

Prática Avaliada 1 – Administração de Banco de Dados

Objetivo

Avaliar a capacidade do aluno em:

- Compreender o modelo de dados do minimundo Chaves Imóveis.
- Utilizar comandos SQL para manipulação e consulta de dados.
- Gerenciar usuários e permissões no PostgreSQL.

Instruções Gerais

Ambiente:

Utilize o PostgreSQL (pode ser local ou via Docker). Você deve usar o script de criação de tabelas e dados fornecido pelo professor conforme o modelo do minimundo Chaves Imóveis.

Entrega:

- Envie um arquivo .sql com todos os comandos utilizados e comentários explicativos.
- Inclua um arquivo .txt com as respostas das consultas (copie e cole os resultados).
- Organize os comandos e respostas por item.

Recursos:

- Consulte a documentação oficial do PostgreSQL e material de aula se necessário.
- Não é permitido o trabalho em grupo.

Atividades

- 1) Criação de Usuários e Roles (20 pontos)
 - a. Crie um usuário chamado corretor com senha.

CREATE USER corretor WITH PASSWORD 'SenhaForte123';

The screenshot shows a PostgreSQL terminal window. The command 'CREATE USER corretor WITH PASSWORD 'SenhaForte123'' was entered. The response in the 'Messages' tab indicates an error: 'ERROR: role "corretor" already exists'. The 'SQL state' is shown as '42710'.

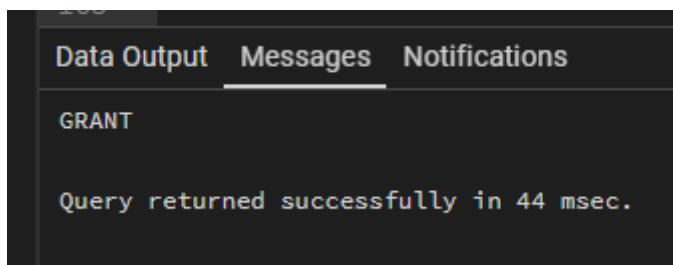
- b. Crie uma role chamada gerente sem login.

CREATE ROLE gerente NOLOGIN;

The screenshot shows a PostgreSQL terminal window. The command 'CREATE ROLE gerente NOLOGIN;' was entered. The response in the 'Messages' tab indicates an error: 'ERROR: role "gerente" already exists'. The 'SQL state' is shown as '42710'.

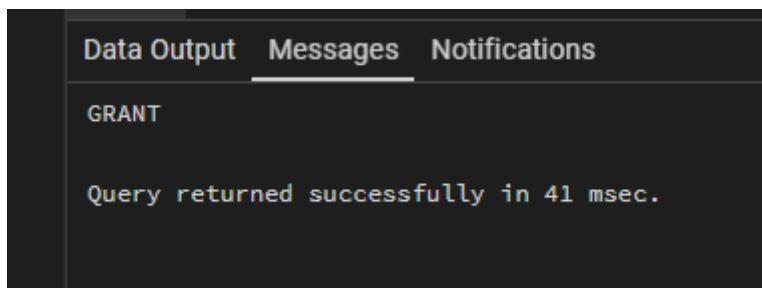
- c. Conceda à role gerente permissão total (DDL e DML) sobre todas as tabelas do banco.

```
GRANT CREATE ON SCHEMA public TO gerente;
```



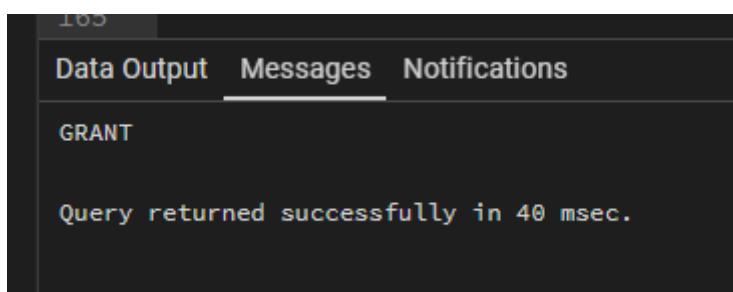
```
105
Data Output Messages Notifications
GRANT
Query returned successfully in 44 msec.
```

```
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO gerente;
```



```
105
Data Output Messages Notifications
GRANT
Query returned successfully in 41 msec.
```

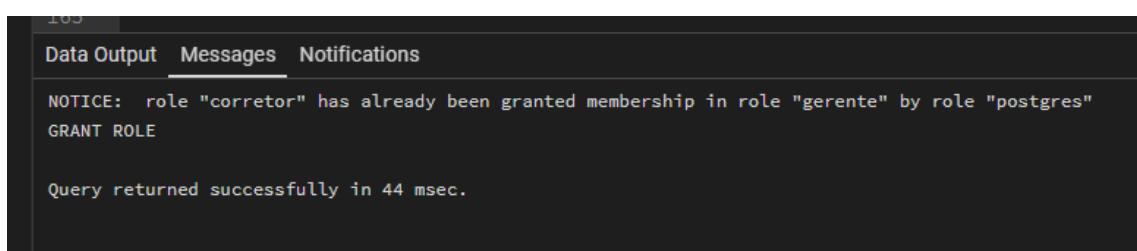
```
GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA public TO gerente;
```



```
105
Data Output Messages Notifications
GRANT
Query returned successfully in 40 msec.
```

- d. Adicione o usuário corretor à role gerente.

```
GRANT gerente TO corretor;
```



```
105
Data Output Messages Notifications
NOTICE:  role "corretor" has already been granted membership in role "gerente" by role "postgres"
GRANT ROLE
Query returned successfully in 44 msec.
```

- 2) Consultas SQL (40 pontos) utilize o modelo de dados do minimundo Chaves Imóveis para responder:
- Liste todos os imóveis disponíveis para aluguel, mostrando endereço, valor e nome do proprietário.

```
SELECT i.endereco, i.bairro, i.valor_aluguel, p.nome AS proprietario
```

```

FROM imoveis i
LEFT JOIN proprietarios p ON i.proprietario_id = p.id
WHERE i.disponivel_aluguel = TRUE
ORDER BY i.endereco;

```

	endereco character varying (200)	bairro character varying (100)	valor_aluguel numeric (12,2)	proprietario character varying (100)
1	Av B, 200	Jardim	2500.00	João Pereira
2	Rua A, 100	Centro	1500.00	Mariana Silva
3	Rua D, 400	Vila Nova	1200.00	Ana Costa

b. Mostre o nome dos corretores que não realizaram nenhuma venda.

```

SELECT c.nome
FROM corretores c
LEFT JOIN vendas v ON c.id = v.corretor_id
WHERE v.id IS NULL;

```

	nome character varying (100)
1	Rafael Alves

c. Exiba o total de imóveis vendidos por cada corretor, ordenando do maior para o menor.

```

SELECT c.nome AS corretor, COUNT(v.id) AS total_vendas
FROM corretores c
LEFT JOIN vendas v ON c.id = v.corretor_id
GROUP BY c.nome
ORDER BY total_vendas DESC;

```

101 com clientes cl

Data Output Messages Notifications

	corretor character varying (100)	total_vendas bigint
1	Carlos Souza	2
2	Fernanda Lima	1
3	Rafael Alves	0

- d. Liste os clientes que alugaram imóveis mais de uma vez, mostrando o nome e a quantidade de aluguéis.

```
SELECT cl.nome AS cliente, COUNT(a.id) AS qtd_alugueis
FROM clientes cl
JOIN alugueis a ON cl.id = a.cliente_id
GROUP BY cl.nome
HAVING COUNT(a.id) > 1
ORDER BY qtd_alugueis DESC;
```

200 (D) Realize o valor do aluguel

Data Output Messages Notifications

	cliente character varying (100)	qtd_alugueis bigint
1	Pedro Santos	3

3) Manipulação de Dados (20 pontos)

- a. Insira um novo imóvel para aluguel, associando a um proprietário já existente.

```
INSERT INTO imoveis (endereco, bairro, valor_aluguel, disponivel_aluguel, proprietario_id) VALUES
('Rua Nova, 77', 'Centro', 2100.00, TRUE, 2);
```

Data Output Messages Notifications

```
INSERT 0 1
Query returned successfully in 55 msec.
```

- b. Atualize o valor do aluguel de todos os imóveis localizados em um determinado bairro, aumentando em 10%.

```
UPDATE imoveis SET valor_aluguel = ROUND(valor_aluguel * 1.10, 2) WHERE bairro = 'Jardim';
```

The screenshot shows a database interface with a 'Messages' tab selected. The message area displays the result of an UPDATE query: 'UPDATE 2' followed by 'Query returned successfully in 48 msec.'

- c. Remova todos os registros de aluguel com data anterior a 2020.

```
DELETE FROM alugueis
```

```
WHERE data_contrato < DATE '2020-01-01';
```

The screenshot shows a database interface with a 'Messages' tab selected. The message area displays the result of a DELETE query: 'DELETE 1' followed by 'Query returned successfully in 40 msec.'

4) Segurança e Permissões (20 pontos)

- a. Revogue da role gerente a permissão de remover (DELETE) registros das tabelas.

```
REVOKE DELETE ON ALL TABLES IN SCHEMA public FROM gerente;
```

The screenshot shows a database interface with a 'Messages' tab selected. The message area displays the result of a REVOKE query: 'REVOKE' followed by 'Query returned successfully in 39 msec.'

- b. Conceda ao usuário corretor permissão apenas de leitura (SELECT) sobre a tabela de clientes.

```
GRANT SELECT ON clientes TO corretor;
```

A screenshot of a database terminal window. At the top, there are three tabs: "Data Output", "Messages" (which is underlined), and "Notifications". Below the tabs, the word "GRANT" is typed. At the bottom of the window, the message "Query returned successfully in 39 msec." is displayed.

```
GRANT
Query returned successfully in 39 msec.
```

Observações

- Comente cada comando explicando o que ele faz.
- Certifique-se de que os comandos executam sem erros.
- Utilize transações (BEGIN/COMMIT) quando necessário.