MongoDB - Esercitazione (Aggregate)

# $project e $match

## Yelpbusiness

1. Restituire, per ogni business, un campo totalStars ottenuto moltiplicando stars per review\_count

db.yelpbusiness.aggregate([{  
 $project: {  
 "totalStars": {   
 $multiply: [ “$stars”, “$review\_count” ]  
 },  
 "stars": 1,  
 "review\_count": 1  
 }  
}])

1. Restituire, per ogni business, un campo totalStars2 ottenuto moltiplicando stars per review\_count e sottraendo il campo is\_open al risultato
2. Restituire, per ogni business, un campo isNevada che indichi se il business si trova o meno nello stato del Nevada (operatore $eq)
3. Restituire, per i soli business che si trovano in Nevada, il campo totalStars come definito in query 1
4. Restituire, per ogni business, un campo totalStars3 ottenuto moltiplicando stars per review\_count se il business si trova nel Nevada, oppure dividendo stars per review\_count in caso contrario (operatore $cond)

## Games

1. Restituire (in campi separati) l’anno, il mese ed il giorno di ogni partita
2. Restituire un campo che indichi quanti anni fa è stata disputata ciascuna partita

(sottrarre l’anno di *new Date()* dall’anno della partita)

1. Restituire un campo che indichi quanti giorni fa è stata disputata ciascuna partita

(sottrarre *new Date()* dalla data della partita e trasformare da millisecondi a giorni)

1. Restituire, per ogni partita, un campo “team1” ed un campo “team2” che contengono rispettivamente il contenuto del primo e del secondo oggetto all’interno dell’array *teams*

db.games.aggregate([{  
 $project: {  
 "team1": {$arrayElemAt: ["$teams", 0]},  
 "team2": {$arrayElemAt: ["$teams", 1]}­­  
 }  
},{  
 $project: {  
 "diffPunti": {   
 $abs: {   
 $subtract: ["$team1.score","$team2.score"]   
 }  
 },  
 "punti1": "$team1.score",  
 "punti2": "$team2.score"  
 }   
}])

Data la query 9, restituire, per ogni partita, la differenza tra i punteggi delle due squadre (utilizzare l’operatore “$abs”)  
db.games.aggregate([{  
 $project: {  
 "team1": {$arrayElemAt: ["$teams", 0]},  
 "team2": {$arrayElemAt: ["$teams", 1]}­­  
 }  
},{  
 $project: {  
 "diffPunti": {   
 $abs: {   
 $subtract: ["$team1.score","$team2.score"]   
 }  
 },  
 "punti1": "$team1.score",  
 "punti2": "$team2.score"  
 }   
}])

1. Dalla query precedente, restituire anche un campo booleano che indichi se la partita si è conclusa con un punto di differenza (true) o con più punti di differenza (false)
2. Data la query precedente, filtrare solo le partite in cui la differenza è di un unico punto
3. Data la query precedente, filtrare solo le partite in cui la differenza è di un unico punto E in cui ha vinto la squadra di casa

## Restaurants

1. Restituire un unico campo “fullAddress” che metta insieme (nell’ordine indicato) i campi zipcode, street e building contenuti dentro ad address
2. Restituire i primi 3 caratteri del campo borough
3. Restituire i primi 3 caratteri del campo borough in lettere maiuscole

# $group e $sort

## Restaurants

1. Raggruppare per borough
2. Raggruppare per borough e contare i ristoranti
3. Raggruppare per cuisine e contare i ristoranti; ordinare il risultato alfabeticamente
4. Raggruppare per cuisine e contare i ristoranti; ordinare il risultato per numero decrescente di ristoranti

## Yelp-business

1. Raggruppare per state e calcolare il punteggio medio (stars)
2. Raggruppare per city e per state e calcolare il punteggio medio (stars)
3. Raggruppare per state e calcolare il punteggio medio (stars) e la somma di review ricevute
4. Raggruppare per city del Nevada e calcolare il punteggio medio (stars) e la somma di review ricevute
5. Data la query 24, mantenere solo le city che hanno almeno 4.5 di punteggio medio
6. Data la query 24, mantenere solo le city che hanno almeno 4.5 di punteggio medio E almeno 100 review
7. Raggruppare per state ed individuare per ciascuno il punteggio più elevato
8. Raggruppare per state ed individuare per ciascuno il punteggio più elevato, ma senza utilizzare l’operatore $max
9. Raggruppare per state ed individuare per ciascuno l’elenco distinto di città (usando l’operatore $addToSet)

# $unwind

## Games

1. Per ogni partita, creare un campo idPartita che indichi la data di svolgimento e le squadre che hanno giocato (ad esempio, “1985-10-25: Chicago Bulls vs Cleveland Cavaliers)
   1. non richiede l’uso di $unwind
   2. $dateToString: {format: "%Y-%m-%d", date: "$date"}
2. Eseguire l’unwind dell’array teams
3. Eseguire l’unwind dell’array teams e raggruppare per \_ID della partita per ottenere il totale di punti segnati
4. Come la query precedente, ma raggruppare per idPartita (come definito in query 30)
5. Eseguire l’unwind dell’array teams e raggruppare per anno per ottenere il totale di punti segnati
6. Dalla query precedente, raggruppare per anno e squadra per ottenere il totale di punti segnati
7. Dalla query precedente, ordinare per anno e punti segnati (decrescente) per ottenere le classifiche annuali
8. Eseguire l’unwind dell’array box e dell’array players
9. Dalla query precedente, raggruppare per player per ottenere il totale di punti segnati; ordinare il risultato per visualizzare per primi i migliori giocatori del’NBA

## Restaurants

1. Eseguire l’unwind dell’array grades
2. Dalla query 39, raggruppare per borough e ottenere lo score medio
3. Dalla query 39, raggruppare per cuisine e ottenere lo score medio
4. Ripetere la query 41 considerando solo i voti assegnati nel 2015
5. Dalla query 39, raggruppare per anno e contare il numero di voti assegnati

# Esercizi conclusivi

## Yelp-business

1. Individuare i 10 attributes più frequenti
2. Individuare i 10 attributes più frequenti nello stato del Nevada
3. Classificare gli stati in base al numero di business aperti che hanno un punteggio di 5 stelle
4. Individuare le 5 categorie col punteggio medio più alto
5. Elencare, per ogni state, le city in cui sono presenti dei business
6. Sommare il numero di review per ogni state e per ogni city

## Restaurants

1. Individuare, per ogni borough, il punteggio medio ricevuto di anno in anno; ordinare i risultati per borough e per anno
2. Come la query precedente, ma considerare solamente i casi in cui grade = A

## Games

1. Individuare la classifica delle squadre nell’anno 2010
2. Individuare, per ogni anno, la classifica dei giocatori
3. Dalla query precedente, mantenere per ogni anno il solo giocatore che ha totalizzato il punteggio maggiore
4. Come la query precedente, ma calcolare il punteggio di ogni giocatore considerando i campi “fg” (“field goal” = tiro da 2 punti), “fg3” (tiro da 3 punti) e “ft” (“free throw” = tiro libero da 1 punto)