

Banco de Dados 2

8 – Segurança (Parte 2) e Visões (Views).

Segurança da Informação

- No **PostgreSQL**, a segurança na comunicação entre cliente (ex.: psql, pgAdmin, aplicações) e servidor é garantida principalmente através de **criptação de dados em trânsito**, usando **SSL/TLS**.
- Isso evita que **informações sensíveis** (logins, senhas, dados de tabelas) sejam capturadas por ataques de rede (como **sniffing** ou **man-in-the-middle**).

Segurança da Informação

- Quando falamos em **segurança de comunicação com o banco de dados**, dois ataques clássicos que a **encriptação via SSL/TLS** ajuda a evitar são o ***sniffing*** e o ***man-in-the-middle* (MITM)**.

Segurança da Informação

- **Sniffing** É a interceptação passiva do tráfego de rede.
- Um atacante usa um **software** (ex.: **Wireshark**, **tcpdump**) para “escutar” os pacotes que trafegam na rede.
- Se a **comunicação não estiver criptografada**, o atacante **consegue ver usuário, senha e dados em texto puro**.

Segurança da Informação

- Um usuário mal-intencionado na mesma rede que o servidor pode capturar pacotes e descobrir credenciais de login ou dados de tabelas.
- Com a **criptografia SSL/TLS**, mesmo que os pacotes sejam capturados, estarão ilegíveis sem a chave de decriptação.

Segurança da Informação

- **Man-in-the-Middle (MITM)** é quando o atacante se coloca entre o cliente e o servidor, interceptando e alterando a comunicação sem que nenhum dos lados perceba.
- O cliente acha que está falando com o servidor, mas na verdade está se conectando ao atacante.
- O servidor acha que está falando com o

Segurança da Informação

- **Durante uma conexão sem criptografia ou sem verificação de certificado, o atacante pode:**
 - interceptar credenciais de login;
 - alterar resultados de consultas;
 - modificar comandos enviados ao servidor (ex.: UPDATE saldo SET valor=0).

Segurança da Informação

- Defesa contra **Man-in-the-Middle (MITM)**:
- Usar **SSL/TLS** com **verificação de certificados (sslmode=verify-full)**.
- Isso garante que o cliente realmente está falando com o servidor legítimo, e não com um impostor.

Segurança da Informação

- O **postgresql.conf** é o arquivo principal de configuração do PostgreSQL.
- Ele não fica “dentro” do pgAdmin 4, mas sim no sistema operacional onde o **servidor PostgreSQL** está rodando.

Segurança da Informação

- O **PostgreSQL** pode estar em **Linux**, **Windows**, ou dentro de um **container Docker**.
- O caminho do arquivo varia.
- Para descobrir **o caminho exato no seu servidor**, você pode rodar no **Query Tool** do **pgAdmin 4** (ou **psql**):

Segurança da Informação

The screenshot shows the pgAdmin 4 interface. At the top, it displays the connection information: 'segurança/postgres@PostgreSQL 17'. Below the connection bar are standard database management icons. The main area is divided into two tabs: 'Query' (which is selected) and 'Query History'. In the 'Query' tab, there is a single line of SQL code: 'SHOW config_file;'. Below the query area are three tabs: 'Data Output', 'Messages', and 'Notifications'. Under 'Data Output', the results of the query are displayed in a table. The table has one row with the following data:

	config_file
1	text C:/Program Files/PostgreSQL/17/data/postgresql.conf

A lock icon is visible next to the column header 'config_file'. At the bottom of the pgAdmin window, there is a toolbar with several icons.

- Isso retorna o **caminho completo** do **postgresql.conf**.

Segurança da Informação

- Abra o **arquivo** com um **editor de texto** (dependendo do SO):
 - Linux: nano, vim ou gedit com privilégios de **administrador**.
 - Windows: abrir com **Notepad** ou **Notepad++** em modo **administrador**.
- Altere os parâmetros desejados (ex.: habilitar **SSL**).

Segurança da Informação

```
-- Habilitação do SSL:  
ssl = on  
ssl_cert_file = 'server.crt'  
ssl_key_file = 'server.key'
```

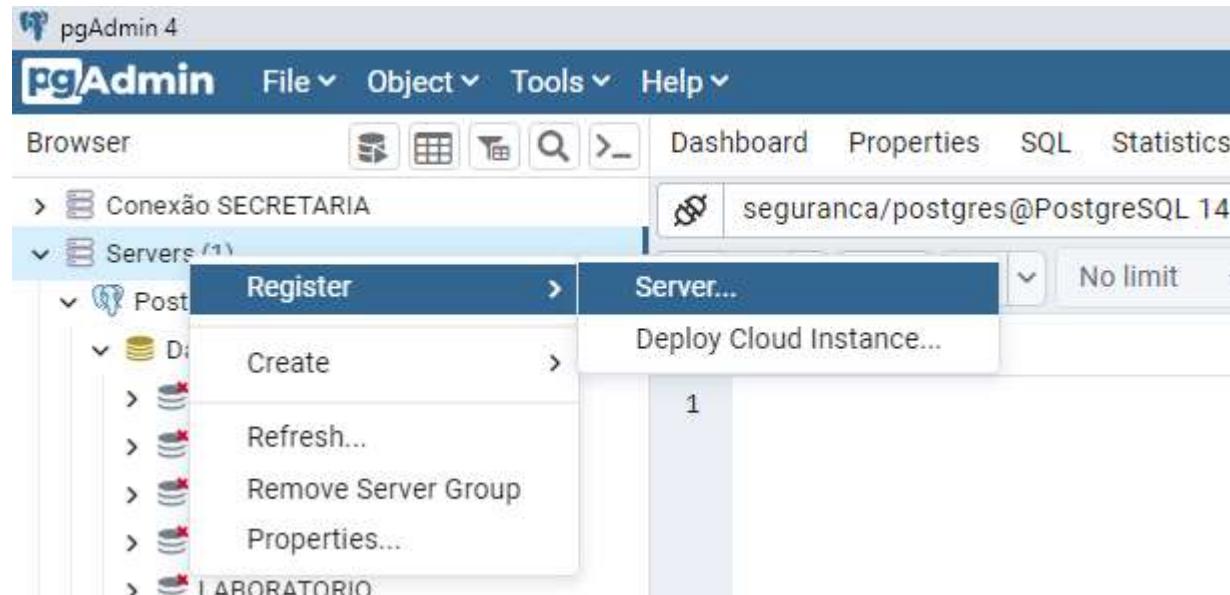
Após a alteração de
postgresql.conf é necessário
reiniciar o servidor do
PostgreSQL.

- **server.key** → é a chave privada do servidor.
- **server.crt** → é o certificado digital correspondente (pode ser autoassinado ou emitido por uma Autoridade Certificadora - CA).
- Eles não vêm prontos na instalação do **PostgreSQL**. Você precisa gerar ou obter esses arquivos.

Segurança da Informação

- Existe uma maneira alternativa de configurar algumas características do **postgresql.conf** sem manipular diretamente o próprio arquivo.

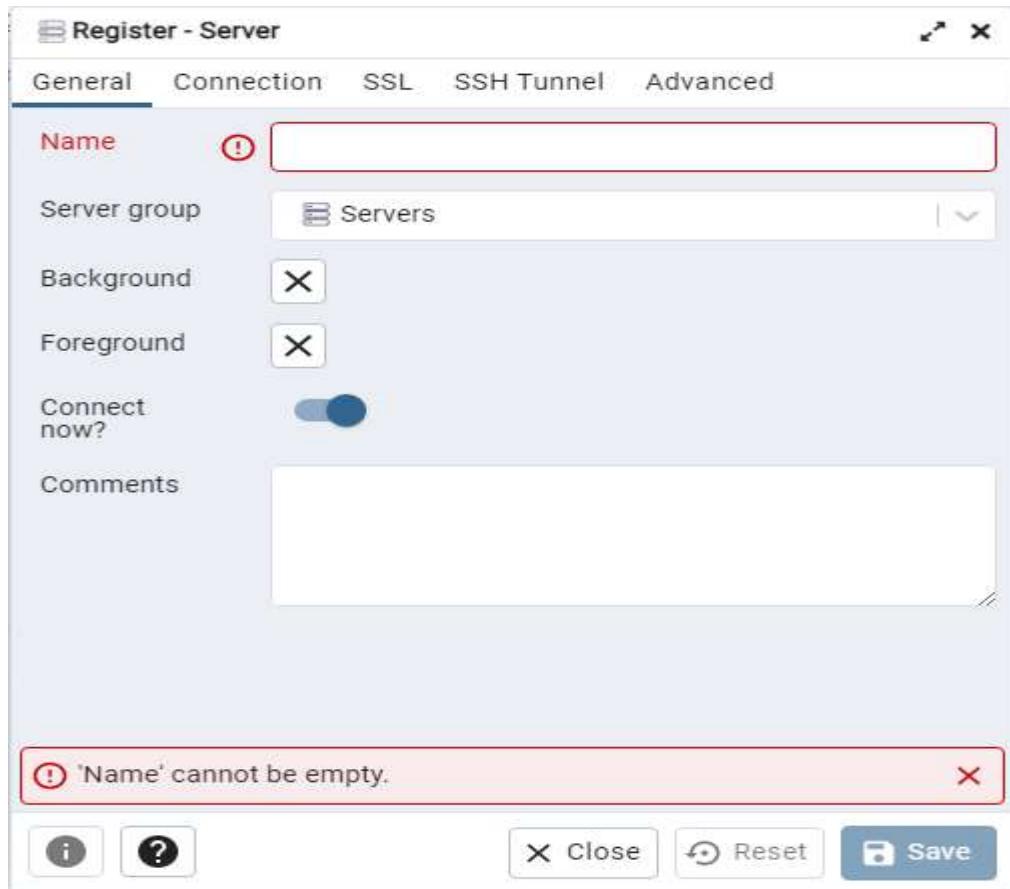
Segurança da Informação



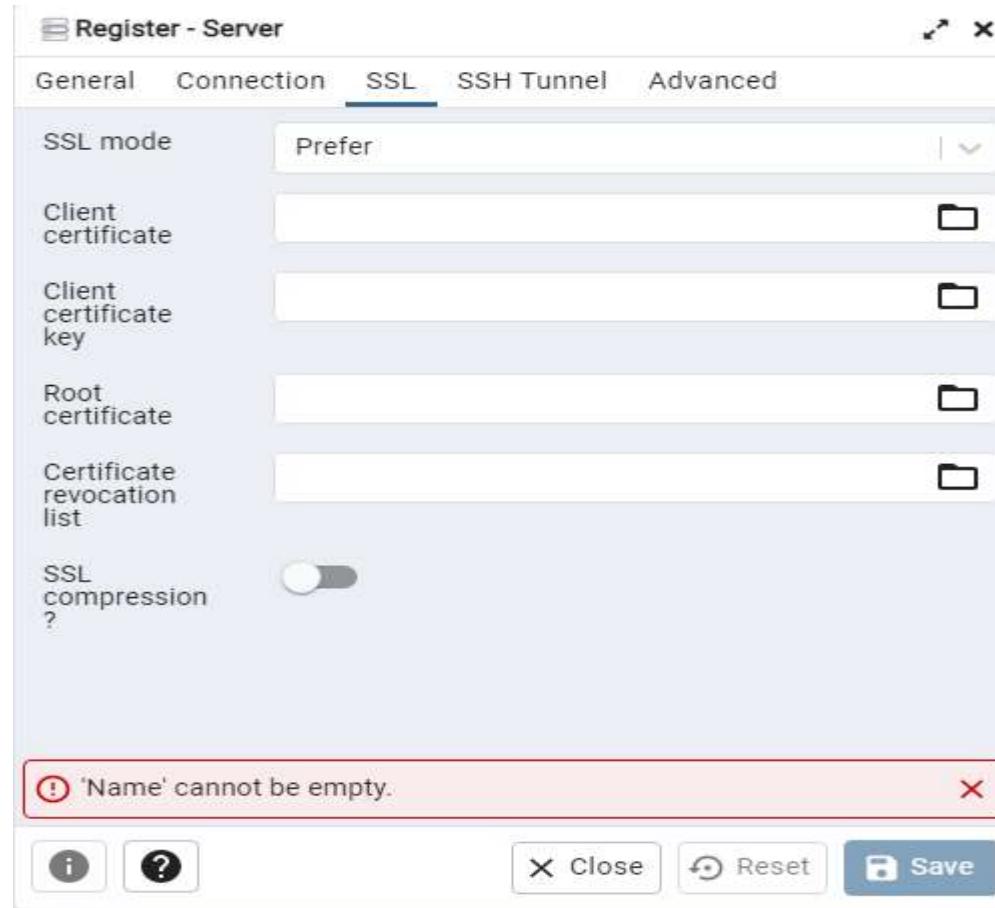
- Clique com o botão direito sobre o **Servidor**.
- Selecione **Register**.
- Selecione **Server...**

Segurança da Informação

- Clique na aba **SSL**.



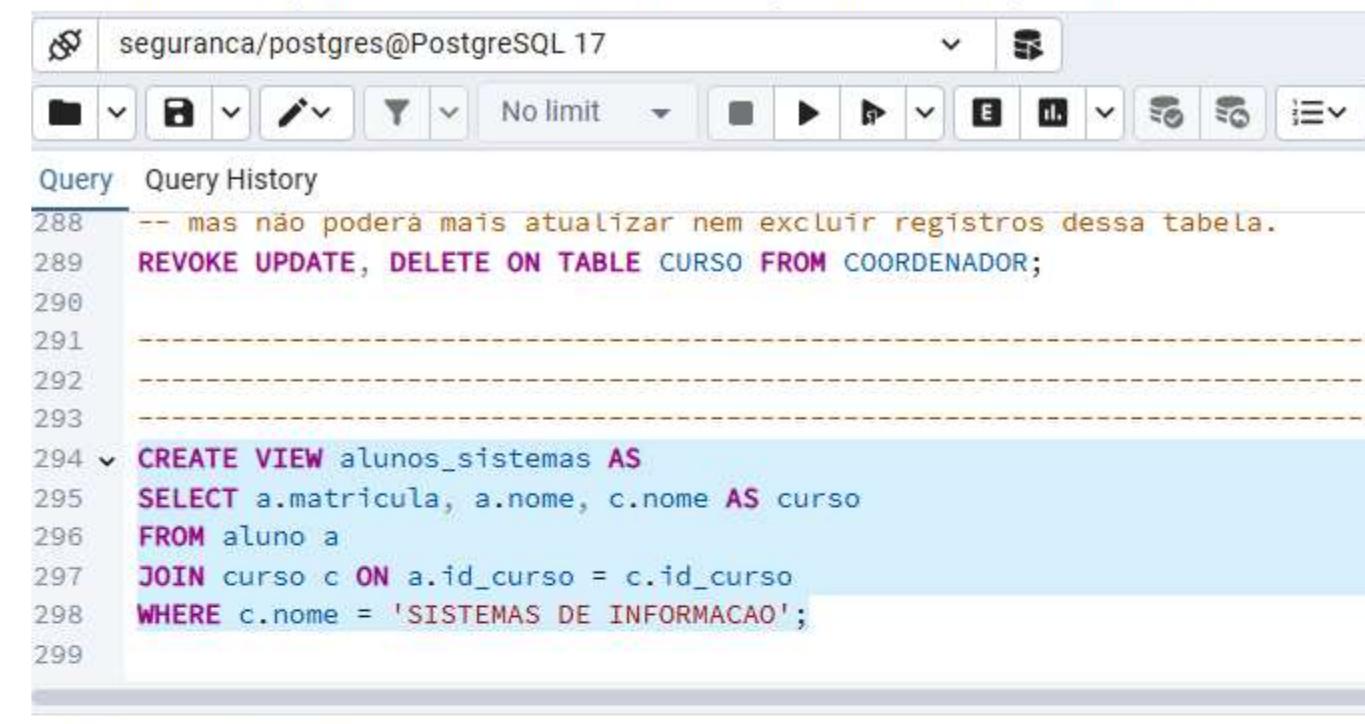
Segurança da Informação



Visão (View)

- Uma **VIEW** (ou *visão*) é um **objeto de banco de dados que armazena uma consulta SQL pré-definida**, permitindo que você a utilize como se fosse uma **tabela virtual**.
- A **VIEW não armazena dados fisicamente** (a não ser que seja ***materializada***, o que veremos em breve).
- Sempre que a **view** é **consultada**, o **PostgreSQL executa a consulta SQL definida por trás dela e retorna o resultado como se fosse uma tabela**.

Visão (View)



The screenshot shows the pgAdmin 4 interface with a PostgreSQL connection named 'segurança/postgres@PostgreSQL 17'. The query history tab is selected, displaying the following SQL code:

```
288 -- mas não poderá mais atualizar nem excluir registros dessa tabela.  
289 REVOKE UPDATE, DELETE ON TABLE CURSO FROM COORDENADOR;  
290  
291 -----  
292 -----  
293 -----  
294 CREATE VIEW alunos_sistemas AS  
295 SELECT a.matricula, a.nome, c.nome AS curso  
296 FROM aluno a  
297 JOIN curso c ON a.id_curso = c.id_curso  
298 WHERE c.nome = 'SISTEMAS DE INFORMAÇÃO';  
299
```

- A **criação** de uma **VIEW** dos **alunos matriculados no curso de SISTEMAS DE INFORMAÇÃO.**

CREATE VIEW

Query returned successfully in 136 msec.

Visão (View)

```
291
292
293
294 ✓ CREATE VIEW alunos_sistemas AS
295     SELECT a.matricula, a.nome, c.nome AS curso
296     FROM aluno a
297     JOIN curso c ON a.id_curso = c.id_curso
298     WHERE c.nome = 'SISTEMAS DE INFORMACAO';
299
300    SELECT * FROM ALUNO;
```

Data Output Messages Notifications



	matricula [PK] integer	nome character varying (45)	sexo character (1)	nascimento date	id_curso integer
1	2099	FRANCISCO BUARQUE DE HOLANDA	M	1992-09-22	200
2	2100	CAETANO VELOSO	M	1993-01-21	100
3	2101	MILTON NASCIMENTO	M	1994-04-03	100
4	2102	AGNETHA FALTSKOG	F	1995-05-01	300
5	2200	JOHN LENNON	M	1995-07-21	300
6	2300	ANNY FRID LYNGSTAD	F	1996-03-25	300

- Na **tabela ALUNO** há um total de **6 alunos distribuídos por 3 cursos diferentes: 100, 200 e 300.**

Visão (View)

```
293
294 ✓ CREATE VIEW alunos_sistemas AS
295     SELECT a.matricula, a.nome, c.nome AS curso
296     FROM aluno a
297     JOIN curso c ON a.id_curso = c.id_curso
298     WHERE c.nome = 'SISTEMAS DE INFORMACAO';
299
300     SELECT * FROM alunos_sistemas;
```

Data Output Messages Notifications

≡+ SQL

	matricula integer	nome character varying (45)	curso character varying (45)
1	2102	AGNETHA FALTSKOG	SISTEMAS DE INFORMACAO
2	2200	JOHN LENNON	SISTEMAS DE INFORMACAO
3	2300	ANNY FRID LYNGSTAD	SISTEMAS DE INFORMACAO

- Quando fazemos uma **consulta** a partir da **VIEW alunos_sistemas** encontramos **apenas 3 alunos em vez de 6.**
- **Todos matriculados em Sistemas de Informação.**

Visão (View)

```
294 ✓ CREATE VIEW alunos_sistemas AS
295   SELECT a.matricula, a.nome, c.nome AS curso
296   FROM aluno a
297   JOIN curso c ON a.id_curso = c.id_curso
298   WHERE c.nome = 'SISTEMAS DE INFORMACAO';
299
300   SELECT * FROM alunos_sistemas WHERE MATRICULA = 2200;
```

Data Output Messages Notifications

matricula nome curso

	integer	character varying (45)	character varying (45)
1	2200	JOHN LENNON	SISTEMAS DE INFORMACAO

- Podemos efetuar qualquer tipo de consulta comum sobre a VIEW como se fosse uma tabela de verdade.

Visão (View)

```
293
294 -+
294 ✓ CREATE VIEW alunos_sistemas AS
295   SELECT a.matricula, a.nome, c.nome AS curso
296   FROM aluno a
297   JOIN curso c ON a.id_curso = c.id_curso
298   WHERE c.nome = 'SISTEMAS DE INFORMACAO';
299
300   SELECT * FROM alunos_sistemas WHERE sexo = 'F';
```

Data Output Messages Notifications

ERROR: coluna "sexo" não existe
LINE 1: SELECT * FROM alunos_sistemas WHERE sexo = 'F';
^

ERRO: coluna "sexo" não existe
SQL state: 42703
Character: 37

- Logicamente não é possível filtrar os registros por SEXO.
- A coluna SEXO existe em ALUNO mas não foi utilizada na VIEW alunos_sistemas.

Visão (View)

```
293 -----  
294 ✓ CREATE VIEW alunos_sistemas AS  
295   SELECT a.matricula, a.nome, c.nome AS curso  
296   FROM aluno a  
297   JOIN curso c ON a.id_curso = c.id_curso  
298   WHERE c.nome = 'SISTEMAS DE INFORMACAO';  
299  
300 INSERT INTO ALUNOS_SISTEMAS VALUES (3000, 'AGNALDO RAYOL', 'SISTEMAS DE INFORMAÇÃO');
```

Data Output Messages Notifications

ERROR: não é possível inserir na visão "alunos_sistemas"

Visões que não selecionam de uma única tabela ou visão não são automaticamente atualizáveis.

ERRO: não é possível inserir na visão "alunos_sistemas"

SQL state: 55000

Detail: Visões que não selecionam de uma única tabela ou visão não são automaticamente atualizáveis.

Hint: Para ativar a inserção na visão, forneça um gatilho INSTEAD OF INSERT ou uma regra incondicional ON INSERT DO INSTEAD.

Visão (View)

- **Não foi possível inserir um novo registro através da VIEW pela seguinte razão:**
 - A VIEW não possui colunas como SEXO, ID_CURSO e NASCIMENTO (todas de preenchimento obrigatório).
 - Apesar de estarmos inserindo na VIEW o novo registro será inserido na tabela ALUNO.

Visão (View)

```
293 -----
294 ✓ CREATE VIEW alunos_sistemas AS
295   SELECT a.matricula, a.nome, c.nome AS curso
296   FROM aluno a
297   JOIN curso c ON a.id_curso = c.id_curso
298   WHERE c.nome = 'SISTEMAS DE INFORMACAO';
299
300 INSERT INTO ALUNOS_SISTEMAS VALUES (3000,'PAULA TOLLER','ARQUITETURA E URBANISMO');
301
```

Data Output Messages Notifications

ERROR: não é possível inserir na visão "alunos_sistemas"

Visões que não selecionam de uma única tabela ou visão não são automaticamente atualizáveis.

ERRO: não é possível inserir na visão "alunos_sistemas"

SQL state: 55000

Detail: Visões que não selecionam de uma única tabela ou visão não são automaticamente atualizáveis.

Hint: Para ativar a inserção na visão, forneça um gatilho INSTEAD OF INSERT ou uma regra incondicional ON INSERT DO INSTEAD.

Visão (View)

- Além das restrições anteriores essa tentativa de inserção de um novo registro em ALUNO não pode ocorrer pois a VIEW está restrita a alunos de SISTEMAS DE INFORMAÇÃO.
- Aqui tentou-se inserir uma nova aluna no curso de ARQUITETURA E URBANISMO.

Vantagens do Uso de Views

- **Simplificação de consultas:**
 - Consultas complexas com várias junções e filtros podem ser encapsuladas em uma *view*.
 - O usuário final só consulta a *view*, sem precisar repetir a query completa.
- **Reuso de consultas:**
 - Uma mesma lógica SQL pode ser usada por várias aplicações e relatórios.
 - Isso evita duplicação de código SQL.

Vantagens do Uso de Views

- **Segurança e controle de acesso**
 - Você pode dar permissão a um usuário para consultar apenas a *view*, sem acesso direto às tabelas originais.
 - Exemplo: permitir que a secretaria veja apenas alunos e notas aprovadas, sem expor a tabela inteira.
- **Independência lógica**
 - Se a estrutura das tabelas mudar, você pode ajustar a *view* sem alterar os programas que dependem dela.
 - Isso ajuda na **manutenção** e na **evolução** do banco.

Vantagens do Uso de Views

- **Consistência de dados**
 - Regras de negócio podem ser centralizadas em *views*.
 - Assim, todos os usuários recebem os dados já padronizados.

View

```
302  
303 ✓ CREATE OR REPLACE VIEW alunos_aprovados AS  
304  
SELECT  
305     a.matricula,  
306     a.nome AS aluno,  
307     d.nome AS disciplina,  
308     h.ano,  
309     h.semestre,  
310     h.nota,  
311     h.frequencia,  
312     c.descr_condicao  
313 FROM historico h  
314 JOIN aluno a  
      ON h.matricula = a.matricula  
316 JOIN disciplina d  
      ON h.id_disciplina = d.id_disciplina  
318 JOIN condicao c  
      ON h.condicao = c.id_condicao  
319 WHERE h.condicao = 'APR';  
320  
321
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 111 msec.

- A **visão** (**view**) **alunos_aprovados**.
- Qual a vantagem?
- **Simplifica a consulta por parte do usuário final** que não precisa efetuar os **JOINS** que lhe são **internos**.

View

```
302  
303 ✓ CREATE OR REPLACE VIEW alunos_aprovados AS  
304  
305     a.matricula,  
306     a.nome AS aluno,  
307     d.nome AS disciplina,  
308     h.ano,  
309     h.semestre,  
310     h.nota,  
311     h.frequencia,  
312     c.descr_condicao  
313 FROM historico h  
314 JOIN aluno a  
315     ON h.matricula = a.matricula  
316 JOIN disciplina d  
317     ON h.id_disciplina = d.id_disciplina  
318 JOIN condicao c  
319     ON h.condicao = c.id_condicao  
320 WHERE h.condicao = 'APR';  
321
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 111 msec.

- Existe ainda uma outra vantagem.
- O usuário final que tiver privilégios de acesso a essa view (mas não à tabela HISTORICO) somente poderá visualizar e manipular registros de alunos aprovados.

View e Segurança

The screenshot shows a PostgreSQL client interface with the following details:

- Connection:** segurança/postgres@PostgreSQL 17
- Toolbar:** Includes icons for file operations, search, and various database functions.
- Tab:** "Query" is selected, showing the query history.
- Code Area:** Displays the following SQL script:

```
1 --- CONTEXTO DE RECURSOS HUMANOS (SETEMBRO DE 2025)
2
3
4
5
6 ✓ CREATE TABLE SETOR (
7     ID_SETOR CHAR(4) NOT NULL PRIMARY KEY,
8     NOME_SETOR VARCHAR(40) NOT NULL
9 );
10
11 ✓ INSERT INTO SETOR VALUES ('REI','REITORIA'), ('NTI','NUCLEO DE TECNOLOGIA DA INFORMAÇÃO'),
12                               ('CSI','COORDENADORIA DE SISTEMAS DE INFORMACAO'),
13                               ('CCE','COORDENADORIA DE CIENCIAS ECONOMICAS');
14
15 ✓ CREATE TABLE FUNCAO (
16     ID_FUNCAO CHAR(4) NOT NULL PRIMARY KEY,
17     NOME_FUNCAO VARCHAR(40) NOT NULL
18 );
19
20 ✓ INSERT INTO FUNCAO VALUES ('PROF','PROFESSOR'),('REIT','REITOR'),('COOR','COORDENADOR'),
21     ('TECA','TECNICO ADMINISTRATIVO'),('GERE','GERENTE ADMINISTRATIVO');
```
- Data Output Tab:** Shows the result of the last query: "INSERT 0 5".
- Messages Tab:** Shows the message: "Query returned successfully in 83 msec."

View e Segurança

```
23 ✓ CREATE TABLE FUNCIONARIO (
24     MATRICULA SERIAL NOT NULL PRIMARY KEY,
25     NOME VARCHAR(45) NOT NULL,
26     SEXO CHAR(1) NOT NULL,
27     NASCIMENTO DATE NOT NULL,
28     DT_INGRESSO DATE NOT NULL,
29     ID_SETOR CHAR(4) NOT NULL,
30
31         FOREIGN KEY (ID_SETOR) REFERENCES SETOR (ID_SETOR) ON DELETE RESTRICT
32
33 );
34
35 ✓ INSERT INTO FUNCIONARIO (NOME, SEXO, NASCIMENTO, DT_INGRESSO, ID_SETOR)
36     VALUES ('CINTIA ANDRADE RIBEIRO','F','1970-03-21','1995-05-13','CSI'),
37     ('EDUARDO LICOLINI','M','1967-09-19','1992-09-23','CSI'),
38     ('RONALDO CARVALHO CAMPOS','M','1968-06-21','1995-02-14','CSI'),
39     ('CLAUDIA GONCALVES MACHADO','F','1969-06-13','1992-09-22','CCE'),
40     ('ANA PAULA XAVIER DE ABREU','F','1967-08-29','1997-10-03','CCE'),
41     ('ANA PAULA LASCASAS','F','1968-06-10','1999-04-03','CCE'),
42     ('HUMBERTO TAVARES NOGUEIRA','M','1969-09-12','2000-06-06','CCE'),
43     ('THORSTEN BERG','M','1970-01-21','2000-01-02','CSI'),
44     ('LEONARDO PEREIRA','M','1971-01-21','2001-07-21','NTI'),
45     ('RENATA CRISTINA LARANJA','F','1970-06-23','2002-03-28','NTI'),
46     ('FABRICIA PARAGUACU','F','1969-12-03','2008-07-25','REI');
47
```

Data Output Messages Notifications

INSERT 0 11

Query returned successfully in 163 msec.

View e Segurança

```
47
48 ✓ CREATE TABLE HISTORICO_FUNCIONAL (
49     MATRICULA_FUNC INTEGER NOT NULL,
50     ID_SETOR CHAR(4) NOT NULL,
51     ID_FUNCAO CHAR(4) NOT NULL,
52     DT_INICIAL DATE NOT NULL,
53     DT_FINAL DATE,
54
55     PRIMARY KEY(MATRICULA_FUNC, ID_SETOR, ID_FUNCAO, DT_INICIAL),
56
57     FOREIGN KEY (MATRICULA_FUNC) REFERENCES FUNCIONARIO (MATRICULA)
58     ON DELETE RESTRICT,
59
60     FOREIGN KEY (ID_SETOR) REFERENCES SETOR (ID_SETOR)
61     ON DELETE RESTRICT,
62
63     FOREIGN KEY (ID_FUNCAO) REFERENCES FUNCAO (ID_FUNCAO)
64     ON DELETE RESTRICT
65
66 );
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 76 msec.

View e Segurança

```
67
68 ✓ INSERT INTO FUNCIONARIO (NOME, SEXO, NASCIMENTO, DT_INGRESSO, ID_SETOR)
69   VALUES ('JOSE MAGALHAES NETO','M','1969-10-12','1995-05-13','CCE');
70
71 ✓ INSERT INTO HISTORICO_FUNCIONAL VALUES
72   (2,'CSI','PROF','1992-09-23', NULL),
73   (2,'CSI','COOR','2002-01-02', '2005-09-01'),
74   (2,'REI','REIT','2005-9-02', '2009-09-01'),
75   (4,'CCE','PROF','1992-09-22', NULL),
76   (1,'CSI','PROF','1995-05-13', NULL),
77   (1,'REI','REIT','2009-09-02','2017-09-01'),
78   (3,'CSI','PROF','1995-02-14', NULL),
79   (5,'CCE','PROF','1997-10-03', NULL),
80   (6,'CCE','PROF','1999-04-03', '2025-01-30'),
81   (6,'CCE','COOR','2003-01-02', '2017-09-01'),
82   (6,'REI','REIT','2017-09-02', '2025-01-30'),
83   (7,'CCE','PROF','2000-06-06', NULL),
84   (8,'CSI','PROF','2000-01-02', NULL),
85   (9,'NTI','TECA','2001-07-21', NULL),
86   (9,'NTI','GERE','2018-01-02', NULL),
87   (10,'NTI','TECA','2002-02-28', NULL),
88   (11,'REI','TECA','2008-07-25', NULL),
89   (12,'CCE','PROF','1995-05-13', NULL);
90
```

Data Output Messages Notifications

INSERT 0 18

Query returned successfully in 110 msec.

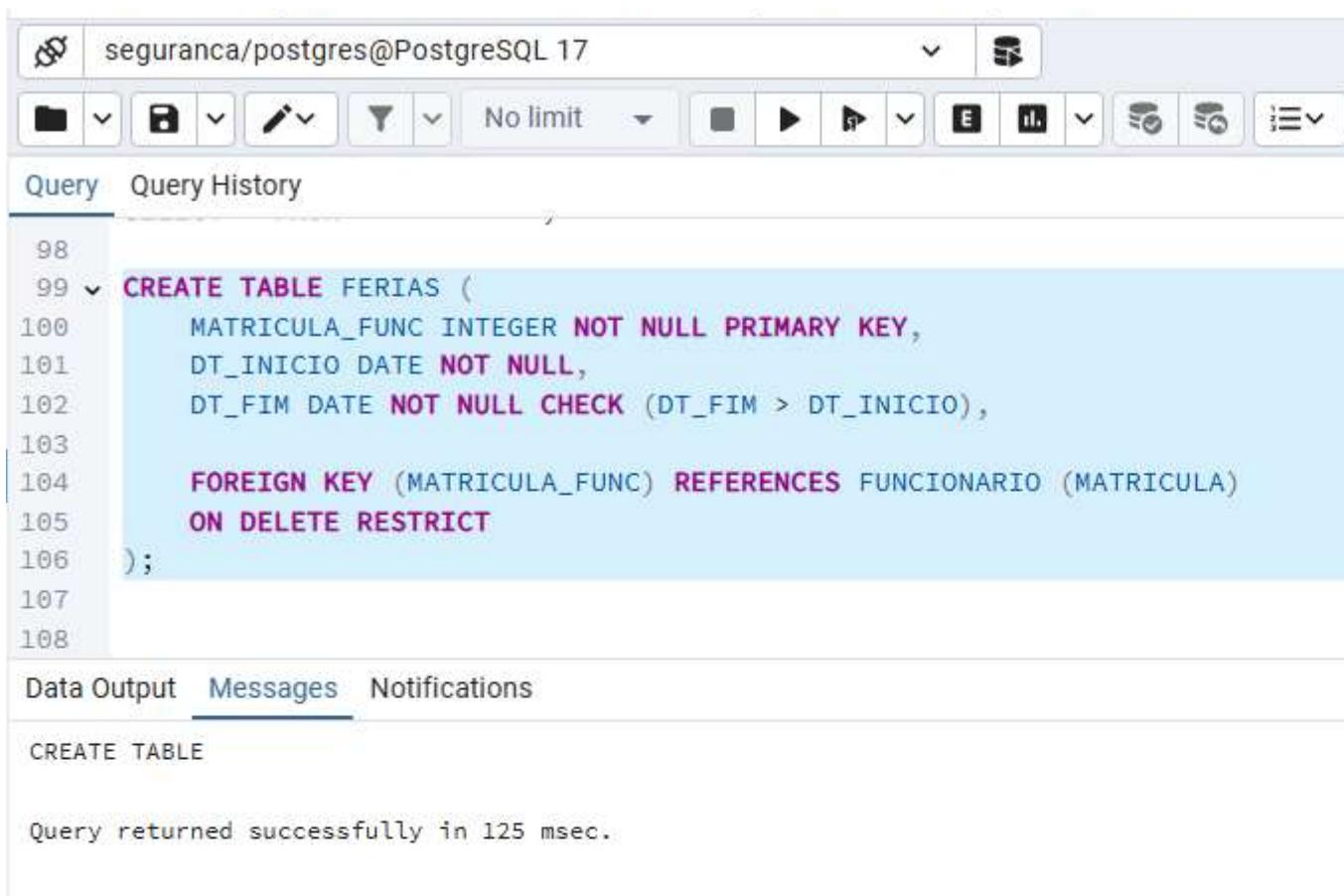
View e Segurança

```
91  
92 ALTER TABLE FUNCIONARIO ADD COLUMN SENHA VARCHAR(06);  
93  
94 UPDATE FUNCIONARIO SET SENHA = '123456' WHERE SEXO = 'M';  
95 UPDATE FUNCIONARIO SET SENHA = '456789' WHERE SEXO = 'F';  
96  
97 SELECT * FROM FUNCIONARIO;  
98
```

Data Output Messages Notifications

	matricula [PK] integer	nome character varying (45)	sexo character (1)	nascimento date	dt_ingresso date	id_setor character (4)	senha character varying (6)
1	2	EDUARDO LICOLINI	M	1967-09-19	1992-09-23	CSI	123456
2	3	RONALDO CARVALHO CAMP...	M	1968-06-21	1995-02-14	CSI	123456
3	7	HUMBERTO TAVARES NOGU...	M	1969-09-12	2000-06-06	CCE	123456
4	8	THORSTEN BERG	M	1970-01-21	2000-01-02	CSI	123456
5	9	LEONARDO PEREIRA	M	1971-01-21	2001-07-21	NTI	123456
6	12	JOSE MAGALHAES NETO	M	1969-10-12	1995-05-13	CCE	123456
7	1	CINTIA ANDRADE RIBEIRO	F	1970-03-21	1995-05-13	CSI	456789
8	4	CLAUDIA GONCALVES MAC...	F	1969-06-13	1992-09-22	CCE	456789
9	5	ANA PAULA XAVIER DE ABREU	F	1967-08-29	1997-10-03	CCE	456789
10	6	ANA PAULA LASCASAS	F	1968-06-10	1999-04-03	CCE	456789
11	10	RENATA CRISTINA LARANJA	F	1970-06-23	2002-03-28	NTI	456789
12	11	FABRICIA PARAGUACU	F	1969-12-03	2008-07-25	REI	456789

View e Segurança



The screenshot shows a PostgreSQL client interface with the following details:

- Connection:** seguranca/postgres@PostgreSQL 17
- Toolbar:** Includes icons for file operations, search, and various database management functions.
- Query Tab:** Active tab, showing the SQL code for creating the 'FERIAS' table.
- Code:**

```
98
99 CREATE TABLE FERIAS (
100     MATRICULA_FUNC INTEGER NOT NULL PRIMARY KEY,
101     DT_INICIO DATE NOT NULL,
102     DT_FIM DATE NOT NULL CHECK (DT_FIM > DT_INICIO),
103
104     FOREIGN KEY (MATRICULA_FUNC) REFERENCES FUNCIONARIO (MATRICULA)
105     ON DELETE RESTRICT
106 );
107
108
```
- Data Output Tab:** Shows the command "CREATE TABLE".
- Messages Tab:** Active tab, showing the message "Query returned successfully in 125 msec."
- Notifications Tab:** Shows a blank area.

```
109  
110 -- View para REITORIA  
111 ✓ CREATE OR REPLACE VIEW vw_funcionarios_reitoria AS  
112   SELECT MATRICULA, NOME, SEXO, NASCIMENTO, DT_INGRESSO, ID_SETOR  
113   FROM FUNCIONARIO  
114   WHERE ID_SETOR = 'REI';  
115  
116 -- View para NTI  
117 ✓ CREATE OR REPLACE VIEW vw_funcionarios_nti AS  
118   SELECT MATRICULA, NOME, SEXO, NASCIMENTO, DT_INGRESSO, ID_SETOR  
119   FROM FUNCIONARIO  
120   WHERE ID_SETOR = 'NTI';  
121  
122 -- View para CSI  
123 ✓ CREATE OR REPLACE VIEW vw_funcionarios_csi AS  
124   SELECT MATRICULA, NOME, SEXO, NASCIMENTO, DT_INGRESSO, ID_SETOR  
125   FROM FUNCIONARIO  
126   WHERE ID_SETOR = 'CSI';  
127  
128 -- View para CCE  
129 ✓ CREATE OR REPLACE VIEW vw_funcionarios_cce AS  
130   SELECT MATRICULA, NOME, SEXO, NASCIMENTO, DT_INGRESSO, ID_SETOR  
131   FROM FUNCIONARIO  
132   WHERE ID_SETOR = 'CCE';  
133
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 83 msec.

Criamos uma VIEW para cada SETOR existente.

Nela podemos visualizar todas as colunas existentes na tabela FUNCIONARIO, exceto a coluna SENHA.

Cada SETOR somente pode “enxergar” os funcionários de seu próprio setor.

View e Segurança

```
524 -- REITORIA
525 ✓ CREATE OR REPLACE VIEW vw_ferias_reitoria AS
526   SELECT f.matricula_func, f.dt_inicio, f.dt_fim
527   FROM ferias f
528   JOIN funcionario fu ON fu.matricula = f.matricula_func
529   WHERE fu.id_setor = 'REI';
530
531 -- NTI
532 ✓ CREATE OR REPLACE VIEW vw_ferias_nti AS
533   SELECT f.matricula_func, f.dt_inicio, f.dt_fim
534   FROM ferias f
535   JOIN funcionario fu ON fu.matricula = f.matricula_func
536   WHERE fu.id_setor = 'NTI';
537
```

- Também criamos uma VIEW sobre a TABELA FÉRIAS para cada SETOR existente.

View e Segurança

```
537
538    -- CSI
539 ✓ CREATE OR REPLACE VIEW vw_ferias_csi AS
540     SELECT f.matricula_func, f.dt_inicio, f.dt_fim
541     FROM ferias f
542     JOIN funcionario fu ON fu.matricula = f.matricula_func
543     WHERE fu.id_setor = 'CSI';
544
545    -- CCE
546 ✓ CREATE OR REPLACE VIEW vw_ferias_cce AS
547     SELECT f.matricula_func, f.dt_inicio, f.dt_fim
548     FROM ferias f
549     JOIN funcionario fu ON fu.matricula = f.matricula_func
550     WHERE fu.id_setor = 'CCE';
551
552
```

View e Segurança

```
544  
545 -- CCE  
546 ✓ CREATE OR REPLACE VIEW vw_ferias_cce AS  
547   SELECT f.matricula_func, f.dt_inicio, f.dt_fim  
548   FROM ferias f  
549   JOIN funcionario fu ON fu.matricula = f.matricula_func  
550   WHERE fu.id_setor = 'CCE';  
551  
552  
553 INSERT INTO VW_FERIAS_CSI VALUES (1,'2025-08-29','2025-09-27');  
554
```

Data Output Messages Notifications

ERROR: não é possível inserir na visão "vw_ferias_csi"

Visões que não selecionam de uma única tabela ou visão não são automaticamente atualizáveis.

ERRO: não é possível inserir na visão "vw_ferias_csi"

SQL state: 55000

Detail: Visões que não selecionam de uma única tabela ou visão não são automaticamente atualizáveis.

Hint: Para ativar a inserção na visão, forneça um gatilho INSTEAD OF INSERT ou uma regra incondicional ON INSERT DO INSTEAD.

- Perceba que não foi possível fazer um INSERT diretamente através da VIEW.

```

552
553 INSERT INTO VW_FERIAS_CSI VALUES (1,'2025-08-29','2025-09-27');
554
555 -- INSERT em vw_ferias_reitoria
556 CREATE OR REPLACE RULE vw_ferias_reitoria_insert AS
557 ON INSERT TO vw_ferias_reitoria DO INSTEAD
558 INSERT INTO ferias (matricula_func, dt_inicio, dt_fim)
559 VALUES (NEW.matricula_func, NEW.dt_inicio, NEW.dt_fim);
560
561 -- UPDATE em vw_ferias_reitoria
562 CREATE OR REPLACE RULE vw_ferias_reitoria_update AS
563 ON UPDATE TO vw_ferias_reitoria DO INSTEAD
564 UPDATE ferias
565 SET dt_inicio = NEW.dt_inicio,
566     dt_fim = NEW.dt_fim
567 WHERE matricula_func = OLD.matricula_func
568     AND dt_inicio = OLD.dt_inicio
569     AND dt_fim = OLD.dt_fim;
570
571 -- DELETE em vw_ferias_reitoria
572 CREATE OR REPLACE RULE vw_ferias_reitoria_delete AS
573 ON DELETE TO vw_ferias_reitoria DO INSTEAD
574 DELETE FROM ferias
575 WHERE matricula_func = OLD.matricula_func
576     AND dt_inicio = OLD.dt_inicio
577     AND dt_fim = OLD.dt_fim;
578

```

Data Output Messages Notifications

CREATE RULE

Query returned successfully in 142 msec.

- Agora criamos RULES para cada view, de forma que quando você inserir/alterar/excluir por elas, o PostgreSQL redirecione a operação para a tabela FERIAS.
- Você repetiria o mesmo para as views vw_ferias_nti, vw_ferias_csi e vw_ferias_cce.

```
-- 582 -- INSERT em vw_ferias_csi
-- 583 ✓ CREATE OR REPLACE RULE vw_ferias_csi_insert AS
-- 584 ON INSERT TO vw_ferias_csi DO INSTEAD
-- 585 INSERT INTO ferias (matricula_func, dt_inicio, dt_fim)
-- 586 VALUES (NEW.matricula_func, NEW.dt_inicio, NEW.dt_fim);
-- 587
-- 588 -- UPDATE em vw_ferias_csi
-- 589 ✓ CREATE OR REPLACE RULE vw_ferias_csi_update AS
-- 590 ON UPDATE TO vw_ferias_csi DO INSTEAD
-- 591 UPDATE ferias
-- 592 SET dt_inicio = NEW.dt_inicio,
-- 593         dt_fim = NEW.dt_fim
-- 594 WHERE matricula_func = OLD.matricula_func
-- 595     AND dt_inicio = OLD.dt_inicio
-- 596     AND dt_fim = OLD.dt_fim;
-- 597
-- 598 -- DELETE em vw_ferias_csi
-- 599 ✓ CREATE OR REPLACE RULE vw_ferias_csi_delete AS
-- 600 ON DELETE TO vw_ferias_csi DO INSTEAD
-- 601 DELETE FROM ferias
-- 602 WHERE matricula_func = OLD.matricula_func
-- 603     AND dt_inicio = OLD.dt_inicio
-- 604     AND dt_fim = OLD.dt_fim;
```

Data Output Messages Notifications

CREATE RULE

Query returned successfully in 73 msec.

View e Segurança

```
608  
609  
610 INSERT INTO VW_FERIAS_CSI VALUES (1,'2025-08-29','2025-09-27');
```

Data Output Messages Notifications

```
INSERT 0 1
```

```
Query returned successfully in 73 msec.
```

- Agora foi possível INSERIR um registro em FÉRIAS através da VIEW VW_FERIAS_CSI.

View e Segurança

Query Query History

```
1
2 CREATE ROLE COORDENADOR_CSI;
3
4 -- Revogar todos os privilégios sobre todas as tabelas no schema 'public' para o papel 'COORDENADOR_CSI'
5 REVOKE ALL PRIVILEGES ON ALL TABLES IN SCHEMA public FROM COORDENADOR_CSI;
6
7 GRANT ALL PRIVILEGES ON vw_funcionarios_csi, vw_ferias_csi TO COORDENADOR_CSI;
8
9 CREATE ROLE CINTIA_ANDRADE LOGIN PASSWORD '123456' IN ROLE COORDENADOR_CSI;
10
11 SET ROLE CINTIA_ANDRADE;
12
13 SELECT * FROM FUNCIONARIO;
```

Data Output Messages Notifications

ERROR: permissão negada para tabela funcionario

ERRO: permissão negada para tabela funcionario

SQL state: 42501

View e Segurança

```
2 CREATE ROLE COORDENADOR_CSI;
3
4 -- Revogar todos os privilégios sobre todas as tabelas no schema 'public' para o papel 'COORDENADOR_CSI'
5 REVOKE ALL PRIVILEGES ON ALL TABLES IN SCHEMA public FROM COORDENADOR_CSI;
6
7 GRANT ALL PRIVILEGES ON vw_funcionarios_csi, vw_ferias_csi TO COORDENADOR_CSI;
8
9 CREATE ROLE CINTIA_ANDRADE LOGIN PASSWORD '123456' IN ROLE COORDENADOR_CSI;
10
11 SET ROLE CINTIA_ANDRADE;
12
13 SELECT * FROM FUNCIONARIO;
14
15 SELECT * FROM VW_FUNCIONARIOS_CSI;
16
```

Data Output Messages Notifications

The screenshot shows a PostgreSQL client interface with a toolbar at the top containing icons for file operations, a dropdown menu, and a SQL tab. Below the toolbar is a table with eight columns. The first column contains row numbers (1, 2, 3, 4). The second column is labeled 'matricula' with type 'integer'. The third column is labeled 'nome' with type 'character varying (45)'. The fourth column is labeled 'sexo' with type 'character (1)'. The fifth column is labeled 'nascimento' with type 'date'. The sixth column is labeled 'dt_ingresso' with type 'date'. The seventh column is labeled 'id_setor' with type 'character (4)'. The eighth column has a lock icon. The table displays four rows of data:

	matricula	nome	sexo	nascimento	dt_ingresso	id_setor	
1	2	EDUARDO LICOLINI	M	1967-09-19	1992-09-23	CSI	
2	3	RONALDO CARVALHO CAMPOS	M	1968-06-21	1995-02-14	CSI	
3	8	THORSTEN BERG	M	1970-01-21	2000-01-02	CSI	
4	1	CINTIA ANDRADE RIBEIRO	F	1970-03-21	1995-05-13	CSI	

View e Segurança

```
7 GRANT ALL PRIVILEGES ON vw_funcionarios_csi, vw_ferias_csi TO COORDENADOR_CSI;
8
9 CREATE ROLE CINTIA_ANDRADE LOGIN PASSWORD '123456' IN ROLE COORDENADOR_CSI;
10
11 SET ROLE CINTIA_ANDRADE;
12
13 SELECT * FROM FUNCIONARIO;
14
15 SELECT * FROM VW_FUNCIONARIOS_CSI;
16
17 SELECT * FROM VW_FERIAS_CSI;
18
```

Data Output Messages Notifications

	matricula_func integer	dt_inicio date	dt_fim date
1	1	2025-08-29	2025-09-27

View e Segurança

```
8
9  CREATE ROLE CINTIA_ANDRADE LOGIN PASSWORD '123456' IN ROLE COORDENADOR_CSI;
10
11 SET ROLE CINTIA_ANDRADE;
12
13 SELECT * FROM FUNCIONARIO;
14
15 SELECT * FROM VW_FUNCIONARIOS_CSI;
16
17 SELECT * FROM VW_FERIAS_CSI;
18
19 SELECT FROM FERIAS;
20
```

Data Output Messages Notifications

ERROR: permissão negada para tabela ferias

ERRO: permissão negada para tabela ferias

SQL state: 42501

View e Segurança

```
7 GRANT ALL PRIVILEGES ON vw_funcionarios_csi, vw_ferias_csi TO COORDENADOR_CSI;
8
9 CREATE ROLE CINTIA_ANDRADE LOGIN PASSWORD '123456' IN ROLE COORDENADOR_CSI;
10
11 SET ROLE CINTIA_ANDRADE;
12
13 SELECT * FROM FUNCIONARIO;
14
15 SELECT * FROM VW_FUNCIONARIOS_CSI;
16
17 SELECT * FROM VW_FERIAS_CSI;
18
19 SELECT FROM FERIAS;
20
21 UPDATE VW_FERIAS_CSI SET DT_FIM = '2025-09-13' WHERE MATRICULA_FUNC = 1 AND DT_INICIO = '2025-08-29';
```

Data Output Messages Notifications

UPDATE 1

Query returned successfully in 107 msec.

View e Segurança

```
11 SET ROLE CINTIA_ANDRADE;
12
13 SELECT * FROM FUNCIONARIO;
14
15 SELECT * FROM VW_FUNCIONARIOS_CSI;
16
17 SELECT * FROM VW_FERIAS_CSI;
18
19 SELECT FROM FERIAS;
20
21 UPDATE VW_FERIAS_CSI SET DT_FIM = '2025-09-13' WHERE MATRICULA_FUNC = 1 AND DT_INICIO = '2025-08-29';
22
23
```

Data Output Messages Notifications



	matricula_func integer	dt_inicio date	dt_fim date
1	1	2025-08-29	2025-09-13

View e Segurança

```
10  
11 SET ROLE CINTIA_ANDRADE;  
12  
13 SELECT * FROM FUNCIONARIO;  
14  
15 SELECT * FROM VW_FUNCIONARIOS_CSI;  
16  
17 SELECT * FROM VW_FERIAS_CSI;  
18  
19 SELECT FROM FERIAS;  
20  
21 UPDATE VW_FERIAS_CSI SET DT_FIM = '2025-09-13' WHERE MATRICULA_FUNC = 1 AND DT_INICIO = '2025-08-29';  
22  
23 RESET ROLE;
```

Data Output Messages Notifications

RESET

Query returned successfully in 295 msec.

VIEW MATERIALIZADA

- As **views materializadas** são um tipo especial de **view** no **PostgreSQL** e em outros bancos de dados.
- Enquanto as **views tradicionais** são **dinâmicas**, ou seja, recalculadas toda vez que você as consulta, as **views materializadas** são **armazenadas fisicamente** e podem ser **atualizadas periodicamente**.

VIEW MATERIALIZADA

- Uma **view materializada** é uma tabela **que contém os resultados de uma consulta SQL, armazenados fisicamente no banco de dados.**
- Ao contrário das views convencionais, que são calculadas em tempo real, uma **view materializada armazena o resultado da consulta em disco, proporcionando melhor desempenho para consultas complexas ou com grandes volumes de dados.**

VIEW MATERIALIZADA

Vantagens das Views Materializadas

Desempenho:

- Como o resultado da consulta é armazenado, consultas subsequentes são muito mais rápidas, pois não precisam recalcular o conjunto de dados.
- Muito útil em cenários com consultas complexas ou agregações pesadas, onde o tempo de resposta pode ser um problema.

Redução de Carga no Banco:

- Em vez de recalcular os dados toda vez que a view é acessada, a view materializada reduz a carga no banco de dados, especialmente em ambientes com alto volume de consultas.

Cache de Resultados:

- Funciona como um **cache persistente** dos dados, tornando a visualização de resultados mais eficiente.

VIEW MATERIALIZADA

Desvantagens

Desatualização de Dados:

- A view materializada precisa ser **atualizada manualmente** (ou automaticamente em intervalos) para refletir mudanças nas tabelas subjacentes. Caso contrário, ela pode se tornar **obsoleta**.

Espaço de Armazenamento:

- Como os resultados são armazenados fisicamente, isso ocupa espaço adicional no banco de dados.

Manutenção:

- É necessário implementar a lógica para atualizar a view materializada quando os dados subjacentes mudam, o que pode adicionar complexidade.

VIEW MATERIALIZADA

```
22
23  RESET ROLE;
24
25 ✓ CREATE MATERIALIZED VIEW mv_funcionarios_ativos AS
26    SELECT MATRICULA, NOME, SEXO, NASCIMENTO, DT_INGRESSO, ID_SETOR
27    FROM FUNCIONARIO;
28
```

Data Output Messages Notifications

SELECT 12

Query returned successfully in 201 msec.

VIEW MATERIALIZADA

```
24
25 ✓ CREATE MATERIALIZED VIEW mv_funcionarios_ativos AS
26   SELECT MATRICULA, NOME, SEXO, NASCIMENTO, DT_INGRESSO, ID_SETOR
27   FROM FUNCIONARIO;
28
29 ✓ INSERT INTO FUNCIONARIO (NOME, SEXO, NASCIMENTO, DT_INGRESSO, ID_SETOR)
30   VALUES ('ASDRUBAL RIBEIRO SOARES','M','1956-04-03','1979-01-02','REI');
31
```

Data Output Messages Notifications

INSERT 0 1

Query returned successfully in 79 msec.

```

22
23  RESET ROLE;
24
25 ✓ CREATE MATERIALIZED VIEW mv_funcionarios_ativos AS
26   SELECT MATRICULA, NOME, SEXO, NASCIMENTO, DT_INGRESSO, ID_SETOR
27   FROM FUNCIONARIO;
28
29 ✓ INSERT INTO FUNCIONARIO (NOME, SEXO, NASCIMENTO, DT_INGRESSO, ID_SETOR)
30   VALUES ('ASDRUBAL RIBEIRO SOARES', 'M', '1956-04-03', '1979-01-02', 'REI');
31
32  SELECT * FROM FUNCIONARIO;

```

Data Output Messages Notifications

	matricula [PK] integer	nome character varying (45)	sexo character (1)	nascimento date	dt_ingresso date	id_setor character (4)	senha character varying (6)
1	2	EDUARDO LICOLINI	M	1967-09-19	1992-09-23	CSI	123456
2	3	RONALDO CARVALHO CAMP...	M	1968-06-21	1995-02-14	CSI	123456
3	7	HUMBERTO TAVARES NOGU...	M	1969-09-12	2000-06-06	CCE	123456
4	8	THORSTEN BERG	M	1970-01-21	2000-01-02	CSI	123456
5	9	LEONARDO PEREIRA	M	1971-01-21	2001-07-21	NTI	123456
6	12	JOSE MAGALHAES NETO	M	1969-10-12	1995-05-13	CCE	123456
7	1	CINTIA ANDRADE RIBEIRO	F	1970-03-21	1995-05-13	CSI	456789
8	4	CLAUDIA GONCALVES MAC...	F	1969-06-13	1992-09-22	CCE	456789
9	5	ANA PAULA XAVIER DE ABREU	F	1967-08-29	1997-10-03	CCE	456789
10	6	ANA PAULA LASCASAS	F	1968-06-10	1999-04-03	CCE	456789
11	10	RENATA CRISTINA LARANJA	F	1970-06-23	2002-03-28	NTI	456789
12	11	FABRICIA PARAGUACU	F	1969-12-03	2008-07-25	REI	456789
13	15	ASDRUBAL RIBEIRO SOARES	M	1956-04-03	1979-01-02	REI	[null]

VIEW MATERIALIZADA

- O novo registro de FUNCIONARIO para “Asdrúbal Soares Ribeiro” foi inserido com sucesso na tabela.
- Todavia, não aparece na VIEW MATERIALIZADA.

```

24
25 ✓ CREATE MATERIALIZED VIEW mv_funcionarios_ativos AS
26   SELECT MATRICULA, NOME, SEXO, NASCIMENTO, DT_INGRESSO, ID_SETOR
27   FROM FUNCIONARIO;
28
29 ✓ INSERT INTO FUNCIONARIO (NOME, SEXO, NASCIMENTO, DT_INGRESSO, ID_SETOR)
30   VALUES ('ASDRUBAL RIBEIRO SOARES', 'M', '1956-04-03', '1979-01-02', 'REI');
31
32   SELECT * FROM FUNCIONARIO;
33
34   SELECT * FROM MV_FUNCIONARIOS_ATIVOS;

```

Data Output Messages Notifications

	matricula integer	nome character varying (45)	sexo character (1)	nascimento date	dt_ingresso date	id_setor character (4)
1	2	EDUARDO LICOLINI	M	1967-09-19	1992-09-23	CSI
2	3	RONALDO CARVALHO CAMP...	M	1968-06-21	1995-02-14	CSI
3	7	HUMBERTO TAVARES NOGU...	M	1969-09-12	2000-06-06	CCE
4	8	THORSTEN BERG	M	1970-01-21	2000-01-02	CSI
5	9	LEONARDO PEREIRA	M	1971-01-21	2001-07-21	NTI
6	12	JOSE MAGALHAES NETO	M	1969-10-12	1995-05-13	CCE
7	1	CINTIA ANDRADE RIBEIRO	F	1970-03-21	1995-05-13	CSI
8	4	CLAUDIA GONCALVES MAC...	F	1969-06-13	1992-09-22	CCE
9	5	ANA PAULA XAVIER DE ABREU	F	1967-08-29	1997-10-03	CCE
10	6	ANA PAULA LASCASAS	F	1968-06-10	1999-04-03	CCE
11	10	RENATA CRISTINA LARANJA	F	1970-06-23	2002-03-28	NTI
12	11	FABRICIA PARAGUACU	F	1969-12-03	2008-07-25	REI

VIEW MATERIALIZADA

```
34  
35  SELECT * FROM MV_FUNCIONARIOS_ATIVOS;  
36  
37  REFRESH MATERIALIZED VIEW mv_funcionarios_ativos; -- Atualização da View Materializada  
38  
39  SELECT * FROM MV_FUNCIONARIOS_ATIVOS;  
40
```

Data Output Messages Notifications

	matricula integer	nome character varying (45)	sexo character (1)	nascimento date	dt_ingresso date	id_setor character (4)
1	2	EDUARDO LICOLINI	M	1967-09-19	1992-09-23	CSI
2	3	RONALDO CARVALHO CAMP...	M	1968-06-21	1995-02-14	CSI
3	7	HUMBERTO TAVARES NOGU...	M	1969-09-12	2000-06-06	CCE
4	8	THORSTEN BERG	M	1970-01-21	2000-01-02	CSI
5	9	LEONARDO PEREIRA	M	1971-01-21	2001-07-21	NTI
6	12	JOSE MAGALHAES NETO	M	1969-10-12	1995-05-13	CCE
7	1	CINTIA ANDRADE RIBEIRO	F	1970-03-21	1995-05-13	CSI
8	4	CLAUDIA GONCALVES MAC...	F	1969-06-13	1992-09-22	CCE
9	5	ANA PAULA XAVIER DE ABREU	F	1967-08-29	1997-10-03	CCE
10	6	ANA PAULA LASCASAS	F	1968-06-10	1999-04-03	CCE
11	10	RENATA CRISTINA LARANJA	F	1970-06-23	2002-03-28	NTI
12	11	FABRICIA PARAGUACU	F	1969-12-03	2008-07-25	REI
13	15	ASDRUBAL RIBEIRO SOARES	M	1956-04-03	1979-01-02	REI