

Element Operators

1. \$exists

ব্যাখ্যা:

নির্দিষ্ট কোনো ফিল্ড ডকুমেন্টে আছে কিনা তা যাচাই করে।

উদাহরণ:

address ফিল্ড যেসব ডকুমেন্টে আছে, সেগুলো নির্বাচন করতে:

```
javascript
```

```
db.collection.find({ "address": { "$exists": true } })
```

2. \$type

ব্যাখ্যা:

নির্দিষ্ট ফিল্ডের ডেটাতাইপ চেক করে (যেমন: string, int, array ইত্যাদি)।

উদাহরণ:

age ফিল্ড যেসব ডকুমেন্টে সংখ্যা (NumberInt) টাইপের, তা খুঁজে বের করতে:

```
javascript
```

```
db.collection.find({ "age": { "$type": "int" } })
```

Evaluation Operators

3. \$expr

ব্যাখ্যা:

কোয়েরিতে এগ্রিগেশন এক্সপ্রেশন ব্যবহার করার সুযোগ দেয়।

উদাহরণ:

যেসব ইউজারের বয়স ৩০ বছরের বেশি:

```
javascript
```

```
db.collection.find({ "$expr": { "$gt": ["$age", 30] } })
```

4. \$jsonSchema

ব্যাখ্যা:

ডকুমেন্টের স্ট্রাকচার বা ডেটা ভ্যালিডেশন করার জন্য JSON Schema ব্যবহার করা হয়।

উদাহরণ:

`email` ফিল্ড অবশ্যই থাকতে হবে এবং তা স্ট্রিং হতে হবে এমন ভ্যালিডেশন:

```
javascript
```

```
db.collection.find({
  "$jsonSchema": {
    "required": ["email"],
    "properties": {
      "email": { "bsonType": "string" }
    }
  }
})
```

5. \$mod

ব্যাখ্যা:

কোনো ফিল্ডের মানকে একটি সংখ্যা দিয়ে ভাগ করে নির্দিষ্ট রিমাইন্ডার থাকলে সিলেক্ট করে।

উদাহরণ:

age যেসব ইউজারের বয়স ৫ দিয়ে ভাগ করলে বাকি থাকে ২:

```
javascript
```

```
db.collection.find({ "age": { "$mod": [5, 2] } })
```

6. \$regex

ব্যাখ্যা:

রেগুলার এক্সপ্রেশন ব্যবহার করে ফিল্ডের ভ্যালু ম্যাচ করে।

উদাহরণ:

যেসব ইউজারের **email** ফিল্ডে 'washingtonpost' আছে:

```
javascript
```

```
db.collection.find({ "email": { "$regex": "washingtonpost" } })
```

7. \$text

ব্যাখ্যা:

ফুল টেক্সট সার্চ করার জন্য ব্যবহৃত হয়। (MongoDB Self-Managed সার্ভারে ব্যবহৃত হয়)

উদাহরণ:

occupation বা অন্যান্য টেক্সট ফিল্ডে 'Advisor' শব্দ খুঁজতে:

javascript

```
db.collection.find({ "$text": { "$search": "Advisor" } })
```

Note:

Atlas-এ থাকলে **Atlas Search** ব্যবহার করা উচিত।

8. \$where

ব্যাখ্যা:

কাস্টম JavaScript এক্সপ্রেশন দিয়ে ডকুমেন্ট সিলেক্ট করে।

উদাহরণ:

যেসব ইউজারের বয়স ৫০ বছরের বেশি এবং তারা France-এ থাকে:

javascript

```
db.collection.find({
  "$where": function() {
    return this.age > 50 && this.address.country === "France";
  }
})
```

Array Query Operators

9. \$all

ব্যাখ্যা:

একটি অ্যারে ফিল্ডে নির্দিষ্ট সব এলিমেন্ট একসাথে আছে কিনা তা যাচাই করে।

উদাহরণ:

যেসব ইউজারের **friends** লিস্টে "Mizanur Rahman" এবং "Fahim Ahammed Firoz" দুজনেই আছে:

javascript

```
db.collection.find({ "friends": { "$all": ["Mizanur Rahman", "Fahim Ahammed Firoz"] } })
```

10. \$elemMatch

ব্যাখ্যা:

অ্যারের মধ্যে কোনো একটি এলিমেন্ট যেটি একাধিক শর্ত পূরণ করে, তা নির্বাচন করতে।

উদাহরণ:

education অ্যারের মধ্যে এমন ডকুমেন্ট যাদের ডিগ্রি "Master of Education" এবং মেজর "Biology":

javascript

```
db.collection.find({
  "education": {
    "$elemMatch": {
      "degree": "Master of Education",
      "major": "Biology"
    }
  }
})
```

11. \$size

ব্যাখ্যা:

অ্যারে ফিল্ডের সাইজ (এলিমেন্ট সংখ্যা) চেক করে।

উদাহরণ:

যাদের **languages** অ্যারেতে মোট ৩টি ভাষা আছে:

javascript

```
db.collection.find({ "languages": { "$size": 3 } })
```

✓ সংক্ষেপে টিপস:

- `$exists` ও `$type` দিয়ে ফিল্ড যাচাই করা হয়।
 - `$expr`, `$where`, ও `$jsonSchema` ব্যবহার হয় জটিল লজিকের জন্য।
 - `$mod`, `$regex`, `$text` দিয়ে কন্ডিশনাল সার্চ করা হয়।
 - `$all`, `$elemMatch`, `$size` দিয়ে অ্যারে আইটেম নিয়ে কাজ করা হয়।
-

আপনার কাজ বা প্রজেক্টের জন্য চাইলে আমি এগুলোর রেডিমেড কোড টেমপ্লেটও সাজিয়ে দিতে পারি।
আপনি কি চান, প্রতিটি অপারেটরের আরো বাস্তব ভিত্তিক উদাহরণ (real-life example) দেই? 🚀