



**Ministério da Educação
Instituto Federal de Mato Grosso
Campus Cuiabá – Cel. Octayde Jorge da Silva
Departamento de Área de Informática**



Aula RED 12 – PostgreSQL: Comandos SQL – continuação (Fundamentos de BD)

Profª Juliana Saragiotto Silva

IFMT - Campus Cuiabá

Cuiabá, MT – 15/03/2021



Agenda

1. Objetivo da aula
2. PostgreSQL (revisão):
 - Ambiente de trabalho (PGAdmin);
 - Comandos SQL de inserção e seleção
3. Comandos SQL:
 - Update (atualização de dados)
 - Delete (exclusão de dados)
4. Atividade da semana



1. Objetivo da aula

- Revisar os comandos básicos de SQL aprendidos na semana passada: INSERT e SELECT);
- Apresentar os comandos SQL para atualização (UPDATE) e exclusão (DELETE) de dados.



2. PostgreSQL (revisão): PGAdmin

← → ↻ ⓘ 127.0.0.1:35203/browser/

pgAdmin File ▾ Object ▾ Tools ▾ Help ▾

Browser

- Servers (1)
 - PostgreSQL 13
 - Databases (2)
 - PROTOTIPO**
 - Cast
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Schemas (1)

→

- Schemas (1)
 - public**
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Procedures
 - Sequences
 - Tables (2)**
 - departamento
 - funcionario
 - Trigger Functions



2. PostgreSQL (revisão):

Tools -> Query Tool

The screenshot displays the pgAdmin 4 web interface. The top navigation bar includes 'pgAdmin', 'File', 'Object', 'Tools', and 'Help'. The 'Tools' menu is open, showing options: 'New ERD Project (Beta)', 'Query Tool' (highlighted), and 'Schema Diff'. The left sidebar shows a tree view with 'Servers (1)' expanded to 'PostgreSQL 13', which is further expanded to 'Databases (2)'. The main area shows the 'Query Editor' tab with a single line of text '1'. A 'Notifications' panel at the bottom shows a table with headers: 'Recorded time', 'Event', 'Process ID', and 'Payload'. The table is currently empty, with the text 'No data found' at the bottom.

Recorded time	Event	Process ID	Payload
No data found			



2. PostgreSQL (revisão):

Tools -> Query Tool

Dashboard Properties SQL Statistics Dependencies Dependents PROTOTIPO/postgres@PostgreSQL 13 *

Query Editor Query History Scratch Pad

1 **SELECT** *
2 **FROM** departamento

F5

Data Output Explain Messages

id_departamento [PK] integer	nome character varying (100)	sigla character varying (15)	ramal character varying (11)	responsavel character varying (100)



2. PostgreSQL (revisão):

Comando SQL INSERT

Exemplo (Departamento):

INSERT into **Departamento**

(id_departamento, nome, sigla, ramal, responsavel)

VALUES

(1, 'Tecnologia da Informação', 'TI', '1402', 'Gabriel');



2. PostgreSQL (revisão):

Comando SQL SELECT

Query Editor Query History

```
1 SELECT *
2 FROM departamento
```

Data Output Explain Messages

	id_departamento [PK] integer	nome character varying (100)	sigla character varying (15)	ramal character varying (11)	responsavel character varying (100)
1	1	Tecnologia da Informação	TI	1402	Gabriel



2. PostgreSQL (revisão):

Comando SQL INSERT (bloco)

Exemplo (Departamento):

INSERT into Departamento

(id_departamento, nome, sigla, ramal, responsavel)

VALUES

(2, 'Gestão de Pessoas', 'GP', '1403', 'Keila'),

(3, 'Financeiro', 'FN', '1404', 'Renan'),

(4, 'Comercial', 'CM', '1405', 'Lucas');

2. PostgreSQL (revisão):

Comando SQL SELECT

Query Editor

Query History

Scratch Pad

1

SELECT *

2

FROM Departamento

Data Output

Explain

Messages

		id_departamento [PK] integer	nome character varying (100)	sigla character varying (15)	ramal character varying (11)	responsavel character varying (100)
1		1	Tecnologia da Informação	TI	1402	Gabriel
2		2	Gestão de Pessoas	GP	1403	Keila
3		3	Financeiro	FN	1404	Renan
4		4	Comercial	CM	1405	Lucas



3. Comandos SQL: UPDATE

Sintaxe:

UPDATE Nome_tabela

SET atributo1 = novo_valor -- atributo a ser alterado

WHERE -- filtro? -- qual registro?

atributo1 ou atributon = identificador_do registro
ou valor procurado



3. Comandos SQL: UPDATE

Exemplo (Departamento): **opção 1**

UPDATE Departamento

SET responsavel = 'Glaucia'

WHERE

id_departamento = 1;



3. Comandos SQL: UPDATE

Exemplo (Departamento): **opção 2** (OU)

UPDATE Departamento

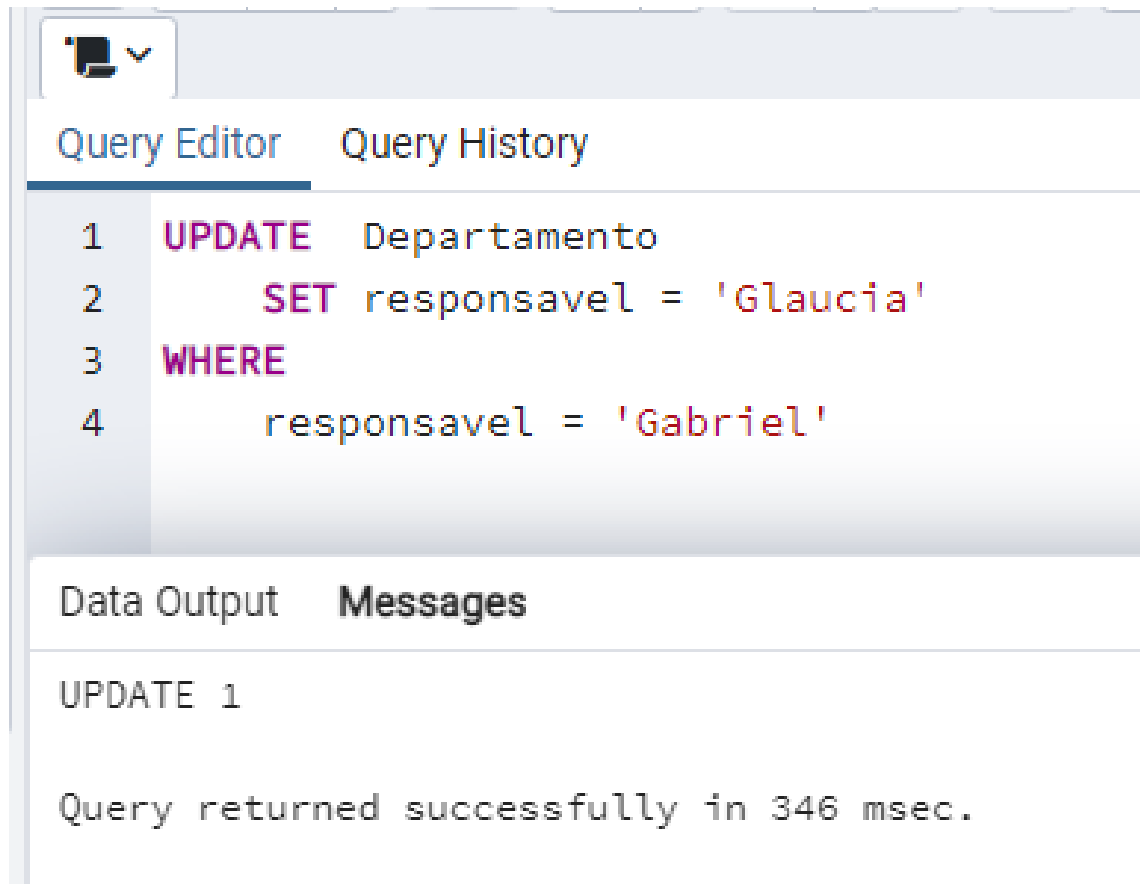
SET responsavel = 'Glaucia'

WHERE

responsavel = 'Gabriel';



3. Comandos SQL: UPDATE (Resultado)



The screenshot shows a SQL query editor with a toolbar at the top. Below the toolbar are two tabs: 'Query Editor' (active) and 'Query History'. The 'Query Editor' tab displays a SQL UPDATE statement with line numbers 1 through 4. Below the query editor are two more tabs: 'Data Output' and 'Messages'. The 'Messages' tab is active and shows the execution result: 'UPDATE 1' and 'Query returned successfully in 346 msec.'

```
1 UPDATE Departamento
2     SET responsavel = 'Glaucia'
3 WHERE
4     responsavel = 'Gabriel'
```

UPDATE 1

Query returned successfully in 346 msec.



3. Comandos SQL: UPDATE (Resultado)

Query Editor Query History

```
1 SELECT *
2 FROM Departamento
```

Output Messages

departamento { integer	nome character varying (100)	sigla character varying (15)	ramal character varying (11)	responsavel character varying (100)
2	Gestão de Pessoas	GP	1403	Keila
3	Financeiro	FN	1404	Renan
4	Comercial	CM	1405	Lucas
1	Tecnologia da Informação	TI	1402	Glaucia ←

Query Editor Query History

```
1 SELECT *
2 FROM Departamento
3 WHERE
4     id_departamento = 1
```

Data Output Messages

id_departamento [PK] integer	nome character varying (100)	sigla character varying (15)	ramal character varying (11)	responsavel character varying (100)
1	1 Tecnologia da Informação	TI	1402	Glaucia



3. Comandos SQL: DELETE

Sintaxe:

DELETE

FROM Nome_Tabela

WHERE -- filtro? Qual registro eu quero excluir?

atributo1 = valor_deseja_excluir



3. Comandos SQL: Registros na Tabela

```
1 SELECT *|
2 FROM Funcionario
```

Data Output Messages

	id_funcionario [PK] integer	nome character varying (100)	sexo character (1)	endereco character varying (200)	telefone character varying (11)	dta_nasc date	email character
1	2	Jessica	F	Rua 20, Qd 5, B. Morada do O...	65999892222	2002-03-08	jessica@
2	3	João Marcos	M	Rua X, B. CPA	65988982525	2001-01-07	joaomarc
3	4	Lucas Adriano	M	Rua Z, Qd Y, B. Centro	65998980202	1995-04-26	lucasadria
4	5	William	M	Rua A, Qd 10, B. Cachorro Pel...	65997970404	1998-03-08	william@



3. Comandos SQL: DELETE

Exemplo (Funcionario): **opção 1**

DELETE

FROM Funcionario

WHERE

id_funcionario = 5;



3. Comandos SQL: DELETE (Resultado)

Query Editor

Query History

```
1 DELETE
2 FROM Funcionario
3 WHERE
4 id_funcionario = 5;
5
```

Data Output Messages

DELETE 1

Query returned successfully in 264 msec.

Query Editor

Query History

```
1 SELECT *
2 FROM Funcionario
3
```

Data Output Messages

	id_funcionario [PK] integer	nome character varying (100)	sexo character (1)	endereco character varying (200)	telefone character varying (11)	dta_nasc date	email char
1	2	Jessica	F	Rua 20, Qd 5, B. Morada do O...	65999892222	2002-03-08	jessie
2	3	João Marcos	M	Rua X, B. CPA	65988982525	2001-01-07	joao
3	4	Lucas Adriano	M	Rua Z, Qd Y, B. Centro	65998980202	1995-04-26	lucas



3. Comandos SQL: DELETE

Exemplo (Funcionario): **opção 2**

DELETE

FROM Funcionario

WHERE

Nome_Funcionario LIKE 'João%';



3. Comandos SQL: SELECT (com mais de 1 tabela)

Sintaxe:

SELECT atributo_x, atributo_y, atributo_z -- atributos que você
necessita trazer na consulta

FROM Tabela_1, Tabela_2 -- pode ter mais de 2 tabelas

WHERE Tabela_1.atributo_z = Tabela_2.atributo_z

-- atributo que tem em comum nas 2 tabelas (garantir a integridade)

ORDER BY 1, 2 – ordenar a consulta SQL, pelos 2 primeiros atributos



3. Comandos SQL: SELECT (com mais de 1 tabela)

Exemplo:

```
SELECT Departamento.Nome, Funcionario.Nome,  
        Email, Telefone  
  
FROM Departamento, Funcionario  
  
WHERE Departamento.Id_Departamento = Funcionario.Id_Departamento  
  
ORDER BY 1, 2
```



3. Comandos SQL: SELECT (com mais de 1 tabela)

Sintaxe:

SELECT atributo_x, atributo_y, atributo_z -- atributos que você
necessita trazer na consulta

FROM Tabela_1, Tabela_2 -- pode ter mais de 2 tabelas

WHERE Tabela_1.atributo_z = Tabela_2.atributo_z

-- atributo que tem em comum nas 2 tabelas (garantir a integridade)

ORDER BY 1, 2 – ordenar a consulta SQL, pelos 2 primeiros atributos



3. Comandos SQL: SELECT (com mais de 1 tabela)

Exemplo:

```
SELECT Departamento.Nome, Funcionario.Nome,  
        Email, Telefone  
  
FROM Departamento, Funcionario  
  
WHERE Departamento.Id_Departamento = Funcionario.Id_Departamento  
  
ORDER BY 1, 2
```




3. Comandos SQL: SELECT (com mais de 1 tabela)

PROTÓTIPO/postgres@PostgreSQL 13 ▾

Query Editor Query History

```
1 SELECT Departamento.Nome, Funcionario.Nome,  
2     Email, Telefone  
3 FROM Departamento, Funcionario  
4 WHERE Departamento.Id_Departamento = Funcionario.Id_Departamento  
5 ORDER BY 1, 2  
6
```

Data Output Messages

	nome character varying (100)	nome character varying (100)	email character varying (50)	telefone character varying (11)
1	Financeiro	William	william@gmail.com	65997970404
2	Gestão de Pessoas	João Marcos	joaomarcos@gmail.com	65988982525
3	Gestão de Pessoas	Lucas Adriano	lucasadriano@gmail.com	65998980202
4	Tecnologia da Informação	Jessica	jessica@gmail.com	65999892222



3. Comandos SQL: SELECT

(com mais de 1 tabela – usando o recurso AS - renomear)

PROTOTIPO/postgres@PostgreSQL 13 ▾

Query Editor Query History

```
1  SELECT Departamento.Nome AS "Nome do Depto", -- O comando AS renomeia um atributo
2      Funcionario.Nome AS "Nome do Funcionário",
3      Email AS "Email do Funcinário", Telefone
4  FROM Departamento, Funcionario
5  WHERE Departamento.Id_Departamento = Funcionario.Id_Departamento |
6  ORDER BY 1, 2
7
8
```

in

Data Output Messages

	Nome do Depto character varying (100) 🔒	Nome do Funcionário character varying (100) 🔒	Email do Funcinário character varying (50) 🔒	telefone character varying (11) 🔒
1	Gestão de Pessoas	João Marcos	joaomarcos@gmail.com	65988982525
2	Gestão de Pessoas	Lucas Adriano	lucasadriano@gmail.com	65998980202
3	Tecnologia da Informação	Jessica	jessica@gmail.com	65999892222



4. Atividade da semana

Fazer o *upload* de um arquivo contendo as operações de atualização (UPDATE) e exclusão (DELETE) de dados realizadas sobre os seguintes cenários:

=> Funcionario_Departamento;

- *update* em algum registro de departamento;
- *update* de um endereço de algum funcionario;
- *delete* em 1 registro de funcionario;
- depois faça a inserção de 1 novo registro em funcionario.
- desenvolva 1 consulta SQL utilizando 2 tabelas, bem como os comandos AS e ORDER BY.

=> Clinica_Medica:

- *update* em algum registro de medico, paciente e de consulta;
- *delete* em 2 registros de consulta;
- depois faça a inserção de 2 novos registros em consulta.
- desenvolva 2 consultas SQL: (1) utilizando 2 tabelas; (outra) utilizando 3 tabelas. Obs.: use os comandos AS e ORDER BY.



4. Atividade da semana (exemplo)

Exemplo:

SELECT xxxx, aaa – escolha os atributos que desejar



FROM Medico, Consulta, Paciente

WHERE Medico.Id_Medico = Consulta.Id_Medico -- atributo de ligação

AND Consulta.Id_Paciente = Paciente.Id_Paciente -- atributo de ligação

ORDER BY 1, 2



Obrigada!

Juliana Saragiotto Silva

Email: juliana.silva@cba.ifmt.edu.br