# Python

Fundamentos de **BIG DATA** e **DATA ANALYTCS**

Levi Marques | **Professor**

# Fluxo de Controle

> Break
> Continue

# Fluxo de Controle Break

**O que é a instrução Break em Python ?**

É uma instrução que encerra o Loop e retoma a execução da próxima instrução

# Fluxo de Controle Break

**O que é a instrução Continue em Python ?**

É uma instrução que retorna o controle para o inicio do loop while. Ela rejeita todas as instruções restantes na interação atual do loop.

# Banco de Dados

> Definição
> Características
> Tipos
  - **SQL**
  - **NoSQL**
> Comando em SQL

# Banco de dados | Definição

**Qual a definição de banco de dados ?**

É uma coleção organizada de informações - ou dados - estruturadas, normalmente armazenadas eletronicamente em um sistema de computador. Um banco de dados é geralmente controlado por um sistema de gerenciamento de banco de dados (DBMS).

# Banco de dados | Caracteristicas

## Quais suas características ?

É uma coleção organizada de informações - ou dados - estruturadas, normalmente armazenadas eletronicamente em um sistema de computador. Um banco de dados é geralmente controlado por um sistema de gerenciamento de banco de dados (DBMS).

# Banco de dados | SQL

## O que é SQL ?

***S**tructured **Q**uery **L**anguage*, **Linguagem de consulta estruturada.**

SQL é uma linguagem de programação usada por quase todos os bancos de dados **relacionais** para consultar, manipular e definir dados e fornecer controle de acesso.

# Banco de dados | NoSQL

## O que é NoSQL ?

***No Only Structured Query Language*, Não somente Linguagem de consulta estruturada.**

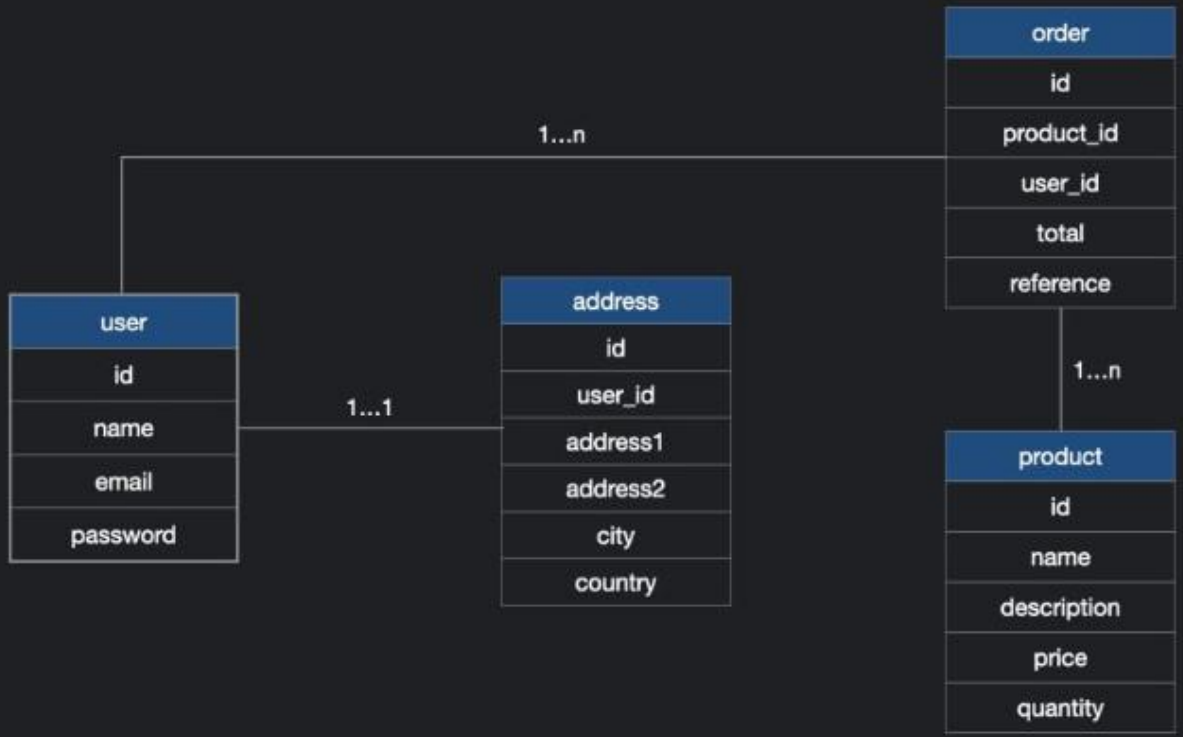
NoSQL é uma linguagem de programação usada para bancos de dados **não relacionais**. Ele pode conter SQL, porém há outras formas de manipular os dados.

# Banco de dados | Banco Relacional

## O que é um Banco Relacional ?

É um formato de banco rigidamente estruturado, baseado em tabelas. Os campos tem relacionamento entre si.

# Banco de dados | Banco Relacional



# Banco de dados | Banco Não Relacional

## O que é um Banco Não Relacional ?

É qualquer banco de dados que não segue o modelo relacional fornecido pelos sistemas tradicionais de gerenciamento de bancos de dados relacionais (SGBDR).

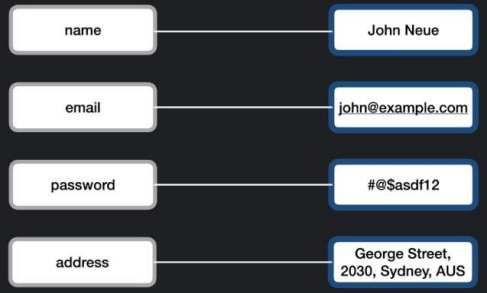
# Banco de dados | Banco Não Relacional

## Os tipos de Bancos Não Relacionais :

> Key-value stores
> Graph stores
> Column stores
> Document stores

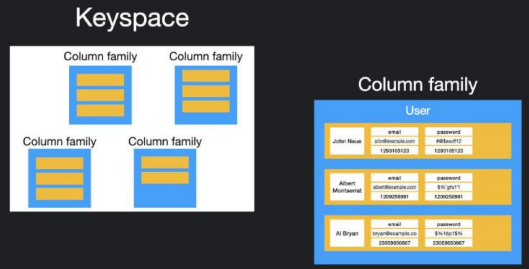
# Banco de dados | Banco Não Relacional

## > Key-value stores




## > Graph stores

## > Column stores




## > Document stores

```
[
  {
    id: 1293105123,
    name: "John Neue",
    email:"john@example.com",
    password:"#@$asdf12",
  },
  {
    id: 1209256991,
    name: "Albert Montserrat",
    email: "albert@example.com",
    password: "$%^gfal+!",
  },
  {
    id: 23059650867,
    name: "Al Bryan",
    email: "bryan@example.com",
    password: "$%16pl$%",
  }
]
```

# Banco de dados | Comandos SQL

> **CREATE**: cria novas tabelas em um banco de dados
> **ALTER**: alterar uma tabela já criada
> **INSERT**: adiciona registros a uma tabela
> **UPDATE**: atualiza os registros já inseridos
> **DELETE**: exclui registros de uma tabela
> **SELECT**: busca registros na tabela
> **GRANT**: permite acesso a objetos do banco de dados
> **REVOKE**: remove o acesso a objetos do banco de dados
> **DENY**: bloqueia o acesso para objetos e usuários específicos
> **DROP**: exclui uma tabela do banco de dados

# Banco de dados | Comandos SQL

> **CREATE**: cria novas tabelas em um banco de dados

```
CREATE TABLE usuarios (
id INTEGER PRIMARY KEY AUTOINCREMENT,
nome VARCHAR(50),
email VARCHAR(100),
senha VARCHAR(50)
);
```

# Banco de dados | Comandos SQL

> **ALTER**: alterar uma tabela já criada

```sql
ALTER TABLE usuarios ADD data_nascimento VARCHAR(10);
```

# Banco de dados | Comandos SQL

> **INSERT**: adiciona registros a uma tabela

```
INSERT INTO usuarios(nome, email, senha, data_nascimento)
VALUES ('Joao','joao_e_maria@gmail.com','Maria123','21/05/2001');
```

```
INSERT INTO usuarios(nome, email, senha, data_nascimento)
VALUES ('Maria','maria_e_joao@gmail.com','Joao123','21/05/2001');
```

```
INSERT INTO usuarios(nome, email, senha, data_nascimento)
VALUES ('Ana','joao_e_ana@gmail.com', 'Joao123','21/05/2001');
```

# Banco de dados | Comandos SQL

> **UPDATE**: atualiza os registros já inseridos

```
UPDATE usuarios SET senha="Ana123" WHERE nome="Joao"
```

# Banco de dados | Comandos SQL

> **DELETE**: exclui registros de uma tabela

```
DELETE FROM usuarios WHERE email="joao_e_maria@gmail.com"
```

# Banco de dados | Comandos SQL

> **SELECT**: busca registros na tabela

```
SELECT * FROM usuarios
```

```
SELECT * FROM usuarios WHERE nome="Ana“
```

```
SELECT nome FROM usuarios WHERE senha="Joao123“
```

```
SELECT * FROM usuarios WHERE senha="Joao123" ORDER BY nome DESC
```

# Banco de dados | Comandos SQL

> **GRANT**: permite acesso a objetos do banco de dados

```
GRANT SELECT, INSERT, UPDATE ON usuarios TO usuario_db_01
```

# Banco de dados | Comandos SQL

> **REVOKE**: remove o acesso a objetos do banco de dados

```
REVOKE SELECT ON usuarios FROM usuario_db_01
```

# Banco de dados | Comandos SQL

> **DENY**: bloqueia o acesso para objetos e usuários específicos

```
DENY SELECT ON usuarios TO usuario_db_01
```

# Banco de dados | Comandos SQL

> **DROP**: exclui uma tabela do banco de dados

```
DROP TABLE usuários
```

```
DROP DATABASE db
```

# Obrigado !