Mongo Database

Scale the web

Course Material

You can access course material via this URL:

http://tinyurl.com/iti-mongo

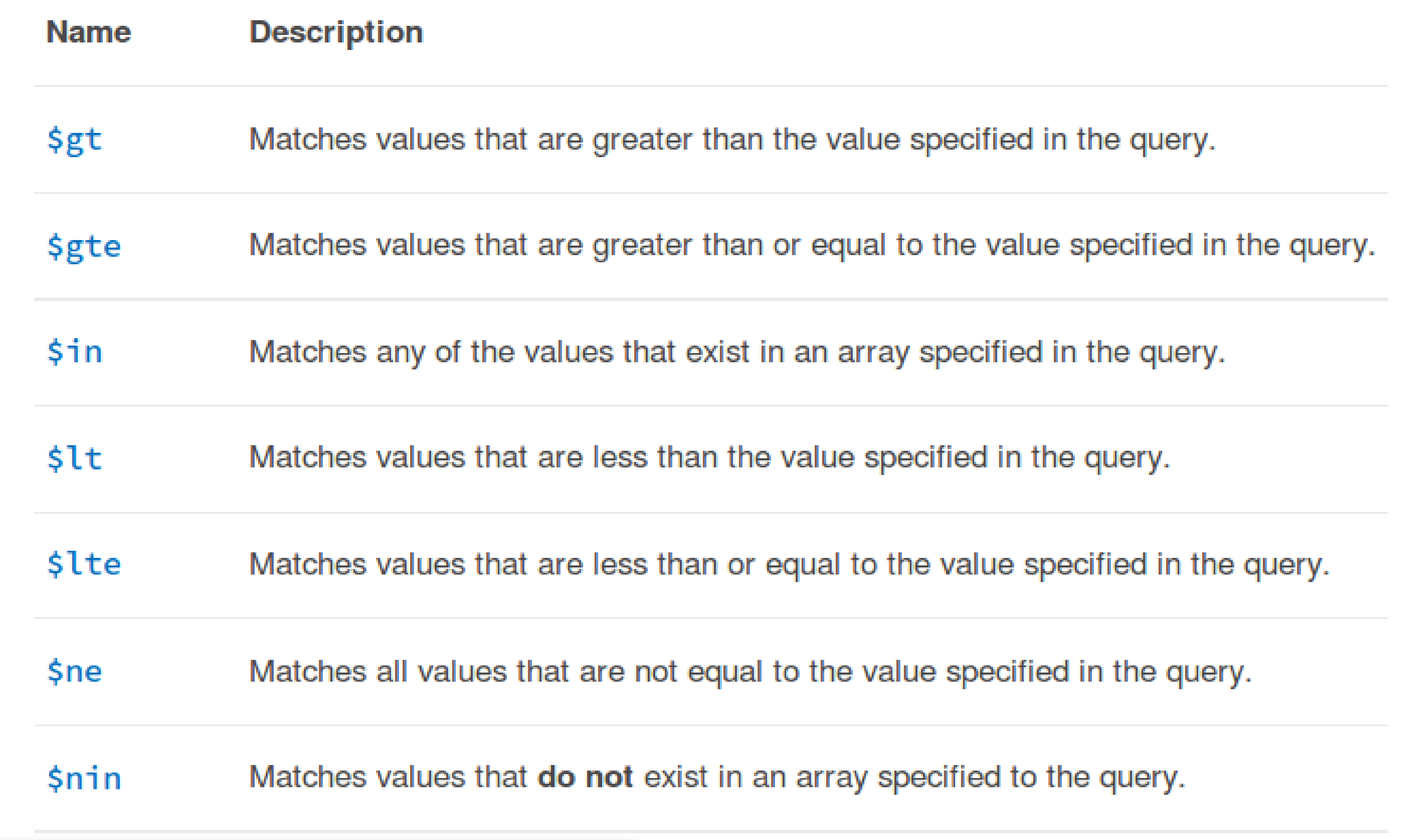
Agenda

- Mongo query operators
  - Comparison
  - Logical
  - Arrays
  - Element
- Update and upsert
- Mongo indexes
- Drop Databases
- Export and import data

Mongo Query Operators

Mongo Query Operators

Query Comparison Operators



Name	Description
$gt	Matches values that are greater than the value specified in the query.
$gte	Matches values that are greater than or equal to the value specified in the query.
$in	Matches any of the values that exist in an array specified in the query.
$lt	Matches values that are less than the value specified in the query.
$lte	Matches values that are less than or equal to the value specified in the query.
$ne	Matches all values that are not equal to the value specified in the query.
$nin	Matches values that **do not** exist in an array specified to the query.

Mongo Query Operators

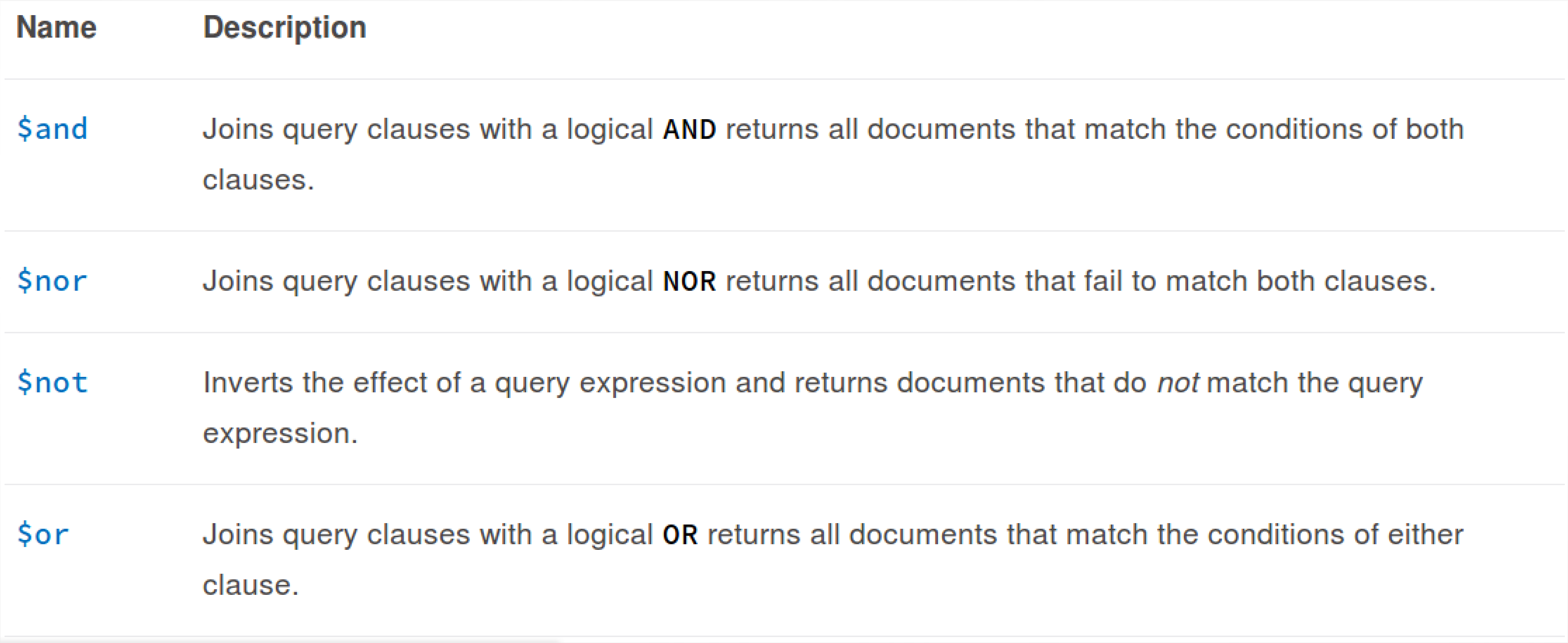
Example:

```
db.inventory.find( { type: { $in: [ 'food', 'snacks' ] } } )
```

```
db.inventory.find( { qty: { $gt: 20 } } )
```

Mongo Query Operators

Query Logical Operators



Name	Description
$and	Joins query clauses with a logical AND returns all documents that match the conditions of both clauses.
$nor	Joins query clauses with a logical NOR returns all documents that fail to match both clauses.
$not	Inverts the effect of a query expression and returns documents that do *not* match the query expression.
$or	Joins query clauses with a logical OR returns all documents that match the conditions of either clause.

Mongo Query Operators

Example:

```
db.inventory.find( { $and: [ { price: { $ne: 1.99 } }, { price: { $exists: true } } ] } )
```

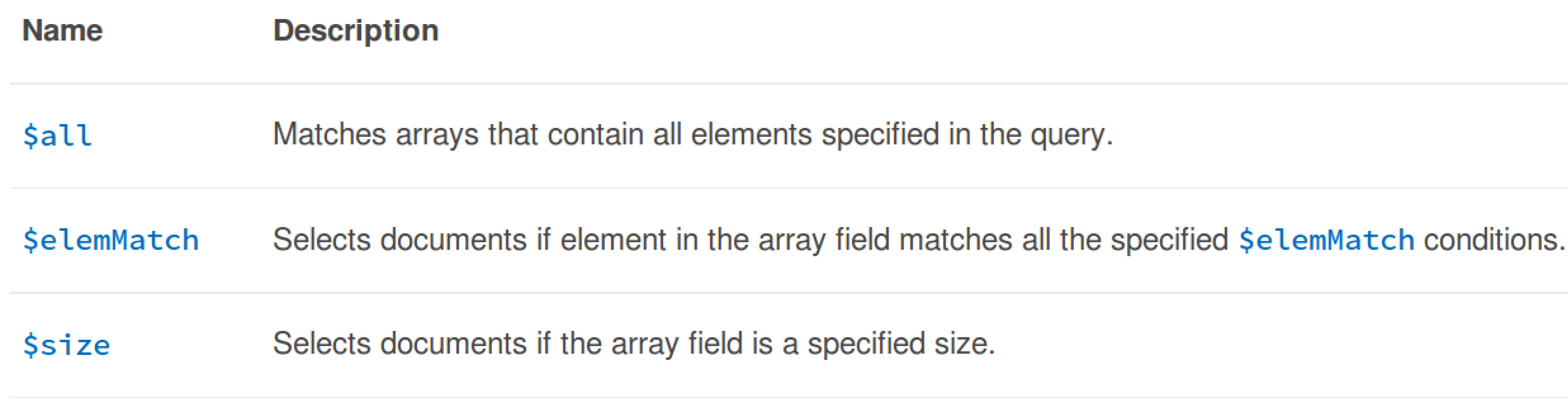
- the price field value is not equal to 1.99 and
- the price field exists.

```
db.inventory.find( { price: { $not: { $gt: 1.99 } } } )
```

- the price field value is less than or equal to 1.99 or
- the price field does not exist

Mongo Query Operators

Query Arrays Operators



Name	Description
`$all`	Matches arrays that contain all elements specified in the query.
`$elemMatch`	Selects documents if element in the array field matches all the specified `$elemMatch` conditions.
`$size`	Selects documents if the array field is a specified size.

Mongo Query Operators

Example:

```
{ tags: { $all: [ "ssl" , "security" ] } }
```

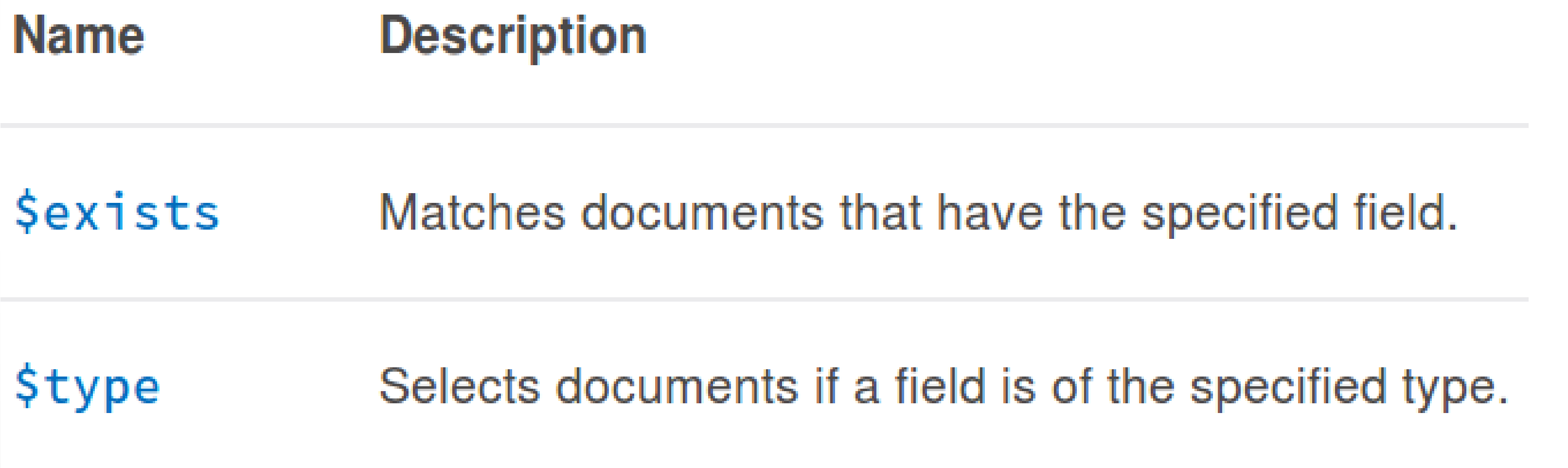
- the tags field value is an array and should contain ssl, security elements

```
db.scores.find(
   { results: { $elemMatch: { $gte: 80, $lt: 85 } } }
)
```

- The score is an array and should contains at least one element matches this query (i.e. 82)

Mongo Query Operators

Query Element Operators



Name	Description
`$exists`	Matches documents that have the specified field.
`$type`	Selects documents if a field is of the specified type.

Mongo Query Operators

Example:

```
db.inventory.find( { qty: { $exists: true, $nin: [ 5, 15 ] } } )
```

- This query will select all documents in the inventory collection where the qty field exists and its value does not equal 5 or 15.

```
db.inventory.find( { tags: { $type : 2 } } );
```

- This will list all documents containing a tags field that is either a string or an array holding at least one string.

http://docs.mongodb.org/manual/reference/operator/query/type/#type

Update and Upsert

Update and Upsert

- MongoDB provides the update() method to update the documents of a collection. The method accepts as its parameters:
  - an update conditions document to match the documents to update,
  - an update document to specify the modification to perform, and
  - an options document.

- To specify the update condition, use the same structure and syntax as the query conditions.

Update and Upsert

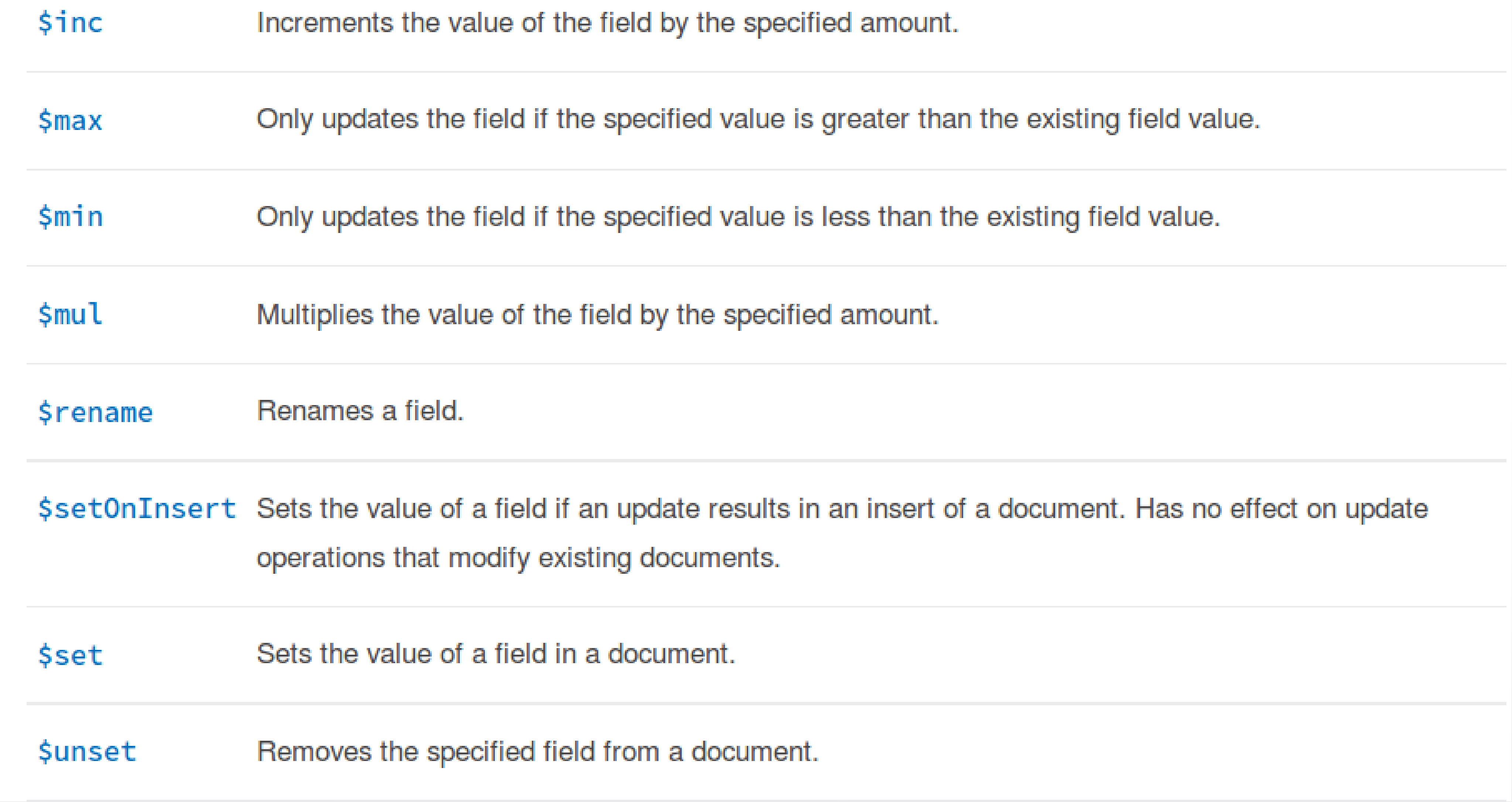
Examples:

```
db.inventory.update(
   { item: "MNO2" },
        {
        $set: {
            category: "apparel",
            details: { model: "14Q3", manufacturer: "XYZ Company" }
        },
   }
)
```

*** Note The multi operator and the embedded documents update feature**

Update and Upsert

Update Operators



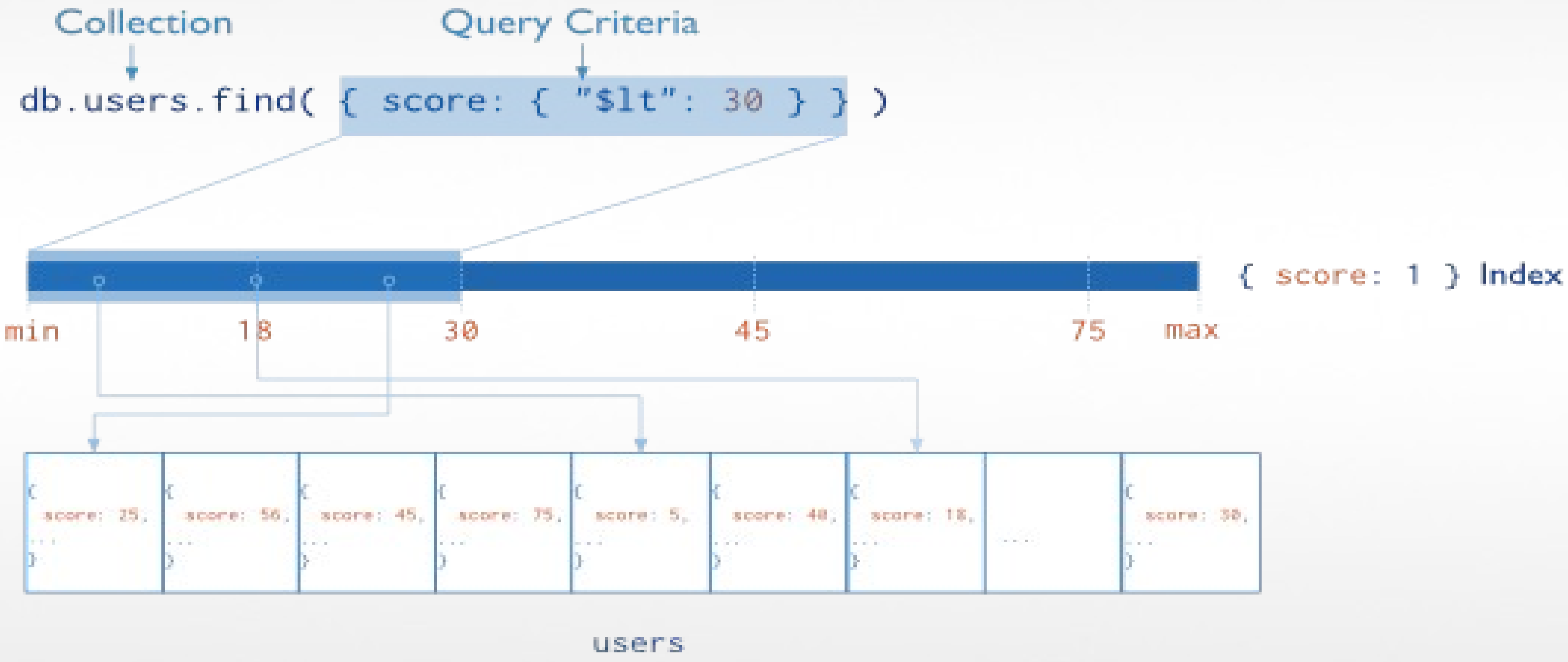
`$inc`	Increments the value of the field by the specified amount.
`$max`	Only updates the field if the specified value is greater than the existing field value.
`$min`	Only updates the field if the specified value is less than the existing field value.
`$mul`	Multiplies the value of the field by the specified amount.
`$rename`	Renames a field.
`$setOnInsert`	Sets the value of a field if an update results in an insert of a document. Has no effect on update operations that modify existing documents.
`$set`	Sets the value of a field in a document.
`$unset`	Removes the specified field from a document.

Mongo Indexes

Mongo Indexes

- Indexes support the efficient execution of queries in MongoDB.
- Without indexes, **MongoDB** must scan every document in a collection to select those documents that match the query statement.
- Indexes are special data structures [1] that store a small portion of the collection’s data set in an easy to traverse form.
- The index stores the value of a specific field or set of fields, ordered by the value of the field.
- indexes in MongoDB are similar to indexes in other database systems.
- MongoDB defines indexes at the collection level and supports indexes on any field or sub-field of the documents in a MongoDB collection.

Mongo Indexes



Mongo Indexes

Example:

```
db.friends.ensureIndex( { "name" : 1 } )
```

- **Compare the output of explain() function before and after creating index**
- **Try to add unique option for creating Index**

Drop Databases

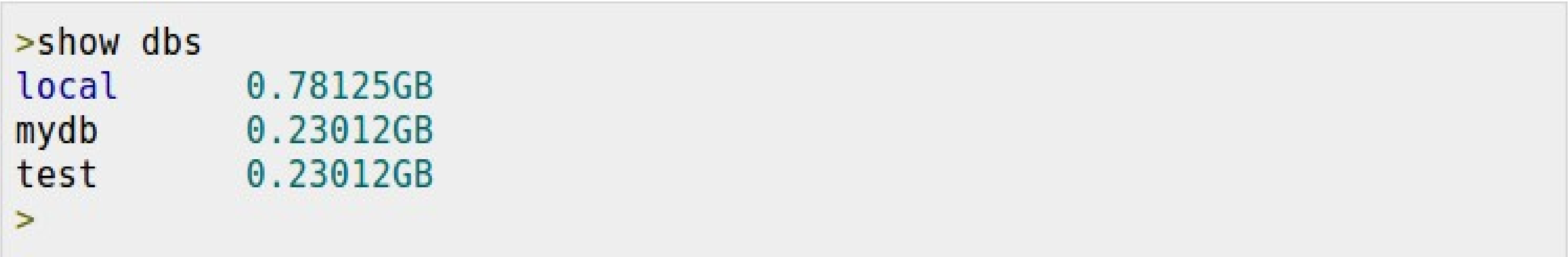
Drop Databases

Basic syntax of **dropDatabase()** command is as follows:

```
db.dropDatabase()
```

Example:

First, check the list available databases by using the command show dbs:



```
>show dbs
local        0.78125GB
mydb         0.23012GB
test         0.23012GB
>
```

Drop Databases

If you want to delete database called 'mydb' , then dropDatabase() command would be as follows:

```
> use mydb
switched to db mydb
> db.dropDatabase()
> { "dropped" : "mydb", "ok" : 1 }
```

Export and Import

Export and Import

```
$mongodump
$mongorestore
```