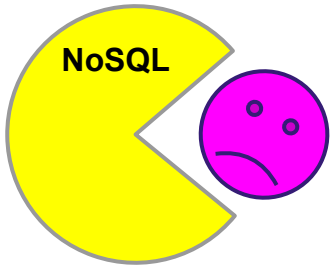
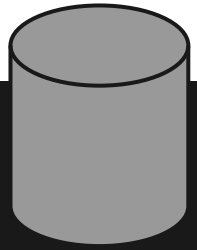


CouchDB

André Camargo
Lucas Pereira



O que garantiu a hegemonia dos bancos de dados relacionais?

- Persistência de dados;
- Concorrência;
- Padrão SQL;
- Integração entre aplicações;
- Relacionamento de dados.

Características do NoSQL

- Não há uma definição formal;
- Modelo não relacional;
- Sem esquema fixo;
- Sem interface SQL;
- Tendência para *clusters*;
- Normalmente *Open Source*.

Características do NoSQL

- Baixa latência;
- Tolerância a falhas;
- Escalabilidade;
- Alta disponibilidade;
- Esquemas flexíveis;
- Baixo custo;
- Busca com heurística.

Aplicações do NoSQL

- Alta quantidade de dados;
- Arquitetura distribuída;
- *Joins* não são suficientes;
- Esquema muda constantemente.

Prós e contras do NoSQL

Prós

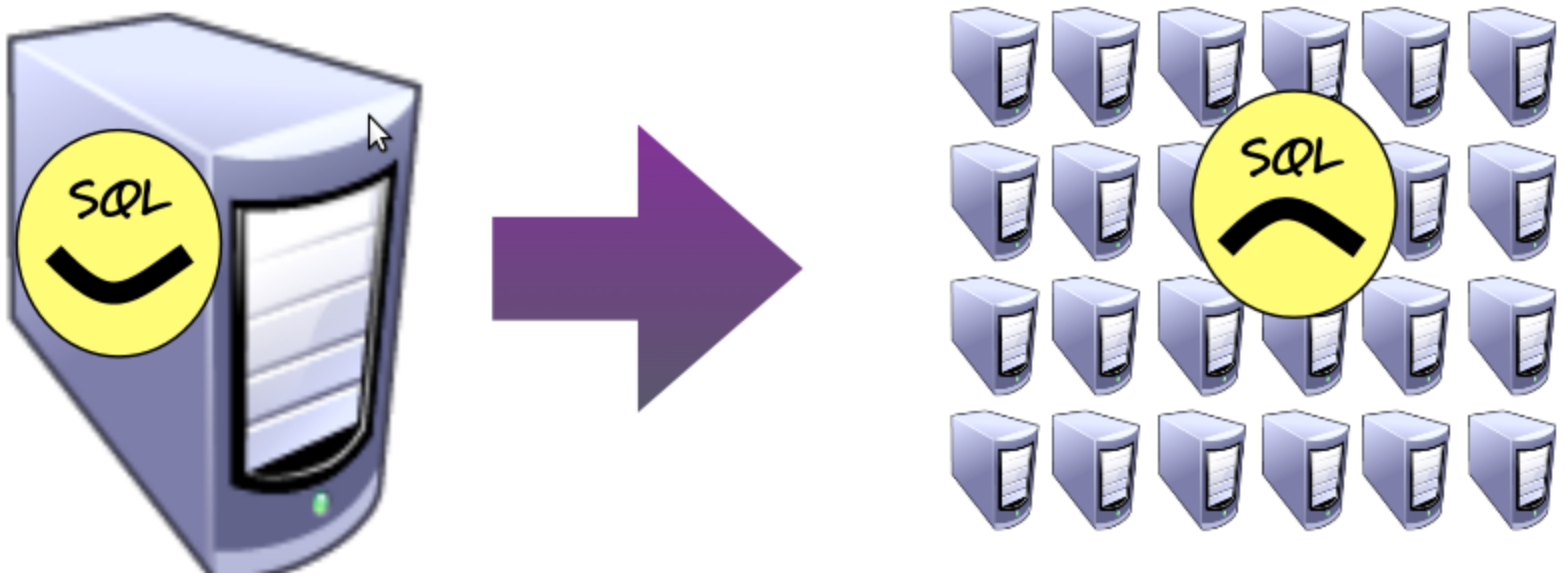
- Alta disponibilidade;
- Escalabilidade;
- Baixo custo;
- Esquemas flexíveis;
- Dados distribuídos.

Contras

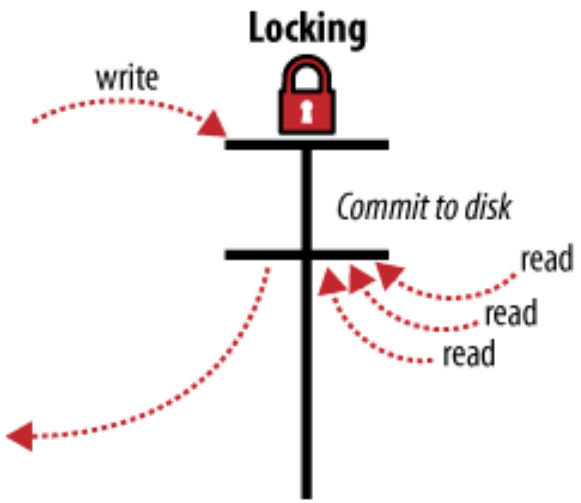
- Buscas dinâmicas limitadas;
- Dificuldade por consistência;
- Sem padrão;
- Sem controle de acesso;
- Busca textual não tão completa como SQL.

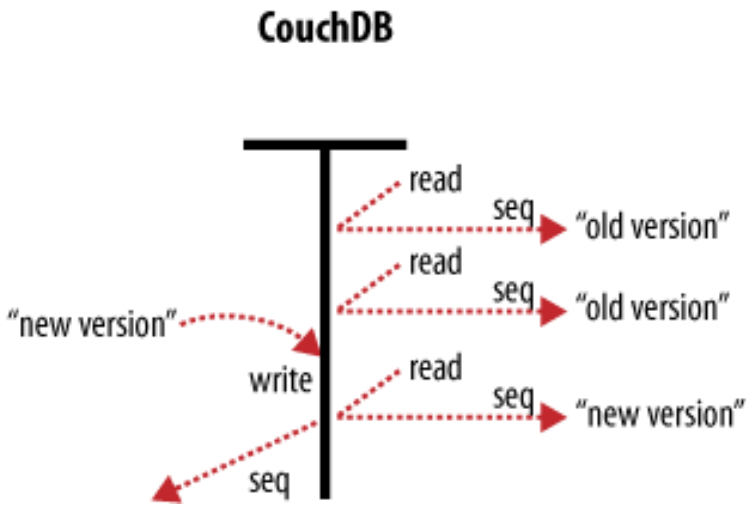
Escalabilidade do SQL

Vertical x Horizontal



Locks: SQL e NoSQL

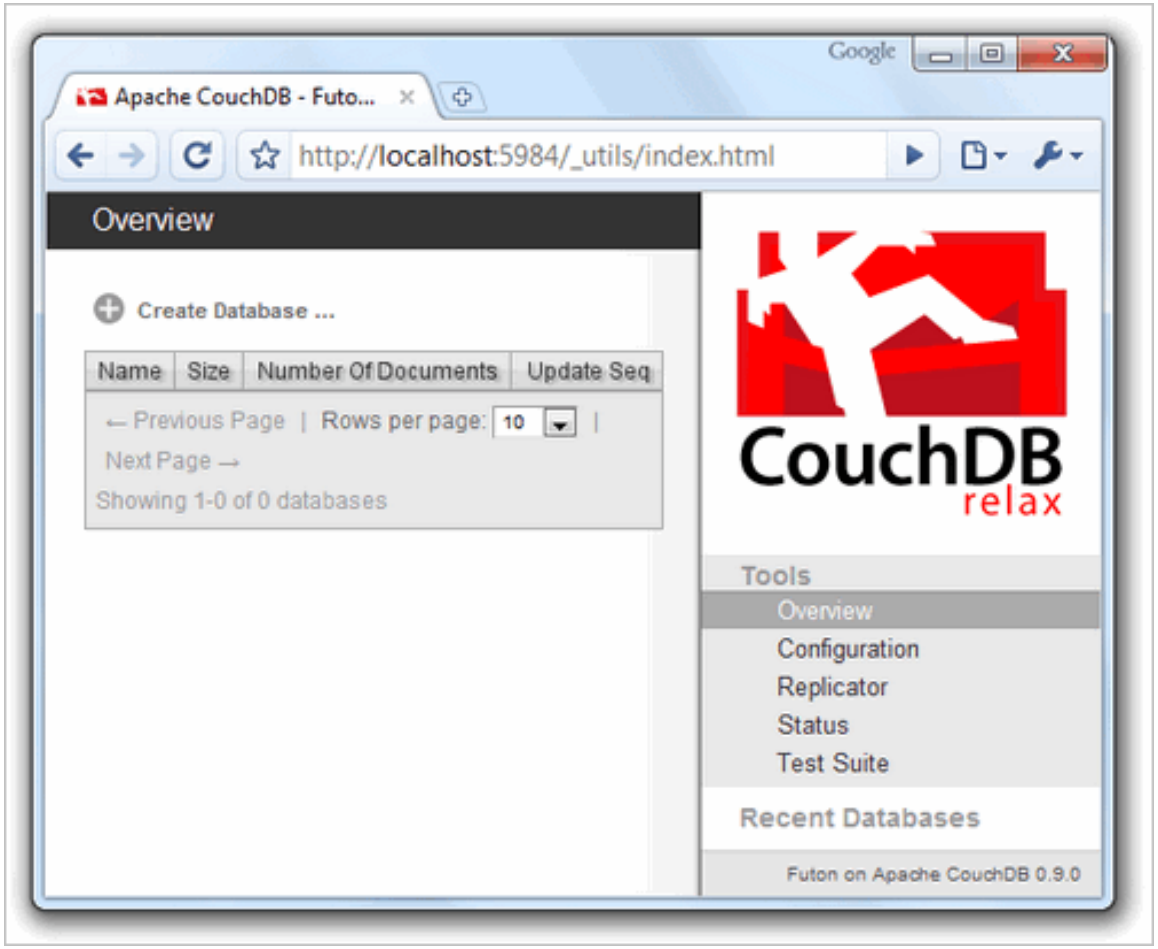




CouchDB

- Cluster of Unreliable Commodity Hardware (Conjunto de Hardware não confiáveis);
- Código aberto;
- Facilidade na Replicação de dados;
- Dados armazenados como uma coleção de documentos no formato JSON;
- MapReduces (agregação e filtro);
- API REST para manipulação dos dados.

Futon


Google
Apache CouchDB - Futo...
http://localhost:5984/_utils/index.html
Overview
Create Database ...
Name
Size
Number Of Documents
Update Seq
← Previous Page | Rows per page: 10 |
Next Page →
Showing 1-0 of 0 databases
CouchDB
relax
Tools
Overview
Configuration
Replicator
Status
Test Suite
Recent Databases
Futon on Apache CouchDB 0.9.0

Organização física

Estrutura: B+ *Tree*

- Excelente para armazenar grandes quantidades de dados, e tem acesso rápido;
- Garantem um tempo de acesso inferior à 10ms;
- CouchDB armazena as folhas em um dispositivo lento, como *hard disk*;
- *Append-only design*;
- Leituras simultâneas de versões diferentes de um mesmo documento.

API REST

- Utilização da Web;
- Arquitetura Orientada a Recursos;
- REST;
- Aplicações Web;
- GET, HEAD, POST, PUT, DELETE e COPY;
- URIs.

API REST

- /
- /{nomeDoBanco}
- /{nomeDoBanco}/{identificadorDoDocumento}
- /{nomeDoBanco}/{identificadorDoDocumento}/{nomeDoAnexo}
- /{nomeDoBanco}/_design/{nomeDoDesing}
- /{nomeDoBanco}/_design/{nomeDoDesing}/_view/{nomeDaView}
- /{nomeDoBanco}/_design/{nomeDoDesing}/_show/{nomeDaShowFunction}/{identificadorDoDocumento}
- /{nomeDoBanco}/_design/{nomeDoDesing}/_list/{nomeDaListFunction}/{nomeDaView}

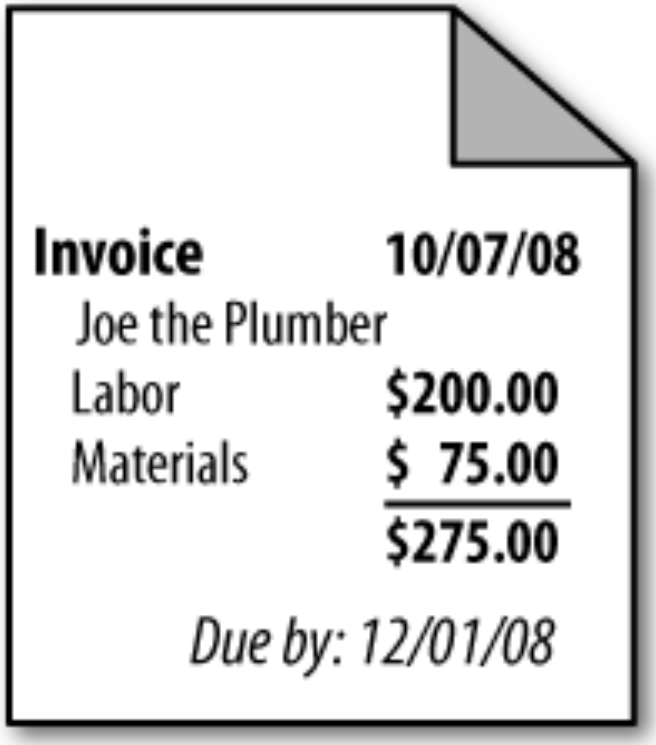
JSON

- Formato para representação de objetos;
- Leve e rápido (em comparação com XML);
- Pode ser analisado pela função `JSON.parse()`.

```
var dados = { "aluno" : [
{"nome":"João","provas":[{"nota":8},{"nota":6},{"nota":10},{"nota":2}]},
{"nome":"Maria","provas":[{"nota":3},{"nota":5},{"nota":8},{"nota":1}]},
{"nome":"Pedro","provas":[{"nota":7},{"nota":6},{"nota":6},{"nota":8}]}]};
 var dadosJson = JSON.parse(dados);
dadosJson.aluno[1].nome  // Maria
```

Documentos

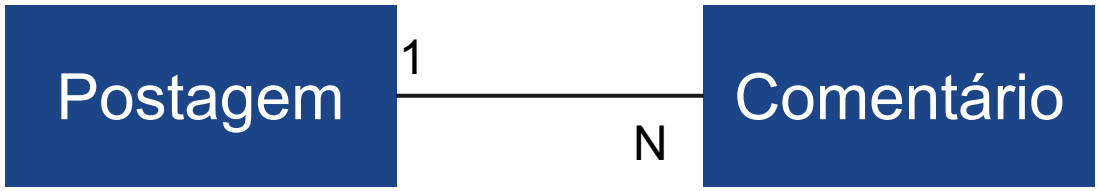
Real-world data is managed as real-world documents



Documentos

- Formato JSON;
- `_id`
- `_rev`
- `_attachments`
- `/{nomeDoBanco}/{identificadorDoDocumento}`
- `/{nomeDoBanco}/{identificadorDoDocumento}/{nomeDoAnexo}`
- Modelado de diversas formas.

Documentos



Documentos

Modelagem 1: postagem

```
{
   titulo: "Título da postagem",
   texto: "...",
   data: "11/09/2001",
   tipo: "postagem"
}
```

Documentos

Modelagem 1: comentário

```
{
   autor: "...",
   comentario: "...",
   identificadorDaPostagem:
      "33a483fea6e8d3fae8fb338d9b00179d",
   tipo: "comentario"
}
```

Documentos

Modelagem 2: postagem

```
{
    titulo: "Título da postagem",
    texto: "...",
    data: "11/09/2001",
    identificadoresDosComentarios: [
        "6beffd0c0c9bf04e72d0dcc029034f2c",
        "9eb51ac3d4df5eb1859ffb972488c7e4",
        "e7d760af60bd8ee418a81a8ed15dcd36"
    ],
    tipo: "postagem"
}
```

Documentos

Modelagem 2: comentário

```
{
   autor: "...",
   comentario: "...",
   tipo: "comentario"
}
```

Documentos

Modelagem 3: postagem e comentários

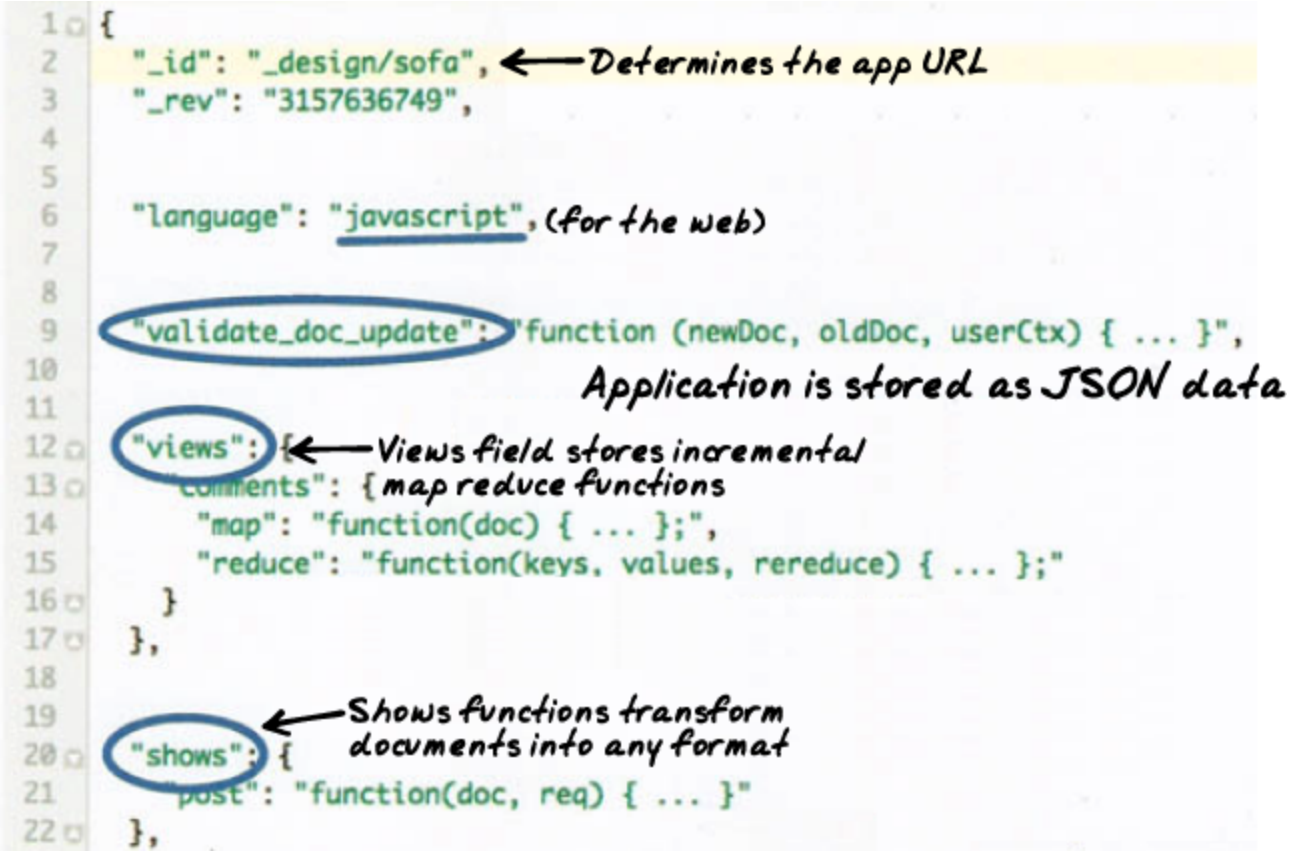
```
{
    titulo: "Título da postagem",
    texto: "...",
    data: "11/09/2001",
    comentarios: [
        {autor: "...", comentario: "..."},
        {autor: "...", comentario: "..."},
        {autor: "...", comentario: "..."}
    ],
    tipo: "postagem"
}
```

Design documents

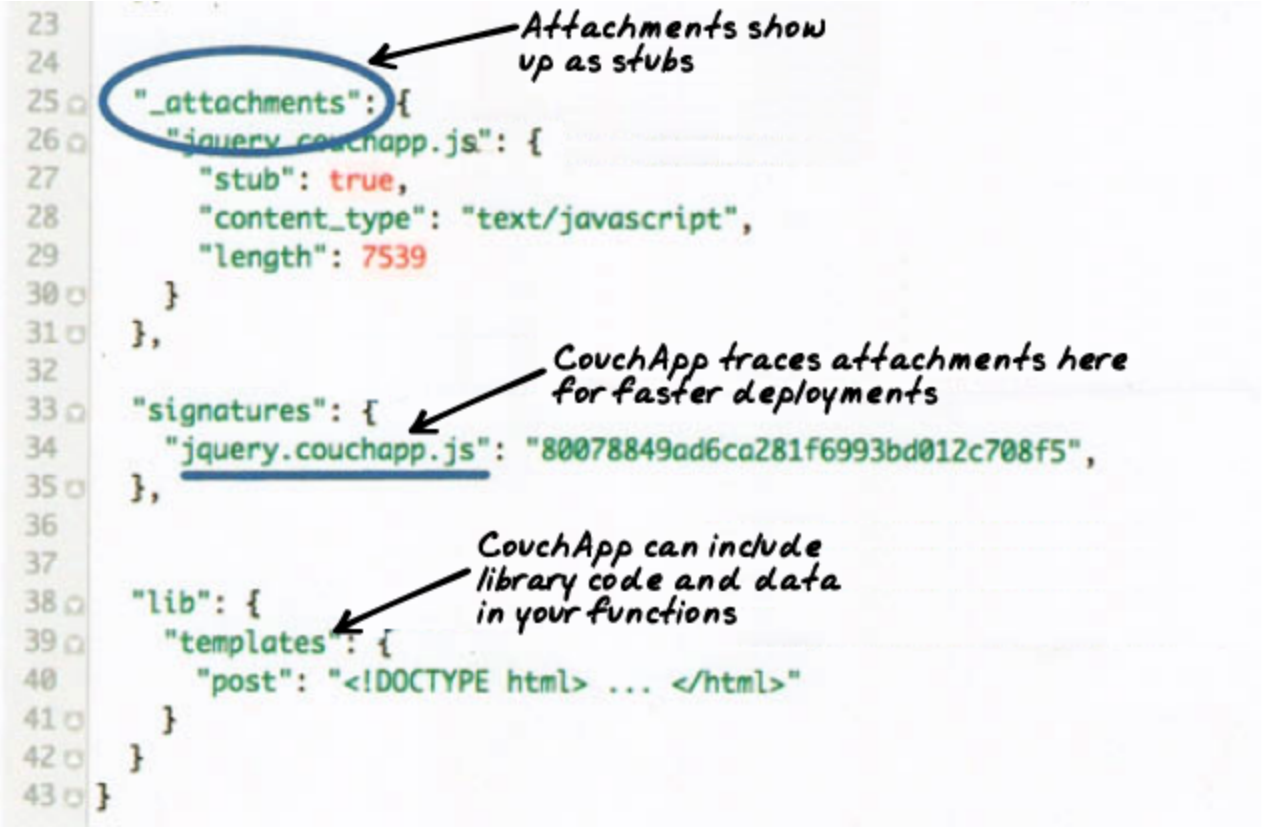
Design documents

- Código de aplicação;
- *Validate functions*, *show functions*, *list functions* e *views*;
- `/{nomeDoBanco}/{identificadorDoDocumento}`
- `/{nomeDoBanco}/_design/{nomeDoDesing}`
- São documentos;
- Múltiplos *design documents*.

Design documents



Design documents



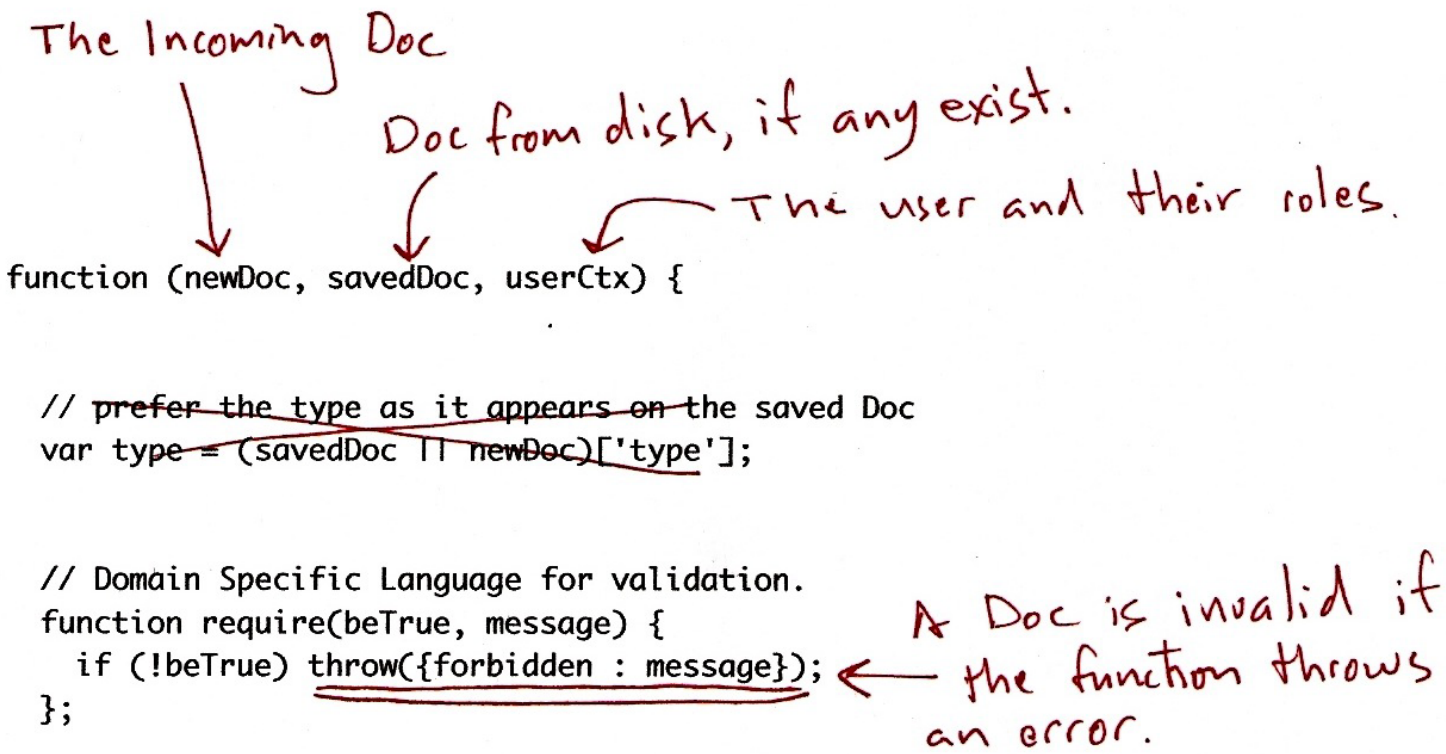

```
{
  "_attachments": {
    "jquery.couchapp.js": {
      "stub": true,
      "content_type": "text/javascript",
      "length": 7539
    }
  },

  "signatures": {
    "jquery.couchapp.js": "80078849ad6ca281f6993bd012c708f5",
  },

  "lib": {
    "templates": {
      "post": "<!DOCTYPE html> ... </html>"
    }
  }
}
```

Design documents

Validade function

- `validate_doc_update`
- Função JavaScript;
- Erro HTTP;
- Novo documento, documento antigo e contexto de usuário;
- `throw({forbidden: "Cai fora!"});`
- Apenas uma por *design document*.

Design documents



Design documents

```
// Ensure the timestamp hasn't changed.
require((savedDoc && savedDoc.created_at == newDoc.created_at),
        "You may not alter the created_at time.");


// Post-specific validations.
if (type == 'post') {
  validate([
    newDoc.author,                "Posts must have an author.",
    newDoc.title,                 "Posts must have a title.",
    newDoc.body,                  "Posts must have a body.",
    newDoc.format,                "Posts require a format.",
    newDoc.html,                  "Posts must have an html field.",
    newDoc.slug,                  "Posts must have a slug.",
    newDoc.slug == newDoc._id,    "Post slugs must be used as the _id.",
    newDoc.created_at,            "Posts must have a created_at date."
  ]);
}

}
```

If a user or application were to save a doc with a mismatched timestamp, ~~the function~~ CouchDB would return a 403 Forbidden HTTP status.

* We'll cover authorization later.

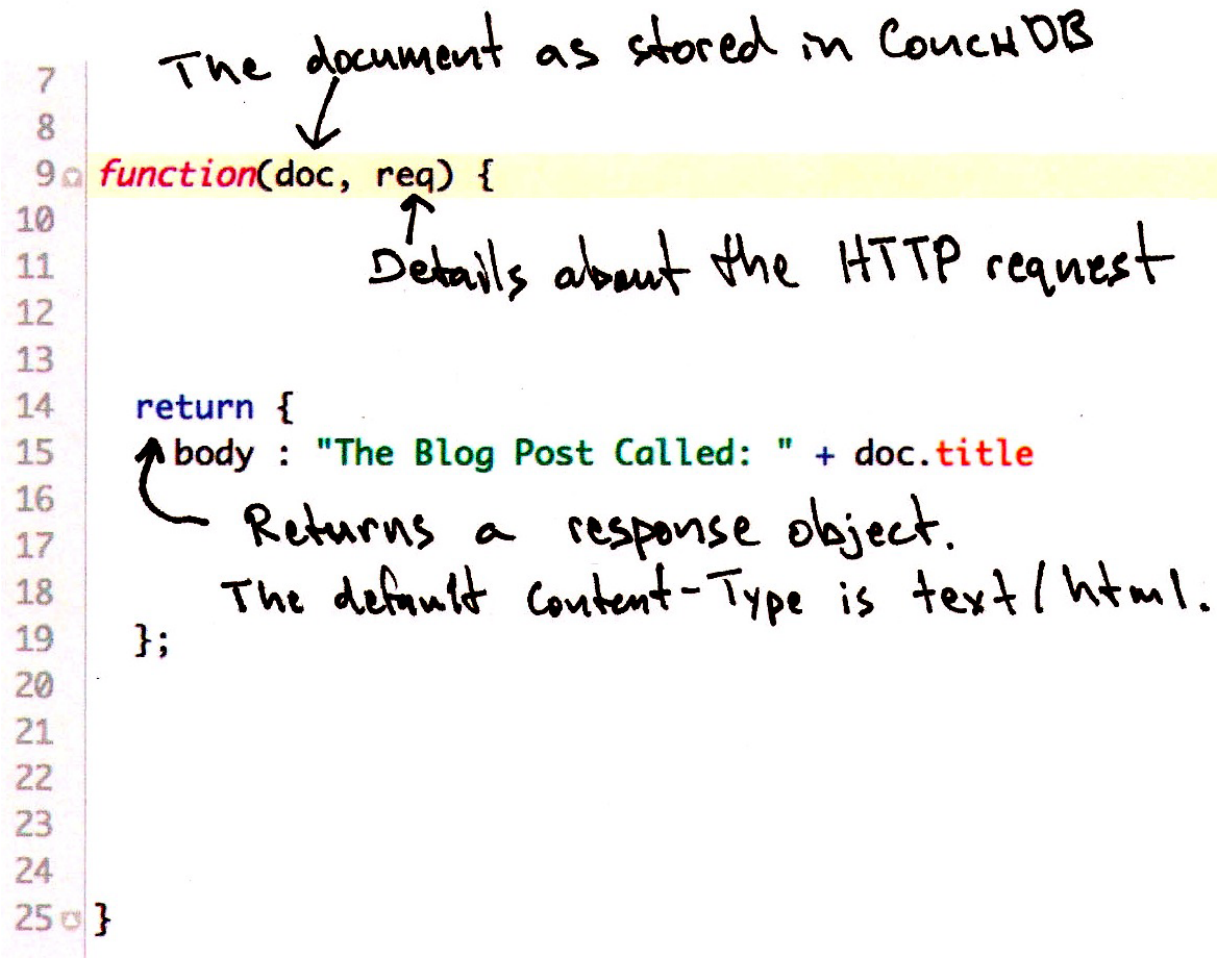
The messages sent to client browsers when something is invalid.

Design documents

Show function

- Documentos em outros formatos;
- Documento e detalhes da requisição;
- `/{nomeDoBanco}/_design/{nomeDoDesing}/_show/{nomeDaShowFunction}/{identificadorDoDocumento}`
- Várias por *design document*.

Design documents



Design documents

List function

- Semelhante a *show function*, porém aplicadas a views;
- `/{nomeDoBanco}/_design/{nomeDoDesing}/_list/{nomeDaListFunction}/{nomeDaView}`
- Várias por *design document*.

Design documents

Views

- Buscas;
- *MapReduce*;
- Map: filtragem e ordenação;
- Reduce: agregação.

Design documents

Views

- Executada uma vez para todos documentos;
- Executada ao atualizar um documento;
- Primeiro acesso demorado.
- `/{nomeDoBanco}/_design/{nomeDoDesing}/_view/{nomeDaView}`

Design documents

Map

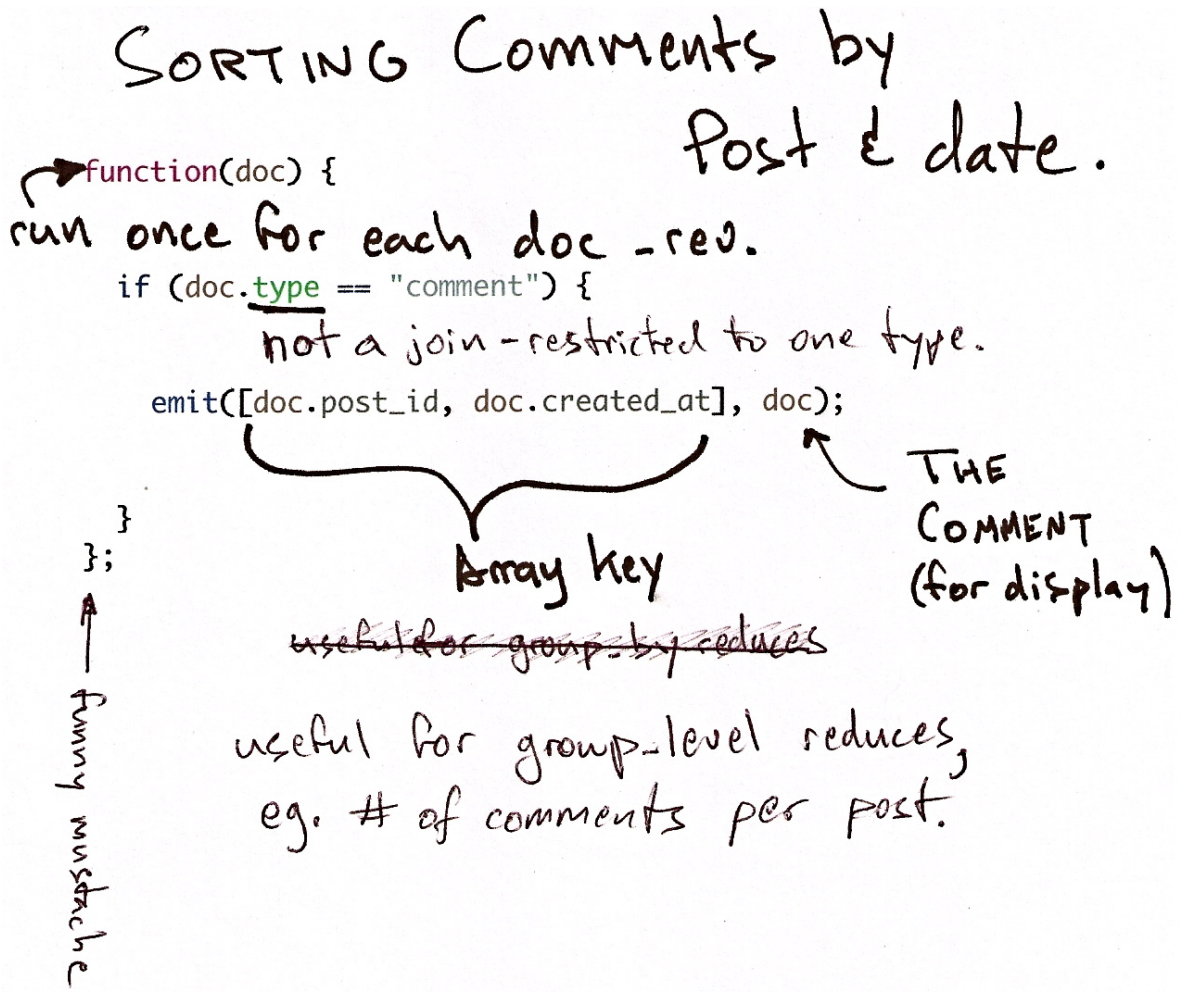
- Documento;
- `emit`
- Chave e valor;
- Chave complexa.

Design documents

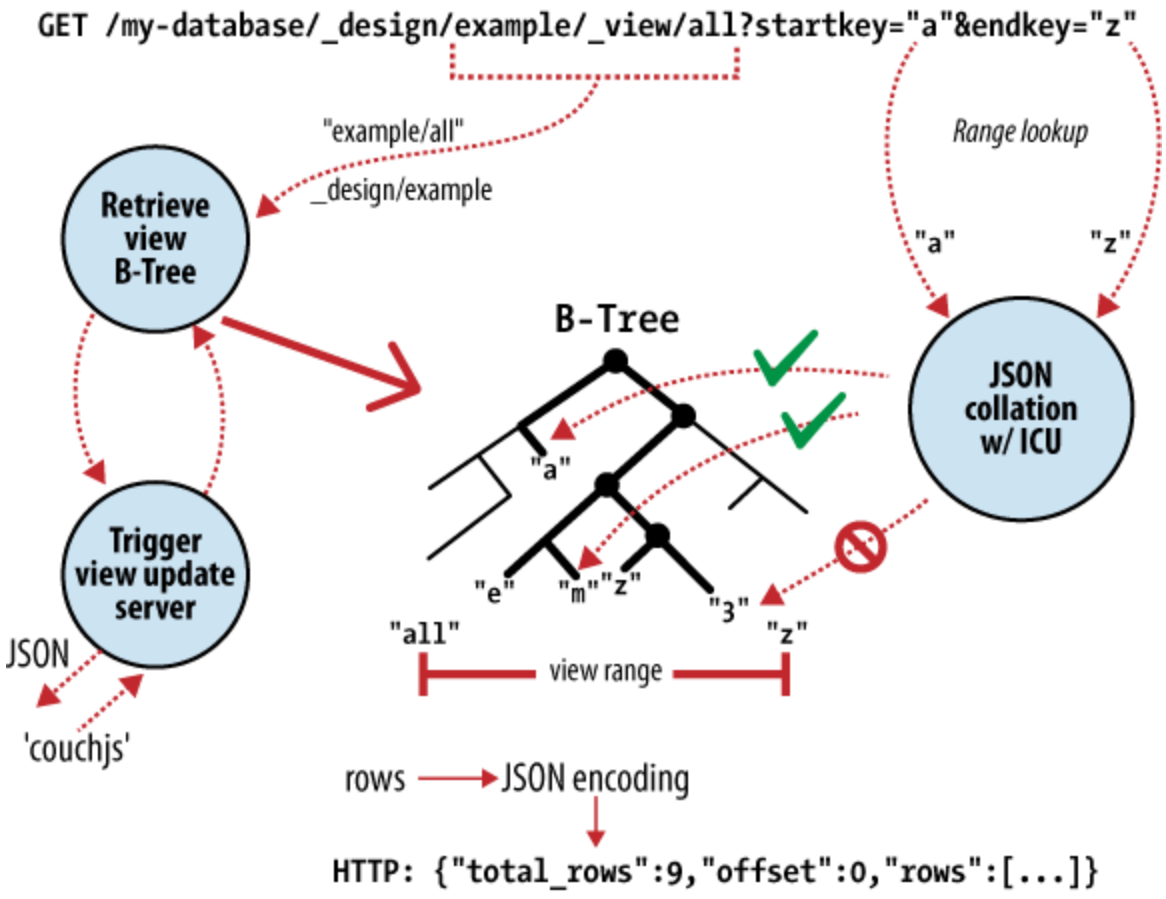
Map

- Seleção;
- Projeção;
- Múltiplos *emits*;
- Árvore B em arquivo separado.

Design documents



Design documents

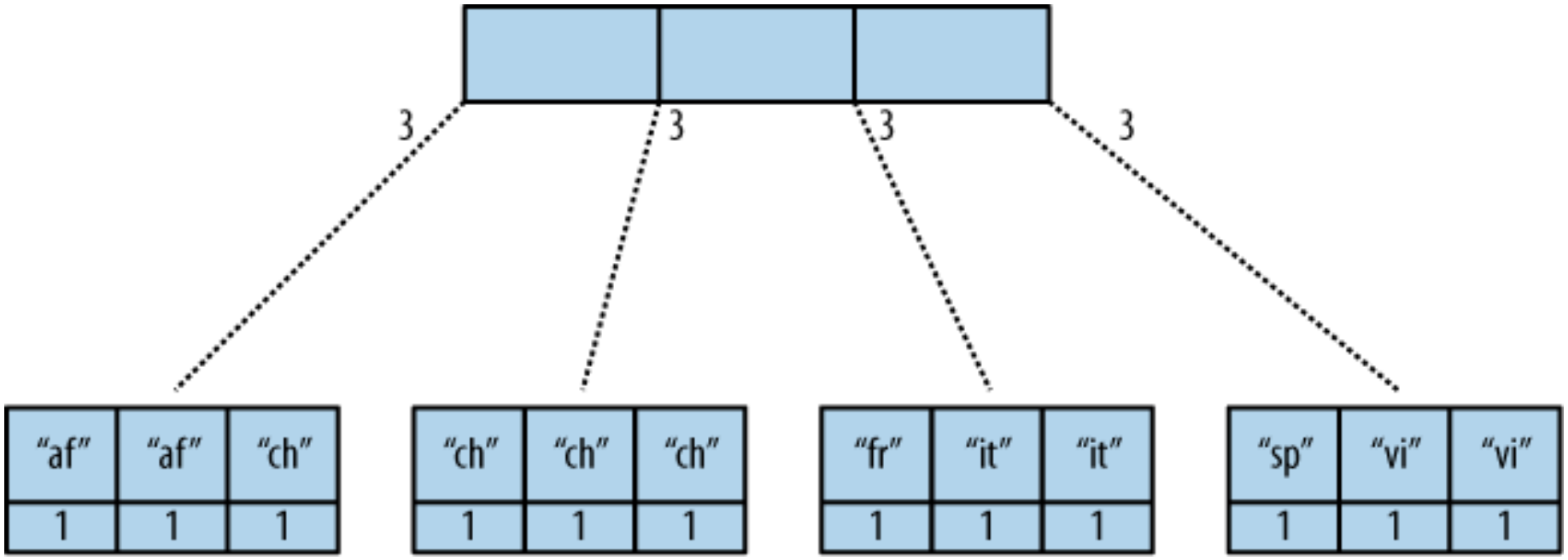


Design documents

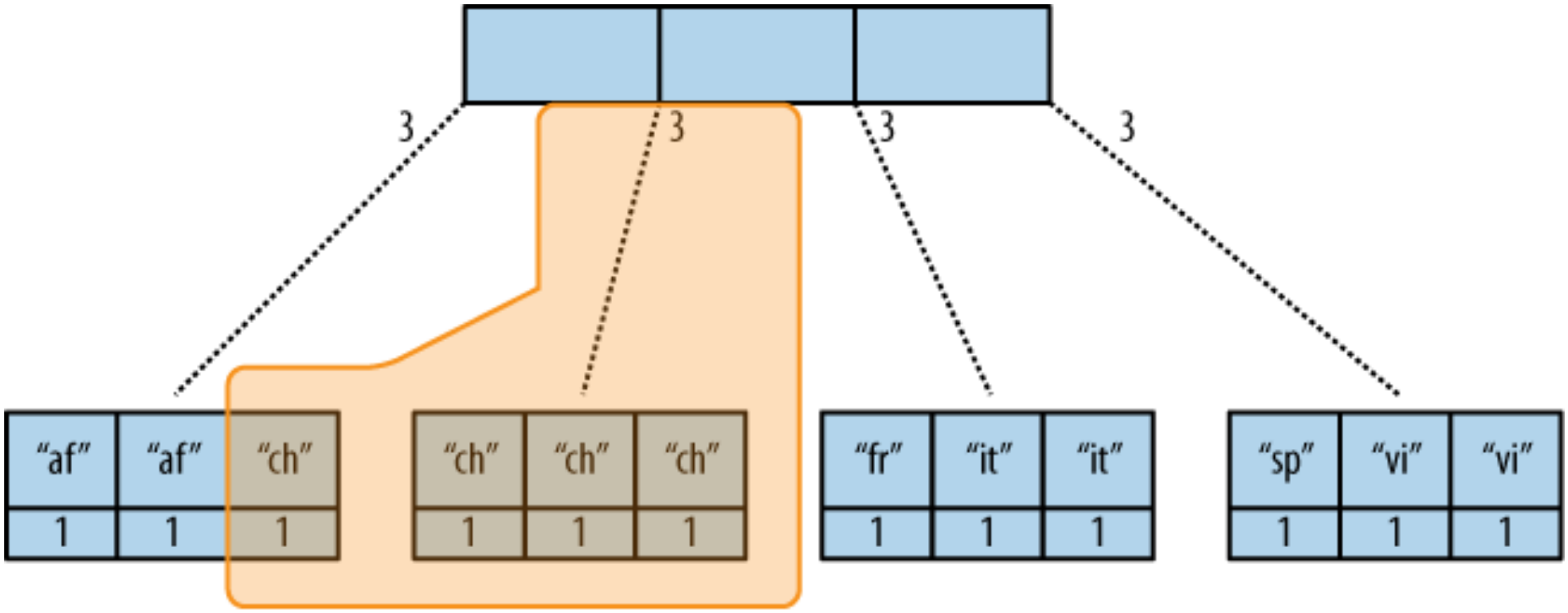
Reduce

- Contar registros, somar valores, achar o máximo e outros;
- Agrupamentos de nodos folhas da árvore B proveniente do *map* são aplicados ao *reduce*;
- `?group=true`
- `?group_level={nivel}`
- Chaves, valores e `rereduce = false`;
- Nulo, *reduces* anteriores e `rereduce = true`;
- Paralelismo.

Design documents



Design documents

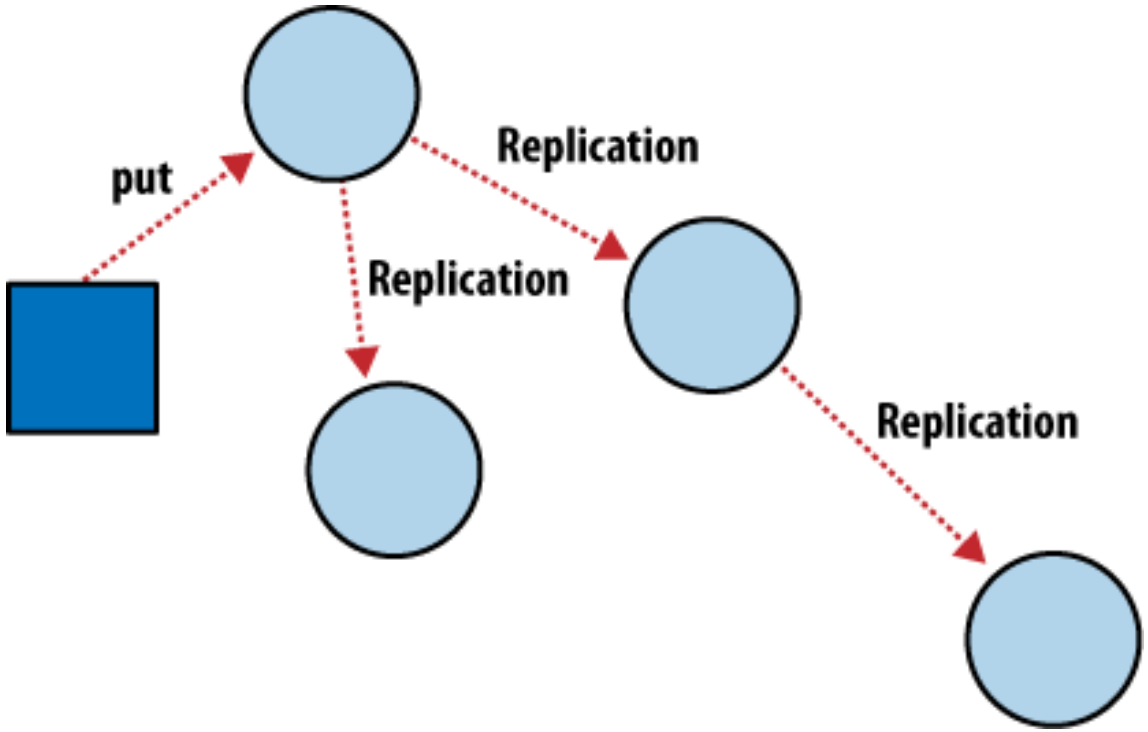


Design documents

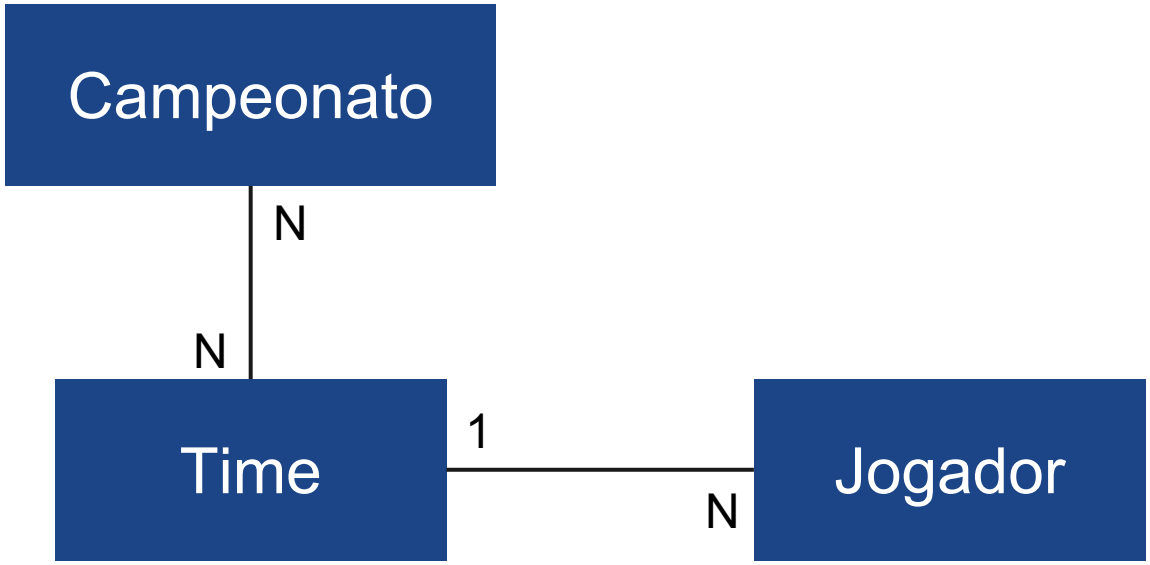
View

- `/{nomeDoBanco}/_design/{nomeDoDesing}/_view/{nomeDaView}`
- `emit`
- `?key={chave}`
- `?startkey={chave}` e `?endkey={chave}`
- `?descending=true`
- `?limit={limite}`

Replicação



Exemplo SQL x CouchDB



Exemplo SQL x CouchDB

Time

```
{
   "nome": "Avaí Futebol Clube",
   "dataDeFundacao": "1923-09-01",
   "estado": "SC",
   "capacidadeDoEstadio": 17537,
   "rankingNacional": 20,
   "tipo": "time"
}
```

Exemplo SQL x CouchDB

Exemplo 1

```
SELECT * FROM times
```

Exemplo SQL x CouchDB

```
SELECT * FROM times
```

```
function (documento) {
    if (documento.tipo === "time") {
        emit(documento._id, documento);
    }
}
```

```
/futebol/_design/futebol/_view/times
```

Exemplo SQL x CouchDB

Exemplo 2

```
SELECT nome, dataDeFundacao
FROM times
```

Exemplo SQL x CouchDB

```
SELECT nome, dataDeFundacao
FROM times
```

```
function (documento) {
    if (documento.tipo === "time") {
        emit(documento._id, [documento.nome, documento.dataDeFundacao]);
    }
}
```

/futebol/_design/futebol/_view/timesNomeDataDeFundacao

Exemplo SQL x CouchDB

Exemplo 3

```
SELECT *
FROM times
WHERE _id =
“bb50e9d6ff29c2f1331c6e6fa101ba8f”
```

```
/futebol/bb50e9d6ff29c2f1331c6e6fa101a727
```

```
/futebol/_design/futebol/_view/times?key="
bb50e9d6ff29c2f1331c6e6fa101a727"
```

Exemplo SQL x CouchDB

Exemplo 4

```
SELECT nome
FROM times
WHERE estado = “SC”
```

Exemplo SQL x CouchDB

```
SELECT nome
FROM times
WHERE estado = “SC”
```

```
function (documento) {
    if (documento.tipo === "time") {
        if (documento.estado === "SC") {
            emit(documento._id, [documento.nome]);
        }
    }
}
```

/futebol/_design/futebol/_view/timesDoEstadoDeSantaCatarina

Exemplo SQL x CouchDB

```
SELECT nome
FROM times
WHERE estado = “SC”
```

```
function (documento) {
    if (documento.tipo === "time") {
        emit(documento.estado, [documento.nome]);
    }
}
```

/futebol/_design/futebol/_view/timesDoEstadoDeSantaCatarinaComIndiceEstado?key="SC"

Exemplo SQL x CouchDB

Exemplo 5

```
SELECT nome
FROM times
WHERE dataDeFundacao BETWEEN 1910 AND 1920
```

Exemplo SQL x CouchDB

```
SELECT nome
FROM times
WHERE dataDeFundacao BETWEEN 1910 AND 1920
```

```
function (documento) {
    if (documento.tipo === "time") {
        emit(documento.dataDeFundacao, [documento.nome]);
    }
}
```

/futebol/_design/futebol/_view/timesComIndiceDataDeFundacao?startkey="1910-01-01"&endkey="1920-12-31"

Exemplo SQL x CouchDB

Exemplo 6

```
SELECT nome, capacidadeDoEstadio
FROM time
ORDER BY capacidadeDoEstadio
```

Exemplo SQL x CouchDB

```
SELECT nome, capacidadeDoEstadio
FROM time
ORDER BY capacidadeDoEstadio
```

```
function (documento) {
    if (documento.tipo === "time") {
        email(documento.capacidadeDoEstadio, [documento.nome, documento.capacidadeDoEstadio]);
    }
}
```

```
/futebol/_design/futebol/_view/timesComIndiceCapacidadeDoEstadio
```

Exemplo SQL x CouchDB

Exemplo 7

```
SELECT nome, capacidadeDoEstadio
FROM time
ORDER BY capacidadeDoEstadio DESC
```

Exemplo SQL x CouchDB

```
SELECT nome, capacidadeDoEstadio
FROM time
ORDER BY capacidadeDoEstadio DESC
```

```
function (documento) {
    if (documento.tipo === "time") {
        email(documento.capacidadeDoEstadio, [documento.nome, documento.capacidadeDoEstadio]);
    }
}
```

/futebol/_design/futebol/_view/timesComIndiceCapacidadeDoEstadio?descending=true

Exemplo SQL x CouchDB

Exemplo 8

```
SELECT t.nome, j.nome
FROM jogadores j JOIN times t
ON j.identificadorDoTime = b._id
```

Exemplo SQL x CouchDB

Jogador

```
{
   "nome": "Marcelo Toscano",
   "posicao": "Atacante",
   "identificadorDoTime":
      "bb50e9d6ff29c2f1331c6e6fa1020373",
   "tipo": "jogador"
}
```

Exemplo SQL x CouchDB

```
SELECT t.nome, j.nome
FROM jogadores j JOIN times t
ON j.identificadorDoTime = b._id
```

```
function (documento) {
    if (documento.tipo === "time") {
        emit([documento._id, 0], documento.nome);
    } else if (documento.tipo === "jogador") {
        emit([documento.identificadorDoTime, 1], documento.nome);
    }
}
```

/futebol/_design/futebol/_view/timesJuncaoComJogadores

Exemplo SQL x CouchDB

Exemplo 9

```
SELECT t.nome, j.nome
FROM jogadores j JOIN times t
ON j.identificadorDoTime = b._id
```

Exemplo SQL x CouchDB

Time

```
{
   "nome": "Fluminense Football Club",
   ...
   "identificadorDosJogadores": [
      "4c19e4e54f081740444a3095ee003dd3",
      "4c19e4e54f081740444a3095ee0049c9",
      "4c19e4e54f081740444a3095ee003450"
   ],
   "tipo": "time"
}
```

Exemplo SQL x CouchDB

```
SELECT t.nome, j.nome
FROM jogadores j JOIN times t
ON j.identificadorDoTime = b._id
```

```
function (documento) {
    if (documento.tipo === "time" && documento.identificadorDosJogadores) {
        var jogadores = documento.identificadorDosJogadores;
        for (var indice in jogadores) {
            emit([jogadores[indice], 0], documento.nome);
        }
    } else if (documento.tipo === "jogador") {
        emit([documento._id, 1], documento.nome);
    }
}
```

/futebol/_design/futebol/_view/timesJuncaoComJogadoresComRepeticao

Exemplo SQL x CouchDB

Exemplo 10

```
SELECT nome FROM times
WHERE nome LIKE “%Atlético%”
```

Exemplo SQL x CouchDB

```
SELECT nome FROM times
WHERE nome LIKE "%Atlético%"
```

```
function (documento) {
    if (documento.tipo === "time") {
        var nomes = documento.nome.split(" ");
        for (var indice in nomes) {
            emit(nomes[indice], documento.nome);
        }
    }
}
```

```
/futebol/_design/futebol/_view/timesComIndiceParteDoNome?key="Atlético"
```

Exemplo SQL x CouchDB

Exemplo 11

```
SELECT COUNT(*) FROM times
```

Exemplo SQL x CouchDB

```
SELECT COUNT(*) FROM times
```

```
function (documento) {
    if (documento.tipo === "time") {
        emit(documento._id, null);
    }
}
```

```
function (chaves, valores, rereduce) {
    if (rereduce) {
        return sum(valores);
    } else {
        return valores.length;
    }
}
```

```
/futebol/_design/futebol/_view/timesContagem
```

Exemplo SQL x CouchDB

Exemplo 12

```
SELECT COUNT(estado), estado
FROM times GROUP BY estado
```

Exemplo SQL x CouchDB

```
SELECT COUNT(estado), estado
FROM times GROUP BY estado
```

```
function (documento) {
    if (documento.tipo === "time") {
        emit(documento.estado, null);
    }
}
```

```
function (chaves, valores, rereduce) {
    return valores.length;
}
```

/futebol/_design/futebol/_view/timesContagemPorEstado?group=true

Exemplo SQL x CouchDB

Exemplo 13

```
SELECT MAX(capacidadeDoEstadio)
FROM times
```

Exemplo SQL x CouchDB

```
SELECT MAX(capacidadeDoEstadio)
FROM times
```

```
function (documento) {
    if (documento.tipo === "time") {
        emit(documento._id, documento.capacidadeDoEstadio);
    }
}
```

```
function (chaves, valores, rereduce) {
    var maximo = -Infinity;
    for (var indice in valores) {
        var valor = valores[indice];
        if (valor > maximo) {
            maximo = valor;
        }
    }
    return maximo;
}
```

/futebol/_design/futebol/_view/timesMaiorCapacidadeDoEstadio