**Segurança**

O objetivo deste tutorial é demonstrar como se configura e implementa a concessão de privilégios, controle de acesso e backup no SGBD MySQL.

Carlos Henrique Reis - 30415

Mateus Henrique Toledo - 34849

Victor Rodrigues da Silva - 31054

**1. Estudar e praticar o controle de acesso de usuário em seu banco.**

Para que algum usuários acessem o SGBD, deve ser informado o nome de usuário e a senha, da seguinte forma:

mysql -h [ip\_maquina] -u <usuário> -p

O comando acima basicamente executa o MySQL, sendo que -h informa o *host*, -u informa o usuário (no caso *root*) e -p informa que deve ser solicitado a senha do usuário. Ao instalar o SGBD é definida a senha do usuário *root*, e os demais usuários podem ser criados com permissões específicas.

Para criar um usuário, deve ser executado o seguinte comando no MySQL:

CREATE USER ‘nomeusuario’;

Podem ser criados usuários com senha também, da seguinte forma (**mais recomendado**):

CREATE USER ‘nomeusuario’@’host\_name’ IDENTIFIED BY ‘senha’;

Podemos observar que a criação do usuário não lhe concede a permissão no banco de dados, sendo que estas permissões devem ser cedidas posteriormente ao usuário.

**2. Estudar e praticar a concessão de privilégios em seu banco.**

Para que o usuário criado possa realizar determinadas ações no banco de dados devem ser concedidos privilégios para ele. Esses privilégios podem ser concedidos em diferentes níveis, seguindo a sintaxe básica:

GRANT <privilégios> ON <itens> TO ‘<usuario>’@‘<ip da máquina cliente>';

Caso o usuário informado não exista no banco, é necessário incluir IDENTIFIED BY '[senha]’ no fim do comando criado acima. **Mesmo sendo possível criar um usuário utilizando o GRANT, isso não é recomendado**.

**2.1 Tipos de privilégios**

* **I. Privilégios a nível global:**

Privilégios a nível global são privilégios administrativos ou aplicados a todos os bancos de dados. Para especificar um privilégio como global basta utilizar **\*.\*** na sintaxe, ou seja, o usuário terá privilégio para todos os bancos e todas as suas tabelas. Podem ser cedidos os seguintes privilégios neste nível: CREATE TABLESPACE, CREATE USER, FILE, PROCESS, RELOAD, REPLICATION CLIENT, REPLICATION SLAVE, SHOW DATABASES, SHUTDOWN, e SUPER.

Exemplos:

*GRANT ALL ON \*.\* TO ‘someuser’@’somehost’;*

*GRANT UPDATE,DELETE ON \*.\* TO ‘someuser’@’somehost’;*

* **II. Privilégios a nível de banco:**

À nível de banco podem ser especificados os privilégios CREATE, DROP, EVENT, GRANT OPTION, LOCK TABLES e REFERENCES.

Exemplos:

*GRANT ALL ON nomebanco.\* TO nomeusuario;*

*GRANT CREATE ON nomebanco.\* TO nomeusuario;*

* **III. Privilégios a nível de tabela:**

Os privilégios à nível de tabela são aplicados a todas as colunas da tabela. Os privilégios que podem ser concedidos a nível de tabela são ALTER, CREATE VIEW, CREATE, DELETE, DROP, GRANT OPTION, INDEX, INSERT, REFERENCES, SELECT, SHOW VIEW, TRIGGER, e UPDATE.

Exemplo:

*GRANT ALL ON nomebanco.nometabela TO nomeusuario;*

* **IV. Privilégios a nível de coluna:**

Os privilégios a nível de coluna são cedidos de uma forma um pouco diferente. Ao especificar que operação pode ser feita após o GRANT, as colunas que podem ser afetadas por essa operação são indicadas entre parênteses, como no exemplo abaixo. Os privilégios específicos a nível de tabela são INSERT, REFERENCES, SELECT, e UPDATE.

Exemplo:

*GRANT SELECT (coluna1), UPDATE (coluna2,coluna3) ON nomebanco.nometabela TO nomeusuario;*

* **V. Privilégios a nível de rotinas salvas:**

Os privilégios ALTER ROUTINE, CREATE ROUTINE, EXECUTE e GRANT OPTION podem ser aplicados em *procedures* e funções. Exceto para CREATE ROUTINE os privilégios são cedidos a nível global ou de banco.

Exemplos:

*GRANT CREATE ROUTINE ON nomebanco.\* TO ‘someuser’@’somehost’;*

*GRANT EXECUTE ON PROCEDURE nomebanco.nomeprocedure TO 'someuser'@'somehost';*

* **VI. Privilégios *proxy*:**

Os privilégios proxy podem ser cedidos conforme descrito no link:<https://dev.mysql.com/doc/refman/5.7/en/grant.html>

**3. Como listar os usuários cadastrados no banco?**

Para listar os usuários cadastrados no banco, utiliza o comando abaixo:

SELECT user FROM mysql.user;

**4. Como listar os privilégios dos usuários cadastrados no banco?**

Para listar os privilégios de um usuário cadastrado, basta utilizar o comando:

SHOW GRANTS FOR ‘user’@’host’;

**5. Criar uma view no banco. Dar privilégios para o usuário apenas na view e verificar como o banco de dados é apresentado para esse usuário. O que ele pode ver?**

Ele poderá realizar somente operações sobre a *view,* qualquer outra operação no banco é negada.

**6. Verificar:**

**a) O SGBD permite configurar acessos em diferentes redes? Por exemplo, se o usuário estiver na rede da empresa, ele tem determinados privilégios. Caso contrário, ele possui outros privilégios.**

Sim, o MySQL permite configurar acessos baseado na rede em que o usuário se encontra. Basta criar usuários com o mesmo nome e senha, porém com *host* diferente e conceder privilégios diferentes para cada *host*.

**b) O SGBD permite configurar privilégios para grupos de usuários? Como?**

As versões estáveis do MySQL, não possuem suporte para configurar privilégios para grupos de usuários, porém, as versões mais recentes (instáveis) possuem a funcionalidade ROLE. Como no exemplo abaixo:

CREATE ROLE administrador;

GRANT SHOW DATABASES ON \*.\* TO administrador;

GRANT administrador to carlos;

**c) Veja o exemplo: o usuário deve ter acesso a todas as tabelas de um banco, exceto a tabela ‘x’. Neste caso, é possível dar privilégios ao banco de dados e posteriormente remover o privilégio apenas da tabela x?**

Sim, é possível dar privilégios ao banco de dados. Exemplos:

REVOKE ALL PRIVILEGES ON TABLE Table\_1 FROM PUBLIC CASCADE;

REVOKE EXECUTE ON SPECIFIC ROUTINE some\_routine FROM sam CASCADE;

**7. Uma boa prática de segurança em qualquer SGBD é manter sempre o**

**backup atualizado. Verifique como é o sistema de backup do banco. Quais opções ele dá? É possível agendar? É possível salvar o backup como binário? Faça um teste e verifique a diferença de tamanho dos arquivos.**

O backup no MySQL é totalmente personalizável. O agendamento pode ser realizado com algum tipo de *script* para que o sistema operacional execute o *mysqldump* diariamente ou por meio de alguma variável *date* configurada diretamente no comando para executar o *dump* (mais complexo).

Como existem diversas opções para o backup, serão apresentadas algumas das mais utilizadas no banco de dados teste chamado *world*:

1. *Backup* de um banco de dados específico ou todos os bancos:

mysqldump -u <usuario> -p <nomebanco> > <nomearquivo>.sql

Ex.: mysqldump -u root -p world > worlddump.sql

Ex.: mysqldump -u root -p --all-databases > dump.sql

2. *Backup* de uma tabela específica do banco:

mysqldump -u <usuario> -p <nomebanco> <nometabela> > <nomearquivo>.sql

Ex.: mysqldump -u root -p world country > countriesdump.sql

3. *Backup* com algumas tabelas do banco:

mysql -u <usuario> -p <nomebanco> <tabela1 tabela2 ... tabelaN> > nomearquivo.sql

Ex.: mysql -u root -p world country countrylanguage > countrytables.sql

4. Como restaurar o *backup* do banco:

Mysql -u <usuario> -p < <nomearquivo>

Ex.: mysql -u root -p < worlddump.sql

Outra forma de salvar o *backup* do banco é por *log* binário, sendo essa forma bem mais complexa. O *log* binário guarda todas as alterações DDL ou DML realizadas no banco desde o momento em que ele foi habilitado. A configuração é feita editando o arquivo *my.cnf* e também há um executável do MySQL para *log* binário, o *mysqlbinlog*.