

# How do I rename fields in MongoDB?

Authored by  
**stats writer**

June 30, 2024

## RECOMMENDED CITATION

stats writer (2024). *How do I rename fields in MongoDB?*. PSYCHOLOGICAL SCALES.  
Retrieved from <https://scales.arabpsychology.com/?p=162280>

MongoDB is a popular NoSQL database management system that allows for flexible and efficient storage of data. One of the key features of MongoDB is the ability to rename fields within a collection. This functionality allows for better organization and management of data as well as easier querying and manipulation. To rename fields in MongoDB, one must use the update command in conjunction with the \$rename operator. This allows for the targeted renaming of specific fields within a collection. Furthermore, MongoDB also offers the option to use the \$unset operator to remove unwanted fields entirely. Overall, the ability to rename fields in MongoDB provides users with increased control and flexibility in managing their data.

## Rename Fields in MongoDB (3 Examples)

You can use the following methods to rename fields in MongoDB:

### Method 1: Rename One Field

```
db.collection.updateMany({},  
{ $rename: { "oldField": "newField" } }, false, true)
```

### Method 2: Rename Multiple Fields

```
db.collection.updateMany({}, { $rename: { "old1": "new1",  
"old2": "new2" } }, false, true)
```

### Method 3: Rename Subfield

```
db.collection.updateMany({},  
{ $rename: { "field.oldSub": "field.newSub" } }, false, true)
```

Note that the `false`, `true` in the `$rename` function stands for `{upsert:false, multi:true}`.

You need the `multi:true` to update the field name in all of your documents.

The following examples show how to use each method with a collection `teams` with the following documents:

```
db.teams.insertOne({team: "Mavs", class:
{conf:"Western", div:"A"}, points: 31})
db.teams.insertOne({team: "Spurs", class:
{conf:"Western", div:"A"}, points: 22})
db.teams.insertOne({team: "Jazz", class:
{conf:"Western", div:"B"}, points: 19})
db.teams.insertOne({team: "Celtics", class:
{conf:"Eastern", div:"C"}, points: 26})
db.teams.insertOne({team: "Cavs", class:
{conf:"Eastern", div:"D"}, points: 33})
db.teams.insertOne({team: "Nets", class:
{conf:"Eastern", div:"D"}, points: 38})
```

Example 1: Rename One Field

We can use the following code to rename the `team` field

to `new_team`:

```
db.teams.updateMany({},  
{ $rename: { "team": "new_team" } }, false, true)
```

Here are what the documents look like now:

```
{ _id: ObjectId("62017ce6fd435937399d6b58"),  
  class: { conf: 'Western', div: 'A' },  
  points: 31,  
  new_team: 'Mavs' }  
{ _id: ObjectId("62017ce6fd435937399d6b59"),  
  class: { conf: 'Western', div: 'A' },  
  points: 22,  
  new_team: 'Spurs' }  
{ _id: ObjectId("62017ce6fd435937399d6b5a"),  
  class: { conf: 'Western', div: 'B' },  
  points: 19,  
  new_team: 'Jazz' }  
{ _id: ObjectId("62017ce6fd435937399d6b5b"),  
  class: { conf: 'Eastern', div: 'C' },  
  points: 26,  
  new_team: 'Celtics' }  
{ _id: ObjectId("62017ce6fd435937399d6b5c"),
```

```
class: { conf: 'Eastern', div: 'D' },
points: 33,
new_team: 'Cavs' }
{ _id: ObjectId("62017ce6fd435937399d6b5d"),
class: { conf: 'Eastern', div: 'D' },
points: 38,
new_team: 'Nets' }
```

Notice that the team field has been renamed to new\_team for every document.

#### Example 2: Rename Multiple Fields

We can use the following code to rename the team field to new\_team and the points field to new\_points:

```
db.teams.updateMany({},
{$rename:{"team":"new_team",
"points":"new_points"}}, false, true)
```

Here are what the documents look like now:

```
{ _id: ObjectId("62017ce6fd435937399d6b58"),
class: { conf: 'Western', div: 'A' },
new_team: 'Mavs',
```

```
new_points: 31 }
{ _id: ObjectId("62017ce6fd435937399d6b59"),
  class: { conf: 'Western', div: 'A' },
  new_team: 'Spurs',
  new_points: 22 }
{ _id: ObjectId("62017ce6fd435937399d6b5a"),
  class: { conf: 'Western', div: 'B' },
  new_team: 'Jazz',
  new_points: 19 }
{ _id: ObjectId("62017ce6fd435937399d6b5b"),
  class: { conf: 'Eastern', div: 'C' },
  new_team: 'Celtics',
  new_points: 26 }
{ _id: ObjectId("62017ce6fd435937399d6b5c"),
  class: { conf: 'Eastern', div: 'D' },
  new_team: 'Cavs',
  new_points: 33 }
{ _id: ObjectId("62017ce6fd435937399d6b5d"),
  class: { conf: 'Eastern', div: 'D' },
  new_team: 'Nets',
  new_points: 38 }
```

**Notice that the team field and the points field have both been renamed in each document.**

### Example 3: Rename Subfield

We can use the following code to rename the div subfield within the class field to division:

```
db.teams.updateMany({},  
{$rename:{"class.div":"class.division"}}, false, true)
```

Here are what the documents look like now:

```
{_id: ObjectId("62017e21fd435937399d6b5e"),  
team: 'Mavs',  
class: { conf: 'Western', division: 'A' },  
points: 31 }  
{_id: ObjectId("62017e21fd435937399d6b5f"),  
team: 'Spurs',  
class: { conf: 'Western', division: 'A' },  
points: 22 }  
{_id: ObjectId("62017e21fd435937399d6b60"),  
team: 'Jazz',  
class: { conf: 'Western', division: 'B' },  
points: 19 }  
{_id: ObjectId("62017e21fd435937399d6b61"),  
team: 'Celtics',  
class: { conf: 'Eastern', division: 'C' },
```

```
points: 26 }
{ _id: ObjectId("62017e21fd435937399d6b62"),
team: 'Cavs',
class: { conf: 'Eastern', division: 'D' },
points: 33 }
{ _id: ObjectId("62017e21fd435937399d6b63"),
team: 'Nets',
class: { conf: 'Eastern', division: 'D' },
points: 38 }
```

Notice that the div subfield within the class field has been renamed to division in each document.

**Note:** You can find the complete documentation for the `$rename` function .

#### Additional Resources

The following tutorials explain how to perform other common operations in MongoDB: