



# **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



## LISTA DE FIGURAS

FIGURA 1 - CRIANDO USUÁRIOS.....	3
FIGURA 2 - CONCEDENDO PERMISSÕES AOS USUÁRIOS .....	6
FIGURA 3- USERONE ACESSANDO A BASE DE DADOS.....	6
FIGURA 4- USERONE FAZENDO UMA CONSULTA NA TABELA PRODUTO. ....	7
FIGURA 5- USERTWO ACESSANDO A BASE DE DADOS. ....	7
FIGURA 6- EXIBINDO COLUNAS DA TABELA USER. ....	8
FIGURA 7- LISTANDO OS USUÁRIOS DO SGBD .....	8
FIGURA 8- LISTANDO PRIVILÉGIOS DO USUÁRIOS. ....	9
FIGURA 9- REMOVENDO AS PERMISSÕES DO USERONE .....	9
FIGURA 10- CRIANDO VIEW. ....	9
FIGURA 11- CONCEDENDO PERMISSÃO PARA USERONE ACESSAR A VIEW RECÉM CRIADA. 9	
FIGURA 12- ACESSO DO USUÁRIO USERTWO A BASE DE DADOS.....	10
FIGURA 13- CRIANDO USUÁRIO COM ESPECIFICAÇÃO DE HOST.....	10
FIGURA 14- CRIANDO ROLE.....	11
FIGURA 15- ATRIBUINDO PERMISSÕES PARA A ROLE.....	11
FIGURA 16- ATRIBUINDO ROLE PARA OS USUÁRIOS.....	11
FIGURA 17 - ESPAÇO OCUPADO POR CADA BACKUP. ....	14
FIGURA 18– COMANDOS PARA O BACKUP FÍSICO.....	14
FIGURA 19– COMANDOS PARA BACKUP LÓGICO.....	15



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

No MySQL os usuários são identificados pelo seu nome e host de acesso dessa forma sempre que for criado um usuário novo você deve atribuir o host de onde ele acessará. Como consequência dessa política do MySQL sempre que você quiser atribuir um novo host de acesso a um usuário existente, um novo usuário com mesmo nome deve ser criado com o host desejado especificado.

Para que o usuário possa acessar a base dados de qualquer host pode ser especificado como host o símbolo “%” que significa todos.

Para criar um usuário no MySQL a seguinte sintaxe deve ser utilizada: **CREATE USER ‘<usuário>’ @‘<host de acesso>’ IDENTIFIED BY ‘<senha>’**; Caso o host não seja especificado por padrão o símbolo “%” será atribuído ao usuário recém criado.

```
XAMPP for Windows - mysql -u root
Rodrigo Maia@LAPTOP-N9UR32SE c:\xampp\mysql\bin
# mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 10.1.21-MariaDB mariadb.org binary distribution
Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE USER 'userOne' IDENTIFIED BY 'master1';
Query OK, 0 rows affected (0.19 sec)

MariaDB [(none)]> CREATE USER 'userTwo' IDENTIFIED BY 'master2';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> create database com231;
```

*Figura 1 - Criando usuários.*

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

A concessão de privilégios no MySQL assim como os demais SGBDs é efetuada através do comando **GRANT** na seguinte sintaxe: **GRANT <lista de tipos de privilégio> (<colunas privilegiadas>) ON <objeto alvo> TO ‘<usuário>’ @‘<host>’ WITH <propriedades>**, onde:

- Lista de tipos de privilégio – São os privilégios que serão concedidos ao usuário, podendo ser:



*Tabela 1 - Privilégios disponíveis no MySQL.*

PRIVILÉGIOS	DESCRIÇÃO
<b>ALL [PRIVILEGES]</b>	Concede todos os privilégios no nível de acesso especificado, exceto GRANT OPTION e PROXY.
<b>ALTER</b>	Permite ao usuário o uso de ALTER TABLE. Níveis: Global, banco de dados, tabela.
<b>ALTER ROUTINE</b>	Permite ao usuário alterar e deletar rotinas armazenadas. Níveis: Global, banco de dados, procedimento.
<b>CREATE</b>	Permite ao usuário a criação de banco de dados e de tabelas. Níveis: Global, banco de dados, tabela.
<b>CREATE ROUTINE</b>	Permite ao usuário a criação de rotina armazenada. Níveis: Global, banco de dados.
<b>CREATE TABLESPACE</b>	Permite ao usuário a criação, alteração ou remoção de espaços de tabela e grupos de arquivos de log. Nível: Global.
<b>CREATE TEMPORARY TABLES</b>	Permite ao usuário o uso de CREATE TEMPORARY TABLE. Níveis: Global, banco de dados.
<b>CREATE USER</b>	Permite ao usuário o uso de CREATE USER, DROP USER, RENAME USER e REVOKE ALL PRIVILEGES. Nível: Global.
<b>CREATE VIEW</b>	Permite ao usuário a criação e alteração de views. Níveis: Global, banco de dados, tabela.
<b>DELETE</b>	Permite ao usuário o uso de DELETE. Nível: Global, banco de dados, tabela.
<b>DROP</b>	Permite ao usuário a exclusão de bancos de dados, tabelas e views. Níveis: Global, banco de dados, tabela.
<b>EVENT</b>	Permite ao usuário o uso de eventos para o Agendador de eventos. Níveis: Global, banco de dados
<b>EXECUTE</b>	Habilita o usuário para executar rotinas armazenadas. Níveis: Global, banco de dados, tabela.
<b>FILE</b>	Habilita o usuário à efetuar no servidor leitura ou escrita de arquivos. Nível: Global.
<b>GRANT OPTION</b>	Permite que privilégios sejam concedidos ou removidos de outras contas. Níveis: Global, banco de dados, tabela, procedimento, proxy.
<b>INDEX</b>	Permite ao usuário a criação e remoção de índices. Níveis: Global, banco de dados, tabela.
<b>INSERT</b>	Permite ao usuário o uso de INSERT. Níveis: Global, banco de dados, tabela, coluna.



<b>LOCK TABLES</b>	Permite ao usuário uso de LOCK TABLES em tabelas para as quais ele tenha o privilégio SELECT. Níveis: Global, banco de dados.
<b>PROCESS</b>	Permite que o usuário veja todos os processos com SHOW PROCESSLIST. Nível: Global.
<b>PROXY</b>	Habilita o proxy do usuário. Nível: De usuário para usuário.
<b>REFERENCES</b>	Permite ao usuário a criação de chaves estrangeiras. Níveis: Global, banco de dados, tabela, coluna.
<b>RELOAD</b>	Permite ao usuário o uso de operações FLUSH. Nível: Global.
<b>REPLICATION CLIENT</b>	Permite que o usuário pergunte onde são os servidores mestre ou escravo. Nível: Global.
<b>REPLICATION SLAVE</b>	Permite ao usuário criar escravos de replicação para ler eventos de log binário do mestre. Nível: Global.
<b>SELECT</b>	Permite ao usuário o uso de SELECT. Níveis: Global, banco de dados, tabela, coluna.
<b>SHOW DATABASES</b>	Permite ao usuário a visualização de todos os bancos na base de dados. Níveis: Global.
<b>SHOW VIEW</b>	Permite ao usuário o uso de SHOW CREATE VIEW. Níveis: Global, banco de dados, tabela.
<b>SHUTDOWN</b>	Permite ao usuário o desligamento da base dados. Nível: Global.
<b>SUPER</b>	Permite o uso de outras operações administrativas, como o comando CHANGE MASTER TO, KILL, PURGE BINARY LOGS, SET GLOBAL e mysqladmin debug. Nível: Global.
<b>TRIGGER</b>	Permite ao usuário o uso de operações de disparo. Níveis: Global, banco de dados, tabela.
<b>UPDATE</b>	Permite ao usuário o uso de UPDATE. Níveis: Global, banco de dados, tabela, coluna.
<b>USAGE</b>	Sinônimo de sem privilégios.

- Colunas privilegiadas – São as colunas que o usuário obterá os privilégios especificado anteriormente, podendo ser eles SELECT, UPDATE e etc. Obs.: Apenas utilizadas em privilégios a nível de tabela.
- Objeto alvo – Determina o nível do privilegio, que pode ser Global; banco de dados; tabela; coluna, e o alvo, nome de uma tabela, banco ou etc..
- Usuário e host – Determina o usuário que receberá os privilégios especificados anteriormente.
- Propriedades – Em geral determina limites sobre a concessão especificada anteriormente. Segue a tabela abaixo das opções:



Tabela 2- Propriedades disponiveis no MySQL

OPÇÕES	DESCRIÇÃO
<b>MAX_QUERIES_PER_HOUR</b>	Máximo de execuções por hora.
<b>MAX_UPDATES_PER_HOUR</b>	Máximo de atualizações por hora.
<b>MAX_CONNECTIONS_PER_HOUR</b>	Máximo de conexões por hora.
<b>GRANT OPTION</b>	Permitir ao usuário conceder a outros usuários o privilégio especificado anteriormente.

Assim como a concessão de privilégios é efetuada pelo comando **GRANT** a remoção é efetuada pelo comando **REVOKE** que pode ser utilizado com a seguinte sintaxe: **REVOKE** <lista de tipos de privilégio> (<colunas privilegiadas>) **ON** <objeto alvo> **FROM** '<usuário>' @ '<host>'.

Após a execução de ambos os comandos **GRANT** e **REVOKE** deve se atualizar o banco os privilégios que estão em memória, isso é possível através do comando **FLUSH PRIVILEGES**.

Imagens dos conceitos anteriores sendo aplicado no banco:

```
XAMPP for Windows - mysql -u root
MariaDB [com231]> show tables;
+-----+
| Tables_in_com231 |
+-----+
| produto          |
| venda            |
| vendedor         |
+-----+
3 rows in set (0.00 sec)

MariaDB [com231]> GRANT SELECT,INSERT ON com231.produto to userOne;
Query OK, 0 rows affected (0.05 sec)

MariaDB [com231]> GRANT SELECT ON com231.produto to userTwo;
Query OK, 0 rows affected (0.00 sec)

MariaDB [com231]>
```

Figura 2 - Concedendo permissões aos usuários

```
Prompt de Comando - mysql -u userOne -h 25.68.17.37 -p
Microsoft Windows [versão 10.0.16299.248]
(c) 2017 Microsoft Corporation. Todos os direitos reservados.

C:\Users\windows>cd C:\xampp\mysql\bin

C:\xampp\mysql\bin>mysql -u userOne -h 25.68.17.37 -p
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 24
Server version: 10.1.21-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| com231   |
| information_schema |
+-----+
2 rows in set (0.09 sec)

MariaDB [(none)]> use com231
Database changed
MariaDB [com231]> show tables;
+-----+
| Tables_in_com231 |
+-----+
| produto          |
+-----+
1 row in set (0.08 sec)

MariaDB [com231]> insert into produto values (1,'Sapato','Nike');
Query OK, 1 row affected (0.15 sec)
```

Figura 3- UserOne acessando a base de dados.



```
cmd Prompt de Comando - mysql -u userOne -h 25.68.17.37 -p

MariaDB [com231]> select * from produto;
+-----+-----+-----+
| prod_cod | prod_nome | prod_marca |
+-----+-----+-----+
| 1 | Sapato | Nike |
| 2 | Sapato | Adidas |
| 3 | Camiseta | Quick Silver |
| 4 | Oculos | Oakley |
| 5 | Meia | Puma |
| 6 | Perfume | Ferrari Black |
+-----+-----+-----+
6 rows in set (0.07 sec)

MariaDB [com231]>
```

Figura 4- UserOne fazendo uma consulta na tabela produto.

```
cmd Command Prompt - mysql -u userTwo -h 25.68.17.37 -p

MariaDB [(none)]> use com231;
Database changed
MariaDB [com231]> show tables;
+-----+
| Tables_in_com231 |
+-----+
| produto |
+-----+
4 rows in set (0.07 sec)

MariaDB [com231]> select * from produto;
+-----+-----+-----+
| prod_cod | prod_nome | prod_marca |
+-----+-----+-----+
| 1 | Sapato | Nike |
| 2 | Sapato | Adidas |
| 3 | Camiseta | Quick Silver |
| 4 | Oculos | Oakley |
| 5 | Meia | Puma |
| 6 | Perfume | Ferrari Black |
+-----+-----+-----+
6 rows in set (0.03 sec)

MariaDB [com231]> insert into produto values(7,'Calca','Levi');
ERROR 1142 (42000): INSERT command denied to user 'userTwo'@'DESKTOP-CEDUFEV' for table 'produto'
MariaDB [com231]>
```

Figura 5- UserTwo acessando a base de dados.

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

Os usuários do MySQL são armazenados fisicamente em uma tabela de nome “**user**” no banco de dados “**mysql**” presente por padrão no SGBD, dessa forma todos os usuários podem ser listado através de um comando **SELECT** na tabela **user**.

A tabela **user** também armazena privilégios globais concedidos aos usuários, então é aconselhável fazer um filtro sobre as colunas desejadas que no caso desse relatório serão as colunas **host**, **user**, e **password**.



```
XAMPP for Windows - mysql -u root
MariaDB [(none)]> use mysql;
Database changed
MariaDB [mysql]> describe user;
```

Field	Type	Null	Key	Default	Extra
Host	char(60)	NO	PRI		
User	char(80)	NO	PRI		
Password	char(41)	NO			
Select_priv	enum('N','Y')	NO		N	
Insert_priv	enum('N','Y')	NO		N	
Update_priv	enum('N','Y')	NO		N	
Delete_priv	enum('N','Y')	NO		N	
Create_priv	enum('N','Y')	NO		N	
Drop_priv	enum('N','Y')	NO		N	
Reload_priv	enum('N','Y')	NO		N	
Shutdown_priv	enum('N','Y')	NO		N	
Process_priv	enum('N','Y')	NO		N	
File_priv	enum('N','Y')	NO		N	
Grant_priv	enum('N','Y')	NO		N	
References_priv	enum('N','Y')	NO		N	
Index_priv	enum('N','Y')	NO		N	
Alter_priv	enum('N','Y')	NO		N	
Show_db_priv	enum('N','Y')	NO		N	
Super_priv	enum('N','Y')	NO		N	
Create_tmp_table_priv	enum('N','Y')	NO		N	
Lock_tables_priv	enum('N','Y')	NO		N	
Execute_priv	enum('N','Y')	NO		N	
Repl_slave_priv	enum('N','Y')	NO		N	
Repl_client_priv	enum('N','Y')	NO		N	
Create_view_priv	enum('N','Y')	NO		N	
Show_view_priv	enum('N','Y')	NO		N	
Create_routine_priv	enum('N','Y')	NO		N	
Alter_routine_priv	enum('N','Y')	NO		N	
Create_user_priv	enum('N','Y')	NO		N	
Event_priv	enum('N','Y')	NO		N	
Trigger_priv	enum('N','Y')	NO		N	
Create_tablespace_priv	enum('N','Y')	NO		N	
ssl_type	enum('', 'ANY', 'X509', 'SPECIFIED')	NO			
ssl_cipher	blob	NO		NULL	
x509_issuer	blob	NO		NULL	
x509_subject	blob	NO		NULL	
max_questions	int(11) unsigned	NO		0	
max_updates	int(11) unsigned	NO		0	

Figura 6- Exibindo colunas da tabela user.

```
XAMPP for Windows - mysql -u root
MariaDB [mysql]> select host,user,password from user;
```

host	user	password
localhost	root	
127.0.0.1	root	
:::1	root	
%	userOne	*0902208CD43C41A7897699FA332550A4C61997DE
%	userTwo	*4C6CAB89211EF945B522FD271D0419B460A37E0C

5 rows in set (0.02 sec)

```
MariaDB [mysql]>
```

Figura 7- Listando os usuários do SGBD

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

Como dito anteriormente os privilégios globais estão armazenados na tabela **user** juntamente com seus usuários, enquanto os privilégios a nível de tabela estão presente na tabela **“tables\_priv”**, com essas informações seria possível construir uma consulta que mostrasse todos os privilégios de um usuário no entanto o MySQL já disponibiliza uma ferramenta para essa função, executando o comando **show GRANTS for <usuários> @ <host>**.





```
XAMPP for Windows - mysql -u root
MariaDB [mysql]> show grants for userOne;
+-----+
| Grants for userOne@% |
+-----+
| GRANT USAGE ON *.* TO 'userOne'@'%' IDENTIFIED BY PASSWORD '*0902208CD43C41A7897699FA332550A4C61997DE' |
| GRANT SELECT, INSERT ON `com231`.`produto` TO 'userOne'@'%' |
+-----+
2 rows in set (0.00 sec)

MariaDB [mysql]> show grants for userTwo;
+-----+
| Grants for userTwo@% |
+-----+
| GRANT USAGE ON *.* TO 'userTwo'@'%' IDENTIFIED BY PASSWORD '*4C6CAB89211EF945B522FD271D0419B460A37E0C' |
| GRANT SELECT ON `com231`.`produto` TO 'userTwo'@'%' |
+-----+
2 rows in set (0.00 sec)

MariaDB [mysql]>
```

Figura 8- Listando privilégios do usuários.

**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?**

1. O usuário utilizado para responder essa questão será o userTwo, por isso inicialmente será removido todos os seus privilégios.

```
XAMPP for Windows - mysql -u root
MariaDB [com231]> revoke all on com231.* from userTwo;
Query OK, 0 rows affected (0.00 sec)
```

Figura 9- Removendo as permissões do userOne

2. Agora será criada a view a ser utilizada, que será uma view construída a partir de uma junção entre as tabelas de vendas, produto e vendedor.

```
XAMPP for Windows - mysql -u root
MariaDB [com231]> create view vendasRealizadas as select p.prod_nome, p.prod_marca, vd.vend_nome, vd.vend_cpf
-> from venda v natural join produto p
-> natural join vendedor vd;
Query OK, 0 rows affected (0.09 sec)
```

Figura 10- Criando View.

3. Concedendo permissão sobre a view para o usuário userTwo.

```
XAMPP for Windows - mysql -u root
MariaDB [com231]> grant select on com231.vendasRealizadas to userTwo;
Query OK, 0 rows affected (0.11 sec)

MariaDB [com231]>
```

Figura 11- Concedendo permissão para userOne acessar a View recém criada.



4. Acesso do usuário userTwo a base de dados.

```
Command Prompt - mysql -u userTwo -h 25.68.17.37 -p
MariaDB [(none)]> use com231;
Database changed
MariaDB [com231]> show tables;
+-----+
| Tables_in_com231 |
+-----+
| vendasrealizadas |
+-----+
1 row in set (0.11 sec)

MariaDB [com231]> select * from vendasrealizadas;
+-----+-----+-----+-----+
| prod_nome | prod_marca | vend_nome | vend_cpf |
+-----+-----+-----+-----+
| Sapato    | Nike      | Marcos Moreira | 22222222222 |
| Sapato    | Adidas    | Maria Marques  | 33333333333 |
| Meia      | Puma      | Maria Marques  | 33333333333 |
| Sapato    | Adidas    | Charles Teodoro | 44444444444 |
| Perfume   | Ferrari Black | Charles Teodoro | 44444444444 |
+-----+-----+-----+-----+
5 rows in set (0.64 sec)

MariaDB [com231]> insert into vendasrealizadas values('Camisa', 'Nike', 'Marcos Moreira', '22222222222');
ERROR 1394 (HY000): Can not insert into join view 'com231.vendasrealizadas' without fields list
MariaDB [com231]>
```

Figura 12- Acesso do usuário userTwo a base de dados.

## QUESTÃO 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 SGBD permite que configurações de acesso para diferentes hosts sejam efetuadas para os usuários, no entanto como referido na questão 1, o MySQL trata como identificador o host de acesso e o nome do usuário, dessa forma para esse tipo de configuração deve ser criado um novo usuário com o mesmo nome e senha, mas com um host diferente.

Na questão 1 foi criado usuários que possui acesso a todos os hosts nessa questão será criado um novo usuário com especificações de host.

```
XAMPP for Windows - mysql -u root
MariaDB [mysql]> create user 'userTwo'@'25.68.68.253' identified by 'master2';
Query OK, 0 rows affected (0.16 sec)

MariaDB [mysql]> grant select, insert on com231.* to userTwo;
Query OK, 0 rows affected (0.06 sec)

MariaDB [mysql]>
```

Figura 13- Criando usuário com especificação de host.

A partir de agora o usuário userTwo através da máquina de IP 25.68.68.253 é capaz de selecionar e inserir em todas as tabelas do banco



com231, enquanto em outras máquinas ele apenas é capaz de visualizar a view criada na questão 5.

## 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:

```
XAMPP for Windows - mysql -u root
MariaDB [mysql]> create role com2312;
Query OK, 0 rows affected (0.00 sec)
```

*Figura 14- Criando role.*

Em seguida ao invés de atribuirmos privilégios a um usuário, vamos atribuir estes privilégios a Roler com2312 que criamos no passo anterior.

```
XAMPP for Windows - mysql -u root
MariaDB [mysql]> create role com2312;
Query OK, 0 rows affected (0.00 sec)

MariaDB [mysql]> grant create, select, update on com231_2 to com2312;
Query OK, 0 rows affected (0.05 sec)
```

*Figura 15- Atribuido permissões para a role.*

Agora que temos uma Role criada e com os seus privilégios definidos vamos criar um usuário modelo para inserirmos neste grupo com2312.

```
XAMPP for Windows - mysql -u root
MariaDB [mysql]> create role com2312;
Query OK, 0 rows affected (0.00 sec)

MariaDB [mysql]> grant create, select, update on com231_2 to com2312;
Query OK, 0 rows affected (0.05 sec)

MariaDB [mysql]> grant com2312 to userOne, userTwo@25.68.68.253;
Query OK, 0 rows affected (0.00 sec)
```

*Figura 16- Atribuindo role para os usuários.*



Da mesma maneira que atribuímos privilégios para uma Role, podemos remover utilizando o comando REVOKE na Role com2312, que os usuários 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?**

Na versão do MySQL que está sendo utilizado pelo grupo não é permitido esse tipo de operação.

**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; existe a variação nesse comando que permite salvar em binário o backup;

\*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, está disponível apenas para a versão MySQL Enterprise Backup, a versão empresarial do MySQL; terceiros podem possibilitar tais ações também;

### Diferenças de Tamanhos:

Foi realizado tanto o backup lógico, como o físico, neste se foi necessário realizar o backup de cada tabela e depois da estrutura do banco. O banco de dados utilizado era



composto de 2 tabelas uma com todas as cidades do Brasil e o outro de todos os estados, pode-se perceber que o backup lógico consumiu 365KB, enquanto o físico 329KB (considerando os 3 itens dele, cidadesBackupFisico(325KB), estadosBackupFisico(1KB) e o backupFisico(3KB)), em questão de tempo o lógico foi mais rápido por ter de se realizar apenas um comando. Nas figuras abaixo apresentam-se as informações da quantidade de espaço ocupado por cada arquivo, e os comandos para realizar cada backup; ambos os backup foram realizados de maneira local, e não foi realizado o backup incremental pois o mesmo é necessário realizar o físico e depois para cada operação realizada se salva os conteúdos, como a única operação pedida era a de comparação de tamanho, então ele possuiria o mesmo tamanho que o backup físico.

 backupFisico	19/03/2018 15:40	SQL File	3 KB
 backupLogico	19/03/2018 15:34	SQL File	365 KB
 cidadesBackupFisico	19/03/2018 15:38	File	325 KB
 estadosBackupFisico	19/03/2018 15:39	File	1 KB

*Figura 17 - Espaço ocupado por cada backup.*

```
C:\xampp\mysql\bin>mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 20418
Server version: 10.1.21-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> use testando;
Database changed
MariaDB [testando]> select * into outfile 'cidadesBackupFisico' from cidades;
Query OK, 5565 rows affected (0.02 sec)

MariaDB [testando]> select * into outfile 'estadosBackupFisico' from estados;
Query OK, 27 rows affected (0.01 sec)

MariaDB [testando]> exit
Bye

C:\xampp\mysql\bin>mysqldump -u root -p Brasil > backupFisico.sql
Enter password:

C:\xampp\mysql\bin>
```

*Figura 18– Comandos para o backup físico.*



```
Command Prompt - mysqldump -u root -p testando
Microsoft Windows [Version 10.0.16299.248]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Kevin>cd C:\xampp\mysql\bin

C:\xampp\mysql\bin>mysqldump -u root -p Brasil > backupLogico.sql
Enter password:

C:\xampp\mysql\bin>
```

*Figura 19– Comandos para backup lógico.*

### 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.