Banco de Dados

Turma CIC0097 – 01/2025

Profa. Maristela Holanda

Carlos Iunes – 22/2012738

Laíssa Soares – 22/2032982

Robson Bezerra da Silva – 22/2035661

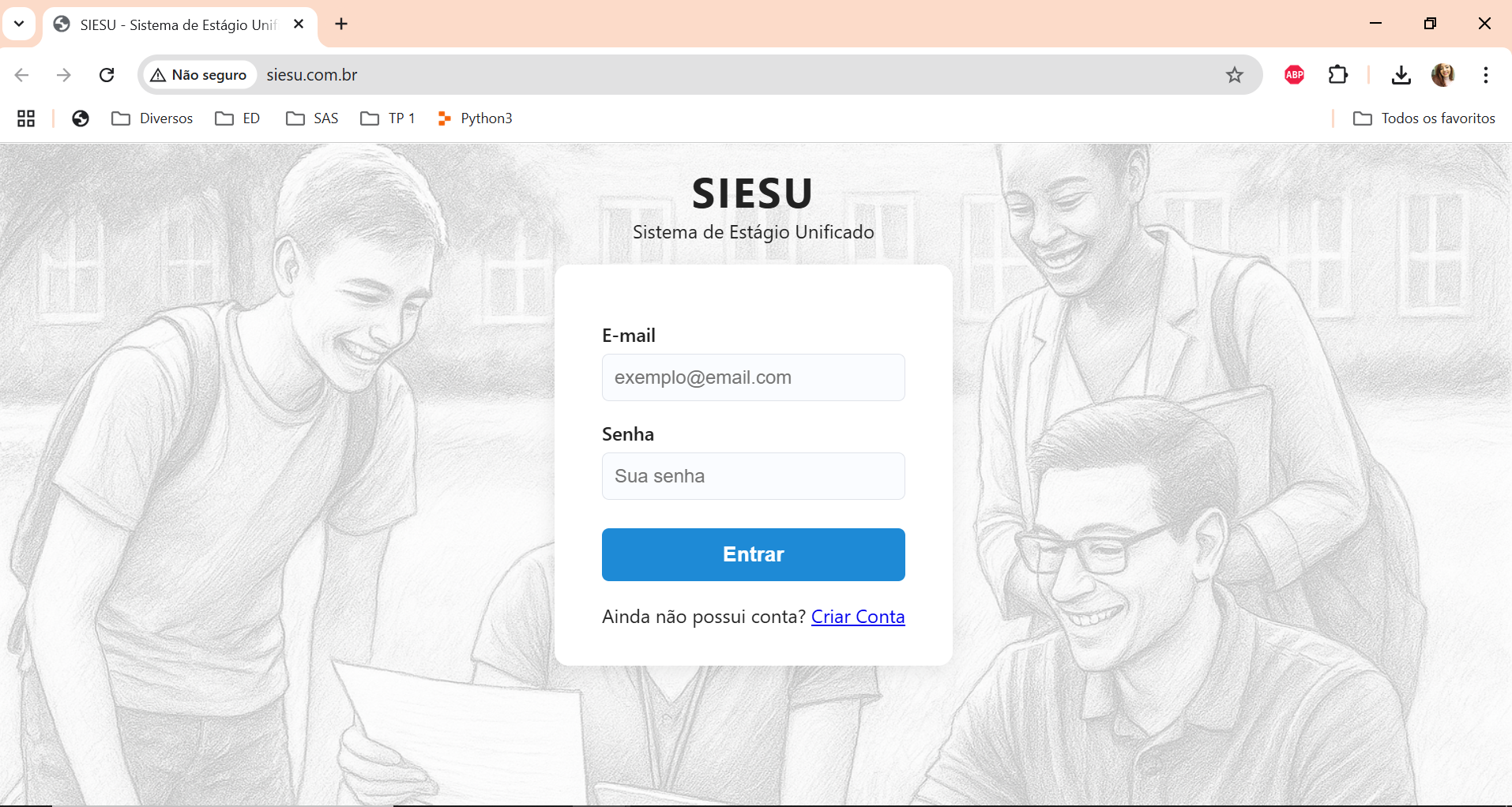
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**SIESU –** **SISTEMA PARA GESTÃO DE ESTÁGIOS OBRIGATÓRIOS SUPERVISIONADO**

1. **Introdução:**

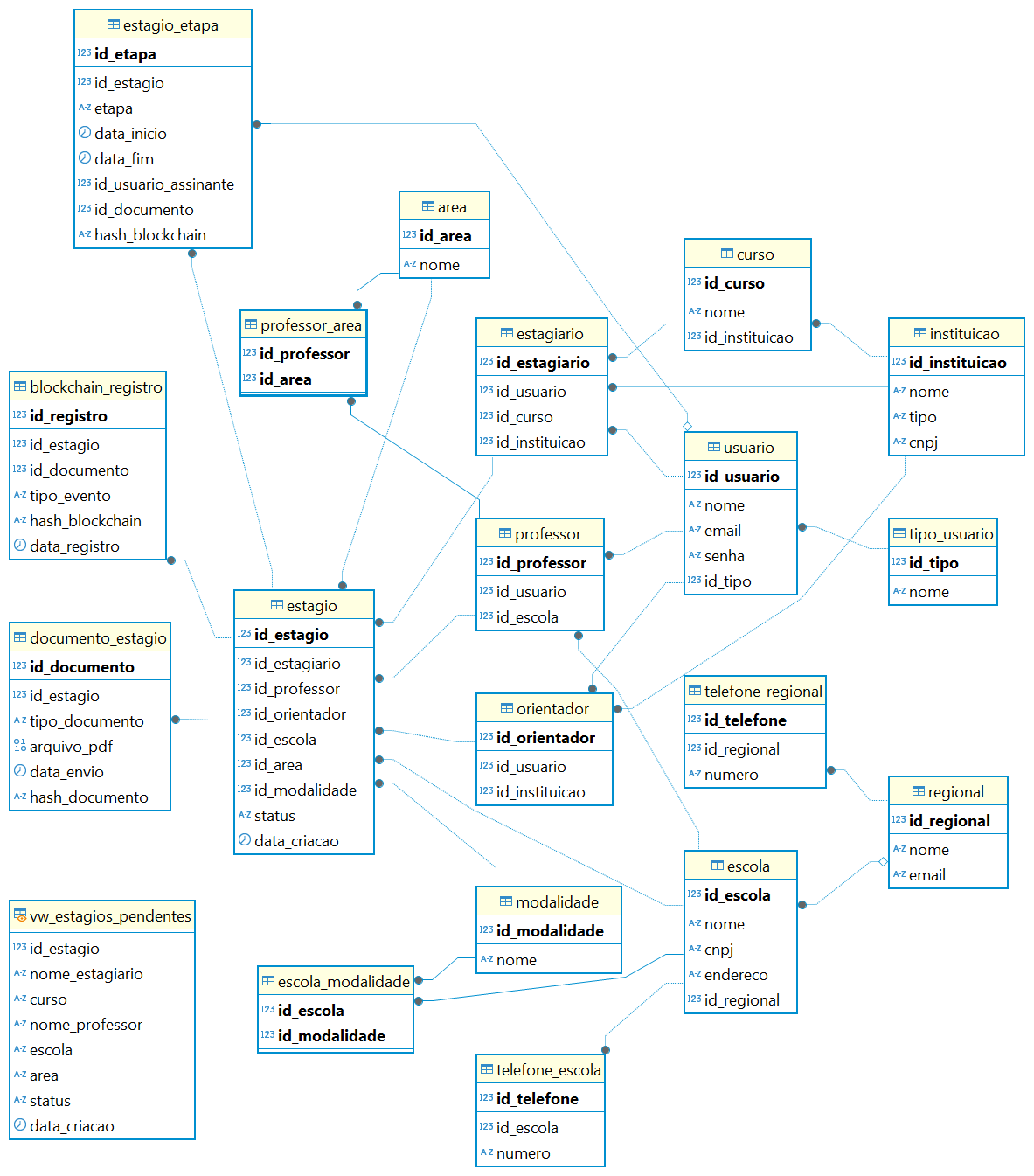
O sistema [***SIESU – Sistema de Estágio Supervisionado***](http://siesu.com.br/) ***–*** visa facilitar a **gestão de estágios**, envolvendo instituições, estudantes, escolas, professores, documentos e registros em blockchain.

Ele permitirá o cadastro de orientadores, professores da rede de ensino, e estagiários de quaisquer escolas/faculdades **cadastradas**. Além disso, serve para realizar *login*, pesquisar, registrar etapas e acompanhar a documentação necessária durante um estágio.



*Figura 1 - Página Inicial do Sistema de Estágio Unificado.*

1. **Modelo Entidade Relacionamento (MER):**
2. **Modelo Relacional (MR):**

****

*Figura 3 - Modelagem Lógica.*

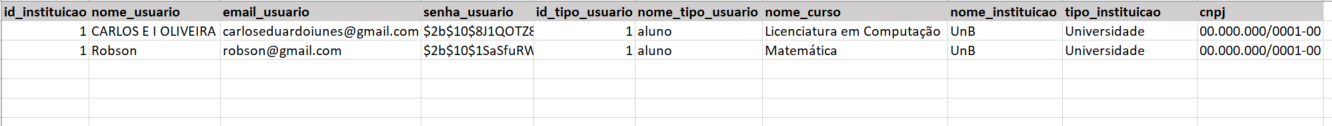
1. **Álgebra Relacional:** 
   1. **Consulta 1:** tabela com o nome de cada estagiário, o nome do curso que ele faz e a instituição à qual pertence.
   2. **Consulta 2:** apresenta um estágio, a escola onde acontece, sua modalidade e o nome do orientador.
   3. **Consulta 3:** apresenta os telefones das escolas e sua respectiva regional.
   4. **Consulta 4:** apresenta os professores e suas áreas atuando em estágios.
   5. **Consulta 5:** apresenta os estágios e seus documentos registrados em blockchain.
2. **Formas Normais:**

Para a avaliação das formas normais em cinco tabelas, usando a consulta a seguir, gerando a tabela denominada de tb\_estagiario\_detalhado, da imagem abaixo.

* 1. ***Script* da consulta da \_estagiario\_detalhado:**

| **SELECT**  ***e*.\*,**  ***u*.nome AS *nome\_usuario*,**  ***u*.email AS *email\_usuaio*,**  ***u*.senha AS *senha\_usuario*,**  ***u*.id\_tipo AS *id\_tipo\_usuario*,**  ***tu*.nome AS *nome\_tipo\_usuario*,**  ***c*.nome AS *nome\_curso*,**  ***i*.nome AS *nome\_instituicao*,**  ***i*.tipo AS *tipo\_instituicao*,**  ***i*.cnpj**  **FROM estagiario *e***  **INNER JOIN usuario *u***  **ON *e*.id\_usuario = *u*.id\_usuario**  **INNER JOIN tipo\_usuario *tu***  **ON *u*.id\_tipo = *tu*.id\_tipo**  **INNER JOIN curso *c***  **ON *e*.id\_curso = *c*.id\_curso**  **INNER JOIN instituicao *i***  **ON *e*.id\_instituicao = *i*.id\_instituicao**  **WHERE *tu*.nome = 'aluno';** |
| --- |

* 1. **Resultado tb\_estagiario\_detalhado:**

****

*Figura 4 - Tabela para normalização.*

* 1. **1ª Forma Normal (1FN):**

**Regra:**

* Todos os campos contêm valores atômicos e não repetidos em uma única célula.

Portanto, está em 1FN.

* 1. **2ª Forma Normal (2FN):**

**Regra:**

* Estar em 1FN.
* Todos os atributos **não chave** dependem da chave primária **inteira** e não apenas de parte dela.

A única chave candidata dessa tabela é **id\_estagiario**, pois ela identifica unicamente cada linha.

Portanto, está em 2FN.

* 1. **3ª Forma Normal (3FN):**

**Regra:**

* Estar em 2FN.
* Nenhum atributo não-chave pode depender transitivamente da chave primária.

Existem dependências transitivas na tabela, veja:

* *nome\_usuario, email\_usuario, senha\_usuario, id\_tipo\_usuario* **dependem** do **id\_usuario**;
* *nome\_tipo\_usuario* **depende** do **id\_tipo\_usuario***;*
* *nome\_curso* **depende** do **id\_curso**;
* *nome\_institucao, tipo\_instituicao, cnpj* **dependem** do **id\_instituicao**.

Para normalizar em 3FN, deve-se separar as entidades em:

| **tb\_estagiario** | **tb\_usuario** | **tb\_tipo\_usuario** | **tb\_curso** | **tb\_instituicao** |
| --- | --- | --- | --- | --- |
| id\_estagiario (PK)  id\_usuario (FK)  id\_curso (FK)  id\_instituicao (FK) | id\_usuario (PK)  nome  email  senha  id\_tipo\_usuario (FK) | id\_tipo (PK)  nome | id\_curso (PK)  nome | id\_instituicao (PK)  nome  tipo  cnpj |

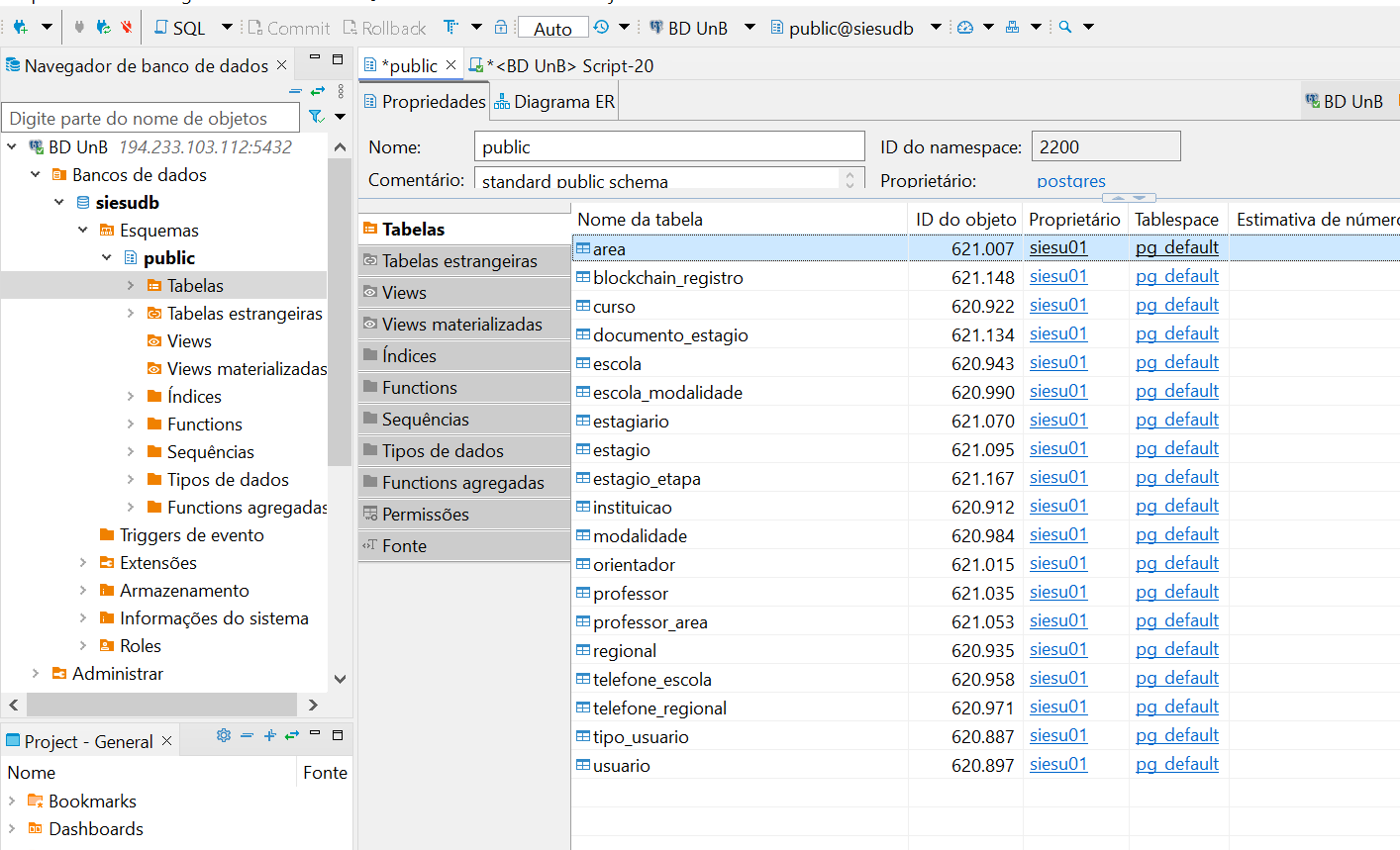
Assim cada atributo fica **dependendo apenas da chave primária da sua entidade**.

Portanto, o modelo está em 3FN.

1. **Modelo Físico - *Script* SQL:**

| **1** | -- Tabela de regional  **CREATE** **TABLE** public.regional (  id\_regional **serial4** **NOT** **NULL**,  nome **varchar**(100) **NOT** **NULL**,  email **varchar**(100) **NULL**,  **CONSTRAINT** regional\_pkey **PRIMARY** **KEY** (id\_regional)  ); |
| --- | --- |
| **2** | -- Tabela de escolas  **CREATE** **TABLE** public.escola (  id\_escola **serial4** **NOT** **NULL**,  nome **varchar**(150) **NOT** **NULL**,  cnpj **varchar**(18) **NOT** **NULL**,  endereco **varchar**(200) **NULL**,  id\_regional **int4** **NULL**,  **CONSTRAINT** escola\_cnpj\_key **UNIQUE** (cnpj),  **CONSTRAINT** escola\_pkey **PRIMARY** **KEY** (id\_escola)  );  -- public.escola chaves estrangeiras  **ALTER** **TABLE** public.escola **ADD** **CONSTRAINT** escola\_id\_regional\_fkey **FOREIGN** **KEY** (id\_regional) **REFERENCES** public.regional(id\_regional); |
| **3** | -- Tabela de telefone regional  **CREATE** **TABLE** public.telefone\_regional (  id\_telefone **serial4** **NOT** **NULL**,  id\_regional **int4** **NOT** **NULL**,  numero **varchar**(20) **NOT** **NULL**,  **CONSTRAINT** telefone\_regional\_pkey **PRIMARY** **KEY** (id\_telefone)  );  -- public.telefone\_regional chaves estrangeiras  **ALTER** **TABLE** public.telefone\_regional **ADD** **CONSTRAINT** telefone\_regional\_id\_regional\_fkey **FOREIGN** **KEY** (id\_regional) **REFERENCES** public.regional(id\_regional) **ON** **DELETE** **CASCADE**; |
| **4** | -- Tabela de telefone escola  **CREATE** **TABLE** public.telefone\_escola (  id\_telefone **serial4** **NOT** **NULL**,  id\_escola **int4** **NOT** **NULL**,  numero **varchar**(20) **NOT** **NULL**,  **CONSTRAINT** telefone\_escola\_pkey **PRIMARY** **KEY** (id\_telefone)  );  -- public.telefone\_escola chaves estrangeiras  **ALTER** **TABLE** public.telefone\_escola **ADD** **CONSTRAINT** telefone\_escola\_id\_escola\_fkey **FOREIGN** **KEY** (id\_escola) **REFERENCES** public.escola(id\_escola) **ON** **DELETE** **CASCADE**; |
| **5** | -- Tabela de instituição  **CREATE** **TABLE** public.instituicao (  id\_instituicao **serial4** **NOT** **NULL**,  nome **varchar**(150) **NOT** **NULL**,  tipo **varchar**(50) **NOT** **NULL**,  cnpj **varchar**(18) **NULL**,  **CONSTRAINT** instituicao\_cnpj\_key **UNIQUE** (cnpj),  **CONSTRAINT** instituicao\_pkey **PRIMARY** **KEY** (id\_instituicao)  ); |
| **6** | -- Tabela de curso  **CREATE** **TABLE** public.curso (  id\_curso **serial4** **NOT** **NULL**,  nome **varchar**(100) **NOT** **NULL**,  id\_instituicao **int4** **NOT** **NULL**,  **CONSTRAINT** curso\_pkey **PRIMARY** **KEY** (id\_curso)  );  -- public.curso chaves estrangeiras  **ALTER** **TABLE** public.curso **ADD** **CONSTRAINT** curso\_id\_instituicao\_fkey **FOREIGN** **KEY** (id\_instituicao) **REFERENCES** public.instituicao(id\_instituicao); |
| **7** | -- Tabela de estagiário  **CREATE** **TABLE** public.estagiario (  id\_estagiario **serial4** **NOT** **NULL**,  id\_usuario **int4** **NOT** **NULL**,  id\_curso **int4** **NOT** **NULL**,  id\_instituicao **int4** **NOT** **NULL**,  **CONSTRAINT** estagiario\_id\_usuario\_key **UNIQUE** (id\_usuario),  **CONSTRAINT** estagiario\_pkey **PRIMARY** **KEY** (id\_estagiario)  );  -- public.estagiario chaves estrangeiras  **ALTER** **TABLE** public.estagiario **ADD** **CONSTRAINT** estagiario\_id\_curso\_fkey **FOREIGN** **KEY** (id\_curso) **REFERENCES** public.curso(id\_curso);  **ALTER** **TABLE** public.estagiario **ADD** **CONSTRAINT** estagiario\_id\_instituicao\_fkey **FOREIGN** **KEY** (id\_instituicao) **REFERENCES** public.instituicao(id\_instituicao);  **ALTER** **TABLE** public.estagiario **ADD** **CONSTRAINT** estagiario\_id\_usuario\_fkey **FOREIGN** **KEY** (id\_usuario) **REFERENCES** public.usuario(id\_usuario) **ON** **DELETE** **CASCADE**; |
| **8** | -- Tabela do tipo do usuário  **CREATE** **TABLE** public.tipo\_usuario (  id\_tipo **serial4** **NOT** **NULL**,  nome **varchar**(30) **NOT** **NULL**,  **CONSTRAINT** tipo\_usuario\_nome\_key **UNIQUE** (nome),  **CONSTRAINT** tipo\_usuario\_pkey **PRIMARY** **KEY** (id\_tipo)  ); |
| **9** | -- Tabela de usuário  **CREATE** **TABLE** public.usuario (  id\_usuario **serial4** **NOT** **NULL**,  nome **varchar**(100) **NOT** **NULL**,  email **varchar**(100) **NOT** **NULL**,  senha **varchar**(100) **NOT** **NULL**,  id\_tipo **int4** **NOT** **NULL**,  **CONSTRAINT** usuario\_email\_key **UNIQUE** (email),  **CONSTRAINT** usuario\_pkey **PRIMARY** **KEY** (id\_usuario)  );  -- public.usuario chaves estrangeiras  **ALTER** **TABLE** public.usuario **ADD** **CONSTRAINT** usuario\_id\_tipo\_fkey **FOREIGN** **KEY** (id\_tipo) **REFERENCES** public.tipo\_usuario(id\_tipo); |
| **10** | -- Tabela de professor  **CREATE** **TABLE** public.professor (  id\_professor **serial4** **NOT** **NULL**,  id\_usuario **int4** **NOT** **NULL**,  id\_escola **int4** **NOT** **NULL**,  **CONSTRAINT** professor\_id\_usuario\_key **UNIQUE** (id\_usuario),  **CONSTRAINT** professor\_pkey **PRIMARY** **KEY** (id\_professor)  );  -- public.professor chaves estrangeiras  **ALTER** **TABLE** public.professor **ADD** **CONSTRAINT** professor\_id\_escola\_fkey **FOREIGN** **KEY** (id\_escola) **REFERENCES** public.escola(id\_escola);  **ALTER** **TABLE** public.professor **ADD** **CONSTRAINT** professor\_id\_usuario\_fkey **FOREIGN** **KEY** (id\_usuario) **REFERENCES** public.usuario(id\_usuario) **ON** **DELETE** **CASCADE**; |
| **11** | -- Tabela de área  **CREATE** **TABLE** public.area (  id\_area **serial4** **NOT** **NULL**,  nome **varchar**(100) **NOT** **NULL**,  **CONSTRAINT** area\_pkey **PRIMARY** **KEY** (id\_area)  ); |
| **12** | -- Tabela de área do professor  **CREATE** **TABLE** public.professor\_area (  id\_professor **int4** **NOT** **NULL**,  id\_area **int4** **NOT** **NULL**,  **CONSTRAINT** professor\_area\_pkey **PRIMARY** **KEY** (id\_professor, id\_area)  );  -- public.professor\_area chaves estrangeiras  **ALTER** **TABLE** public.professor\_area **ADD** **CONSTRAINT** professor\_area\_id\_area\_fkey **FOREIGN** **KEY** (id\_area) **REFERENCES** public.area(id\_area) **ON** **DELETE** **CASCADE**;  **ALTER** **TABLE** public.professor\_area **ADD** **CONSTRAINT** professor\_area\_id\_professor\_fkey **FOREIGN** **KEY** (id\_professor) **REFERENCES** public.professor(id\_professor) **ON** **DELETE** **CASCADE**; |
| **13** | -- Tabela de modalidade  **CREATE TABLE public.modalidade (**  **id\_modalidade serial4 NOT NULL,**  **nome varchar(50) NOT NULL,**  **CONSTRAINT modalidade\_pkey PRIMARY KEY (id\_modalidade)**  **);** |
| **14** | -- Tabela de modalidade da escola  **CREATE** **TABLE** public.escola\_modalidade (  id\_escola **int4** **NOT** **NULL**,  id\_modalidade **int4** **NOT** **NULL**,  **CONSTRAINT** escola\_modalidade\_pkey **PRIMARY** **KEY** (id\_escola, id\_modalidade)  );  -- public.escola\_modalidade chaves estrangeiras  **ALTER** **TABLE** public.escola\_modalidade **ADD** **CONSTRAINT** escola\_modalidade\_id\_escola\_fkey **FOREIGN** **KEY** (id\_escola) **REFERENCES** public.escola(id\_escola) **ON** **DELETE** **CASCADE**;  **ALTER** **TABLE** public.escola\_modalidade **ADD** **CONSTRAINT** escola\_modalidade\_id\_modalidade\_fkey **FOREIGN** **KEY** (id\_modalidade) **REFERENCES** public.modalidade(id\_modalidade) **ON** **DELETE** **CASCADE**; |
| **15** | -- Tabela de orientador  **CREATE** **TABLE** public.orientador (  id\_orientador **serial4** **NOT** **NULL**,  id\_usuario **int4** **NOT** **NULL**,  id\_instituicao **int4** **NOT** **NULL**,  **CONSTRAINT** orientador\_id\_usuario\_key **UNIQUE** (id\_usuario),  **CONSTRAINT** orientador\_pkey **PRIMARY** **KEY** (id\_orientador)  );  -- public.orientador chaves estrangeiras  **ALTER** **TABLE** public.orientador **ADD** **CONSTRAINT** orientador\_id\_instituicao\_fkey **FOREIGN** **KEY** (id\_instituicao) **REFERENCES** public.instituicao(id\_instituicao);  **ALTER** **TABLE** public.orientador **ADD** **CONSTRAINT** orientador\_id\_usuario\_fkey **FOREIGN** **KEY** (id\_usuario) **REFERENCES** public.usuario(id\_usuario) **ON** **DELETE** **CASCADE**; |
| **16** | -- Tabela de estágio  **CREATE** **TABLE** public.estagio (  id\_estagio **serial4** **NOT** **NULL**,  id\_estagiario **int4** **NOT** **NULL**,  id\_professor **int4** **NOT** **NULL**,  id\_orientador **int4** **NOT** **NULL**,  id\_escola **int4** **NOT** **NULL**,  id\_area **int4** **NOT** **NULL**,  id\_modalidade **int4** **NOT** **NULL**,  status **varchar**(30) **NOT** **NULL**,  data\_criacao **timestamp** **DEFAULT** CURRENT\_TIMESTAMP **NULL**,  **CONSTRAINT** estagio\_pkey **PRIMARY** **KEY** (id\_estagio)  );  -- public.estagio chaves estrangeiras  **ALTER** **TABLE** public.estagio **ADD** **CONSTRAINT** estagio\_id\_area\_fkey **FOREIGN** **KEY** (id\_area) **REFERENCES** public.area(id\_area);  **ALTER** **TABLE** public.estagio **ADD** **CONSTRAINT** estagio\_id\_escola\_fkey **FOREIGN** **KEY** (id\_escola) **REFERENCES** public.escola(id\_escola);  **ALTER** **TABLE** public.estagio **ADD** **CONSTRAINT** estagio\_id\_estagiario\_fkey **FOREIGN** **KEY** (id\_estagiario) **REFERENCES** public.estagiario(id\_estagiario);  **ALTER** **TABLE** public.estagio **ADD** **CONSTRAINT** estagio\_id\_modalidade\_fkey **FOREIGN** **KEY** (id\_modalidade) **REFERENCES** public.modalidade(id\_modalidade);  **ALTER** **TABLE** public.estagio **ADD** **CONSTRAINT** estagio\_id\_orientador\_fkey **FOREIGN** **KEY** (id\_orientador) **REFERENCES** public.orientador(id\_orientador);  **ALTER** **TABLE** public.estagio **ADD** **CONSTRAINT** estagio\_id\_professor\_fkey **FOREIGN** **KEY** (id\_professor) **REFERENCES** public.professor(id\_professor); |
| **17** | -- Tabela de documento do estágio  **CREATE** **TABLE** public.documento\_estagio (  id\_documento **serial4** **NOT** **NULL**,  id\_estagio **int4** **NOT** **NULL**,  tipo\_documento **varchar**(50) **NOT** **NULL**,  caminho\_pdf **varchar**(200) **NOT** **NULL**,  data\_envio **timestamp** **DEFAULT** CURRENT\_TIMESTAMP **NULL**,  **CONSTRAINT** documento\_estagio\_pkey **PRIMARY** **KEY** (id\_documento)  );  -- public.documento\_estagio chaves estrangeiras  **ALTER** **TABLE** public.documento\_estagio **ADD** **CONSTRAINT** documento\_estagio\_id\_estagio\_fkey **FOREIGN** **KEY** (id\_estagio) **REFERENCES** public.estagio(id\_estagio); |
| **18** | -- Tabela de blockchain do registro  **CREATE** **TABLE** public.blockchain\_registro (  id\_registro **serial4** **NOT** **NULL**,  id\_estagio **int4** **NOT** **NULL**,  id\_documento **int4** **NULL**,  tipo\_evento **varchar**(50) **NOT** **NULL**,  hash\_blockchain **varchar**(150) **NOT** **NULL**,  data\_registro **timestamp** **DEFAULT** CURRENT\_TIMESTAMP **NULL**,  **CONSTRAINT** blockchain\_registro\_pkey **PRIMARY** **KEY** (id\_registro)  );  -- public.blockchain\_registro chaves estrangeiras  **ALTER** **TABLE** public.blockchain\_registro **ADD** **CONSTRAINT** blockchain\_registro\_id\_documento\_fkey **FOREIGN** **KEY** (id\_documento) **REFERENCES** public.documento\_estagio(id\_documento);  **ALTER** **TABLE** public.blockchain\_registro **ADD** **CONSTRAINT** blockchain\_registro\_id\_estagio\_fkey **FOREIGN** **KEY** (id\_estagio) **REFERENCES** public.estagio(id\_estagio); |
| **19** | -- Tabela de etapa do estágio  **CREATE** **TABLE** public.estagio\_etapa (  id\_etapa **serial4** **NOT** **NULL**,  id\_estagio **int4** **NOT** **NULL**,  etapa **varchar**(50) **NOT** **NULL**,  data\_inicio **timestamp** **DEFAULT** CURRENT\_TIMESTAMP **NOT** **NULL**,  data\_fim **timestamp** **NULL**,  id\_usuario\_assinante **int4** **NULL**,  id\_documento **int4** **NULL**,  hash\_blockchain **varchar**(150) **NULL**,  **CONSTRAINT** estagio\_etapa\_pkey **PRIMARY** **KEY** (id\_etapa)  );  -- public.estagio\_etapa chaves estrangeiras  **ALTER** **TABLE** public.estagio\_etapa **ADD** **CONSTRAINT** estagio\_etapa\_id\_documento\_fkey **FOREIGN** **KEY** (id\_documento) **REFERENCES** public.documento\_estagio(id\_documento);  **ALTER** **TABLE** public.estagio\_etapa **ADD** **CONSTRAINT** estagio\_etapa\_id\_estagio\_fkey **FOREIGN** **KEY** (id\_estagio) **REFERENCES** public.estagio(id\_estagio);  **ALTER** **TABLE** public.estagio\_etapa **ADD** **CONSTRAINT** estagio\_etapa\_id\_usuario\_assinante\_fkey **FOREIGN** **KEY** (id\_usuario\_assinante) **REFERENCES** public.usuario(id\_usuario); |

* 1. **Tabelas criadas no banco siesudb:**

*Figura 5 - Tabelas criadas no PostgreSQL.*

1. **View** ***vw\_estagios\_pendentes*:**
   1. **Objetivo:**

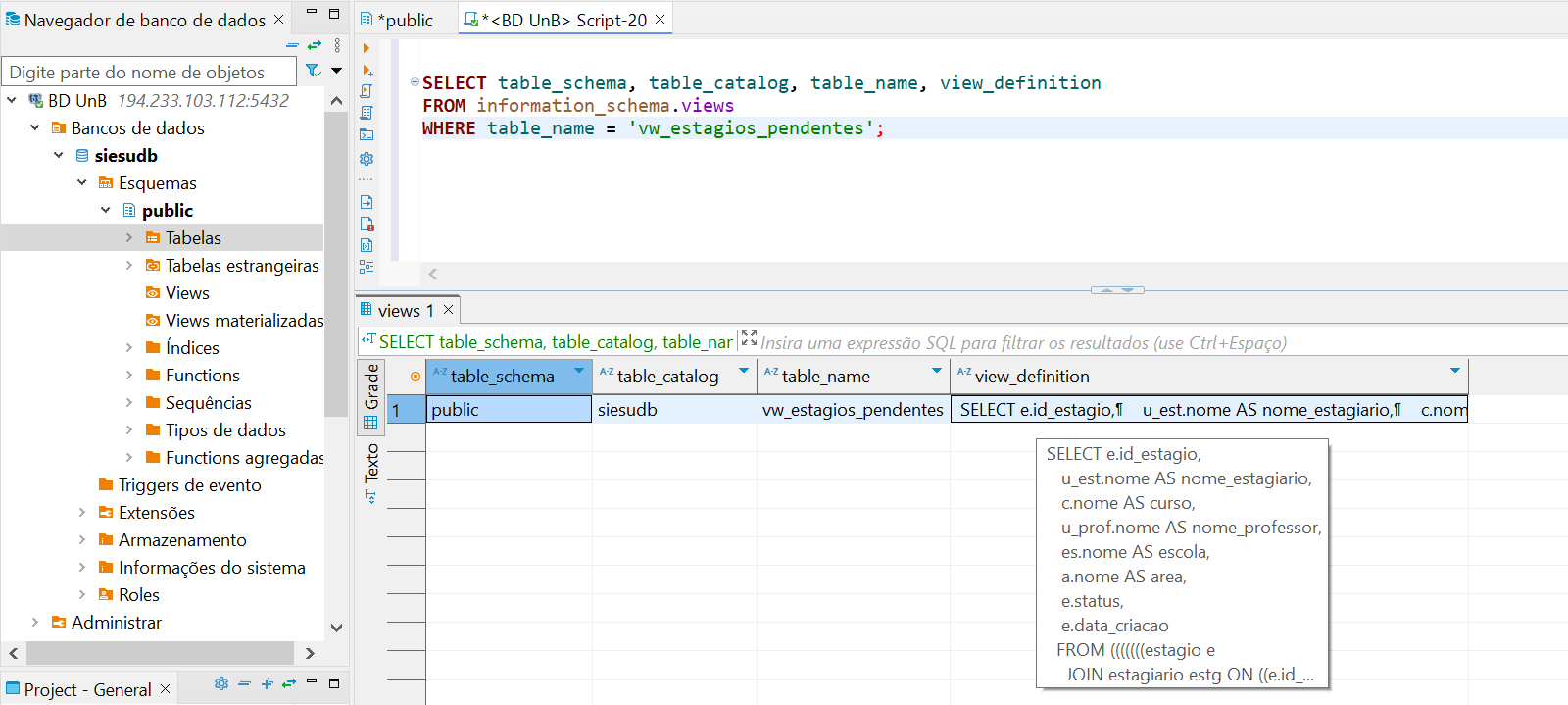
Selecionar e exibir informações dos estágios que possuem *status* igual a **'Pendente'**, permitindo a visualização dos dados do *estagiário, curso, professor responsável, escola, área de atuação, status* e *data\_criacao*.

A *view* facilita o acompanhamento de estágios que ainda estão pendentes, fornecendo recurso de consulta rápida e organizada para controle.

* 1. **Código SQL:**

| **CREATE** **VIEW** vw\_estagios\_pendentes **AS**  **SELECT**  e.id\_estagio,  u\_est.nome **AS** nome\_estagiario,  c.nome **AS** curso,  u\_prof.nome **AS** nome\_professor,  es.nome **AS** escola,  a.nome **AS** area,  e.status,  e.data\_criacao  **FROM** estagio e  **INNER** **JOIN** estagiario estg  **ON** e.id\_estagiario = estg.id\_estagiario  **INNER** **JOIN** usuario u\_est  **ON** estg.id\_usuario = u\_est.id\_usuario  **INNER** **JOIN** curso c  **ON** estg.id\_curso = c.id\_curso  **INNER** **JOIN** professor p  **ON** e.id\_professor = p.id\_professor  **INNER** **JOIN** usuario u\_prof  **ON** p.id\_usuario = u\_prof.id\_usuario  **INNER** **JOIN** escola es  **ON** e.id\_escola = es.id\_escola  **INNER** **JOIN** area a  **ON** e.id\_area = a.id\_area  **WHERE** **upper**(e.status) = 'PENDENTE'; |
| --- |

* 1. **Verificação da View no banco siesudb:**



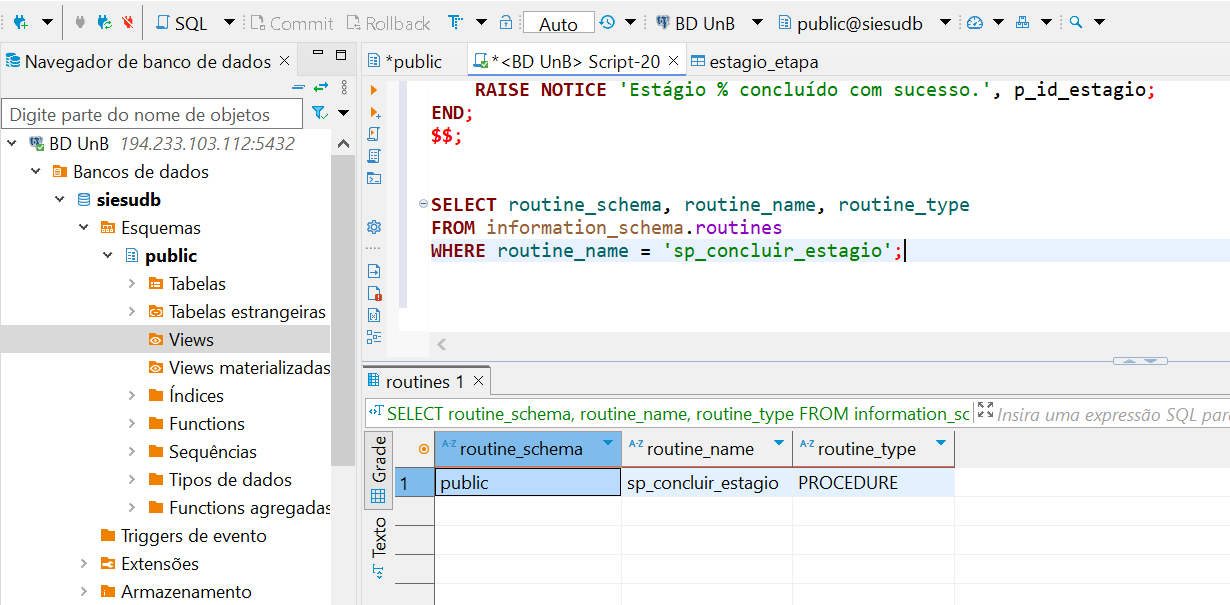
## *Figura 6 - Verificação da view no PostgreSQL.*

1. **Procedure *sp\_concluir\_estagio*:**
   1. **Objetivo:**  
       Atualizar o status de um estágio para 'Concluído', mas somente se ele tiver pelo menos uma etapa com etapa = 'Finalizado'.

* Se existir essa etapa, atualiza o registro.
* Se não existir, emite uma mensagem de aviso e não faz nada.
  1. **Código SQL:**

| **CREATE** **OR** **REPLACE** **PROCEDURE** sp\_concluir\_estagio(p\_id\_estagio **INT**)  **LANGUAGE** plpgsql  **AS** **$$**  **DECLARE**  v\_count\_etapas **INT**;  **BEGIN**  -- Verifica quantas etapas "Finalizado" existem para o estágio informado  **SELECT** **COUNT**(\*) **INTO** v\_count\_etapas  **FROM** estagio\_etapa  **WHERE** id\_estagio = p\_id\_estagio  **AND** etapa = 'Finalizado';  -- Se não encontrou nenhuma etapa finalizada, não atualiza  **IF** v\_count\_etapas = 0 **THEN**  **RAISE** **NOTICE** 'Nenhuma etapa finalizada encontrada. O estágio não pode ser concluído.';  **RETURN**;  **END** **IF**;  -- Caso contrário, atualiza o status do estágio  **UPDATE** estagio  **SET** status = 'Concluído'  **WHERE** id\_estagio = p\_id\_estagio;  **RAISE** **NOTICE** 'Estágio % concluído com sucesso.', p\_id\_estagio;  **END**;  **$$**; |
| --- |

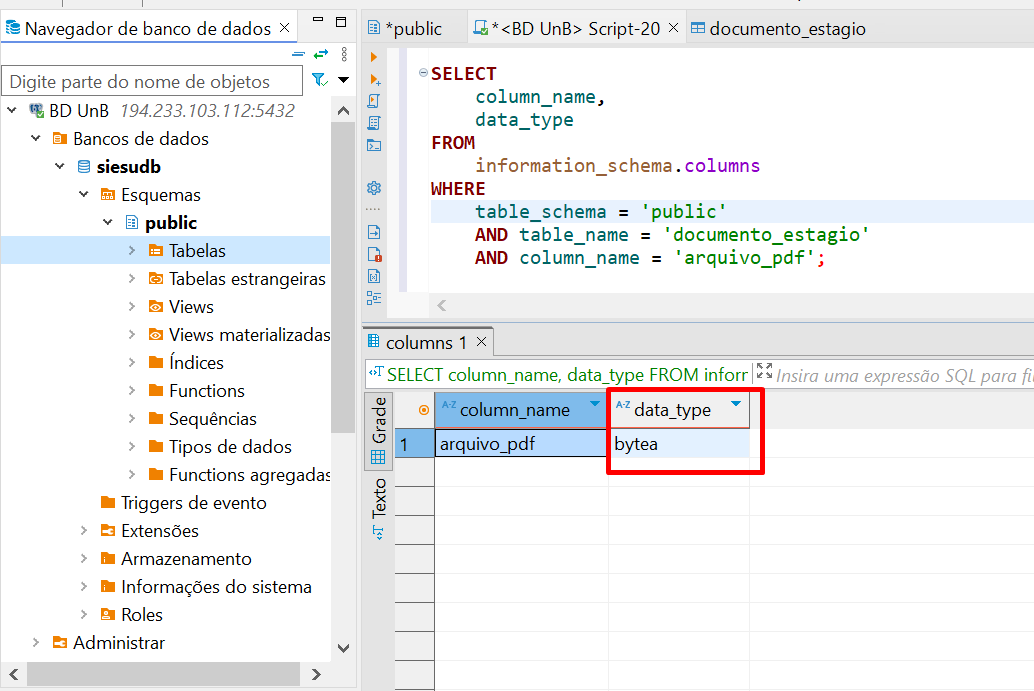
* 1. **Verificação da Procedure no banco siesudb:**



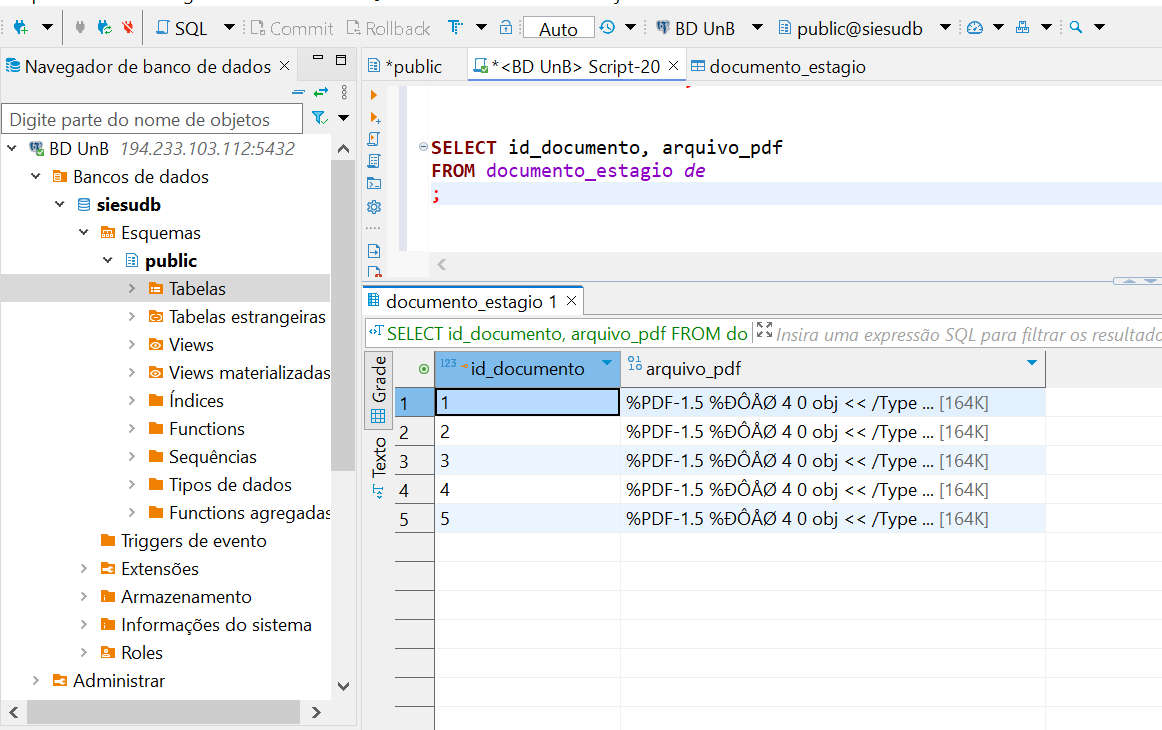
## *Figura 7 - Verificação do Procedure no PostgreSQL.*

1. **Dados Binários no banco siesudb**

Para formalizar o estágio, os estudantes de instituições de ensino superior, precisam do documento TCE - Termo de Compromisso de Estágio. Este, por sua vez, será adicionado à plataforma e armazenado no banco de dados no atributo **arquivo\_pdf** na tabela **documento\_estágio**. Assim, esse documento será o dado binário que teremos no banco de dados.



*Figura 8 - Verificação do tipo da coluna arquivo\_pdf.*

****

*Figura 9 - Verificação dos dados inseridos na coluna arquivo\_pdf na tabela documento\_estagio.*

1. **Acesso aos códigos**

[Repositório Trabalho Final - BD](https://github.com/laissabiia/trabalho-final-bd.git)