

Fazendo backup do banco

Transcrição

No dia seguinte, tivemos uma nova reunião com os diretores da *Mutillidae*, onde foi tratado sobre o banco de dados da empresa, que contém informações em uma tabela de `produtos`, e uma tabela de `usuarios`. Mas, eles tiveram um problema que colocou em risco todas as informações do banco.

A tarefa desta aula é montar um script para fazer um backup dessas informações, para que seja possível restabelecer as informações através desse arquivo de backup, caso venha perder os dados.

Abrindo o terminal, acessaremos o **MySQL** para ver como o banco da *Mutillidae* está estruturado.

```
$ sudo mysql -u root
```

Depois, colocamos a senha do nosso usuário.

Legal! Agora que já estamos no **MySQL**, usaremos um comando que mostrará os bancos presentes:

```
mysql> show databases;
```

É retornado essa tabela de todos os bancos da empresa para nós:

```
+-----+
| Database          |
+-----+
| information_schema |
| mutillidae        |
| mysql             |
| performance_schema |
| sys               |
+-----+
5 rows in set (0.00 sec)
```

Uma vez que localizamos o banco `mutillidae`, podemos dizer que queremos "usá-lo":

```
mysql> use mutillidae
```

E em seguida, podemos perguntar sobre as tabelas desse banco:

```
mysql> show tables;
```

E como resultado, temos essas tabelas:

```
+-----+
| Tables_in_mutillidae |
+-----+
| produtos              |
| usuarios              |
+-----+
2 rows in set (0.00 sec)
```

O comando a seguir realiza uma consulta e retorna o **conteúdo** das mesmas. Primeiro, retornaremos o conteúdo da tabela `produtos` :

```
mysql> select * from produtos;
```

Dentro da tabela de `produtos` , temos informações referentes aos livros que a *Mutillidae* faz o cadastro lá na plataforma. Entre as colunas da tabela, temos `livro` , `autor` , `preco_ebook` e `preco_livro` . São no total, 50 linhas presentes na tabela `produtos` .

Agora é a vez de verificar o conteúdo da tabela `usuarios` :

```
mysql> select * from usuarios;
```

Então, temos 100 linhas de registro de dados dos usuários cadastrados na plataforma. Existem as colunas `nome` , `email` , `telefone` e `cidade` .

Agora que já vimos que temos essas duas tabelas, trabalharemos em como realizar o *backup* dessas informações.

Vamos sair com o comando `exit` , e entraremos no diretório `/Scripts` para montar um novo script de *backup*.

```
$ cd Scripts/
```

Criaremos um novo arquivo com o comando `nano` , e vamos chamá-lo de `backup-dados-mysql.sh` .

Como já sabemos, a primeira linha de cada script, será o **interpretador**! Depois, para pegarmos os dados presentes no banco de dados da *Mutillidae*, temos que fazer o **dump**.

Será feito o **dump** do banco para salvar em um arquivo de backup. No script, colocaremos assim:

```
#!/bin/bash

mysqldump -u root
```

No nosso caso, não temos uma senha configurada, e por isso temos o acesso direto.

Depois de ter especificado o usuário (`-u`), especificaremos o banco que gostaríamos de realizar o *backup* das informações. Podemos colocar tanto de forma estática (o próprio nome do banco), quanto podemos pegar essa informação como parâmetro do usuário, assim podemos pegar informação de qualquer banco, se necessário! Seria mais interessante para o nosso aprendizado escolher a segunda opção, não é mesmo?

```
#!/bin/bash

mysqldump -u root $1
```

Agora precisamos salvar a informação do **dump** em um arquivo. Com isso, vamos *redirecionar* o resultado da consulta para o arquivo de mesmo nome do banco, na qual foi passado como parâmetro.

```
#!/bin/bash

mysqldump -u root $1 > $1.sql
```

Para não ficar informações perdidas em vários lugares, concentraremos tudo em um diretório específico chamado de `/backup`.

Voltemos a `/Home`. Se não tiver o diretório `/backup` na Home, então pediremos para criar! O primeiro passo é entrar na `/Home` **pelo script** para fazer a verificação.

```
#!/bin/bash

cd /home/rafael
```

Quando essa linha for executada, estaremos na `/Home` do usuário "rafael". Faremos a verificação utilizando o `if`. "Na `/Home`, existe algum diretório chamado de `/backup`?" Se **não** existe, vamos criá-lo!

```
#!/bin/bash

cd /home/rafael
if [ ! -d backup ]
then
    mkdir backup
fi
```

Agora que já criamos esse diretório, podemos dizer que o arquivo `$1.sql` será salvo no diretório `/backup`. Para ficar um pouco mais fácil, podemos isolar o caminho em uma **constante**, assim como fizemos anteriormente:

```
#!/bin/bash

CAMINHO_HOME=/home/rafael

cd $CAMINHO_HOME
if [ ! -d backup ]
then
    mkdir backup
fi

mysqldump -u root $1 > $CAMINHO_HOME/backup/$1.sql
```

A partir daqui, nós estaremos realizando o *backup* das informações do banco de dados, que o mesmo será passado como um parâmetro pelo usuário. O *backup* terá o mesmo nome do banco original. Agora só nos resta verificar se o

comando `mysqldump` está sendo executado com sucesso, ou se tem algum problema.

Para isso, verificaremos o seu **status de saída**. Caso ele seja `0`, significa que o comando foi executado com sucesso. E caso ele seja diferente de `0`, algo de errado aconteceu.

```
mysqldump -u root $1 > $CAMINHO_HOME/backup/$1.sql
if [ $? -eq 0 ]
then
    echo "Backup foi realizado com sucesso"
else
    echo "Houve um problema no backup"
fi
```

Saímos com o "Ctrl + X" e "Y" para salvar.

Como nós fazemos o **dump** do banco da *Mutillidae*, precisamos subir o privilégio para o usuário root, para conseguirmos executar o script.

```
$ sudo bash backup-dados-mysql.sh mutillidae
```

De acordo com a mensagem retornada, não houve problemas na execução do script. Se acessarmos o diretório `/backup`, encontraremos o arquivo criado pelo nosso script, tendo o mesmo nome do banco de dados. Agora que já realizamos o *backup*, vamos ver se esse *backup* realmente funciona!

Voltemos ao banco *mutillidae* com o comando `sudo mysql -u root`.

Logo depois, vamos especificar o banco de dados que queremos usar:

```
mysql> use mutillidae
```

Faremos uma simulação PERDENDO todas as informações da tabela `produtos` e `usuarios`.

```
mysql> delete from produtos;
```

Após deletar a tabela de produtos, deletaremos a tabela de usuários:

```
mysql> delete from usuarios;
```

Agora não temos mais nenhum dado! (Lembrando que estamos simulando um problema para poder utilizar o backup). Tentaremos restabelecer as informações. Podemos sair do `mysql` com o comando `exit`.

Para restabelecer as informações, primeiro temos que dar o privilégio para o usuário root, e em seguida, diremos qual banco será restabelecido. Em seguida, pegaremos as informações de um arquivo (backup), e passaremos para o banco:

```
$ sudo mysql -u root mutillidae < ~/backup/mutillidae.sql
```

Agora, vamos voltar ao banco sql da *Mutillidae* para conferir se realmente as informações foram restabelecidas, realizando uma **consulta**.

```
$ sudo mysql -u root
```

```
mysql> use mutillidae
```

```
mysql> select * from produtos;
```

Muito bem! Temos 50 linhas de informações sobre os produtos, assim como tínhamos antes. Faremos o mesmo teste com a tabela `usuarios` :

```
mysql> select * from usuarios;
```

Foram retornados 100 linhas de informações dos usuários, e com isso, conseguimos com sucesso restabelecer as informações que estavam nas tabelas de `produtos` e `usuarios` .

Entretanto, se pensarmos melhor, o que pode acontecer? Bom, estamos fazendo o *backup* somente nessa máquina! E se por um acaso, essa máquina para de funcionar, ou tiver algum problema? O *backup* será perdido!

A próxima etapa é verificar como podemos enviar essas informações para um **serviço de Cloud**, como a Amazon.