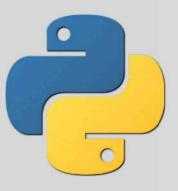


Python Fundamentals



SQL

8.1



Objetivos da Aula

- ✓ Entender o que são Banco de Dados.
- ✓ Introdução à linguagem SQL.

SQL

Nesta aula abordaremos conceitos, características e possibilidades de trabalho com a linguagem SQL.

Anotações			

4LINUX Introdução ao SQL

- Banco de dados é um conjunto de dados guardados para a extração de informações no futuro.
- A linguagem padrão universal para manipular banco de dados relacionais é o SQL (Structured Query Language).
- Esta linguagem permite interagir com um SGBD (Sistema Gerenciador de Banco de Dados) para executar tarefas como: inserir registros, gerenciar usuários, criar consultas, entre outras.



Introdução ao SQL

O conceito sobre Banco de Dados deve ser compreendido com clareza.

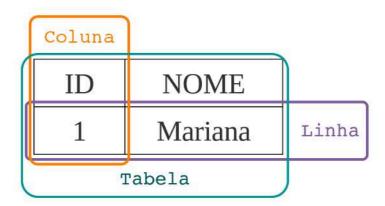
Exemplo: 09/06/2017

09/06/2017 é somente um dado, aparentemente não apresenta nenhum nenhum significado. Porém, se mudarmos o contexto para: O Cliente João foi cadastrado na data de 09/06/2017.

Atribuído um contexto para o dado extraído, traz informação com a qual podemos trabalhar. Sendo assim, utilizamos o banco de dados para conservar registros que recuperados em determinado contexto, geram informações que podem resultar em valor para o negócio.

4LINUX Introdução ao SQL

- ✓ Na computação utilizamos ferramentas SGBD (Sistemas Gerenciadores de Bancos de Dados) para armazenamento e o manuseio desses dados.
- ✓ Dados são organizados em tabelas, colunas e linhas.



Durante a maior parte do tempo como desenvolvedor, você criará ou encontrará aplicações que utilizam alguma forma de armazenamento de dados. Estes dados podem ser armazenados em simples arquivos, no sistema de arquivos de uma máquina ou, armazenados em estruturas mais complexas, como um banco de dados.

Bancos de dados, em sua grande maioria, são constituídos com base nas relações existentes entre suas entidades, daí o nome: Bancos de Dados Relacionais.

Anotações			



Anotações		

4LINUX Comandos para SGBD

DDL: Data Definition Language, para manipular a estrutura da tabela, com comandos como: CREATE, ALTER, DROP.

DML: Data Manipulation Language, manipulamos os dados, através de comandos como: INSERT, UPDATE e DELETE.

DQL: Data Query Language, para consulta de dados, por meio de comandos como: SELECT, SHOW.

Anotações	



Python Fundamentals



PostgreSQL

8.2

4LINUX PostgreSQL

Objetivos da Aula

- ✓ Conhecer o PostgreSQL.
- ✓ Trabalhar com PostgreSQL.

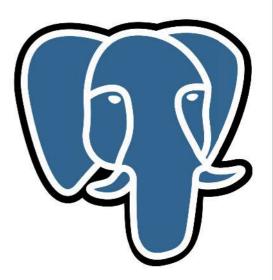
PostgreSQL

Nesta aula você aprenderá como trabalhar com PostgreSQL, criando tabelas, inserindo dados e fazendo consultas.

Anotações	

4LINUX Introdução ao PostgreSQL

- Constitui um Sistema Gerenciador de Banco de Dados (SGBD) de código aberto.
- Otimizado para aplicações complexas, que trabalham informações críticas ou grandes volumes de dados.
- ✓ Exemplo: e-commerce de médio porte.
- ✔ PostgreSQL é mais indicado, por ser capaz de lidar com o volume da dados gerado pelas consultas de venda.



O PostgreSQL é um poderoso sistema gerenciador de banco de dados objeto-relacional de código aberto e 100% livre. Considerado o banco de dados livre, mais avançado do mundo, por várias Razões.

Suporta largamente os padrões ANSI-SQL 92/99 e respeita a normativa ACID. É altamente extensível, detém vários tipos de índices para diversos tipos de aplicações. Utilizado em larga escala por empresas como Skype, Caixa Econômica Federal e Datasus. Python trabalha muito bem com o PostgreSQL, possui módulos de trabalho, assim como ocorre com MySQL.

Anotações		

Trabalhando com PostgreSQL

Quando não há usuário criado, o sistema o disponibiliza para gerenciar o banco.

Logue com o usuário root: 1

\$ su - root

Logue com usuário postgres e acesse o banco: 2

su - postgres

psql

Anotações

10

<u>41</u>	Trabalhando com PostgreSQL
1	Crie um usuário para acesso a aplicação: =# CREATE USER admin PASSWORD '41inux';
2	Crie um banco de dados para a aplicação: =# CREATE DATABASE projeto OWNER admin;
3	Saia do sistema com o usuário postgres: =# \q
4	Logue no banco denominado 'projeto' com usuário 'admin': =# psql -h localhost -U admin projeto

Anotações	

<u>41.</u>	Trabalhando com PostgreSQL
1	Verifique as tabelas criadas: =# \dt
2	Gere uma tabela para inserir scripts: =#CREATE TABLE scripts(id SERIAL, nome VARCHAR(50), conteudo TEXT);
3	Veja as informações sobre a tabela: =# \d scripts
4	Veja o conteúdo da tabela: =# SELECT * FFROM scripts;

Anotações	

notações		

4 L	Trabalhando com PostgreSQL
	Trabamanao com r ostgresq2
1	Para contar dados específicos, execute: =# SELECT * FROM scripts WHERE id=1;
2	Para consultar dados por um intervalo, execute o seguinte comando: =# SELECT * FROM scripts WHERE id BETWEEN 1 AND 4;
3	Consultas a dados por parte do campo, execute:
•	=# SELECT * FROM scripts WHERE nome LIKE '%.py';
4	Ordene dados de forma decrescente com o comando:
7	=# SELECT * FROM scripts WHERE nome LIKE '%.py%' ORDER BY id DESC;
∧ nota	

_
_
_
_
_
_



Python Fundamentals



MySQL

8.3

Anotações			



Objetivos da Aula

- ✓ Conhecer o MySQL.
- ✓ Trabalhar com MySQL.

MySQL

Nesta aula demonstraremos como trabalhar com MySQL criando tabelas, inserindo dados e fazendo consultas.

Allotações			
	-		

- Consiste um dos SGBD's mais populares.
- ✓ Amplamente empregado em serviços de hospedagem de sites.
- ✓ Focado em agilidade, efetivo se a aplicação demanda retornos rápidos, não envolvendo operações complexas.
- ✓ Exemplo: armazenamento de conteúdo de site/fórum.



Anotações	

<u>41</u>	Trabalhando com MySQL	18
1	Faça login no MySQL: \$ mysql -u root -p	
2	Verifique os bancos existentes: > SHOW DATABASES;	
3	Para criar um banco de dados, execute: > CREATE DATABASE projetos;	
4	Acesse a base: > USE projetos;	

Anotações			

<u>41.</u>	Trabalhando com MySQL
5	A fim de inserir usuário para administrar o banco, execute: > GRANT ALL PRIVILEGES on projetos.* to admin@'localhost' IDENTIFIED BY '4linux' WITH GRANT OPTION;
6	Atualize os acessos: > FLUSH PRIVILEGES;
7	Saia do MySQL: > EXIT;
6	Acesse com o novo usuário: # mysql -u admin -p

Anotações	

<u>41.</u>	Trabalhando com MySQL	20
1	Verifique as tabelas criadas: > SHOW TABLES;	
2	A fim de criar uma tabela para inserir clientes, execute: > CREATE TABLE clientes(id INTEGER PRIMARY KEY NOT NULL AUTO_INCREMENT, nome VARCHAR(255), endereco VARCHAR(255));	
3	Veja informações sobre a tabela: > DESCRIBE clientes;	
4	Para ver o conteúdo da tabela, execute: > SELECT * FROM clientes;	

Anotações		

Anotações		

Wesley Ângelo Ferreira / wesleyangeloferreira@gmail.com	4 <u>1</u> .
Ferreira / v	1
vesleyange	2
loferreira@	3
gmail.com	

Trabalhando com MySQL

- Criar uma nova tabela chamada projetos, execute o comando:

 > CREATE TABLE projetos(id INTEGER PRIMARY KEY AUTO_INCREMENT NOT NULL, nome VARCHAR(255) NOT NULL, cliente VARCHAR(255));
- Faça a inserção de um novo registro:

 > INSERT INTO projetos(nome, cliente) VALUES
 ('Virtualização', 'Mariana');
- Verificar apenas o cliente que contratou algum projeto

 > SELECT * FROM projetos WHERE cliente IN (SELECT nome FROM clientes);

Anotações		



Python Fundamentals



Estrutura de Dados no MongoDB

8.4

Anotações			

4LINUX Estrutura de Dados no MongoDB Objetivos da Aula

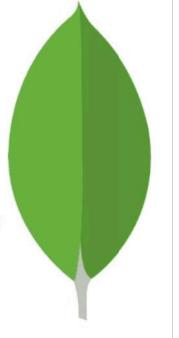
- ✓ Conhecer o MongoDB.
- ✓ Compreender sua estrutura de dados.

Estrutura de Dados no MongoDB

Nesta aula você conhecerá o MongoDB e a sua estrutura de dados.

Anotações	

- ✓ Trata-se de um banco de dados do tipo noSQL (Not Only SQL).
- ✓ Os dados são armazenados no formato JSON, visando a alta performance.
- ✔ Bancos de dados noSQL, normalmente ocupam mais espaço em disco, uma vez que não dispõem do recurso JOIN entre os dados, sua leitura e escrita porém, são bem superiores.



Estrutura de Dados no MongoDB

MongoDB não é um substituto dos bancos relacionais, embora seja possível trabalhar somente com ele, sem a necessidade de outros bancos de dados.

A escolha pelo MongoDB, se dá com a necessidade de velocidade em leitura e escrita de dados. Em seu site, encontramos benchmarks mostrando sua performance, efetuando cerca de 270 mil operações de leitura e escrita por segundo.

Também é muito utilizado para o controle de filas. Existem muitos artigos na internet, mostrando como é possível substituir o RabbitMQ pelo MongoDB.

MongoDB Performance Testing:

https://www.mongodb.com/blog/post/performance-testing-mongodb-30-part-1-throughput-improvements-measured-ycsb

4LINUX MongoDB

- ✓ Trabalha com documentos do tipo Chave Valor.
- ✓ Armazena documentos do tipo JSON, o formato mais usado atualmente para enviar dados na web através de API's que utilizam o protocolo REST.

Anotações		

4LINUX MongoDB

Sua estrutura de armazenamento de dados também é diferente do SQL, então temos:

Collections: armazena os documentos, semelhante as tabelas nos bancos SQL.

Documentos: guarda os dados no formato JSON, equivale aos registros do SQL.

Em MongoDB, um registro é um documento, este, consiste uma estrutura de dados composta por chave e valor (como um dicionário em Python). Uma chave pode ter qualquer valor (até outro dicionário ou umalista).

```
Exemplo:
```

```
{
    nome: "Mariana",
    cargo: Analista,
    grupos: ["consultoria", "treinamento"]
}
```

Para saber mais sobre collections, visite o site com a documentação oficial: https://docs.mongodb.com/v3.2/core/databases-and-collections/

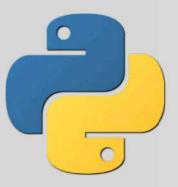
4LINUX Exemplo de Documento

Instale da seguinte maneira:

#fazer a instalação
\$apt-get install mongodb
#iniciar o serviço
\$ service mongodb start
#acessar o banco
\$ mongo



Python Fundamentals



Comandos no MongoDB

8.5

Anotações		

4LINUX Comandos no MongoDB

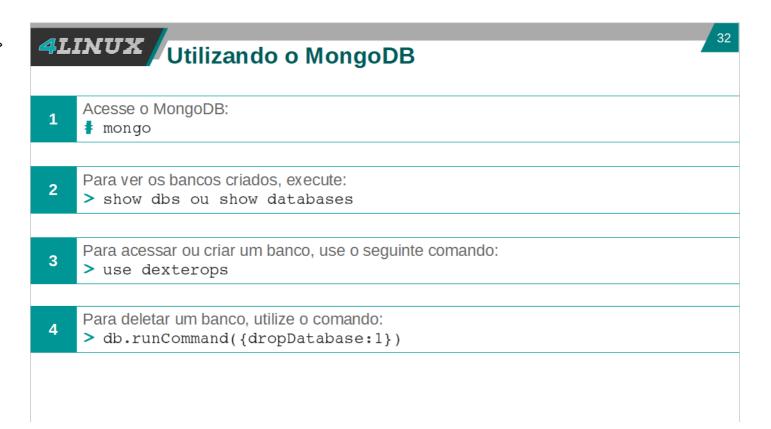
Objetivos da Aula

- ✓ Criar e excluir bases.
- ✓ Inserção, atualização, exclusão e consultano MongoDB.

Trabalhar com comandos no MongoDB

Nesta aula você aprenderá como trabalhar com o MongoDB, criando collections, inserindo dados e fazendo as consultas.

Anotações	



No console do MongoDB, encontramos comandos criados para a compatibilidade com usuários do MySQL. Exemplo:

```
show databases
show tables
```

O comando show tables foi criado especificamente para essa compatibilidade, já que esse banco de dados trabalha com documentos JSON ao invés de tabelas.

Recomenda-se cuidado ao digitar o nome do banco de dados, pois o MongoDB cria um novo banco de dados cada vez que for digitado um nome errado.

Sendo assim, é importante verificar os bancos existentes, sempre apagando as criações errôneas, evitando assim a existência do que chamamos de "sujeira" em banco de dados.

Além dos comandos mostrados, contamos com os seguintes:

db.help() — mostra comandos para administrar o MongoDB.

db.<nome-da-collection>.help() — exibe as opções de comandos existentes para você administrar a sua collection.

show users — exibe os usuários criados. Por padrão mongodo não vem com usuários criados.

Para criar um usuário digite o seguinte comando:

Na versão 2 do MongoDB o comando é: addUser().

4LINUX Utilizando o MongoDB

- Para verificar documentos de modo mais fácil quanto a leitura, execute: 1
 - > db.file.find().pretty()
- Pesquisas com filtro, são realizadas com o comando: 2
- > db.fila.find({" id":2}).pretty()
- Para fazer alterações no documento e apagar dados, execute: 3
 - > db.fila.update({"_id":1},{"status":1})
- Para fazer uma alteração no documento e permanecer com os dados, execute: 4
- > db.fila.update({"_id":1},{"\$set":{"status":1}})

O parâmetro id não é obrigatório, porém, caso não seja informado, MongoDB criará um id parecido com esse: { "_id" : ObjectId("569946c973fdfb8dacf877f3")

Esta ação, garante a unicidade dos documentos, possibilitando inserir o mesmo documento várias vezes sem informar o id. MongoDB acrescentará esses documentos, fazendo com que pelo menos um parâmetro assegure a diferença entre eles.

Ao utilizar o update, nunca esquecer de usar a chave \$set para atualizar um valor em específico. Caso a chave não seja informada, o documento que se pretende atualizar, será alterado para o parâmetro passado.

```
Exemplo:
```

```
db.file.update({" id":1},{"status":1})
```

O resultado será:

```
{" id":1,"status":1}
```

Com o \$set conforme o exemplo, o resultado será:

```
{" id":1,"servico":"intranet","status":1}
```

Também importante, cuidado ao utilizar o método remove, se não forem especificados parâmetros, serão removidos todos os documentos.

Criar Documento com Subdocumento

Anotações		

Alterando um Subdocumento

Anotações		

Removendo um Subdocumento

Anotações		