

**Київський національний університет
Імені Тараса Шевченка.**

Кафедра: Мережєвих та інтернет-технологій.

Лабораторна робота №6

з дисципліни: Бази даних та інформаційні системи

На тему: «Практичне використання Aggregation Framework у MongoDB»

Студента 3 курсу:
Групи МІТ-31
Мулико Володимира

Київ - 2025р.

Хід роботи

1. Встановлюємо MongoDB Shell

```
C:\Users\Vovam>mongosh --version
2.4.2
```

2. Запускаємо MongoShell

```
C:\Users\Vovam>mongosh|
```

```
Current Mongosh Log ID: 67dca4d5dbed8b2fe3b71235
Connecting to:          mongodb://127.0.0.1:27017/
2.4.2
Using MongoDB:          8.0.5
Using Mongosh:          2.4.2

For mongosh info see: https://www.mongodb.com/docs

-----
The server generated these startup warnings when booting
2025-03-20T01:04:32.334+02:00: Access control is not enabled
configuration is unrestricted
-----

test>
```

3. Починаємо створювати колекції та писати запити

3.1. Створення нової колекції у mongosh:

```
test> use myDatabase
switched to db myDatabase
myDatabase>
```

3.2. Створення колекції Orders.

```
myDatabase> db.createCollection("orders")
... db.orders.insertMany([
...   {
...     "orderId": "ORD001",
...     "customerId": ObjectId(),
...     "date": ISODate("2024-01-12"),
...     "items": [
...       { "product": "Laptop", "quantity": 1, "price": 1200 },
...       { "product": "Mouse", "quantity": 2, "price": 50 }
...     ],
...     "status": "Completed"
...   }
... ])
...
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67dca809dbed8b2fe3b71237') }
}
```

3.3. Створення колекції Customers

```
myDatabase> db.createCollection("customers")
... db.customers.insertMany([
...   {
...     "_id": ObjectId(),
...     "name": "John Doe",
...     "email": "john.doe@example.com",
...     "city": "New York",
...     "registeredAt": ISODate("2021-03-15")
...   }
... ])
...
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId('67dca874dbed8b2fe3b71238') }
}
```

3.4. Створення колекції Product

```
myDatabase> db.createCollection("products")
... db.products.insertMany([
...   {
...     "_id": ObjectId(),
...     "name": "Laptop",
...     "category": "Electronics",
...     "price": 1200,
...     "stock": 15
...   }
... ])
...
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId('67dca946dbed8b2fe3b71239') }
}
```

4. Базові агрегаційні операції

4.1. Відфільтруйте замовлення за останні 3 місяці

```
myDatabase> db.orders.aggregate([
...   {
...     $match: {
...       "date": { $gte: new Date(new Date().setMonth(new Date().getMonth() - 3)) }
...     }
...   }
... ])
...
```

4.2. Групування замовлень за місяцем

```
myDatabase> db.orders.aggregate([
...   {
...     $group: {
...       _id: { $month: "$date" },
...       totalOrders: { $sum: 1 }
...     }
...   }
... ])
...
[ { _id: 1, totalOrders: 2 } ]
```

4.3. Сортування за сумою замовлення

```
myDatabase> db.orders.aggregate([
...   {
...     $addFields: {
...       totalAmount: { $sum: { $map: { input: "$items", as: "item", in: { $multiply: ["$$item.quantity", "$$item.price"] } } } }
...     }
...   },
...   { $sort: { totalAmount: -1 } }
... ])
[
...
...
...]
```

Вивід програми:

```
{
  _id: ObjectId('67db51d97d368e49edb71237'),
  orderId: 'ORD001',
  customerId: ObjectId('67db51d97d368e49edb71236'),
  date: ISODate('2024-01-12T00:00:00.000Z'),
  items: [
    { product: 'Laptop', quantity: 1, price: 1200 },
    { product: 'Mouse', quantity: 2, price: 50 }
  ],
  status: 'Completed',
  totalAmount: 1300
},
{
  _id: ObjectId('67dca809dbed8b2fe3b71237'),
  orderId: 'ORD001',
  customerId: ObjectId('67dca809dbed8b2fe3b71236'),
  date: ISODate('2024-01-12T00:00:00.000Z'),
  items: [
    { product: 'Laptop', quantity: 1, price: 1200 },
    { product: 'Mouse', quantity: 2, price: 50 }
  ],
  status: 'Completed',
  totalAmount: 1300
}
```

5. Робота з масивами

5.1. Розгорніть масив items у замовленнях

```
myDatabase> db.orders.aggregate([
...   { $unwind: "$items" }
... ])
...
[
```

Вивід програми:

```
[
{
  _id: ObjectId('67db51d97d368e49edb71237'),
  orderId: 'ORD001',
  customerId: ObjectId('67db51d97d368e49edb71236'),
  date: ISODate('2024-01-12T00:00:00.000Z'),
  items: { product: 'Laptop', quantity: 1, price: 1200 },
  status: 'Completed'
},
{
  _id: ObjectId('67db51d97d368e49edb71237'),
  orderId: 'ORD001',
  customerId: ObjectId('67db51d97d368e49edb71236'),
  date: ISODate('2024-01-12T00:00:00.000Z'),
  items: { product: 'Mouse', quantity: 2, price: 50 },
  status: 'Completed'
}
]
```

5.2. Підрахуйте кількість проданих одиниць товарів

```
myDatabase> db.orders.aggregate([
...   { $unwind: "$items" },
...   {
...     $group: {
...       _id: "$items.product",
...       totalSold: { $sum: "$items.quantity" }
...     }
...   }
... ])
...
[ { _id: 'Mouse', totalSold: 4 }, { _id: 'Laptop', totalSold: 2 } ]
```

6. З'єднання колекцій (\$lookup)

6.1. Отримання інформації про клієнтів у замовленнях

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

Вивід програми:

```
[
  {
    _id: ObjectId('67db51d97d368e49edb71237'),
    orderId: 'ORD001',
    customerId: ObjectId('67db51d97d368e49edb71236'),
    date: ISODate('2024-01-12T00:00:00.000Z'),
    items: [
      { product: 'Laptop', quantity: 1, price: 1200 },
      { product: 'Mouse', quantity: 2, price: 50 }
    ],
    status: 'Completed',
    customerInfo: []
  },
  {
    _id: ObjectId('67dca809dbed8b2fe3b71237'),
    orderId: 'ORD001',
    customerId: ObjectId('67dca809dbed8b2fe3b71236'),
    date: ISODate('2024-01-12T00:00:00.000Z'),
    items: [
      { product: 'Laptop', quantity: 1, price: 1200 },
      { product: 'Mouse', quantity: 2, price: 50 }
    ],
    status: 'Completed',
    customerInfo: []
  }
]
```

6.2. Визначте найбільш активних клієнтів

```
myDatabase> db.orders.aggregate([
...   {
...     $group: {
...       _id: "$customerId",
...       totalOrders: { $sum: 1 }
...     }
...   },
...   { $sort: { totalOrders: -1 } }
... ])
...
[
  { _id: ObjectId('67dca809dbed8b2fe3b71236'), totalOrders: 1 },
  { _id: ObjectId('67db51d97d368e49edb71236'), totalOrders: 1 }
]
```

7. Оптимізація запитів

7.1. Перевірте продуктивність запиту

```
myDatabase> db.orders.explain("executionStats").aggregate([
...   { $match: { status: "Completed" } }
... ])
...
{
```

Вивід програми:

```
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'myDatabase.orders',
    parsedQuery: { status: { '$eq': 'Completed' } },
    indexFilterSet: false,
    queryHash: '5D6543D9',
    planCacheShapeHash: '5D6543D9',
    planCacheKey: '405CB45D',
    optimizationTimeMillis: 1,
    optimizedPipeline: true,
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    prunedSimilarIndexes: false,
    winningPlan: {
      isCached: false,
      stage: 'COLLSCAN',
      filter: { status: { '$eq': 'Completed' } },
      direction: 'forward'
    },
    rejectedPlans: []
  },
}
```

```
  executionStats: {
    executionSuccess: true,
    nReturned: 2,
    executionTimeMillis: 3,
    totalKeysExamined: 0,
    totalDocsExamined: 2,
    executionStages: {
      isCached: false,
      stage: 'COLLSCAN',
      filter: { status: { '$eq': 'Completed' } },
      nReturned: 2,
      executionTimeMillisEstimate: 0,
      works: 3,
      advanced: 2,
      needTime: 0,
      needYield: 0,
      saveState: 0,
      restoreState: 0,
      isEOF: 1,
      direction: 'forward',
      docsExamined: 2
    }
  }
}
```

Продовження виводу програми від пункту 7.1

```
}
queryShapeHash: '5C3C276017285326CB71C5BCB2DDEA4325F7D9CC2D839AE6DAA1125667A729A9',
command: {
  aggregate: 'orders',
  pipeline: [ { '$match': { status: 'Completed' } } ],
  cursor: {},
  '$db': 'myDatabase'
},
serverInfo: {
  host: 'ASUS_ROG',
  port: 27017,
  version: '8.0.5',
  gitVersion: 'cb9e2e5e552ee39de1e39d7859336456d0c9820'
},
serverParameters: {
  internalQueryFacetBufferSizeBytes: 104857600,
  internalQueryFacetMaxOutputDocSizeBytes: 104857600,
  internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
  internalDocumentSourceGroupMaxMemoryBytes: 104857600,
  internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
  internalQueryProhibitBlockingMergeOnMongoS: 0,
  internalQueryMaxAddToSetBytes: 104857600,
  internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600,
  internalQueryFrameworkControl: 'trySbeRestricted',
  internalQueryPlannerIgnoreIndexWithCollationForRegex: 1
},
ok: 1
}
```

7.2. Оптимізуйте агрегаційний запит

```
myDatabase> db.orders.createIndex({ "date": 1 })
date_1
```

8. Додаткові завдання

8.1. Визначення категорій товарів із найбільшою кількістю продажів

```
myDatabase> db.orders.aggregate([
...   { $unwind: "$items" },
...   {
...     $lookup: {
...       from: "products",
...       localField: "items.product",
...       foreignField: "name",
...       as: "productInfo"
...     }
...   },
...   { $unwind: "$productInfo" },
...   {
...     $group: {
...       _id: "$productInfo.category",
...       totalSold: { $sum: "$items.quantity" }
...     }
...   },
...   { $sort: { totalSold: -1 } }
... ])
```

Вивід програми:

```
[ { _id: 'Electronics', totalSold: 4 } ]
```

8.2. Розрахунок середньої ціни товарів у кожній категорії

```
myDatabase> db.products.aggregate([
...   {
...     $group: {
...       _id: "$category",
...       avgPrice: { $avg: "$price" }
...     }
...   }
... ])
...
[ { _id: 'Electronics', avgPrice: 1200 } ]
```

8.3. Пошук клієнтів, які зробили більше одного замовлення

```
myDatabase> db.orders.aggregate([
...   {
...     $group: {
...       _id: "$customerId",
...       orderCount: { $sum: 1 }
...     }
...   },
...   { $match: { orderCount: { $gt: 1 } } }
... ])
...
```

Висновок: В цій лабораторній роботі ми почали знайомство з новим інтерфейсом створення Баз Даних – MongoDB. Розібралися з установкою та створенням нових таблиць, записали код оптимізації та написання запитів, зрозуміли що звичайний SQL набагато зручніший.