MongoDB - Esercitazione

# Query semplici (find)

## Restaurants

1. Visualizzare tutti i ristoranti.
   1. **db.restaurants.find()**
2. Visualizzare quartiere (borough) e tipo di cucina (cuisine) di tutti i ristoranti.
   1. **db.restaurants.find({}, {borough: 1, cuisine: 1})**
3. Visualizzare quartiere (borough) e tipo di cucina (cuisine) di tutti i ristoranti, ma senza \_id.
   1. **db.restaurants.find({}, {borough: 1, cuisine: 1, \_id: 0})**
4. Visualizzare quartiere (borough), tipo di cucina (cuisine) e via (address.street) di tutti i ristoranti.
   1. **db.restaurants.find({}, {borough: 1, "cuisine": 1, "address.street": 1})**
5. Visualizzare iI ristorante il cui zipcode è 11225
   1. **db.restaurants.find({"address.zipcode": "11225"})**
6. Visualizzare i ristoranti il cui tipo di cucina è Hamburgers
   1. **db.restaurants.find({cuisine: "Hamburgers"})**
7. Visualizzare i ristoranti il cui tipo di cucina NON è Hamburgers
   1. **db.restaurants.find({cuisine: {$ne: "Hamburgers"} })**
   2. **db.restaurants.find({cuisine: {$not: { $in: ["Hamburgers"] } } })**
8. Visualizzare i ristoranti il cui tipo di cucina è tra Hamburgers, Bakery o Irish
   1. **db.restaurants.find({cuisine: {$in: ["Hamburgers", "Bakery", "Irish"]} })**
   2. **db.restaurants.find({$or: [{cuisine: "Hamburgers"}, {cuisine: "Bakery"}]})**
9. Visualizzare i ristoranti il cui tipo di cucina NON è tra Hamburgers, Bakery o Irish
   1. **db.restaurants.find({cuisine: {$nin: ["Hamburgers", "Bakery", "Irish"]} }**
   2. **db.restaurants.find({cuisine: {$not: {$in: ["Hamburgers", "Bakery", "Irish"]}} })**
10. Visualizzare i ristoranti il cui tipo di cucina non esiste
    1. **db.restaurants.find({cuisine: {$exists: false} })**

## Yelpbusiness

1. Visualizzare tutti i business che hanno ricevuto almeno 10 recensioni (review\_count) E si trovano in Arizona o in Nevada (state = NV o AZ)
2. **db.yelpbusiness.find({review\_count: {$gte: 10 }, state: {$in: ["NV","AZ"]} })**
3. 37113 risultati
4. Visualizzare tutti i business che hanno ricevuto almeno 10 recensioni (review\_count) O si trovano in Arizona o in Nevada (state = NV o AZ)
5. **db.yelpbusiness.find({$or: [{review\_count: {$gte: 10 } }, {state: {$in: ["NV","AZ"]} }]})**
6. 79898 risultati
7. Visualizzare tutti i business che hanno Grocery tra le categorie
   1. **db.yelpbusiness.find({categories: "Grocery"})**
   2. **db.yelpbusiness.find({categories: {$in: ["Grocery"]} })**
   3. 2850 risultati
8. Visualizzare tutti i business che hanno Grocery E Barbeque tra le categorie
   1. **db.yelpbusiness.find({categories: {$all: ["Grocery", "Barbeque"]} })**
   2. 4 risultati
9. Visualizzare tutti i business che hanno Grocery O Barbeque tra le categorie
   1. **db.yelpbusiness.find({categories: {$in: ["Grocery", "Barbeque"]} })**
   2. 4125 risultati
10. Visualizzare tutti i business che hanno SOLAMENTE Grocery E Barbeque tra le categorie
    1. **db.yelpbusiness.find({categories: ["Grocery", "Barbeque"]})**
    2. **db.yelpbusiness.find({categories: {$all: ["Grocery", "Barbeque"], $size: 2} })**
    3. 0 risultati
11. Visualizzare tutti i business che hanno 5 categorie
    1. **db.yelpbusiness.find({categories: {$size: 5 } })**
    2. 17227 risultati
12. Visualizzare tutti i business che hanno 5 categorie e la quinta categoria è Food
    1. **db.yelpbusiness.find({$and: [{categories: {$size: 5 } }, {"categories.4": "Food"} ] })**
    2. **db.yelpbusiness.find({categories: {$size: 5 }, "categories.4": "Food"} )**
    3. 604 risultati
13. Visualizzare le prime due categorie di ogni business
    1. **db.yelpbusiness.find({}, {categories: {$slice: 2} })**
    2. 144072 risultati (tutti: si tratta solo di modificare la proiezione)
14. Visualizzare unicamente le categorie dei business del Nevada
    1. **db.yelpbusiness.find({state: "NV"}, {"categories": 1, "state": 1})**
    2. 28214 risultati
15. Visualizzare i valori distinti del campo city
    1. **db.yelpbusiness.distinct("city")**
    2. 1 array con 878 valori distinti
16. Visualizzare i valori distinti del campo city nello stato del Nevada
    1. **db.yelpbusiness.distinct("city", {state: "NV"})**
    2. 1 array con 41 valori distinti
17. Visualizzare i valori distinti dell’array categories
    1. **db.yelpbusiness.distinct("categories")**
    2. 1 array con 1192 valori distinti

## Games

1. Visualizzare le partite disputate nel 2010.
   1. **db.games.find({date: {$gte: new Date("2010-01-01"), $lte: new Date("2010-12-31")} })**
   2. **db.games.find({$where: function(){ return this.date.getFullYear()==2010 } })**
2. Visualizzare le partite in cui una delle squadre ha totalizzato almeno 160 punti
   1. **db.games.find({teams: {$elemMatch: {score: {$gte: 160} } } })**
   2. **db.games.find({"teams.score": {$gte: 160} })**
3. Visualizzare le partite in cui la squadra di casa ha perso
   1. **db.games.find({teams: {$elemMatch: {won: 0, home: true} } })**
   2. ~~db.games.find({teams: {won: 0, home: true} })~~ match esatto
   3. ~~db.games.find({"teams.won": 0, "teams.home": true})~~ condizioni in OR
4. Visualizzare le partite in cui ha giocato Michael Jordan
   1. **db.games.find({box: {$elemMatch: {players: {$elemMatch: {player: "Michael Jordan"} } } } })**
   2. **db.games.find({box: {$elemMatch: {"players.player": "Michael Jordan"} } })**
   3. **db.games.find({"box.players.player": "Michael Jordan"} )**
5. Visualizzare le partite in cui Michael Jordan ha segnato più di 60 punti
   1. **db.games.find({box: {$elemMatch: {players: {$elemMatch: {player: "Michael Jordan", pts: {$gt: 60} } } } } })**
   2. **db.games.find({"box.players": {$elemMatch: {player: "Michael Jordan", pts: {$gt: 60} } } })**
6. Visualizzare le prime 10 partite memorizzate nella collection
   1. **db.games.find().limit(10)**
7. Visualizzare le prime 10 partite in ordine di data
   1. **db.games.find().sort({date: 1}).limit(10)**
8. Visualizzare le prime 10 partite del 2010 in ordine di data
   1. **db.games.find({date: {$gte: new Date("2010-01-01"), $lte: new Date("2010-12-31")} })  
      .sort({date: 1}).limit(10)**
9. Visualizzare le seconde 10 partite (dalla 11° del 2010 in ordine di data
   1. **db.games.find({date: {$gte: new Date("2010-01-01"), $lte: new Date("2010-12-31")} })  
      .sort({date: 1}).skip(10).limit(10)**

# Indici di testo

## Restaurants

1. Creare un indice di testo sul campo nome
   1. **db.restaurants.createIndex({name: "text"})**
2. Cercare i ristoranti che hanno il termine “baker” nel nome; verificare i risultati
   1. **db.restaurants.find({$text: {$search: "baker" }})**
3. Cercare i ristoranti che hanno il termine “bake” nel nome; verificare i risultati
   1. **db.restaurants.find({$text: {$search: "bake"}})**

## YelpBusiness

1. Creare un indice di testo sul campo categories
   1. **db.yelpbusiness.createIndex({categories: "text"})**
2. Contare i business che utilizzano il termine “service”
   1. **db.yelpbusiness.find({$text: {$search: "service"}})**
3. Contare i business che utilizzano il termine “services”
   1. **db.yelpbusiness.find({$text: {$search: "services"}})**
4. Contare i business che utilizzano i termini “professional” o “service”
   1. **db.yelpbusiness.find({$text: {$search: "professional service"}})**
5. Contare i business che utilizzano i termini “professional” e “service”
   1. **db.yelpbusiness.find({$text: {$search: "\"professional service\""}})**
6. Contare i business che utilizzano il termine “professional” senza “service”
   1. **db.yelpbusiness.find({$text: {$search: "professional -service"}})**

# Query complesse (aggregate: $project e $match)

Vedi file .txt.