/tulios/json-viewer
3
4 {
5   "id": 12635853,
6   "name": "json-viewer",
7   "full_name": "tulios/json-viewer",
8   "owner": {
9     "login": "tulios",
10    "id": 33231,
11    "avatar_url": "https://avatars.githubusercontent.com/u/33231?v=3",
12    "gravatar_id": "",
13    "url": "https://api.github.com/users/tulios",
14    "html_url": "https://github.com/tulios",
15    "followers_url": "https://api.github.com/users/tulios/followers",
16    "following_url": "https://api.github.com/users/tulios/following{/other_user}",
17    "gists_url": "https://api.github.com/users/tulios/gists{/gist_id}",
18    "starred_url": "https://api.github.com/users/tulios/starred{/owner}{/repo}",
19    "subscriptions_url": "https://api.github.com/users/tulios/subscriptions",
20    "organizations_url": "https://api.github.com/users/tulios/orgs",
21    "repos_url": "https://api.github.com/users/tulios/repos",
22    "events_url": "https://api.github.com/users/tulios/events{/privacy}",
23    "received_events_url": "https://api.github.com/users/tulios/received_events",
24    "type": "User",
25    "site_admin": false
26  },
27  "private": false,
28  "html_url": "https://github.com/tulios/json-viewer",
29  "description": "It is a Chrome extension for printing JSON and JSONP."
30 }
```

```
1 // 20150625171154
2 // https://api.github.com/repos/tulios/json-viewer
3
4 {
5   "id": 12635853,
6   "name": "json-viewer",
7   "full_name": "tulios/json-viewer",
8   "owner": {
9     "login": "tulios",
10    "id": 33231,
11    "avatar_url": "https://avatars.githubusercontent.com/u/33231?v=3",
12    "gravatar_id": "",
13    "url": "https://api.github.com/users/tulios",
14    "html_url": "https://github.com/tulios",
15    "followers_url": "https://api.github.com/users/tulios/followers",
16    "following_url": "https://api.github.com/users/tulios/following{/other_user}",
17    "gists_url": "https://api.github.com/users/tulios/gists{/gist_id}",
18    "starred_url": "https://api.github.com/users/tulios/starred{/owner}{/repo}",
19    "subscriptions_url": "https://api.github.com/users/tulios/subscriptions",
20    "organizations_url": "https://api.github.com/users/tulios/orgs",
21    "repos_url": "https://api.github.com/users/tulios/repos",
22    "events_url": "https://api.github.com/users/tulios/events{/privacy}",
23    "received_events_url": "https://api.github.com/users/tulios/received_events",
24    "type": "User",
25    "site_admin": false
26  },
27  "private": false,
28  "html_url": "https://github.com/tulios/json-viewer",
29  "description": "It is a Chrome extension for printing JSON and JSONP.",
30  "fork": false,
```



<https://gist.githubusercontent.com/Urdzik/de477f8e3d7baf4366c9d797fce63531/raw/38c6afa2937ef222323392cc34c8c8c77e02fc40/Movie.json>

תרגילים

אנו דטה-ביס חדש לתרגילים!

משימה 1

צור אוסף חדש (collection) בשם users והגדיר בו שדה (field) בשם email שיהיה ייחודי (unique) בעזרת המתודה createIndex

משימה 2

- בעזרת המתודה insert הוסף לאוסף users מספר documents חדשים כאשר כל document יכול את השדות name,email,password,city,city

משימה 3

בעזרת המתודה insertMany צור אוסף חדש (collection) בשם products ובו מסמכים שיכילו שדות כגון: title,article,price וכו'

משימה 4

בעזרת המתודה insert צור אוסף חדש בשם tester שבו יהיה מסמך שיכילו מספר שדות לבחירתך (כולל ערך) אבל יש להגדיר את ה - pi בצורה ידנית

משימה 5

- צור בסיס נתונים חדש בשם blog

- הוסף לבסיס נתונים זה אוסף חדש בשם posts שבו יהיו מסמכים הכוללים את השדות הבאים:

כותרת הפוסט (title), מאמר הפוסט (article), מכתב ע"י (by) שיכיל את שם הכותב, תגובות (comments) שיכיל מערך של מחוזות תום - תשובות גולשים על הפוסט, תאריך יצירת הפוסט (date)

- הערכים של כל אחד מהשדות לבחירתך... אך צור לפחות שני מסמכים!

פתרונות:

use lec1practice

משימה 1

צור אוסף חדש users והגדיר בו שדה email (field) בשם unique (value) ייחודי (unique) באמצעות המתודה createIndex (collection) בשם users

```
db.users.createIndex({email: 1}, {unique: true})
```

משימה 2

- באמצעות המתודה insert הוסף לאוסף users מספר documents חדשים כאשר כל document יכיל את השדות name,email,password,city

```
db.users.insert([{email: 'moe@gmail.com', password: '123456', city: 'tlv', name: 'moe'}])
```

משימה 3

באמצעות המתודה insertMany צור אוסף חדש products (collection) בשם products ובו מסמכים שייכלו שדות כדוגמתם title,article,price וכו'

```
lec1practice > db.products.insertMany([
```

```
    {title: 'title1', price: 20, article: 'article1'},
    {title: 'title2', price: 30, article: 'article2'},
    {title: 'title3', price: 40, article: 'article3'}
```

```
])
```

פתרונות:

use lec1practice

משימה 4

בעזרת המתוודה `insert` צור אוסף חדש בשם `tester` שבו יהיה מסמך שיכילן מספר שדות לבחירתך (כולל ערך) אבל יש להגדיר את ה- `-po` בצוות ידנית

```
db.tester.insert({_id: 1, foo: 'bar'})
```

משימה 5

- הוסף לבסיס נתונים זה אוסף חדש בשם posts שבו יהיו מסמכים הכלולים את השדות הבאים:
 - כותרת הפוסט (title), מאמר הפוסט (article), נכתב ע"י (by) שיכיל את שם הכותב, תגובות (comments) שיכיל מערך של מחרוזות תיימס - תשובות גולשים על הפוסט, תאריך יצירת הפוסט (date)
 - הערכים של כל אחד מהשדות לבחירתך...אך צור לפחות שני מסמכים!
 - צור בסיס נתונים חדש בשם blog

use blogi

```
db.posts.insertMany([
  {
    title: 'A Post About MySQL',
    article: "Querying Data",
    by: "Itamar",
    comments: ["nice", "Good!"],
    date: new Date()
  }
])
```

שיעור בית - להתקין Mongosh

<https://www.mongodb.com/docs/mongodb-shell/install/>

להוסיף את Path mongosh של חלונות

(אפשר לפתח את mongosh מכל מקום במחשב)

[https://www.google.com/search?
q=mongosh+windows&sca_esv=584847290&rlz=1C5CHFA_enIL707IL707&tbo=vid&sxsrf=AM9HkKnoWV5O3s0MGRO_5RNjCl9urAH-LQ:1700744288357&source=lhms&sa=X&ved=2ahUKEwiX2KSZltqCAxXO0glHHTj8CY4Q_AUoAXoECAEQAw&biw=2043&bih=1010&dpr=2](https://www.google.com/search?q=mongosh+windows&sca_esv=584847290&rlz=1C5CHFA_enIL707IL707&tbo=vid&sxsrf=AM9HkKnoWV5O3s0MGRO_5RNjCl9urAH-LQ:1700744288357&source=lhms&sa=X&ved=2ahUKEwiX2KSZltqCAxXO0glHHTj8CY4Q_AUoAXoECAEQAw&biw=2043&bih=1010&dpr=2)

שיעור בית

צרו דטה ביביס חדש בשם lec1hw1 בעזרת הפקודה use

הוסיפו לדטה ביביס אוסף בשם pets

הוסיפו לדטה ביביס אוסף בשם products

הוסיפו לדטה ביביס אוסף בשם users

הציגו את שמות כל האוספים

לאוסף users יש להוסיף אינדקס ייחודי לכתובת האימייל

לאוסף users יש להוסיף 2 משתמשים - לכל משתמש יש שם פרטי, שם משפחה, סיסמה וכתובת אימייל

יש להוסיף 2 מסמכים לאוסף Pets:

לכל חיה יש pi_ לבחירתכם (לא אוטומטי)

לכל חיה יש: סוג החיה, שם חיבת, צבע פרווה, אוכל מועדף

יש להוסיף 3 מסמכים לאוסף products:

לכל מוצר יש שם מוצר, תיאור המוצר, מחיר, קטגוריה ומערך של מילוט מפתח (מערך של מחרוזות)

בחרו בדטה ביביס חדש בשם lec1hw2 בעזרת הפקודה use

צרו אוסף חדש בשם mytest

צרו אוסף חדש בשם mytest2

שנו את שם האוסף mytest ל test

מחקו את האוסף mytest2

מחקו את הדטה ביביס