

# Conceitos de Banco de Dados

## Aula 09 - Linguagem SQL - Segurança

# Apresentação

---

Caro aluno, na aula anterior, você estudou os conceitos de funções, como criar, executar e apagar essas estruturas. Nesta aula, você estudará segurança de sistema e segurança de banco de dados. Em segurança de sistemas, você verá como criar uma conta de usuário no *MySQL* e como acessar o sistema *MySQL* a partir de uma determinada conta. Em seguida, estudará segurança de banco de dados. Aprenderá como controlar o que os usuários podem fazer com os objetos (tabelas, visões, funções e *stored procedures*) baseados nos privilégios atribuídos a cada usuário. Finalizando nossa aula, você verá o comando que revoga permissões dos usuários.



## **Vídeo 01** - Apresentação

### Objetivos

- Criar contas no sistema *MySQL*.
- Acessar o sistema *MySQL* a partir da linha de comando do Windows.
- Conceder permissões aos usuários.
- Revogar permissões dos usuários.

# Segurança

---

Em ambientes com múltiplos usuários, é importante proteger o banco de dados de alterações indevidas nos dados ou nas estruturas das tabelas, as quais podem comprometer a integridade do banco de dados. Além disso, evita o acesso de determinados usuários a dados sigilosos, como, por exemplo, a folha de pagamento dos empregados de uma empresa. Com esse propósito, os SGBD possuem um conjunto de regras e mecanismos de proteção de acesso ao banco de dados denominado segurança ou autorização.

A segurança em banco de dados pode ser classificada em duas categorias, vejamos.

- **Segurança de sistema:** relaciona-se com o controle de acesso ao banco de dados no nível de sistema, como, por exemplo, nome de usuário e senha.
- **Segurança de banco de dados:** relaciona-se com o controle de uso dos objetos do banco de dados e as ações que esses usuários podem realizar sobre os objetos.



**Vídeo 02** - Sistema de Autenticação no *MySQL*

## Segurança de sistema

---

Até o momento, trabalhamos com apenas um único usuário em nosso banco de dados, o usuário *root*, que por definição é o primeiro usuário do SGBD. Como nos demais sistemas, o usuário *root* possui o controle completo sobre o banco de dados,

tendo inclusive a permissão de incluir novos usuários no banco de dados.

Mas, como permitir que mais usuários utilizem o banco de dados? Todos eles devem acessar o banco de dados através da conta *root*? Permitir o acesso a um sistema através de uma única conta com todas as permissões, como a conta *root*, é geralmente perigoso. Cada usuário deve possuir um *login*, ao qual está associada uma conta de acesso ao banco de dados com determinadas permissões, conforme for o caso.

Então, como proceder para adicionar uma nova conta de acesso? A sintaxe para adicionar uma conta ao sistema *MySQL* é descrita no destaque a seguir.

```
1 mysql> CREATE USER nome_da_conta  
2 IDENTIFIED BY 'password';
```

Perceba que o comando `CREATE USER` cria uma nova conta no *MySQL*. Para cada conta criada, esse comando insere uma nova linha na tabela `mysql.user` (notação que indica a tabela `user` do banco chamado `mysql`) sem qualquer privilégio. A tabela `mysql.user` é mantida pelo SGBD e contém informações sobre todas as contas de acesso (login, senha e o que cada conta tem permissão de fazer em cada banco de dados).

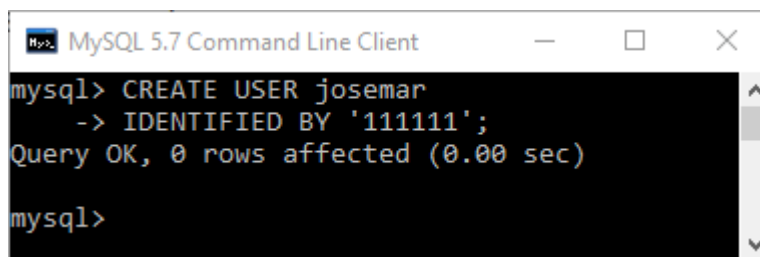
No *MySQL*, o usuário é constituído de um nome mais o *host* de onde ele poderá acessar o servidor do banco de dados (`usuario@host`). Caso você não informe o *host* para o usuário, o *MySQL* assumirá "%", isto é, todos os *hosts*.

Vamos criar, inicialmente, um usuário com *login* **josemar** e senha (*password*) 111111 no SGBD *MySQL*? Para criarmos a conta `josemar`, utilizamos o seguinte comando.

```
1 mysql> CREATE USER josemar  
2 IDENTIFIED BY '111111';
```

A resposta do sistema SGBD, no caso *MySQL*, para o comando `CREATE USER`, é ilustrada na **Figura 1**.

**Figura 01** - Tela do MySQL após o comando CREATE USER.



```
mysql> CREATE USER josemar
-> IDENTIFIED BY '111111';
Query OK, 0 rows affected (0.00 sec)

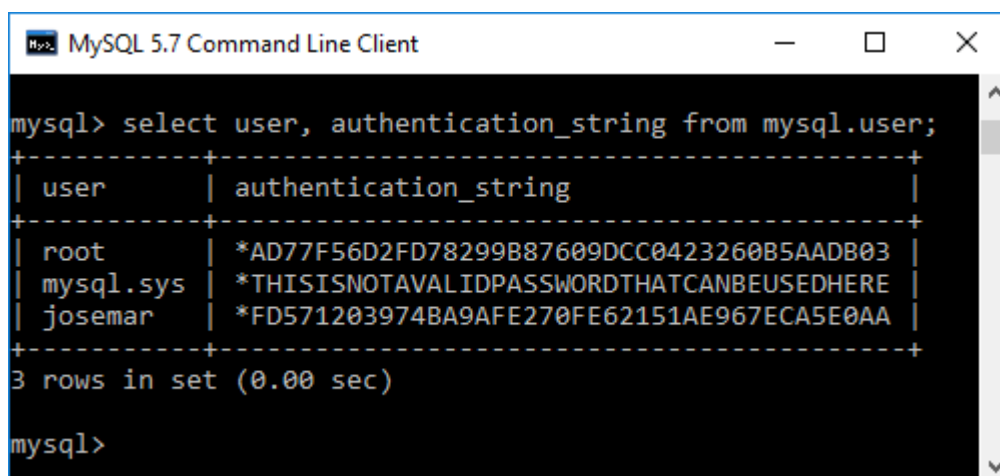
mysql>
```

**Fonte:** MySQL 5.7 Command Line Client

É importante esclarecer que a linguagem SQL não especifica como gerenciar os usuários. A criação dos usuários varia de sistema para sistema. Sendo necessário consultar a documentação para encontrar o comando correto para criar um usuário em cada SGBD.

Para visualizarmos os atributos *user* e *authentication\_string* (password) da tabela *mysql.user*, utilizamos o comando SELECT, conforme apresentado na **Figura 2**. Observe que as informações referentes ao campo password estão codificadas.

**Figura 02** - Tela do MySQL após o comando SELECT.



```
mysql> select user, authentication_string from mysql.user;
+-----+-----+
| user      | authentication_string |
+-----+-----+
| root      | *AD77F56D2FD78299B87609DCC0423260B5AADB03 |
| mysql.sys | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE |
| josemar   | *FD571203974BA9AFE270FE62151AE967ECA5E0AA |
+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

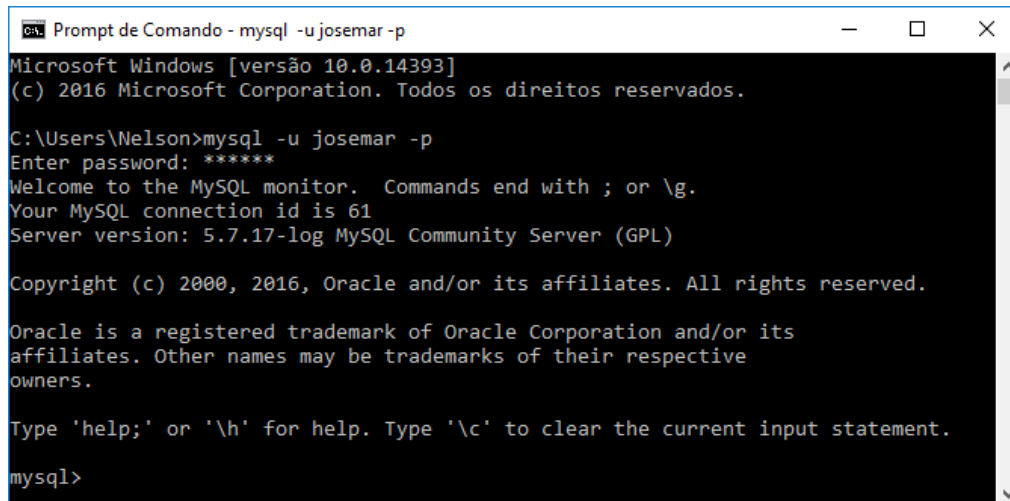
**Fonte:** MySQL 5.7 Command Line Client

Você deve estar se perguntando: “Como proceder para acessar o banco de dados com uma conta diferente da *root*?”. Bom, para ter acesso ao MySQL a partir de uma conta qualquer e começar a interagir com seu banco de dados, você deve procurar o prompt de comando do *Windows*, acessível diretamente pelo menu Iniciar/Acessórios do seu ambiente *Windows*. Ao clicar no ícone, aparecerá a tela da linha de comando, na qual se deve digitar o seguinte comando para entrar no SGBD MySQL.

```
1 >mysql -u login_do_usuario -p;
```

Como resposta a esse comando, será solicitada a senha do usuário. Digite a senha e tecla *Enter*. O sistema SGBD será acessado, aparecendo a tela de boas-vindas do ambiente *MySQL*, conforme você pode ver na **Figura 3**.

**Figura 03** - Tela inicial do *MySQL*.



```
Prompt de Comando - mysql -u josemar -p
Microsoft Windows [versão 10.0.14393]
(c) 2016 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Nelson>mysql -u josemar -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 61
Server version: 5.7.17-log MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

**Fonte:** Prompt de Comando

## Atividade 01

---

1. Adicione um novo usuário no seu SGBD.
2. Acesse o banco de dados com a conta criada na questão 1.

## Segurança de banco de dados

---

Para executar qualquer atividade em um banco de dados, o usuário deve ter as permissões adequadas. Diferentemente da conta *root*, os novos usuários que forem criados utilizando o comando *CREATE USER*, conforme foi descrito, não possuem permissão para executar nenhum comando SQL. Portanto, para cada novo usuário, é necessário especificar quais dados e comandos ele terá a permissão de acessar e utilizar, evitando assim o uso não autorizado, através da concessão de permissão.

Para conceder permissão no *MySQL*, você deve utilizar o comando **GRANT**. Esse comando concede permissões específicas no objeto (tabela, visão, função e *stored procedures*) para um ou mais usuários ou grupos de usuário. Essas permissões são adicionadas às já concedidas, caso existam. A sintaxe resumida do comando **GRANT** é exibida no destaque a seguir.

```
1 mysql> GRANT lista_de_privilegios ON lista_do_objeto TO lista-de-usuarios ;
```

No comando mostrado anteriormente, você pode observar que o primeiro item a ser informado é a lista de privilégios a serem concedidos aos usuários. Os privilégios mais comuns são:

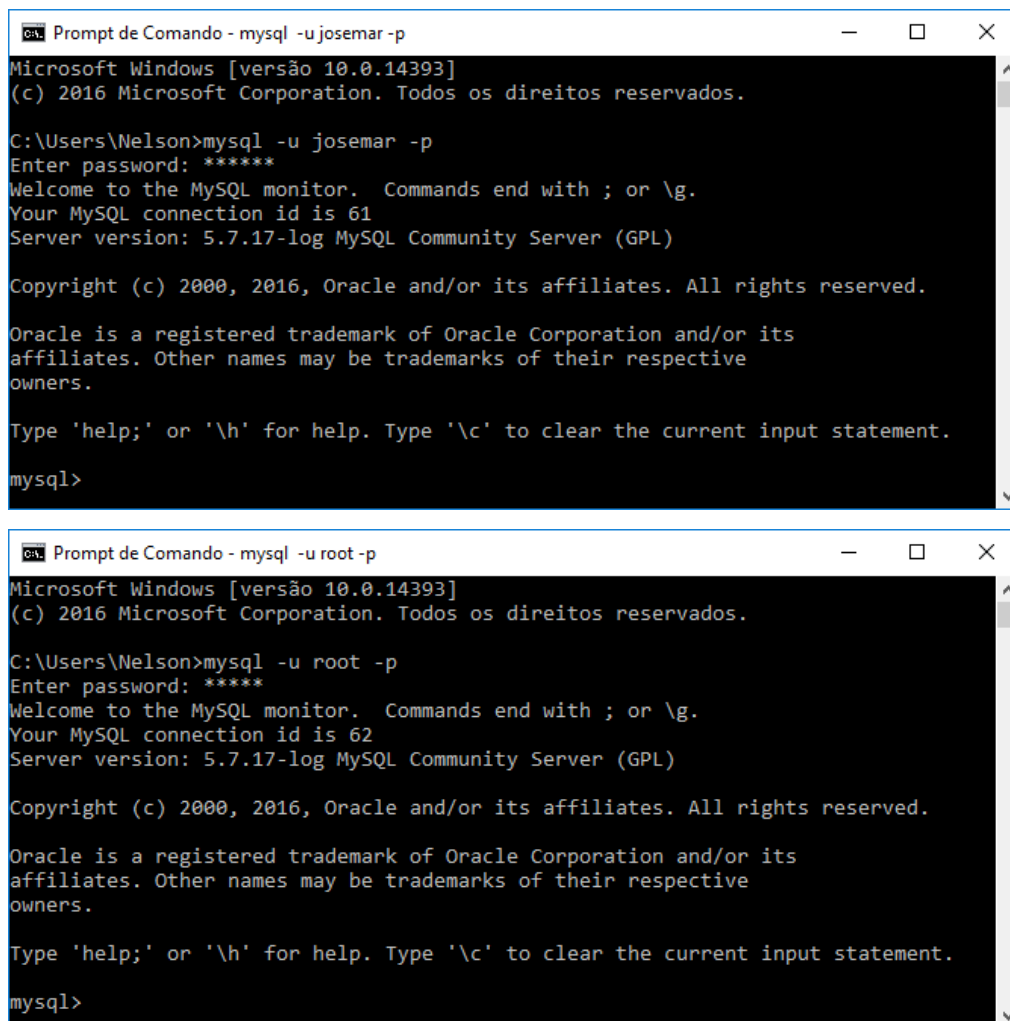
- **SELECT**: permite consultar qualquer coluna da tabela, visão ou sequência especificada.
- **INSERT**: permite incluir novas linhas na tabela especificada.
- **DELETE**: permite excluir linhas da tabela especificada.
- **UPDATE**: permite modificar os dados de qualquer coluna da tabela especificada.
- **ALTER**: permite modificar a estrutura da tabela especificada.
- **CREATE**: permite criar tabelas.
- **DROP**: permite excluir tabelas especificadas.
- **SHOW DATABASES**: permite exibir todos os bancos de dados.
- **ALL**: concede todos os privilégios descritos nessa lista de uma só vez.

Uma vez informados os privilégios do usuário, deverá ser indicada a lista de objetos a qual o privilégio se aplica, sendo possível especificar três níveis:

- \*.\* - Privilégio global.
- db.\* - Qualquer tabela do banco de dados denominado de **db**.
- db.tb - Apenas a tabela **tb** do banco de dados **db**.

Depois da lista de objetos, deverá ser indicada a lista de usuários, para os quais os privilégios se aplicam.

Vamos praticar o comando GRANT, concedendo sucessivamente diversas permissões ao usuário **josemar**, utilizando para tanto uma janela de linha de comando do sistema MySQL conectado como *root*. Para efeito de verificação das permissões concedidas, abrimos uma segunda janela de linha de comando do sistema MySQL, conectado como o usuário **josemar** (**Figura 4**).



```
Prompt de Comando - mysql -u josemar -p
Microsoft Windows [versão 10.0.14393]
(c) 2016 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Nelson>mysql -u josemar -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 61
Server version: 5.7.17-log MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

```
Prompt de Comando - mysql -u root -p
Microsoft Windows [versão 10.0.14393]
(c) 2016 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Nelson>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 62
Server version: 5.7.17-log MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```



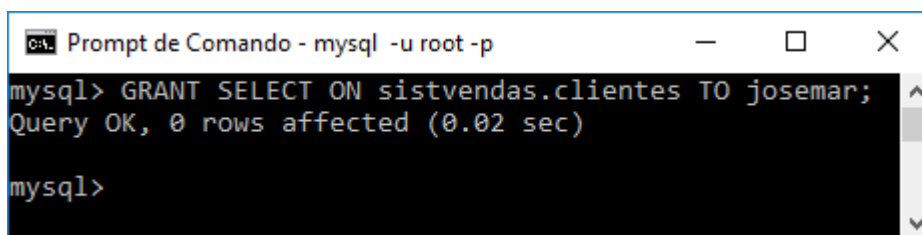
**Figura 04** - Telas do ambiente *MySQL* para os usuários *root* e **josemar**. **Fonte:** Prompt de Comando

Inicialmente, usando a conta *root*, vamos conceder ao usuário **josemar** o direito de selecionar os dados da tabela **clientes** do nosso banco de dados **sistvendas**.

```
1 mysql> GRANT SELECT ON sistvendas.clientes TO josemar;
```

A resposta do sistema *MySQL* ao comando é *QUERY OK*, a qual informa que o comando foi executado com sucesso, conforme ilustrado na **Figura 5**.

**Figura 05** - Tela do *MySQL* após o comando *GRANT SELECT*.



```
mysql> GRANT SELECT ON sistvendas.clientes TO josemar;
Query OK, 0 rows affected (0.02 sec)

mysql>
```

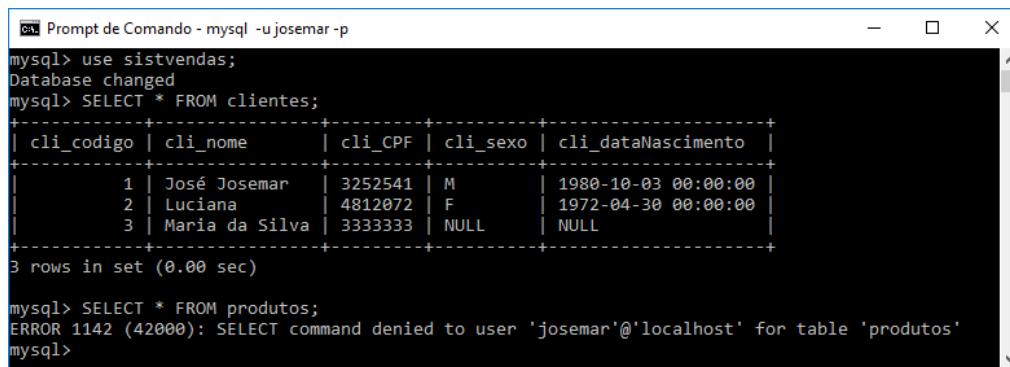
**Fonte:** Prompt de Comando

Nesse momento, é interessante verificar a permissão concedida ao usuário **josemar**. Para tanto, utilizando a janela do *MySQL* conectado com o usuário **josemar**, você irá selecionar todos os dados da tabela **clientes** através do comando *SELECT*, conforme é apresentado no quadro a seguir. Lembre-se de antes informar ao sistema que deseja trabalhar com o banco de dados **sistvendas** utilizando o comando *USE*. Para rever este banco de dados, você pode consultar a **Aula 9**.

```
1 mysql> SELECT * FROM clientes;
```

A resposta do sistema ao comando anterior é ilustrada na **Figura 6**. Conforme pode ser visualizado na figura, o usuário **josemar** pode selecionar e visualizar os dados da tabela **clientes**. Entretanto, não é permitido a esse usuário visualizar as informações pertencentes a nenhuma outra tabela desse banco de dados (**produtos** e **compras**). Caso esse usuário tente selecionar os dados das tabelas **produtos** ou **compras**, terá como resposta do sistema a mensagem que o uso do comando *SELECT*, nessas tabelas, foi negado ao usuário **josemar**, conforme ilustrado na **Figura 6**.

**Figura 06** - Tela do MySQL após os comandos SELECT.



```
Prompt de Comando - mysql -u josemar -p
mysql> use sistvendas;
Database changed
mysql> SELECT * FROM clientes;
+-----+-----+-----+-----+-----+
| cli_codigo | cli_nome | cli_CPF | cli_sexo | cli_dataNascimento |
+-----+-----+-----+-----+-----+
| 1 | José Josemar | 3252541 | M | 1980-10-03 00:00:00 |
| 2 | Luciana | 4812072 | F | 1972-04-30 00:00:00 |
| 3 | Maria da Silva | 3333333 | NULL | NULL |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> SELECT * FROM produtos;
ERROR 1142 (42000): SELECT command denied to user 'josemar'@'localhost' for table 'produtos'
mysql>
```

**Fonte:** Prompt de Comando

Para que o usuário **josemar** tenha acesso aos dados de todas as tabelas pertencentes ao banco de dados **sistvendas**, você deve conectar como *root* e executar o seguinte comando.

```
1 mysql> GRANT SELECT ON sistvendas.* TO josemar;
```

Até o momento, o usuário **josemar** só tem permissão de visualizar os dados nas tabelas pertencentes ao banco de dados **sistvendas**, mas não tem permissão de incluir, atualizar ou excluir nenhum registro nessas tabelas. Para que esse usuário tenha permissão para realizar tais tarefas, é necessário que o usuário *root* lhe conceda as permissões através do seguinte comando.

```
1 mysql> GRANT INSERT, UPDATE, DELETE ON sistvendas.* TO josemar;
```

É importante destacar que para especificar apenas algumas colunas de uma determinada tabela, essas deverão ser listadas ao lado do privilégio (*priv (colunas)*).

Para maior esclarecimento, você deve analisar o exemplo a seguir.

```
1 mysql> GRANT SELECT (cli_nome) ON locadora.clientes TO josemar;
```

Nesse exemplo, é concedido ao usuário **josemar** a permissão de visualizar apenas os nomes dos clientes do banco de dados **locadora**.

Vale salientar que não são apenas as tabelas que podem ser objeto de permissões, essas podem ser estendidas para outros objetos do banco de dados como as visões e as *stored procedures* vistas nas aulas anteriores, conforme mostra o exemplo a seguir.

```
1 mysql> GRANT SELECT ON pagamentos.funcionario TO josemar;
```

Nesse exemplo, **funcionario** é uma visão pertencente ao banco de dados **pagamentos**. Consulte a **Aula 08** para rever este banco de dados.

O usuário *root* também pode conceder a qualquer usuário o direito de repassar para um terceiro o privilégio concedido a ele. Para isso, basta acrescentar a cláusula **WITH GRANT OPTION** no final de um comando **GRANT** qualquer. Veja o seguinte exemplo:

```
1 mysql> GRANT ALL ON cineonline.* TO josemar WITH GRANT OPTION;
```

Com esse comando, o usuário **josemar** não só possui todos os privilégios em todas as tabelas do banco de dados **cineonline**, como pode conceder, a outro usuário, qualquer um dos seus privilégios nas tabelas do banco de dados **cineonline**.

É importante esclarecer que mesmo que o usuário possua várias permissões, ele só poderá conceder, a outros usuários, aqueles privilégios que lhe forem atribuídos com a cláusula **WITH GRANT OPTION**.

Suponha que o usuário **josemar**, por alguma razão, não deve mais ter acesso ao banco de dados **locadora**. O que se deve fazer? Excluir esse usuário e novamente adicioná-lo, concedendo novamente todas as suas permissões com exceção daquelas referentes ao banco de dados **locadora**? Embora isso seja possível, não é a maneira mais prática.

De maneira semelhante ao que foi utilizado para conceder privilégios a usuários, existe um comando para remover ou excluir privilégios concedidos. O comando **REVOKE** permite ao administrador de sistemas (usuário *root*) revogar permissões concedidas.

A sintaxe do comando **REVOKE** é semelhante à sintaxe do comando **GRANT**, entretanto, ao invés de utilizar a palavra **GRANT**, utiliza-se a palavra **REVOKE**, e ao invés de **TO**, utiliza-se **FROM**, conforme ilustrado no quadro a seguir.

```
1 mysql> REVOKE lista_de_privilegios ON lista_do_objeto  
2 FROM lista-de-usuarios;
```

Para revogar toda e qualquer permissão que o usuário **josemar** tenha sobre as tabelas do banco de dados **locadora**, o usuário *root* deve executar o comando a seguir.

```
1 mysql> REVOKE ALL ON locadora.* FROM josemar;
```

Para revogar um privilégio com a cláusula **WITH GRANT OPTION** no qual o usuário pode conceder seus direitos a outros usuários, você deve utilizar o comando semelhante ao exemplo a seguir.

```
1 mysql> REVOKE GRANT OPTION ON cineonline.*  
2 FROM josemar;
```

Nesse exemplo, o usuário **josemar** ainda poderá utilizar todos os comandos nas tabelas do banco de dados **cineonline**, mas não poderá conceder a mais ninguém a permissão de utilizar os comandos SQL.

Suponha que o usuário Josemar passou a permissão de atualizar dados de **cineonline**(UPDATE) para outro usuário chamado Pedro. Quando a permissão de atualizar (UPDATE) de Josemar é revogada (comando a seguir), também é revogada a permissão de Pedro de atualizar tabelas.

```
1 mysql> REVOKE UPDATE ON cineonline.*  
2 FROM josemar;
```

É importante destacar que é possível unir os comandos **CREATE USER** e **GRANT** em um só comando, criando um usuário e lhe concedendo as permissões devidas. Para tanto, você deve utilizar o comando **GRANT** acrescido da cláusula **IDENTIFIED BY**, conforme o exemplo a seguir.

```
1 mysql> GRANT SELECT ON sistvendas.* TO Jose  
2 IDENTIFIED BY '22222';
```

No comando anterior, criamos um usuário chamado Jose no nosso sistema de banco de dados e lhe concedemos a permissão de visualizar todos os dados contidos no banco de dados **sistvendas**.



### Vídeo 03 - Comandos GRANT e REVOKE

## Atividade 02

---

Considerando o banco de dados **pagamentos**, desenvolvido na **Aula 07**, contendo tabelas (**empregados**, **pagamentos** e **descontos**) e visões (**funcionario** e **salario**), faça o que se pede.

1. Crie os seguintes usuários: João, Maria e José.
2. Usando comandos em SQL, fixe as seguintes regras de privilégios para os usuários:
  - a. João deve ter acesso às tabelas podendo alterar, inserir e excluir dados. Porém, ele terá acesso somente às tabelas e a nenhuma visão.
  - b. Maria deve ter acesso somente à tabela **clientes**. Ela poderá inserir e alterar dados na tabela, mas não excluí-los.
  - c. José poderá apenas visualizar os dados das tabelas clientes e **descontos**.
  - d. João concede a José os direitos de alterar, inserir e excluir dados na tabela **clientes**.
  - e. Maria perde o direito de acessar e visualizar os dados da tabela **clientes**.
  - f. João perde o direito de acessar a tabela **clientes**. Com isso, José também deve perder os direitos concedidos por João.

## Definindo Permissões

---



### **Vídeo 04** - Definindo Permissões

## Conclusão

---

Encerramos por aqui nossa aula sobre segurança de sistemas e dados na linguagem SQL. Na próxima aula, aprenderemos como integrar uma aplicação desenvolvida em Java com o seu banco de dados *MySQL*.

Faça a autoavaliação com atenção e veja se precisa parar e refletir mais um pouco sobre como modelar, criar e manipular dados utilizando a linguagem SQL. É uma boa prática escrever no seu caderno todos os comandos SQL (e respectivas funções) que você estudou para não esquecer.

Bons estudos e boa sorte!

## Resumo

---

Nesta aula, você estudou sobre segurança de sistemas e de banco de dados. Em segurança de sistemas, viu que o comando `CREATE USER` cria uma conta de usuário no *MySQL*. Aprendeu como ter acesso ao *MySQL* a partir de uma conta qualquer. Em segurança de banco de dados, estudou o comando `GRANT`, que permite controlar exatamente o que os usuários podem fazer com os objetos (tabelas, visões, funções e *stored procedures*) baseados nos privilégios atribuídos a cada usuário. Estudou, ainda, como utilizar o comando `REVOKE` para revogar as permissões de um usuário.

## Autoavaliação

---

1. Considere o banco de dados **CursoX**, criado na Autoavaliação da **Aula 03**, cuja estrutura de tabelas é mostrada a seguir:

ATRIBUTO	TIPO	DESCRIÇÃO
<b>aluno_cod</b>	Número inteiro	Código do aluno
<b>aluno_nome</b>	Alfanumérico	Nome do aluno
<b>aluno_endereco</b>	Alfanumérico	Endereço do aluno
<b>aluno_cidade</b>	Alfanumérico	Cidade do aluno

**Tabela:** Alunos

ATRIBUTO	TIPO	DESCRIÇÃO
<b>dis_cod</b>	Número inteiro	Código da disciplina

ATRIBUTO	TIPO	DESCRIÇÃO
<b>dis_nome</b>	Alfanumérico	Nome da disciplina
<b>dis_carga</b>	Número inteiro	Carga horária da disciplina
<b>dis_professor</b>	Alfanumérico	Professor da disciplina

**Tabela:** Disciplina

ATRIBUTO	TIPO	DESCRIÇÃO
<b>prof_cod</b>	Número inteiro	Código do professor
<b>prof_nome</b>	Alfanumérico	Nome do professor
<b>prof_endereco</b>	Alfanumérico	Endereço do professor
<b>prof_cidade</b>	Alfanumérico	Cidade do professor

**Tabela:** Professores

Considere os comandos a seguir e as tabelas pertencentes ao banco de dados **CursoX**:

- CREATE USER prof IDENTIFIED BY = '111111';
- CREATE USER coord IDENTIFIED BY = '222222';
- CREATE USER maria IDENTIFIED BY = '333333';
- CREATE USER marcos IDENTIFIED BY = '444444';
- GRANT SELECT ON Cursox.alunos TO marcos;



- GRANT ALL ON Cursox.\* TO coord WITH GRANT OPTION;
- GRANT SELECT, UPDATE (aluno\_endereco, aluno\_cidade) ON Cursox.alunos TO Maria;
- GRANT SELECT, UPDATE, INSERT ON Cursox.professores TO Maria;
- REVOKE SELECT ON Cursox.alunos TO marcos;
- REVOKE INSERT ON Cursox.professores TO maria;

Considerando a execução dos comandos citados, responda às questões propostas.

- a. Quais os nomes das pessoas que podem se conectar ao banco de dados **CursoX**? O que cada uma delas está autorizada a fazer nesse banco de dados? Explique.
- b. O que o usuário **maria** pode fazer?
- c. O usuário **coord** poderá conceder a outro usuário permissão para atualizar a tabela **professores**? Explique.
- d. O usuário **marcos** poderá cadastrar um novo professor? Explique.
- e. O usuário **maria** poderá cadastrar um novo aluno? Explique.

## Referências

---

BEIGHLEY, L. **Use a cabeça SQL**. Rio de Janeiro: Editora AltaBooks, 2008.

MySQL 5.7 Reference Manual. Disponível em: <<http://dev.mysql.com/doc/refman/5.7/en/>>. Acesso em: 28 jan. 2017.

WIKIPÉDIA. **SQL**. Disponível em: <<http://pt.wikipedia.org/wiki/SQL>>. Acesso em: 26 set. 2012.