**إنشاء قاعدة البيانات Collection وإضافة مستندات لها Documents**

تتألف قاعدة البيانات في MongoDB من مجموعات Collections، بداخلها مستندات Documents، بداخلها حقول Fields.

أمر عرض قواعد البيانات الموجودة، والتي تحوي داخلها مجموعة واحدة على الأقل:

show dbs

لإنشاء قاعدة البيانات أو الانتقال لقاعدة بيانات موجودة مسبقًا نستخدم الأمر:

use <name>

نضيف مكان <name> اسم قاعدة البيانات.

لعرض اسم قاعدة البيانات التي نعمل عليها:

db

أو

db.getName()

أمر عرض المجموعات ضمن قاعدة البيانات الحالية:

show collections

لإضافة مستند جديد إلى مجموعة:

db <collections>.insert(<document>)

نضيف مكان <collections> اسم المجموعة ومكان <document> بيانات المستند الجديد بصيغة JSON.

**الاستعلام عن البيانات Find**

أمر عرض جميع المستندات ضمن مجموعة في قاعدة البيانات الحالية:

db.<collection>.find()

نضيف مكان <collection> اسم المجموعة.

لإظهار البيانات بشكل قابل للقراءة، نستدعي التابع `pretty` في نهاية الاستعلام، أو التابع `forEach` ونمرر القيمة `printjson`:

db.<collection>.find().pretty()

أو

db.<collection>.find().forEach(printjson)

نضيف مكان <collection> اسم المجموعة.

أمر عرض البيانات على شكل مصفوفة:

db.<collection>.find().toArray()

نضيف مكان <collection> اسم المجموعة.

يمكن استعمال تعليمات جافاسكربت داخل سطر أوامر MongoDB لمعالجة البيانات.

**المفتاح الأولي Primary Key والدالة Count والدالة Limit**

تُنشئ MongoDB مفتاحًا أوليًا بالاسم `‎\_id` لكل مستند يتم إضافته، وذلك في حال لم يتم تعيينه يدويًا، وطوله 12 بايت ويعرض بالصيغة الست عشرية Hexadecimal.

تتألف قيمة المعُرّف `‎\_id` من قيمة العلامة الزمنية ليونكس UNIX Timestamp لأول 4 بايت، و3 بايت من مُعرّف الجهاز، و 2 بايت من مُعرّف العملية PID، و 3 بايت قيمة عشوائية في كل مرة.

ObjectId("68f88d0ee810658ca73ffe8b"),

لعرض عدد المستندات الموجودة ضمن المجموعة نستخدم التابع `count`:

db.<collection>.count()

نضيف مكان <collection> اسم المجموعة.

لتحديد عدد المستندات ضمن النتيجة نستخدم التابع `limit`:

db.<collection>.find().limit(<number>)

نضيف مكان <collection> اسم المجموعة، ومكان <number> عدد السجلات الذي نريد عرضه.

يعيد التابع `findOne` أول مستند من النتيجة:

db.<collection>.findOne()

نضيف مكان <collection> اسم المجموعة.

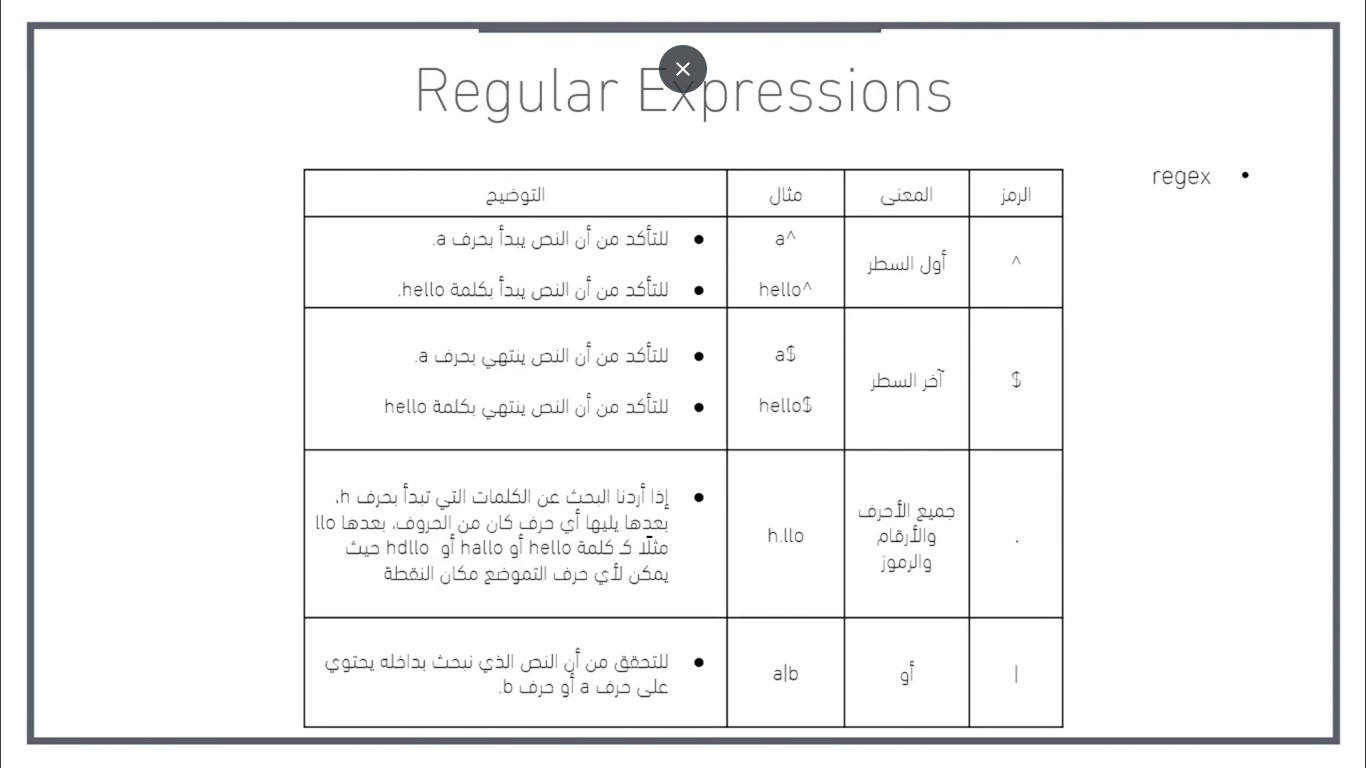
**التعرف على Regular Expressions وبعض الاختصارات المهمة**

**تستخدم التعابير النظامية لتحديد أنماط لمطابقة السلاسل النصية.**

* يدل الرمز `^` على بداية السطر.
* يدل الرمز `$` على نهاية السطر.
* يدل رمز النقطة `.` محرف حرف أو رقم أو رمز.
* يجمع الرمز `|` بين عبارتين، ويعني أو.
* يعبر القوسان `[]` عن تطابق مع أحد المحارف داخلهما، أو مجال من الأحرف أو الأرقام.
* يعبر الرمز `+` على واحد أو أكثر مما قبله.
* يعبر الرمز `?` على صفر أو واحد فقط مما قبله.
* يعبر القوسان `{}` وبداخلهما عدد، على عدد محدد من مرات تكرار ما قبلهما.

**يمكن استخدام عوامل مقارنة وبعدها نقطتان والقيمة التي سنقارن بها، ومن تلك العوامل هي**

* `gt` لأكبر من،
* `lt` لأصغر من،
* `eq` ليساوي،
* `nq` لعدم المساواة،
* `ge` لأكبر أو يساوي،
* `le` لأصغر أو يساوي.





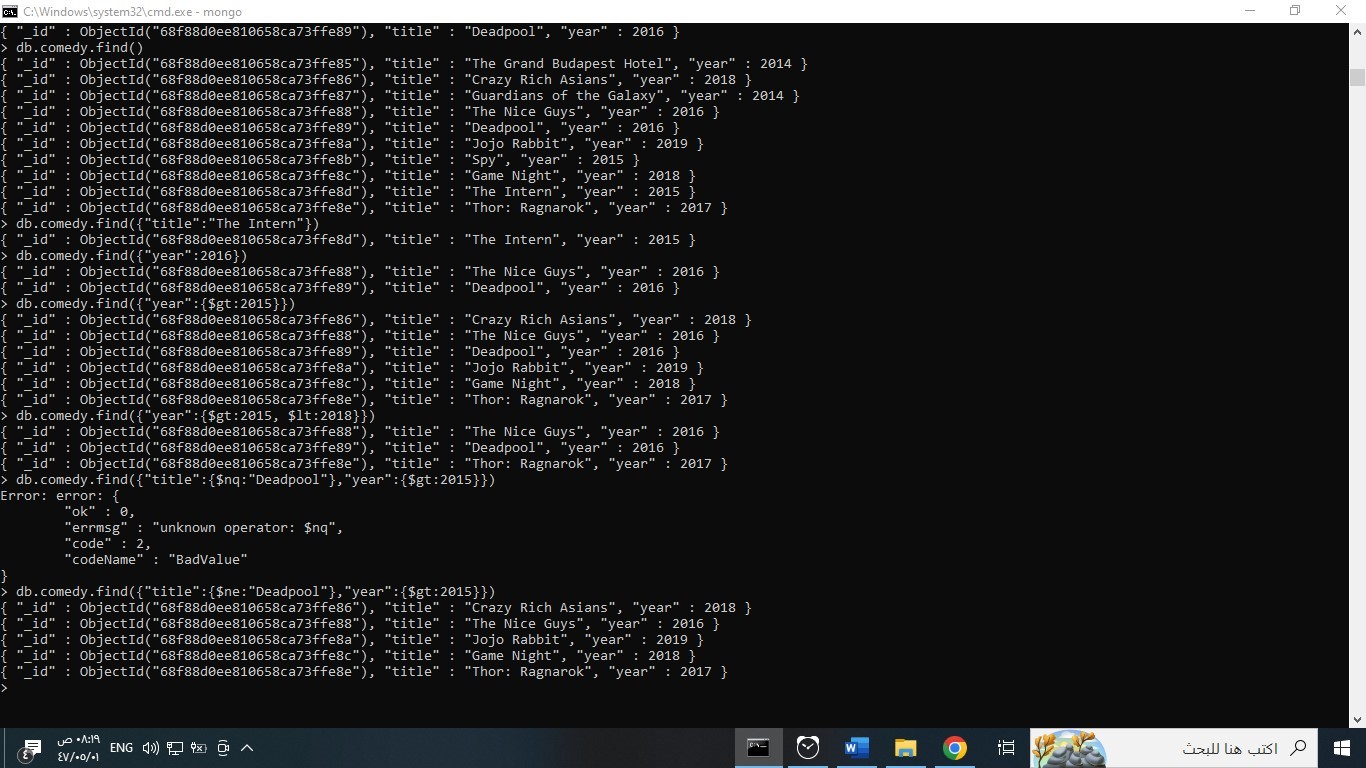


**الاستعلام عن البيانات باستخدام Regular Expressions وعرض بعض الحقول Fields وترتيب النتائج Sort**

يمكن تمرير كائن يعبر عن شرط إظهار البيانات للتابع `find`:

db.<collection>.find({ key1: value, key2: value })

يمكن أن تكون القيم للمفاتيح قيمة صريحة مثل سلسلة نصية أو عدد، أو تكون كائن يحوي مفتاح أو أكثر تبدأ كل منها بالرمز `$` وتعبر عن عوامل المقارنة.



يمكن تمرير تعبير نمطي Regular Expression داخل كائن الاستعلام لمطابقة الحقل معه، وذلك ضمن مفتاح بالاسم `‎$regex`، ويمكن تمرير خيارات مطابقة التعبير النمطي كقيمة للمفتاح `‎$options`:

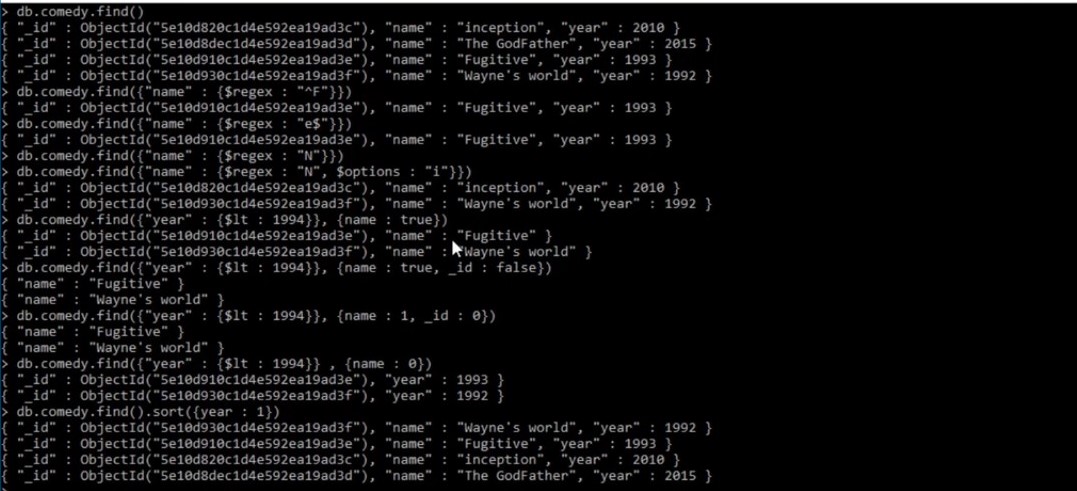
db.<collection>.find({ key: { $regex: "value", $options: "…"})

يمكن تمرير معامل ثاني للتابع `find` نحدد فيه الحقول التي نريد إظهارها ضمن النتيجة بتمرير قيمة منطقية تعبر عن إظهار أو إخفاء الحقل:

db.<collection>.find({ key: value }, { key: true, \_id: false })

يمكن إظهار جميع الحقول عدا حقول معينة، بتمرير تلك الحقول ضمن كائن المعامل الثاني للتابع `find`، وتحديد قيمتها `false` أو `0`.

يرتب التابع `sort` البيانات بحسب الحقول الممررة له، ويكون الترتيب تصاعديًا في حال كانت قيمة الحقل `1` وتنازليًا إذا كانت `-1



تعديل البيانات (الوثائق) UPDATE Documents

يعدل التابع `update` البيانات ضمن المجموعات، فنمرر له الاستعلام كمعامل أول، وكائن يعبر عن التحديث كمعامل ثانِ، وضمن المفتاح `‎$set` نمرر كائن يعبر عن البيانات المُحدثة أو الجديدة:

استعلام إضافة بيانات جديدة لمستندات أو تحديث بياناته:

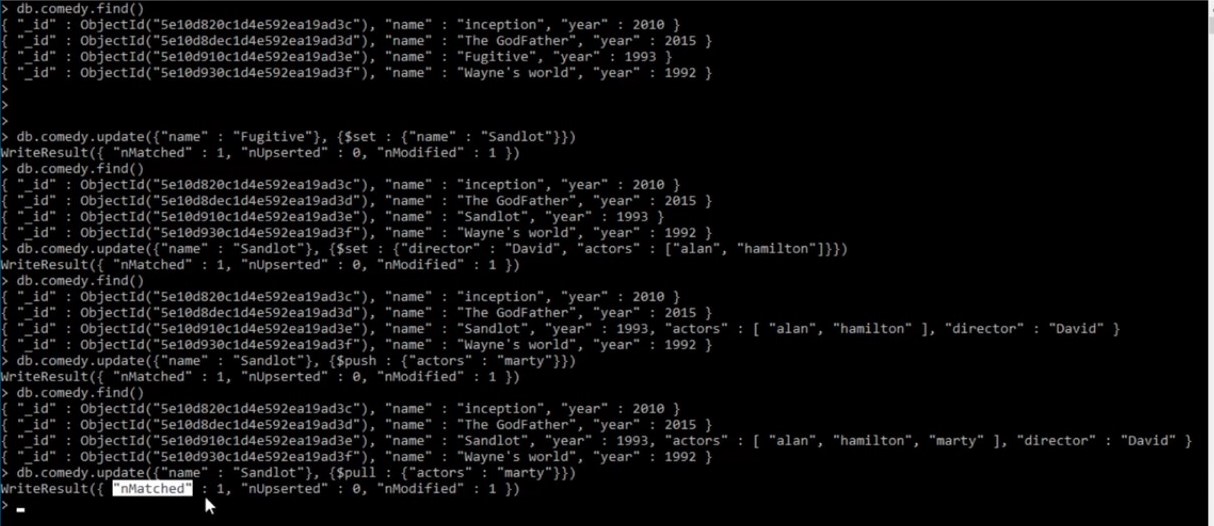
db.<collection>.update({ key: value }, { $set: { key: value } })

يعبر المفتاح `‎$push` عن بيانات نريد إضافتها لقيمة مصفوفة ضمن المستند:

db.<collection>.update({ key: value }, { $push: { key: value } })

يعبر المفتاح `‎$pull` عن بيانات نريد حذفها من قيمة مصفوفة ضمن المستند:

db.<collection>.update({ key: value }, { $pull: { key: value } })



حذف الوثائق Documents والمجموعات Collections وقاعدة البيانات

يحذف التابع `remove` مستندات من المجموعة، التي توافق شرطًا محددًا نمرره له كمعامل أول:

db.<collection>.remove({ key: value })

لحذف جميع المستندات من مجموعة نمرر كائنًا فارغًا:

db.<collection>.remove({})

يحذف التابع `drop` المجموعة بشكل كامل:

db.<collection>.drop()

يحذف التابع `dropDatabase()‎` قاعدة البيانات الحالية:

db.dropDatabase()

لتعديل اسم مجموعة ما collection

db. collection.renameCollection("<collection >")

<collection >" نحط مكانه اسم المجموعة الجديد

**حذف الوثائق Documents والمجموعات Collections وقاعدة البيانات**

يحذف التابع `remove` مستندات من المجموعة، التي توافق شرطًا محددًا نمرره له كمعامل أول:

db.<collection>.remove({ key: value })

لحذف جميع المستندات من مجموعة نمرر كائنًا فارغًا:

db.<collection>.remove({})

يحذف التابع `drop` المجموعة بشكل كامل:

db.<collection>.drop()

يحذف التابع `dropDatabase()‎` قاعدة البيانات الحالية:

db.dropDatabase()

**استيراد البيانات Importing**

يخرج الأمر `exit` من جلسة الاستعلام الحالية في سطر أوامر mongo.

يستورد الأمر `mongoimport` البيانات من الملفات ويدرجها ضمن مستندات جديدة في قاعدة البيانات:

mongoimport --db <db> --collection <collection> --type <type> --file <filename>

مع تبديل القيم <db> باسم قاعدة البيانات، و <collection> باسم المجموعة، و <type> بنوع الملف الذي نحاول استيراده إما json أو csv، و <filename> باسم ملف البيانات، وعند استيراد ملف من نوع csv يجب إضافة الخيار `‎--headerline` قبل الخيار `‎--**file`.**

**تصدير البيانات Exporting**

يُصدر الأمر `mongoexport` البيانات من قاعدة بيانات MongoDB إلى ملف:

mongoexport --db <db> --collection <collection> --type <type> --out <filename>

مع تبديل القيم <db> باسم قاعدة البيانات، و <collection> باسم المجموعة، و <type> بنوع الملف الذي نحاول استيراده إما json أو csv، و <filename> باسم ملف البيانات.

مثال

C:\Users\G.B\Desktop>mongoexport --db Cinema --collection movies --type json --out movies.json

عند تصدير ملف من نوع csv يجب إضافة الخيار `‎--fields` وبعده أسماء الحقول المراد استخراجها مفصول بينها بفاصلة `,` من دون فراغات:

mongoexport --db <db> --collection <collection> --type csv --out <filename> --fields key1,key2,…

مثال

C:\Users\G.B\Desktop>mongoexport --db Cinema --collection movies --type csv --out movies.csv --fields \_id,rank,title,id

**حل مشروع الوحدة**

**استدعي البيانات في مونقو دبي مع أنا عبارة عن مصفوفة من الكائنات**

mongoimport --db ECommerce --collection market --jsonArray --file Market.json

**اعرض الاسم واللون للمنتجات التي أكثر من 90 دولار**

> db.market.find({"Price": {$gt:90} }, { Name: true, "Color":true, \_id: false })

{ "Color" : "green", "Name" : "Sweater" }

{ "Color" : "black", "Name" : "Shoes" }

{ "Color" : "Yellow", "Name" : "Pants" }

{ "Color" : "green", "Name" : "Sweater" }

{ "Color" : "black", "Name" : "Shoes" }

{ "Color" : "Yellow", "Name" : "Pants" }

{ "Color" : "green", "Name" : "Sweater" }

{ "Color" : "black", "Name" : "Shoes" }

{ "Color" : "Yellow", "Name" : "Pants" }

{ "Color" : "Yellow", "Name" : "Pants" }

{ "Color" : "Yellow", "Name" : "Pants" }

**>**