

Introdução ao Banco de Dados

Mentor: Carlos Júnior

1 - Arquitetura Cliente e Servidor

- Lembra dela?
- Onde entra o Banco de Dados nela?

2 - Banco de Dados Relacional (SQL)

- O que é?
- Quando usá-lo?

3 - Banco de Dados Não Relacional (NoSQL)

- O que é?
- Quando usá-lo?

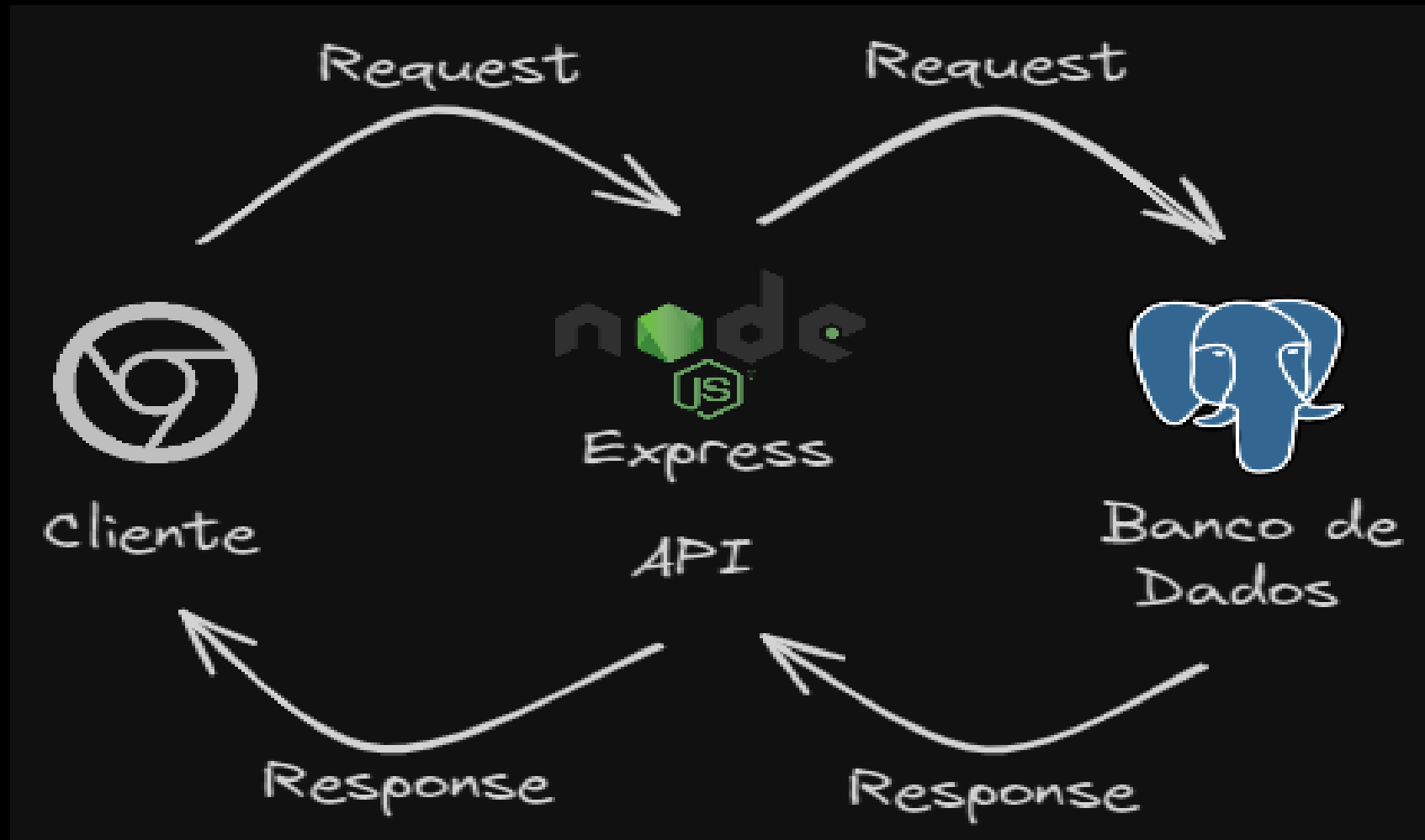
4 - Sistema de Gerenciamento de Banco de Dados (SGBD)

- O que é?
- Para que serve?

5 - SGBD: PostgreSQL

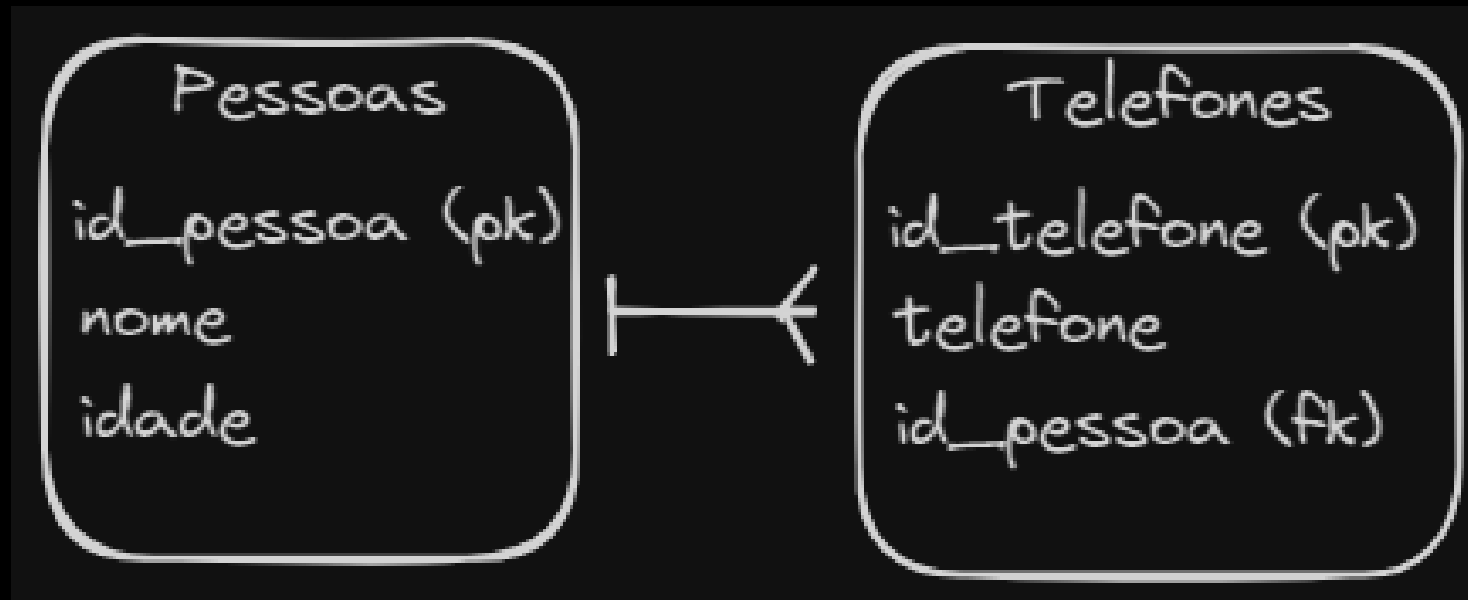
- Instalação
- Criando uma Base de Dados e suas tabelas

1 - Arquitetura Cliente e Servidor



Banco de Dados Relacional (SQL)

Tipo de Banco de Dados que trabalha com **tabelas** que possuem informações de uma entidade. Uma tabela pode possuir vários **registros** (linhas) que possuem os mesmos **atributos** (colunas). Cada registro possui um identificador único. A Base de Dados que adota esse modelo pode armazenar várias tabelas e acessa as informações **por registro**. As tabelas possuem relacionamentos entre si.



Banco de Dados Relacional (SQL)

Esse tipo de Banco de Dados é adequado quando é necessária alguma das seguintes situações:

1. Estrutura de dados bem definida
2. Integridade dos dados críticos
3. Consultas complexas e agregações

Banco de Dados Não Relacional (NoSQL)

Tipo de Banco de Dados que trabalha com **modelos de esquemas flexíveis** e que **não se relacionam**. Os principais modelos de esquemas são:

Orientado a Documentos

Tipo de Banco de Dados que trabalha com **coleções** que possuem informações de um tipo (entidade). Uma coleção pode possuir vários **documentos** (registros) que possuem **atributos**. Os atributos de uma coleção podem ser distintos, além de permitirem valores complexos. Cada documento possui um identificador único. A Base de Dados que adota esse modelo pode armazenar várias coleções e acessa as informações **por documento**. As coleções **não** possuem relacionamentos entre si.

Banco de Dados Não Relacional (NoSQL)

Chave e Valor

Tipo de Banco de Dados que trabalha com **uma tabela** que possui informações de um ou vários tipos (entidades). Uma tabela pode possuir vários **itens** (registros) que possuem **atributos**. Os atributos de um item podem ser distintos, além de permitirem valores complexos. Cada item possui um identificador único que será usado como chave. A Base de Dados que adota esse modelo pode armazenar vários itens e acessar as informações pela **chave dos itens**.

Banco de Dados Não Relacional (NoSQL)

Colunas Largas

Tipo de Banco de Dados que trabalha com **tabelas** que possuem informações de uma entidade. Uma tabela pode possuir vários **registros** (linhas) que possuem **atributos** (colunas). Os atributos de um registro podem ser distintos, além de permitirem valores complexos. Cada registro possui um identificador único. A Base de Dados que adota esse modelo pode armazenar várias tabelas e acessa as informações **por atributo**. As tabelas **não** possuem relacionamentos entre si.

Banco de Dados Não Relacional (NoSQL)

Grafos

Tipo de Banco de Dados que trabalha com **nós** (registros), que possuem informações de uma entidade, e **arestas**, que possuem informações de relacionamentos ou ações entre os nós. Um nó possui **atributos** que permite valores complexos. Cada nó possui um identificador único. A Base de Dados que adota esse modelo pode armazenar vários nós com várias **arestas** e acessa as informações **por nó**.

Banco de Dados Não Relacional (NoSQL)

Esse tipo de Banco de Dados é adequado quando é necessária alguma das seguintes situações:

1. Estrutura de dados flexível
2. Integridade dos dados não importa
3. Consultas simples

Sistema de Gerenciamento de Banco de Dados (SGBD)

É um software que fornece uma interface entre os dados armazenados em um Banco de Dados e os aplicativos que acessam esses dados. Por essa interface é possível criar, excluir e gerenciar um Banco de Dados. Em Bancos de Dados Relacionais, os SGBD's utilizam uma linguagem denominada de Structured Query Language (SQL).

SGBD: PostgreSQL

Uma vez instalado, podemos nos conectar ao PostgreSQL de diversas formas: via terminal, via softwares de interface gráfica (pgAdmin, etc) e extensões no VSCode (PostgreSQL, etc). Em todas elas, quatro informações serão solicitadas:

1. **Host:** endereço/domínio da máquina em que se encontra o PostgreSQL
2. **Port:** porta da máquina em que o PostgreSQL está acessível
3. **User:** usuário para se autenticar no PostgreSQL
4. **Password:** senha do usuário em questão

SGBD: PostgreSQL

Conexão via terminal:

```
psql -h <host> -p <port> -U <user>
```

Caso queira conectar localmente e na porta padrão:

```
psql -U <user>
```

SGBD: PostgreSQL

Criar uma base de dados:

```
postgres=# create database <db_name>;
```

Listar todas as bases de dados existentes:

```
postgres=# \l
```

Selecionar uma base de dados específica:

```
postgres=# \c <db_name>
```

SGBD: PostgreSQL

Criar uma tabela na base de dados:

```
<db_name>=# create table <tb_name> (  
    <col_name> <col_type> [constraint] [default <value>],  
    <col_name> <col_type> [constraint] [default <value>],  
);
```

Listar todas as tabelas de uma base de dados:

```
<db_name>=# \dt
```

Listar todas as informações das colunas de uma tabela específica de uma base de dados:

```
<db_name>=# \d <tb_name>
```


SGBD: PostgreSQL

Inserir um registro em uma tabela de uma base de dados:

```
<db_name>=# insert into <tb_name> (col_name, col_name) values (val_1, val_2);
```

Inserir mais de um registro, na mesma consulta, em uma tabela de uma base de dados:

```
<db_name>=# insert into <tb_name> (col_name, col_name) values (val_1, val_2), (val_3, val_4);
```

Listar todos os registros de uma tabela com todas as suas colunas:

```
<db_name>=# select * from <tb_name>;
```

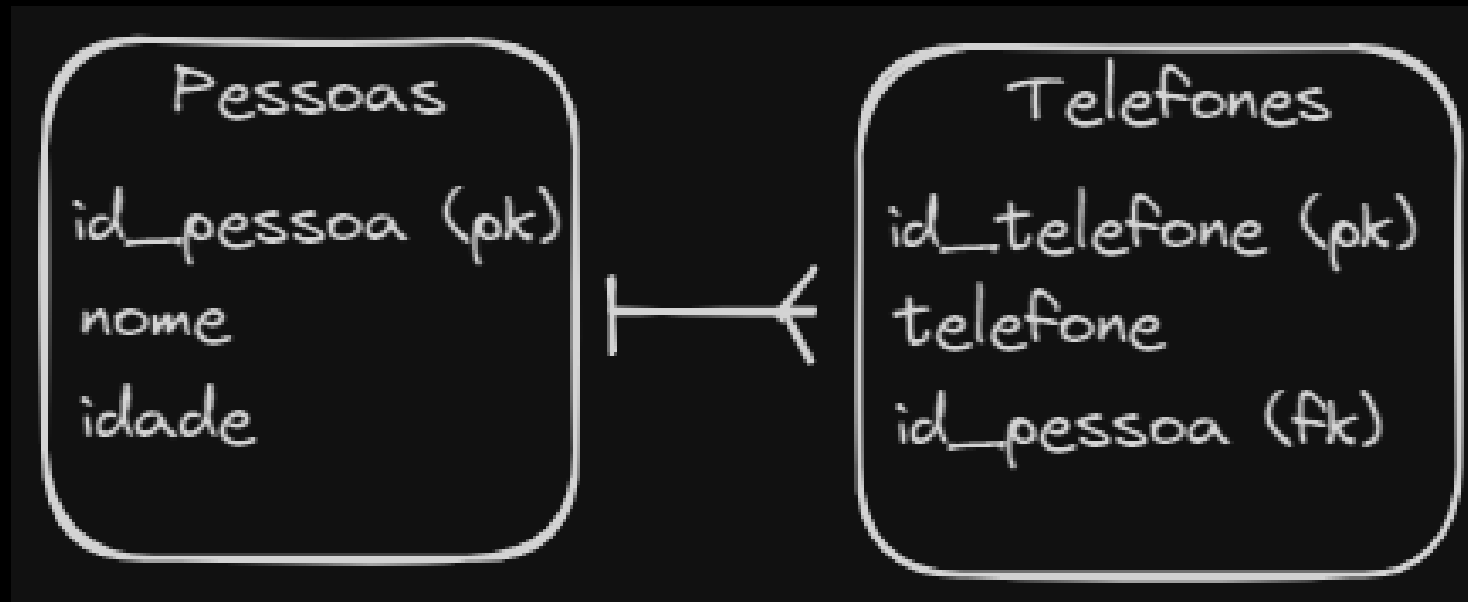
Sua vez!

Até aqui criamos informações: base de dados, tabelas e registros. A partir dos exercícios abaixo, descubra como deletá-las.

1. Como delete um registro específico de uma tabela? E todos os registros de uma vez?
2. Como delete uma tabela específica? E todas as tabelas de uma vez?
3. Como delete uma base de dados específica?

Sua vez!

4. Crie, via terminal, uma base de dados chamada agenda_telefonica. As tabelas e suas colunas são descritas na figura abaixo. Todos os campos de todas as tabelas não podem ser nulos. Além disso, a coluna telefone da tabela Telefones deve conter valores únicos. Escolha o tipo das colunas, mas tenha em mente a justificativa da escolha deles.



Sua vez!

5. Insira cinco pessoas, sendo duas delas com mais de um telefone cadastrado e uma que não possui telefone cadastrado.
6. Delete o telefone ou um dos telefones da primeira pessoa cadastrada.
7. Delete a pessoa que não possui telefone cadastrado.
8. Delete uma pessoa que possui algum telefone cadastrado. Caso tenha obtido algum erro, explique o motivo e como fez para contornar. Posteriormente, pesquise sobre a palavra reservada do SQL *ON DELETE* e explique para que serve e as opções que ela possui.