Laborator 13 – Operatori de agregare

1. Obiective

- Crearea fluxurilor de agregare;
- Utilizarea operatorilor de agregare.

2. Fluxuri de agregare

Operaţiunile de agregare procesează mai multe documente şi returnează rezultatele calculate. Operaţiunile de agregare pot fi folosite pentru a:

- grupa împreună valorile din mai multe documente;
- efectua operaţiuni asupra datelor grupate pentru a returna un singur rezultat;
- analiza modificările datelor în timp.

Există 2 abordări în realizarea operaţiunilor de agregare:

- *fluxuri de lucru de agregare* – reprezintă metoda preferată pentru efectuarea agregărilor;
- *metode de agregare cu scop unic* - sunt simple, dar nu au capacităţile unui flux de lucru de agregare.

Un flux de lucru de agregare constă din una sau mai multe etape care procesează documente:

- Fiecare etapă efectuează o operaţie asupra documentelor de intrare. De exemplu, o etapă poate filtra documente, grupa documente şi calcula valori.
- Documentele care sunt generate de o etapă sunt transmise către etapa următoare.
- Un flux de lucru de agregare poate returna rezultate pentru grupuri de documente. De exemplu, poate returna valorile totale, medii, maxime şi minime.

În *Tabelul 1* sunt prezentaţi cei mai importanţi operatori de agregare.

Tabel 1. Operatori agregare

Operator	Descriere operator
`$match`	Filtrează documentele pe baza unui predicat de interogare specificat. Documentele rezultate sunt transmise către următoarea etapă a procesului.
`$group`	Combină mai multe documente cu acelaşi câmp, câmpuri sau expresie într-un singur document, în funcţie de o cheie de grup. Rezultatul este un document pentru fiecare cheie de grup unică.
`$project`	Transmite documentele cu câmpurile solicitate către etapa următoare din fluxul de lucru. Câmpurile specificate pot fi câmpuri existente din documentele de intrare sau câmpuri nou calculate.
`$sort`	Sortează documentele de la intrarea în etapa respectivă şi le returnează în fluxul de agregare.
`$limit`	Limitează numărul de documente transmise către etapa următoare din fluxul de lucru.
`$sum`	Calculează şi returnează suma colectivă a valorilor numerice. `$sum` ignoră valorile non-numerice.
`$count`	Trimite către etapa următoare un document care conţine numărul de documente introduse în etapa respectivă.
`$lookup`	Efectuează o joncţiune externă la stânga cu o colecţie din aceeaşi bază de date pentru a filtra documentele din colecţia externă în vederea procesării. Etapa `$lookup` adaugă un nou câmp de tip vector la fiecare document de intrare.
`$out`	Preia documentele returnate de fluxul de agregare şi le scrie în colecţia specificată. Se poate specifica baza de date de ieşire.
`$unwind`	Descompune un câmp de tip vector din documentele de intrare şi *generează un document pentru fiecare element din vector*. Fiecare document de ieşire este format din documentul de intrare în care valoarea câmpului de tip vector este înlocuită cu elementul din vector.

2.1. $match

Sintaxa operatorului de agregare `$match` este următoarea:

```
{ $match: { predicat_interogare } }
```

unde *predicat_interogare* reprezintă predicatul după care se face filtrarea documentelor şi are sintaxa similară cu cea a predicatelor de interogare folosite drept argument în metoda `find()`.

Etapa `$match` elimină un document din rezultatul fluxului de agregare dacă *predicat_interogare* returnează o valoare `0`, nulă sau falsă pentru documentul respectiv, sau predicatul utilizează un câmp care lipseşte din documentul respectiv.

Exemplu: Se returnează toate documentele din colecţia `emp` care au valoarea `20` pentru câmpul `deptno`, rezultatul fiind ilustrat în imagine.

```
db.emp.aggregate([{$match:{deptno:20}}])
```

	{} _id	{} deptno	{} empno	{} ename	{} hiredate	{} job	{} mgr	{} sal
1	689cba201c570cc05476fb0a	20	7369	SMITH	1980-12-17T00:00:00.000Z	CLERK	7902	850
2	689cba201c570cc05476fb0d	20	7566	JONES	1981-04-02T00:00:00.000Z	MANAGER	7839	2975
3	689cba201c570cc05476fb11	20	7788	SCOTT	1982-12-09T00:00:00.000Z	ANALYST	7566	3000
4	689cba201c570cc05476fb14	20	7876	ADAMS	1983-01-12T00:00:00.000Z	CLERK	7788	1150
5	689cba201c570cc05476fb16	20	7902	FORD	1981-12-03T00:00:00.000Z	ANALYST	7566	3000

2.2. $group

Sintaxa operatorului de agregare `$group` este prezentată mai jos, unde

- `_id` - este câmp obligatoriu ce indică câmpul după care se realizează gruparea documentelor. Dacă pentru acest câmp este specificată o valoare nulă sau orice altă valoare constantă, `$group` returnează un singur document, calculat pentru toate documentele de intrare.
- *câmp_nou1* - este câmp opţional ce va apărea în documentul rezultat, câmp calculat prin intermediul operatorilor de tip acumulator (grup). Lista cu cei mai importanţi operatori de tip acumulator poate fi găsită în notiţele de curs.

```
{ $group: {
    _id: expresie, // Cheie grupare
    câmp_nou1: { op_acumulator1: expresie1 },
    ...
} }
```

Exemplu: Se afișează din colecția `emp` media salariilor pentru fiecare tip de job, rezultatul fiind ilustrat în imagine.

```
db.emp.aggregate([
        {$group:
                {_id:"$job",
                avg_sal:{$avg:"$sal"}
                }
        }
])
```

	{} _id	{} avg_sal
1	MANAGER	2758.3333333333335
2	PRESIDENT	5000
3	SALESMAN	1400
4	CLERK	1087.5
5	ANALYST	3000

2.3. $project

Sintaxa operatorului de agregare `$project` este următoarea:

```
{ $project: { specificații } }
```

Operatorul acceptă drept parametru un document care poate specifica ce câmpuri să fie incluse în realizarea operației de proiecție, suprimarea câmpului `_id`, adăugarea de câmpuri noi și resetarea valorilor câmpurilor existente. Alternativ, se poate specifica ce câmpuri să fie excluse.

Cele mai folosite forme pentru specificații sunt prezentate mai jos și indică dacă se dorește includerea sau excluderea de la operația de proiecție a câmpului `câmp`

sau a câmpului `_id`. *`val_logică`* poate lua valorile `0` sau `false` (pentru excludere) respectiv `1` sau `true` (pentru includere).

- `câmp`: *`val_logică`*
- `_id`: *`val_logică`*

Exemplu: Se afişează doar numele departamentelor din documentele colecţiei `dept`, rezultatul fiind ilustrat în imagine.

```
db.dept.aggregate([{$project:{_id:0,dname:1}}])
```

	{} dname
1	ACCOUNTING
2	RESEARCH
3	SALES
4	OPERATIONS
5	PUBLIC RELATIONS

2.4. $sort

Sintaxa operatorului de agregare `$sort` este prezentată mai jos.

```
{ $sort: { câmp1: tip_s, câmp2: tip_s ... } }
```

unde

`câmp` – indică acel câmp în funcţie de care se face sortarea documentelor;

`tip_s` – indică tipul sortării (1 – sortare crescătoare, -1 – sortare descrescătoare).

Exemplu: Se aranjează în ordine crescătoare după locaţie, documentele din colecţia `dept`, rezultatul fiind ilustrat în imagine.

```
db.dept.aggregate([{$sort:{loc:1}}])
```

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

2.5. $limit

Sintaxa operatorului de agregare `$limit` este următoarea:

```
{ $limit: nr_întreg }
```

unde *nr_întreg* reprezintă numărul maxim de documente transmise către etapa următoare din fluxul de agregare.

Exemplu: Sunt afişate două numere de departamente din colecţia `dept`. Toate numerele de departamente sunt obţinute prin gruparea documentelor din colecţia `dept` după câmpul `deptno`. Rezultatul este ilustrat în figura de mai jos.

```
db.emp.aggregate([
                    {$group:{_id:"$deptno"}},
                    {$limit:2}
                    ])
```

	_id
1	20
2	10

2.6. $count

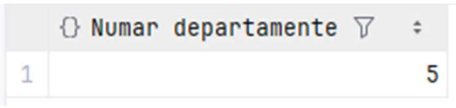
Sintaxa operatorului de agregare `$count` este următoarea:

```
{ $count: nume_câmp }
```

unde *nume_câmp* este un şir de caractere ce reprezintă numele câmpului de ieşire ce va avea valoarea returnată de operator.

Exemplu: Se afișează numărul documentelor din colecția `dept`.

```
db.dept.aggregate([
        {$count:"Numar departamente"}
])
```



	Numar departamente
1	5

2.7. $lookup

Etapa `$lookup` adaugă un nou câmp de tip vector la fiecare document de intrare. Noul câmp de tip vector conține documentele corespunzătoare din colecția externă. Etapa `$lookup` transmite aceste documente remodelate către etapa următoare. Sintaxa operatorului de agregare `$lookup` este prezentată mai jos.

```
{$lookup:{
      from: colectie_externa,
      localField: camp_intern,
      foreignField: camp_extern,
      let: { var_1: expr1, …, var_n: exprn },
      pipeline: [ agregare ],
      as: nume_câmp_iesire
}}
```

unde:

- *colectie_externa* - reprezintă numele colecției externe cu care se face joncțiunea;
- *camp_intern* - câmp din documentele din colecția internă (colecție primită de la stadiul anterior din fluxul de agregare);
- *camp_extern* - câmp din documentele din colecția externă (cu care se face joncțiunea);
- *let* - specifică variabilele care vor fi utilizate în etapele fluxului de agregare. Este opțional. Utilizați expresiile variabile pentru a accesa câmpurile din documentele colecției locale care sunt introduse în fluxul de agregare.
- *agregare* - indică un flux de agregare ce va fi rulat asupra colecției externe; câmpul **`pipeline`** este opțional dacă **`localField`** și **`foreignField`** sunt specificați;

- *nume_câmp_iesire* - indică numele câmpului de tip vector ce va fi adăugat documentelor de la intrarea în fluxul de agregare. El conține documentele din colecția externă ce corespund unui document din colecția internă în urma realizării joncțiunii. **as** este câmp obligatoriu.

Exemplu: Se realizează o echi-joncțiune între colecțiile `dept` și `emp`, prin intermediul câmpului `deptno` care se regăsește în ambele colecții. Documentele de la ieșirea fluxului de agregare au un câmp de tip vector, numit `rezultat`.

```
db.dept.aggregate([
        {$lookup:{
                from:"emp",
                localField:"deptno",
                foreignField:"deptno",
                as:"rezultat"}
        }
])
```

Rezultatul fluxului de agregare este prezentat mai jos.

	{} _id	{} deptno	{} dname	{} loc	{} rezultat
1	689cb9f41c570cc05476faff	10	ACCOUNTING	NEW YORK	[{"_id": new ObjectId("689cba201c570cc05476fb10"), "empno": new NumberInt("7782"), "ename": "CLARK",
2	689cb9f41c570cc05476fb00	20	RESEARCH	DALLAS	[{"_id": new ObjectId("689cba201c570cc05476fb0a"), "empno": new NumberInt("7369"), "ename": "SMITH",
3	689cb9f41c570cc05476fb01	30	SALES	CHICAGO	[{"_id": new ObjectId("689cba201c570cc05476fb0b"), "empno": new NumberInt("7499"), "ename": "ALLEN",
4	689cb9f41c570cc05476fb02	40	OPERATIONS	BOSTON	[]
5	68a5c9a4edef6412d4d10857	50	PUBLIC RELATIONS	DALLAS	[]

Mai jos este prezentată forma JSON a primului document obținut în urma agregării (pentru `deptno:10`), formă obținută în MongoDB Compass pentru o mai bună vizualizare a câmpului rezultat.

```
{
  _id: ObjectId('689cb9f41c570cc05476faff'),
  deptno: 10,
  dname: 'ACCOUNTING',
  loc: 'NEW YORK',
  rezultat: [
    {
      id: ObjectId('689cba201c570cc05476fb10'),
      empno: 7782,
      ename: 'CLARK',
      job: 'MANAGER',
      mgr: 7839,
      hiredate: 1981-06-09T00:00:00.000Z,
      sal: 2450,
      deptno: 10
    },
    {
      id: ObjectId('689cba201c570cc05476fb12'),
      empno: 7839,
      ename: 'KING',
      job: 'PRESIDENT',
```

```
      hiredate: 1981-11-17T00:00:00.000Z,
      sal: 5000,
      deptno: 10
    },
    {
      id: ObjectId('689cba201c570cc05476fb17'),
      empno: 7934,
      ename: 'MILLER',
      job: 'CLERK',
      mgr: 7782,
      hiredate: 1982-01-23T00:00:00.000Z,
      sal: 1350,
      deptno: 10
    }
  ]
}
```

2.8. $out

Preia documentele returnate de fluxul de agregare şi le scrie în colecţia specificată. Se poate specifica baza de date de ieşire. Etapa `$out` trebuie să fie ultima etapă a fluxului de agregare. Sintaxa operatorului `$out` este următoarea:

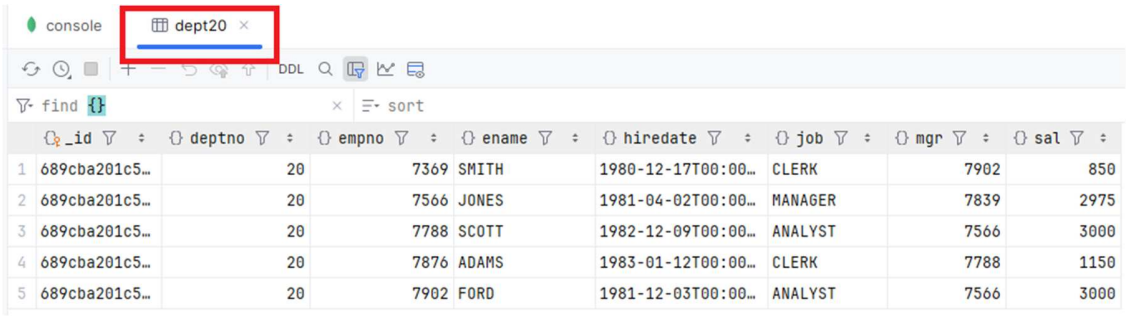
```
{ $out: "nume_colecție" }
```

Unde *nume_colecție* reprezintă numele colecţiei în care vor fi salvate rezultatele fluxului de agregare. Dacă nume_colecţie există deja, atunci `$out` înlocuieşte colecţia existentă cu noua colecţie rezultată la finalizarea agregării.

Exemplu: Se crează colecţia `dept20` în care vor fi salvate toate documentele din colecţia `emp` care au valoarea `20` pentru câmpul `deptno`. Rezultatul este ilustrat în imaginea de mai jos.

```
db.emp.aggregate([
        {$match:{deptno:20}},
        {$out:"dept20"}
        ])
```

Observaţie: Lista cu colecţiile din baza de date `emp_dept_salgrade` nu se actualizează automat. Pentru a verifica crearea colecţiei trebuie selectată opţiunea *Refresh* (CTRL+F5).


console dept20

find {} sort

	_id	deptno	empno	ename	hiredate	job	mgr	sal
1	689cba201c5…	20	7369	SMITH	1980-12-17T00:00…	CLERK	7902	850
2	689cba201c5…	20	7566	JONES	1981-04-02T00:00…	MANAGER	7839	2975
3	689cba201c5…	20	7788	SCOTT	1982-12-09T00:00…	ANALYST	7566	3000
4	689cba201c5…	20	7876	ADAMS	1983-01-12T00:00…	CLERK	7788	1150
5	689cba201c5…	20	7902	FORD	1981-12-03T00:00…	ANALYST	7566	3000

2.9. $unwind

Sintaxa operatorului `$unwind` este următoarea:

```
{ $unwind: "$nume_câmp_vector" }
```

Unde *nume_câmp_vector* reprezintă numele câmpului de tip vector din documentele de intrare în etapa de agregare.

Exemplu: Se descompune vectorul `numbers`. Cele două figuri prezintă colecţia `numbers` şi respectiv rezultatul interogării.

```
db.phone.aggregate([{$unwind:"$numbers"}])
```

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

	_id	empno	numbers
1	68a82e6b092095a82182b29c	7902	0732096123
2	68a82e6b092095a82182b29c	7902	0732096455
3	68a82e6b092095a82182b29c	7902	0732096980
4	68a83089092095a82182b29e	7844	0732021644

3. Probleme propuse

1. Să se creeze o colecţie cu numele `clerk` care să conţină documentele corespunzătoare angajaţilor cu jobul “`CLERK`”.
2. Să se afişeze media salariilor pentru fiecare departament.
3. Să se afişeze doar numele şi jobul pentru toţi angajaţii din colecţia `emp`.

4. Să se afişeze documentele din colecţia `emp` sortate crescător, în funcţie de câmpul `ename`.
5. Să se afişeze numele, salariul şi numărul departamentului pentru 5 angajaţi.
6. Să se afişeze câţi angajaţi au postul de "`SALESMAN`" şi salariul de `1250`.
7. Să se afişeze pentru fiecare angajat numele său şi datele corespunzătoare managerului său.
8. Să se afişeze pentru fiecare angajat numele său şi oraşul în care lucrează.