MongoDB

Web programiranje

Jurica Maltar

mongoDB

- NOSQL (“not only SQL”)

- mongoDB:
  - Kolekcije (collection)
  - Dokumenti (document)
  - Polje (field)

- SQL:
  - Tablica (table)
  - Retci (row)
  - Ćelija/stupac (column)

mongoDB

- Instalacija: MongoDB server, MongoDB shell

- Početak rada:
  1. Stvoriti `C:\data\db`
  2. Pokrenuti server (`C:\Program Files\MongoDB\Server\8.0\bin\mongod.exe`)
  3. Pokrenuti shell (`C:\Program Files\mongosh\mongosh.exe`)

- Napomena: u starijim verzijama (< 6.0) shell je

```
C:\Program Files\MongoDB\Server\x.y\bin\mongo.exe
```

CRUD

- Operacije nad podacima grupiramo u:
    - **C**reate
    - **R**ead
    - **U**pdate
    - **D**elete
- Asocirajte CRUD operacije s vrstama HTTP zahtjeva

mongoDB (JavaScript)

- `mongo ./mongo_script.js`

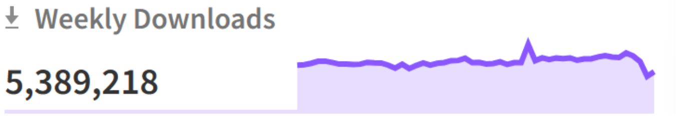
```
const conn = new Mongo();
db = conn.getDB("nastava");
print(db)
print(db.getCollectionNames());

let cursor = db.predavanja.find({});

while (cursor.hasNext())
    printjson(cursor.next());
```

mongoDB (Node.js, službeni paket)

- npm install mongodb



⤓ Weekly Downloads

5,389,218

- require('mongodb’)
- MongoClient
  - connect
- Client
  - db

Stvaranje kolekcije

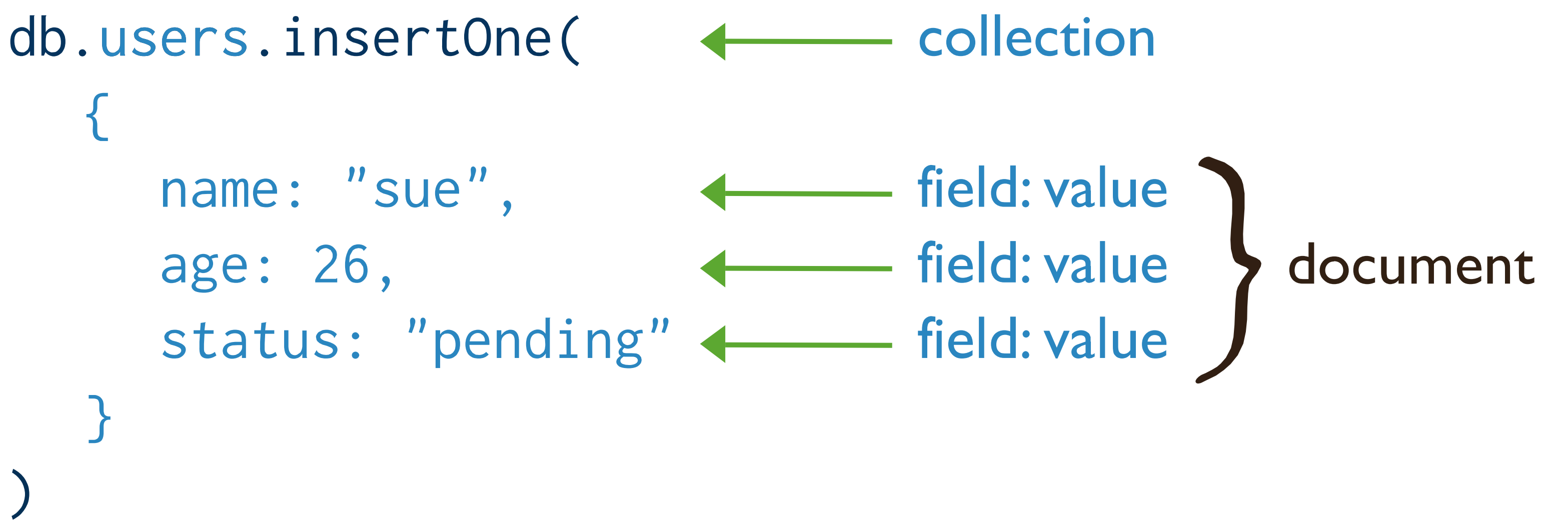
- SQL:

```
CREATE TABLE people (
    id MEDIUMINT NOT NULL
        AUTO_INCREMENT,
    user_id Varchar(30),
    age Number,
    status char(1),
    PRIMARY KEY (id)
);
```

- mongoDB:

```
db.createCollection("people");
```

Create



```
db.users.insertOne(
   {
      name: "sue",
      age: 26,
      status: "pending"
   }
)
```

collection

field: value

field: value

field: value

document

Create

- `insertOne`/`insertMany`
- SQL:

```
INSERT INTO people(user_id, age, status)
VALUES ("bcd001", 45, "A");
```

- mongoDB:

```
db.people.insertOne({
    user_id: "bcd001",
    age: 45,
    status: "A"
});
```

Read

```
db.users.find(
   { age: { $gt: 18 } },
   { name: 1, address: 1 }
).limit(5)
```

collection

query criteria

projection

cursor modifier

Read

- findOne/find
- SQL:

```
SELECT * FROM people;
```

- mongoDB:

```
db.people.find({});
```

Update

```
db.users.updateMany(
   { age: { $lt: 18 } },
   { $set: { status: "reject” } }
)
```

collection

update filter

update action

Update

- updateOne/updateMany/replaceOne
- SQL:

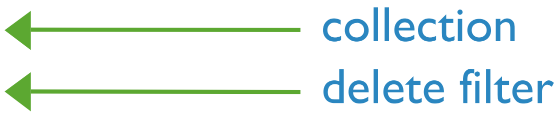
```
UPDATE people SET status = "C"
WHERE age > 25;
```

- mongoDB:

```
db.people.updateMany(
    { age: { $gt: 25 } },
    { $set: { status: "C" } }
);
```

Delete

```
db.users.deleteMany(
   { status: "reject" }
)
```



collection

delete filter

Delete

- `deleteOne`/`deleteMany`
- SQL:

```
DELETE FROM people
WHERE status = "D";
```

- mongoDB:

```
db.people.deleteMany(
    { status: "D" }
);
```