

Лабораторна робота №6

Тема: Практичне використання Aggregation Framework у MongoDB

З дисципліни: Базы даних та інформаційні системи

Студента групи МІТ-31: Ларіонова С.О.

Створення та наповнення бази даних:

```
use mydatabase
db.products.insertMany([
  { "name": "Laptop", "category": "Electronics", "price": 1200, "stock": 15 },
  { "name": "Sofa", "category": "Furniture", "price": 700, "stock": 10 },
  { "name": "Running Shoes", "category": "Clothing", "price": 100, "stock": 25 },
  ...
  { "name": "Drone", "category": "Electronics", "price": 600, "stock": 10 }
])
```



```
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('67e965e7b59b7fb03bb52138'),
    '1': ObjectId('67e965e7b59b7fb03bb52139'),
    '2': ObjectId('67e965e7b59b7fb03bb5213a'),
    '3': ObjectId('67e965e7b59b7fb03bb5213b'),
    '4': ObjectId('67e965e7b59b7fb03bb5213c'),
    '5': ObjectId('67e965e7b59b7fb03bb5213d'),
    '6': ObjectId('67e965e7b59b7fb03bb5213e'),
    '7': ObjectId('67e965e7b59b7fb03bb5213f'),
  }
}
```

Рисунок 6.1 – Результат виконання

```
db.customers.insertMany([ { "name": "Alice Smith", "email":
"alice.smith@example.com", "city": "Los Angeles", "registeredAt":
ISODate("2022-01-10") },
  { "name": "Bob Johnson", "email": "bob.johnson@example.com", "city":
"Chicago", "registeredAt": ISODate("2023-06-25") },
  { "name": "Charlie Brown", "email": "charlie.brown@example.com", "city":
"Houston", "registeredAt": ISODate("2020-11-30") },
  ...
  { "name": "Daniel Bennett", "email": "daniel.bennett@example.com", "city":
"Albuquerque", "registeredAt": ISODate("2023-08-14") }
])
```

```
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('67e965e7b59b7fb03bb52138'),
    '1': ObjectId('67e965e7b59b7fb03bb52139'),
    '2': ObjectId('67e965e7b59b7fb03bb5213a'),
    '3': ObjectId('67e965e7b59b7fb03bb5213b'),
    '4': ObjectId('67e965e7b59b7fb03bb5213c'),
    '5': ObjectId('67e965e7b59b7fb03bb5213d'),
    '6': ObjectId('67e965e7b59b7fb03bb5213e'),
    '7': ObjectId('67e965e7b59b7fb03bb5213f'),
    '8': ObjectId('67e965e7b59b7fb03bb52140'),
```

Рисунок 6.2 – Результат виконання

```
db.orders.insertMany([
  {
    "orderId": "ORD001",
    "customerId": ObjectId("67e9674ab59b7fb03bb52156"),
    "date": ISODate("2024-01-12"),
    "items": [
      { "product": "Laptop", "quantity": 1, "price": 1200 },
      { "product": "Mouse", "quantity": 2, "price": 50 }
    ],
    "status": "Completed"
  },
  ...
  {
    "orderId": "ORD030",
    "customerId": ObjectId("67e9674ab59b7fb03bb52156"),
    "date": ISODate("2024-01-15"),
    "items": [
      { "product": "Refrigerator", "quantity": 1, "price": 1200 },
      { "product": "Microwave", "quantity": 1, "price": 300 }
    ],
    "status": "Shipped"
  }
])
```

```

{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('67e96c1fb59b7fb03bb52174'),
    '1': ObjectId('67e96c1fb59b7fb03bb52175'),
    '2': ObjectId('67e96c1fb59b7fb03bb52176'),
    '3': ObjectId('67e96c1fb59b7fb03bb52177'),
    '4': ObjectId('67e96c1fb59b7fb03bb52178'),
    '5': ObjectId('67e96c1fb59b7fb03bb52179'),
    '6': ObjectId('67e96c1fb59b7fb03bb5217a'),
    '7': ObjectId('67e96c1fb59b7fb03bb5217b'),
    '8': ObjectId('67e96c1fb59b7fb03bb5217c'),
  }
}

```

Рисунок 6.3 – Результат виконання

Виконання запитів:

Частина 1: Базові агрегаційні операції

1. Відфільтруйте замовлення за останні 3 місяці

```
const threeMonthsAgo = new Date()
```

```
threeMonthsAgo.setMonth(threeMonthsAgo.getMonth()-3)
```

```
db.orders.aggregate([{$match: {date:{$gt: threeMonthsAgo}}}])
```

```

{
  _id: ObjectId('67e96c1fb59b7fb03bb52176'),
  orderId: 'ORD003',
  customerId: ObjectId('67e9674ab59b7fb03bb52156'),
  date: 2025-03-15T00:00:00.000Z,
  items: [
    {
      product: 'Refrigerator',
      quantity: 1,
      price: 1200
    },
    {
      product: 'Microwave',
      quantity: 1,
      price: 300
    }
  ],
  status: 'Shipped'
}
{
  _id: ObjectId('67e96c1fb59b7fb03bb52182'),
  orderId: 'ORD015',
  customerId: ObjectId('67e9674ab59b7fb03bb52158'),
  date: 2025-02-22T00:00:00.000Z,
  items: [
    {
      product: 'Oven',
      quantity: 1,
      price: 1000
    }
  ],
  status: 'Pending'
}

```

Рисунок 6.4 – Результат виконання запиту

2. Групування замовлень за місяцем

```
db.orders.aggregate({$group: {_id: {$month: "$date"}, orders: {$push : "$$ROOT"}}})
```

```
< {
  _id: 1,
  orders: [
    {
      _id: ObjectId('67e96c1fb59b7fb03bb52174'),
      orderId: 'ORD001',
      customerId: ObjectId('67e9674ab59b7fb03bb52156'),
      date: 2024-01-12T00:00:00.000Z,
      items: [
        {
          product: 'Laptop',
          quantity: 1,
          price: 1200
        },
        {
          product: 'Mouse',
          quantity: 2,
          price: 50
        }
      ]
    },
    {
      _id: ObjectId('67e96c1fb59b7fb03bb52179'),
      orderId: 'ORD006',
      customerId: ObjectId('67e9674ab59b7fb03bb5215a'),
      date: 2024-01-18T00:00:00.000Z,
      items: [
        {
          product: 'Dress',
          quantity: 2,
```

Рисунок 6.5 – Результат виконання запиту

3. Сортуювання за сумою замовлення

```
db.orders.aggregate(
[
  {
    $project: {
      orderId: 1,
      totalPrice: {
        $sum: {
          $map: {
            input: '$items',
            as: 'item',
            in: {
              $multiply: [
                '$$item.quantity',
                '$$item.price'
              ]
            }
          }
        }
      }
    }
  }
])
```

```

    }
  }
}
},
{ $sort: { totalPrice: -1 } }
]
)

```

```

< {
  _id: ObjectId('67e96c1fb59b7fb03bb52186'),
  orderId: 'ORD019',
  totalPrice: 2500
}
{
  _id: ObjectId('67e96c1fb59b7fb03bb52184'),
  orderId: 'ORD017',
  totalPrice: 2200
}
{
  _id: ObjectId('67e96c1fb59b7fb03bb5217c'),
  orderId: 'ORD009',
  totalPrice: 1740
}
{
  _id: ObjectId('67e96c1fb59b7fb03bb5217e'),
  orderId: 'ORD011',
  totalPrice: 1700
}
{
  _id: ObjectId('67e96c1fb59b7fb03bb5218f'),
  orderId: 'ORD028',
  totalPrice: 1620
}
{
  _id: ObjectId('67e96c1fb59b7fb03bb5217a'),
  orderId: 'ORD007',
  totalPrice: 1600
}

```

Рисунок 6.6 – Результат виконання запиту

Частина 2: Робота з масивами

4. Розгорніть масив items у замовленнях

```

db.orders.aggregate([
  {
    $unwind: {
      path: "$items"
    }
  }
])

```

```

{
  _id: ObjectId('67e96c1fb59b7fb03bb52174'),
  orderId: 'ORD001',
  customerId: ObjectId('67e9674ab59b7fb03bb52156'),
  date: 2024-01-12T00:00:00.000Z,
  items: {
    product: 'Laptop',
    quantity: 1,
    price: 1200
  },
  status: 'Completed'
}
{
  _id: ObjectId('67e96c1fb59b7fb03bb52174'),
  orderId: 'ORD001',
  customerId: ObjectId('67e9674ab59b7fb03bb52156'),
  date: 2024-01-12T00:00:00.000Z,
  items: {
    product: 'Mouse',
    quantity: 2,
    price: 50
  },
  status: 'Completed'
}

```

Рисунок 6.7 – Результат виконання запиту

5. Підрахуйте кількість проданих одиниць товарів

```

db.orders.aggregate(
[
  { $unwind: { path: '$items' } },
  {
    $group: {
      _id: '$items.product',
      bought: { $sum: '$items.quantity' }
    }
  }
]
)

```

```
  _id: 'Dishwasher',  
  bought: 1  
}  
{  
  _id: 'Headphones',  
  bought: 5  
}  
{  
  _id: 'Drone',  
  bought: 2  
}  
{  
  _id: 'Smartphone',  
  bought: 3  
}  
{  
  _id: 'Refrigerator',  
  bought: 3  
}  
{  
  _id: 'Sofa',  
  bought: 1  
}
```

Рисунок 6.8 – Результат виконання запиту

Частина 3: З'єднання колекцій (\$lookup)

6. Отримання інформації про клієнтів у замовленнях

```
db.orders.aggregate([  
  {$lookup:{  
    from: "customers",  
    localField: "customerId",  
    foreignField: "_id",  
    as: "customerData"  
  }}  
])
```

```

{
  _id: ObjectId('67e96c1fb59b7fb03bb52174'),
  orderId: 'ORD001',
  customerId: ObjectId('67e9674ab59b7fb03bb52156'),
  date: 2024-01-12T00:00:00.000Z,
  items: [
    {
      product: 'Laptop',
      quantity: 1,
      price: 1200
    },
    {
      product: 'Headphones',
      quantity: 2,
      price: 150
    }
  ],
  status: 'Completed',
  customerData: [
    {
      _id: ObjectId('67e9674ab59b7fb03bb52156'),
      name: 'Alice Smith',
      email: 'alice.smith@example.com',
      city: 'Los Angeles',
      registeredAt: 2022-01-10T00:00:00.000Z
    }
  ]
}

```

Рисунок 6.9 – Результат виконання запиту

7. Визначте найбільш активних клієнтів

```

db.orders.aggregate(
[
  {
    $group: {
      _id: '$customerId',
      orderAmount: { $sum: 1 }
    }
  },
  {
    $lookup: {
      from: 'customers',
      localField: '_id',
      foreignField: '_id',
      as: 'customerData'
    }
  },
  { $sort: { orderAmount: -1 } },
  { $limit: 5 }
]
);

```



```

    _id: ObjectId('67e9674ab59b7fb03bb52156'),
    orderAmount: 5,
    customerData: [
      {
        _id: ObjectId('67e9674ab59b7fb03bb52156'),
        name: 'Alice Smith',
        email: 'alice.smith@example.com',
        city: 'Los Angeles',
        registeredAt: 2022-01-10T00:00:00.000Z
      }
    ]
  }
}
{
  _id: ObjectId('67e9674ab59b7fb03bb52159'),
  orderAmount: 4,
  customerData: [
    {
      _id: ObjectId('67e9674ab59b7fb03bb52159'),
      name: 'Daisy Miller',
      email: 'daisy.miller@example.com',
      city: 'Phoenix',
      registeredAt: 2021-04-19T00:00:00.000Z
    }
  ]
}
{
  _id: ObjectId('67e9674ab59b7fb03bb5215d'),
  orderAmount: 3,
  customerData: [
    {
      _id: ObjectId('67e9674ab59b7fb03bb5215d'),

```

Рисунок 6.10 – Результат виконання запиту

Частина 4: Оптимізація запитів

8. Перевірте продуктивність запиту

Для перевірки продуктивності певного запиту можна скористатись `.explain("executionStats")`. За допомогою цього метода можна детально ознайомитись зі статистикою кожного етапу агрегації.

На прикладі попереднього запиту:

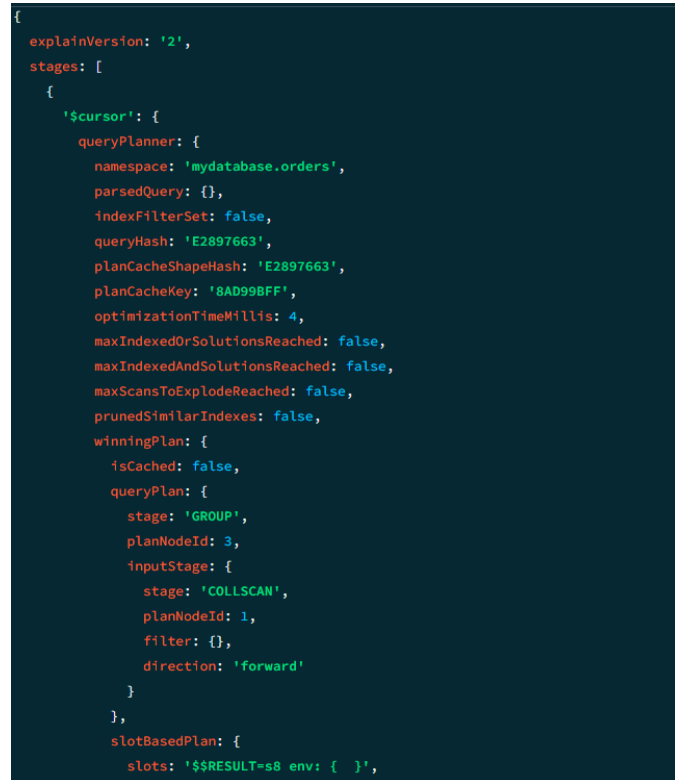
```

db.orders.aggregate(
[
  {
    $group: {
      _id: '$customerId',
      orderAmount: { $sum: 1 }
    }
  },
  {
    $lookup: {
      from: 'customers',
      localField: '_id',
      foreignField: '_id',
      as: 'customerData'
    }
  }
]
)
```

```

    }
  },
  { $sort: { orderAmount: -1 } },
  { $limit: 5 }
]
).explain("executionStats")

```



```

{
  explainVersion: '2',
  stages: [
    {
      '$cursor': {
        queryPlanner: {
          namespace: 'mydatabase.orders',
          parsedQuery: {},
          indexFilterSet: false,
          queryHash: 'E2897663',
          planCacheShapeHash: 'E2897663',
          planCacheKey: '8AD99BFF',
          optimizationTimeMillis: 4,
          maxIndexedOrSolutionsReached: false,
          maxIndexedAndSolutionsReached: false,
          maxScansToExplodeReached: false,
          prunedSimilarIndexes: false,
          winningPlan: {
            isCached: false,
            queryPlan: {
              stage: 'GROUP',
              planNodeId: 3,
              inputStage: {
                stage: 'COLLSCAN',
                planNodeId: 1,
                filter: {},
                direction: 'forward'
              }
            },
            slotBasedPlan: {
              slots: '$$RESULT=s8 env: { }',

```

Рисунок 6.11 – Результат виконання запиту

9. Оптимізуйте агрегаційний запит

Для оптимізації запиту №7 можемо використати \$project, щоб отримувалися тільки потрібні поля з колекції customers:

```

db.orders.aggregate([
  {
    $group: {
      _id: "$customerId",
      orderAmount: { $sum: 1 }
    }
  },
  {
    $lookup: {
      from: "customers",
      localField: "_id",
      foreignField: "_id",
      pipeline: [{ $project: { name: 1 } }],
      as: "customerData"
    }
  }
])

```

```

    },
    { $sort: { orderAmount: -1 } },
    { $limit: 5 }
  ]
})

```

Порівняємо швидкість виконання запиту за допомогою `.explain("executionStats")`:

Unoptimized Query (Left)	Optimized Query (Right)
<code>executionStats: {</code>	<code>executionStats: {</code>
<code> executionSuccess: true,</code>	<code> executionSuccess: true,</code>
<code> nReturned: 14,</code>	<code> nReturned: 14,</code>
<code> executionTimeMillis: 6,</code>	<code> executionTimeMillis: 5,</code>
<code> totalKeysExamined: 0,</code>	<code> totalKeysExamined: 0,</code>
<code> totalDocsExamined: 30,</code>	<code> totalDocsExamined: 30,</code>
<code> executionStages: {</code>	<code> executionStages: {</code>
<code> stage: 'project',</code>	<code> stage: 'project',</code>

Рисунок 6.12 – Фрагмент статистики виконання неоптимізованого запиту (зліва) та оптимізованого (справа)

На порівнянні статистики (рис. 6.12) можна побачити приріст швидкості приблизно на 16.66%.

Додаткові завдання

10. Визначте категорії товарів із найбільшою кількістю продажів. Використайте `$group` для підрахунку загальної кількості проданих товарів за категоріями. Відсортуйте результат за спаданням.

```

db.orders.aggregate(
[
  { $project: { items: 1, _id: 0 } },
  { $unwind: { path: '$items' } },
  {
    $group: {
      _id: '$items.product',
      amount: { $sum: '$items.quantity' }
    }
  },
  {
    $lookup: {
      from: 'products',
      localField: '_id',
      foreignField: 'name',
      pipeline: [
        { $project: { category: 1, _id: 0 } }
      ]
    }
  }
]
)

```

```

    ],
    as: 'category'
  }
},
{ $unwind: { path: '$category' } },
{
  $group: {
    _id: '$category.category',
    amount: { $sum: '$amount' }
  }
},
{ $sort: { amount: -1 } }
]
)

```

```

< {
  _id: 'Clothing',
  amount: 28
}
{
  _id: 'Electronics',
  amount: 28
}
{
  _id: 'Furniture',
  amount: 19
}
{
  _id: 'Appliances',
  amount: 13
}

```

Рисунок 6.13 – Результат виконання запиту

11. Розрахуйте середню ціну товарів у кожній категорії. Використайте **\$group** для підрахунку середньої ціни товарів у кожній категорії.

```

db.products.aggregate(
[
  {
    $group: {
      _id: '$category',
      averagePrice: { $avg: '$price' }
    }
  }
]
)

```

```
< {
  _id: 'Clothing',
  averagePrice: 74.375
}
{
  _id: 'Electronics',
  averagePrice: 708.75
}
{
  _id: 'Appliances',
  averagePrice: 636.6666666666666
}
{
  _id: 'Furniture',
  averagePrice: 506.25
}
```

Рисунок 6.14 – Результат виконання запиту

12. Знайдіть користувачів, які зробили більше одного замовлення. Використайте **\$group** і **\$match**, щоб знайти клієнтів, які мали більше одного замовлення.

```
db.orders.aggregate(
[
  {
    $group: {
      _id: '$customerId',
      orderAmount: { $sum: 1 }
    }
  },
  { $match: { orderAmount: { $gt: 0 } } },
  {
    $lookup: {
      from: 'customers',
      localField: '_id',
      foreignField: '_id',
      as: 'customerData'
    }
  }
]
)
```

```

< [
  {
    _id: ObjectId('67e9674ab59b7fb03bb5215e'),
    orderAmount: 2,
    customerData: [
      {
        _id: ObjectId('67e9674ab59b7fb03bb5215e'),
        name: 'Ian Taylor',
        email: 'ian.taylor@example.com',
        city: 'San Jose',
        registeredAt: 2022-11-07T00:00:00.000Z
      }
    ]
  },
  {
    _id: ObjectId('67e9674ab59b7fb03bb52160'),
    orderAmount: 2,
    customerData: [
      {
        _id: ObjectId('67e9674ab59b7fb03bb52160'),
        name: 'Kevin Walker',
        email: 'kevin.walker@example.com',
        city: 'Jacksonville',
        registeredAt: 2021-07-09T00:00:00.000Z
      }
    ]
  },
  {
    _id: ObjectId('67e9674ab59b7fb03bb5215c'),
    orderAmount: 1,
    customerData: [

```

Рисунок 6.15 – Результат виконання запиту

Висновок: під час роботи було вивчено базові принципи MongoDB та засобів агрегації в MongoDB. Серед викликів були труднощі із розумінням логіки певних етапів конвеєру агрегації, але використання MongoDB Compass допомогло з формуванням запитів завдяки поетапній візуалізації агрегації. До найкращих підходів оптимізації відносять індексацію та коригування логіки запиту для уникання роботи з зайвими даними.