

# CouchDB

BaCil. Vojta Hordějčuk

<http://voho.cz/>

Středisko UN\*Xových technologií

30. 11. 2010

**tohle na ní nezkoušejte. . .**

~~SELECT \* FROM girl WHERE (name = 'Dorothy') LIMIT 1~~



# Charakteristika

- ▶ dokumentově orientovaná databáze

# Charakteristika

- ▶ dokumentově orientovaná databáze
- ▶ The Apache Software Foundation

# Charakteristika

- ▶ dokumentově orientovaná databáze
- ▶ The Apache Software Foundation
- ▶ engine: Erlang

# Charakteristika

- ▶ dokumentově orientovaná databáze
- ▶ The Apache Software Foundation
- ▶ engine: Erlang
- ▶ data: JSON

# Charakteristika

- ▶ dokumentově orientovaná databáze
- ▶ The Apache Software Foundation
- ▶ engine: Erlang
- ▶ data: JSON
- ▶ pohledy: Javascript

# Charakteristika

- ▶ dokumentově orientovaná databáze
- ▶ The Apache Software Foundation
- ▶ engine: Erlang
- ▶ data: JSON
- ▶ pohledy: Javascript
- ▶ HTTP REST API



# Charakteristika

- ▶ žádné schéma → libovolná struktura

# Charakteristika

- ▶ žádné schéma → libovolná struktura
- ▶ plochý adresní prostor

# Charakteristika

- ▶ žádné schéma → libovolná struktura
- ▶ plochý adresní prostor
- ▶ žádné transakce

# Charakteristika

- ▶ žádné schéma → libovolná struktura
- ▶ plochý adresní prostor
- ▶ žádné transakce
- ▶ automatická detekce konfliktů

# Charakteristika

- ▶ žádné schéma → libovolná struktura
- ▶ plochý adresní prostor
- ▶ žádné transakce
- ▶ automatická detekce konfliktů
- ▶ snadná replikace

# Anticharakteristika

- ▶ relační databáze?

# Anticharakteristika

- ▶ relační databáze? **NE**

# Anticharakteristika

- ▶ relační databáze? **NE**
- ▶ náhrada za relační databáze?



# Anticharakteristika

- ▶ relační databáze? **NE**
- ▶ náhrada za relační databáze? **NE**

# Anticharakteristika

- ▶ relační databáze? **NE**
- ▶ náhrada za relační databáze? **NE**
- ▶ objektová databáze?

# Anticharakteristika

- ▶ relační databáze? **NE**
- ▶ náhrada za relační databáze? **NE**
- ▶ objektová databáze? **NE**

# Anticharakteristika

- ▶ relační databáze? **NE**
- ▶ náhrada za relační databáze? **NE**
- ▶ objektová databáze? **NE**
- ▶ náhrada za objektové databáze?

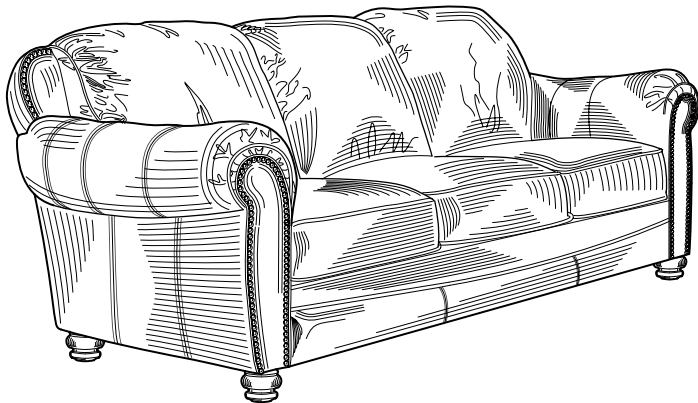
# Anticharakteristika

- ▶ relační databáze? **NE**
- ▶ náhrada za relační databáze? **NE**
- ▶ objektová databáze? **NE**
- ▶ náhrada za objektové databáze? **ANO**

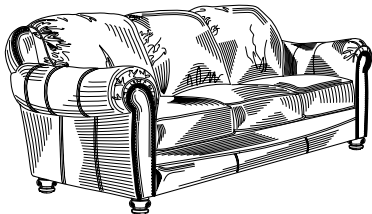
## Anticharakteristika

- ▶ relační databáze? **NE**
- ▶ náhrada za relační databáze? **NE**
- ▶ objektová databáze? **NE**
- ▶ náhrada za objektové databáze? **ANO** **NE**, dělám si srandu

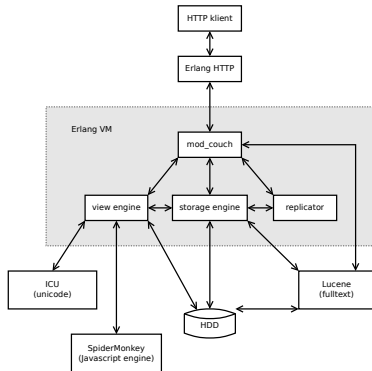
# Architektura



# Architektura

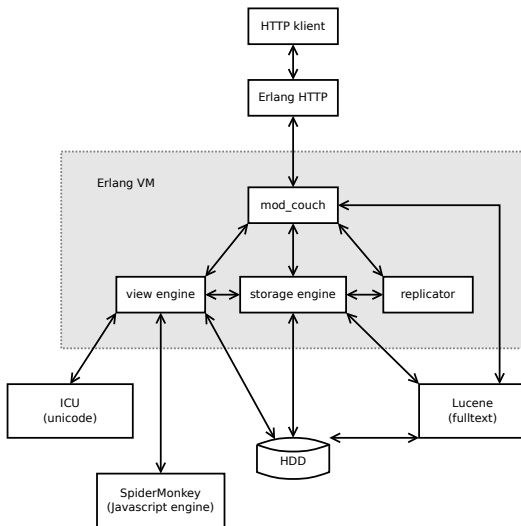


=



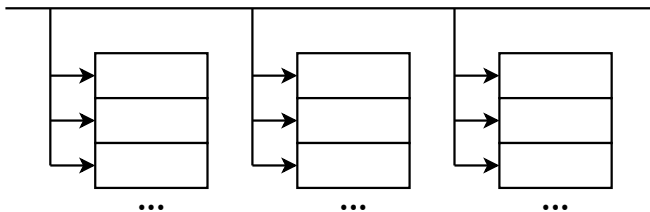


# Architektura



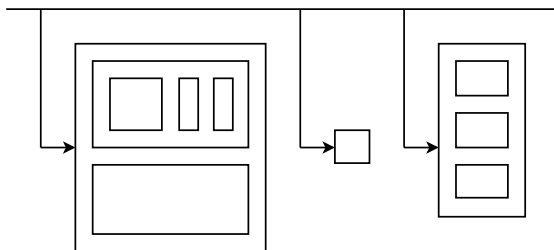
# Relační databáze

databáze – tabulka – řádek – atribut



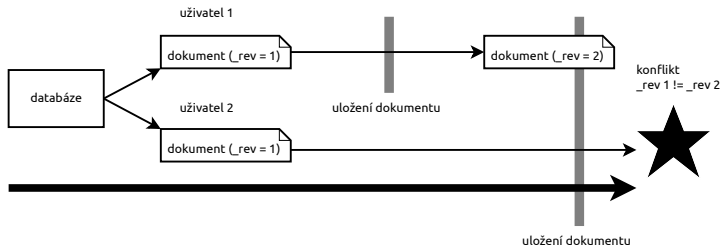
# Dokumentová databáze

databáze – dokument – hodnota – ... – hodnota



## Společné vlastnosti dokumentů

- ▶ **\_id** = identifikátor
- ▶ **\_rev** = číslo revize
- ▶ revize → ochrana proti konfliktům



# JSON

- ▶ JavaScript Object Notation
- ▶ JSON [džejsn]

objekt ::= { } | { páry }

páry ::= pár | pár , páry

pár ::= řetězec : hodnota

pole ::= [ ] | [ prvky ]

prvky ::= hodnota | hodnota , prvky

hodnota ::= řetězec | číslo | objekt | pole | true | false | null

## JSON – Příklad 1

```
{
  "název" : "Legrační video",
  "vložil" : "Jan Novotný",
  "datum" :
  {
    "den" : 10,
    "měsíc" : 3,
    "rok" : 2010
  },
  "komentáře" :
  [
    "Tak to je hodně dobrý!",
    "Fakt super video!",
    "Nuda, nic moc..."
  ]
}
```

## JSON – Příklad 2

```
{
  "jméno" : "Karel",
  "příjmení" : "Novák",
  "věk" : 30,
  "adresa" :
  {
    "ulice" : "Blanická 42",
    "město" : "Liptákov",
    "psč" : 12331,
    "země" : "Česká republika"
  },
  "zájmy" :
  [
    "zahrádka",
    "turistika",
    "fotografie",
    "chovatelství"
  ]
}
```

## Přílohy

- ▶ příloha = jakýkoliv soubor
- ▶ k dokumentu se přímo ukládá jen tzv. **stub**
- ▶ s přílohami se pracuje odděleně → nezatěžují zbytečně dokument
  - ▶ download – HTTP GET
  - ▶ upload – HTTP PUT

### Co je to “stub”?

Stub je záznam v dokumentu představující přílohu. Obsahuje informace o příloze - její název, velikost, MIME typ. . .



## Vložení dokumentu (HTTP požadavek)

```
PUT /example/some_doc_id HTTP/1.0
```

```
Content-Length: 29
```

```
Content-Type: application/json
```

```
{"greetings": "Hello, World!"}
```

## Vložení dokumentu (HTTP odpověď)

```
HTTP/1.0 201 Created
Server: CouchDB/0.9.0 (Erlang OTP/R12B)
Etag: "1-518824332"
Date: Wed, 24 Jun 2009 13:33:11 GMT
Content-Type: text/plain;charset=utf-8
Content-Length: 51
Cache-Control: must-revalidate
```

```
{"ok":true,"id":"some_doc_id","rev":"1-518824332"}
Connection closed by foreign host.
```

## Co je to vlastně dotaz?

dotaz ~ pohled + parametry

### Vyvolání

požadavek přes HTTP GET → HTTP odpověď (JSON)

### Definice

pohledy definované v databázi pomocí funkcí **map/reduce**

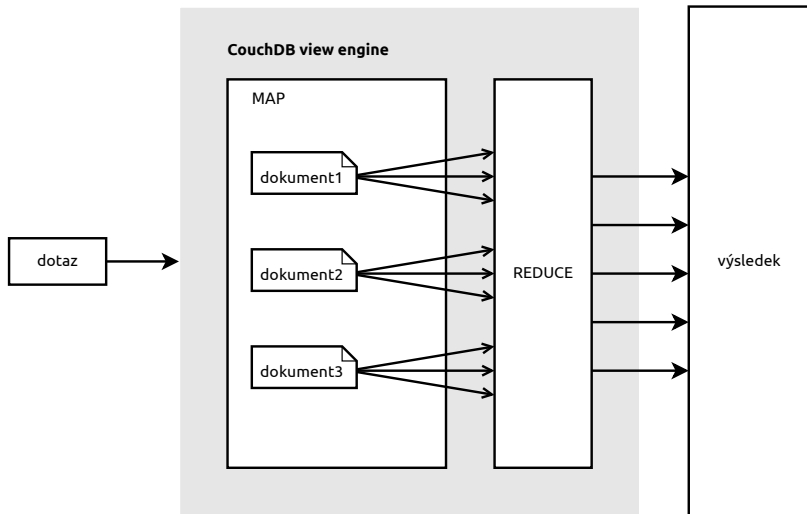
### Parametry

key, start key, end key, offset, limit, stale...

# Princip Map & Reduce

- ▶ Google
- ▶ JavaScript
- ▶ **map** = vytváří páry (klíč, hodnota)
- ▶ idempotence → inkrementální vyhodnocování
- ▶ **reduce** = redukuje páry
- ▶ výsledky seřazené podle klíče
- ▶  $\text{null} < 1 < \text{"ahoj"} < [] < \{ \}$

## Princip Map & Reduce



## SELECT \* FROM girl ORDER BY name

[http://localhost:5984/db/\\_design/girl/\\_view/list](http://localhost:5984/db/_design/girl/_view/list)

```
{
  "_id": "_design/girl",
  "_rev": "1-29b34c427276b2222b070c54494bbfd5",
  "language": "javascript",
  "views": {
    "list": {
      "map": "
        function(doc) {
          if (doc.name) {
            emit(doc.name, doc);
          }
        }"
    }
  }
}
```

# SELECT name FROM girl WHERE age BETWEEN 18 AND 25

[http://localhost:5984/db/\\_design/girl/\\_view/list?startkey=18&endkey=25](http://localhost:5984/db/_design/girl/_view/list?startkey=18&endkey=25)

```
{
  "_id": "_design/girl",
  "_rev": "1-29b34c427276b2222b070c54494bbfd5",
  "language": "javascript",
  "views": {
    "list": {
      "map": "
        function(doc) {
          if (doc.name) {
            emit(doc.age,doc.name);
          }
        }"
    }
  }
}
```

## SELECT age,COUNT(\*) FROM girl GROUP BY age

[http://localhost:5984/db/\\_design/girl/\\_view/count\\_age](http://localhost:5984/db/_design/girl/_view/count_age)

```
{
  "_id": "_design/girl",
  "_rev": "1-29b34c427276b2222b070c54494bbfd5",
  "language": "javascript",
  "views": {
    "count_age": {
      "map":
        "function(doc) {if (doc.name) {emit(doc.age,1);}}",
      "reduce":
        "function(key,values) {return sum(values);}"
    }
  }
}
```



## Výsledek předchozího pohledu

```
{
  "total_rows": 4, "offset": 0, "rows":
  [
    {"key":17, "value":3},
    {"key":18, "value":2},
    {"key":20, "value":1},
    {"key":22, "value":4}
  ]
}
```

# Srovnání

## Relační databáze

- ▶ integritní omezení
- ▶ mnoho funkcí
- ▶ transakce
- ▶ časem ověřené a vyspělé
- ▶ rozšířené a známé
- ▶ podporované

## Dokumentová databáze

- ▶ nehlídá ~ neomezuje
- ▶ základní funkce
- ▶ bez transakcí
- ▶ neznámé, spíše experiment
- ▶ rozšiřuje se
- ▶ knihy, wiki, IRC kanál

Každá technologie má své místo jinde, ale  $\cap \neq \emptyset$ .

# Knihovny, API, frameworky

## Java

jcouchdb, ektorp, JRelax, CouchDB4J, DroidCouch

## PHP

PHP-on-Couch

## C

libcouchdb + curl

## Haskell

(Hackage)

## LISP

Cl-Couch, Chillax

To je pro dnešek všechno :)

Děkuji za pozornost!  
Ptejte se.