

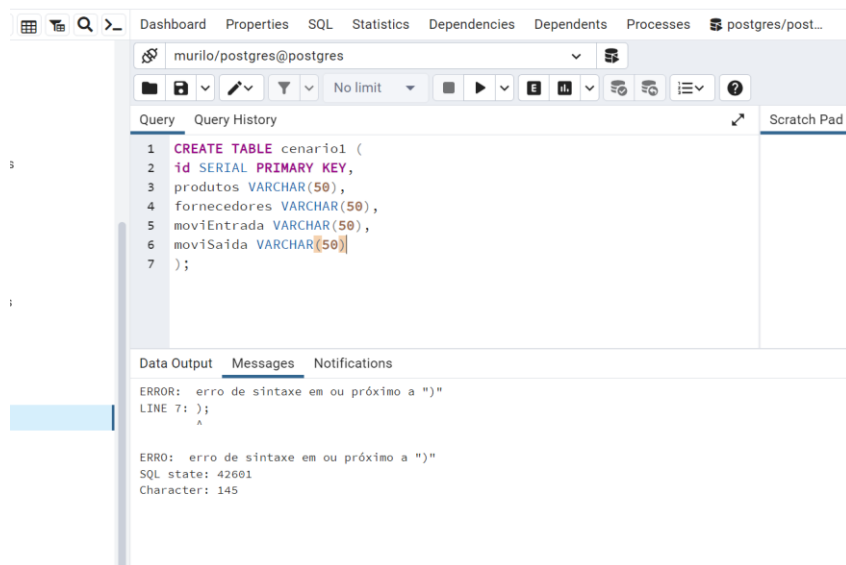
Cenário 1

Uma empresa possui um sistema de controle de estoque, onde precisa gerenciar informações

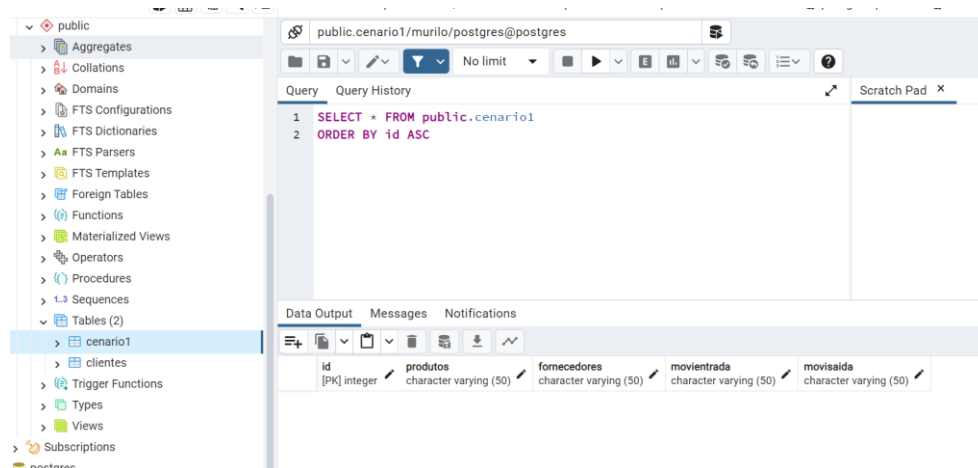
sobre produtos, fornecedores, e movimentações de entrada e saída de mercadorias. A

estrutura é previsível e baseada em relações claras entre produtos e fornecedores.

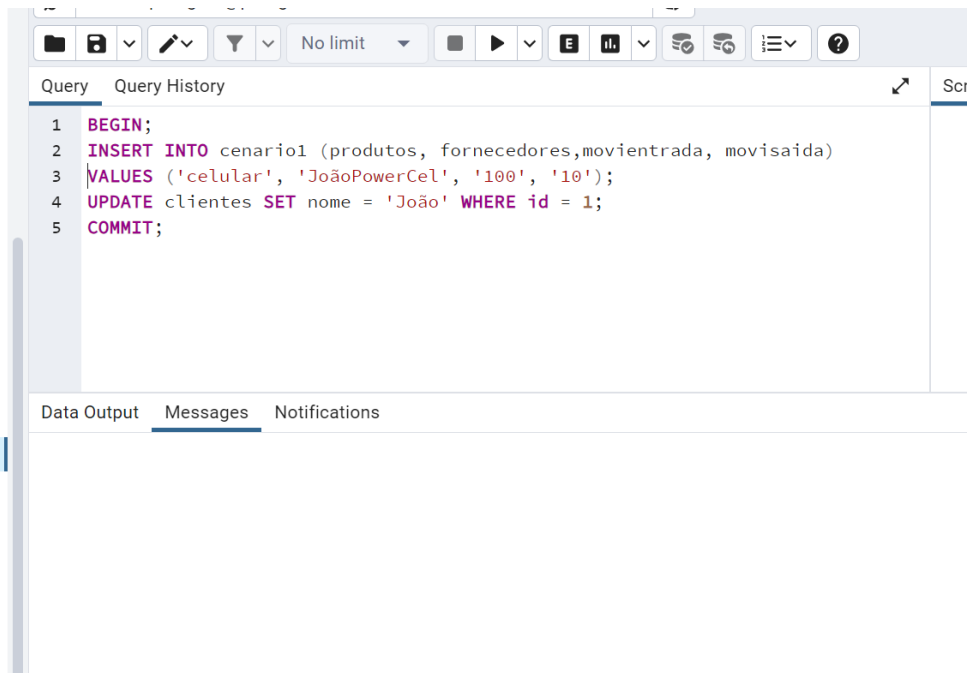
Etapa 1. Criação da tabela “CENARIO1”



Etapa 2. Visualização da tabela criada.



Etapa 3.Inserção de dados na tabela

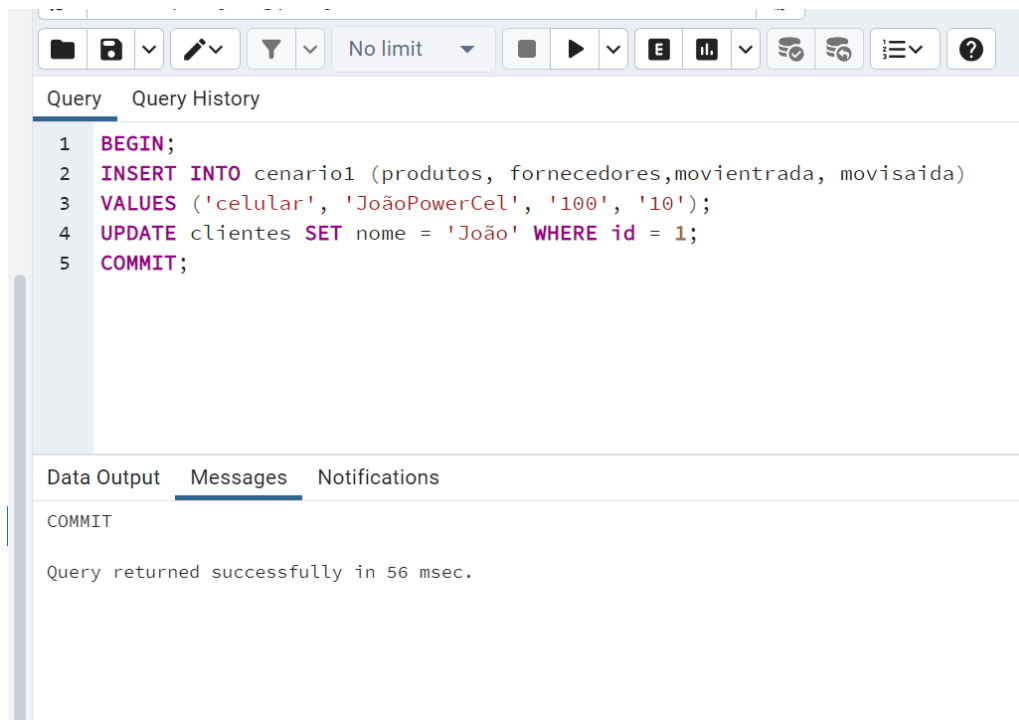


The screenshot shows a SQL IDE interface. At the top, there is a toolbar with icons for file operations, query execution, and settings. Below the toolbar, there are tabs for 'Query' and 'Query History'. The 'Query' tab is active, displaying the following SQL code:

```
1 BEGIN;  
2 INSERT INTO cenario1 (produtos, fornecedores,movientrada, movisaida)  
3 VALUES ('celular', 'JoãoPowerCel', '100', '10');  
4 UPDATE clientes SET nome = 'João' WHERE id = 1;  
5 COMMIT;
```

Below the query window, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is currently selected, but it is empty.

Etapa 3.1 – Confirmação da inserção de dados.



This screenshot shows the same SQL IDE interface as the previous one, but now the 'Messages' tab is active. It displays the following information:

```
COMMIT
```

Query returned successfully in 56 msec.

Etapa 4. Seleção e visualização dos dados inseridos na tabela.

- > Domains
- > FTS Configurations
- > FTS Dictionaries
- > **Aa** FTS Parsers
- > FTS Templates
- > Foreign Tables
- > Functions
- > Materialized Views
- > Operators
- > Procedures
- > Sequences
- > Tables (2)
- > **cenario1**
- > clientes
- > Trigger Functions
- > Types
- > Views
- > Subscriptions

Query
Query History
Scratch Pad

```
1 SELECT * FROM public.cenario1
2 ORDER BY id ASC
```

Data Output
Messages
Notifications

	id [PK] integer	produtos character varying (50)	fornecedores character varying (50)	movientrada character varying (50)	movisaida character varying (50)
1	1	celular	JoãoPowerCel	100	10

Etapa 5. Alteração dos dados inseridos na tabela.

```
1 BEGIN;  
2 UPDATE cenario1 SET produtos = 'Tablet' WHERE id = 1;  
3 COMMIT;
```

Etapa 5.1. Confirmação da alteração dos dados.

Data Output					
<div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> </div>					
id	produtos	fornecedores	movientrada	movisaida	
[PK] integer	character varying (50)	character varying (50)	character varying (50)	character varying (50)	
1	1 Tablet	JoãoPowerCel	100	10	

Neste cenário, encontramos a necessidade de criar uma tabela para gerenciar um estoque de produtos, fornecedores, movimentações de entrada e movimentações de saída.

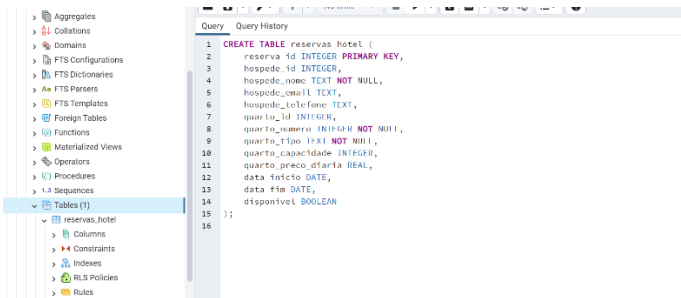
Nesse processo foi utilizado os comandos em sql para criação de uma tabela em um banco de dados.

Cenário 3: Crie um sistema de reservas para um hotel, onde é necessário gerenciar informações sobre

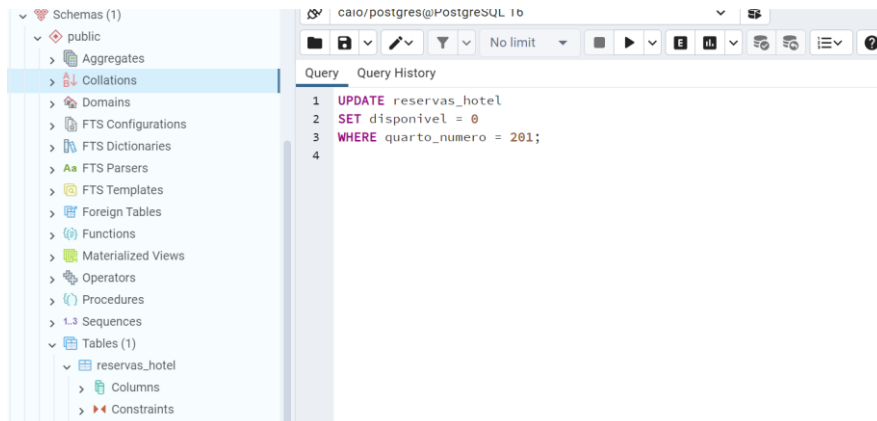
quartos, hóspedes, datas de reserva e disponibilidade. A estrutura deve seguir um modelo

relacional para garantir consistência e integridade dos dados.

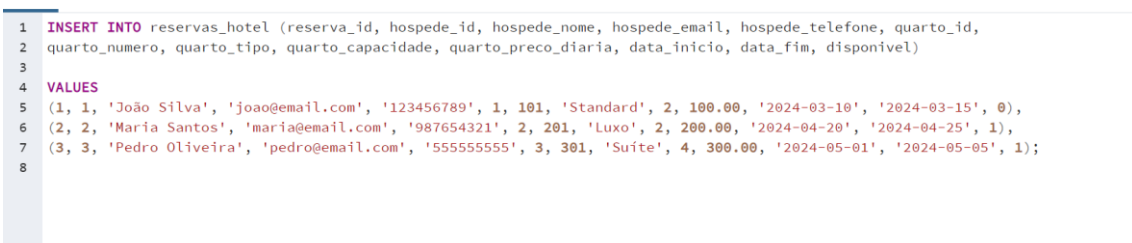
Etapa 1. Criação da tabela



Etapa 2: Inserção de dados na tabela



Etapa 3: Alteração de dados da tabela

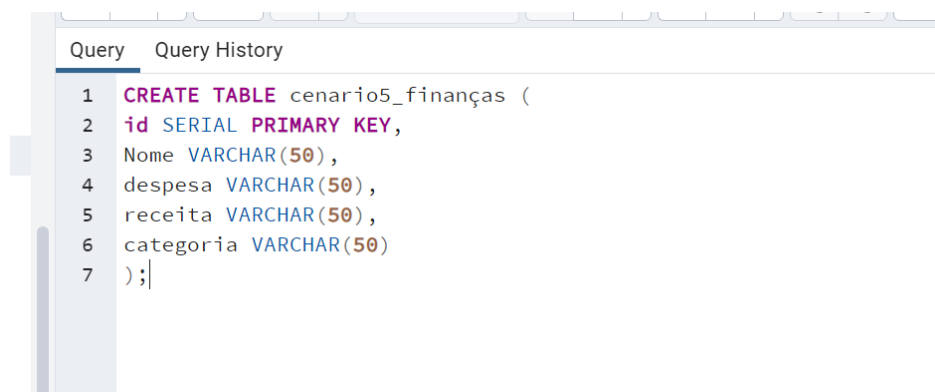


Neste cenário, identificamos a necessidade de criar uma estrutura de banco de dados para gerenciar as operações de um hotel, incluindo reservas, informações sobre hóspedes, quartos e outras atividades relacionadas. Para atender a essa demanda, foram utilizados comandos SQL para criar tabelas no banco de dados.

Cenário 5: Sistema de Gerenciamento de Finanças Pessoais

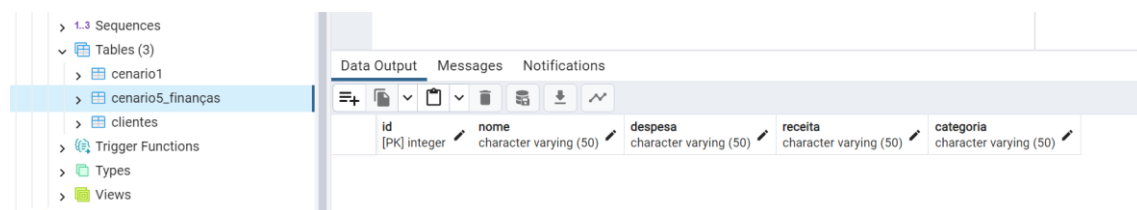
Crie um sistema para o gerenciamento de finanças pessoais, onde os usuários podem registrar despesas, receitas e categorias. A estrutura deve ser organizada de forma relacional para garantir precisão e coesão nas informações financeiras.

Etapa 1.0 Criação da tabela

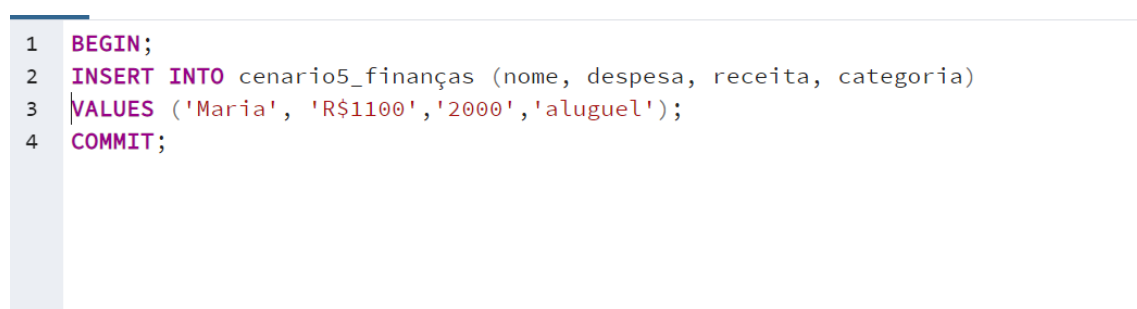


```
Query  Query History
1 CREATE TABLE cenario5_finanças (
2   id SERIAL PRIMARY KEY,
3   Nome VARCHAR(50),
4   despesa VARCHAR(50),
5   receita VARCHAR(50),
6   categoria VARCHAR(50)
7 );
```

Etapa 1.1 Confirmação da tabela criada

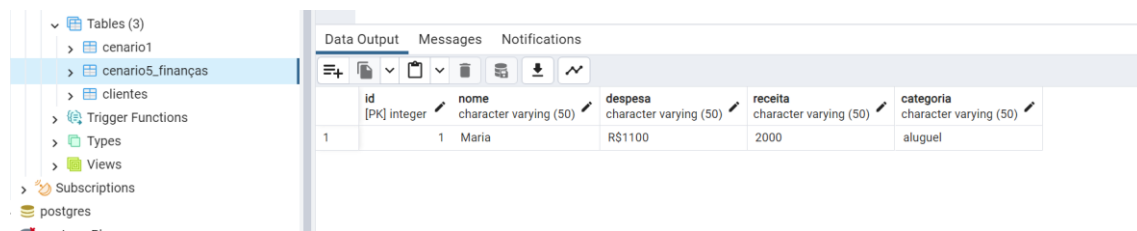


Etapa 2.0 Inserção de dados na tabela



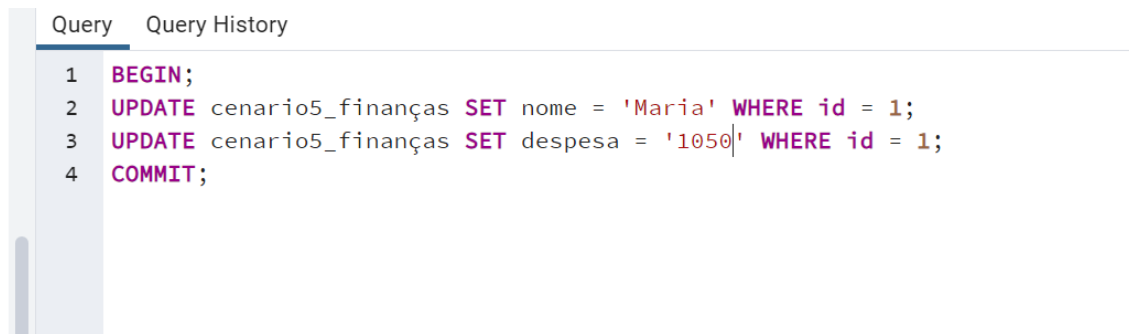
```
1 BEGIN;
2 INSERT INTO cenario5_finanças (nome, despesa, receita, categoria)
3 VALUES ('Maria', 'R$1100', '2000', 'aluguel');
4 COMMIT;
```

Etapa 2.1 Confirmação da inserção de dados



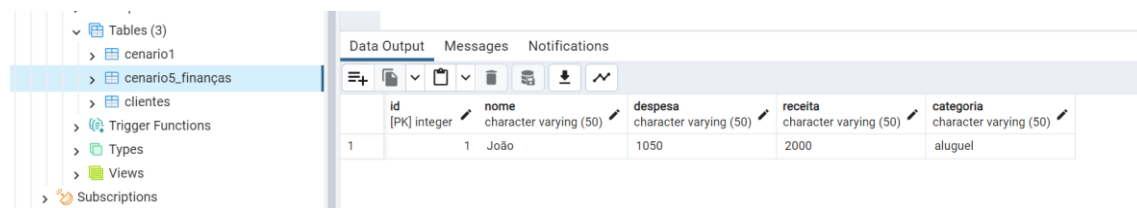
id	nome	despesa	receita	categoria
1	Maria	R\$1100	2000	aluguel

Etapa 3.0 Alteração de dados da tabela



```
1 BEGIN;  
2 UPDATE cenario5_financas SET nome = 'Maria' WHERE id = 1;  
3 UPDATE cenario5_financas SET despesa = '1050' WHERE id = 1;  
4 COMMIT;
```

Etapa 3.1 Confirmação da alteração dos dados



id	nome	despesa	receita	categoria
1	João	1050	2000	aluguel

Neste cenário, deparamo-nos com a necessidade de estabelecer um sistema de gestão financeira para uma empresa, visando o controle eficiente das finanças, incluindo registros de despesas, receitas, e categorias. Para atender a essa demanda, foram empregados comandos SQL para criar tabelas no banco de dados, formando uma estrutura coesa e organizada.

Cenário 7: Sistema de Gestão Escolar (SGBD Relacional)

Crie um sistema de gestão escolar, onde é necessário armazenar informações sobre alunos, professores, disciplinas e notas. A estrutura deve seguir um modelo relacional para garantir a consistência das informações acadêmicas.

Etapa 1 Criação da tabela

```
Query    Query History
1  CREATE TABLE cenario7_gestão_escola (
2  id SERIAL PRIMARY KEY,
3  aluno VARCHAR(50),
4  professor VARCHAR(50),
5  disciplina VARCHAR(50),
6  nota VARCHAR(5)
7  );|
```

Etapa 2 Confirmação da criação da tabela

The screenshot shows a database management tool interface. On the left, a tree view lists several databases, with 'cenario7_gestão_escola' selected. On the right, the 'Data Output' tab displays the schema of the selected table. The schema consists of five columns: 'id' (integer, primary key), 'aluno' (character varying (50)), 'professor' (character varying (50)), 'disciplina' (character varying (50)), and 'nota' (character varying (5)).

id	aluno	professor	disciplina	nota
[PK] integer	character varying (50)	character varying (50)	character varying (50)	character varying (5)

Etapa 3.0 Inserção de dados na tabela

```
Query    Query History
1  BEGIN;
2  INSERT INTO cenario7_gestão_escola (aluno,professor,disciplina,nota)
3  VALUES ('urilo','Diogo','SGBD','10');|
4  COMMIT;
```

Etapa 3.1 Confirmação da inserção de dados

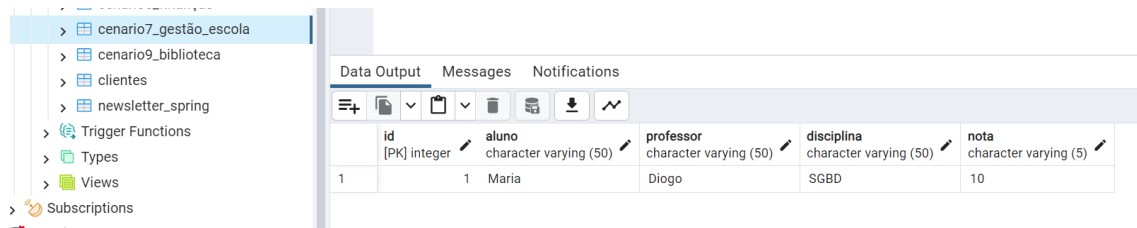
The screenshot shows the same database management tool interface as before. The 'Data Output' tab now displays the data inserted into the 'cenario7_gestão_escola' table. The data consists of one row with the following values: '1' for 'id', '1' for 'aluno', 'Murilo' for 'professor', 'Diogo' for 'disciplina', and 'SGBD' for 'nota'.

id	aluno	professor	disciplina	nota
[PK] integer	character varying (50)	character varying (50)	character varying (50)	character varying (5)
1	1	Murilo	Diogo	SGBD

Etapa 4.0 Alteração de dados da tabela

```
Query  Query History
1  BEGIN;
2  UPDATE cenário7_gestão_escola SET aluno = 'Maria' WHERE id = 1;
3  COMMIT;
```

Etapa 4.1 Confirmação da alteração dos dados



The screenshot shows a database management interface. On the left, a tree view lists several databases: cenário7_gestão_escola, cenário9_biblioteca, clientes, newsletter_spring, Trigger Functions, Types, Views, and Subscriptions. The 'cenário7_gestão_escola' database is selected. On the right, the 'Data Output' tab is active, displaying the contents of the 'cenário7_gestão_escola' table. The table has six columns: id (integer, primary key), aluno (character varying (50)), professor (character varying (50)), disciplina (character varying (50)), and nota (character varying (5)). The data row shows id=1, aluno='Maria', professor='Diogo', disciplina='SGBD', and nota='10'.

id	aluno	professor	disciplina	nota
1	Maria	Diogo	SGBD	10

Neste cenário, surge a necessidade de implementar um sistema de gestão escolar para uma instituição de ensino, buscando eficiência no controle das atividades acadêmicas, incluindo registros de alunos, professores, disciplinas e notas. Para atender a essa demanda, foram utilizados comandos SQL para criar tabelas no banco de dados, resultando em uma estrutura coesa e organizada.

Cenario 9:

Crie um sistema para o controle de uma biblioteca, onde é necessário gerenciar informações

sobre livros, autores, leitores e empréstimos. A estrutura deve seguir um modelo relacional

para garantir a integridade das informações bibliotecárias.

Etapa 1.0 Criação da tabela

```
Query    Query History
1  CREATE TABLE cenario9_biblioteca (
2  id SERIAL PRIMARY KEY,
3  Livro VARCHAR(50),
4  Autor VARCHAR(50),
5  Leitor VARCHAR(50),
6  EmpréstimoData VARCHAR(50)
7  );
```

Etapa 1.1 Confirmação da criação da tabela.

Tables (4) cenario1 cenario5_finanças cenario9_biblioteca clientes Trigger Functions	Data Output Messages Notifications				
	id [PK] integer	livro character varying (50)	autor character varying (50)	leitor character varying (50)	empréstimodata character varying (50)

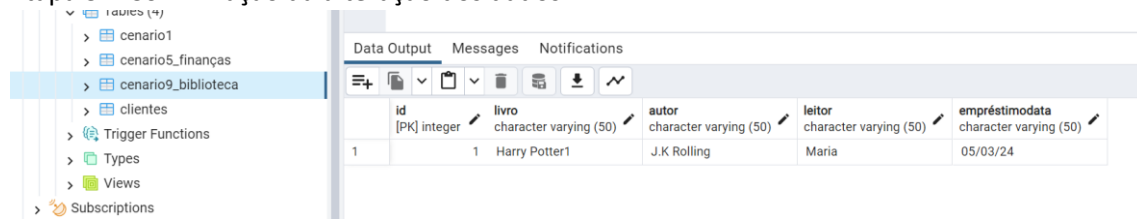
Etapa 2. Inserção de dados na tabela.

```
Query    Query History
1  BEGIN;
2  INSERT INTO cenario9_bibliotecas (livro,autor, leitor, empréstimodata)
3  VALUES ('Harry Potter - 1', 'J.K Rolling', 'João', '04/03/24');
4  COMMIT;
```

Etapa 3. Alteração de dados na tabela.

```
Query    Query History
1  BEGIN;
2  UPDATE cenario9_biblioteca SET leitor = 'Maria' WHERE id = 1;
3  UPDATE cenario9_biblioteca SET empréstimodata = '05/03/24' WHERE id = 1;
4  COMMIT;
```

Etapa 3.1 Confirmação da alteração dos dados.



The screenshot shows a database management interface. On the left, a tree view lists various database objects, with 'cenario9_biblioteca' selected. The main area displays the 'Data Output' tab for this table. The table has five columns: 'id' (integer, primary key), 'livro' (character varying (50)), 'autor' (character varying (50)), 'leitor' (character varying (50)), and 'empréstimodata' (character varying (50)). A single record is shown with the following values: id=1, livro=Harry Potter1, autor=J.K Rolling, leitor=Maria, and empréstimodata=05/03/24.

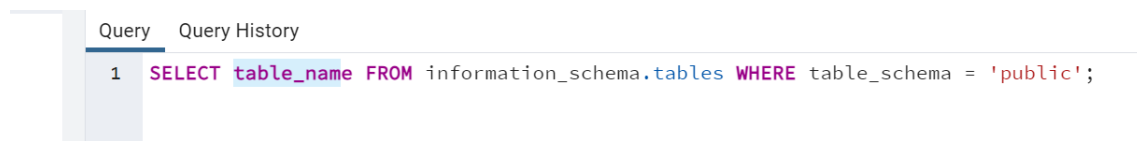
	id [PK] integer	livro character varying (50)	autor character varying (50)	leitor character varying (50)	empréstimodata character varying (50)
1	1	Harry Potter1	J.K Rolling	Maria	05/03/24

Neste contexto, surge a necessidade de estabelecer um sistema de gestão para uma biblioteca, visando um controle eficiente dos recursos bibliográficos, incluindo registros de livros, autores, leitores e datas de empréstimo. Para atender a essa demanda, foram utilizados comandos SQL para criar tabelas no banco de dados, proporcionando uma estrutura coesa e organizada.

Comando SQL para listar todas as tabelas do banco

SELECT table_name FROM information_schema.tables WHERE table_schema = 'public';

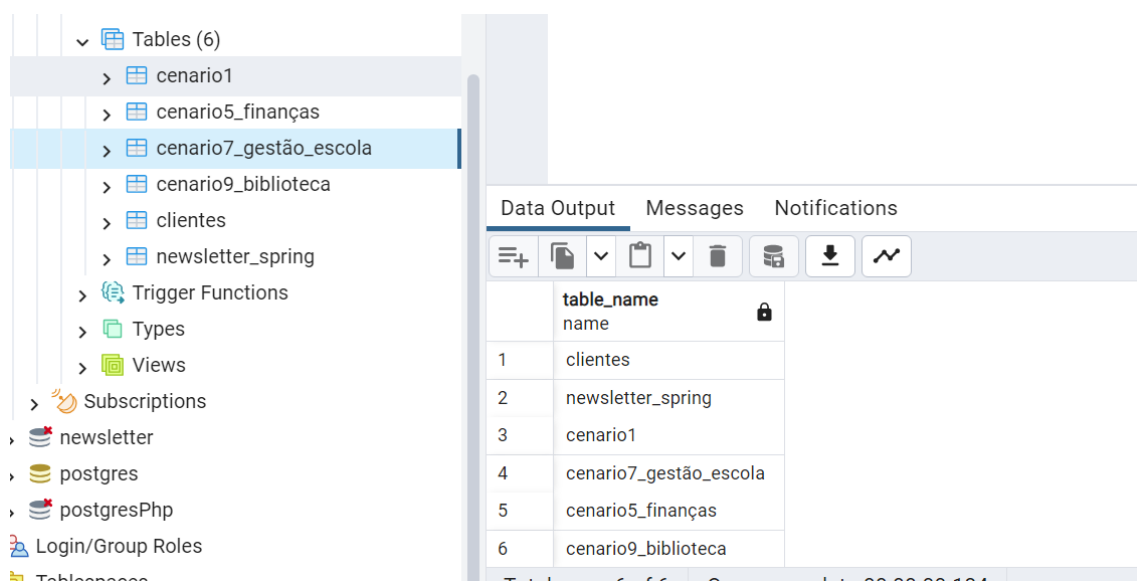
Etapa 1 Comando para listar todas as tabelas do banco



The screenshot shows a database management interface with the 'Query' tab selected. A SQL query is entered in the text area: 'SELECT table_name FROM information_schema.tables WHERE table_schema = 'public';'. The query is numbered '1'.

Query	Query History
1	SELECT table_name FROM information_schema.tables WHERE table_schema = 'public';

Etapa 2 Lista das tabelas no banco



The screenshot shows a database management interface. On the left, a tree view lists various database objects, with 'cenario7_gestão_escola' selected. The main area displays the 'Data Output' tab for the query. The table has two columns: 'table_name' and 'name'. Six records are shown, listing the names of the tables in the public schema: clientes, newsletter_spring, cenario1, cenario7_gestão_escola, cenario5_finanças, and cenario9_biblioteca.

	table_name	name
1	clientes	
2	newsletter_spring	
3	cenario1	
4	cenario7_gestão_escola	
5	cenario5_finanças	
6	cenario9_biblioteca	