

Alcuni esercizi risolti ed alcuni esercizi da risolvere.....

Primi comandi:

show dbs

use restaurant (prova)

show collections

Codice Lettura:

```
db.restaurants.find() /* digitare it per andare avanti */
```

```
db.restaurants.find({name:"Glorious Food"})
```

```
db.restaurants.find().sort( { restaurant_id: 1 } ).limit(1).pretty()
```

```
/* Count restaurants */
```

```
db.restaurants.count() /* 25359 */
```

```
/* Count restaurants with American (Vegetarian, Chinese) cuisine */
```

```
/* How many different types of cooking */
```

```
db.restaurants.distinct("cuisine").length
```

```
/* Count no Vegetarian restaurants */
```

Codice Inserimento:

```
r = {} /* crea nuovo oggetto */
```

```
/* Create multiple fields for that object and assign a value to them */
```

```
r.restaurant_id = "101"
```

```
r.name = "Da Mario"
```

```
r.cuisine = "Italiano"
```

```
a = {}
```

```
a.building = "1-A"
```

```
/* Create an array and assign it to the field coord of object a */
```

```
a.coord = [ 41.8902102, 12.4922309 ]
```

```
a.street = "Piazza del Colosseo"
```

```
r.address = a
```

```
/* Create an empty array */
```

```
r.grades = []
```

```
/* This is a faster way to add data inside an object. ISODate is an object and handles dates and time */
```

```
g = { date: ISODate("2016-07-31T13:00:00Z"), grade: "A", score: 10 }
```

```
/* Add the object to the last position of the array r.grades */
```

```
r.grades.push(g)
```

```
g = { date: ISODate(), grade: "D", score: 4 }
```

```
r.grades.push(g)
```

```
/* Finally add the created object into restaurants */
```

```
db.restaurants.insert(r)
```

OPPURE

Codice Inserimento:

_____ (provare una insert completa come da
slide)_____

Codice Lettura:

```
/* Find every restaurant with the value of borough equal to  
Brooklyn*/
```

```
/* Show every distinct value for the field borough*/
```

```
db.restaurants.distinct("borough")
```

```
/* Count how many they are */
```

```
/*projection: trovare I nomi dei ristorante di 'Brooklin'*/
```

```
/* Trovare quelli che hanno almeno un grade A ordinandoli per  
nome o per quartiere */
```

E poi con ordine inverso

```
/* grades written after the date "2013-10-01" */
```

```
/* Documents with all the grades but A */
```

```
/* object_id, quartieri e nomi di tutti i ristoranti ordinati per  
cucina */
```

```
/* quartieri e nomi di tutti i ristoranti ordinati per cucina */
```

```
/*name contain string coffee, ignoreCase when a i is added after  
the last "/" */
```

```
db.restaurants.find({name: /coffee/i}, {name: 1, _id: 0}).pretty();
```

```
/*name starts with Starbucks*/
```

```
db.restaurants.find({name: /^Starbucks/}, {name: 1, _id:  
0}).pretty();
```

```
/*name ends with shop*/
```

```
db.restaurants.find({name: /Shop$/}, {name: 1, _id: 0}).pretty();
```

```
/*ignoreCase with i*/
```

```
db.restaurants.find({name: /^starbucks/i}, {name: 1, _id:  
0}).pretty();
```