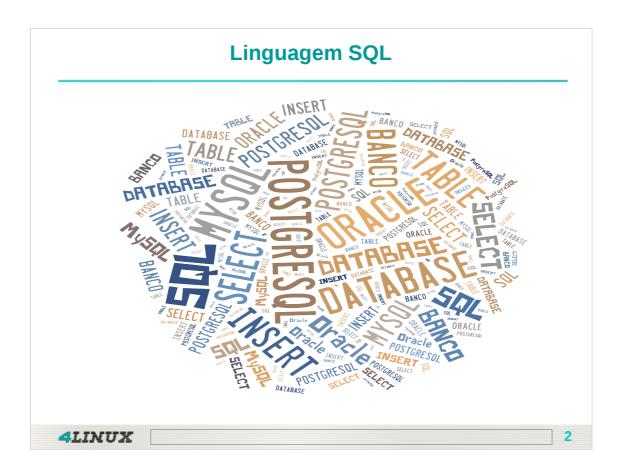


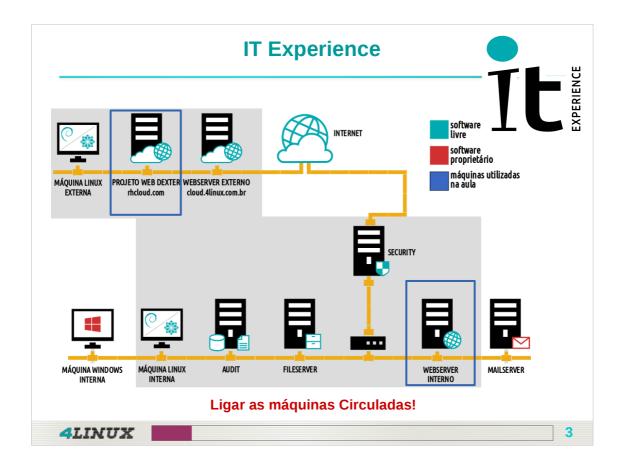
## **Curso 4451**

## Linux Security Administration in Cloud



## Linguagem SQL

"Structured Query Language", ou Linguagem de Consulta Estruturada ou SQL, é uma linguagem para banco de dados relacional.



| Anotações: |      |      |      |
|------------|------|------|------|
|            |      |      |      |
|            |      |      |      |
|            | <br> | <br> | <br> |
|            | <br> |      |      |
|            | <br> | <br> | <br> |
|            |      |      |      |

## **Objetivos da Aula**

## Aula 06

- ➤ Introdução ao SGBD;
- > Criar banco de dados e tabelas;
- Cadastrar, pesquisar e remover dados.



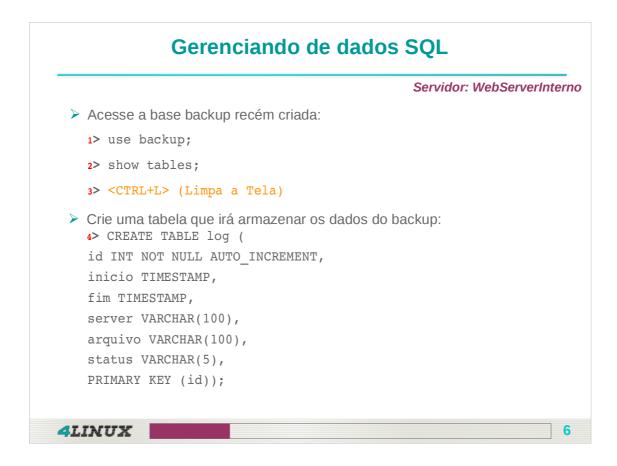
4LINUX

| Anotações: |      |      |  |
|------------|------|------|--|
|            | <br> |      |  |
|            | <br> |      |  |
|            | <br> | <br> |  |
|            | <br> | <br> |  |
|            |      |      |  |
|            |      |      |  |
|            |      |      |  |
|            |      |      |  |
|            |      |      |  |
|            |      |      |  |
|            |      |      |  |
|            |      |      |  |
|            |      |      |  |

## Gerenciando de dados SQL Servidor: WebServerInterno Introdução ao SGBD: > Um SGBD é um sistema de gestor de base de dados, que disponibiliza uma interface para que clientes (usuários) possam interagir com o banco de dados de várias maneiras, como inserir dados, pesquisar, excluir, entre outras; > Acesse o banco de dados MySQL do WebServerInterno: 1# mysql -u root -p (Senha 123456) 2> show databases; 3> CREATE database backup; 4> show databases;

A Linguagem SQL possui classificações para as instruções executadas, feitas de acordo com a função de cada instrução/comando:

- **DCL Linguagem de Controle de Dados:** Comandos SQL responsáveis pela administração/autorização de dados e licença de usuários;
- **DDL Linguagem de Definição de Dados:** Conjunto de comandos responsáveis pela criação de bancos, esquemas, tabelas e constraints.
- **DML Linguagem de Manipulação de Dados:** Comandos responsáveis por operações de manipulação como Inserir, consultar, atualizar ou excluir dados.
- **DQL Linguagem de Consulta de Dados:** O único comando que pode ser definido como DQL é o SELECT usado para realização de consultas.



O servidor MySql armazena as informações dos usuários em quatro tabelas localizadas no banco de dados padrão chamado Mysql.

## São estas:

**Tabela user:** Armazena as informações de todos os usuários do banco e seus privilégisos a nível global;

Tabela db: Armazena os privilégios dos usuários específicos de um banco de dados;

**Tabela tables\_priv e columns\_priv:** Armazenam os privilégios associados a tabelas e colunas, respectivamente;

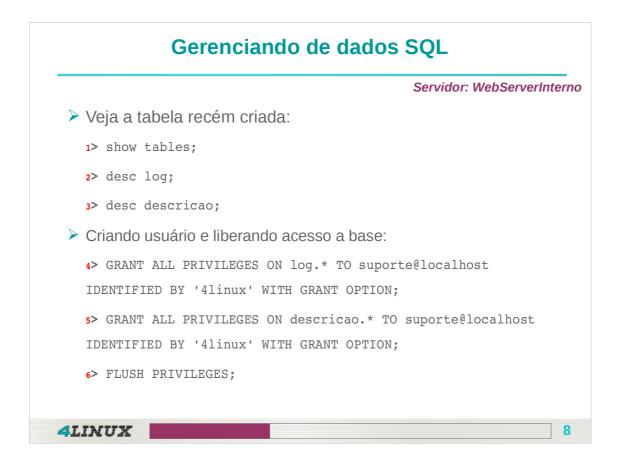
Servidor: WebServerInterno

> Crie uma tabela que irá armazenar a descrição dos servidores:

```
1> CREATE TABLE descricao (
id INT NOT NULL AUTO_INCREMENT,
descricao VARCHAR(100),
localizacao VARCHAR(6),
PRIMARY KEY (id));
```

4LINUX

| Anotações: |      |
|------------|------|
|            | <br> |
|            |      |
|            |      |
|            |      |
|            |      |
|            |      |
|            |      |
|            |      |
|            |      |
|            |      |
|            |      |
|            |      |
|            |      |



## Explorando a Liberação de Acesso:

A instrução **GRANT ALL PRIVILEGES** atribui **TODOS** os privilégios à todas as tabelas do banco "log" ao usuário "suporte", a partir da máquina "localhost", cuja senha é "4linux".

Muito cuidado com a opção **WITH GRANT OPTION**, pois ela garante ao usuário suporte, a possibilidade de atribuir permissões a outros usuários do banco, ou seja, o usuário suporte possui privilégios de administração sobre a tabela log.

A instrução **"FLUSH PRIVILEGES"** atualiza as novas alterações no "daemon" do MySQL. Caso o usuário "suporte" não exista, um novo usuário será criado.

Servidor: WebServerInterno

Inserindo dados na tabela log:

```
1> INSERT INTO log (inicio,fim,server,arquivo,status) VALUES
('2015-03-04 15:21:21','2015-03-04 15:25:21','intranet', 'etc-
2015-03-04-15:21:21.tar.gz','OK');
```

```
2> INSERT INTO log (inicio,fim,server,arquivo,status) VALUES
('2015-03-04 15:21:21','2015-03-04 15:25:21','intranet', 'etc-
2015-03-04-15:21:21.tar.gz','FAIL');
```

**4LINUX** 

| Anotações: |  |  |
|------------|--|--|
|            |  |  |
|            |  |  |
|            |  |  |
|            |  |  |
|            |  |  |
|            |  |  |
|            |  |  |
|            |  |  |
|            |  |  |
|            |  |  |
|            |  |  |
|            |  |  |

Servidor: WebServerInterno

➤ Inserindo dados na tabela descrição:

```
1> INSERT INTO descricao (descricao,localizacao) VALUES
('Servidor WebserverInterno - Backup Server','Cloud');
```

```
2> INSERT INTO descricao (descricao,localizacao) VALUES
('Servidor Intranet - Intranet Server','Fisico');
```

4LINUX

| Anotações: |      |      |
|------------|------|------|
|            | <br> | <br> |
|            |      |      |
|            |      |      |
|            |      |      |
|            |      |      |
|            |      |      |
|            |      |      |
|            |      |      |
|            |      |      |
|            |      |      |
|            |      |      |
|            |      |      |

Servidor: WebServerInterno

> Realizando consulta nas tabelas:

Sintaxe: SELECT <campo1,campo2> FROM <nome da tabela>

- 1> select \* from log;
- 2> select \* from descricao;
- > Realizando consulta com filtro:
  - 3> SELECT server,inicio,fim FROM log WHERE status='OK';
  - 4> SELECT descricao FROM descricao WHERE localizacao='Cloud';

4LINUX

| Anotações: |      |      |
|------------|------|------|
|            | <br> | <br> |
|            | <br> | <br> |
|            |      |      |
|            |      |      |
|            | <br> | <br> |
|            |      |      |
|            | <br> | <br> |
|            |      |      |
|            |      |      |
|            |      |      |
|            |      |      |
|            |      |      |
|            |      |      |

## Gerenciando de dados SQL Servidor: WebServerInterno Realizando consulta nas tabelas: Para realizar uma pesquisa ordenada por determinada coluna: 1> select \* from log ORDER BY arquivo; Dessa forma, a consulta será ordenada por ordem alfabética com referencia ao campo arquivo.

> Realizando consulta nas tabelas:

- 2> SELECT server,inicio,fim FROM log WHERE status='OK';
- 3> SELECT descricao FROM descricao WHERE localizacao='Cloud';



| Anotações: |      |
|------------|------|
|            |      |
|            |      |
|            | <br> |

## Gerenciando de dados SQL Servidor: WebServerInterno Alterar o nome do servidor intranet para WebServerInterno: 1> UPDATE log SET server="Intranet" WHERE status='FAIL'; 2> select \* from log; Apagar o registro com FAIL da base: 1> DELETE FROM log WHERE status='FAIL'; 2> select \* from log; 3> exit

**4LINUX** 

| Anotações: |      |  |
|------------|------|--|
|            |      |  |
|            |      |  |
|            | <br> |  |
|            | <br> |  |
|            |      |  |
|            |      |  |
|            |      |  |
|            |      |  |
|            |      |  |
|            |      |  |
|            | <br> |  |
|            |      |  |

Servidor: WebServerInterno

- ➤ É extremamente importante ter um dump da base de dados, para realizar a tarefa segue, abaixo, a instrução:
- > Na linha de comando execute:
- 1# mysqldump -h localhost -u root -p123456 backup > backup.sql
- 2# cat backup.sql

Onde:

- -h localhost: Host a ser feito o dump
- -u root: Usuário que irá realizar o dump
- -p 123456: Senha para se conectar ao banco

Backup: Nome do banco de dados a ser feito o dump

- Para restaurar um backup execute:
- nysql -h localhost -u root -p123456 backup < backup.sql</pre>

4LINUX

| Anotações: |      |      |
|------------|------|------|
|            | <br> | <br> |
|            |      |      |
|            |      |      |
|            |      |      |
|            |      |      |
|            |      |      |
|            |      |      |
|            |      |      |
|            |      |      |
|            |      |      |
|            |      |      |
|            |      |      |



Servidor: WebServerInterno

- ➤ Iremos agora criar um shell script no servidor **WebServerInterno** que irá gerenciar o backup dos servidores **WebServerCloud** e **WebServerInterno**;
- Esse shell script deverá ser colocado no cron para que seja criada uma rotina de backup diário;
- Acesse no browser da máquina Linux Interna o sistema de backup web:

http://192.168.200.130/bkpreport

4LINUX 15

| Anotações: |      |  |
|------------|------|--|
|            |      |  |
|            |      |  |
|            | <br> |  |

## Laboratório Dexter Servidor: WebServerInterno Vamos criar o Shell Script que fará backup do Servidor WebServer Interno e agendá-lo no cron: 1# cd /etc 2# mkdir backup 3# cd backup 4# vim backupdexter.sh #!/bin/bash ## Validate ## Se não for especificado o Diretório como parâmetro ele cancela o script.

4LINUX

test ! -z \$1 || exit

| Anotações: |      |
|------------|------|
|            |      |
|            |      |
|            | <br> |

## ## Variáveis SERVER="\$(hostname -s)" DIRECTORY="\$1" TODAY=\$(date +%d-%m-%Y) FILE="bkp\_\$(echo \$DIRECTORY | sed -e 's/\///g')\_\$SERVER\_\$TODAY.tar.bz2" ## Begin Script # Exporta variável START registrando o horário que começa o backup export START=\$(date +%Y-%m-%d\ %H:%M:%S)

| Anotações: |      |         |
|------------|------|---------|
|            |      |         |
|            |      |         |
|            |      |         |
|            | <br> | <br>    |
|            | <br> | <br>    |
|            |      | <br>    |
|            | <br> | <br>- : |
|            |      |         |



Servidor: WebServerInterno

## Armazena no log a hora em que irá iniciar o backup.
echo -e " Backup do \$DIRECTORY no \$SERVER iniciado as \$START
" >> /var/log/backup.log

## Faz o backup usando o comando tar com BZIP2
tar cjf /backup/\$FILE \$DIRECTORY > /dev/null 2> /dev/null

4LINUX

**18** 

# Anotações:



Servidor: WebServerInterno

```
## Condição que valida se o backup foi realizado com Sucesso ou
Não.
if [ $? -eq 0 ]; then
STATUS="OK"
export END=$(date +%Y-%m-%d\ %H:%M:%S )
echo -e " Backup do $DIRECTORY no $SERVER finalizado em $END " >>
/var/log/backup.log

else
STATUS="FAIL"
export END=$(date +%Y-%m-%d\ %H:%M:%S )
echo -e " Backup do $DIRECTORY no $SERVER finalizado em $END " >>
/var/log/backup.log
fi
```

4LINUX

19

## Anotações:



Servidor: WebServerInterno

## Insere os valores no Banco para que possa aparecer no Sistema Web

```
mysql -u root -p123456 << TERMINAR
use backup;
INSERT INTO log (inicio,fim,server,arquivo,status) VALUES
('$START','$END','$SERVER','$FILE','$STATUS');
TERMINAR</pre>
```

4LINUX

20

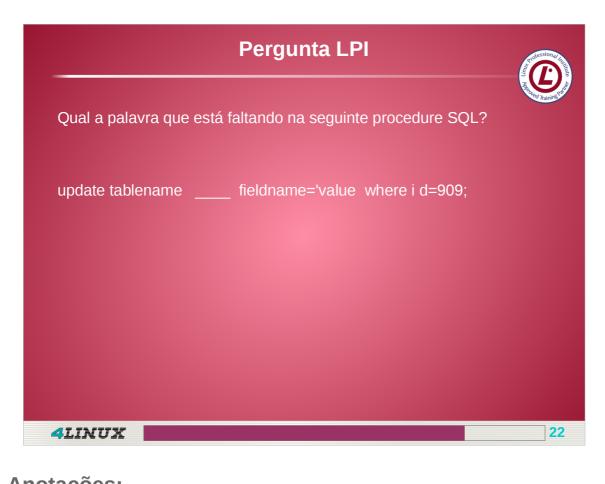
## Anotações:

Servidor: WebServerInterno

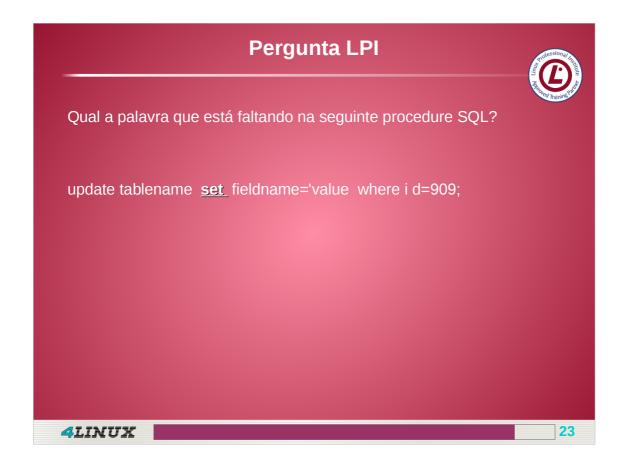
- Liberando acesso a base backup e agendando no cron:
  - 1> GRANT ALL PRIVILEGES ON backup.\* TO suporte@localhost IDENTIFIED BY '4linux' WITH GRANT OPTION;
  - 2> FLUSH PRIVILEGES;
  - 3# vim /etc/crontab
  - 00 4 \* \* \* root /etc/backup/backupdexter.sh /home
  - 10 4 \* \* \* root /etc/backup/backupdexter.sh /etc

4LINUX 21

| Anotações: |      |  |
|------------|------|--|
|            |      |  |
|            |      |  |
|            | <br> |  |
|            |      |  |
|            | <br> |  |
|            | <br> |  |
|            | <br> |  |
|            |      |  |
|            | <br> |  |
|            |      |  |
|            | <br> |  |
|            | <br> |  |



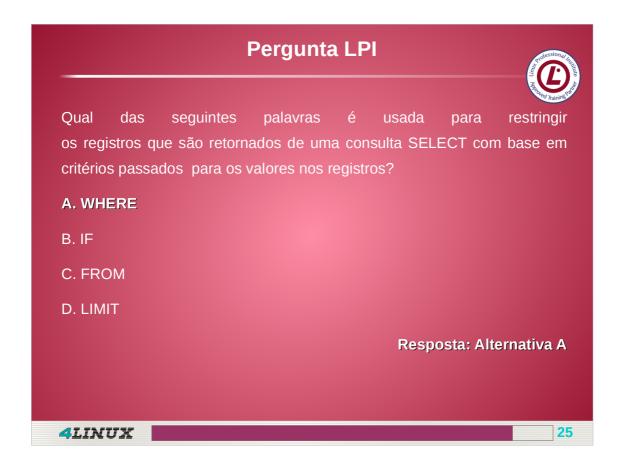
| Anotações: |                                       |      |  |
|------------|---------------------------------------|------|--|
|            |                                       |      |  |
|            |                                       | <br> |  |
|            |                                       | <br> |  |
|            |                                       | <br> |  |
|            |                                       |      |  |
|            |                                       |      |  |
|            |                                       |      |  |
|            |                                       |      |  |
|            |                                       |      |  |
|            | · · · · · · · · · · · · · · · · · · · | <br> |  |
|            |                                       |      |  |
|            |                                       |      |  |



RESPOTA CORRETA: set

## Pergunta LPI Qual das seguintes palavras é usada para restringir os registros que são retornados de uma consulta SELECT com base em critérios passados para os valores nos registros? A. WHERE B. IF C. FROM D. LIMIT

| Anotações: |      |   |
|------------|------|---|
|            |      |   |
|            |      |   |
|            | <br> | <br>· · · · · · · · · · · · · · · · · · · |
|            | <br> | <br>                                      |
|            | <br> | <br>· · · · · ·                           |
|            | <br> | <br>· · · · · · ·                         |
|            |      |   |
|            | <br> | <br>· · · · · ·                           |
|            |      |   |
|            | <br> | <br>                                      |
|            |      |   |
|            | <br> | <br>                                      |



## Alternativa A: RESPOSTA CORRETA!

A clausula **WHERE** permite a delimitação de uma consulta SQL com base em critérios lógicos.

## NOTA:

A cláusula LIMIT não foi utilizada em aula, sua função é limitar o número de resultados de uma consulta.

## **Próximos Passos**

Para que você tenha um melhor aproveitamento do curso, participes das seguintes atividades disponíveis no Netclass:

- > Executar as tarefas do Practice Lab;
- Executar a **ETAPA II** do **Dexter Project** e postar o resultado no Fórum Temático;
- Responder as questões do Teste de Conhecimento sobre o conteúdo visto em aula.

Mãos à obra!

4LINUX

