

Inlämningsuppgift 3

Vad ni ska skicka in:

- For G:
 - 1 text-fil i plain-text-format med era svar osv.
 - Filen måste vara .txt (pdf, doc, docs, rtf, ... accepteras inte.)
 - Skriv namn och frågor och svar, till example: fråga 1: ... svar : mongosats + resultat fråga 2:
 - svar: mongosats + resultat osv...
- For VG:
 - Samtliga krav för G plus:
 - 1 text-fil i plaintextformat med era svar, och en länk till erat github visual studio projekt.
 - OBS: För VG För att få VG, måste ni klarat G. Om ni inte klarat G och ni får en retur (komplettering), gäller komplettering bara för betyget G även om VG delen var korrekt. Man kan alltså inte få två chanser för VG.
 - Allt måste skickas innan 3/2/2020 klockan 23.50. Om ni skickar efter deadline får ni inte VG även om ni har gjort den delen.
 - Komplettering är endast för G delen. Inte för VG.
 - Koden måste kommenteras om ni ska få VG.

Instruktion

Task1 G-uppgiften

Skriv alla satser och resultat från MongoDB som du använder i följande frågor i en txt fil:

1. Skapa en databas vid namn citydb.
2. Sätt in följande dokument i en kollektion med namn: cities
 - a. `_id:1, name: Stockholm, population: 1372565`
 - b. `_id:2, name: Göteborg, population: 549839`
 - c. `_id:3, name: Malmö, population: 280415`
 - d. `_id:4, name: Uppsala, population: 140454`
 - e. `_id:5, name: Västerås, population: 110877`
3. Hitta ett dokument där namnet är "Malmö" och visa bara "namn" och "population".
4. Uppdatera dokumentet som har "namn" "Göteborg" med "population" 549890
5. Ta bort dokumentet med `_id:4`
6. Uppdatera dokumentet som har "namn" "Västerås" och öka (med increment) "population" fält så att det blir 110879. (increment by 2)
7. Lägg till ett fält "country" och värdet "Sweden" till alla dokument.
8. Byt namn på fältet "country" till "Country".

Task 2

Även här ska ni skriva både satserna och vad mongo svarar (resultat). Ladda ned animals.js från ITHS Distans.

1. Skapa en databas vid namn animalsdb.
2. Gå till databasen animalsdb.
3. Skriv i mongo-skalet : `load ("animals.js")`. Vad gör den raden? Visa resultatet.
4. Hitta alla data från collection animals.
5. Hitta hur många djur det finns totalt i databasen animals?
6. Lägg till en array "favorit_food" med "blueberry, honey och ants" för "Bear" .
7. Ta bort ants och honey från arrayen "favorit_food" och "rounded ears" från arrayen "data" för "Bear".
8. Hitta hur många gula djur det finns i kollektionen.
9. Hitta alla namnet på alla djur vars färg är brun och som lever i Asien och visa deras namn, color och found_in .
10. Lägg till fältet "litter_size" med värdet "4-6-kits" för "Fox".
11. Lägg till ett fält med namn "likes meat" i arrayen "data" för alla med "order": "Carnivora" och som är gula till färgen.
12. Gruppera efter färger och visa antal djur per färg.
13. Skriv ut antalet djur som väger mer än 100.

14. Skriv ut djur med "order" "Carnivora", sorterat efter weight i fallande ordning.
15. Hitta alla dokument som har "mammal" eller "carnivore" i deras "data" fält.
16. Hitta alla dokument som har "mammal" och "carnivore" i deras "data" fält.

Task 3

Även här ska ni skriva både satserna och vad mongo svarar (resultat).

1. Sätt in följande record i kollektion orders:

```
{
  "Id": 100,
  "Name": "Eva",
  "Subscriber": true,
  "Payment": {
    "Type": "Credit-Card",
    "Total": 400,
    "Success": true
  },
  "Note": "1st Complete Record"
},
{
  "Id": 101,
  "Name": "Johan",
  "Subscriber": true,
  "Payment": {
    "Type": "Debit-Card",
    "Total": 500,
    "Success": true
  },
  "Note": null
},
{
  "Id": 102,
  "Name": "Matilda",
  "Subscriber": true,
  "Payment": {
    "Type": "Credit-Card",
    "Total": 700,
    "Success": false
  }
},
{
  "Id": 103,
  "Name": "Mikael",
```

```

    "Subscriber": false,
    "Payment": null,
    "Note": "Payment is Null"
  },
  {
    "Id": 104,
    "Name": "Rikard",
    "Subscriber": false,
    "Payment": {
      "Type": "Debit-Card",
      "Total": 300,
      "Success": false
    },
    "Note": "Completed Payment"
  }
}

```

2. Hitta alla dokument där fältet "Note" är "null" eller saknas .
3. Hitta alla dokument där fältet "Note" saknas.
4. Hitta alla dokument som har "Type": "Credit-Card" sorterat efter "Total" i stigande ordning.
5. Sortera alla dokument efter Total i fallande ordning och visa bara dokument 2 och 3 (från resultatet).
6. Aggregera "Success" och visa antal i fallande ordning. (Hur många "Success" är "true" och hur många är "false").

Exempel på hur svarsfilen för G-uppgiften ska se ut, formmässigt

Namn: Johan Mongosson Task 1. Fråga 1.

Skapa en databas vid namn fruits

> <<här skriver du ditt mongo-kommand>>

Switched to db fruits ← Detta är vad mongoddb svarar! Skall vara med.

Fråga 2. Skapa en collection vid namn "citrus"

> <<här skriver du ditt mongo-kommando>>

{ "ok" : 1 } ← Detta är vad mongoddb svarar, det ska du ta med i din fil.

Fråga 3. Hitta alla dokument i collection "citrus"

> <<här skriver du ditt mongo-kommando>>

{ "_id" : 1, "name" : "orange" } { "_id" : 2, "name" : "lemon" } osv...

VG-uppgiften

Skriv en C# klass (ni bestämmer själva vad den ska hetta).

Ni kan använda lämpliga MongoDB paket ifrån NuGet

Ert program ska:

Ansluta till mongodb (skapa en klient)

Skapa en ny databas via klienten till exempel "lab3"

Skapa en kollektion vid namn "restaurants" med följande innehåll: OBS! Ni ska skapa detta i ert program med C#

```
{
  "_id" : ObjectId("5c39f9b5df831369c19b6bca"),
  "name" : "Sun Bakery Trattoria",
  "stars" : 4,
  "categories" : [ "Pizza", "Pasta", "Italian", "Coffee", "Sandwiches" ]
},
{
  "_id": ObjectId("5c39f9b5df831369c19b6bcb"),
  "name" : "Blue Bagels Grill",
  "stars" : 3,
  "categories" : [
    "Bagels", "Cookies", "Sandwiches"
  ]
},
{
  "_id" : ObjectId("5c39f9b5df831369c19b6bcc"),
  "name" : "Hot Bakery Cafe",
  "stars" : 4,
  "categories" : [
    "Bakery",
    "Cafe",
    "Coffee",
    "Dessert"
  ]
},
{
  "_id" : ObjectId("5c39f9b5df831369c19b6bcd"),
  "name" : "XYZ Coffee Bar",
  "stars" : 5,
  "categories" : [
    "Coffee", "Cafe", "Bakery", "Chocolates"
  ]
}
```

```

    ]
  },
  {
    "_id" : ObjectId("5c39f9b5df831369c19b6bce"),
    "name" : "456 Cookies Shop",
    "stars" : 4,
    "categories" : [
      "Bakery", "Cookies", "Cake", "Coffee"
    ]
  }
}

```

Skriv en metod som skriver ut (Console.WriteLine) alla dokument i samlingen.

Skriv en metod som skriver ut namnet på alla dokument som har kategorin "Cafe"
OBS! Exkludera id så att bara namn visas

Skriv en metod som uppdaterar genom increment "stars" för den restaurang som har "name"
"XYZ Coffee Bar" så att nya värdet på stars blir 6.

OBS! Ni ska använda increment .

OBS! Skriv ut alla restauranger igen, så att jag kan se att "stars" blivit 6, för denna restaurang när jag kör ert program.

Skriv en metod som uppdaterar "name" för "456 Cookies Shop" till "123 Cookies Heaven"

OBS! Skriv ut alla restauranger igen, så att jag kan se att namnet ändrats för denna restaurang när jag kör ert program.

Skriv en metod som aggregerar en lista med alla restauranger som har 4 eller fler "stars" och skriver ut endast "name" och "stars"

OBS! Metoderna ska skriva ut via Console.WriteLine resultatet, det vill säga, när jag kör ert program ska jag se resultatet från utskrifterna.

Exempel på utskriber från ert program Alla dokument:

```

{
  "_id" :
  {
    "$oid" : "5c3a0af0df83137e38e4b0da"
  },
  "name" : "Sun Bakery Trattoria",
  "stars" : 4,
  "categories" : [
    "Pizza", "Pasta", "Italian", "Coffee", "Sandwiches"
  ]
}

```

```
},
{
  "_id" : {
    "$oid" : "5c3a0af0df83137e38e4b0db"
  },
  "name" : "Blue Bagels Grill",
  "stars" : 3,
  "categories" : [
    "Bagels", "Cookies", "Sandwiches"
  ]
},
{
  "_id" : {
    "$oid" : "5c3a0af0df83137e38e4b0dc"
  },
  "name" : "Hot Bakery Cafe",
  "stars" : 4,
  "categories" : [
    "Bakery", "Cafe", "Coffee", "Dessert"
  ]
},
{
  "_id" : {
    "$oid" : "5c3a0af0df83137e38e4b0dd"
  },
  "name" : "XYZ Coffee Bar",
  "stars" : 5,
  "categories" : [
    "Coffee", "Cafe", "Bakery", "Chocolates"
  ]
},
{
  "_id" : {
    "$oid" : "5c3a0af0df83137e38e4b0de"
  },
  "name" : "456 Cookies Shop",
  "stars" : 4,
  "categories" : [
    "Bakery", "Cookies", "Cake", "Coffee"
  ]
}
```


Alla namn på rest. med kategori Cafe:

```
{  "name" : "Hot Bakery Cafe" }
```

```
{  "name" : "XYZ Coffee Bar" }
```

Och så vidare för varje VG-uppgift... Lycka till!