Создание, обновление и удаление документов

Вставка документов

Вставка – основной метод добавления данных в MongoDB. Чтобы вставить один документ, используйте метод коллекции insertOne:

```
> db.movies.insertOne({"title" : "Stand by Me"})
```

insertOne добавит в документ ключ "_id" (если вы его не предоставили) и сохранит документ в MongoDB.

Вставка документов

Если вам нужно вставить несколько документов в коллекцию, можно использовать метод insertMany. Этот метод позволяет передавать массив документов в базу данных, что гораздо более эффективно. В оболочке можно опробовать это следующим образом:

```
> db.movies.insertMany([
{"title" : "Ghostbusters"}, {"title" : "E.T."},
{"title" : "Blade Runner"}]);
```

Вставка документов

При выполнении массовой вставки с использованием метода insertMany, если документ посреди массива выдает какую-либо ошибку, что произойдет, зависит от того, выбрали ли вы упорядоченные или неупорядоченные операции. В качестве второго параметра insertMany можно указать документ опций. Укажите значение true для ключа "ordered" в документе параметров, чтобы обеспечить вставку документов в том порядке, в котором они были предоставлены. Укажите значение false, и MongoDB может изменить порядок вставок для повышения производительности.

Вставка документов

```
db.movies.insertMany([
  {"_id" : 2, "title" : "Sixteen Candles"},
  {"_id" : 3, "title" : "The Terminator"},
  {"_id" : 3, "title" : "The Princess Bride"},
  {"_id" : 4, "title" : "Scarface"}],
  {"ordered" : false}
)
```

В этом примере, поскольку упорядоченные вставки используются по умолчанию, будут вставлены только первые два документа. Третий документ выдаст ошибку, потому что нельзя вставить два документа с одинаковым "_id“

Если вместо этого мы указываем неупорядоченные вставки, первый, второй и четвертый документы в массиве вставляются. Единственная неудачная вставка – третий документ, опять же по причине дублирующейся ошибки "_id"

Удаление документов

Теперь, когда в нашей базе данных есть данные, давайте удалим их. Для этой цели CRUD API предоставляет методы deleteOne и deleteMany. Оба этих метода принимают документ фильтра в качестве первого параметра.

Фильтр задает набор критериев для сопоставления при удалении доку-ментов. Чтобы удалить документ со значением "_id", равным 4, мы использу-ем метод deleteOne в оболочке mongo:

```
> db.movies.deleteOne({"_id" : 4})
```

Удаление документов

В этом примере мы использовали фильтр, который может соответствовать только одному документу, поскольку значения "_id" уникальны в коллекции. Однако мы также можем указать фильтр, который соответствует нескольким документам в коллекции. В этом случае метод deleteOne удалит первый найденный документ, который соответствует фильтру. Какой документ будет найден первым, зависит от нескольких факторов, в том числе от порядка, в котором были вставлены документы, того, какие обновления были внесены в них (для некоторых подсистем хранения), и того, какие индексы указаны. Как и при любой операции с базой данных, убедитесь, что вы знаете, как использование метода deleteOne повлияет на ваши данные.

Удаление документов

Чтобы удалить все документы, которые соответствуют фильтру, используйте метод deleteMany:

```
>db.movies.deleteMany({"year" : 1984})
```

Можно использовать метод deleteMany, чтобы удалить все документы в коллекции:

```
> db.movies.deleteMany({})
```

Удаление документов обычно является довольно быстрой операцией.

Однако если вы хотите очистить всю коллекцию, ее быстрее удалить с помощью метода drop:

```
> db.movies.drop()
```

Удаление документов

Как только данные удаляются, они исчезают навсегда. Операции drop и delete невозможно отменить, равно как нельзя восстановить удаленные документы, разумеется, если только речь не идет о восстановлении предварительно сохраненной версии данных.