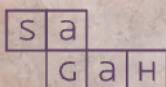


MODELAGEM E DESENVOLVIMENTO DE BANCO DE DADOS

Pedro Henrique Chagas Freitas



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS



Linguagem de controle de dados: DCL (*Data Control Language*)

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Identificar a linguagem DCL.
- Exemplificar a linguagem DCL.
- Implementar a linguagem DCL.

Introdução

Neste capítulo, você vai estudar a utilização das instruções DCL (*Data Control Language*) para implementação de permissionamento de acesso em bancos de dados. Para isso, você conhecerá as instruções GRANT e REVOKE. Enquanto a instrução GRANT é responsável por conceder permissão de acesso a um determinado perfil, o REVOKE tem por finalidade restringir o acesso a um perfil.

A linguagem DCL: conceitos

Nos bancos de dados, existem muitas tabelas que possuem dados estratégicos, dados que não podem ser divulgados ou até mesmo dados que não podem ser acessados por usuários sem privilégio (permissão) de acesso a esses dados. Pensando nisso, a linguagem SQL evoluiu para abarcar instruções de segurança e delimitação de acesso a dados, o que foi chamado de linguagem de controle de dados ou DCL (*Data Control Language*).

Por meio da linguagem DCL, é possível, então, implementar controles de segurança de acesso à tabela, restringindo e implementando privilégios de acesso a diferentes tipos de usuários de banco de dados. Isso, por sua vez, abre caminho para que tenhamos um alinhamento entre a área de segurança

da informação e a área de banco de dados, tendo em vista que as instruções de permissionamento de dados, na maioria dos casos, são funções da área de banco de dados, enquanto a implementação e a orquestração das políticas de segurança são responsabilidades da área de segurança da informação.

Assim, temos o GRANT como uma instrução de controle de dados que permite a concessão de privilégios (acessos) para um dado perfil de usuário em um banco de dados, podendo ser o acesso a um dado, a uma coluna, a uma tabela ou acesso ao próprio banco de dados.

A instrução REVOKE é o oposto do GRANT, servindo para restringir o acesso ou revogando a permissão concedida anteriormente por uma instrução DCL GRANT. É comum a utilização do REVOKE quando temos mudanças nas regras de negócio, ou seja, a estratégia do negócio muda e os privilégios de acessos a determinados grupos de usuários de bancos de dados muda; ou simplesmente um funcionário que tinha um dado privilégio de acesso sai da empresa ou é promovido e, a partir de uma dada situação, não pode mais ter acesso a uma tabela, a um banco de dados ou a um dado em uma determinada base de dados. Logo, a instrução REVOKE tem a finalidade de restringir acesso a um usuário de banco de dados ou a um perfil de banco de dados.

Logo, o DCL (*Data Control Language*) é uma linguagem que compõe o SQL e se baseia na definição de acessos a bases de dados, ou seja, por meio do DCL, temos uma implementação de segurança no nível de acesso ao banco de dados com os respectivos comandos:

- **GRANT:** concede permissões ao(s) usuário(s) do banco de dados ou aos perfis de usuários para realizar acesso ou operações nas bases de dados, tabelas, colunas, campos ou no próprio dado.
- **REVOKE:** retira ou restringe o acesso ou permissões de acesso de usuário(s) ou perfil(is) de usuários para realizar acesso ou operações nas bases de dados, tabelas, colunas, campos ou no próprio dado.

Assim, como mostra a Figura 1, as instruções DCL **GRANT** e **REVOKE** são responsáveis por conceder privilégios aos usuários de bancos de dados.



Figura 1. Privilégios das instruções DCL.



Fique atento

Como a instrução REVOKE é utilizada para remover privilégio de acesso aos objetos de uma base de dados, pressupõe-se que esses objetos tenham recebido determinado privilégio, ou seja, para remover um privilégio de acesso por meio do REVOKE, esse privilégio foi atribuído anteriormente por um GRANT. Assim, é correto pressupor que, primeiramente, houve um GRANT, ou seja, foram atribuídos privilégios de acesso a um usuário ou perfil de usuário e que, a partir disso, decide-se pela utilização do REVOKE para retirar os privilégios atribuídos. Assim sendo, não cabe falar de privilégios que não foram concedidos, e a ordem é que haja um GRANT para, posteriormente, haver um REVOKE se assim for necessário.

Quando falamos de perfis de usuários, estamos dizendo que temos vários tipos de usuários que acessam um banco de dados, além dos DBAs. Logo, é comum a criação de perfis, a partir dos quais são atribuídos acessos (GRANTS). Tomemos como exemplo o seguinte: Perfil Chefe_Divisão_RH terá acesso a todas as tabelas pertinentes à divisão de RH, bem como seus colaboradores dentro da divisão. Não faria sentido conceder acesso à base de dados Finanças ao Perfil Chefe_Divisão_RH. Obviamente, o chefe da divisão de RH pode, sim, solicitar acesso a dados financeiros, mas, em tese, não seria concedido acesso a uma base de dados específica a um perfil não pertinente à base de dados que é solicitada.

O oposto também pode ocorrer, de modo que podemos ter um coordenador da divisão de finanças que se tornou coordenador da divisão de RH, de modo

que não faz sentido ele ter mais acesso, pelo menos não o mesmo acesso que tinha quando era coordenador da divisão de finanças. Assim, os privilégios (acessos) devem ser restringidos (retirados) e, para isso, utilizamos o REVOKE.

Exemplificando a utilização da linguagem DCL

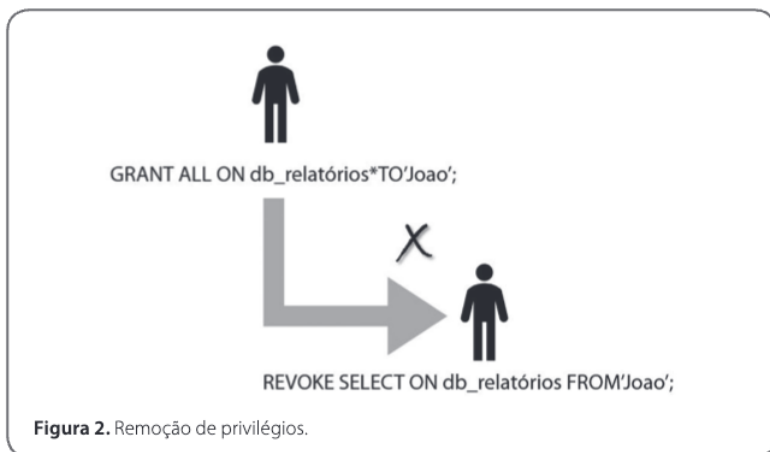
Se tivéssemos que atribuir acesso a um dado usuário, teríamos que utilizar a instrução GRANT para conceder acesso. Logo, primeiramente, precisamos criar o usuário:

```
CREATE USER 'Joao'  
IDENTIFIED BY '123456'  
GRANT ALL ON db_relatorios.* TO 'Joao';
```

Estamos dizendo, então, que João foi criado e que, na sua criação, já estamos concedendo acesso ao banco de dado de relatórios (db_relatorios). Caso João não pudesse mais realizar consultas à base de dados de relatórios, removeríamos o privilégio anteriormente dado:

```
REVOKE SELECT ON db_relatorios FROM 'Joao';
```

Nesse caso, são removidos os privilégios dados a João pertinentes à base de dados db_relatorios (Figura 2).



Implementando a linguagem DCL

Vamos apresentar, aqui, a sintaxe utilizada para a instrução GRANT na atribuição de privilégios para usuários em um banco de dados MySQL. Temos, então, os seguintes aspectos na implementação do GRANT para acesso em um banco de dados MySQL:

Caso desejássemos criar um usuário ainda inexistente no banco de dados, poderíamos atribuir, no momento da criação, por meio da instrução GRANT, algumas permissões de acesso. Inicialmente, se desejássemos somente que houvesse acesso a uma base de dados, faríamos:

```
CREATE USER 'Usuario1'@'localhost' IDENTIFIED BY 'password';  
GRANT ALL ON BancodeDados1.* TO 'Usuario1'@'localhost';
```

Nesse caso, estamos atribuindo ao usuário criado Usuario1 acesso à base de dados que chamamos de Banco de Dados1. Normalmente, um administrador de banco de dados (DBA) usa, primeiramente, CREATE USER para criar um usuário ou perfil e, depois, define suas características (privilégios ou não privilégios), como sua senha, por exemplo.

A instrução GRANT também garante o uso de conexões seguras por autorização de acesso, o que colabora no acesso aos recursos do servidor em que está o banco de dados. Isso ocorre porque, quando são utilizados privilégios a partir do GRANT, menos recursos ficam disponíveis para alocação imediata. É importante ressaltar, também, que a instrução ALTER USER pode ser usada para alterar as características de contas já existentes.

```
CREATE USER 'jeffrey'@'localhost' IDENTIFIED BY 'password';  
GRANT ALL ON db1.* TO 'jeffrey'@'localhost';
```

Temos, aqui, o IDENTIFIED BY, que garante que, no momento da criação do usuário, este será identificado por um *password*, aumentando, assim, o nível de segurança do usuário criado. Nesse caso, o usuário criado é Jeffrey, que terá acesso local ao banco de dados, chamado **db1**.

Teremos, então, os seguintes privilégios que podem ser combinados com a instrução DCL GRANT para atribuir privilégios a usuários (Quadro 1):

Quadro 1. Privilégios da instrução DCL

Privilégio	Níveis significativos e concedíveis
ALL [PRIVILEGES]	Concede todos os privilégios no nível de acesso especificado.
ALTER	Ativa o uso de ALTER TABLE para alteração de tabelas por usuários.
ALTER ROUTINE	Ativa as rotinas armazenadas para serem alteradas ou eliminadas.
CREATE	Ativar a criação de banco de dados e tabelas.
CREATE ROUTINE	Ativa a criação de rotina armazenada. Níveis: global, banco de dados.
CREATE TABLESPACE	Ativa os <i>tablespaces</i> e os grupos de arquivos de log para serem criados, alterados ou eliminados.
CREATE TEMPORARY TABLES	Ativa a criação de uma tabela temporária.
CREATE USER	Permite a criação de usuários, exclusão de usuário e renomear usuários.
CREATE VIEW	Ativa visões para serem criadas ou alteradas.
DELETE	Ativa a autorização para exclusão de dados.
DROP	Ativa a autorização para exclusão de bancos de dados e tabelas.
EVENT	Ativa a criação de eventos para o banco de dados.
EXECUTE	Permite que o usuário execute rotinas armazenadas.
FILE	Permite que o usuário faça com que o servidor leia ou grave arquivos.
GRANT OPTION	Ativa privilégios a serem concedidos ou removidos de outras contas.
INDEX	Ativa índices a serem criados ou eliminados.
INSERT	Ativa a autorização para inserção de dados.
PROXY	Ativa o proxy do usuário.
REFERENCES	Ativa criação de chave estrangeira.
SELECT	Ativa a criação de consultas.
UPDATE	Ativa o uso de atualizações nos dados.

Todas essas instruções podem ser combinadas com a instrução DCL GRANT, a fim de atribuir algum tipo de privilégio de acesso a um determinado usuário. O mesmo pode ser realizado com o REVOKE, que também pode ser combinado com essas instruções para retirar privilégios atribuídos a um usuário.

A instrução REVOKE nos permitiria restringir privilégios em um banco de dados, ou seja, a partir do REVOKE, privilégios são removidos de usuários em um banco de dados, como exemplificado a seguir:

```
REVOKE INSERT ON *db1* FROM 'jeffrey'@'localhost';
```

Nesse caso, estamos retirando a autorização para Jeffrey na base de dados db1. Para revogar todos os privilégios atribuídos a Jeffrey, faríamos:

```
REVOKE ALL PRIVILEGES
```

O REVOKE é uma instrução DCL que poderia ser combinada com todos os privilégios mostrados para o GRANT, a fim de retirá-los. Assim sendo, as instruções GRANT e REVOKE se complementam na linguagem DCL visando o controle de acesso nos bancos de dados.



Leituras recomendadas

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados*. 6. ed. São Paulo: Pearson, 2010.

HEUSER, C. A. *Projeto de banco de dados*. 6. ed. Porto Alegre: Bookman, 2010. (Série Livros Didáticos Informática UFRGS, 4).

KORTH, H. F.; SILBERSHATZ, A.; SUDARSHAN, S. *Sistemas de banco de dados*. 6. ed. Rio de Janeiro: Campus, 2012.

RAMAKRISHNAN, R. *Sistemas de gerenciamento de bancos de dados*. 3. ed. Porto Alegre: Penso, 2009.

SETZER, V. W. *Banco de dados: conceitos, modelos, gerenciadores, projeto lógico, projeto físico*. 3. ed. São Paulo: Blucher, 2002.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:

