**MYSQL – SEGURANÇA E CONTROLE DE ACESSO**

LUCAS LAMOUNIER GONCALVES DUARTE - 2016012688

KEVIN VIEIRA PEREIRA - 2016015385

RODRIGO APARECIDO SILVA MAIA - 2016013095

Itajubá, 20 de março de 2018

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

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

**QUESTÃO 3: Como listar os usuários cadastrados no banco?**

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

**QUESTÃO 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?**

**QUESTÃO 6: Verificar:**

1. **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 SGBD permite que privilégios estejam associados ao endereço IP que o usuário está usando para fazer a conexão no banco. No exemplo a seguir daremos privilégios para o usuário KEVIN acessar a tabela financeira do banco de dados empresa1 apenas quando ele estiver em uma máquina dentro do host da empresa.

GRANT INSERT, SELECT, REMOVE

ON empresa1.financeiro

TO [KEVIN@”%.empresa1.com.br](mailto:KEVIN@)”

IDENTIFIED BY “KEVIN”

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

Sim, o SGBD permite que o root configure privilégios tanto para usuários específicos, como para grupos de usuários. Essa segunda opção é denominada “Roles”. A seguir é apresentado um exemplo de uma Role sendo criada e definindo privilégios para os usuários que estiverem dentro dela.

Primeiro vamos criar a Role:

CREATE ROLE

ADMINISTRADOR;

Em seguida ao invés de atribuirmos privilégios a um usuário, vamos atribuir este privilégios a Roler ADMINISTRADOR que criamos no passo anterior.

GRANT CREATE, INSERT, SELECT

ON empresa1.\*

TO ADMINISTRADOR;

Agora que temos uma Role criada e com os seus privilégios definidos vamos criar um usuário modelo para inserirmos neste grupo ADMINISTRADOR.

GRANT ADMINISTRADOR

TO LUCAS, RODRIGO, KEVIN;

Da mesma maneira que atribuímos privilégios para uma Role, podemos remover utilizando o comando REVOKE na Role ADMINISTRADOR, que os usuários LUCAS, RODRIGO e KEVIN perdem os privilégios automaticamente.

**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, você deve dar permissão total no banco com um GRANT dando os privilégios desejados a todo o banco, e logo após dar um REVOKE na tabela que deseja remover o acesso do usuário. Segue abaixo um exemplo de como é realizado este procedimento.

GRANT SELECT, INSERT, DELETE

ON empresa1.\*

TO LUCAS;

Permissão dada para o usuário LUCAS acessar todo o banco “empresa1”. Agora existe uma tabela relacionada aos dados financeiros da “empresa1” que o usuário LUCAS não deve ter permissão de acessar. Para isso daremos um REVOKE na tabela financeiroempresa1 a seguir.

REVOKE SELECT, INSERT, DELETE

ON empresa1.financeiro

FROM LUCAS;

**QUESTÃO 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**

**Tipos de backups:**

**Backup Físico:** Consiste em copiar os diretórios e arquivos que contém os conteúdos do banco de dados, é recomendado para base de dados importantes que necessitam ser recuperados rapidamente quando um problema ocorre;

Comando do Backup: no terminal de comando, executar o comando para se encontra o MySQL, no caso desse computador é necessário executar o comando **cd C:\xampp\mysql\bin** primeiro, após isso pode-se executar o comando:

**FLUSH TABLES listaDeTabelas WITH READ LOCK**; este comando faz com que as tabelas estejam disponíveis apenas para a leitura e não escrita, possibilitando assim a cópia física das tabelas sem risco de inserção de dados;

\*Pode ser usado o comando **SELECT \* INTO OUTFILE 'nomeDoArquivo' FROM nomeDaTabela**, este comando salvará a tabela escolhida no formato de texto, contento apenas os dados e não a estrutura da tabela. Salvará apenas no servidor o arquivo, e para a realização do backup do banco inteiro será necessário realizar o comando para cada tabela, apesar de ser lento para ser feito, a recuperação do banco pode chegar a ser 20 vezes mais rápida que qualquer outro método;

**Backup Lógico:** Salva informações representadas de maneira lógica pelo banco, como os comandos usados na base de dados, **create table, insert**; mais adequado para bancos em que possa haver a necessidade de se editar o conteúdo;

Comando do Backup: no terminal de comando, executar o comando para se encontra o MySQL, no caso desse computador é necessário executar o comando **cd C:\xampp\mysql\bin** primeiro, após isso pode-se executar o comando:

**mysqldump -u nomeDoUsuario -p nomeDoBanco> nomeDoArquivo.sql**, depois é necessário fornecer a senha do banco;

**Backup Incremental**: consistem em salvar apenas as mudanças dado um ponto no tempo, para ser possível isso é necessário habilitar o **binary** **log** do servidor, que permite que o servidor salve a data das mudanças e quais comandos foram realizados;

Comando do Backup: depois de ativado o **binary log,** será necessário escrever os logs em um arquivo para que o MySQL seja capaz de ler, então para tal será usado o comando no terminal de comandos **mysqlbinlog binlog.001 > Localizacao/nomeDoArquivo.sql** e depois será executado o comando para realizar tal backup **mysql -u nomeDoUsuário -p -e "Localizacao/nomeDoArquivo.sql";**

**Snapshot**: esse tipo permite que seja o feito o backup lógico dado em um ponto no tempo, sem realizar a cópia física do sistema inteiro; é disponibilizado por terceiros como Veritas, LVM e ZFS;

**Opções para o Backup:**

**Online**: backup pode ser realizado quando o servidor está funcionando, é menos intrusivo para os clientes, pois como o servidor vai estar funcionando ainda, será possível para os mesmos acessar os dados da base de dados;

**Offline**: nesse caso o servidor dever estar parado para ser possível realizar o backup, os clientes podem ser afetados, caso desejem ter acesso aos dados;

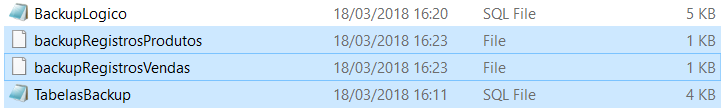
**Local**: o backup é iniciado no mesmo host em que a aplicação MySQL está sendo executado;

**Remoto**: é possível iniciar o backup em um host diferente do que contém a aplicação;

\*As opções de Agendamento, Criptografar, e comprimir o backup, não está para o MySQL em si, apenas para o MySQL Enterprise Backup, versão empresarial do MySQL; terceiros podem possibilitar tais ações também;

**Diferenças de Tamanhos:**

Foi realizado tanto o backup lógico, como o físico, os 3 itens que estão selecionados são do backup físico, em que se foi necessário realizar o backup de cada tabela e depois da estrutura do banco, enquanto o backup lógico é o primeiro item. O banco de dados utilizado possuía 2 tabelas com 10 registros cada, pode-se perceber que o backup lógico consumiu 5kb e o físico 6kb, em questão de tempo o lógico foi mais rápido por ter de se realizar apenas um comando.



**REFERÊNCIA**:

MySQL 5.7 Reference Manual, Oracle, 2018. Disponível em: <https://dev.mysql.com/doc/refman/5.7/en/data-types.html >. Acesso em 18/03/2018.