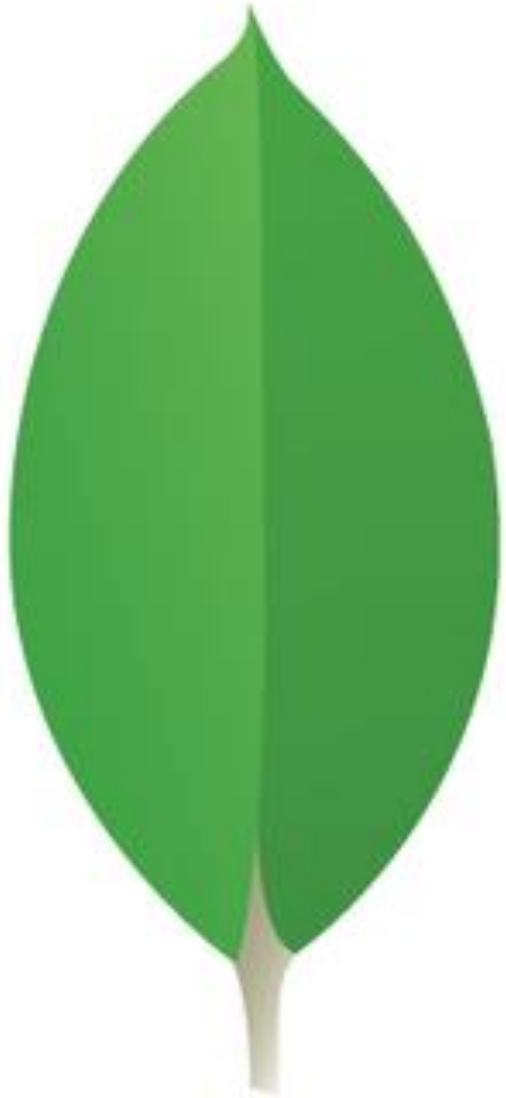


MongoDB 2

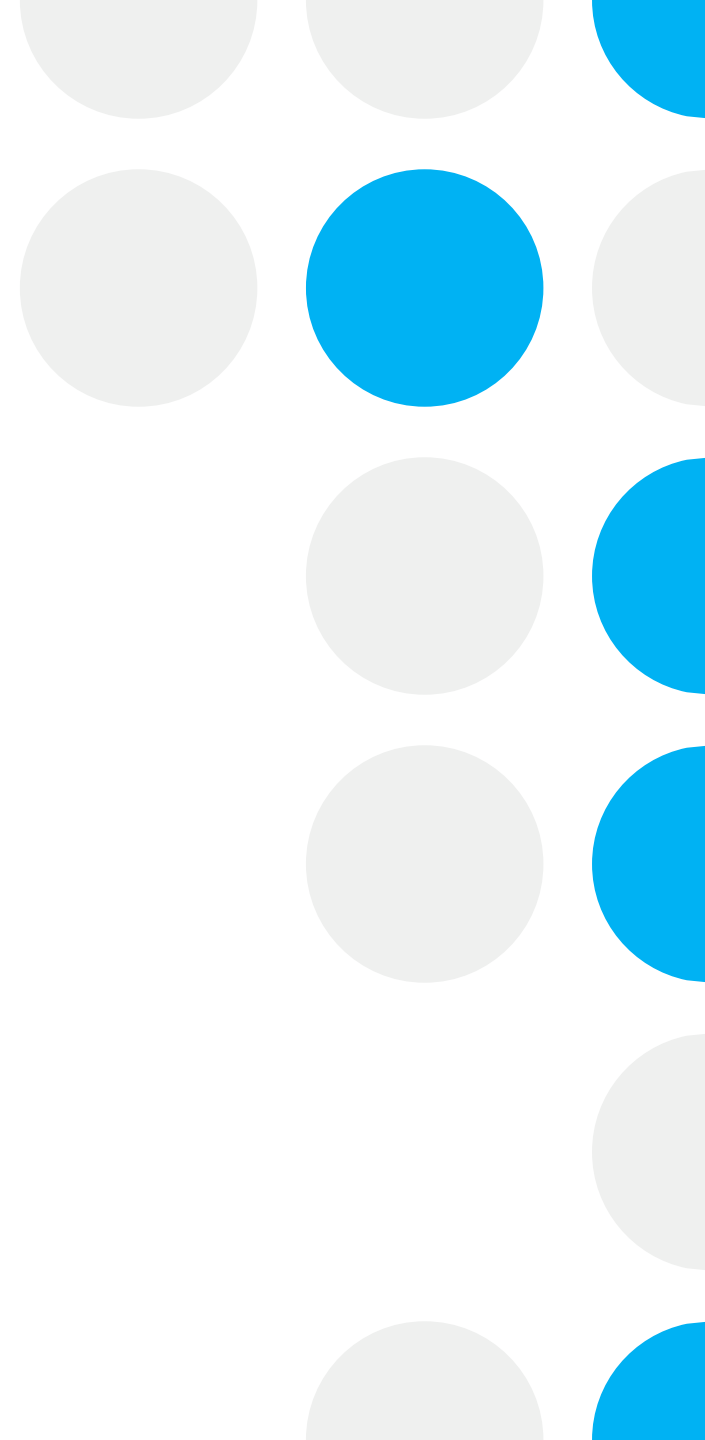


mongoDB

Advanced CRUD & Aggregation

Content

- Advanced CRUD
- Aggregation Pipelines



Advanced CRUD



Advanced CRUD – Upsert

- Most update operations allow the flag `upsert:true`
- If exists, update
- If does not exist, insert

```
use session2

db.example7.updateOne({ name: "John Doe" },
  $set: {
    age: 33,
    address: {
      country: "France",
      city: "Paris"
    }
  }, { upsert: true })

db.example7.find()

db.example7.updateOne({ name: "John Doe" },
  $set: {
    age: 34,
    address: {
      country: "France",
      city: "Marseille"
    }
  }, { upsert: true })

db.example7.find()
```

Advanced CRUD – findOneAndUpdate

- updateOne() + findOne() causes race condition
- Multiple simultaneous updates

```
use session2

db.example8.updateOne({ name: "John Doe" }, {
  $set: {
    viewed: 0
  }
}, { upsert: true })

// User 1
db.example8.updateOne({ name: "John Doe" }, {
  $inc: {
    viewed: 1
  }
})

// User 2
db.example8.updateOne({ name: "John Doe" }, {
  $inc: {
    viewed: 1
  }
})

// User 1
db.example8.findOne({ name: "John Doe" })

// User 2
db.example8.findOne({ name: "John Doe" })
```

Advanced CRUD – findOneAndUpdate

- findOneAndUpdate solves this problem
- { returnNewDocument : true }

```
use session2

db.example9.updateOne({ name: "John Doe" }, {
  $set: {
    viewed: 0
  }
}, { upsert: true })

// User 1
db.example9.findOneAndUpdate({ name: "John Doe" }, {
  $inc: {
    viewed: 1
  }
})

// User 2
db.example9.findOneAndUpdate({ name: "John Doe" }, {
  $inc: {
    viewed: 1
  }
})

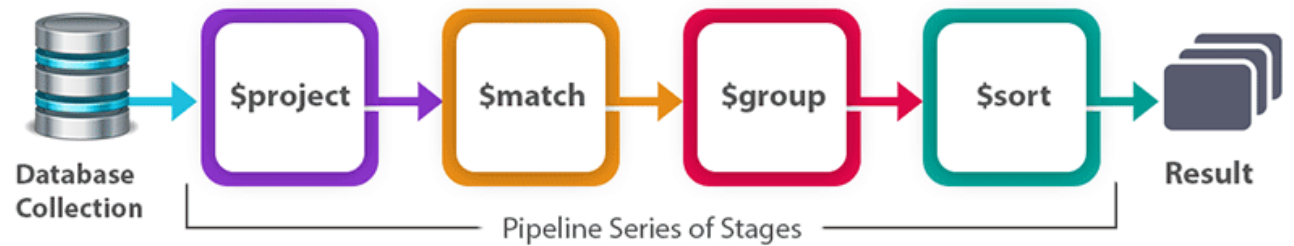
// Get updated document
db.example9.findOneAndUpdate({ name: "John Doe" }, {
  $inc: {
    viewed: 1
  }
}, { returnNewDocument: true })
```

Aggregation Pipeline



Aggregation

- Retrieval operations can filter:
 - What documents
 - What fields
 - What order
- Aggregation allows us to:
 - Calculate new fields
 - Summarize and group values
 - Reshape documents



Aggregation – Basic Stages

\$match

\$project

\$sort

\$limit

\$skip

\$count

Aggregation – Arithmetic Expressions

- \$subtract
- \$add
- \$multiply
- \$divide

```
use sample_airbnb

db.listingsAndReviews.aggregate([
  {
    $project: {
      name: 1,
      doublePrice: {
        $multiply: [
          "$price", 2
        ]
      },
      pricePlus10: {
        $add: [
          "$price", 10
        ]
      },
      price: 1,
      room_type: 1
    }
  }
])
```

Aggregation – String Expressions

- \$concat
- \$ltrim
- \$indexOfCP
- \$strLenCP
- \$split

```
use sample_airbnb

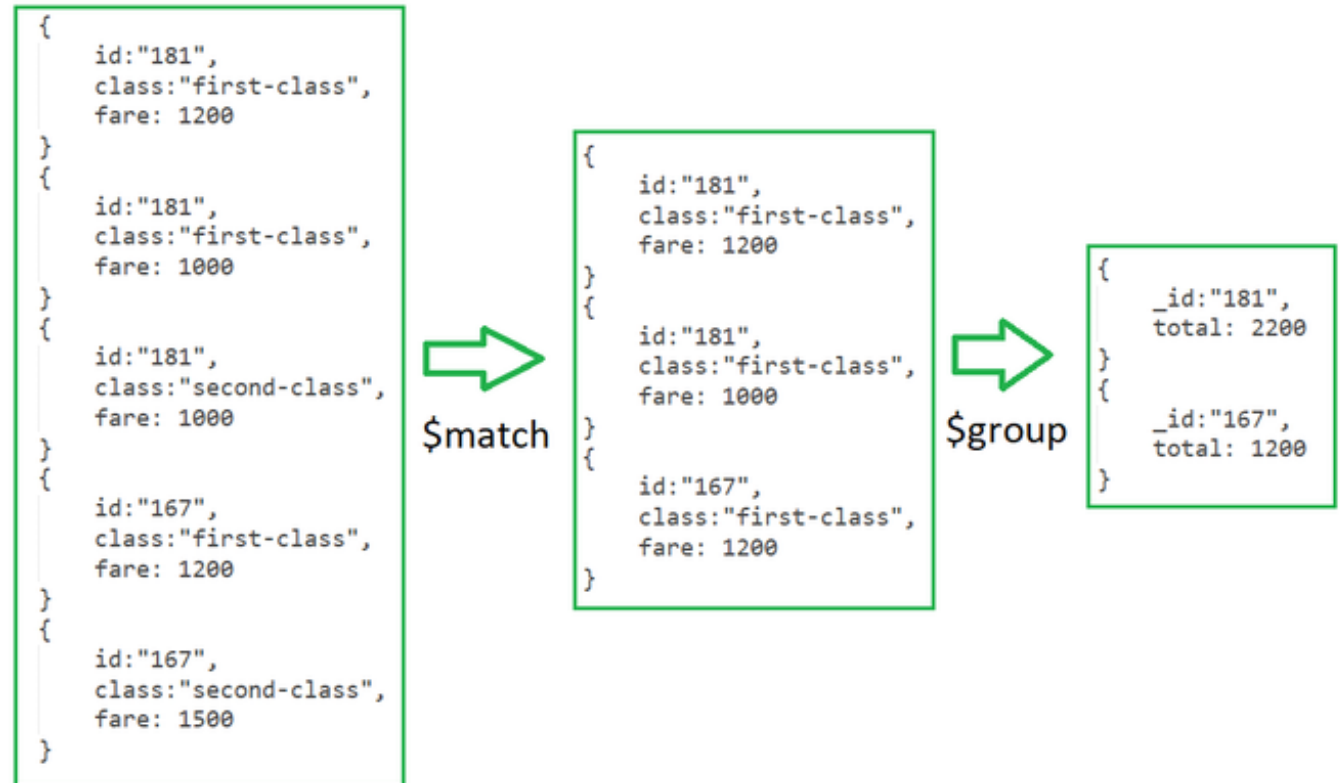
db.listingsAndReviews.aggregate([
  {
    $project: {
      name: 1,
      price: 1,
      room_type: 1,
      nameAndRoomType: {
        $concat: [
          "$name", " - ", "$room_type"
        ]
      },
      nameLength: {
        $strLenCP: "$name"
      },
      roomTypeWords: {
        $split: ["$room_type", " "]
      }
    }
  }
])
```

Aggregation - Group

- We can group our document using \$group stage
- _id is what MongoDB uses as a unique field

```
db.train.aggregate( [  
  {$match:{class:"first-class"}},  
  {$group:{_id:"id",total:{$sum:"$fare"}}} ] )
```

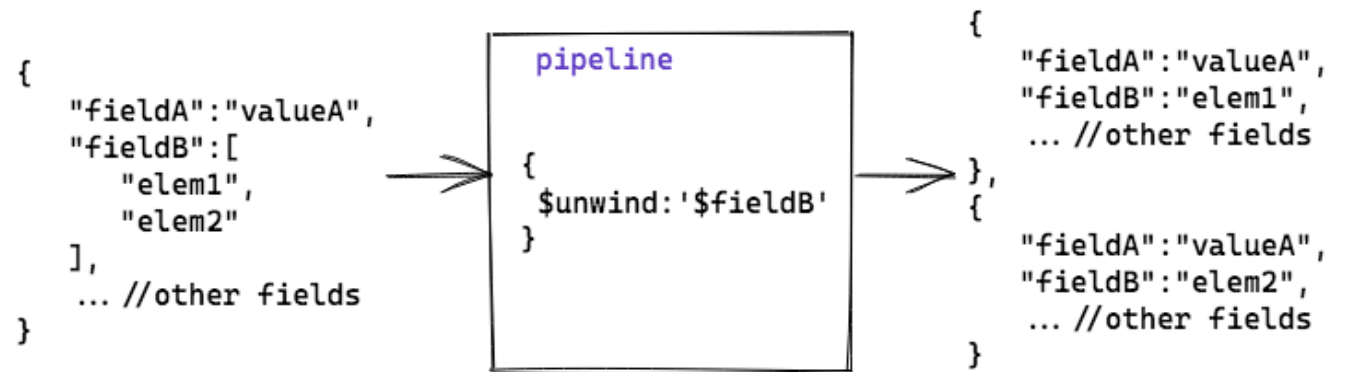
} pipeline stages



Aggregation - Unwind

- \$unwind stage
- Opposite of \$group
- Converts one document to many
- One document per value in array

MongoDB Aggregate \$unwind



Aggregation - Join

- \$lookup stage
- Like Left Outer Join
- Embed results as array in parent document

```
use session2
db.example12.insertMany([
  {
    _id: 0,
    type: 1,
  }, {
    _id: 1,
    type: 1,
  }, {
    _id: 2,
    type: 2,
  }, {
    _id: 3,
    type: 3,
  }
])

db.example13.insertMany([
  {
    _id: 1,
    name: 'MSFT'
  }, {
    _id: 2,
    name: 'ORCL'
  }, {
    _id: 3,
    name: 'MDB'
  }
])

db.example12.aggregate([
  {
    $lookup: {
      from: 'example13',
      localField: 'type',
      foreignField: '_id',
      as: 'type_details'
    }
  }
])
```

Aggregation – More Grouping

\$bucket

\$bucketAuto

\$sortByCount

Aggregation – More Stages

\$set

\$out

\$merge

\$replaceRoot

\$sample

Questions?

