

1. Criando um Usuário e Concedendo Acesso Básico

Primeiro, você deve criar o usuário aluno e conceder a ele a capacidade de fazer login e se conectar ao banco de dados restaurante.

- **Comando de criação do usuário:**

SQL

```
CREATE USER aluno WITH PASSWORD '1234';
```

- **Explicação:** Este comando cria um usuário (USER) chamado aluno e define sua senha (PASSWORD).

- **Comando para conceder permissão de conexão:**

SQL

```
GRANT CONNECT ON DATABASE restaurante TO aluno;
```

- **Explicação:** A permissão CONNECT é necessária para que o usuário aluno possa se conectar ao banco de dados restaurante. Sem essa permissão, ele não conseguirá acessar o banco de dados.

2. Concedendo Privilégios de Leitura (SELECT)

Para permitir que o aluno visualize os dados de todas as tabelas, mas não os altere, você pode conceder o privilégio SELECT em todas as tabelas.

- **Comando para conceder privilégios de leitura:**

SQL

```
GRANT SELECT ON ALL TABLES IN SCHEMA public TO aluno;
```

- **Explicação:** Este comando concede permissão (GRANT) para SELECT em todas as tabelas (ALL TABLES) no schema public para o usuário aluno. O schema public é o padrão e é onde as tabelas do seu script foram criadas¹.

3. Removendo Privilégios

Para demonstrar como revogar permissões, você pode remover o acesso de leitura que acabou de conceder.

- **Comando para remover privilégios de leitura:**

SQL

```
REVOKE SELECT ON ALL TABLES IN SCHEMA public FROM aluno;
```

- **Explicação:** O comando REVOKE é o oposto do GRANT. Ele remove a permissão SELECT de todas as tabelas do schema public para o usuário aluno.

4. Concedendo Privilégios para Tabelas Específicas

Para uma abordagem mais granular, você pode dar ao aluno permissão de leitura apenas em tabelas específicas, como mesas e produtos.

- **Comandos para conceder permissões em tabelas específicas:**

SQL

```
GRANT SELECT ON TABLE mesas TO aluno;
```

```
GRANT SELECT ON TABLE produtos TO aluno;
```

- **Explicação:** Esses comandos concedem a permissão de SELECT apenas para as tabelas mesas e produtos, respectivamente. O aluno não poderá visualizar dados de outras tabelas, como funcionarios ou vendas.

5. Usando Roles para Gerenciamento de Permissões

Uma "role" no PostgreSQL pode ser usada como um "grupo" para gerenciar permissões, o que simplifica a administração, especialmente quando há muitos usuários².

- **O que é uma Role?**

Uma role é um objeto que pode conter privilégios. Em vez de conceder permissões a cada usuário individualmente, você pode concedê-las a uma role e, em seguida, adicionar usuários a essa role. Isso simplifica o gerenciamento: se você precisar adicionar ou remover permissões, basta modificar a role, e todos os usuários que pertencem a ela serão afetados automaticamente.

- **Criando e Usando a Role libertas:**

Vamos criar a role libertas e usar o usuário aluno como membro.

- **Comando para criar a role:**

SQL

```
CREATE ROLE libertas;
```

- **Comando para adicionar o usuário à role:**

SQL

```
GRANT libertas TO aluno;
```

- **Explicação:** Este comando faz com que o usuário aluno seja um membro da role libertas⁵.

- **Concedendo privilégios à role:**

Agora, em vez de conceder privilégios ao usuário aluno, você os concede à role libertas.

SQL

```
GRANT SELECT ON ALL TABLES IN SCHEMA public TO libertas;
```

```
GRANT INSERT, UPDATE, DELETE ON TABLE vendas TO libertas;
```

- **Explicação:** O primeiro comando concede permissão de leitura em todas as tabelas do schema public para a role libertas. O segundo comando concede permissões de inserção, atualização e exclusão apenas na tabela vendas para a role⁶.

- **Verificação:**

O usuário aluno agora tem acesso de leitura a todas as tabelas e acesso total de escrita na tabela vendas, porque ele é um membro da role libertas.

- **Gerenciando a Role:**

- **Para remover um privilégio de todos os membros:**

SQL

```
REVOKE SELECT ON TABLE mesas FROM libertas;
```

Isso retira a permissão de SELECT da role libertas na tabela mesas.

- **Para remover um usuário da role:**

SQL

```
REVOKE libertas FROM aluno;
```

Isso remove o usuário aluno da role libertas, e ele perde todos os privilégios herdados dela.

Para conceder permissões gerais a role libertas no DB restaurante.

1. Concedendo Privilégios de Acesso ao Banco de Dados

O primeiro passo é garantir que a role libertas tenha todos os privilégios no banco de dados restaurante. Isso inclui a capacidade de criar, modificar e excluir schemas, bem como outras operações de alto nível.

- **Comando para conceder todos os privilégios no banco de dados:**

SQL

```
GRANT ALL PRIVILEGES ON DATABASE restaurante TO libertas;
```

- **Explicação:** Este comando concede todas as permissões no nível do banco de dados, como CREATE (para criar schemas e tabelas), CONNECT, TEMPORARY e outras.

2. Concedendo Privilégios em Schemas

Para que a role libertas possa manipular os objetos dentro dos schemas (como o schema padrão public), você precisa conceder privilégios de CREATE e USAGE no schema.

- **Comando para conceder privilégios no schema public:**

SQL

```
GRANT ALL ON SCHEMA public TO libertas;
```

- **Explicação:** Esta permissão concede o direito de criar objetos (CREATE) e usar (USAGE) o schema public.

3. Concedendo Privilégios em Tabelas Existentes

Para garantir que a role libertas (e, por herança, o usuário aluno) tenha acesso total a todas as tabelas já existentes no banco de dados, você deve conceder ALL PRIVILEGES nas tabelas.

- **Comando para conceder todos os privilégios em tabelas existentes:**

SQL

```
GRANT ALL ON ALL TABLES IN SCHEMA public TO libertas;
```

- **Explicação:** Este comando concede todas as permissões, incluindo SELECT, INSERT, UPDATE, DELETE, TRUNCATE, REFERENCES e TRIGGER, em todas as tabelas que já existem no schema public.

4. Concedendo Privilégios em Tabelas Futuras

Para garantir que a role libertas também tenha acesso total a qualquer tabela que seja criada no futuro, use o ALTER DEFAULT PRIVILEGES.

- **Comando para conceder privilégios padrão para tabelas futuras:**

SQL

```
ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT ALL ON TABLES TO libertas;
```

- **Explicação:** Esta configuração garante que, quando um usuário criar uma nova tabela no schema public, a role libertas receberá automaticamente todos os privilégios sobre ela.

Ao executar todos esses comandos, a role libertas terá acesso total ao banco de dados restaurante, e o usuário aluno, como membro da role, herdará todos esses privilégios.