Pinczel Balázs, ELTE IK, 2013. április 15.

Emlékeztető: NoSQL

- **Célok**:
  - Nagy teljesítmény
  - Magas rendelkezésre állás
- **Elosztott működés**
- **Következmények**:
  - Egyszerűsített adatmodell *(kulcs-érték, dokumentum, big table, stb.)*
  - Egyszerűsített műveletek *(nincs join)*
  - Másféle megközelítés
    - Nem deklaratív lekérdezőnyelvek
    - Backup helyett replikálás
    - Kevésbé kidolgozott tranzakciós modell
    - Stb.

Emlékeztető: mongoDB

- **Adatmodell**: dokumentumok

```
{
   "_id" : ObjectId("5114e0bd42…"),
   "first" : "John",
   "last" : "Doe",
   "age" : 39,
   "interests" : [
      "Reading",
      "Mountain Biking" ]
   "favorites" : {
      "color" : "Blue",
      "sport" : "Soccer" }
}
```

Emlékeztető: mongoDB

- **Alapfogalmak**:

RDBMS		MongoDB
Database	⇨	Database
Table, View	⇨	Collection
Row	⇨	Document (BSON)
Column	⇨	Field
Index	⇨	Index
Join	⇨	Embedded Document
Foreign Key	⇨	Reference
Partition	⇨	Shard

http://docs.mongodb.org/manual/reference/sql-comparison/

Emlékeztető: mongoDB

- **Lehetőségek**:
  - Másodlagos indexek
  - Replikálás
  - Shardolás
  - MapReduce
  - In-Memory működés
  - Sokféle API:
    - C, C++, C#, Erlang, Java, Javascript, Node.js, Perl, PHP, Python, Ruby, Scala

Előkészületek

1. Az operációs rendszernek megfelelő MongoDB csomag letöltése a honlapjukról:
   - http://www.mongodb.org/downloads
2. A letöltött állomány kitömörítése
3. Kliens futtatása
   - A `bin` könyvtárban: `mongo(.exe)` futtatható fájl
   - Indítás: `mongo eszakigrid108.inf.elte.hu`

Adatbázisok

- **Létrehozás**:
  - Automatikusan létrejön, amikor adatok kerülnek bele
- **Listázás**:
  - `show dbs`
- **Váltás**:
  - `use <adatbázis>`
  - *(Nem létezőre is lehet)*

Gyűjtemények

- **Létrehozás**:
  - Automatikusan létrejön, amikor adatok kerülnek bele
- **Listázás**:
  - **`db.getCollectionNames()`**
- **Beszúrás**:
  - **`db.`***`<gyűjtemény>`***`.insert(`***`<dokumentum>`***`)`**
- **Lekérdezés**:
  - **`db.`***`<gyűjtemény>`***`.find()`**
  - **`db.`***`<gyűjtemény>`***`.findOne()`**

Példa

```
> use pinczel
switched to db pinczel
> db.getCollectionNames()
[ ]
> db.test.insert({a:1})
> db.test.insert({a:2})
> db.test.count()
2
> db.test.find()
{ "_id" : ObjectId("516b4ee10f1e45670d23e965"), "a" : 1 }
{ "_id" : ObjectId("516b4ef30f1e45670d23e966"), "a" : 2 }
> db.test.findOne()
{ "_id" : ObjectId("516b4ee10f1e45670d23e965"), "a" : 1 }
> db.getCollectionNames()
[ "system.indexes", "test" ]
>
```

JavaScript

- A mongo kliens, mint JavaScript értelmező:

```
> var i = 3
> i
3
> i+1
4
> i%2
1
> db.test.insert({a:i})
> for (i = 4; i <= 10; ++i ) {
... db.test.insert({a:i})
... }
> db.test.find()
{ "_id" : ObjectId("516b4ee10f1e45670d23e965"), "a" : 1 }
{ "_id" : ObjectId("516b4ef30f1e45670d23e966"), "a" : 2 }
{ "_id" : ObjectId("516b51630f1e45670d23e968"), "a" : 3 }
{ "_id" : ObjectId("516b51aa0f1e45670d23e969"), "a" : 4 }
{ "_id" : ObjectId("516b51aa0f1e45670d23e96a"), "a" : 5 }
{ "_id" : ObjectId("516b51aa0f1e45670d23e96b"), "a" : 6 }
{ "_id" : ObjectId("516b51aa0f1e45670d23e96c"), "a" : 7 }
{ "_id" : ObjectId("516b51aa0f1e45670d23e96d"), "a" : 8 }
{ "_id" : ObjectId("516b51aa0f1e45670d23e96e"), "a" : 9 }
{ "_id" : ObjectId("516b51aa0f1e45670d23e96f"), "a" : 10 }
```

Egyszerű szűrések és vetítések

- **Szűrés adott mezők értékére**:
  - **`find({`*`<mező`*$_1$*`>`*`: `*`<érték`*$_1$*`>`*`, …, `*`<mező`*$_n$*`>`*`: `*`<érték`*$_n$*`>`*`})`**
  - Az összes feltételnek teljesülnie kell
  - Lista típusú mezőnél akkor teljesül, ha legalább egyszer benne van az érték
  - Beágyazva is lehet szűrni: pl. `"favorites.color": "Blue"`
- **Vetítés adott mezőkre**:
  - **`find({…}, {`*`<mező`*$_1$*`>`*`: true, …, `*`<mező`*$_n$*`>`*`: true})`**
    - Visszaadja a megadott mezőket és az `_id`-t
  - **`find({…}, {`*`<mező`*$_1$*`>`*`: false, …, `*`<mező`*$_n$*`>`*`: false})`**
    - Visszaad a megadott mezőkön kívül mindent

JavaScript szűrés

- **Példák**:
  - `find("this.a == 5")`
  - `find(function(){return this.a == 5})`
  - `find({$where: "this.a == 5"})`
  - `find({$where: function(){return this.a == 5}})`

- **Előnyök**:
  - Nagyobb kifejezőerő
- **Hátrányok**:
  - Nem tudja használni az indexeket
  - Nem párhuzamosítható
  - Nem szakítható meg

Szűrési operátorok

- **Operátorokkal érdemes megoldani amit csak lehet**
  - (ld. JavaScript szűrés hátrányai)

- **Összehasonlítás**: `$gt`, `$gte`, `$lt`, `$lte`, `$ne`
- **Halmazhoz tartozás**: `$in`, `$nin`, `$all`
- **Logikai**: `$and`, `$or`, `$not`, `$nor`
- **Mező létezése**: `$exists`
- (**Reguláris kifejezés**: `$regex`)

http://docs.mongodb.org/manual/reference/operators/

Példa

```
> db.test.find({a: {$gt: 7} })
{ "_id" : ObjectId("516b51aa0f1e45670d23e96d"), "a" : 8 }
{ "_id" : ObjectId("516b51aa0f1e45670d23e96e"), "a" : 9 }
{ "_id" : ObjectId("516b51aa0f1e45670d23e96f"), "a" : 10 }
> db.test.find({a: {$in: [3,7,9]} })
{ "_id" : ObjectId("516b51630f1e45670d23e968"), "a" : 3 }
{ "_id" : ObjectId("516b51aa0f1e45670d23e96c"), "a" : 7 }
{ "_id" : ObjectId("516b51aa0f1e45670d23e96e"), "a" : 9 }
> db.test.find({a: {$not: {$mod: [3, 0]} } })
{ "_id" : ObjectId("516b4ee10f1e45670d23e965"), "a" : 1 }
{ "_id" : ObjectId("516b4ef30f1e45670d23e966"), "a" : 2 }
{ "_id" : ObjectId("516b51aa0f1e45670d23e969"), "a" : 4 }
{ "_id" : ObjectId("516b51aa0f1e45670d23e96a"), "a" : 5 }
{ "_id" : ObjectId("516b51aa0f1e45670d23e96c"), "a" : 7 }
{ "_id" : ObjectId("516b51aa0f1e45670d23e96d"), "a" : 8 }
{ "_id" : ObjectId("516b51aa0f1e45670d23e96f"), "a" : 10 }
```

Példa

```
> db.people.find({}, {_id: false})
{ "first" : "John", "last" : "Doe", "age" : 39, "interests" :
                [ "Reading", "Mountain Biking" ] }
{ "first" : "Jane", "last" : "Doe", "age" : 36, "interests" :
                [ "Shopping", "Programming", "Reading" ] }
{ "first" : "Mark", "last" : "Smith", "age" : 24, "interests" :
                [ "Skiing", "Playing Guitar" ], "homepage" : "http://example.com" }
> db.people.find({$or : [ {age: {$gt: 37}}, {homepage: {$exists: true}} ] }, {first: true})
{ "_id" : ObjectId("516b68cf0f1e45670d23ea63"), "first" : "John" }
{ "_id" : ObjectId("516b6a570f1e45670d23ea65"), "first" : "Mark" }
> db.people.find({interests: /S.*ing/}, {first: true})
{ "_id" : ObjectId("516b69160f1e45670d23ea64"), "first" : "Jane" }
{ "_id" : ObjectId("516b6a570f1e45670d23ea65"), "first" : "Mark" }
```

Egyéb lehetőségek

- **Eredmények számolása**:
  - `find(…).count()`
  - `count(…)`
- **Eredmények rendezése**:
  - `find(…).sort({`*`<mező`*$_1$*`>`*`: `*`<irány`*$_1$*`>`*`, …})`
  - Irány:
    - Pozitív érték → növekvő
    - Negatív érték → csökkenő
- **Eredmények formázása**: `find(…).pretty()`
- **Egyéb**: `find(…).limit(`*`<n>`*`)`, `find(…).skip(`*`<n>`*`)`

http://docs.mongodb.org/manual/reference/mongo-shell/#queries

Twitter adatok a feladatokhoz

- Host: `eszakigrid108.inf.elte.hu`
- Adatbázis: `twitter`
- Gyűjtemény: `messages`

- Adatok:
  - Beérkező tweetek
    - https://dev.twitter.com/docs/platform-objects/tweets
  - Törölt tweetek
    - https://dev.twitter.com/docs/streaming-apis/messages