إنشاء قاعدة البيانات Collection وإضافة مستندات لها Documents

تتألف قاعدة البيانات في MongoDB من مجموعاتCollections ، بداخلها مستنداتDocuments ، 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 البياناتFind

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 ١٢ بايت ويعرض بالصيغة الست عشرية Hexadecimal.

تتألف قيمة المعُرّف `_id` من قيمة العلامة الزمنية ليونكس UNIX Timestamp لأول ٤ بايت، و٣ بايت من مُعرّف الجهاز، و ٢ بايت من مُعرّف العملية PID، و ٣ 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.

Regular Expressions

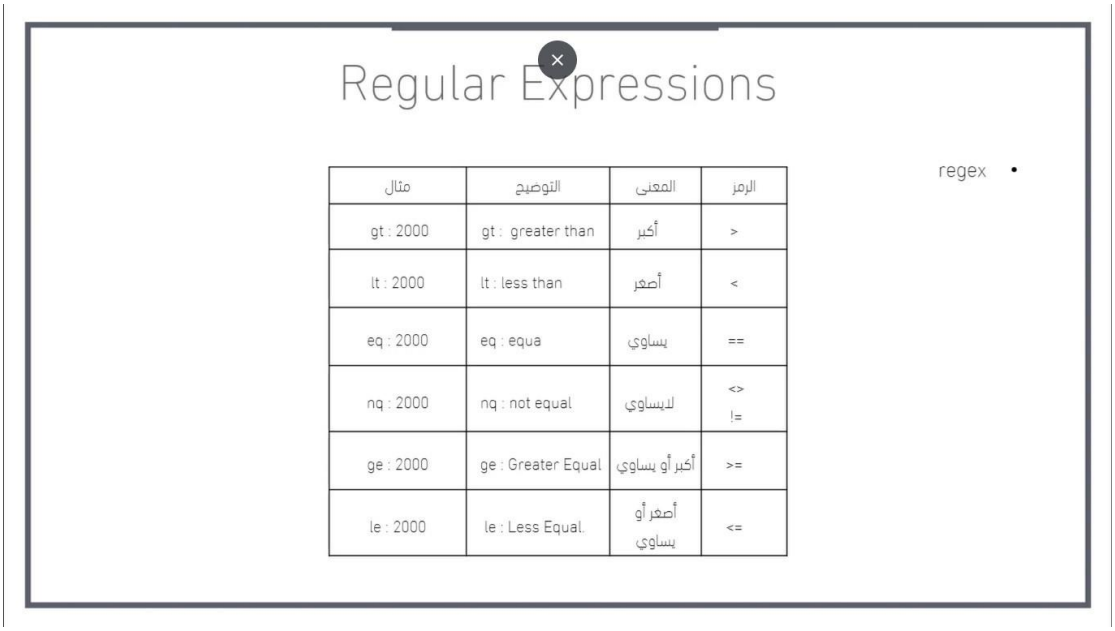
- regex

الرمز	المعنى	مثال	التوضيح
^	أول السطر	a^ hello^	• للتأكد من أن النص يبدأ بحرف a. • للتأكد من أن النص يبدأ بكلمة hello.
$	آخر السطر	a$ hello$	• للتأكد من أن النص ينتهي بحرف a. • للتأكد من أن النص ينتهي بكلمة hello
.	جميع الأحرف والأرقام والرموز	h.llo	• إذا أردنا البحث عن الكلمات التي تبدأ بحرف h، بعدها يليها أي حرف كان من الحروف، بعدها llo مثلاً ك كلمة hello أو hallo أو hdllo حيث يمكن لأي حرف التموضع مكان النقطة
|	أو	a|b	• للتحقق من أن النص الذي نبحث بداخله يحتوي على حرف a أو حرف b.

Regular Expressions

- regex

الرمز	المعنى	مثال	التوضيح
[ ]	أو	[abc] [a-f] [1-9]	• للتحقق من أن النص الذي نبحث فيه يحتوي على حرف a أو حرف b أو حرف c. • للتحقق من أن النص الذي نبحث فيه يحتوي على الأحرف التي بين الحرف a والحرف f • للتحقق من أن النص الذي نبحث فيه يحتوي على أعداد بين العدد 1 والعدد 9
+	يوجد مرة واحدة على الأقل أو أكثر	a+	• للتحقق من أن النص يحتوي على الأقل حرف a واحد أو أكثر.
?	لايوجد أبدًا أو يوجد مرة واحدة فقط	a?	• للتحقق من أن النص لا يحتوي على حرف a أو يحتوي على حرف a واحد فقط.
{n}	تحديد عدد مرات التكرار	a{2}	• الكلمات التي تحتوي على حرف a مكرر مرتين فقط

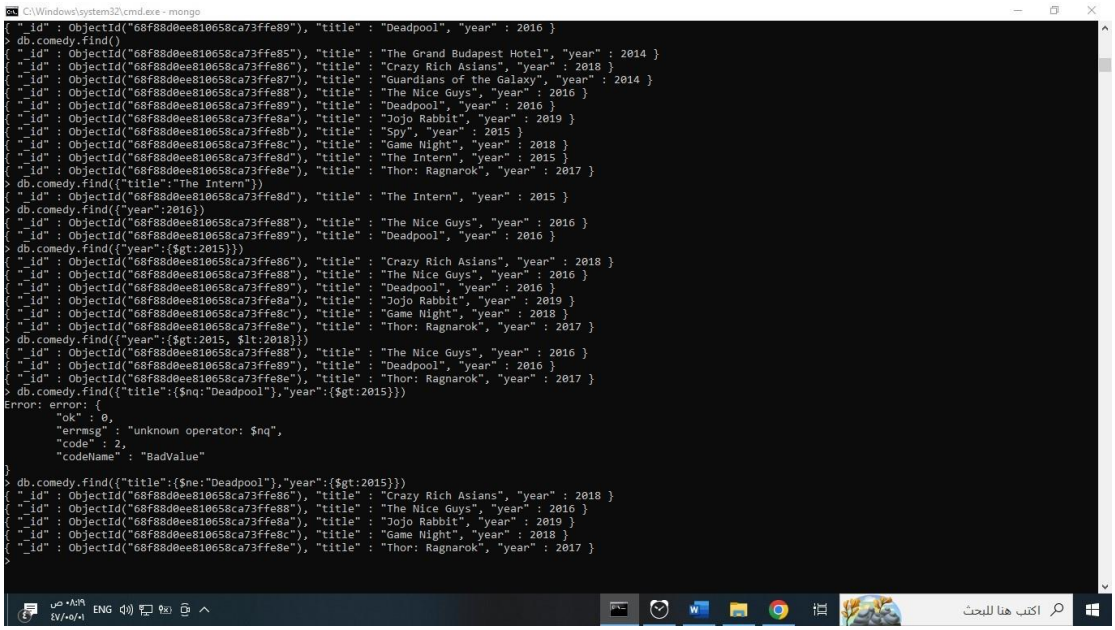


الاستعلام عن البيانات باستخدام Regular Expressions وعرض بعض الحقول Fields وترتيب النتائج Sort

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 `٠`.

البيانات بحسب الحقول الممررة له، ويكون الترتيب تصاعديًا `sort` يرتب التابع في حال كانت قيمة الحقل `١` وتنازليًا إذا كانت `-١`

```
> db.comedy.find()
{ "_id" : ObjectId("5e10d820c1d4e592ea19ad3c"), "name" : "inception", "year" : 2010 }
{ "_id" : ObjectId("5e10d8dec1d4e592ea19ad3d"), "name" : "The GodFather", "year" : 2015 }
{ "_id" : ObjectId("5e10d910c1d4e592ea19ad3e"), "name" : "Fugitive", "year" : 1993 }
{ "_id" : ObjectId("5e10d930c1d4e592ea19ad3f"), "name" : "Wayne's world", "year" : 1992 }
> db.comedy.find({"name" : {$regex : "^F"}})
{ "_id" : ObjectId("5e10d910c1d4e592ea19ad3e"), "name" : "Fugitive", "year" : 1993 }
> db.comedy.find({"name" : {$regex : "e$"}})
{ "_id" : ObjectId("5e10d910c1d4e592ea19ad3e"), "name" : "Fugitive", "year" : 1993 }
> db.comedy.find({"name" : {$regex : "N"}})
> db.comedy.find({"name" : {$regex : "N", $options : "i"}})
{ "_id" : ObjectId("5e10d820c1d4e592ea19ad3c"), "name" : "inception", "year" : 2010 }
{ "_id" : ObjectId("5e10d930c1d4e592ea19ad3f"), "name" : "Wayne's world", "year" : 1992 }
> db.comedy.find({"year" : {$lt : 1994}}, {name : true})
{ "_id" : ObjectId("5e10d910c1d4e592ea19ad3e"), "name" : "Fugitive" }
{ "_id" : ObjectId("5e10d930c1d4e592ea19ad3f"), "name" : "Wayne's world" }
> db.comedy.find({"year" : {$lt : 1994}}, {name : true, _id : false})
{ "name" : "Fugitive" }
{ "name" : "Wayne's world" }
> db.comedy.find({"year" : {$lt : 1994}}, {name : 1, _id : 0})
{ "name" : "Fugitive" }
{ "name" : "Wayne's world" }
> db.comedy.find({"year" : {$lt : 1994}} , {name : 0})
{ "_id" : ObjectId("5e10d910c1d4e592ea19ad3e"), "year" : 1993 }
{ "_id" : ObjectId("5e10d930c1d4e592ea19ad3f"), "year" : 1992 }
> db.comedy.find().sort({year : 1})
{ "_id" : ObjectId("5e10d930c1d4e592ea19ad3f"), "name" : "Wayne's world", "year" : 1992 }
{ "_id" : ObjectId("5e10d910c1d4e592ea19ad3e"), "name" : "Fugitive", "year" : 1993 }
{ "_id" : ObjectId("5e10d820c1d4e592ea19ad3c"), "name" : "inception", "year" : 2010 }
{ "_id" : ObjectId("5e10d8dec1d4e592ea19ad3d"), "name" : "The GodFather", "year" : 2015 }
```

تعديل البيانات (الوثائق) UPDATE Documents

يعدل التابع `update` البيانات ضمن المجموعات، فنمرر له الاستعلام كمعامل أول، وكائن يعبر عن التحديث كمعامل ثانٍ، 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 البياناتImporting

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 ٩٠ 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" }
>
```