	Bases de dades – MongoDB	
Nom i Cognoms:		Data:

## Documents a Insertar

Introduïu els següents documents en una col·lecció trucada `movies`.

`title : Fight Club`

`writer : Chuck Palahniuk`

`year : 1999`

`actors : [`

`Brad Pitt`

`Edward Norton`

`]`

`title : Pulp Fiction`

`writer : Quentin Tarantino`

`year : 1994`

`actors : [`

`John Travolta`

`Uma Thurman`

`]`

`title : Inglorious Basterds`

`writer : Quentin Tarantino`

`year : 2009`

`actors : [`

`Brad Pitt`

`Diane Kruger`

`Eli Roth`

`]`

`title : The Hobbit: An Unexpected Journey`

`writer : J.R.R. Tolkein`

`year : 2012`

`franchise : The Hobbit`

`title : The Hobbit: The Desolation of Smaug`

`writer : J.R.R. Tolkein`

`year : 2013`

`franchise : The Hobbit`

`title : The Hobbit: The Battle of the Five Armies`

`writer : J.R.R. Tolkein`


`year : 2012`


`franchise : The Hobbit`

`synopsis : Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness.`

`title : Pee Wee Herman's Big Adventure`

`title : Avatar`

	Codi: Exàmens	Data del Format:24-10-2014	Pàgina 1 de 8
	C/ Ferrer i Busquets, 17	25230-Mollerussa	E-mail: <a href="mailto:lasallemollerussa@lasalle.cat">lasallemollerussa@lasalle.cat</a>
	Tel. 973 600 270	Fax 973 710 599	<a href="http://www.mollerussa.lasalle.cat">http://www.mollerussa.lasalle.cat</a>

	Bases de dades – MongoDB	
Nom i Cognoms:		Data:

## Consultes / Buscar documents

Realitzeu les següents consultes a la col·lecció movies:

1. Obtenir tots els documents

```
public static void consulta1() {
    String uri = "mongodb+srv://administrador:pFh46b1AjbAH1aAw@cluster0.x4jmrz0.mongodb.net/?retryWrites=true&w=majority";

    try (MongoClient mongoClient = MongoClient.create(uri))
    {
        MongoDB database = mongoClient.getDatabase(s: "Practica_MongoDB_1");
        MongoCollection<Document> collection = database.getCollection(s: "Practic");
        System.out.println("-----");
        FindIterable<Document> documento = collection.find();
        for (Document doc : documento) {
            System.out.println(doc.toJson());
        }
    }
}
```

2. Obtenir documents amb writer igual a "Quentin Tarantino"

```
public static void consulta2() {
    String uri = "mongodb+srv://administrador:pFh46b1AjbAH1aAw@cluster0.x4jmrz0.mongodb.net/?retryWrites=true&w=majority";

    try (MongoClient mongoClient = MongoClient.create(uri))
    {
        MongoDB database = mongoClient.getDatabase(s: "Practica_MongoDB_1");
        MongoCollection<Document> collection = database.getCollection(s: "Practic");
        System.out.println("-----");
        FindIterable<Document> write = collection.find(eq(fieldName: "writer", value: "Quentin Tarantino"));
        for (Document document : write) {
            System.out.println(document.toJson());
        }
    }
}
```

3. Obtenir documents amb actors inclosos a "Brad Pitt"

Nom i Cognoms:

Data:

```
public static void consulta3() {
    String uri = "mongodb+srv://administrador:pFh46b1AjbaH1aAw@cluster0.x4jmrz0.mongodb.net/?retryWrites=true&w=majori";

    try (MongoClient mongoClient = MongoClient.create(uri))
    {
        MongoDB database = mongoClient.getDatabase(s: "Practica_MongoDB_1");
        MongoCollection<Document> collection = database.getCollection(s: "Practic");
        System.out.println("-----");
        FindIterable<Document> actors = collection.find(eq( fieldName: "actors", value: "Brad Pitt"));
        for (Document document : actors) {
            System.out.println(document.toJson());
        }
    }
}
```

4. Obtenir documents amb franchise igual a "El Hobbit"

```
public static void consulta4() {
    String uri = "mongodb+srv://administrador:pFh46b1AjbaH1aAw@cluster0.x4jmrz0.mongodb.net/?retryWrites=true&w=majori";

    try (MongoClient mongoClient = MongoClient.create(uri))
    {
        MongoDB database = mongoClient.getDatabase(s: "Practica_MongoDB_1");
        MongoCollection<Document> collection = database.getCollection(s: "Practic");
        System.out.println("-----");
        FindIterable<Document> franchise = collection.find(eq( fieldName: "franchise", value: "The Hobbit"));
        for (Document document : franchise) {
            System.out.println(document.toJson());
        }
    }
}
```

5. Obtenir totes les pel·lícules dels anys 90.

```
public static void consulta5() {
    String uri = "mongodb+srv://administrador:pFh46b1AjbaH1aAw@cluster0.x4jmrz0.mongodb.net/?retryWrites=true&w=majori";

    try (MongoClient mongoClient = MongoClient.create(uri))
    {
        MongoDB database = mongoClient.getDatabase(s: "Practica_MongoDB_1");
        MongoCollection<Document> collection = database.getCollection(s: "Practic");
        System.out.println("-----");
        FindIterable<Document> noventa = collection.find(and(gt( fieldName: "year", value: 1900),lt( fieldName: "year", value: 2000)));
        for (Document document : noventa) {
            System.out.println(document.toJson());
        }
    }
}
```

6. Obtenir les pel·lícules estrenades entre l'any 2000 i 2010.

Nom i Cognoms:

Data:

```
public static void consulta6() {
    String uri = "mongodb+srv://administrador:pFh46b1AjbaH1aAw@cluster0.x4jmrz0.mongodb.net/?retryWrites=true&w=majori

    try (MongoClient mongoClient = MongoClient.create(uri))
    {
        MongoDB database = mongoClient.getDatabase(s: "Practica_MongoDB_1");
        MongoCollection<Document> collection = database.getCollection(s: "Practic");
        System.out.println("-----");
        FindIterable<Document> estreno = collection.find(and(gt( fieldName: "year", value: 2000),lt( fieldName: "year", value:
        for (Document document : estreno) {
            System.out.println(document.toJson());
        }
    }
}
```

## Actualitzar documents

1. Agregar sinopsis a " The Hobbit: An Unexpected Journey ": "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."

```
public static void actualizar1() {
    String uri = "mongodb+srv://administrador:pFh46b1AjbaH1aAw@cluster0.x4jmrz0.mongodb.net/?retryWrites=true&w=majori

    try (MongoClient mongoClient = MongoClient.create(uri))
    {
        Scanner reader = new Scanner(System.in);
        MongoDB database = mongoClient.getDatabase(s: "Practica_MongoDB_1");
        MongoCollection<Document> collection = database.getCollection(s: "Practic");
        System.out.println("-----");
        String titulo = "The Hobbit: An Unexpected Journey";
        String opinion;
        System.out.println("ESCRIBE TU RESEÑA SOBRE |The Hobbit: An Unexpected Journey|");
        opinion = reader.nextLine();
        collection.updateOne(filters.eq( fieldName: "title", titulo),Updates.set("synopsis", opinion));
        System.out.println("Sinopsis actualizada correctamente para " + titulo);
    }
}
```

2. Agregar sinopsis a "The Hobbit: The Desolation of Smaug" : "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."

Nom i Cognoms:

Data:

```

Usage
public static void actualizar2() {
    String uri = "mongodb+srv://administrador:pFh46bIAjbaH1aAw@cluster0.x4jmrz0.mongodb.net/?retryWrites=true&w=majori

    try (MongoClient mongoClient = MongoClient.create(uri))
    {
        Scanner reader = new Scanner(System.in);
        MongoDB database = mongoClient.getDatabase(s: "Practica_MongoDB_1");
        MongoCollection<Document> collection = database.getCollection(s: "Practic");
        System.out.println("-----");
        String titulo = "The Hobbit: The Desolation of Smaug";
        String opinion;
        System.out.println("ESCRIBE TU RESEÑA SOBRE |The Hobbit: The Desolation of Smaug|");
        opinion = reader.nextLine();
        collection.updateOne(Filters.eq( fieldName: "title", titulo),Updates.set("synopsis", opinion));
        System.out.println("Sinopsis actualizada correctamente para " + titulo);
    }
}

```

3. Agregar un actor anomenat "Samuel L. Jackson" a la pel·lícula "Pulp Fiction"

```

public static void actualizar3() {
    String uri = "mongodb+srv://administrador:pFh46bIAjbaH1aAw@cluster0.x4jmrz0.mongodb.net/?retryWrites=true&w=majori

    try (MongoClient mongoClient = MongoClient.create(uri))
    {
        Scanner reader = new Scanner(System.in);
        MongoDB database = mongoClient.getDatabase(s: "Practica_MongoDB_1");
        MongoCollection<Document> collection = database.getCollection(s: "Practic");
        System.out.println("-----");
        String titulo = "Pulp Fiction";
        String actor;
        System.out.println("AGREGA UN ACTOR A LA PELICULA DEL |Pulp Fiction|");
        actor = reader.nextLine();
        collection.updateOne(Filters.eq( fieldName: "title", titulo),Updates.push( fieldName: "actors", actor));
        System.out.println("Actor agregado correctamente para " + titulo);
    }
}

```

## Búsqueda de Text / Cerca de text

1. Trobar les pel·lícules que hi ha a la sinopsi contenguin la paraula "Bilbo"

Nom i Cognoms:

Data:

```
public static void busqueda1() {
    String uri = "mongodb+srv://administrador:pFh46bLAjbaH1aAw@cluster0.x4jmrz0.mongodb.net/?retryWrites=true&w=majori";

    try (MongoClient mongoClient = MongoClient.create(uri))
    {
        MongoDB database = mongoClient.getDatabase(s: "Practica_MongoDB_1");
        MongoCollection<Document> collection = database.getCollection(s: "Practic");
        System.out.println("-----");
        FindIterable<Document> synopsis = collection.find(Filters.regex(fieldName: "synopsis", pattern: "Bilbo", options: "i"));
        for (Document resultado : synopsis) {
            System.out.println(resultado.toJson());
        }
    }
}
```

2. Trobar les pel·lícules que en la sinopsi conté la paraula "Gandalf"

```
public static void busqueda2() {
    String uri = "mongodb+srv://administrador:pFh46bLAjbaH1aAw@cluster0.x4jmrz0.mongodb.net/?retryWrites=true&w=majority&a";

    try (MongoClient mongoClient = MongoClient.create(uri))
    {
        MongoDB database = mongoClient.getDatabase(s: "Practica_MongoDB_1");
        MongoCollection<Document> collection = database.getCollection(s: "Practic");
        System.out.println("-----");
        FindIterable<Document> synopsis = collection.find(Filters.regex(fieldName: "synopsis", pattern: "Gandalf", options: "i"));
        for (Document resultado : synopsis) {
            System.out.println(resultado.toJson());
        }
    }
}
```

3. Trobar les pel·lícules que en la sinopsi conté la paraula "Bilbo" i no la paraula "Gandalf"

```
public static void busqueda3() {
    String uri = "mongodb+srv://administrador:pFh46bLAjbaH1aAw@cluster0.x4jmrz0.mongodb.net/?retryWrites=true&w=majori";

    try (MongoClient mongoClient = MongoClient.create(uri))
    {
        MongoDB database = mongoClient.getDatabase(s: "Practica_MongoDB_1");
        MongoCollection<Document> collection = database.getCollection(s: "Practic");
        System.out.println("-----");
        FindIterable<Document> synopsis = collection.find(Filters.and(
            Filters.regex(fieldName: "synopsis", pattern: "Bilbo", options: "i"),
            Filters.not(Filters.regex(fieldName: "synopsis", pattern: "Gandalf", options: "i"))
        ));
        for (Document resultado : synopsis) {
            System.out.println(resultado.toJson());
        }
    }
}
```

4. Trobar les pel·lícules que en la sinopsi contenguin la paraula "dwarves" o "hobbit"

Nom i Cognoms:

Data:

```
public static void busqueda4() {
    String uri = "mongodb+srv://administrador:pFh46b1AjbaH1aAw@cluster0.x4jmrz0.mongodb.net/?retryWrites=true&w=majori";
    try (MongoClient mongoClient = MongoClient.create(uri))
    {
        MongoDBDatabase database = mongoClient.getDatabase(s: "Practica_MongoDB_1");
        MongoCollection<Document> collection = database.getCollection(s: "Practic");
        System.out.println("-----");
        FindIterable<Document> synopsis = collection.find(Filters.or(
            Filters.regex(fieldName: "synopsis", pattern: "dwarves", options: "i"),
            Filters.regex(fieldName: "synopsis", pattern: "hobbit", options: "i")));
        for (Document resultado : synopsis) {
            System.out.println(resultado.toJson());
        }
    }
}
```

5. Trobar les pel·lícules que a la sinopsi conté la paraula "gold" i "dragon"

```
public static void busqueda5() {
    String uri = "mongodb+srv://administrador:pFh46b1AjbaH1aAw@cluster0.x4jmrz0.mongodb.net/?retryWrites=true&w=majori";
    try (MongoClient mongoClient = MongoClient.create(uri))
    {
        MongoDBDatabase database = mongoClient.getDatabase(s: "Practica_MongoDB_1");
        MongoCollection<Document> collection = database.getCollection(s: "Practic");
        System.out.println("-----");
        FindIterable<Document> synopsis = collection.find(Filters.and(
            Filters.regex(fieldName: "synopsis", pattern: "gold", options: "i"),
            Filters.regex(fieldName: "synopsis", pattern: "dragon", options: "i")));
        for (Document resultado : synopsis) {
            System.out.println(resultado.toJson());
        }
    }
}
```

## Eliminar documents

1. Eliminar la pel·lícula "Pee Wee Herman's Big Adventure"



Nom i Cognoms:

Data:

```
public static void eliminar1() {
    String uri = "mongodb+srv://administrador:pFh46b1AjbaH1aAw@cluster0.x4jmrz0.mongodb.net/?retryWrites=true&w=majori

    try (MongoClient mongoClient = MongoClient.create(uri))
    {
        MongoDBDatabase database = mongoClient.getDatabase(s: "Practica_MongoDB_1");
        MongoCollection<Document> collection = database.getCollection(s: "Practic");
        System.out.println("-----");
        String titulo = "Pee Wee Herman's Big Adventure";
        collection.deleteOne(Filters.eq(fieldName: "title", titulo));
        System.out.println("Película eliminada con éxito.");
    }
}
```

## 2. Eliminar la película "Avatar"

```
public static void eliminar2() {
    String uri = "mongodb+srv://administrador:pFh46b1AjbaH1aAw@cluster0.x4jmrz0.mongodb.net/?retryWrites=true&w=majori

    try (MongoClient mongoClient = MongoClient.create(uri))
    {
        MongoDBDatabase database = mongoClient.getDatabase(s: "Practica_MongoDB_1");
        MongoCollection<Document> collection = database.getCollection(s: "Practic");
        System.out.println("-----");
        String titulo = "Avatar";
        collection.deleteOne(Filters.eq(fieldName: "title", titulo));
        System.out.println("Película eliminada con éxito.");
    }
}
```