

Operatore	Funzione	Esempio
{key:value}	equal	{weight:100}
\$ne	not equal	{weight:{\$ne: 700}}
\$lt	less than	{weight:{\$lt: 700}}
\$lte	less than or equal	{weight:{\$lte: 700}}
\$gt	greater than	{weight:{\$gt: 700}}
\$gte	greater than or equal	{weight:{\$gte: 700}}
\$not	nega una condizione	{weight : {\$not:{\$gt:200}}}
\$and	effettua una and tra un array di condizioni	{\$and: [{gender:'m'}, {weight:{\$lt:500}}]}
\$or	effettua una or tra un array di condizioni	{\$or: [{gender:'m'}, {weight:{\$lt:500}}]}
\$in	verifica se uno dei valori del vettore soddisfa la condizione <i>Es. unicorni che amano le mele oppure le arance</i>	{ loves: {\$in: ['apple', 'orange'] } }
\$nin	<i>Es. unicorni che non amano né le mele né le arance</i>	{ loves: {\$nin:['apple', 'orange']}}
\$all	<i>Es. unicorni che amano sia le mele sia le arance</i>	{ loves: {\$all:['apple', 'orange']}}
\$pull	elimina da un campo vettoriale la / le voci indicate	{ \$pull: { loves: {\$in:["apple","orange"]} } }
\$exists	usato per verificare la presenza o l'assenza di un campo	{ vampires: {"\$exists": false}}
\$regex	filtrare una porzione di stringa sulla base di una regular expression	{name:{"\$regex" : "\\b[Mm]aria\\b"}}
.project()	consente di specificare l'elenco dei campi da visualizzare	.project({name:1, _id:0})
.sort()	ordinamento dei record (1 = cresc, -1 = decres)	.sort({name: 1, vampires: -1})
.limit()	limitare la quantità di record restituiti	.limit(2)

.skip()	saltare dei record restituiti (all'inizio)	.skip(1)
.count()	contare i record restituiti	.count()
.findOne()	restituisce un solo record, ottimizzando la query	.findOne({ "name": "Aurora" }, {"projection":{"hair":1, "weight":1}},...);
.distinct()	filtra gli elementi duplicati, restituendoli una sola volta	.distinct("loves");
.deleteOne/Many()	Rimuove i record individuati dal filtro	.deleteOne({_id: "a13"})
.update(filtro, campi da aggiornare, opzioni?) (upsert, multi...)	per default aggiorna un solo il 1° record incontrato => updateMany() o opzione multi	.update({name: 'Roodles'}, {\$set: {weight: 590}})
Operatori di update		
\$set{ }	imposta campo indicato al valore indicato.	{\$set: {weight: 590}}
\$unset{ }	rimuove il campo indicato	{\$unset: {vampires: 1}}
\$inc{ }	incrementa / decrementa campi di tipo numerico (sia interi che float)	{\$inc: {vampires: -2}}
\$mul{ }	moltiplica campi di tipo numerico (accetta sia interi che float)	{\$mul: {vampires: 3}}
\$rename{ }	rinomina il campo indicato	{\$rename: {'vampires':'nVampiri'}}
\$push{ }	aggiunge nuovi elementi in campi di tipo vettoriale	{\$push: {loves: 'sugar'}}
\$addToSet{ }	aggiunge nuovi elementi in campi di tipo vettoriale solo se non sono già esistenti	{\$addToSet: {loves: 'sugar'}}
aggregate() NOTA: Quando un nome di campo viene utilizzato come valore a destra dei due punti, occorre anteporre davanti il simbolo del \$		
\$match	applica un filtro sul recordset ricevuto. Stessa sintassi di find()	{ \$match: { status: "A" } }

\$group	raggruppa i record sulla base di uno o più campi indicati	{ \$group: { _id:"\$cust_id", amount: { \$sum:"\$amount" } } }
\$sum, \$avg, \$first (primo elemento di quel gruppo), \$count:{} (dentro group conta gli elementi restituiti, passato con json vuoto) ecc...		
\$project	proiezione sui campi che dovranno essere utilizzati	
\$unwind	consente di “srotolare” un campo costituito da un vettore enumerativo producendo un elenco di N documenti ciascuno contenente una sola delle voci presenti nel vettore	
\$sort	esegue un ordinamento sul recordset finale. Accetta come parametro un JSON avente come chiavi una o più delle chiavi restituite dall’operatore \$group	
\$limit	i due operatori vengono eseguiti nell’ordine in cui sono scritti	{ \$limit:100}
\$skip		{ \$skip:10}
\$expr	eseguire dei confronti fra campi diversi oppure eseguendo a monte una elaborazione sul contenuto del campo	{"\$expr": {"\$gt": ["\$spesa", "\$budget"]}}
Date		
\$year	anno della data	{ \$year: "\$data" }
\$month	mese della data	{ \$month: "\$data" }
\$dayOfMonth	giorno della data	{ \$dayOfMonth: "\$data" }
\$hour	ora	{ \$hour: "\$data" }
\$minute	minuti	{ \$minute: "\$data" }
\$second	secondi	{ \$second: "\$data" }
\$millisecond	millisecondi	{ \$millisecond: "\$data" }
\$dayOfYear	giorno dell’anno	{ \$dayOfYear: "\$data" }
\$dayOfWeek	giorno della settimana	{ \$dayOfWeek: "\$data" }
\$week	settimana dell’anno	{ \$week: "\$data" }