Privilégios

A permissão para realizar operações com os objetos do banco de dados depende dos privilégios que o usuário tem sobre o objeto.

O dono de um objeto pode realizar qualquer operação sobre o objeto. Para os demais usuários é necessário atribuir os privilégios ao usuário ou a um grupo ao qual o usuário pertença.

Usuários que possuem o atributo de superusuário podem acessar qualquer objeto.

Privilégios

Podem ser atribuídos os seguintes privilégios:

SELECT - Permite consultar qualquer coluna, ou as colunas especificadas da tabela, visão ou seqüência especificada. Também permite utilizar o comando COPY TO. Para as seqüências, este privilégio também permite o uso da função currval.

INSERT - Permite inserir novas linhas na tabela especificada. Se forem relacionadas colunas específicas, apenas essas colunas podem ser referencias no comando INSERT, as demais colunas receberão o valor default. Também permite utilizar o comando COPY FROM.

UPDATE - Permite modificar os dados de qualquer coluna, ou das colunas especificadas da tabela especificada. Para as seqüências, este privilégio permite o uso das funções nextval e setval.

DELETE - Permite excluir linhas da tabela especificada.

TRUNCATE - Permite truncar a tabela especificada.

Privilégios

REFERENCES - Para criar uma restrição de chave estrangeira é necessário possuir este privilégio, tanto na tabela que faz referência quanto na tabela que é referenciada.

TRIGGER - Permite criar gatilhos na tabela especificada.

CREATE - Para bancos de dados, permite a criação de novos esquemas no banco de dados. Para esquemas, permite a criação e renomeação de novos objetos no esquema. Para espaços de tabelas, permite a criação de tabelas e índices no espaço de tabelas, e permite a criação de bancos de dados possuindo este espaço de tabelas como seu espaço de tabelas padrão.

CONNECT – Permite a conexão ao banco de dados.

TEMPORARY ou **TEMP** - Permite a criação de tabelas temporárias ao usar o banco de dados.

EXECUTE - Permite utilizar a função especificada e qualquer operador implementado utilizando a função.

Privilégios

USAGE - Para as linguagens procedurais, permite o uso da linguagem especificada para criar funções nesta linguagem. Para os esquemas, permite acessar os objetos contidos no esquema especificado. Para as sequências, esse privilégio permite o uso das funções currval e nextval.

ALL [PRIVILEGES] - Concede todos os privilégios disponíveis de uma só vez. A palavra chave PRIVILEGES é opcional no PostgreSQL, embora seja requerida pelo padrão SQL.

Papéis

Para facilitar o gerenciamento dos privilégios dos usuários, os privilégios podem ser atribuídos por grupo e os usuários terão os mesmos privilégios dos grupos aos quais pertencerem.

No PostgreSQL, tanto usuários quanto grupos são papéis (roles). A diferença é que usuários podem fazer login no banco de dados e grupos não podem.

Existem comandos separados para criação de usuários (CREATE USER) e para criação de grupos (CREATE GROUP), mas no PostgreSQL os dois comandos são apenas aliases para o comando para criação de papéis (CREATE ROLE).

CREATE ROLE

O comando **CREATE ROLE** adiciona um novo papel ao agrupamento de bancos de dados do PostgreSQL. Apenas os superusuários do banco de dados podem usar este comando.

```
CREATE ROLE nome [ [ WITH ] opção [ ... ] ]
```

As opções disponíveis são as mesmas que para o comando **CREATE USER**. A diferença entre os dois comandos é que para **CREATE USER**, o padrão é assumir a opção **LOGIN**, enquanto **NOLOGIN** é o padrão para **CREATE ROLE**.

Exemplo:

```
CREATE ROLE grupo;
```

ALTER ROLE

O comando ALTER ROLE altera um papel do banco de dados.

```
ALTER ROLE name [ [ WITH ] option [ ... ] ]
```

Onde option pode ser:

```
      SUPERUSER | NOSUPERUSER
    | CREATEDB | NOCREATEDB
    | CREATEROLE | NOCREATEROLE
    | CREATEUSER | NOCREATEUSER
    | INHERIT | NOINHERIT
    | LOGIN | NOLOGIN
    | CONNECTION LIMIT connlimit
    | [ ENCRYPTED | UNENCRYPTED ] PASSWORD 'password'
    | VALID UNTIL 'timestamp'
```

Exemplo:

```
ALTER ROLE teste WITH PASSWORD 'teste';
```

ALTER GROUP

Para alterar os atributos de um grupo, adicionar ou remover usuários de um grupo, pode ser utilizado o comando ALTER GROUP.

Como grupos e usuários foram unificados como papéis, essas operações também podem ser realizadas com os comandos ALTER ROLE, GRANT e REVOKE.

Para acrescentar um usuário em um grupo utiliza-se:

```
ALTER GROUP groupname ADD USER username [, ... ]
```

Exemplo:

```
ALTER GROUP grupo ADD USER teste;
```

Para remover um usuário em um grupo utiliza-se:

```
ALTER GROUP groupname DROP USER username [, ... ]
```

Exemplo:

```
ALTER GROUP grupo DROP USER teste;
```

GRANT

O comando GRANT permite atribuir privilégios para um papel. Para atribuir privilégios para todos usuários é utilizado PUBLIC no lugar do nome do usuário e para que o usuário/grupo ao qual foi atribuído privilégio passe a ter permissão para atribuir privilégios para o objeto, é utilizada a cláusula WITH GRANT OPTION.

GRANT

Para incluir um papel em um grupo utiliza-se:

```
GRANT role [, ...] TO rolename [, ...] [ WITH ADMIN OPTION ]
```

Exemplo:

```
GRANT grupo TO teste;
```

GRANT

Para atribuir privilégios para uma tabela ou todas tabelas de um esquema utiliza-se:

```
GRANT { { SELECT | INSERT | UPDATE | DELETE | TRUNCATE | REFERENCES
    | TRIGGER } [,...] | ALL [ PRIVILEGES ] }
    ON { [ TABLE ] tablename [, ...]
         | ALL TABLES IN SCHEMA schema_name [, ...] }
    TO { [ GROUP ] rolename | PUBLIC } [, ...] [ WITH GRANT OPTION ]
```

Exemplo:

```
GRANT SELECT ON aluno TO grupo;
```

GRANT

Para atribuir privilégios para colunas de uma tabela utiliza-se:

```
GRANT { { SELECT | INSERT | UPDATE | DELETE | REFERENCES }
    ( column_name [, ...] ) [,...] | ALL [ PRIVILEGES ] }
    ON { [ TABLE ] tablename [, ...]
    TO { [ GROUP ] rolename | PUBLIC } [, ...] [ WITH GRANT OPTION ]
```

Exemplo:

```
GRANT SELECT ( codigo, nome ) ON cliente TO grupo;
```

GRANT

Para atribuir privilégios em uma sequência utiliza-se:

```
GRANT { { USAGE | SELECT | UPDATE }
    [,...] | ALL [ PRIVILEGES ] }
    ON SEQUENCE sequencename [, ...]
    TO { [ GROUP ] rolename | PUBLIC } [, ...] [ WITH GRANT OPTION ]
```

GRANT

Para atribuir privilégios em um banco de dados utiliza-se:

```
GRANT { {CREATE|CONNECT|TEMPORARY|TEMP} [,...] | ALL [PRIVILEGES] }
    ON DATABASE dbname [, ...]
    TO { [ GROUP ] rolename | PUBLIC } [, ...] [ WITH GRANT OPTION ]
```

GRANT

Para atribuir privilégios em uma função ou todas as funções de um esquema utiliza-se:

```
GRANT { EXECUTE | ALL [ PRIVILEGES ] }
    ON { FUNCTION funcname ([[ argmode ] [ argname ] argtype[,...]])
       [,...] | ALL FUNCTIONS IN SCHEMA schema_name [, ...] }
    TO { [ GROUP ] rolename | PUBLIC } [, ...] [ WITH GRANT OPTION ]
```

GRANT

Para atribuir privilégios em uma linguagem utiliza-se:

```
GRANT { USAGE | ALL [ PRIVILEGES ] }
    ON LANGUAGE langname [, ...]
    TO { [ GROUP ] rolename | PUBLIC } [, ...] [ WITH GRANT OPTION ]
```

GRANT

Para atribuir privilégios em um esquema utiliza-se:

```
GRANT { { CREATE | USAGE } [,...] | ALL [ PRIVILEGES ] }
    ON SCHEMA schemaname [, ...]
    TO { [ GROUP ] rolename | PUBLIC } [, ...] [ WITH GRANT OPTION ]
```

GRANT

Para atribuir privilégios em espaço de tabelas utiliza-se:

```
GRANT { CREATE | ALL [ PRIVILEGES ] }
    ON TABLESPACE tablespacename [, ...]
    TO { [ GROUP ] rolename | PUBLIC } [, ...] [ WITH GRANT OPTION ]
```

REVOKE

O comando REVOKE permite revogar os privilégios de um papel.

Se for especificado GRANT OPTION FOR somente a opção de concessão do privilégio é revogada, e não o próprio privilégio. Caso contrário, tanto o privilégio quanto a opção de concessão serão revogados.

Se o usuário possui um privilégio com opção de concessão, e concedeu este privilégio para outros usuários, então os privilégios que estes outros usuários possuem são chamados de privilégios dependentes. Se o privilégio ou a opção de concessão que o primeiro usuário possui for revogada, e existirem privilégios dependentes, estes privilégios dependentes também serão revogados se for especificado CASCADE, senão a ação de revogar falhará. Esta revogação recursiva somente afeta os privilégios que foram concedidos através de uma cadeia de usuários começando pelo usuário objeto deste comando REVOKE. Portanto, os usuários afetados poderão manter o privilégio, se o privilégio também tiver sido concedido por outros usuários.

REVOKE

Exemplo:

```
REVOKE SELECT ON aluno FROM grupo;
```

ALTER DEFAULT PRIVILEGES

O comando **ALTER DEFAULT PRIVILEGES** permite alterar os privilégios que serão atribuídos por padrão para as novas tabelas, views, sequências, funções e tipos que forem criados. Esse comando não afeta os privilégios dos objetos já existentes.

```
ALTER DEFAULT PRIVILEGES
    [ FOR { ROLE | USER } target_role [, ...] ]
    [ IN SCHEMA schema_name [, ...] ]
    abbreviated_grant_or_revoke
```

Onde `abbreviated_grant_or_revoke` especifica os privilégios padrão para os objetos.

Exemplo:

```
ALTER DEFAULT PRIVILEGES GRANT SELECT ON TABLES TO grupo;
```