Centro Paula Souza

Fatec Jacareí
Prof. Francisco de Moura

BANCO DE DADOS NÃO RELACIONAL

Operações CRUD no MongoDB

Professora: Lucineide Pimenta

Tópicos da Aula

Fatec
Jacareí
Prof. Francisco de Moura

- **Objetivo:**
  - Compreender como realizar operações CRUD (Create, Read, Update e Delete) no MongoDB, comparando com o PostgreSQL.
- **Tópicos:**
  - Coleções no MongoDB
  - Inserir documentos na coleção
  - Ler documentos da coleção
  - Operadores de igualdade
  - Operadores de comparação
  - Operadores de intervalo
  - Operador de expressão regular
- **Exercícios Práticos**

O comando Use

Fatec
Jacareí
Prof. Francisco de Moura

- Crie um banco de dados chamado *dbcursos*:

  Use dbcursos

- **O comando use é usado:**
  - para especificar o BD que queremos submeter os nossos comandos.
  - criará o BD se ele não existir.
  - O comando **use** não cria fisicamente o BD até que um **documento** seja inserido.

- Liste os bancos de dados:

  show databases

Coleções no MongoDB

Fatec
Jacareí
Prof. Francisco de Moura

- Para se criar um **documento** dentro do **BD**, primeiro cria-se uma **coleção** e automaticamente o documento é criado.
- Crie a coleção *alunos* no *bdcursos* com os campos: *nome*, *idade*, *genero*.

```
db.alunos.insertOne({
nome: "Gabriela",
idade: 21,
genero: "F"
})
```

db: faz referência ao bd atual (*dbcursos*)
alunos: é a coleção que vamos inserir no bd. Ela será criada se não existir.
insertOne: é o método que insere o documento na coleção *aluno*.

nome, idade, genero:
São os campos do documento que recebem os valores.

O campo **_id** serve como identificador único.
É gerado automaticamente a cada novo documento.
Um **_id** específico pode ser fornecido durante a inserção.
O **_id** é indexado para otimizar operações de busca e garantir a sua unicidade.

Coleções no MongoDB

Fatec
Jacareí
Prof. Francisco de Moura

- Liste as coleções do BD atual:

  Show tables

  Show collections

Inserir documentos na coleção

Fatec
Jacareí
Prof. Francisco de Moura

- **insertOne()** recebe **um objeto** JSON com os campos e valores do **documento** a ser inserido;
- **insertMany()** recebe **um array** onde **cada elemento** é um objeto JSON com os **campos e valores** do documento a ser inserido.

```
db.alunos.insertOne({
nome: “Vitória",
idade: 24,
genero: "F"
})
```

Inserir documentos na coleção

Fatec Jacareí
Prof. Francisco de Moura

- insertOne() recebe um objeto JSON com os campos e valores do documento a ser inserido;
- insertMany() recebe um array onde cada elemento é um objeto JSON com os campos e valores do documento a ser inserido.

```
db.alunos.insertOne({
nome: "Vitória",
idade: 24,
genero: "F"
})
```

```
db.alunos.insertMany([
{
nome: "Valdirene",
idade: 20,
genero: "F"
}
nome: "Tiago",
idade: 25,
genero: "M"
}
])
```

O método e o comando drop

Fatec
Jacareí
Prof. Francisco de Moura

- O método drop é usado para remover a coleção do BD:

db.alunos.drop()

show collections

O comando db.dropDatabase() remove o BD em uso:

db.drop.database()

show databases

Inserir documentos na coleção alunos

Fatec
Jacareí
Prof. Francisco de Moura

```
db.alunos.drop()
db.alunos.insertMany([
  { nome: "Pedro", idade: 25, genero: "M" },
  { nome: "Ana", idade: 20, genero: "F" },
  { nome: "Maria", idade: 21, genero: "F" },
 { nome: "Lucas", idade: 28, genero: "M" },
 { nome: "João", idade: 22, genero: "M" },
 { nome: "Renata", idade: 24, genero: "F" },
 { nome: "Paulo", idade: 23, genero: "M" },
 { nome: "Bruna", idade: 27, genero: "F" },
 { nome: "Irene", idade: 20, genero: "F" },
  { nome: "Roberto", idade: 21, genero: "M" },
  { nome: "Yuri",genero: "M" }
])
```

Ler documentos na coleção

Fatec
Jacareí
Prof. Francisco de Moura

- Liste **todos** os documentos da coleção *alunos*:

```
db.alunos.find()
```

- Em um SGBD relacional o método `find` estaria executando o comando SQL:

```
select * from alunos
```

Consulta com critérios de seleção

Fatec Jacareí
Prof. Francisco de Moura

- Liste todos os documentos da coleção *alunos* com **gênero** F e **idade** = 20:

```
db.alunos.find(
{
genero:"F",
idade:20
})
```

- O método find recebe um objeto JSON com os critérios de filtragem.

- Se fosse em um SGBD relacional o método find estaria executando o comando SQL:

```
select * from alunos
where genero = "F"
and idade = 20
```

Consulta com projeção de campos

Fatec
Jacareí
Prof. Francisco de Moura

- Liste apenas os **nomes** de todos os documentos da coleção *alunos*:

```
db.alunos.find(
{},
{
nome:true, _id:false}
)
```

- O método find recebe um objeto JSON com os critérios de filtragem.

- Se fosse em um SGBD relacional o método find estaria executando o comando SQL:

select nome from alunos;

Consulta com opções

Fatec
Jacareí
Prof. Francisco de Moura

- Liste os *alunos* ordenados por **idade**, **ignorando** os 2 primeiros documentos e **mostrando** os 4 últimos:

```
db.alunos.find(
{},
{
  nome:true,
  idade:true,
  _id:false
},
{
  sort:{ idade:1 },
  skip: 2,
  limit: 4
})
```

- Se fosse em um SGBD relacional o exemplo estaria executando o seguinte comando SQL:

```
select nome, idade
from alunos
order by idade asc
limit 4
offset 2;
```

sort: *especifica a ordenação dos documentos retornados (1 para ascendente e -1 para descendente).*
skip: *pula um número especificado de documentos antes de começar a retornar os resultados.*
limit: *limita o número de documentos retornados pela consulta ao valor especificado.*

Consulta básica findOne()

Fatec
Jacareí
Prof. Francisco de Moura

- Liste o **primeiro** documento da coleção *alunos*:

```
db.alunos.findOne()
```

- Se fosse em um SGBD relacional o exemplo seria equivalente ao comando SQL:

```
select * from alunos
limit 1;
```

Consulta com critérios de seleção e projeção de campos

Fatec Jacareí
Prof. Francisco de Moura

- Retorne o **1º documento** da coleção que tenha o **gênero** F e **idade** = 20:

```
db.alunos.findOne({
    genero:"F",
    idade:20
},
{
    nome:true,
    _id:false
})
```

- Se fosse em um SGBD relacional o exemplo seria equivalente ao comando SQL:

```
select nome
from alunos
where genero = "F" and
idade = 20
limit 1;
```

Operadores

Fatec Jacareí
Prof. Francisco de Moura

- Os operadores de comparação são usados para realizar consultas mais específicas e filtrar documentos com base em critérios de comparação.
- São usados com os métodos find e findOne().
- **Operador de igualdade:**

  $eq: comparador “igual a”.

- Liste os documentos que possuem idade = 20 e gênero = F:

```
db.alunos.find(
{
idade: {$eq:20},
genero: {$eq:"F"}
},
{
nome:true,
idade:true,
_id:false
}
)
```

Este comando é equivalente ao comando SQL:

select nome, idade from alunos where idade = 20 and genero = "F";

Operador: $ne

Fatec
Jacareí
Prof. Francisco de Moura

- Liste os documentos que **não** possuem **idade** = 20 e **gênero** =F:

```
db.alunos.find(
{
idade: {$ne:20},
genero: {$eq:"F"}
},
{
nome:true,
idade:true,
_id:false
}
)
```

$ne: comparador "**diferente de**".

Este comando é equivalente ao comando SQL:

select nome, idade from alunos
where idade != 20 and genero = "F";

Operadores de comparação: $gt, $gte, $lt, $lte

Fatec Jacareí
Prof. Francisco de Moura

- Liste os documentos de *alunos* que possuem **idade** entre 22 e 24 anos:

```
db.alunos.find(
{
idade: {$gte:22, $lte:24}
},
{
nome:true,
idade:true,
_id:false
}
)
```

- Este comando é equivalente ao comando SQL:

```
select nome, idade
from alunos
where idade >= 22 and
idade <= 24;
```

$gt, $gte, $lt, $lte:
comparadores "maior que", "maior ou igual a", "menor que" e "menor ou igual a", respectivamente.

Operador de intervalo: $in

Fatec Jacareí
Prof. Francisco de Moura

- Retorne os documentos quando o **nome** for: Maria, João e Bruna:

```
db.alunos.find(
{
nome:{$in:["Maria","João","Bruna"]}
},
{
nome:true,
idade:true,
_id:false
}
)
```

$in: corresponde a **qualquer** um dos valores especificados em um array.

Este comando é equivalente ao comando SQL:

select nome, idade
from alunos
where nome in ('Maria', 'João', 'Bruna');

Operador de intervalo: $nin

Fatec
Jacareí
Prof. Francisco de Moura

- Liste os documentos que **não** possuem o nome de alunos: Maria, João e Bruna:

```
db.alunos.find(
{
nome:{$nin:["Maria","João","Bruna"]}
},
{
nome:true,
idade:true,
_id:false
}
)
```

$nin: não corresponde a **nenhum** dos valores especificados em um array.

Este comando é equivalente ao comando SQL:

select nome, idade from alunos where nome not in ('Maria', 'João', 'Bruna');

Operador de expressão regular: $regex

Fatec Jacareí
Prof. Francisco de Moura

- Liste os documentos que possuem a letra "**o**" em **qualquer** parte do campo nome:

```
db.alunos.find(
{
nome:{ $regex: /o/i }
},
{
nome:true,
idade:true,
_id:false
}
)
```

$regex: seleciona os documentos cujos valores correspondem a expressão regular especificada.

O i indica que a expressão é **insensível** ao case da letra.

O que é CRUD?

Fatec
Jacareí
Prof. Francisco de Moura

- CRUD - As Quatro Operações Básicas de um Banco de Dados:

Operação	PostgreSQL (SQL)	MongoDB (NoSQL)
Create	`INSERT INTO tabela (...) VALUES (...)`	`db.colecao.insertOne({ ... })`
Read	`SELECT * FROM tabela`	`db.colecao.find()`
Update	`UPDATE tabela SET ... WHERE ...`	`db.colecao.updateOne({ ... })`
Delete	`DELETE FROM tabela WHERE ...`	`db.colecao.deleteOne({ ... })`

- **PostgreSQL:** Usa comandos SQL estruturados.
- **MongoDB:** Utiliza JSON e métodos de manipulação de documentos.

Criando Registros (CREATE)

Fatec
Jacareí
Prof. Francisco de Moura

- **Inserção de Dados**
  - **PostgreSQL (SQL - Tabelas Estruturadas)**

    INSERT INTO clientes (nome, idade, cidade)
    VALUES ('Ana', 30, 'São Paulo');
  - **MongoDB (NoSQL - Documentos JSON)**

```
db.clientes.insertOne({
    "nome": "Ana",
    "idade": 30,
    "cidade": "São Paulo"
});
```

Observação:
- *PostgreSQL exige definição de colunas e tipos de dados.*
- *MongoDB permite flexibilidade na estrutura dos documentos.*

Lendo Registros (READ)

Fatec
Jacareí
Prof. Francisco de Moura

- **Consultando Dados**
- **PostgreSQL (Consulta SQL tradicional)**

  *SELECT * FROM clientes WHERE cidade = 'São Paulo';*
- **MongoDB (Consulta JSON-like)**

  db.clientes.find({ "cidade": "São Paulo" });
- Exemplo - Buscar clientes com idade maior que 25:

  db.clientes.find({ "idade": { $gt: 25 } });

Observação:
•O find() do MongoDB retorna documentos completos no formato JSON.
•É possível usar operadores como $gt, $lt, $regex.

Atualizando Registros (UPDATE)

Fatec Jacareí
Prof. Francisco de Moura

- **Alteração de Dados**

  MongoDB (Atualização com $set)

  db.clientes.updateOne(
  { "nome": "Ana" },
  { $set: { "idade": 31 } }
  );

- Para atualizar múltiplos documentos:

  db.clientes.updateMany(
  { "cidade": "São Paulo" },
  { $set: { "status": "Ativo" } }
  );

- **Alteração de Dados**

  PostgreSQL (Modificação de Dados com SQL)

  UPDATE clientes SET idade = 31 WHERE nome = 'Ana';

Observação:
*•MongoDB usa o operador $set para atualizar **somente** os campos desejados.*
*O $updateOne() altera **apenas** o primeiro documento encontrado.*

Excluindo Registros (DELETE)

Fatec
Jacareí
Prof. Francisco de Moura

- **Removendo Dados**
- **PostgreSQL (Removendo Registros com SQL)**

  DELETE FROM clientes WHERE nome = 'Ana';
- **MongoDB (Excluindo Documentos com JSON)**

  db.clientes.deleteOne({ "nome": "Ana" });
- **Exemplo - Remover todos os clientes inativos:**

  db.clientes.deleteMany({ "status": "Inativo" });

Observação:
*O deleteOne() remove **apenas** o primeiro documento encontrado.*
*Para remover **múltiplos** registros, use deleteMany().*

Atividade Prática

Fatec
Jacareí
Prof. Francisco de Moura

- **Atividade prática no MongoDB**
- Criar a coleção *pedidos*.
- Inserir registros de pedidos para diferentes clientes.
- Consultar pedidos por cliente e por valor mínimo.
- Atualizar um pedido já inserido.
- Excluir um pedido específico.

Referências Bibliográficas

- **Material de apoio:**

- Chodorow, Kristina. *MongoDB: The Definitive Guide*. O'Reilly Media, 2013.
- *PostgreSQL Documentation*. Disponível em: https://www.postgresql.org/docs/
- *MongoDB Documentation*. Disponível em: https://www.mongodb.com/docs/
- Cattell, Rick. *Scalable SQL and NoSQL Data Stores*. ACM, 2011.

Bibliografia Básica

Fatec
Jacareí
Prof. Francisco de Moura

- BOAGLIO, Fernando**. MongoDB:** Construa novas aplicações com novas tecnologias. São Paulo: Casa do Código, 2015.
- ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados:** Fundamentos e Aplicações. 7ed. São Paulo: Pearson, 2019.
- SADALAGE, P.; FOWLER, M. **Nosql Essencial:** Um Guia Conciso Para o Mundo Emergente da Persistência Poliglota. São Paulo: Novatec, 2013.
- SINGH, Harry. **Data Warehouse**: conceitos, tecnologias, implementação e gerenciamento. São Paulo: Makron Books, 2001.

Bibliografia Complementar

Fatec
Jacareí
Prof. Francisco de Moura

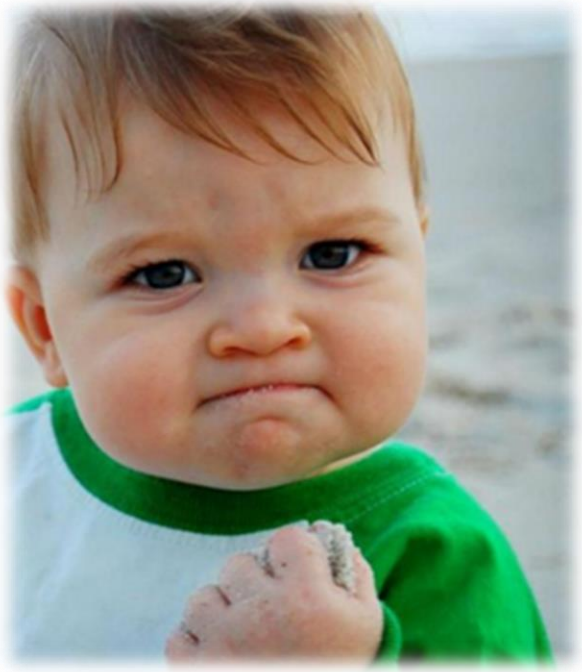
- FAROULT, Stephane. **Refatorando Aplicativos SQL**. Rio de Janeiro: Alta Books, 2009.
- PANIZ, D. **NoSQL:** Como armazenar os dados de uma aplicação moderna. Casa do Código, 2016.
- SOUZA, M. **Desvendando o Mongodb**. Rio de Janeiro: Ciência Moderna, 2015.

Dúvidas?

Fatec
Jacareí
Prof. Francisco de Moura

Considerações Finais

Fatec
Jacareí
Prof. Francisco de Moura



Professor(a):
Lucineide Pimenta

Bom descanso à todos!