Czech Technical University in Prague, Faculty of Information Technology

MIE-PDB: **Advanced Database Systems**

http://www.ksi.mff.cuni.cz/~svoboda/courses/2016-2-MIE-PDB/

Practical Class 4

MongoDB

Martin Svoboda

svoboda@ksi.mff.cuni.cz

28. 4. 2017

Charles University, Faculty of Mathematics and Physics

NDBI040: **Big Data Management and NoSQL Databases**

Data Model

Database system structure

Instance $\rightarrow$ **databases** $\rightarrow$ **collections** $\rightarrow$ **documents**

- Database
- Collection
  - Collection of documents, usually of a similar structure
- Document
  - MongoDB **document** = **one JSON object**
  - Each document…
    - belongs to right one collection
    - has a **unique immutable identifier** `_id`
  - Field name restrictions apply
    - `_id`, `$`, `.`

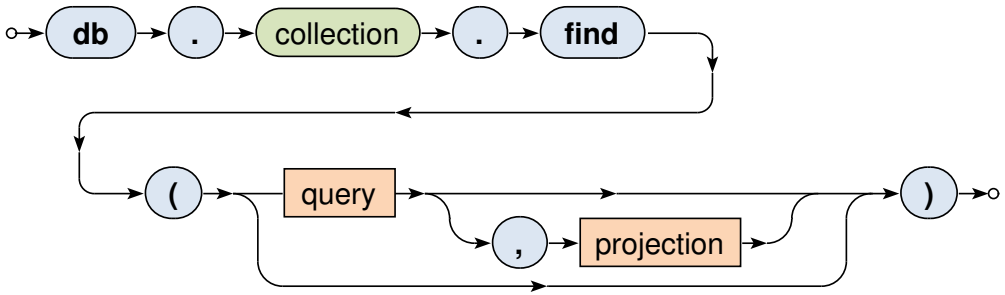
CRUD Operations

Overview

- `db.collection.insert()`
  - Inserts a new document into a collection
- `db.collection.update()`
  - Modifies an existing document / documents or inserts a new one
- `db.collection.remove()`
  - Deletes an existing document / documents
- `db.collection.find()`
  - Finds documents based on filtering conditions
  - Projection and / or sorting may be applied too

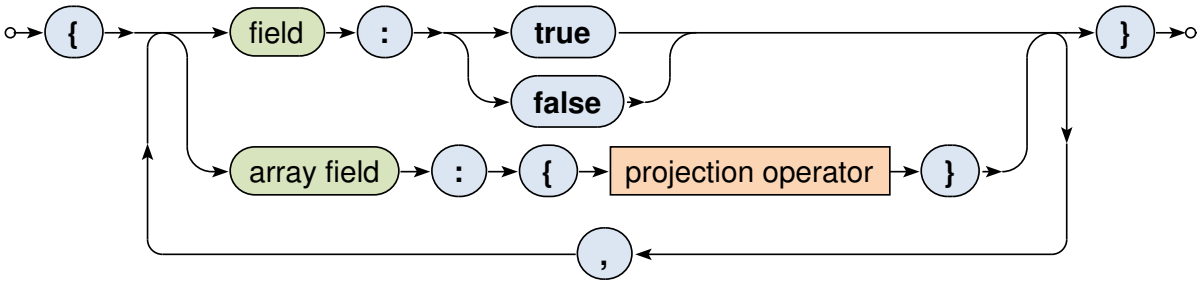
Find Operation

Selects documents from a given collection



Find Operation: Projection

Projection allows us to determine the fields returned in the result

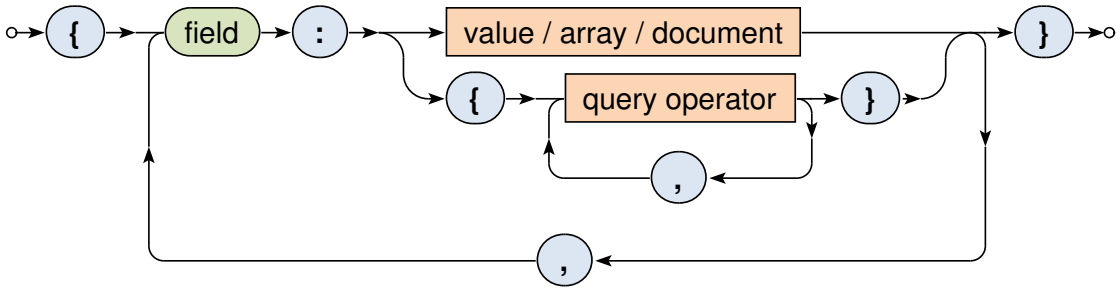


Projection operators

- `$elemMatch`, `$slice`,…

Find Operation: Selection

Query parameter describes the documents we are interested in



Selection operators

- `$eq`, `$neq`, `$lt`, `$lte`, `$gte`, `$gt`
- `$in`, `$nin`
- `$and`, `$or`, `$not`
- `$exists`, `$regex`, `$text`
- …

Sample Queries

Explain the meaning of the following queries

```
db.actors.find({ })
```

```
db.actors.find({ _id: "trojan" })
```

```
db.actors.find({ name: "Ivan Trojan", year: 1964 })
```

```
db.actors.find({ year: { $gte: 1960, $lte: 1980 } })
```

```
db.actors.find({ movies: { $exists: true } })
```

```
db.actors.find({ movies: "medvidek" })
```

```
db.actors.find({ movies: { $in: [ "medvidek", "pelisky" ] } })
```

```
db.actors.find({ movies: { $all: [ "medvidek", "pelisky" ] } })
```

Sample Queries

Explain the meaning of the following queries

```
db.actors.find({ $or: [ { year: 1964 }, { rating: { $gte: 3 } } ] })
```

```
db.actors.find({ rating: { $not: { $gte: 3 } } })
```

```
db.actors.find({ }, { name: 1, year: 1 })
```

```
db.actors.find({ }, { movies: 0, _id: 0 })
```

```
db.actors.find({ }, { name: 1, movies: { $slice: 2 }, _id: 0 })
```

```
db.actors.find().sort({ year: 1, name: -1 })
```

```
db.actors.find().sort({ name: 1 }).skip(1).limit(2)
```

```
db.actors.find().sort({ name: 1 }).limit(2).skip(1)
```

Exercise 1

Express the following MongoDB query

- **Find actors born in *1966* with first name *Jiri***

Exercise 2

Express the following MongoDB query

- **Find movies directed by *Jan Hrebejk***
- Note that the order of fields for first and last names is arbitrary

Exercise 3

Express the following MongoDB query

- **Find actors with first name *Jiri* who played in *Medvidek* movie**
- Return names of these actors only

Exercise 4

Express the following MongoDB query

- **Find movies shot between years *2000* and *2005* such that they have a director specified**
- Return movie identifier only
- Order the result by ratings in descending order and then by years in ascending order

Exercise 5

Express the following MongoDB query

- **Find actors who stared in *Samotari* or *Medvidek* movies**
- Return actor identifier only
- Find two different approaches

Exercise 6

Express the following MongoDB query

- **Find actors who played in both *Samotari* and *Medvidek***
- Return actor identifier only
- Find two different approaches

Exercise 7

Express the following MongoDB query

- **Find movies with Czech title equal to *Vratne lahve***
- Return movie title only
- Note that there are two means how movie titles are defined

Exercise 8

Express the following MongoDB query

- Find actors having their movies defined as an array
- **Return actor identifier and the second and third movie only (if any)**

Exercise 9

Express the following MongoDB query

- **Find movies in which at least one actor with an identifier ending with *ova* played**
- Return movie identifier and the first actor who satisfies such a condition

Exercise 10

Express the following MongoDB query

- **Find movies that have a *Czech Lion* award from *2005***
- Return movie identifier and all awards

Exercise 11

Express the following MongoDB query

- **Find movies that are *comedies* and *dramas* at the same time**
  or
  have a rating *80* or more
- Return movie identifier only