

MongoDB – partea a II-a

Operatorii de comparație

Operatorii logici

Operatorii de actualizare a câmpurilor

Operatorii de actualizare a vectorilor

Tipuri operatori

- **Operatorii de comparație** – filtrează datele pe baza comparațiilor de valori.
- **Operatorii logici** - returnează date pe baza expresiilor care sunt evaluate ca fiind adevărate sau false.
- **Operatorii de actualizare a câmpurilor** - sunt utilizați pentru a realiza modificări asupra câmpurilor din documente.
- **Operatorii de actualizare a vectorilor** - sunt utilizați pentru a realiza modificări asupra vectorilor din documente.

Operatorii de comparație

- returnează (filtrează) date pe baza comparațiilor de valori (egal cu, mai mare, mai mic ș.a.m.d).
- pot fi folosiți în construcții de tipul celor prezentate mai jos, documentul obținut astfel putând fi folosit drept argument pentru metodele din MongoDB.
- *operator* reprezintă unul dintre operatorii de comparație prezentați în tabelul 1.

Forma 1 (primii 6 operatori din tabelul 1)

```
{ câmp: { $operator: valoare } }
```

Forma 2 (ultimii 2 operatori din tabelul 1)

```
{ câmp: { $operator: [val1, val2, ... valN ] } }
```

Tabel 1. Operatori de comparație

Operator	Descriere operator
\$eq	<i>"equal to"</i> - Specifică condiția de egalitate. Operatorul selectează documentele în care valoarea unui câmp este egală cu valoarea specificată.
\$ne	<i>"not equal to"</i> - Selectează documentele în care valoarea câmpului specificat nu este egală cu valoarea specificată. Sunt incluse și documentele care nu conțin câmpul specificat.
\$gt	<i>"greater than"</i> - Selectează documentele în care valoarea câmpului specificat este mai mare decât valoarea specificată.
\$gte	<i>"greater than or equal to"</i> - Selectează documentele în care valoarea câmpului specificat este mai mare sau egală cu valoarea specificată.

Tabel 1. Operatori de comparație

Operator	Descriere operator
\$lt	<i>"less than"</i> - Selectează documentele în care valoarea câmpului specificat este mai mică decât valoarea specificată.
\$lte	<i>"less than or equal to"</i> - Selectează documentele în care valoarea câmpului specificat este mai mică sau egală cu valoarea specificată.
\$in	<i>"in"</i> - Selectează documentele în care valoarea câmpului specificat se află în vectorul specificat.
\$nin	<i>"not in"</i> - Selectează documentele în care valoarea câmpului specificat nu se află în vectorul specificat sau câmpul nu există.

Operatorii de comparație

Exemplu: Se afișează documentele din colecția emp pentru angajații cu post de “MANAGER”, rezultatul interogării fiind ilustrat în imaginea de mai jos.

```
db.emp.find({ job: { $eq: "MANAGER" } })
```

	_id	deptno	empno	ename	hiredate	job	mgr	sal
1	689cba201c570cc05476fb0d	20	7566	JONES	1981-04-02T00:00:00.000Z	MANAGER	7839	2975
2	689cba201c570cc05476fb0f	30	7698	BLAKE	1981-05-01T00:00:00.000Z	MANAGER	7839	2850
3	689cba201c570cc05476fb10	10	7782	CLARK	1981-06-09T00:00:00.000Z	MANAGER	7839	2450

Operatorii de comparație

Exemplu: Se afișează documentele din colecția dept în care numărul departamentului (deptno) este mai mare ca 10, rezultatul interogării fiind ilustrat în imaginea de mai jos.

```
db.dept.find({deptno:{ $gt:10}})
```

	_id	deptno	dname	loc
1	689cb9f41c570cc05476fb00	20	RESEARCH	DALLAS
2	689cb9f41c570cc05476fb01	30	SALES	CHICAGO
3	689cb9f41c570cc05476fb02	40	OPERATIONS	BOSTON
4	68a5c9a4edef6412d4d10857	50	PUBLIC RELATIONS	DALLAS

Operatorii de comparație

Exercițiu: Să se afișează numele, postul și salariul angajaților ce câștigă mai puțin sau egal cu 1600.

```
_id: ObjectId('689cba201c570cc05476fb0a')  
empno : 7369  
ename : "SMITH"  
job : "CLERK"  
mgr : 7902  
hiredate : 1980-12-17T00:00:00.000+00:00  
sal : 850  
deptno : 20
```

?

Operatorii de comparație

Exemplu: Se afișează numele, postul și salariul angajaților ce câștigă mai puțin sau egal cu 1600, rezultatul interogării fiind ilustrat în imaginea de mai jos.

```
db.emp.find(  
  {sal: {$lte: 1600}},  
  {_id: 0, ename: 1, job: 1, sal: 1}  
)
```



	ename ▼ ÷	job ▼ ÷	sal ▼ ÷
1	SMITH	CLERK	850
2	ALLEN	SALESMAN	1600
3	WARD	SALESMAN	1250
4	MARTIN	SALESMAN	1250
5	TURNER	SALESMAN	1500
6	ADAMS	CLERK	1150
7	JAMES	CLERK	1000
8	MILLER	CLERK	1350

Operatorii de comparație

Ultimii doi operatori din tabelul 1, **\$in** și **\$nin**, lucrează cu vectori de valori, deci pot fi folosiți în construcții de tipul prezentat mai jos.

```
{ câmp: { $operator: [val1, val2, ... valN ] } }
```

Operatorii de comparație

Exemplu: Se afișează documentele din colecția `emp` în care numărul angajatului `empno` se regăsește prin valorile din vectorul indicat, rezultatul interogării fiind ilustrat în imaginea de mai jos.

```
db.emp.find({empno: {$in: [7844, 7900, 7934]}})
```

	_id	comm	deptno	empno	ename	hiredate	job	mgr	sal
1	689cba201c570cc05476fb13	0	30	7844	TURNER	1981-09-08T00:00:00.000Z	SALESMAN	7698	1500
2	689cba201c570cc05476fb15	<unset>	30	7900	JAMES	1981-12-03T00:00:00.000Z	CLERK	7698	950
3	689cba201c570cc05476fb17	<unset>	10	7934	MILLER	1982-01-23T00:00:00.000Z	CLERK	7782	1300

Operatorii de comparație

Exemplu: Se afișează din colecția dept documentele corespunzătoare departamentelor care nu au sediul în DALLAS sau BOSTON, rezultatul interogării fiind ilustrat în imaginea de mai jos.

```
db.dept.find({loc: {$nin: ["DALLAS", "BOSTON"]}})
```

	{_id} ▾	÷	{deptno} ▾	÷	{dname} ▾	÷	{loc} ▾	÷
1	689cb9f41c570cc05476faff		10		ACCOUNTING		NEW YORK	
2	689cb9f41c570cc05476fb01		30		SALES		CHICAGO	

Operatorii logici

- returnează date pe baza expresiilor care sunt evaluate ca fiind adevărate sau false.
- pot fi folosiți în construcții de tipul prezentat mai jos, documentul obținut astfel putând fi folosit drept argument pentru metodele din MongoDB.
- *operator* reprezintă unul dintre operatorii logici prezentați în tabelul 2.

```
{ $operator: [ { <expression1> },  
                { <expression2> } , ... ,  
                { <expressionN> }  
              ] }
```

Tabel 2. Operatori logici

Operator	Descriere operator
\$and	Efectuează o operație logică AND pe un șir de una sau mai multe expresii (<expresia1>, <expresia2> și așa mai departe) și selectează documentele care satisfac toate expresiile .
\$or	Efectuează o operație logică OR pe un șir de una sau mai multe expresii și selectează documentele care satisfac cel puțin una dintre ele .
\$nor	Efectuează o operație logică NOR pe un șir de unul sau mai multe predicate de interogare și selectează documentele care nu îndeplinesc niciun predicat de interogare din șir.
\$not	Efectuează o operație logică NOT asupra expresiei specificate și selectează documentele care nu corespund expresiei . Aceasta include și documentele care nu conțin câmpul respectiv.

Operatorii logici

Exemplu: Se afișează din colecția emp documentele asociate angajaților care au numărul angajatului 7900 sau postul de "MANAGER", rezultatul interogării fiind ilustrat în imaginea de mai jos.

```
db.emp.find({$or: [{empno: 7900}, {job: "MANAGER"}]})
```

	{_id ▼ ÷	{deptno ▼ ÷	{empno ▼ ÷	{ename ▼ ÷	{hiredate ▼ ÷	{job ▼ ÷	{mgr ▼ ÷	{sal ▼ ÷
1	689cba201c5...	20	7566	JONES	1981-04-02T00:00:00.000Z	MANAGER	7839	2975
2	689cba201c5...	30	7698	BLAKE	1981-05-01T00:00:00.000Z	MANAGER	7839	2850
3	689cba201c5...	10	7782	CLARK	1981-06-09T00:00:00.000Z	MANAGER	7839	2450
4	689cba201c5...	30	7900	JAMES	1981-12-03T00:00:00.000Z	CLERK	7698	950

Operatorii logici

Exemplu: Se șterg din colecția `emp` documentele corespunzătoare angajaților cu post de “MANAGER” și salariul mai mare de 2500. Rezultatul interogării este ilustrat pe pagina următoare.

```
db.emp.deleteMany(  
  { $and: [  
    { job: "MANAGER" },  
    { sal: { $gt: 2500 } }  
  ]  
})
```


Operatorii logici

	emp_id	deptno	empno	ename	hiredate	job	mgr	sal
1	689cba201c570cc05476fb0a	20	7369	SMITH	1980-12-17T00:00:00.000Z	CLERK	7902	850
2	689cba201c570cc05476fb0b	30	7499	ALLEN	1981-02-20T00:00:00.000Z	SALESMAN	7698	1600
3	689cba201c570cc05476fb0c	30	7521	WARD	1981-02-22T00:00:00.000Z	SALESMAN	7698	1250
4	689cba201c570cc05476fb0d	20	7566	JONES	1981-04-02T00:00:00.000Z	MANAGER	7839	2975
5	689cba201c570cc05476fb0e	30	7654	MARTIN	1981-09-28T00:00:00.000Z	SALESMAN	7698	1250
6	689cba201c570cc05476fb0f	30	7698	BLAKE	1981-05-01T00:00:00.000Z	MANAGER	7839	2850
7	689cba201c570cc05476fb10	10	7782	CLARK	1981-06-09T00:00:00.000Z	MANAGER	7839	2450
8	689cba201c570cc05476fb11	20	7788	SCOTT	1982-12-09T00:00:00.000Z	ANALYST	7566	3000
9	689cba201c570cc05476fb12	10	7839	KING	1981-11-17T00:00:00.000Z	PRESIDENT	<unset>	5000
10	689cba201c570cc05476fb13	30	7844	TURNER	1981-09-08T00:00:00.000Z	SALESMAN	7698	1500
11	689cba201c570cc05476fb14	20	7876	ADAMS	1983-01-12T00:00:00.000Z	CLERK	7788	1150
12	689cba201c570cc05476fb15	30	7900	JAMES	1981-12-03T00:00:00.000Z	CLERK	7698	1000
13	689cba201c570cc05476fb16	20	7902	FORD	1981-12-03T00:00:00.000Z	ANALYST	7566	3000
14	689cba201c570cc05476fb17	10	7934	MILLER	1982-01-23T00:00:00.000Z	CLERK	7782	1350



	emp_id	deptno	empno	ename	hiredate	job	mgr	sal
1	689cba201c570cc05476fb0a	20	7369	SMITH	1980-12-17T00:00:00.000Z	CLERK	7902	850
2	689cba201c570cc05476fb0b	30	7499	ALLEN	1981-02-20T00:00:00.000Z	SALESMAN	7698	1600
3	689cba201c570cc05476fb0c	30	7521	WARD	1981-02-22T00:00:00.000Z	SALESMAN	7698	1250
4	689cba201c570cc05476fb0e	30	7654	MARTIN	1981-09-28T00:00:00.000Z	SALESMAN	7698	1250
5	689cba201c570cc05476fb10	10	7782	CLARK	1981-06-09T00:00:00.000Z	MANAGER	7839	2450
6	689cba201c570cc05476fb11	20	7788	SCOTT	1982-12-09T00:00:00.000Z	ANALYST	7566	3000
7	689cba201c570cc05476fb12	10	7839	KING	1981-11-17T00:00:00.000Z	PRESIDENT	<unset>	5000
8	689cba201c570cc05476fb13	30	7844	TURNER	1981-09-08T00:00:00.000Z	SALESMAN	7698	1500
9	689cba201c570cc05476fb14	20	7876	ADAMS	1983-01-12T00:00:00.000Z	CLERK	7788	1150
10	689cba201c570cc05476fb15	30	7900	JAMES	1981-12-03T00:00:00.000Z	CLERK	7698	1000
11	689cba201c570cc05476fb16	20	7902	FORD	1981-12-03T00:00:00.000Z	ANALYST	7566	3000
12	689cba201c570cc05476fb17	10	7934	MILLER	1982-01-23T00:00:00.000Z	CLERK	7782	1350

Operatorii logici

Exercițiu: Să se afișeze numele și postul pentru primul angajat care nu are postul de “CLERK” sau salariul mai mare ca 1500.

```
_id: ObjectId('689cba201c570cc05476fb0a')  
empno : 7369  
ename : "SMITH"  
job : "CLERK"  
mgr : 7902  
hiredate : 1980-12-17T00:00:00.000+00:00  
sal : 850  
deptno : 20
```

?

Operatorii logici

Exercițiu: Să se afișeze numele și postul pentru primul angajat care nu are postul de “CLERK” sau salariul mai mare ca 1500.

	emp_id	deptno	empno	ename	hiredate	job	mgr	sal
1	689cba201c570cc05476fb0a	20	7369	SMITH	1980-12-17T00:00...	CLERK	7902	850
2	689cba201c570cc05476fb0b	30	7499	ALLEN	1981-02-20T00:00...	SALESMAN	7698	1600
3	689cba201c570cc05476fb0c	30	7521	WARD	1981-02-22T00:00...	SALESMAN	7698	1250
4	689cba201c570cc05476fb0d	20	7566	JONES	1981-04-02T00:00...	MANAGER	7839	2975
5	689cba201c570cc05476fb0e	30	7654	MARTIN	1981-09-28T00:00...	SALESMAN	7698	1250
6	689cba201c570cc05476fb0f	30	7698	BLAKE	1981-05-01T00:00...	MANAGER	7839	2850
7	689cba201c570cc05476fb10	10	7782	CLARK	1981-06-09T00:00...	MANAGER	7839	2450
8	689cba201c570cc05476fb11	20	7788	SCOTT	1982-12-09T00:00...	ANALYST	7566	3000
9	689cba201c570cc05476fb12	10	7839	KING	1981-11-17T00:00...	PRESIDENT	<unset>	5000
10	689cba201c570cc05476fb13	30	7844	TURNER	1981-09-08T00:00...	SALESMAN	7698	1500
11	689cba201c570cc05476fb14	20	7876	ADAMS	1983-01-12T00:00...	CLERK	7788	1150
12	689cba201c570cc05476fb15	30	7900	JAMES	1981-12-03T00:00...	CLERK	7698	1000
13	689cba201c570cc05476fb16	20	7902	FORD	1981-12-03T00:00...	ANALYST	7566	3000
14	689cba201c570cc05476fb17	10	7934	MILLER	1982-01-23T00:00...	CLERK	7782	1350

Operatorii logici

Exercițiu: Să se afișeze numele și postul pentru primul angajat care nu are postul de "CLERK" sau salariul mai mare ca 1500.

Varianta 1

```
db.emp.findOne (
```

```
  { $nor: [  
    { job: { $eq: "CLERK" } },  
    { sal: { $gt: 1500 } }  
  ] },  
  { _id: 0, ename: 1, job: 1 }  
)
```

	ename	job
1	WARD	SALESMAN

Operatorii logici

Exercițiu: Să se afișeze numele și postul pentru primul angajat care nu are postul de "CLERK" sau salariul mai mare ca 1500.

Varianta 2

```
db.emp.findOne (  
    { $and: [  
        { job: { $ne: "CLERK" } },  
        { sal: { $lte: 1500 } }  
    ] },  
    { _id: 0, ename: 1, job: 1 }  
)
```

	ename	job
1	WARD	SALESMAN

Operatorii de actualizare

- Operatorii de actualizare sunt folosiți prin intermediul unor construcții ca cea de mai jos.
- Documentul obținut poate fi folosit drept argument pentru metodele de actualizare din MongoDB.

```
{  
  <operator1>:  
    { <field1>: <value1>, ... },  
  <operator2>:  
    { <field2>: <value2>, ... },  
  ...  
}
```

Exemplu: { \$set: { dname: "PUBLIC RELATIONS" } }

Operatorii de actualizare

Operatorii de actualizare sunt de 3 tipuri:

- Operatori de actualizare a câmpurilor
- Operatori de actualizare a vectorilor
- Operatori de actualizare la nivel de bit

Tabel 3. Operatori de actualizare a câmpurilor

Operator	Descriere operator
\$currentDate	Setează valoarea câmpului la data curentă, fie ca dată calendaristică (Date), fie ca marcaj temporal (Timestamp). Tipul implicit este Date .
\$inc	Incrementează valoarea câmpului cu o valoare specificată.
\$min	Actualizează valoarea câmpului la valoarea specificată dacă valoarea specificată este mai mică decât valoarea curentă a câmpului.
\$max	Actualizează valoarea câmpului la valoarea specificată dacă valoarea specificată este mai mare decât valoarea curentă a câmpului.

Tabel 3. Operatori de actualizare a câmpurilor

Operator	Descriere operator
\$mul	Înmulțește valoarea unui câmp cu un număr.
\$rename	Redenumeste câmpul.
\$set	Setează valoarea unui camp dintr-un document.
\$setOnInsert	Setează valoarea unui câmp dacă actualizarea are ca rezultat inserarea unui document. Nu are niciun efect asupra operațiunilor de actualizare care modifică documentele existente.
\$unset	Elimină câmpul din document.

Operatorii de actualizare a câmpurilor

Exemplu: Se înlocuiește numele departamentului "PR" cu "PUBLIC RELATIONS" în colecția dept, prin modificarea valorii câmpului dname.

```
db.dept.updateOne(  
  { dname: "PR" },  
  { $set: { dname: "PUBLIC RELATIONS" } }  
)
```

Operatorii de actualizare a câmpurilor

	🔗_id 🔽	÷	🔗 deptno 🔽	÷	🔗 dname 🔽	÷	🔗 loc 🔽	÷
1	689cb9f41c570cc05476faff		10		ACCOUNTING		NEW YORK	
2	689cb9f41c570cc05476fb00		20		RESEARCH		DALLAS	
3	689cb9f41c570cc05476fb01		30		SALES		CHICAGO	
4	689cb9f41c570cc05476fb02		40		OPERATIONS		BOSTON	
5	691199a4094ecd45c2f32333		50		PR		DALLAS	



	🔗_id 🔽	÷	🔗 deptno 🔽	÷	🔗 dname 🔽	÷	🔗 loc 🔽	÷
1	689cb9f41c570cc05476faff		10		ACCOUNTING		NEW YORK	
2	689cb9f41c570cc05476fb00		20		RESEARCH		DALLAS	
3	689cb9f41c570cc05476fb01		30		SALES		CHICAGO	
4	689cb9f41c570cc05476fb02		40		OPERATIONS		BOSTON	
5	68a5c9a4edef6412d4d10857		50		PUBLIC RELATIONS		DALLAS	

Operatorii de actualizare a câmpurilor

Exemplu: Se modifică toate documentele din colecția dept, și anume se împarte numărul departamentului deptno la 10, rezultatul obținut fiind ilustrat în imaginea de mai jos.

```
db.dept.updateMany( {}, { $mul: { deptno: 0.1 } } )
```

	_id	deptno	dname	loc
1	689cb9f41c570cc05476faff	10	ACCOUNTING	NEW YORK
2	689cb9f41c570cc05476fb00	20	RESEARCH	DALLAS
3	689cb9f41c570cc05476fb01	30	SALES	CHICAGO
4	689cb9f41c570cc05476fb02	40	OPERATIONS	BOSTON
5	68a5c9a4edef6412d4d10857	50	PUBLIC RELATIONS	DALLAS



	_id	deptno	dname	loc
1	689cb9f41c570cc05476faff	1	ACCOUNTING	NEW YORK
2	689cb9f41c570cc05476fb00	2	RESEARCH	DALLAS
3	689cb9f41c570cc05476fb01	3	SALES	CHICAGO
4	689cb9f41c570cc05476fb02	4	OPERATIONS	BOSTON
5	691199a4094ecd45c2f32333	5	PR	DALLAS

Operatorii de actualizare a câmpurilor

Exemplu: Se mărește valoarea salariului cu 150 pentru angajatul cu codul (empno) 7876.

```
db.emp.updateOne (  
    { empno: 7876 },           // selectează angajatul 7876  
    { $inc: { sal: 150 } }    // mărește salariul cu 150  
)
```

Operatorii de actualizare a câmpurilor

	emp_id	deptno	empno	ename	hiredate	job	mgr	sal
1	689cba201c570cc05476fb0a	20	7369	SMITH	1980-12-17T00:00:00.000Z	CLERK	7902	850
2	689cba201c570cc05476fb0b	30	7499	ALLEN	1981-02-20T00:00:00.000Z	SALESMAN	7698	1600
3	689cba201c570cc05476fb0c	30	7521	WARD	1981-02-22T00:00:00.000Z	SALESMAN	7698	1250
4	689cba201c570cc05476fb0d	20	7566	JONES	1981-04-02T00:00:00.000Z	MANAGER	7839	2975
5	689cba201c570cc05476fb0e	30	7654	MARTIN	1981-09-28T00:00:00.000Z	SALESMAN	7698	1250
6	689cba201c570cc05476fb0f	30	7698	BLAKE	1981-05-01T00:00:00.000Z	MANAGER	7839	2850
7	689cba201c570cc05476fb10	10	7782	CLARK	1981-06-09T00:00:00.000Z	MANAGER	7839	2450
8	689cba201c570cc05476fb11	20	7788	SCOTT	1982-12-09T00:00:00.000Z	ANALYST	7566	3000
9	689cba201c570cc05476fb12	10	7839	KING	1981-11-17T00:00:00.000Z	PRESIDENT	<unset>	5000
10	689cba201c570cc05476fb13	30	7844	TURNER	1981-09-08T00:00:00.000Z	SALESMAN	7698	1500
11	689cba201c570cc05476fb14	20	7876	ADAMS	1983-01-12T00:00:00.000Z	CLERK	7788	1150
12	689cba201c570cc05476fb15	30	7900	JAMES	1981-12-03T00:00:00.000Z	CLERK	7698	1000
13	689cba201c570cc05476fb16	20	7902	FORD	1981-12-03T00:00:00.000Z	ANALYST	7566	3000
14	689cba201c570cc05476fb17	10	7934	MILLER	1982-01-23T00:00:00.000Z	CLERK	7782	1350



	emp_id	deptno	empno	ename	hiredate	job	mgr	sal
1	689cba201c570cc05476fb0a	20	7369	SMITH	1980-12-17T00:00:00.000Z	CLERK	7902	850
2	689cba201c570cc05476fb0b	30	7499	ALLEN	1981-02-20T00:00:00.000Z	SALESMAN	7698	1600
3	689cba201c570cc05476fb0c	30	7521	WARD	1981-02-22T00:00:00.000Z	SALESMAN	7698	1250
4	689cba201c570cc05476fb0d	20	7566	JONES	1981-04-02T00:00:00.000Z	MANAGER	7839	2975
5	689cba201c570cc05476fb0e	30	7654	MARTIN	1981-09-28T00:00:00.000Z	SALESMAN	7698	1250
6	689cba201c570cc05476fb0f	30	7698	BLAKE	1981-05-01T00:00:00.000Z	MANAGER	7839	2850
7	689cba201c570cc05476fb10	10	7782	CLARK	1981-06-09T00:00:00.000Z	MANAGER	7839	2450
8	689cba201c570cc05476fb11	20	7788	SCOTT	1982-12-09T00:00:00.000Z	ANALYST	7566	3000
9	689cba201c570cc05476fb12	10	7839	KING	1981-11-17T00:00:00.000Z	PRESIDENT	<unset>	5000
10	689cba201c570cc05476fb13	30	7844	TURNER	1981-09-08T00:00:00.000Z	SALESMAN	7698	1500
11	689cba201c570cc05476fb14	20	7876	ADAMS	1983-01-12T00:00:00.000Z	CLERK	7788	1300
12	689cba201c570cc05476fb15	30	7900	JAMES	1981-12-03T00:00:00.000Z	CLERK	7698	1000
13	689cba201c570cc05476fb16	20	7902	FORD	1981-12-03T00:00:00.000Z	ANALYST	7566	3000
14	689cba201c570cc05476fb17	10	7934	MILLER	1982-01-23T00:00:00.000Z	CLERK	7782	1350

Operatorii de actualizare a câmpurilor

Exemplu: Se schimbă numele câmpului `dname` cu `dept_name` pentru toate documentele din colecția `dept`.

```
db.dept.updateMany(  
    {},  
    { $rename : { "dname" : "dept_name" } }  
)
```

Observații:

- Noul nume trebuie să fie diferit de numele existent al câmpului.
- Operația se efectuează în doi pași: se șterge câmpul vechi (`$unset`) și se adaugă câmpul nou (`$set`) cu valoarea câmpului vechi.
- Operatorul `$rename` poate muta câmpuri **în** și **din** documente imbricate.

Operatorii de actualizare a câmpurilor

	id	deptno	dname	loc
1	689cb9f41c570cc05476faff	10	ACCOUNTING	NEW YORK
2	689cb9f41c570cc05476fb00	20	RESEARCH	DALLAS
3	689cb9f41c570cc05476fb01	30	SALES	CHICAGO
4	689cb9f41c570cc05476fb02	40	OPERATIONS	BOSTON



	id	dept_name	deptno	loc
1	689cb9f41c570cc05476faff	ACCOUNTING	10	NEW YORK
2	689cb9f41c570cc05476fb00	RESEARCH	20	DALLAS
3	689cb9f41c570cc05476fb01	SALES	30	CHICAGO
4	689cb9f41c570cc05476fb02	OPERATIONS	40	BOSTON

Obs. Se poate și așa...

```
db.dept.updateMany(  
  {},  
  { $rename: { 'dname' : 'dept_name' } }  
)
```


Operatorii de actualizare a câmpurilor

Exemplu: Se înlocuiește salariul cu valoarea maximă dintre salariul curent și 1100.

```
db.emp.updateMany( {}, { $max: { sal: 1100 } } )
```

	emp_id ▾	deptno ▾	empno ▾	ename ▾	hiredate ▾	job ▾	mgr ▾	sal ▾
1	689cba201c570cc05476fb0a	20	7369	SMITH	1980-12-17T00:00:00.000Z	CLERK	7902	850
2	689cba201c570cc05476fb0b	30	7499	ALLEN	1981-02-20T00:00:00.000Z	SALESMAN	7698	1600
3	689cba201c570cc05476fb0c	30	7521	WARD	1981-02-22T00:00:00.000Z	SALESMAN	7698	1250
4	689cba201c570cc05476fb0d	20	7566	JONES	1981-04-02T00:00:00.000Z	MANAGER	7839	2975
5	689cba201c570cc05476fb0e	30	7654	MARTIN	1981-09-28T00:00:00.000Z	SALESMAN	7698	1250
6	689cba201c570cc05476fb0f	30	7698	BLAKE	1981-05-01T00:00:00.000Z	MANAGER	7839	2850
7	689cba201c570cc05476fb10	10	7782	CLARK	1981-06-09T00:00:00.000Z	MANAGER	7839	2450
8	689cba201c570cc05476fb11	20	7788	SCOTT	1982-12-09T00:00:00.000Z	ANALYST	7566	3000
9	689cba201c570cc05476fb12	10	7839	KING	1981-11-17T00:00:00.000Z	PRESIDENT	<unset>	5000
10	689cba201c570cc05476fb13	30	7844	TURNER	1981-09-08T00:00:00.000Z	SALESMAN	7698	1500
11	689cba201c570cc05476fb14	20	7876	ADAMS	1983-01-12T00:00:00.000Z	CLERK	7788	1150
12	689cba201c570cc05476fb15	30	7900	JAMES	1981-12-03T00:00:00.000Z	CLERK	7698	1000
13	689cba201c570cc05476fb16	20	7902	FORD	1981-12-03T00:00:00.000Z	ANALYST	7566	3000
14	689cba201c570cc05476fb17	10	7934	MILLER	1982-01-23T00:00:00.000Z	CLERK	7782	1350

Operatorii de actualizare a câmpurilor

Rezultatul este prezentat mai jos.

```
db.emp.updateMany({ }, { $max: { sal: 1100 } })
```

	🔗_id 🔽	÷ 🔗 deptno 🔽	÷ 🔗 empno 🔽	÷ 🔗 ename 🔽	÷ 🔗 hiredate 🔽	÷ 🔗 job 🔽	÷ 🔗 mgr 🔽	÷ 🔗 sal 🔽
1	689cba201c570cc05476fb0a	20	7369	SMITH	1980-12-17T00:00:00.000Z	CLERK	7902	1100
2	689cba201c570cc05476fb0b	30	7499	ALLEN	1981-02-20T00:00:00.000Z	SALESMAN	7698	1600
3	689cba201c570cc05476fb0c	30	7521	WARD	1981-02-22T00:00:00.000Z	SALESMAN	7698	1250
4	689cba201c570cc05476fb0d	20	7566	JONES	1981-04-02T00:00:00.000Z	MANAGER	7839	2975
5	689cba201c570cc05476fb0e	30	7654	MARTIN	1981-09-28T00:00:00.000Z	SALESMAN	7698	1250
6	689cba201c570cc05476fb0f	30	7698	BLAKE	1981-05-01T00:00:00.000Z	MANAGER	7839	2850
7	689cba201c570cc05476fb10	10	7782	CLARK	1981-06-09T00:00:00.000Z	MANAGER	7839	2450
8	689cba201c570cc05476fb11	20	7788	SCOTT	1982-12-09T00:00:00.000Z	ANALYST	7566	3000
9	689cba201c570cc05476fb12	10	7839	KING	1981-11-17T00:00:00.000Z	PRESIDENT	<unset>	5000
10	689cba201c570cc05476fb13	30	7844	TURNER	1981-09-08T00:00:00.000Z	SALESMAN	7698	1500
11	689cba201c570cc05476fb14	20	7876	ADAMS	1983-01-12T00:00:00.000Z	CLERK	7788	1150
12	689cba201c570cc05476fb15	30	7900	JAMES	1981-12-03T00:00:00.000Z	CLERK	7698	1100
13	689cba201c570cc05476fb16	20	7902	FORD	1981-12-03T00:00:00.000Z	ANALYST	7566	3000
14	689cba201c570cc05476fb17	10	7934	MILLER	1982-01-23T00:00:00.000Z	CLERK	7782	1350

Operatorii de actualizare a câmpurilor

Exemplu: Se înlocuiește salariul cu valoarea maximă dintre salariul curent și 1100.

O variantă alternativă de rezolvare:

```
db.emp.updateMany(  
    {sal: {$lt: 1100}},  
    {$set: {sal: 1100}}  
)
```

Operatorii de actualizare a câmpurilor

Exemplu: Se șterge câmpul cu numele `loc` din documentele colecției `dept`.

```
db.dept.updateMany({}, {$unset: {"loc": ""}})
```

	<code>_id</code>	<code>deptno</code>	<code>dname</code>	<code>loc</code>
1	689cb9f41c570cc05476faff	10	ACCOUNTING	NEW YORK
2	689cb9f41c570cc05476fb00	20	RESEARCH	DALLAS
3	689cb9f41c570cc05476fb01	30	SALES	CHICAGO
4	689cb9f41c570cc05476fb02	40	OPERATIONS	BOSTON



	<code>_id</code>	<code>deptno</code>	<code>dname</code>
1	689cb9f41c570cc05476faff	10	ACCOUNTING
2	689cb9f41c570cc05476fb00	20	RESEARCH
3	689cb9f41c570cc05476fb01	30	SALES
4	689cb9f41c570cc05476fb02	40	OPERATIONS

Operatorii de actualizare a câmpurilor

Exemplu: Se șterge câmpul cu numele `loc` din primul document din colecția `dept` unde câmpul `loc` are valoarea **“CHICAGO”**.

```
db.dept.updateOne({loc: "CHICAGO"},  
                  {$unset: {"loc": ""}})
```

	_id	deptno	dname	loc
1	689cb9f41c570cc05476faff	10	ACCOUNTING	NEW YORK
2	689cb9f41c570cc05476fb00	20	RESEARCH	DALLAS
3	689cb9f41c570cc05476fb01	30	SALES	CHICAGO
4	689cb9f41c570cc05476fb02	40	OPERATIONS	BOSTON
5	68a5c9a4edef6412d4d10857	50	PUBLIC RELATIONS	DALLAS



	_id	deptno	dname	loc
1	689cb9f41c570cc05476faff	10	ACCOUNTING	NEW YORK
2	689cb9f41c570cc05476fb00	20	RESEARCH	DALLAS
3	689cb9f41c570cc05476fb01	30	SALES	<unset>
4	689cb9f41c570cc05476fb02	40	OPERATIONS	BOSTON
5	68a5c9a4edef6412d4d10857	50	PUBLIC RELATIONS	DALLAS

Operatorii de actualizare a câmpurilor

```
_id: ObjectId('689cb9f41c570cc05476faff')  
deptno : 10  
dname : "ACCOUNTING"  
loc : "NEW YORK"
```

```
_id: ObjectId('689cb9f41c570cc05476fb00')  
deptno : 20  
dname : "RESEARCH"  
loc : "DALLAS"
```

```
_id: ObjectId('689cb9f41c570cc05476fb01')  
deptno : 30  
dname : "SALES"
```

```
_id: ObjectId('689cb9f41c570cc05476fb02')  
deptno : 40  
dname : "OPERATIONS"  
loc : "BOSTON"
```

```
_id: ObjectId('68a5c9a4edef6412d4d10857')  
deptno : 50  
dname : "PUBLIC RELATIONS"  
loc : "DALLAS"
```

Operatorii de actualizare a vectorilor

Operatorii de actualizare a vectorilor sunt utilizați pentru a realiza modificări asupra vectorilor din documente. Cei mai reprezentativi operatori de actualizare a vectorilor sunt prezentați în tabelul de mai jos.

Operator	Descriere operator
\$addToSet	Adaugă o valoare la un vector, cu excepția cazului în care valoarea este deja prezentă în vector
\$pop	Elimină primul sau ultimul element dintr-un vector
\$pull	Elimină toate elementele dintr-un vector care se potrivesc cu o interogare specifică
\$push	Adaugă un element într-un vector
\$pullAll	Elimină dintr-un vector toate valorile care se potrivesc

Operatorii de actualizare a vectorilor

- Sintaxa operatorului **\$push** este următoarea:

```
{ $push: { <câmp1>: <valoare1>, ... } }
```

unde

- **<câmp1>** - reprezintă câmpul de tip vector în care se adaugă valoarea;
- **<valoare1>** - este valoarea ce va fi adăugată în câmpul de tip vector.

Operatorii de actualizare a vectorilor

Se presupune existența colecției phone în baza de date emp_dept_salgrade. Colecția conține documente ce indică pentru fiecare angajat care sunt numerele de telefon ale acestuia, ca în imaginea de mai jos.

empno	numbers
7902	["0732096123", "0732096455", "0732096980"]
7844	["0732021644", "0732021123"]

Operatorii de actualizare a vectorilor

Exemplu: Se modifică în colecția `phone` primul documentul asociat angajatului cu numărul 7844 prin adăugarea unui număr de telefon în vectorul `numbers`.

```
db.phone.updateOne(  
  { empno: 7844 },  
  { $push: { numbers: "0722095211" } }  
)
```

empno	numbers
7902	["0732096123", "0732096455", "0732096980"]
7844	["0732021644", "0732021123"]



empno	numbers
7902	["0732096123", "0732096455", "0732096980"]
7844	["0732021644", "0722123123", "0722095211"]

Operatorii de actualizare a vectorilor

- Sintaxa operatorului **\$pop** este următoarea:

{ \$pop: { <câmp>: <-1 | 1>, ... } }

unde

- **<câmp>** - reprezintă câmpul de tip vector din care se șterge elementul;
- **<-1 | 1>** - indică ce element va fi șters din vector.
 - **-1** - indică ștergerea primului element din vector
 - **1** - indică ștergerea ultimului element din vector

Operatorii de actualizare a vectorilor

Exemplu: Se modifică colecția `phone` prin ștergerea ultimului număr de telefon din vectorul `numbers` pentru toate documentele din colecție.

```
db.phone.updateMany( {}, { $pop: { numbers: 1 } } )
```

	<code>_id</code>	<code>empno</code>	<code>numbers</code>
1	68a82e6b092095a82182b29c	7902	["0732096123", "0732096455", "0732096980"]
2	68a83089092095a82182b29e	7844	["0732021644", "0722123123", "0722095211"]



	<code>_id</code>	<code>empno</code>	<code>numbers</code>
1	68a82e6b092095a82182b29c	7902	["0732096123", "0732096455"]
2	68a83089092095a82182b29e	7844	["0732021644", "0722123123"]

Operatorii de actualizare a vectorilor

- Sintaxa operatorului **\$pull** este următoarea:

```
{ $pull: {  
    <câmp1>: <valoare1|condiție1>,  
    <câmp2>: <valoare2|condiție2>, ... } }
```

unde

- **<câmp1>, <câmp2>** - reprezintă câmpuri de tip vector din care se șterg valori;
- **<valoare1>, <valoare2>** - sunt valorile ce vor fi șterse din vector;
- **<condiție1>, <condiție2>** - sunt condițiile ce trebuie îndeplinite de valorile ce vor fi șterse din vector.

Operatorii de actualizare a vectorilor

Exemplu: Se modifică colecția phone prin ștergerea numărului de telefon “0722095211” din vectorul numbers.

```
db.phone.updateMany(  
    {},  
    { $pull: { numbers: "0722095211" } }  
)
```

	{_id}	÷	{empno}	÷	{numbers}	÷
1	68a82e6b092095a82182b29c		7902		["0732096123", "0732096455", "0732096980"]	
2	68a83089092095a82182b29e		7844		["0732021644", "0722123123", "0722095211"]	



	{_id}	÷	{empno}	÷	{numbers}	÷
1	68a82e6b092095a82182b29c		7902		["0732096123", "0732096455", "0732096980"]	
2	68a83089092095a82182b29e		7844		["0732021644", "0722123123"]	

Operatorii de actualizare a vectorilor

Exemplu: Se modifică colecția `phone` prin ștergerea numerelor de telefon “0732096123” și “0722095211” din vectorul `numbers` din documentele colecției unde aceste numere apar.

```
db.phone.updateMany(  
  {},  
  { $pull: { numbers: { $in: ["0732096123", "0722095211"] } } }  
)
```

	_id	empno	numbers
1	68a82e6b092095a82182b29c	7902	["0732096123", "0732096455", "0732096980"]
2	68a83089092095a82182b29e	7844	["0732021644", "0722123123", "0722095211"]



	_id	empno	numbers
1	68a82e6b092095a82182b29c	7902	["0732096455", "0732096980"]
2	68a83089092095a82182b29e	7844	["0732021644", "0722123123"]

Operatorii de actualizare a vectorilor

- Sintaxa operatorului **\$pullAll** este următoarea:

```
{ $pullAll:  
  { <câmp1>: [<valoare1>, <valoare2> ... ], ... }  
}
```

unde

- **<câmp1>** - reprezintă câmpul de tip vector din care se șterg valorile;
- **<valoare1>**, **<valoare2>** - sunt valorile ce vor fi șterse din câmpul de tip vector.

Operatorii de actualizare a vectorilor

Exemplu: Se modifică colecția `phone` prin ștergerea numerelor de telefon “0732021644” și “0722095211” din vectorul `numbers` din documentele colecției.

```
db.phone.updateMany(  
  {},  
  { $pullAll: { numbers: ["0732021644", "0722095211"] } }  
)
```

	_id	empno	numbers
1	68a82e6b092095a82182b29c	7902	["0732096123", "0732096455", "0732096980"]
2	68a83089092095a82182b29e	7844	["0732021644", "0722123123", "0722095211"]



	_id	empno	numbers
1	68a82e6b092095a82182b29c	7902	["0732096455", "0732096980", "0732096123"]
2	68a83089092095a82182b29e	7844	["0722123123"]

Operatorii de actualizare a vectorilor

Observație: Ambele comenzi de mai jos au același efect asupra colecției `phone`, și anume ștergerea numerelor de telefon “0732096123” și “0722095211” din vectorul `numbers` din documentele colecției.

```
db.phone.updateMany(  
  {},  
  { $pull: { numbers: { $in: ["0732096123", "0722095211"] } } }  
)
```

```
db.phone.updateMany(  
  {},  
  { $pullAll: { numbers: ["0732021644", "0722095211"] } }  
)
```

Operatorii de actualizare a vectorilor

*Operatorul de actualizare **\$pull**:*

- Funcționează dacă după “:” este un tablou sau un obiect.
- Funcționează chiar dacă se utilizează orice operator de condiție în sintaxa interogării.
- Este utilizat pentru a elimina elementele care îndeplinesc o anumită condiție.

*Operatorul de actualizare **\$pullAll**:*

- Funcționează numai dacă după “:” este un tablou.
- Funcționează numai dacă nu se utilizează niciun operator de condiție în sintaxa interogării.
- Este utilizat atunci când trebuie să eliminăm anumite elemente din vector (sunt specificate).

Operatorii de actualizare a vectorilor

Se presupune existența colecției `stores`. Colecția conține documente ce indică pentru fiecare magazin ce legume și ce fructe sunt de vânzare.

<code>_id</code>	<code>fruits</code>	<code>vegetables</code>
1	["apples", "pears", "oranges", "grapes", "bananas"]	["carrots", "celery", "squash", "carrots"]
2	["plums", "kiwis", "oranges", "bananas", "apples"]	["broccoli", "zucchini", "carrots", "onions"]

Operatorii de actualizare a vectorilor

Colecția `stores` a fost creată cu instrucțiunea de mai jos.

```
db.stores.insertMany([
  {
    _id: 1,
    fruits: ["apples", "pears", "oranges", "grapes", "bananas"],
    vegetables: ["carrots", "celery", "squash", "carrots"]
  },
  {
    _id: 2,
    fruits: ["plums", "kiwis", "oranges", "bananas", "apples"],
    vegetables: ["broccoli", "zucchini", "carrots", "onions"]
  }
])
```

Operatorii de actualizare a vectorilor

Exerciții:

- Să se șteargă din vectorul **vegetables** din documentul cu identificatorul **1** prima apariție a elementului **"carrots"**.
- Să se șteargă din vectorul **fruits** din documentele colecției **stores** elementele **"bananas"** și **"apples"**.
- Să se adauge în colecția **stores** un document cu următorul conținut:

```
{ _id: 3,  
  fruits: ["plums", "bananas", "kiwis", "oranges"],  
  vegetables: ["onions", "zucchini", "carrots"] }
```
- Să se adauge în vectorul **fruits** din documentele cu **id** mai mare sau egal cu **2** ale colecției **stores** elementele **"pears"** și **"grapes"**.