NoSQL (Not only SQL)

Prof. Álisson R. Arantes

PUC Minas Virtual

Introdução

- Falsa impressão de que o modelo relacional é a única solução possível para o problema da persistência e obtenção de dados

- Entretanto, este modelo possui algumas limitações (“obsessão por representar qualquer informação sob a forma de linhas e colunas”)

Introdução

- Limitações do modelo relacional
  - nem todas as aplicações possuem dados altamente estruturados como no modelo relacional
  - as ferramentas relacionais (SGBDs) impõem controles que nem todas as aplicações precisam

Introdução

- Modelo relacional: foco em dados altamente estruturados e com controle de consistência

- Not Only SQL: complemento e não substituição ao modelo relacional

Introdução

- NoSQL:
  - foco em dados semiestruturados e não estruturados
  - alto desempenho e disponibilidade
  - sem exigência de esquemas rígidos
  - linguagens de consultas menos poderosas

Introdução

- Gama de SGBD’s que não seguem o modelo relacional:
  - modelo chave-valor
  - modelo orientado a documentos
  - modelo orientado a grafos
  - modelo família de colunas

Introdução

- Grandes organizações desenvolveram suas próprias soluções:
  - Google BigTable: modelo de colunas
  - Amazon DynamoDB: chave-valor
  - Apache Cassandra (inicialmente Facebook): híbrido chave-valor e modelo de colunas

Exemplos de Ferramentas NoSQL

- Modelo chave-valor:
  - Redis
  - DynamoDB

- Modelo de documentos:
  - MongoDB
  - CouchDB

Exemplos de Ferramentas NoSQL

- Modelos de grafos:
  - Neo4J
  - GraphBase

- Modelos de colunas:
  - BigTable
  - Hbase

Exemplos de Ferramentas NoSQL

- Híbridos:
  - Cassandra
  - OrientDB

Modelo Chave-Valor

- Foco no alto desempenho, disponibilidade e escalabilidade
- Armazena dados estruturados, semiestruturados e não estruturados
- Constitui-se de uma chave (identificador único) associada a um valor, o dado propriamente dito

Modelo Chave-Valor

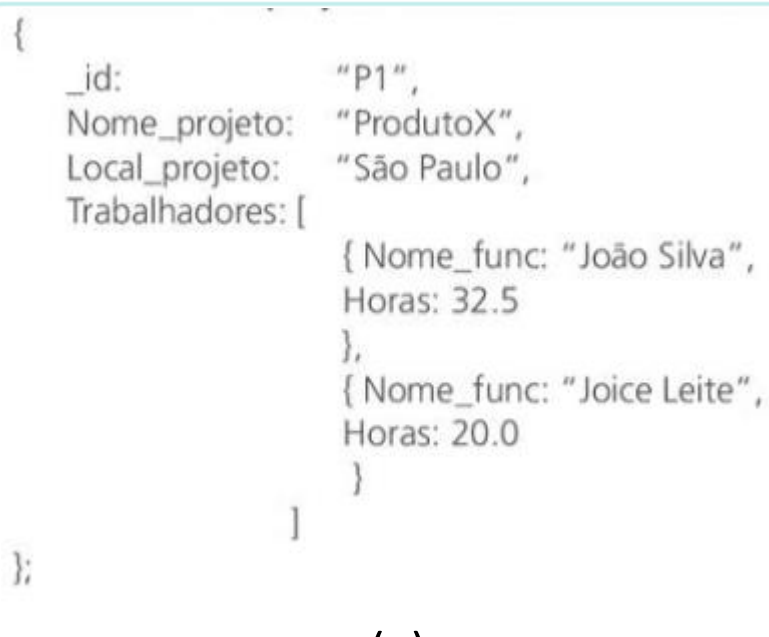
- Mecanismo utilizado para rápida recuperação dos dados
- Responsabilidade da aplicação interpretar a estrutura dos dados

Modelo Orientado a Documentos

- Armazenamento de coleções de documentos
- Um documento é um objeto complexo em formatos como XML ou JSON
- Estrutura dos dados autodescritiva

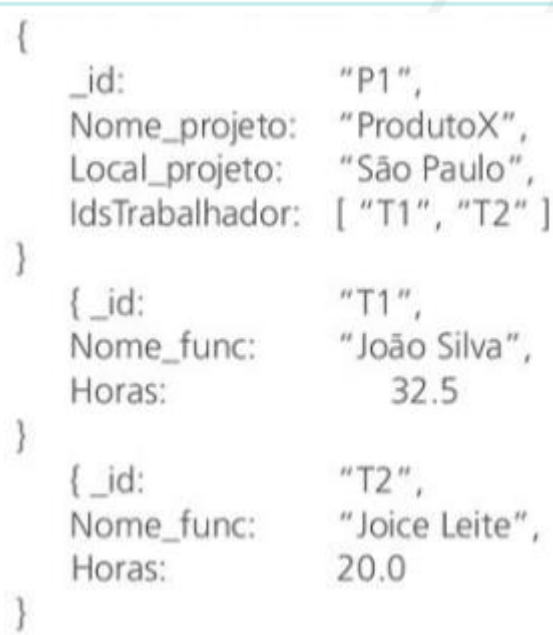
Modelo Orientado a Documentos

- Documentos podem ter elementos de dados diferentes



```
{
    _id:              "P1",
    Nome_projeto:     "ProdutoX",
    Local_projeto:    "São Paulo",
    Trabalhadores: [
                      { Nome_func: "João Silva",
                      Horas: 32.5
                      },
                      { Nome_func: "Joice Leite",
                      Horas: 20.0
                       }
                   ]
};
```

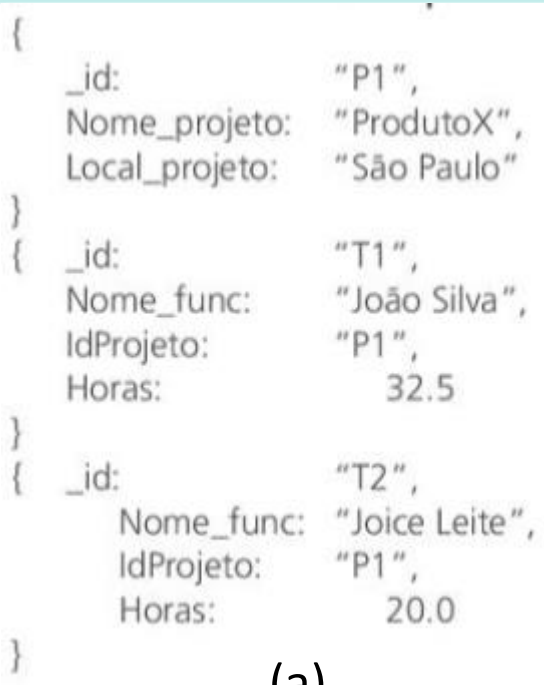
(a)



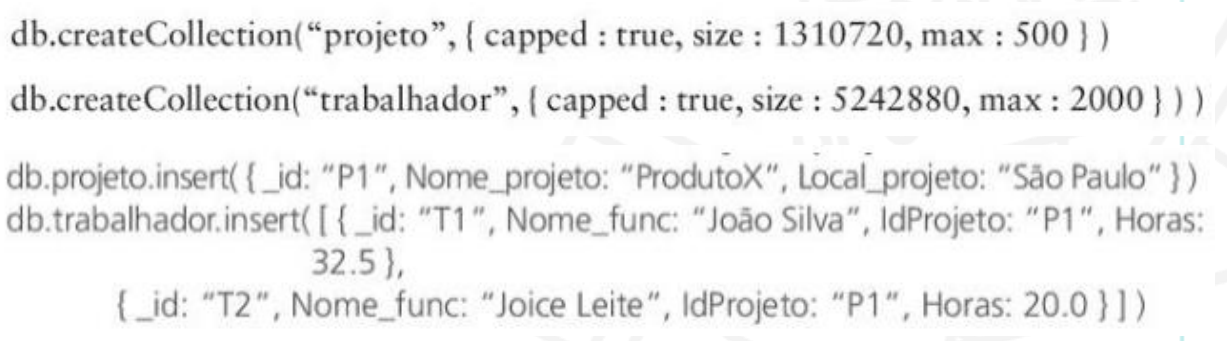
```
{
    _id:              "P1",
    Nome_projeto:     "ProdutoX",
    Local_projeto:    "São Paulo",
    IdsTrabalhador:   [ "T1", "T2" ]
}
    { _id:            "T1",
    Nome_func:        "João Silva",
    Horas:              32.5
}
    { _id:            "T2",
    Nome_func:        "Joice Leite",
    Horas:            20.0
}
```

(b)

(a) Documento de um projeto com subdocumentos embutidos.
(b) Vetor embutido de referências de documentos.
Fonte: Elmasri e Navathe (2019)



(a)



(b)

(a) Documentos normalizados.
(b) Inserindo os documentos de (a) em suas coleções "projeto" e "trabalhador" no MongoDB.
Fonte: Elmasri e Navathe (2019)

Modelo Orientado a Grafos

- Dados representados como um grafo
- Coleções de vértices (nós) e arestas (relacionamentos)
- Armazenamento de dados relativos aos vértices e às arestas

Modelo Orientado a Colunas

- Particionam uma tabela por colunas (particionamento vertical)
- Cada família de colunas armazenada separadamente

```
create 'FUNCIONARIO', 'Nome', 'Endereco', 'Detalhes'
```

(a)

```
put 'FUNCIONARIO', 'row1', 'Nome:Primeiro_nome', 'João'
put 'FUNCIONARIO', 'row1', 'Nome:Ultimo_nome', 'Silva'
put 'FUNCIONARIO', 'row1', 'Nome:Apelido', 'Jô'
put 'FUNCIONARIO', 'row1', 'Detalhes:Cargo', 'Engenheiro'
put 'FUNCIONARIO', 'row1', 'Detalhes:Avalia', 'Bom'
put 'FUNCIONARIO', 'row2', 'Nome:Primeiro_nome', 'Alice'
put 'FUNCIONARIO', 'row2', 'Nome:Ultimo_name', 'Zelaya'
put 'FUNCIONARIO', 'row2', 'Nome:Nome_meio, 'Jennifer'
put 'FUNCIONARIO', 'row2', 'Detalhes:Cargo', 'DBA'
put 'FUNCIONARIO', 'row2', 'Detalhes:Supervisor', 'Jorge Brito'
put 'FUNCIONARIO', 'row3', 'Nome:Primeiro_nome', 'Jorge'
put 'FUNCIONARIO', 'row3', 'Nome:Inicial_meio', 'E'
put 'FUNCIONARIO', 'row3', 'Nome:Ultimo_nome', 'Brito'
put 'FUNCIONARIO', 'row3', 'Nome:Sufixo', 'Jr.'
put 'FUNCIONARIO', 'row3', 'Detalhes:Cargo', 'CEO'
put 'FUNCIONARIO', 'row3', 'Detalhes:Salario', '1000000'
```

(b)

(a) Criando a tabela FUNCIONARIO com três famílias de colunas no Hbase.
(b) Inserindo na tabela FUNCIONARIO.
Fonte: Elmasri e Navathe (2019)

Conclusão

- NoSQL ou SQL: qual usar?
- NoSQL como complemento e não substituto das ferramentas relacionais
- Encontrar a melhor alternativa a cada uma das situações

PUC Minas
Virtual