Dot Notation

MongoDB uses the *dot notation* to access the elements of an array and to access the fields of an embedded document.

Arrays

To specify or access an element of an array by the zero-based index position, concatenate the array name with the dot (`.`) and zero-based index position

- To specify the third element in the `contribs` array, use the dot notation `"contribs.2"`.

Embedded Documents

To specify or access a field of an embedded document with dot notation, concatenate the embedded document name with the dot (`.`) and the field name,

- To specify the field named `last` in the `name` field, use the dot notation `"name.last"`.
- To specify the `number` in the `phone` document in the `contact` field, use the dot notation `"contact.phone.number"`

db.collection.remove()

The `db.collection.remove()` method can have one of two syntaxes. The `remove()` method can take a query document and an optional `justOne` Boolean

Behavior

Write Concern

The `remove()` method uses the `delete` command, which uses the default write concern. To specify a different write concern, include the write concern in the options parameter.

Query Considerations

By default, `remove()` removes all documents that match the `query` expression. Specify the `justOne` option to limit the operation to removing a single document. To delete a single document sorted by a specified order, use the findAndModify() method.

When removing multiple documents, the remove operation may interleave with other read and/or write operations to the collection.

Time Series Collections

You cannot use the `remove()` method on a time series collection.

Sharded Collections

All `remove()` operations for a sharded collection that specify the `justOne: true` option must include the shard key *or* the `_id` field in the query specification.

`remove()` operations specifying `justOne: true` in a sharded collection which do not contain either the shard key or the `_id` field return an error.

Remove a Field from a MongoDB Document $unset

unset

The `$unset` operator deletes a particular field.

The specified value in the `$unset` expression (i.e. `""`) does not impact the operation.

To specify a `<field>` in an embedded document or in an array, use dot notation.

Behavior

Starting in MongoDB 5.0, update operators process document fields with string-based names in lexicographic order. Fields with numeric names are processed in numeric order. See Update Operators Behavior for details.

If the field does not exist, then `$unset` does nothing (i.e. no operation).

When used with `$` to match an array element, `$unset` replaces the matching element with `null` rather than removing the matching element from the array. This behavior keeps consistent the array size and element positions.

Starting in MongoDB 5.0, `mongod` no longer raises an error when you use an update operator like `$unset` with an empty operand expression (`{ }`). An empty update results in no changes and no oplog entry is created (meaning that the operation is a no-op).