QUIZ

Which of the following MongoDB commands would return us detailed information about a query on the `movies` collection, such as how many documents were examined and returned?

```
db.movies.find({actor: “Leonardo DiCaprio”}).find("executionStats");
```

```
db.movies.find({actor: “Leonardo DiCaprio”}).explain("operationStats");
```

```
db.movies.find({actor: “Leonardo DiCaprio”}).explain("executionStats");
```

👏 Correct! This method would return the details about the `.find()` query.

```
db.movies.find({actor: “Leonardo DiCaprio”}).find("operationStats");
```

Which of the following is a benefit of using indexes to support our queries in MongoDB?

Queries that use indexes take longer to execute but produce more accurate results.

Indexes improve speed and efficiency when querying large collections of data.

👏 Correct!

Indexes store the results of our queries for easy access later on.

Indexes give us important metadata about our queries.

What method is used to delete an index in MongoDB?

`.deleteIndex()`

`.dropIndex()`

👏 Correct! This method helps remove an index from a collection.

`.getIndexes()`

`.destroyIndex()`

If we were to execute the following MongoDB command, what fields would the resulting index **NOT** be able to support queries on?

```
db.books.createIndex({ "author": 1, "title": 1, “year_published”: 1 })
```

The `author` field, the `title` field, and the `year_published` field.

The `title` field.

👏 That’s right! Compound indexes can support beginning subsets of indexed fields.

The `author` field.

The `author` field, and the `title` field.

Fill in the code below to create an index on a MongoDB `authors` collection that references the `books` field in ascending order.

```
db.authors.createIndex({ books : 1 });
```

👏 You got it!

How can we create a multikey index in MongoDB?

We can't. Multikey indexes are not possible in MongoDB.

We need to pass the string, `"multikey"` as the second argument to the `.createIndex()` method.

MongoDB automatically creates a multikey index whenever we index a field that has an array value.

That's right! MongoDB automatically creates an index key for each element in the array.

We can use the `.createMultikeyIndex()` method.

Which of the following statements is true about MongoDB indexes?

Each time a document is inserted or removed from a collection, MongoDB also updates all of that collection's indexes.

Correct! Note that the more indexes that need to be updated, the bigger affect on the time to complete a write operation.

Large numbers of indexes have no implications on the performance of CRUD operations for that collection.

Once an index is created, it is not possible to insert or remove documents from the original collection.

Inserting, updating, or removing documents from a collection does not impact the indexes for that collection.

Which of the following statements about MongoDB compound indexes is correct?

A compound index references multiple fields within a document.

 That's exactly right!

It is not possible to create compound indexes in MongoDB.

Compound indexes can not be used to support queries on single fields.

Compound indexes can be created using the `.createCompoundIndex()` method.

Which MongoDB method is used to create an index on a collection?

`.findIndex()`

`.dropIndex()`

`.generateIndex()`

`.createIndex()`

 Correct! The `.createIndex()` method can be used to create new indexes on a collection.