ADMINISTRAÇÃO DE BANCO DE DADOS

Claudia Abreu Paes

SAGAH SOLUÇÕES EDUCACIONAIS INTEGRADAS

Gerenciamento de usuários do banco de dados

Objetivos de aprendizagem

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

- Descrever as etapas de gerenciamento de usuários do banco de dados.
- Determinar como criar, alterar e remover usuários através de *scripts*.
- Reconhecer os conceitos de gerenciamento de usuários.

Introdução

O usuário faz parte do grupo de pessoas que compõem a visão sociotécnica dos sistemas empresariais, formados por *hardware*, dados, *software* e pessoas. Segundo Sommerville (2007), os usuários são pessoas influenciadas pela forma como a organização é gerenciada. Eles também são influenciados pelas suas interações com outras pessoas dentro e fora da organização.

Decorre daí a importância do usuário no banco de dados. A partir dos dados armazenados e por meio de seus processos e atitudes, o usuário atribui o verdadeiro valor ao alcance dos objetivos do negócio. A confidencialidade, ou seja, a proteção aos dados, também é determinada pelo modo como os usuários lidam com as informações. Nesse contexto, é importante conhecer as visões que os usuários possuem em relação ao desenvolvimento de suas atividades. Em síntese, a visão é a forma como o usuário enxerga e aplica os dados. Muitos usuários podem compartilhar uma mesma visão, o que justifica o uso de grupos de usuários.

Neste capítulo, você vai conhecer os conceitos de usuário e grupo de usuários. Você também vai ver a importância de conhecer a visão adequada de cada usuário. Além disso, vai estudar as etapas de gerenciamento e a utilização de *scripts* para a execução de comandos SQL e para a manutenção de usuários.

Usuários, grupos e visões

Segundo Graves (2003), o sistema de um banco de dados é formado pelo banco de dados e pelo ambiente disponível para o seu uso, que inclui usuários, *software* e *hardware*. O *hardware* é necessário para hospedar e prover recursos de interação entre os serviços disponibilizados pelos *softwares*. Por sua vez, o *software* é composto pelo sistema operacional, pelo Sistema Gerenciador de Banco de Dados (SGBD) e por programas de diversas finalidades. Os usuários são as pessoas que, de alguma forma, utilizam os dados, seja para desenvolver os *softwares* — caso dos analistas e programadores —, para gerenciar os dados — caso dos administradores de banco de dados (DBA) —, ou para utilizar os dados — caso dos clientes ou usuários finais (Figura 1).

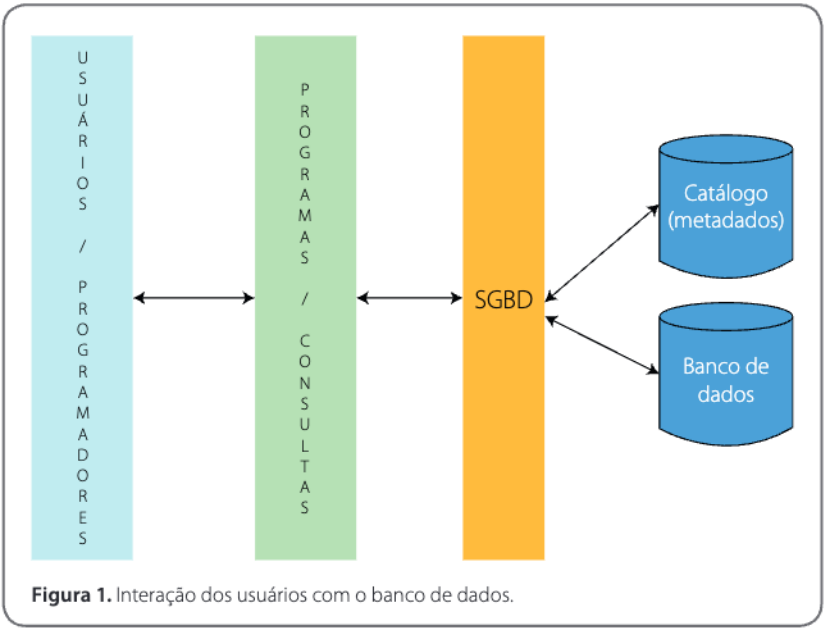

Figura 1. Interação dos usuários com o banco de dados.

Em um ambiente operacional de tecnologia, há servidores que atendem aos sistemas corporativos e bancos de dados que trabalham coordenados pelo sistema operacional do servidor onde estão instalados, mas com autonomia em suas definições. Há também usuários do sistema operacional utilizando os sistemas corporativos, e devem existir os usuários do banco de dados.

Os usuários do banco de dados possuem definição própria em relação às definições no servidor hospedeiro.

Se um usuário é criado no âmbito do banco de dados, o SGBD tem total controle das ações e visões desse usuário. A cada usuário é associado um nome de identificação, que serve de meio para a definição de sua visão no banco de dados. Segundo Macedo (2011), os nomes dos usuários de banco de dados são globais para todo o agrupamento de bancos de dados (e não próprios de cada banco de dados).

Saiba mais

Em algumas instalações, os usuários do banco de dados são criados a partir dos usuários nos servidores hospedeiros, mas é fato que o SGBD possui seus próprios usuários cadastrados, independentemente da origem da criação.

Segundo o dicionário Aurélio (2018), visão é o ato ou efeito de ver. Para o banco de dados, a visão se refere às informações ou ao conjunto de informações que o usuário busca. Assim, entende-se visão como um conjunto virtual de informações formado a partir da junção de informações do banco de dados.

Silberschatz, Korth e Sudarshan (2012) consideram que proporcionar ao usuário uma visão abstrata é um dos maiores benefícios de um banco de dados. Na visão abstrata, o sistema oculta a forma de armazenamento e de manutenção dos dados. Nesse sentido, os autores (2012) propõem três níveis de abstração de usuário, cada um com sua representação conceitual dos dados. Buscando eficiência na abstração dos dados, o SGBD fornece as informações apropriadas para cada nível de abstração de usuário. Assim, cada nível terá uma abstração que estará ocultando a complexidade e simplificando a interação.

Os três níveis de abstração de usuário são: físico, lógico e visão. Veja a seguir.

- O nível de abstração físico ou interno é considerado o nível mais baixo e apresenta os dados como são armazenados no banco de dados (estruturas).
- O nível de abstração lógico ou conceitual é o nível intermediário, que apresenta os dados e os seus relacionamentos.
- O nível de visão ou nível externo é o mais alto, utilizado pelos usuários finais. Ele apresenta somente o que é necessário a esses usuários.

Na Figura 2, você pode ver os níveis de abstração.

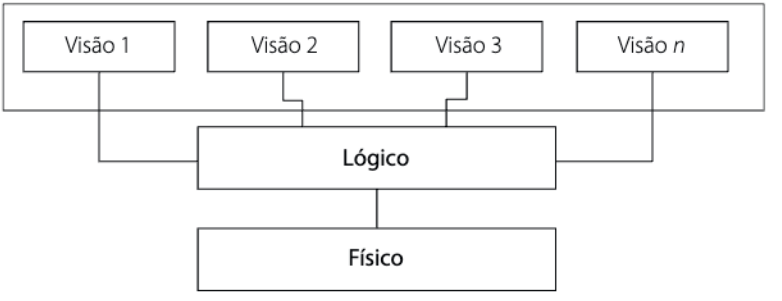

Figura 2. Níveis de abstração.
Fonte: Adaptada de Silberschatz, Korth e Sudarshan (2012).

Ainda de acordo com Silberschatz, Korth e Sudarshan (2012), um banco de dados deve prover informações a todo tipo de usuário, de acordo com suas necessidades específicas. Essas necessidades acabam gerando visões distintas. Dessa forma, há múltiplas visões dos dados. Por exemplo, em uma organização, os funcionários do departamento pessoal precisam ver as informações necessárias para a folha de pagamento e para o recolhimento dos impostos e das obrigações legais. Entre as informações da folha de pagamento, você pode considerar: funcionário, situação (ativo, férias, desligado, licença-maternidade, afastamento, etc.), salário, faltas no mês, folha de ponto, departamento, plano de saúde, entre outras. Assim, o administrador do banco de dados cria uma visão específica com essas informações. Já os funcionários do setor de compras precisam ver as informações relativas a contratos (vendas), estoque, fornecedores, mercado, entre outras.

Cada visão pode atender a vários usuários e, para o banco de dados, os vários usuários formam um grupo. Assim, em um banco de dados, um grupo representa o conjunto de usuários que têm a mesma visão dos dados. Manter grupos de usuários simplifica a administração de usuários em um banco de dados na medida em que as visões e ações podem ser atribuídas a partir do grupo e não individualmente a cada usuário.

Etapas de gerenciamento de usuários

Até agora, você viu que no banco de dados o acesso é feito por usuários e que os dados são apresentados a ele de acordo com as suas necessidades. Além disso, todo usuário deve ser identificado. Assim, é muito importante para a segurança, a integridade e a confidencialidade do banco de dados conhecer os usuários que o utilizam e classificá-los de acordo com as suas necessidades e apropriações de uso.

Portanto, o gerenciamento de usuários controla os usuários do banco de dados, disponibilizando as visões dos objetos que proverão os dados necessários. Os usuários são classificamos em função do tipo de utilização que se dá no banco de dados: administradores, técnicos, especialistas e finais (Figura 3).

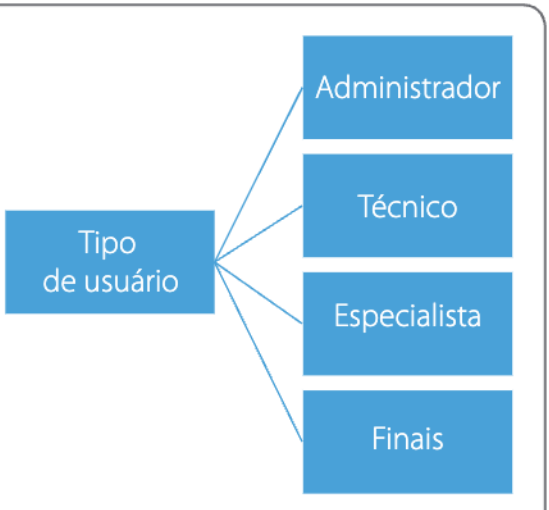

Figura 3. Tipos de usuários.

Usuários administradores

Há dois usuários administradores: administrador de dados e administrador de banco de dados (*DataBase Administrator* [DBA]). Eles atuam no nível físico do banco de dados e têm as funções listadas a seguir.

- Administrador de dados: define os dados, seus relacionamentos e o modo como as aplicações podem compartilhar os dados; atualiza o esquema do banco de dados; mantém a consistência das informações.
- Administrador de banco de dados: gerencia a estrutura de armazenamento, a estratégia e a autorização de acesso aos dados; define controles de integridade e estratégias de *backup* e monitoramento do desempenho. Gerencia e planeja as atividades de manutenção dos bancos de dados, define políticas de segurança e planos de contingência.

Usuários técnicos

Os usuários técnicos atuam no nível lógico, pois utilizam os dados para o desenvolvimento de sistemas diretamente no SGBD. Nessa atuação, é necessária a utilização direta do banco de dados para testes e definições. Analistas de sistemas e desenvolvedores são usuários técnicos, como você pode ver a seguir.

- Analista de sistemas: define os requisitos dos usuários finais e desenvolve especificações para transações que atendam a esses requisitos.
- Desenvolvedor de aplicações: implementa as especificações dos programas, testando, depurando, documentando e fazendo a manutenção.
- Desenvolvedor de banco de dados: trabalha com a construção de *triggers* e linguagens de programação do banco de dados, como: PL/SQL (Oracle), Transact SQL (SQL Server) e PL/pgSQL (PostgreSQL). Essas linguagens costumam ter um bom desempenho, pois ficam armazenadas no banco de dados. O MySQL não possui uma linguagem de programação, mas disponibiliza todos os recursos necessários para o desenvolvimento de *procedures* e *triggers*.

Usuários especialistas

Os usuários especialistas atuam no nível de visão com a função do usuário final, mas também atuam no nível lógico, pois manipulam dados sem o uso de aplicativos e utilizam linguagens de consulta diretamente no banco de dados para obter consultas com o intuito de extração de conhecimento. Para isso, esses usuário precisam conhecer a estrutura do banco de dados. Como exemplo de usuário especialista, você pode considerar o analista de negócio que utiliza o *data warehouse* e ferramentas de *Business Intelligence*.

Usuários finais

Os usuários finais atuam no nível de visão. Eles manipulam os dados do banco de dados por meio de interfaces construídas pelos aplicativos das organizações em atendimento aos processos de negócio. Contudo, desconhecem a estrutura do SGBD.

Saiba mais

O ***data warehouse***, considerado a base para a *business intelligence*, é um armazém de dados utilizado para armazenar dados históricos dos sistemas. Normalmente, os dados são desnormalizados. Eles são utilizados somente para consultas, levando o grupo estratégico das empresas a tomar decisões baseadas em fatos, não em intuições e especulações.

Por sua vez, a *business intelligence* é um processamento de dados que gera informações que embasam a tomada de decisão. Essas informações podem ser apresentadas em formato de relatórios ou *dashboard*. Para saber mais sobre a *business intelligence*, leia o trabalho de Moura (2000), disponível no *link* a seguir.

https://qrgo.page.link/qzvp

Política de segurança da informação

A informação é o elemento básico para que a evolução aconteça e o desenvolvimento humano se realize de forma completa (COURY, 2001 *apud* CARDOSO; OLIVEIRA, 2013). Para Campos (2007, p. 21), "a informação é elemento essencial para todos os processos de negócio da organização, sendo, portanto, um bem ou ativo de grande valor". Dessa forma, é possível dizer que a informação é o coração que pulsa nas organizações.

Portanto, diante de uma variedade de ameaças e vulnerabilidades, o DBA deve se preocupar com a confidencialidade, a integridade e a disponibilidade da informação. A **confidencialidade** é a garantia de que a informação pode ser acessada somente por pessoas autorizadas (ABNT, 2005). Se o acesso acontecer por pessoa não autorizada, ocorrerá a quebra de sigilo, o que pode acarretar danos para a organização. Como exemplo, considere o uso de senhas que não sejam de propriedade do usuário.

A **integridade** é a garantia da exatidão e da completeza da informação e dos métodos de processamento (ABNT, 2005). Assim, "garantir a integridade é permitir que a informação não seja modificada, alterada ou destruída sem autorização, que ela seja legítima e permaneça consistente" (DANTAS, 2011, documento *on-line*). Qualquer violação da informação ocasionará quebra de integridade. Por sua vez, a **disponibilidade** é a garantia de que os usuários autorizados obtenham acesso à informação e aos ativos correspondentes sempre que necessário (ABNT, 2005). A indisponibilidade por quaisquer motivos, voluntários ou não, acarreta quebra de disponibilidade.

A gerência de usuários do banco de dados tem importância fundamental na definição de uma boa política de segurança. A vulnerabilidade se dá também a partir de equipamentos tecnológicos, mas principalmente a partir dos usuários. Isso potencializa a importância da inclusão de regras relacionadas ao comportamento e à postura dos usuários na implantação de uma política de segurança.

A Política de Segurança da Informação é um documento elaborado na empresa que contém normas, métodos e procedimentos que garantem a viabilidade e o uso dos ativos somente por pessoas autorizadas e que realmente necessitam deles para realizar as suas funções na organização (FONTES, 2008). Entre as seções da Política de Segurança da Informação, está a de controle de acesso, diretamente ligada aos usuários, pois é por meio da concessão de acesso que o usuário passa a ter visibilidade dos objetos e do ambiente do banco de dados.

O acesso é fornecido ao usuário por meio de duas informações: nome de usuário, para identificar os usuários, e uma senha, para garantir o acesso restrito ao ambiente. É importante definir uma política de senhas a fim de proteger os dados do banco de dados. A política de senhas pode incluir regras de criação e atualização, como estas:

- adotar prazo de validade para as senhas, obrigando o usuário a alterar a sua senha em um período determinado (30 dias, por exemplo);
- estabelecer um critério de formação, por exemplo: tamanho, utilização de caracteres especiais, maiúscula e minúscula, mistura de números e letras, etc.;
- determinar, por uma lista, um conjunto de senhas proibidas;
- não permitir a repetição de senhas já utilizadas;
- buscar informações fáceis de memorizar, mas não utilizar datas comemorativas pessoais.

A política pode ainda sugerir ao usuário atitudes como:

- não permitir que o observem enquanto digita a senha;
- não fornecer a senha a terceiros (a senha é intransferível e confidencial);
- ter cuidado para utilizar a senha em equipamentos confiáveis.

Dessa forma, o DBA deve buscar conhecer os usuários e as suas necessidades de uso efetivo dos recursos do banco de dados para não gerar visibilidade desnecessária. Outro ponto importante refere-se à conscientização do usuário quanto à importância do cumprimento dos pontos estabelecidos na política de segurança da informação. Há diversos movimentos educacionais focados em desenvolver esse tipo de conscientização.

Link

Para entender melhor a conscientização de usuários, acesse o *link* a seguir.

https://qrgo.page.link/Uv4S

Uso de *scripts*

Um *script* é um conjunto de instruções organizadas para a execução de um procedimento, denominado no MySQL de processamento *batch*. Em síntese, o *script* é um arquivo de texto com a extensão .SQL gerado a partir de *softwares* de edição. No caso de bancos de dados, utiliza-se o *script* para escrever as instruções a realizar, agilizando a execução e gerando uma documentação. O uso de *scripts* favorece também a reutilização. Esse recurso pode ser utilizado, por exemplo, em um conjunto de procedimentos que devem ser rodados várias vezes, como rotinas bimestrais de verificação de existência de usuários inativos.

Considere que a criação de um usuário implica a concessão de privilégios a várias tabelas. Toda vez que se cria um usuário, as permissões devem ser atribuídas. Dessa forma, para não ter de montar o conjunto de instruções sempre que cria um usuário, você pode criar um *script*:

CRIAUSUARIO.SQL

```
Create user TESTE identified by '123456';
Grant all  on table produto to teste;
Grant all  on table cliente to teste;
Grant all  on table venda to teste;
Grant all  on table vendedor to teste;
Grant all  on table cartao to teste;
Grant all  on table banco to teste;
Grant all  on table fornecedor to teste;
Grant all  on table itens to teste;
Grant all  on table pedido to teste;
```

Todo comando oferecido pelo SGBD MySQL pode ser usado em um *script*. Toda instrução deve estar em uma linha e deve haver ponto e vírgula (;) ao final da linha para identificar o fim do comando. O *Manual de Referência do MySQL 8.0* elenca os motivos a seguir para se utilizar um *script* (MYSQL, 2019).

- utilizar consulta frequentemente e evitar redigitação dos comandos;
- reaproveitar instruções já existentes, adaptando-as a novas necessidades;
- otimizar tempo na correção de arquivos extensos. Imagine que você tem 40 linhas de comandos e que ocorre um erro no comando 20. Se você não tiver os comandos em *script*, terá de redigitar os 19 comandos que já haviam sido executados sem erro.

A execução do arquivo *script* se dá por meio do MySQL interativo, a partir da *commandline*, em arquivo *script* e a partir de um script *shell*. Veja a seguir.

Por meio do MySQL interativo

O comando para a execução do *script* é:

SOURCE <endereço>nomeArquivoScript.sql

O endereço é a localização do arquivo no disco. Recomenda-se sempre utilizar o endereço completo. Quando o endereço é omitido, o arquivo de *script* será procurado somente no diretório padrão, ou seja, onde foi submetida a execução. Por exemplo:

Source C:\Program Files\Microsoft\mySQL\scripts\ criausuario.sql

A partir da *commandline*

A submissão a partir da *commandline* implica a utilização do cliente `mysql`. Deve-se ter uma conexão ativa. Veja:

mysql -u root -p <C: \ Usuários \ Usuário \ Desktop \ nomeArquivoScript.sql

Em casos de erros, o *script* para e não executa as instruções a partir do erro. Em alguns casos, é possível permitir a execução mesmo em casos de erro. Nessas condições, deve-se utilizar o parâmetro `-f ou --f`. Veja:

mysql -u root -p <C: \ Usuários \ Usuário \ Desktop \ nomeArquivoScript.sql - f

ou

mysql -u root -p <C: \ Usuários \ Usuário \ Desktop \ nomeArquivoScript.sql --f

Em arquivo *script*

Um arquivo *script* pode executar o comando `SOURCE` para acionar outros *scripts* durante a execução, mas isso não é aconselhável.

A partir de um *script shell*

A linha de execução do *script* SQL pode ser uma linha de comando em um arquivo *shell*. Dessa forma, é possível utilizar algoritmos para atender a necessidades, com instruções de decisão e repetição. Veja um exemplo:

```
EXDB="$(mysql -u root -pPASSWORD -h localhost --silent -N -e 'show databases')"
SESSION_LOG="/tmp/$$.log"
scpSHELL(){
for DB in $ EXDB; do
   mysql -u root -pPASSWORD -h localhost $DB < criausuario.sql
done
   clean
}
scpSHELL > $SESSION_LOG
```

Saiba mais

Script shell, ou simplesmente *script*, é um arquivo, executado pelo nome, que contém uma sequência de um ou mais comandos. Há vários programas *shell*: Bourne Shell, Korn Shell e C Shell.

Procedimentos adicionais

O resultado da execução dos comandos pode:

- ser mostrado em um *pager* por meio do parâmetro `|more`
- SOURCE <endereço>nomeArquivoScript.sql | more
- ser gravado em um arquivo, mas para isso deve-se fornecer o nome do arquivo na execução do *script*
- SOURCE <endereço>nomeArquivoScript.sql > arquivoSaida.out

Para gerar o formato de saída interativo, deve-se usar `mysql -t`. Já para apresentar na tela a saída das instruções que são executadas, deve-se usar `mysql -v`.

Manutenção de usuários

No banco de dados, todos os usuários devem ser identificados para ter acesso aos recursos. Dessa forma, o DBA tem como função manter os usuários no sistema de banco de dados. Para manter o usuário, o SGBD, por meio da linguagem de consulta estruturada (*Structured Query Language* [SQL]), disponibiliza recursos para criação, alteração e exclusão.

Aqui, você deve considerar a sintaxe apresentada pelo SGBD MySQL, mas todos os demais SGBDs possuem instrução semelhante, visto que a SQL é padronizada pelo Instituto Americano de Padronização (ANSI), o que a torna uma linguagem padrão para os bancos de dados que a utilizam.

Fique atento

Caso você tenha dúvidas na composição dos comandos SQL, sempre busque as informações disponíveis no *site* oficial de documentação do SGBD que escolher.

Criação de usuários

A criação de um usuário nos SGBDs é efetivada por meio do comando `CREATE USER`. O DBA deve definir o perfil do usuário para atribuir-lhe a melhor configuração. É recomendado definir um padrão de utilização para o ambiente, de acordo com as necessidades de operação. As opções de configuração do `CREATE USER` são apresentadas na seguinte sintaxe:

```
CREATE USER [IF NOT EXISTS]
  nomeUsuario [opçãoAutenticação] [, nomeUsuário [opçãoAutenticação]] ...
  DEFAULT ROLE nomeLista [, nomeLista ] ...
  [REQUIRE {NONE | opçãoTLS [[AND] opçãoTLS] ...}]
  [WITH opçãoRecurso [opçãoRecurso] ...]
  [opçãoSenha | opçãoBloqueio] ...
```

O `nomeUsuário` representa a identificação e o nome do *host*. A identificação e o nome do *host* devem estar entre aspas e separados pelo caractere especial @. Assim: `'identificação'@'nomeHost'`.

A formação `do nomeUsuario` pode ser com letras, números e caracteres especiais. Além disso, a opção Autenticação define o método de autenticação:

```
IDENTIFIED BY 'senha'
  | IDENTIFIED BY PASSWORD 'senhaHash'
  | IDENTIFIED WITH pluginAutenticação
  | IDENTIFIED WITH pluginAutenticação AS 'senhaHash'
```

A seguir, veja as definições.

- `IDENTIFIED BY`: servidor atribui *plug-in* implicitamente e atribui a senha especificada.
- `IDENTIFIED BY PASSWORD`: servidor atribui *plug-in* implicitamente e atribui o valor *hash* (obtido pelo método `password()`) a uma *string* que deseje a senha.

- `IDENTIFIED WITH`: servidor atribui *plug-in* especificado e a conta não tem senha.
- `IDENTIFIED WITH pluginAutenticação AS 'senhaHash'`: servidor atribui *plug-in* especificado e a conta não tem senha. Armazena a senhaHash, se informada.
- `DEFAULT ROLE nomeLista [, nomeLista ] ...`: define a lista de funções que deverão estar ativas após a autenticação do usuário. A lista de funções equivale à inclusão de um usuário a um grupo.
- `opçãoTLS`: especifica opções relacionadas à conexão. Quando usado o parâmetro NONE a conta não possui requisito SSL ou X 509. Veja:

SSL
| X509
| CIPHER 'cipher'
| ISSUER 'emissor'
| SUBJECT 'objeto'

- `opçãoRecurso`: define limite no uso de recursos do servidor. Veja:

MAX_QUERIES_PER_HOUR valor
| MAX_UPDATES_PER_HOUR valor
| MAX_CONNECTIONS_PER_HOUR valor
| MAX_USER_CONNECTIONS valor

- `opçãoSenha`: define parâmetros para gerenciamento de senhas. Veja:

PASSWORD EXPIRE [DEFAULT | NEVER | INTERVAL N DAY]
| PASSWORD HISTORY {DEFAULT | N}
| PASSWORD REUSE INTERVAL {DEFAULT | N DAY}
| PASSWORD REQUIRE CURRENT [DEFAULT | OPTIONAL]

- `opçãoBloqueio`: bloqueia e desbloqueia contas. Veja:

ACCOUNT LOCK
| ACCOUNT UNLOCK

Suponha que, em um ambiente, o DBA definiu que todo usuário final, quando criado, tem a seguinte configuração em sua conta:

- deve participar do grupo `USUARIOSVISAO`;
- é obrigado a criar uma nova senha no primeiro acesso;
- pode reutilizar senhas em um espaço de 60 dias;
- pode exigir redefinição de senha a cada 30 dias.

O comando de criação do usuário, segundo a especificação, seria:

```
CREATE USER 'ana.silva'@'localhost' IDENTIFIED BY '123456'
    DEFAULT ROLE USUARIOSVISAO
    PASSWORD EXPIRE
    PASSWORD REUSE INTERVAL 60 DAY
    PASSWORD EXPIRE INTERVAL 30 DAY;
```

Fique atento

O nome de contas de usuários é armazenado na tabela **mysql_user**. Cada linha da tabela identifica o usuário, o nome do *host*, o *password* e os privilégios. O SGBD trata a identificação do usuário como *case sensitive*, ou seja, diferencia letras maiúsculas de letras minúsculas. Já para o nome do *host*, isso não se aplica.

Alteração de usuários

A alteração de um usuário nos SGBDs é efetivada por meio do comando `ALTER USER`, cujas opções de configuração são apresentadas na seguinte sintaxe:

```
ALTER USER [IF EXISTS]
    nomeUsuario [opçãoAutenticação] [, nomeUsuário [opçãoAutenticação]] ...
    [REQUIRE {NONE | opçãoTLS [[AND] opçãoTLS] ...}]
    [WITH opçãoRecurso [opçãoRecurso] ...]
    [opçãoSenha | opçãoBloqueio] ...
```

```
ALTER USER [IF EXISTS] USER() opçãoAutenticaçãoUsuario

ALTER USER [IF EXISTS]
    user DEFAULT ROLE
    {NONE | ALL | nomeLista [,nomeLista] ...}
```

Onde:

- `nomeUsuário` representa a identificação e o nome do *host*.
- `opçãoAutenticação` define o método de autenticação.

```
IDENTIFIED BY 'identificação'
[REPLACE 'identificaçãoAtual']
[RETAIN CURRENT PASSWORD]
| IDENTIFIED WITH pluginAutenticação
| IDENTIFIED WITH pluginAutenticação BY 'senha'
[REPLACE 'senhaAtual']
[RETAIN CURRENT PASSWORD]
| IDENTIFIED WITH pluginAutenticação AS 'senhaHash'
| DISCARD OLD PASSWORD
```

- `opçãoAutenticaçãoUsuario` modifica senhas.

```
IDENTIFIED BY 'senha'
[REPLACE 'senhaAtual']
[RETAIN CURRENT PASSWORD]
| DISCARD OLD PASSWORD
```

- `opçãoTLS` especifica opções relacionadas à conexão. Quando é usado o parâmetro NONE, a conta não possui requisito SSL ou X 509.

```
SSL
| X509
| CIPHER 'cipher'
| ISSUER 'emissor'
| SUBJECT 'objeto'
```

- `opçãoRecurso` define limite no uso de recursos do servidor.

```
MAX_QUERIES_PER_HOUR valor
| MAX_UPDATES_PER_HOUR valor
| MAX_CONNECTIONS_PER_HOUR valor
| MAX_USER_CONNECTIONS valor
```

- `opçãoSenha` define parâmetros para gerenciamento de senhas.

```
PASSWORD EXPIRE [DEFAULT | NEVER | INTERVAL N DAY]
| PASSWORD HISTORY {DEFAULT | N}
| PASSWORD REUSE INTERVAL {DEFAULT | N DAY}
| PASSWORD REQUIRE CURRENT [DEFAULT | OPTIONAL]
```

- `opçãoBloqueio` bloqueia e desbloqueia contas.

```
ACCOUNT LOCK
| ACCOUNT UNLOCK
```

Por exemplo, suponha que, em um ambiente, o DBA resolveu que a partir de hoje todos os usuários deverão trocar as suas senhas e estar inseridos nos grupos `VENDEDOR` e `ASSISTENTE`. O comando de alteração do usuário, segundo a especificação, seria:

```
ALTER USER 'ana.silva'@'localhost'
    DEFAULT ROLE VENDEDOR, ASSISTENTE
    PASSWORD EXPIRE;
```

Exclusão de usuários

A exclusão de um usuário nos SGBDs é efetivada por meio do comando `DROP USER`, cujas opções de configuração são apresentadas na seguinte sintaxe:

```
DROP USER [IF EXISTS] nomeUsuario [,nomeUsuario] ...
```

A utilização da cláusula `IF EXISTS` implica o envio de um aviso em caso de erro na execução. Quando se exclui um usuário, todos os recursos disponibilizados a ele devem ser excluídos. No caso do SGBD MySQL, os privilégios são eliminados, mas o DBA deve se encarregar de excluir os objetos criados pelo usuário, pois essa não é uma operação realizada automaticamente. Sugere-se que os

objetos utilizados na produção do negócio sejam criados com o usuário ADMIN, que não será eliminado, pois é o usuário criado na instalação do banco de dados.

Fique atento

Quando ocorre a eliminação de um usuário que está com alguma seção aberta, o SGBD aguarda o fechamento da seção para que a eliminação seja finalizada. A partir daí, uma nova tentativa de conexão será negada.

Criação de listas

Uma lista representa um usuário com várias funções autorizadas para uso. Depois que o usuário é criado, é possível inseri-lo em listas por meio da cláusula `DEFAULT ROLE`. A definição das listas deve ser estabelecida a partir do tipo de visibilidade que o conjunto de usuários pode ter dos objetos. A administração de listas facilita a manutenção e padroniza o uso dos recursos, mas ela nem sempre é possível.

Pode-se adicionar quantos usuários e/ou listas forem necessárias. A criação de uma lista é feita por meio do comando `CREATE ROLE`, como você pode ver a seguir:

```
CREATE ROLE [IF NOT EXISTS nomeLista [, nomeLista] ...
```

Como exemplo, suponha que os estagiários da área financeira de uma empresa devem ter a mesma visibilidade dos usuários do banco de dados. Pode-se criar uma lista para esses estagiários do departamento financeiro. Nesse caso, o comando seria:

```
CREATE ROLE ESTAG-FIN;
```

Alteração de listas

A alteração das configurações das listas deve ser realizada a partir do comando `ALTER USER`.

Exclusão de listas

A exclusão de listas é realizada a partir do comando `DROP ROLE`, conforme sintaxe a seguir:

```
DROP ROLE [IF EXISTS] nomeLista [, nomeLista] ...
```

Referências

ABNT. *ABNT NBR ISO/IEC 27002*: tecnologia da informação: técnicas de segurança: código de prática para a gestão da segurança da informação. Rio de Janeiro: ABNT, 2005.

AURÉLIO. *Qual é o significado de Visão?* Dicionário do Aurélio Online 2018. 2018. Disponível em: https://dicionariodoaurelio.com/visao. Acesso em: 12 abr. 2019.

CAMPOS, A. L. N. *Sistemas de segurança da informação*: controlando riscos. 2. ed. Florianópolis: Visual Books, 2007.

CARDOSO; F. E.; OLIVEIRA, P. C. *Política de segurança da informação nas empresas*. Ourinhos: FATEC, 2013. Disponível em: https://s.profissionaisti.com.br/wp-content/uploads/2013/06/Politica-de-Seguran%C3%A7a-nas-Empresas.pdf. Acesso em: 12 abr. 2019.

DANTAS, M. *Segurança da informação*: uma abordagem focada em gestão de riscos. Olinda: Livro rápido, 2011. Disponível em: http://www.marcusdantas.com.br/files/seguranca_informacao.pdf Acesso em: 12 abr. 2019.

FONTES, E. *Praticando a segurança da informação*. Rio de Janeiro: Brasport, 2008.

GRAVES, M. *Projeto de Banco de Dados com XML*. São Paulo: Pearson, 2003.

MACEDO, D. *Administração de usuários e privilégios no banco de dados*. Diego Macedo: um pouco de tudo sobre T. I., 2011. Disponível em: https://www.diegomacedo.com.br/administracao-de-usuarios-e-privilegios-no-banco-de-dados/. Acesso em: 12 abr. 2019.

MOURA, R. G. *Business Intelligence*. 2000. Monografia (Graduação em Ciência da Computação) — Centro Universitário do Triângulo, Uberlândia, 2000. Disponível em: http://www.computacao.unitri.edu.br/downloads/monografia/39861143143599.pdf. Acesso em: 12 abr. 2019.

MYSQL. *CREATE USER Syntax*. 2019. Disponível em: https://dev.mysql.com/doc/refman/8.0/en/create-user.html. Acesso em: 12 abr. 2019.

MYSQL. *DROP USER Syntax*. 2019. Disponível em: https://dev.mysql.com/doc/refman/8.0/en/drop-user.html. Acesso em: 12 abr. 2019.

MYSQL. *UPDATE USER Syntax*. 2019. Disponível em: https://dev.mysql.com/doc/refman/8.0/en/alter-user.html. Acesso em: 12 abr. 2019.

MYSQL. *Using mysql in Batch Mode*. 2019. Disponível em: https://dev.mysql.com/doc/refman/8.0/en/batch-mode.html. Acesso em: 12 abr. 2019.

PROOF. *Qual a importância da conscientização de usuários para a Segurança da Informação?* Rio de Janeiro, 2019. Disponível em: https://www.proof.com.br/blog/conscientizacao-de-usuarios-seguranca-da-informacao/. Acesso em: 12 abr. 2019.

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

SOMMERVILLE, I. *Engenharia de software*. 8. ed. São Paulo: Pearson; Addison Wesley, 2007.

Leituras recomendadas

BRYLA, B.; LONEY, K. *Oracle Database 11G*: Manual do DBA. Porto Alegre: Bookman, 2009.

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

Conteúdo:

SAGAH
SOLUÇÕES EDUCACIONAIS INTEGRADAS