

# **Manipulação de Dados com SQL**

**Wendel Melo**

Faculdade de Computação

Universidade Federal de Uberlândia

[www.facom.ufu.br/~wendelmelo](http://www.facom.ufu.br/~wendelmelo)

Banco de Dados I

# Structured Query Language (SQL)

- “*Linguagem de consulta estruturada*”;
- **Padrão em SGBD's relacionais**, o que favorece a portabilidade entre diferentes SGBD's;
- Possui um **conjunto de instruções normatizado** (ANSI 1986, ISO 1987, 1992, 1999, 2003, 2006, 2008, 2011, 2016, ...);
- Adicionalmente, SGBD's incorporam recursos adicionais para facilitar o uso;

# Structured Query Language (SQL)

- **Linguagem declarativa:** usuários se preocupam em dizer *o que* deve ser feito, e não *como* deve ser feito, simplificando assim seu uso;
- **Insensível a caixa:** comandos podem estar em maiúsculo ou minúsculo;
- Comandos SQL terminam com o caracter ';' (**ponto e vírgula**).
- Atributos deixados em branco são marcados com o valor **NULL**;

Peculiaridade do NULL: NULL nunca é igual a outro NULL, cada NULL é um valor desconhecido

# Structured Query Language (SQL)

- SQL possui cláusulas para atuar como: Pode cair na prova! Diferença entre DDL E DML
  - Linguagem de Definição de Dados (DDL); Criar a estrutura do BD
  - Linguagem de Manipulação de Dados (DML); Inserir, apagar ou atualizar dadosde um
  - Linguagem de Consulta de Dados (DQL);
  - Linguagem de Transação de Dados (DTL);
  - Linguagem de Controle de Dados (DCL).

# Structured Query Language (SQL)

- SQL possui cláusulas para atuar como:
  - Linguagem de Definição de Dados (DDL);
  - Linguagem de Manipulação de Dados (DML);
  - Linguagem de Consulta de Dados (DQL);
  - Linguagem de Transação de Dados (DTL);
  - Linguagem de Controle de Dados (DCL).

# Linguagem de Manipulação de Dados (DML)

- Através de uma DML, é possível inserir, alterar e remover informações em um banco de dados.

# Linguagem de Manipulação de Dados (DML)

- Através de uma DML, é possível inserir, alterar e remover informações em um banco de dados.
- As principais cláusulas DML na SQL são: **Pode cair na PROVA!**
  - **INSERT**: insere linhas em uma tabela;
  - **DELETE**: remove linhas de uma tabela;
  - **UPDATE**: atualiza (altera) linhas em uma tabela;

# Propriedades ACID

PROVA!

- Operações em SQL devem estar de acordo com as chamadas propriedades ACID (**A**tomicidade, **C**onsistência, **I**solamento e **D**urabilidade):
- **Atomicidade:** Uma operação é considerada como uma unidade de processamento atômica.

Atomicidade: Algo que não pode ser quebrado, ou seja, ou executa tudo sem erro ou não executa nada, te

  - Isso significa que ou a operação é realizada por completo, ou, em caso de falha de alguma das operações, o banco de dados deve ser restaurado ao estado anterior à execução da transação (não há estado intermediário. *É tudo ou nada*).



# Propriedades ACID

- **Consistência:** impõe que se a operação for executada do início ao fim, ela deve levar o banco de dados de um estado consistente até outro estado consistente
  - Dizemos que um estado do banco de dados é consistente se o mesmo atende a todas as restrições definidas no esquema e no banco de dados em geral.

# Propriedades ACID

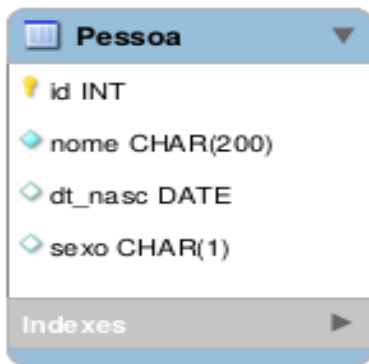
- **Isolamento:** uma operação deve ser encarada como se fosse executada isoladamente de outras operações, isto é, ainda que operações sejam executadas simultaneamente, não devem interferir uma na outra. Isolamento: única que pode ser quebrada
- **Durabilidade (Permanência):** Mudanças aplicadas por uma operação devem persistir no banco de dados. Essas mudanças não devem ser perdidas em caso de alguma falha.

# Inserindo linhas em uma Tabela

A cláusula **INSERT** permite inserir linhas (tuplas) em uma tabela:

```
INSERT INTO  <tabeLa>  VALUES  ( <atributos da linha> );
```

# Inserindo linhas em uma Tabela



PROVA! ->

DECORE ->|V

```
CREATE TABLE Pessoa (  
    id INT,  
    nome CHAR(200) NOT NULL,  
    dt_nasc DATE,  
    sexo CHAR(1) DEFAULT 'I',  
    PRIMARY KEY(id)  
);
```

```
INSERT INTO Pessoa VALUES  
(1, 'Jessica G', '1991-03-19', 'F'),  
(2, 'Leidiana I', '1993-03-28', 'F'),  
(3, 'Lucas S', NULL, 'M');
```

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |

# Inserindo linhas em uma Tabela

Também é possível inserir linhas especificando os atributos a serem preenchidos:

```
INSERT INTO    <tabela>( <atributos> )  VALUES  
                ( <atributos da linha> );
```

# Inserindo linhas em uma Tabela

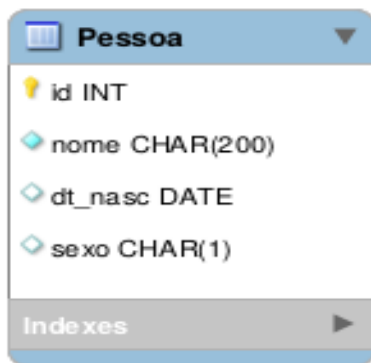


Table structure for 'Pessoa':

- id INT
- nome CHAR(200)
- dt\_nasc DATE
- sexo CHAR(1)

Indexes

```
CREATE TABLE Pessoa (  
    id INT,  
    nome CHAR(200) NOT NULL,  
    dt_nasc DATE,  
    sexo CHAR(1) DEFAULT 'I',  
    PRIMARY KEY(id) );
```

```
INSERT INTO Pessoa (nome, id)  
VALUES  
    ('Rachel M', 7),  
    ('Rodrigo S', 5);
```

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

# Inserindo linhas em uma Tabela

Também é possível inserir linhas especificando os atributos a serem preenchidos:

```
INSERT INTO    <tabela>( <atributos> )  VALUES  
                                     ( <atributos da linha> );
```

Essa forma de inserção é a mais recomendada devido ao fato de que a estrutura de uma tabela pode vir a mudar.

Ao especificar exatamente quais os atributos estão sendo preenchidos e sua respectiva ordem, evita-se potenciais problemas futuros com alterações na estrutura da tabela.

# Inserindo linhas em uma Tabela

Na prática, ao inserir um novo registro em uma tabela cuja chave primária é um inteiro auto-incrementável, pode ser preciso obter o valor da chave primária dado ao novo registro.

Para isso, no PostgreSQL, usamos uma cláusula *RETURNING*

```
INSERT INTO    <tabela> ( <atributos> )  VALUES
                                     ( <atributos da linha> )
                                     RETURNING <colunas>;
```



# Inserindo linhas e obtendo colunas

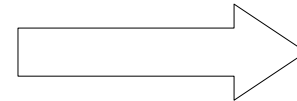
| Projeto             |
|---------------------|
| id SERIAL           |
| titulo VARCHAR(500) |
| dt_inicio DATE      |
| id_coordenador INT  |
| Indexes             |

Projeto

|   | id | titulo                        | dt_inicio  | id_coordenador |
|---|----|-------------------------------|------------|----------------|
| 1 | 1  | Máquina do tempo longitudinal | 2024-01-21 | [NULL]         |
| 2 | 2  | Óculos de raios X             | 2025-02-15 | [NULL]         |

```
INSERT INTO Projeto(titulo, dt_inicio) VALUES  
('Pílula para falar Inglês', '2030-08-03'),  
('Máquina e teletransporte', '2024-03-18')  
RETURNING id;
```

Retorna



|   | id |
|---|----|
| 1 | 3  |
| 2 | 4  |

Projeto

|   | id | titulo                        | dt_inicio  | id_coordenador |
|---|----|-------------------------------|------------|----------------|
| 1 | 1  | Máquina do tempo longitudinal | 2024-01-21 | [NULL]         |
| 2 | 2  | Óculos de raios X             | 2025-02-15 | [NULL]         |
| 3 | 3  | Pílula para falar Inglês      | 2030-08-03 | [NULL]         |
| 4 | 4  | Máquina e teletransporte      | 2024-03-18 | [NULL]         |

# Removendo linhas de uma Tabela

- A cláusula **DELETE** permite a remoção de linhas (tuplas) de uma tabela.

# Removendo linhas de uma Tabela

- A cláusula **DELETE** permite a remoção de linhas (tuplas) de uma tabela.
- Pode ser usada em conjunto com uma cláusula *WHERE* para especificar uma condição a ser satisfeita pelas linhas sendo removidas:

```
DELETE FROM <tabela> [WHERE <condição de seleção>];
```

# Removendo linhas de uma Tabela

- A cláusula **DELETE** permite a remoção de linhas (tuplas) de uma tabela.
- Pode ser usada em conjunto com uma cláusula *WHERE* para especificar uma condição a ser satisfeita pelas linhas sendo removidas:

DELETE FROM <tabela> [WHERE <condição de seleção>];

Opcional: apenas as linhas cuja condição de seleção resultar em *TRUE* serão apagadas.

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
DELETE FROM Pessoa WHERE id = 7;
```

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
DELETE FROM Pessoa WHERE id = 7;
```

Remove da tabela *Pessoa* as linhas cujo campo *id* tem o valor 7.

Como *id* é a chave primária, sabemos de antemão que, no máximo, uma linha será apagada.

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 5  | Rodrigo S  | [NULL]     | I    |

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
DELETE FROM Pessoa WHERE id = 9;
```

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
DELETE FROM Pessoa WHERE id = 9;
```

Como não há linha na tabela que atenda à condição especificada, nenhuma linha será removida.

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |



# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
DELETE FROM Pessoa WHERE sexo = 'F';
```

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
DELETE FROM Pessoa WHERE sexo = 'F';
```

Remove da tabela *Pessoa* as linhas cujo campo *sexo* esteja com o valor 'F'.

|   | id | nome      | dt_nasc | sexo |
|---|----|-----------|---------|------|
| 1 | 3  | Lucas S   | [NULL]  | M    |
| 2 | 7  | Rachel M  | [NULL]  | I    |
| 3 | 5  | Rodrigo S | [NULL]  | I    |

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
DELETE FROM Pessoa WHERE  
    '1990-01-01' <= dt_nasc AND dt_nasc <= '1991-12-31';
```

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
DELETE FROM Pessoa WHERE  
'1990-01-01' <= dt_nasc AND dt_nasc <= '1991-12-31';
```

Remove da tabela *Pessoa* as linhas cujo campo *dt\_nasc* esteja com o valor no intervalo entre 01/01/1990 e 31/12/1991.

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 2  | Leidiana I | 1993-03-28 | F    |
| 2 | 3  | Lucas S    | [NULL]     | M    |
| 3 | 7  | Rachel M   | [NULL]     | I    |
| 4 | 5  | Rodrigo S  | [NULL]     | I    |

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
DELETE FROM Pessoa WHERE  
'1990-01-01' <= dt_nasc AND dt_nasc <= '1991-12-31';
```

Note que é possível utilizar operadores lógicos na condição de seleção.

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 2  | Leidiana I | 1993-03-28 | F    |
| 2 | 3  | Lucas S    | [NULL]     | M    |
| 3 | 7  | Rachel M   | [NULL]     | I    |
| 4 | 5  | Rodrigo S  | [NULL]     | I    |

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
DELETE FROM Pessoa WHERE NOT( dt_nasc <= '1992-01-01' );
```

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
DELETE FROM Pessoa WHERE NOT( dt_nasc <= '1992-01-01' );
```

Remove da tabela Pessoa as linhas cujo campo *dt\_nasc* tenha um valor que não seja menor ou igual que 01/01/1992.

|   | id | nome      | dt_nasc    | sexo |
|---|----|-----------|------------|------|
| 1 | 1  | Jessica G | 1991-03-19 | F    |
| 2 | 3  | Lucas S   | [NULL]     | M    |
| 3 | 7  | Rachel M  | [NULL]     | I    |
| 4 | 5  | Rodrigo S | [NULL]     | I    |

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
DELETE FROM Pessoa WHERE NOT( dt_nasc <= '1992-01-01' );
```

A comparação `NOT( dt_nasc <= '1992-01-01' )` resultará em `UNKNOWN` (desconhecido) para as linhas onde `dt_nasc` está vazio. Por essa razão, essas linhas não são removidas.

|   | id | nome      | dt_nasc    | sexo |
|---|----|-----------|------------|------|
| 1 | 1  | Jessica G | 1991-03-19 | F    |
| 2 | 3  | Lucas S   | [NULL]     | M    |
| 3 | 7  | Rachel M  | [NULL]     | I    |
| 4 | 5  | Rodrigo S | [NULL]     | I    |



# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

- Como remover as linhas cujo o campo *dt\_nasc* esteja vazio?

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

- Como remover as linhas cujo o campo *dt\_nasc* esteja vazio?
- Devemos recordar que quando o campo está vazio, ele é marcado com o valor *NULL*;

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

- Como remover as linhas cujo o campo *dt\_nasc* esteja vazio?
- Devemos recordar que quando o campo está vazio, ele é marcado com o valor *NULL*;
- Todavia, o valor *NULL* é tratado de modo diferente dos demais, pois representa um valor que não se conhece ou não se aplica para a tupla em questão.

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
DELETE FROM Pessoa WHERE dt_nasc = NULL;
```

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
DELETE FROM Pessoa WHERE dt_nasc = NULL;
```

Uma comparação de igualdade ou desigualdade envolvendo o valor *NULL* sempre resulta em *UNKNOWN* (desconhecido).

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
DELETE FROM Pessoa WHERE dt_nasc = NULL;
```

Em SQL um valor *NULL* nunca é considerado igual a outro *NULL*, pois ambos representam algum valor que não se conhece ou não se aplica.

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
DELETE FROM Pessoa WHERE dt_nasc = NULL;
```

Por essa razão, a linha acima não surte qualquer efeito na tabela, pois a condição de seleção sempre resultará em *UNKNOWN* (desconhecido).

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |



# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

- Como remover as linhas cujo o campo *dt\_nasc* esteja vazio?

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

- Como remover as linhas cujo o campo *dt\_nasc* esteja vazio?
- Através do operador ***IS NULL***;

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
DELETE FROM Pessoa WHERE dt_nasc IS NULL;
```

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
DELETE FROM Pessoa WHERE dt_nasc IS NULL;
```

A comparação resultará em *TRUE* para todas as linhas onde *dt\_nasc* estiver com o valor *NULL* (vazio).

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

- Para selecionar valores não nulos, podemos usar o operador ***IS NOT NULL***.

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
DELETE FROM Pessoa WHERE dt_nasc IS NOT NULL;
```

|   | id | nome      | dt_nasc | sexo |
|---|----|-----------|---------|------|
| 1 | 3  | Lucas S   | [NULL]  | M    |
| 2 | 7  | Rachel M  | [NULL]  | I    |
| 3 | 5  | Rodrigo S | [NULL]  | I    |

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

- O que ocorre se o DELETE for utilizado sem WHERE?

```
DELETE FROM Pessoa;
```

# Removendo linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

- O que ocorre se o DELETE for utilizado sem WHERE?

```
DELETE FROM Pessoa;
```

- Todas as linhas da tabela serão apagadas!



# Alterando linhas de uma Tabela

- A cláusula *UPDATE* permite a atualização (alteração) de linhas (tuplas) de uma tabela.

# Alterando linhas de uma Tabela

- A cláusula *UPDATE* permite a atualização (alteração) de linhas (tuplas) de uma tabela.
- Assim como *DELETE*, também pode ser usada em conjunto com uma cláusula *WHERE* para especificar uma condição a ser satisfeita pelas linhas sendo removidas:

*UPDATE*    <tabela>    SET {<coluna> = <expressão>}\*

[WHERE   <condição de seleção>];

# Alterando linhas de uma Tabela

- A cláusula *UPDATE* permite a atualização (alteração) de linhas (tuplas) de uma tabela.
- Assim como *DELETE*, também pode ser usada em conjunto com uma cláusula *WHERE* para especificar uma condição a ser satisfeita pelas linhas sendo removidas:

*UPDATE*    <tabela>    SET {<coluna> = <expressão>}\*

[*WHERE*    <condição de seleção>];

Opcional: apenas as linhas cuja condição de seleção resultar em *TRUE* serão atualizadas.

# Alterando linhas de uma Tabela

Pessoa

|   | 123 id | ABC nome   | dt_nasc    | ABC sexo |
|---|--------|------------|------------|----------|
| 1 | 1      | Jessica G  | 1991-03-19 | F        |
| 2 | 2      | Leidiana I | 1993-03-28 | F        |
| 3 | 3      | Lucas S    | [NULL]     | M        |
| 4 | 7      | Rachel M   | [NULL]     | I        |
| 5 | 5      | Rodrigo S  | [NULL]     | I        |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
UPDATE Pessoa SET nome = 'Jeh Melo' WHERE id = 1;
```

|   | 123 id | ABC nome   | dt_nasc    | ABC sexo |
|---|--------|------------|------------|----------|
| 1 | 2      | Leidiana I | 1993-03-28 | F        |
| 2 | 3      | Lucas S    | [NULL]     | M        |
| 3 | 7      | Rachel M   | [NULL]     | I        |
| 4 | 5      | Rodrigo S  | [NULL]     | I        |
| 5 | 1      | Jeh Melo   | 1991-03-19 | F        |

# Alterando linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
UPDATE Pessoa SET nome = 'Jeh Melo' WHERE id = 1;
```

Atualiza o campo *nome* para 'Jeh Melo' nas linhas cujo campo *id* tenha o valor 1.

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 2  | Leidiana I | 1993-03-28 | F    |
| 2 | 3  | Lucas S    | [NULL]     | M    |
| 3 | 7  | Rachel M   | [NULL]     | I    |
| 4 | 5  | Rodrigo S  | [NULL]     | I    |
| 5 | 1  | Jeh Melo   | 1991-03-19 | F    |

# Alterando linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
UPDATE Pessoa SET sexo = 'F', nome = 'Luna Souza' WHERE id = 3;
```

# Alterando linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
UPDATE Pessoa SET sexo = 'F', nome = 'Luna Souza' WHERE id = 3;
```

Atualiza os campos *sexo* e *nome* nas linhas cujo campo *id* está com o valor 3.

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 7  | Rachel M   | [NULL]     | I    |
| 4 | 5  | Rodrigo S  | [NULL]     | I    |
| 5 | 3  | Luna Souza | [NULL]     | F    |

# Alterando linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
UPDATE Pessoa SET sexo = NULL WHERE sexo = 'I';
```



# Alterando linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
UPDATE Pessoa SET sexo = NULL WHERE sexo = 'I';
```

Atualiza o campo sexo para o valor *NULL* (vazio) nas linhas cujo o campo sexo está com o valor 'I'.

|   | id | nome       | dt_nasc    | sexo   |
|---|----|------------|------------|--------|
| 1 | 1  | Jessica G  | 1991-03-19 | F      |
| 2 | 2  | Leidiana I | 1993-03-28 | F      |
| 3 | 3  | Lucas S    | [NULL]     | M      |
| 4 | 7  | Rachel M   | [NULL]     | [NULL] |
| 5 | 5  | Rodrigo S  | [NULL]     | [NULL] |

# Alterando linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
UPDATE Pessoa SET id = 10*id;
```

# Alterando linhas de uma Tabela

Pessoa

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 1  | Jessica G  | 1991-03-19 | F    |
| 2 | 2  | Leidiana I | 1993-03-28 | F    |
| 3 | 3  | Lucas S    | [NULL]     | M    |
| 4 | 7  | Rachel M   | [NULL]     | I    |
| 5 | 5  | Rodrigo S  | [NULL]     | I    |

| Pessoa  |           |
|---------|-----------|
| id      | INT       |
| nome    | CHAR(200) |
| dt_nasc | DATE      |
| sexo    | CHAR(1)   |
| Indexes |           |

```
UPDATE Pessoa SET id = 10*id;
```

Atualiza o campo id multiplicando o próprio valor do id por 10.

Note que a ausência de *WHERE* faz com que a operação seja aplicada a todas as linhas.

|   | id | nome       | dt_nasc    | sexo |
|---|----|------------|------------|------|
| 1 | 10 | Jessica G  | 1991-03-19 | F    |
| 2 | 20 | Leidiana I | 1993-03-28 | F    |
| 3 | 30 | Lucas S    | [NULL]     | M    |
| 4 | 70 | Rachel M   | [NULL]     | I    |
| 5 | 50 | Rodrigo S  | [NULL]     | I    |

# A Cláusula *WHERE*

- É recomendável sempre testar a condição de seleção especificada por meio da cláusula *WHERE* antes de realizar operações de atualização e remoção.
- Do contrário, pode-se remover ou alterar dados indevidamente;
- O teste pode ser feito através da cláusula *SELECT* (discutida adiante).

# Prática com PostgreSQL

- Para praticar os comandos SQL, necessitaremos de um servidor de banco de dados rodando PostgreSQL. Há duas opções:
  - 1) Instalar o PostgreSQL na sua máquina pessoal (menos recomendável, pois o serviço é relativamente pesado);
  - 2) Utilizar um servidor online (mais recomendável, sem exigir tanto da máquina, porém necessitando de internet)
- Para o uso no laboratório, usaremos o *ElephantSQL*:  
*<https://www.elephantsql.com>*

# Prática com PostgreSQL

- Para criar uma conta, escolha o plano gratuito na página <https://www.elephantsql.com/plans.html> , se cadastre e habilite o serviço criando uma nova instância;
- Obtenha os dados de conexão do seu serviço;
- Para rodar os comandos, é oportuno usar um cliente de banco de dados. Aqui, recomendamos o *DBeaver*: <https://dbeaver.io> ;
- Com seus dados de conexão da conta *ElephantSQL*, abra uma conexão com a mesma no *Dbeaver* (uso o método *host*);
- A partir daí, se tudo tiver dado certo, você está pronto para as práticas com SQL!