ADMINISTRAÇÃO DE BANCO DE DADOS

Claudia Abreu Paes

SAGAH SOLUÇÕES EDUCACIONAIS INTEGRADAS

Gerenciamento de privilégios no banco de dados

Objetivos de aprendizagem

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

- Descrever o que são privilégios de um banco de dados.
- Reconhecer os comandos de permissões e suas consequências.
- Aplicar o uso de comandos de permissões do banco de dados.

Introdução

Um sistema de gerenciamento de banco de dados (SGBD) deve atender a quatro propriedades principais: consistência, concorrência, independência dos dados e segurança. A consistência garante a qualidade da informação armazenada, pois toda operação no banco de dados deve seguir as regras de restrição de integridade. Já a concorrência viabiliza a vários usuários o acesso simultâneo aos dados. Por sua vez, a independência dos dados facilita o desenvolvimento de *software*, gerando também qualidade. Por fim, a segurança entrega confidencialidade, integridade e disponibilidade dos dados.

Neste capítulo, você vai estudar os privilégios, que são essenciais para a garantia da segurança dos bancos de dados. Assim, você vai conhecer os tipos de permissões e ver como aplicá-las.

O que são privilégios?

Nas empresas, a informação é um capital de valor imensurável para os processos operacionais, gerenciais e estratégicos. Por esse motivo, cada vez mais é necessário dispor de recursos de segurança que protejam os dados de uma

empresa. Segundo Silberschatz, Korth e Sudarshan (2012), o sistema de banco de dados deve garantir a segurança das informações armazenadas, apesar das falhas do sistema ou das tentativas de acesso não autorizado. Se os dados são compartilhados entre vários usuários, o sistema deve evitar possíveis resultados anômalos.

Para Elmasri e Navathe (2011), a segurança de banco de dados deve resolver:

- questões legais e éticas relacionadas ao direito de acessar determinadas informações;
- questões políticas em nível governamental, institucional ou corporativo relativas a informações que devem ter sua confidencialidade mantida;
- questões relativas a níveis de sistema, como a definição da forma de tratamento da segurança, se ela será tratada no nível de *hardware*, no nível de sistema operacional ou no nível de SGBD;
- a necessidade de identificar níveis de segurança para a classificação dos dados.

Segundo Date (2003), a segurança de um banco de dados se refere à proteção de dados contra revelação, alteração ou destruição não autorizada, evitando perdas ou degradação de parte ou totalidade do banco de dados. Quando a segurança do banco de dados é violada, pode afetar:

- a integridade dos dados — informações imprecisas, fraudulentas ou errôneas;
- a confidencialidade dos dados — exposição não autorizada de informações confidenciais;
- a disponibilidade dos dados — impossibilidade de acesso aos dados quando necessário.

A segurança do banco de dados se dá de várias formas: criptografia dos dados, controle de acesso por meio da concessão de identificação e senha a cada usuário; atribuição de privilégios do tipo de operação (inclusão, alteração, exclusão, criação) a realizar nos objetos disponíveis no banco de dados; disponibilização de visão.

Saiba mais

No banco de dados, uma visão (*view*) é o conjunto de informações gerado a partir de uma consulta a uma ou várias tabelas. O SGBD não armazena o resultado de uma visão, somente sua estrutura. Por isso, as informações são sempre atualizadas no uso, já que só são disponibilizadas quando requeridas. As visões estão relacionadas à segurança porque cada usuário tem visibilidade somente das informações necessárias e adequadas a ele. A criação de uma visão no SGBD é efetivada com o comando CREATE VIEW.

Privilégios são permissões atribuídas a um usuário ou grupo de usuários. Tais permissões restringem ou permitem o acesso aos dados de acordo com o perfil definido junto ao adminitrador do banco de dados (*DataBase Administrator* [DBA]). Por meio dos privilégios, é possível autorizar o usuário a alcançar determinado recurso do banco de dados. Silberschatz, Korth e Sudarshan (2012) relacionam quatro formas de autorização: leitura, inserção, alteração e exclusão de dados. Utiliza-se uma ou a combinação delas na relação ou visão.

O DBA é o responsável por realizar as tarefas que implementam a segurança, como: criação e eliminação de usuários; concessão e remoção de privilégios; e criação das regras atendendo à política de segurança. Para o controle de acesso, os privilégios são verificados na tentativa de uso do recurso em tabelas próprias do banco de dados, utilizadas para armazenamento. A primeira validação realizada é em relação à autenticação do usuário e, em seguida, às autorizações atribuídas a ele no recurso desejado. O controle de acesso se dá conforme a Figura 1.

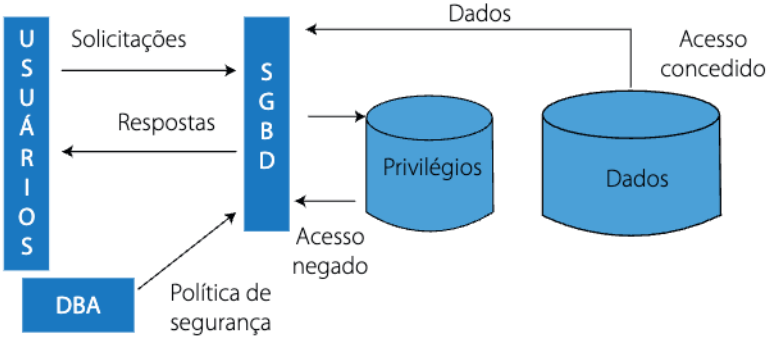

Figura 1. Controle de acesso.

As solicitações feitas pelo usuário serão verificadas nas tabelas de privilégios, por meio do SGBD, e autorizadas ou não. Estando autorizadas, as solicitações são atendidas. Caso contrário, o SGBD apresenta uma mensagem de erro ao usuário.

Link

No caso do SGBD MySQL, a tabela de permissões é carregada na memória no momento da inicialização. Dessa forma, a concessão de permissão é atribuída com base na versão que está em memória. As alterações causadas a partir de instruções de gerenciamento de conta fazem com que o servidor recarregue, automaticamente, a tabela de permissões para a memória. Tais alterações incluem: criação e remoção de usuário e grupos, alteração de senha, atribuição de privilégios a grupos e usuários. Caso o recarregamento não ocorra automaticamente, o DBA deve solicitá-lo por meio de operações de limpeza (*flush*) ou recarga (*reload*). Você pode obter mais informações no *link* a seguir.

https://qrgo.page.link/uXHP

As permissões podem ser atribuídas aos dados e esquemas. O usuário que possui privilégio de criação torna-se proprietário de objetos e pode atribuir privilégios a outros usuários dos objetos de sua propriedade. O DBA tem autoridade máxima, podendo não só autorizar novos usuários como também reestruturar o banco.

A segurança do banco de dados pode considerar duas abordagens: controle discricionário e mandatário. O **controle discricionário** indica que o privilégio pode ser dado a usuários, acessos e objetos diferentes. Por exemplo: o Usuário 1 pode ler apenas a Tabela X e ler e gravar a Tabela Y. Já o Usuário 2 pode somente ler as Tabelas X e Y. Como você pode notar, esse tipo de controle é bem flexível.

De acordo com Lima (2012), o controle discricionário se baseia na ideia de que é o proprietário da informação que deve determinar quem tem acesso a ela. Suponha, por exemplo, que uma criança possui um diário. A criança

controla o acesso ao diário, pois ela pode permitir que alguém o leia (conceder acesso de leitura), ou não permitir que alguém o leia (negar acesso de leitura). A criança permite que a sua mãe leia o diário, mas ninguém mais. Esse é um controle de acesso discricionário, pois o acesso ao diário baseia-se na vontade do dono (SANDHU; SAMARATI, 1994).

Já o **controle mandatário** é a permissão atribuída pelo nível estabelecido ao objeto e ao usuário. Se os níveis são iguais, o acesso é liberado. Por exemplo: a Tabela X é do nível I, e a Tabela Y é do nível 2. Lima (2012) afirma que esse tipo de controle é baseado em uma autorização prévia, sendo a identidade do usuário irrelevante. Ou seja: quando um mecanismo do sistema controla o acesso a um objeto e um usuário não pode alterar esse acesso, o controle é chamado de controle de acesso mandatário (às vezes também chamado de controle de acesso baseado em regras). Um exemplo de controle mandatário é a lei que permite a um tribunal acessar os registros dos condutores de veículos sem autorização prévia dos proprietários. Esse é um controle obrigatório, porque o proprietário do registro não controla o acesso à informação pelo tribunal (SANDHU; SAMARATI, 1994).

Dessa forma, é preciso atribuir permissões para que se possa conceder ao usuário o privilégio de ter acesso aos objetos do banco de dados, sejam tabelas, visões, procedimentos, funções, *triggers* (gatilhos) e outros. Essa permissão pode limitar, liberar ou remover acesso. O foco deve ser, sempre, definir regras que impeçam operações impróprias que violem a confidencialidade, a integridade e a disponibilidade das informações no banco de dados.

Permissões — aplicação e consequências

As permissões são concedidas aos usuários ou grupos de usuários e se referem a determinado recurso. A ideia é permitir acesso e uso desse recurso. Entende-se por recurso o conjunto de procedimentos e conteúdos armazenados no banco de dados. Dessa forma, o administrador do banco de dados tem uma função importante na definição do perfil do usuário, já que concede as permissões cabíveis às necessidades.

As permissões são aplicadas de forma interligada e dependente entre usuários, grupos e recursos, como você pode ver na Figura 2.

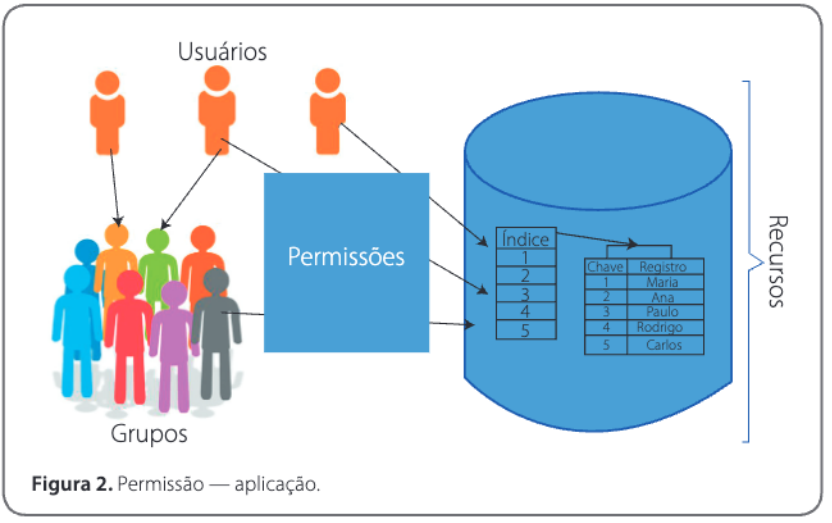

Figura 2. Permissão — aplicação.

Portanto, para implantar um controle de acesso eficiente aos recursos, você deve se preocupar com os usuários, grupos e acessos.

Usuários

A criação do usuário é o primeiro privilégio concedido para acesso aos recursos do banco de dados. Isso é configurado por meio de uma identificação e uma senha. Os usuários são os responsáveis principais por manter a disponibilidade, a confidencialidade e a integridade dos dados armazenados. Afinal, basta que eles violem uma das regras da política de segurança, que é o sigilo de sua senha de identificação, descumprindo a orientação de que a senha é pessoal e intransferível, para criar brechas de invasão e mau uso do ambiente.

Assim, o gerenciamento de usuários é determinante no processo de segurança da informação na empresa. É função da gerência de usuários controlar os privilégios atribuídos aos usuários para que a permissividade fique restrita às reais necessidades. Nesse sentido, você deve considerar que as necessidades mudam e é necessário estar atento a elas.

É indicado, então, que frequentemente o DBA:

- liste os usuários e os seus privilégios e faça atualizações;
- implemente a política de troca de senhas;
- remova privilégios não mais necessários;
- remova usuários.

Grupos

Os grupos devem ser formados a partir de um conjunto de usuários que usem os mesmos privilégios. Os SGBDs disponibilizam recursos para criar, modificar e remover grupos. A todo grupo é atribuída uma identificação, utilizada para realizar as operações.

O gerenciamento de grupos facilita a atribuição e a remoção de permissões. Afinal, cada permissão é concedida uma única vez ao grupo, tornando a autoridade efetiva a todos os usuários que façam parte daquele grupo. Por exemplo, considere uma empresa com fins comerciais que possui 250 vendedores com as mesmas funções. Tais vendedores executam as mesmas funcionalidades no negócio e, consequentemente, utilizam os mesmos recursos do banco de dados. Se houver um grupo VENDEDOR, quando for necessária a concessão ou a remoção de algum recurso, ela será atribuída ao grupo, e não a cada vendedor. Isso otimiza a execução de 250 vezes o mesmo comando.

Um grupo de usuários ou usuários individuais podem ter privilégios diferentes de acordo com o objeto do banco de dados. Ou seja, um usuário pode, por exemplo, consultar a tabela DISCIPLINAS, mas não pode alterá-la, enquanto pode alterar a tabela ALUNOS.

Caso um usuário específico de um grupo, por algum motivo, não faça mais parte do grupo, os SGBDs suportam a remoção de privilégios para usuários individualmente. Por outro lado, um usuário também pode ser incluído no grupo, mesmo que este já esteja criado.

Acessos

O acesso é definido a partir da concessão de privilégios a grupos e/ou usuários. Pode ser interessante a empresa definir uma política de acesso, estabelecendo regras e ajudando, assim, a padronizar e orientar as atitudes na concessão e na remoção de privilégios, mantendo sempre alinhadas as questões de segurança, confidencialidade e integridade do banco de dados.

A concessão de privilégios a usuários ou grupos é definida a partir do tipo de permissão e do recurso a ser autorizado. Cada SGBD utiliza um conjunto de recursos e permissões associadas. Segundo o *Manual de Referência do MySQL 8.0* (MYSQL..., c2019), os tipos de privilégio no SGBD MySQL são aplicados em três níveis diferentes de operação. Veja a seguir.

- Nível administrativo: permite a gerência do servidor MySQL.
- Nível do banco de dados: envolve o banco de dados (*schema*) e todo objeto que pertença a ele.
- Nível dos objetos do banco de dados: envolve tabelas, índices, exibições e rotinas armazenadas.

Os privilégios podem ser estáticos ou dinâmicos. Os privilégios estáticos são incorporados ao servidor. Já os privilégios dinâmicos são definidos em tempo de execução. O diferencial entre os privilégios estáticos e os dinâmicos está na disponibilidade da concessão à conta e nas funções do usuário. No caso de privilégios dinâmicos, eles devem estar registrados para poderem ser concedidos.

Privilégios estáticos

Veja os recursos associados:

- acesso ao arquivo no *host* do servidor;
- administração do servidor;
- bancos de dados;
- tabelas;
- índices;
- rotinas armazenadas;
- colunas;
- *views*.

Agora veja as permissões estáticas associadas:

ALL [PRIVILEGES]	DROP	RELOAD
ALTER	DROP ROLE	REPLICATION CLIENT
ALTER ROUTINE	EVENT	REPLICATION SLAVE
CREATE	EXECUTE	SELECT
CREATE ROLE	FILE	SHOW DATABASES
CREATE ROUTINE	GRANT OPTION	SHOW VIEW
CREATE TABLESPACE	INDEX	SHUTDOWN
CREATE TEMPORARY TABLES	INSERT	SUPER
	LOCK TABLES	TRIGGER
CREATE USER	PROCESS	UPDATE
CREATE VIEW	PROXY	USAGE
DELETE	REFERENCES	

Link

Você pode verificar a descrição de cada permissão estática no endereço oficial do SGBD MySQL:

https://qrgo.page.link/FwFy

Privilégios dinâmicos

Veja os recursos associados:

- administração de *backup*;
- administração de *firewall*;
- administração de *log* de auditoria;
- administração de replicação;
- administração de senha dupla;
- administração do grupo de recursos;
- administração do servidor.

Agora veja as permissões associadas:

BACKUP_ADMIN
BINLOG_ADMIN
BINLOG_ENCRYPTION_ADMIN
FIREWALL_ADMIN
FIREWALL_USER
AUDIT_ADMIN
GROUP_REPLICATION_ADMIN
REPLICATION_SLAVE_ADMIN
APPLICATION_PASSWORD_ADMIN
RESOURCE_GROUP_ADMIN
RESOURCE_GROUP_USER
CONNECTION_ADMIN
ENCRYPTION_KEY_ADMIN
PERSIST_RO_VARIABLES_ADMIN
ROLE_ADMIN
SESSION_VARIABLES_ADMIN
SET_USER_ID
SYSTEM_VARIABLES_ADMIN
TABLE_ENCRYPTION_ADMIN
VERSION_TOKEN_ADMIN
XA_RECOVER_ADMIN

Fique atento

A criação do esquema no banco de dados atribui, automaticamente, o privilégio PUBLIC tanto para usuários individuais quanto para grupos. Isso significa que os usuários e grupos terão autoridade para criar no esquema os recursos do banco de dados que desejarem. É indicado remover todos os privilégios do esquema para um "usuário" PUBLIC.

O privilégio atribuído a um grupo sobrepõe os privilégios de um usuário. Portanto, todos os privilégios de um grupo serão válidos aos usuários, mesmo que eles não tenham tais privilégios individualmente.

Todo SGBD oferece um conjunto de privilégios, associado aos recursos que disponibiliza. Para o melhor uso dos privilégios, é importante:

- criar uma política de acesso;
- definir o perfil de cada usuário;
- utilizar grupos;
- buscar a documentação oficial do SGBD para entender os efeitos das permissões e a sua sintaxe.

Permissões — uso

A concessão e a remoção de privilégios são efetivadas por meio de comandos definidos na linguagem de controle de dados (*Data Control Language* [DCL]), disponibilizada pela linguagem estruturada de consultas (*Structured Query Language* [SQL]). São utilizados os comandos GRANT para concessão e REVOKE para remoção do privilégio.

Todo SGBD utiliza uma sintaxe própria para seus comandos. Tal sintaxe deve ser seguida conforme o site oficial do SGBD. Para usar os comandos, você deve considerar o seguinte:

- as cláusulas entre colchetes [ ] são opcionais;
- as opções de recursos aparecem relacionadas entre chaves { } e separadas pelo símbolo de |, que representa "ou".
- o comando está na língua nativa do SGBD, inglês, e deve ser utilizado com a grafia em que é apresentado.

Comando GRANT

O comando GRANT é utilizado para a concessão de privilégios relacionados aos recursos do banco de dados. Aqui, o *Manual de Referência do MySQL 8.0* é utilizado para a apresentação da sintaxe (PRIVILEGES provided..., c2019). Veja o formato do comando GRANT:

```
GRANT tipoPrivilégio [(column_list)]    [,tipoPrivilégio [(column_list)]]
   ON [tipoObjeto] nívelPrivilégio
   TO nomeUsuario | nomePapel [,nomeUsuario | nomePapel]
   [WITH GRANT OPTION];
GRANT role [, role]   TO user_or_role [, user_or_role]   [WITH ADMIN OPTION];
```

Onde:

- tipoObjeto: { TABLE | FUNCTION | PROCEDURE }
- nívelPrivilégio: { * | *.* | nomeBanco.* | nomeBanco.nomeTabela | nomeTabela | nomeBanco.nomeProcedimento }

O parâmetro GRANT OPTION concede ao usuário o direito de atribuir privilégios a outros usuários. Quando você utilizar o GRANT ALL, todos os privilégios serão concedidos. Contudo, um usuário só pode conceder os privilégios que possui.

O comando GRANT considera os níveis diferentes de operação (administrativo, banco de dados e objetos de banco de dados) para a definição dos tipos de privilégios. Você vai conhecer melhor os privilégios a seguir.

Privilégios administrativos

Os privilégios dinâmicos são todos globais e concedidos sempre de forma global. Os privilégios estáticos CREATE TABLESPACE, CREATE USER, FILE, PROCESS, RELOAD, REPLICATION CLIENT, REPLICATION SLAVE, SHOW DATABASES, SHUTDOWN e SUPER só podem ser concedidos de forma global.

O privilégio global é atribuído a partir do uso do *.*. O asterisco do lado esquerdo do ponto (.) significa o nome do banco de dados, e o asterisco do lado direito refere-se a todas as tabelas. Pode-se especificar somente uma tabela indicando o nome dela. Por exemplo, suponha atribuir privilégio SUPER, que

permite algumas operações de administração de banco de dados, ao usuário USER001:

```
GRANT SUPER ON *.* TO 'USER001'@'localhost';
```

Outros tipos de privilégios, como SELECT, INSERT, DELETE, UPDATE e ALL (engloba todos os outros), podem ser concedidos de forma global ou específica a um usuário ou grupo. Por exemplo, suponha ter de conceder ao usuário USER002 todos os tipos de privilégios relativos a todas as tabelas do banco de dados VENDAS:

```
GRANT ALL ON VENDAS.* TO 'USER002'@'localhost';
```

Privilégios de banco de dados

A aplicação de privilégios em bancos de dados implica a aplicação a todos os objetos do banco de dados em referência. Para isso, você deve utilizar a cláusula:

```
ON nomeBanco.*;
```

O asterisco do lado direito refere-se a todas as tabelas. Por exemplo, suponha atribuir ao grupo VENDEDOR o privilégio de consulta a todas as tabelas do banco de dados VENDAS:

```
GRANT SELECT ON VENDAS.* TO 'VENDEDOR'@'localhost';
```

A omissão do nome do banco de dados indica que a permissão será criada no banco de dados padrão, em que o comando é submetido.

Além dos privilégios SELECT, INSERT, DELETE, UPDATE e ALL, os seguintes privilégios podem ser utilizados no nível do banco de dados: CREATE, DROP, EVENT, GRANT OPTION, LOCK TABLES e REFERENCES.

Privilégios de tabela

Os privilégios utilizados em tabela se aplicam a todos os atributos e não são aplicados a tabelas temporárias. Para isso, você deve utilizar a cláusula:

```
ON nomeBanco.*;
```

O asterisco do lado direito refere-se a todas as tabelas. Você pode especificar somente uma tabela indicando o nome dela. Por exemplo, suponha atribuir o privilégio de inclusão na tabela NOTAFISCAL do banco de dados VENDAS ao usuário USER003:

```
GRANT INSERT ON VENDAS.NOTAFISCAL TO 'USER003@'localhost';
```

A omissão do nome do banco de dados indica que a permissão será criada no banco de dados padrão, em que o comando é submetido.

Além dos privilégios SELECT, INSERT, DELETE, UPDATE e ALL, os seguintes privilégios podem ser utilizados no nível da tabela: ALTER, CREATE VIEW, CREATE, DELETE, DROP, GRANT OPTION, INDEX, INSERT, REFERENCES, SELECT, SHOW VIEW, TRIGGER e UPDATE.

Privilégios de coluna

A concessão de privilégios a atributos/colunas de uma tabela ocorre de forma única ao atributo/coluna especificada. Para isso, você deve colocar entre parênteses o nome do atributo/coluna imediatamente após o tipo de privilégio:

```
GRANT tipoprivilegio (nomeAtributo), insert (nomeAtributo, nomeAtributo) on nomeBanco.nomeTabela TO 'nomeUsuario'@'nomeHost';
```

Dessa forma, é possível atribuir, em um mesmo comando, privilégios a um atributo e/ou a vários. Por exemplo, suponha que, em uma empresa, no sistema de pagamento, é preciso definir privilégio para o atributo SALARIO da tabela PAGAMENTO aos usuários específicos que cuidam da folha de pagamento. A esses usuários, que são os usuários USER001 e USER002, devem ser concedidos todos os privilégios de operação. O comando correspondente ficaria assim:

```
GRANT ALL (SALARIO) ON PAGAMENTO TO
'USER001@'localhost', 'USER002@'localhost';
```

Os privilégios permitidos a atributos são: INSERT, REFERENCES, SELECT e UPDATE.

Privilégios de procedimentos armazenados

A concessão de privilégios a procedimentos armazenados utiliza os tipos ALTER ROUTINE, EXECUTE e GRANT OPTION. Tais privilégios podem ser concedidos a nível global e banco de dados. Para isso, você deve utilizar o comando:

GRANT tipoPrivilégio ON PROCEDURE nomeBanco.nomeProcedimento TO 'nomeUsuario'@'nomeHost';

Por exemplo, suponha que, no banco de dados de uma empresa, o dígito verificador da matrícula do funcionário é calculado por meio de um procedimento sempre que ocorre a inclusão de um novo funcionário. Depois de criar o procedimento, é necessário conceder o privilégio de execução (EXECUTE), denominado "calculoDV", para o usuário USER003. O comando correspondente ficaria assim:

GRANT EXECUTE on PROCEDURE calculoDV TO 'USER003@'localhost';

Concessão de papéis

A criação de papel está relacionada à criação de grupos. No MySQL, cria-se um papel com o comando CREATE ROLE. Depois, atribui-se a esse papel a lista de privilégios necessários e, a partir daí, se concedem aos usuários os papéis como um privilégio. A sintaxe do comando GRANT para a concessão de papéis a usuários é representada da seguinte forma:

GRANT 'nomePapel1 ', 'nomePapel2'' TO 'nomeUsuario1'@'nomeHost', 'nomeUsuario2'@'nomeHost';

Por exemplo, suponha que o papel denominado VENDEDOR representa o conjunto de vendedores da empresa, com os nomes vend001, vend002 e vend003. O comando correspondente ficaria assim:

GRANT 'VENDEDOR' TO 'vend001@'localhost', 'vend002@'localhost', 'vend003@'localhost';

Incluindo a cláusula WITH ADMIN OPTION, cada usuário do mesmo papel (grupo) terá autoridade para conceder funções a outros usuários ou revogar usuários de funções.

Comando REVOKE

O comando REVOKE é utilizado para revogar dos usuários privilégios atribuídos aos recursos do banco de dados. Considere o *Manual de Referência do MySQL 8.0* para a apresentação de toda a sintaxe (PRIVILEGES provided..., c2019). Veja o formato do comando REVOKE:

```
REVOKE
    tipoPrioridade [(listaColunas)] [,tipoPrioridade [(listaColunas)]] ...
    ON [tipoObjeto] nivelPrivilegio
    FROM nomeUsuario | nomePapel [,nomeUsuario | nomePapel] ...

REVOKE ALL [PRIVILEGES], GRANT OPTION
    FROM nomeUsuario | nomePapel [,nomeUsuario | nomePapel] ...

REVOKE nomePapel [,nomePapel] ...
    FROM nomeUsuario | nomePapel [,nomeUsuario | nomePapel] ...
```

As cláusulas apresentadas para o comando REVOKE seguem as mesmas definições descritas para o comando GRANT. A seguir, veja alguns aspectos importantes em relação ao REVOKE.

- O usuário que estiver revogando privilégios deve ter o privilégio GRANT OPTION, além de ter o privilégio que estiver revogando.
- O comando REVOKE ALL PRIVILEGES não revoga as funções. Para isso, você deve usar a cláusula FROM. Depois de revogada a função, as contas de usuários são afetadas e ajustadas na próxima execução.
- Para usar a sintaxe REVOKE, você deve ter o privilégio global CREATE USER ou o privilégio UPDATE para o banco de dados do sistema MySQL.
- A aplicação do comando REVOKE ALL ON *.* significa que todos os privilégios serão revogados, estáticos e dinâmicos.
- REVOKE remove privilégios, não apaga o usuário. Para isso, você deve utilizar o DROP USER.

Como exemplo, suponha que, em uma empresa, o usuário USER001 mudou de função e não pertence mais ao papel/grupo VENDEDOR. Será preciso retirá-lo do papel/grupo VENDEDOR. O comando SQL para retirar o privilégio do papel/grupo VENDEDOR atribuído ao usuário USER001 é:

```
REVOKE VENDEDOR FROM 'USER001';
```

Administração de privilégios

Para um DBA, criar usuários, conceder e revogar privilégios não são as únicas funções que garantem a segurança da informação. É preciso controlar os privilégios concedidos, acompanhar se os usuários que os têm de fato os utilizam e, acima de tudo, verificar se possuem a necessidade de tê-los. É fundamental manter os usuários com privilégios que, de fato, são utilizados.

Para controlar os privilégios, os registros são feitos em tabelas, que podem ser utilizadas com os comandos comuns de consulta, como o SELECT. No MySQL, os privilégios são armazenados em tabelas correspondentes a cada privilégio, como você pode ver no Quadro 1. Por exemplo, suponha que o DBA irá listar todos os privilégios dos usuários. O comando seria:

```
Select * from mysql.user;
```

Quadro 1. Privilégios e tabelas

Privilégios	Nome da tabela
Globais	mysql.user
Banco de dados	mysql.db
Tabela	mysql.tables_priv
Coluna	mysql.columns_priv

Referências

DATE, C. J. *1941 — Introdução a sistemas de bancos de dados*. Rio de Janeiro: Elsevier, 2003.

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados*. 6. ed. São Paulo: Pearson, 2011.

LIMA, A. C. *SACM — Um modelo de controle de acesso baseado em estado e sensível ao contexto*. 2012. 82 f. Dissertação (Mestrado em Ciência da Computação) — Universidade Estadual do Ceará, Fortaleza, 2012. Disponível em: https://ins3rt.s3.amazonaws.com/media/papers/sacm.pdf. Acesso em: 18 abr. 2019.

MYSQL data dictionary. *In*: ORACLE. MySQL 8.0 reference manual. c2019. Disponível em: https://dev.mysql.com/doc/refman/8.0/en/data-dictionary.html. Acesso em: 17 abr. 2019.

PRIVILEGES provided by MySQL. *In*: ORACLE. *MySQL 8.0 reference manual*. c2019. Disponível em: https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_all. Acesso em: 18 abr. 2019.

SANDHU, R.; SAMARATI, P. Access control: principles and practice. *IEEE Communications*, v. 32, n. 9, p. 40–48, 1994.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. *Sistema de banco de dados*. 6. ed. Rio de Janeiro: Elsevier, 2012.

Leituras recomendadas

ELMASRI, R.; NAVATHE, S. B. *Fundamentals of database systems*. 6. ed. Boston: Addison-Wesley, c2010.

LAUDON, K. C.; LAUDON, J. P. *Sistemas de informação*. Rio de Janeiro: LTC, 1999.

OZSU, M. T.; VALDURIEZ, P. *Princípios de sistemas de banco de dados distribuídos*. Rio de Janeiro: Campus, 2001.

PUGA, S.; FRANÇA, E.; GOYA, M. *Banco de dados*: implementação em SQL, PL/SQL e Oracle 11G. São Paulo: Pearson, 2013.

WHEN privilege changes take effect. *In*: ORACLE. *MySQL 8.0 reference manual*. c2019. Disponível em: https://dev.mysql.com/doc/refman/8.0/en/privilege-changes.html. Acesso em: 18 abr. 2019.

Conteúdo:

SAGAH
SOLUÇÕES EDUCACIONAIS INTEGRADAS