

# MongoDB

## 'OR', 'where', 'CRUD',

### 1. OR Queries

The `$or` operator is used to perform a logical OR operation on an array of two or more expressions.

**Example:** Find documents where the name is either "Alice" or "Bob".

```
db.collection.find({
  $or: [
    { name: "Alice" },
    { name: "Bob" }
  ]
})
```

### 2. WHERE Queries

The `$where` operator allows you to specify JavaScript expressions to query documents. It should be used sparingly as it can be slower and less efficient.

**Example:** Find documents where the age is greater than 25.

```
db.collection.find({
  $where: "this.age > 25"
})
```

# MongoDB

## 3. CRUD Operations

- a. Create: To add new documents to a collection, you use the `insertOne` or `insertMany` methods.

Insert One Document:

```
db.collection.insertOne({
  name: "Alice",
  age: 25,
  city: "New York"
})
```

Output:

```
{
  "acknowledged" : true,
  "insertedId" : ObjectId("60d1f68ee5c15c1b5f13d8f9")
}
```

Insert Many Documents:

```
db.collection.insertMany([
  { name: "Bob", age: 30, city: "San Francisco" },
  { name: "Charlie", age: 35, city: "Los Angeles" }
])
```

Output:

```
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("60d1f6c9e5c15c1b5f13d8fa"),
    ObjectId("60d1f6c9e5c15c1b5f13d8fb")
  ]
}
```

# MongoDB

- b. READ:** To retrieve documents, you use the `find` method. You can also specify conditions to filter the results.

Find All Documents:

```
db.collection.find()
```

Find Documents with a Condition

```
db.collection.find({ age: { $gt: 25 } })
```

- c. Update:** To modify existing documents, you use the `updateOne` or `updateMany` methods.

**Update One Document:** Update the age of the document where the name is "Alice".

```
db.collection.updateOne(  
  { name: "Alice" },  
  { $set: { age: 26 } }  
)
```

Output:

```
{  
  "acknowledged" : true,  
  "matchedCount" : 1,  
  "modifiedCount" : 1  
}
```

# MongoDB

**Update Many Documents:** Update the city of all documents where the age is greater than 30

```
db.collection.updateMany(  
  { age: { $gt: 30 } },  
  { $set: { city: "Chicago" } }  
)
```

Output:

```
{  
  "acknowledged" : true,  
  "matchedCount" : 2,  
  "modifiedCount" : 2  
}
```

**d. Delete:** To remove documents, you use the `deleteOne` or `deleteMany` methods.

**Delete One Document:** Delete the document where the name is "Alice".

```
db.collection.deleteOne({ name: "Alice" })
```

Output:

```
{  
  "acknowledged" : true,  
  "deletedCount" : 1  
}
```

**Delete Many Documents:** Delete all documents where the age is less than 30.

```
db.collection.deleteMany({ age: { $lt: 30 } })
```

Output:

```
{  
  "acknowledged" : true,  
  "deletedCount" : 2  
}
```