Marcio Michelluzzi

GitHub - @marciomichelluzzi

Introdução ao MySQL

- MySQL é um Sistema Gerenciador de Banco de Dados (SGBD) relacional.
- Utiliza a linguagem SQL (Structured Query Language, ou traduzindo, Linguagem de Consulta Estruturada).
- MySQL é multiusuário e multitarefas.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Introdução ao MySQL

- Inicialmente desenvolvido para trabalhar com projetos de pequeno e médio porte, com a capacidade de suportar por volta de cem milhões de registros em cada tabela, podendo chegar ao tamanho médio de aproximadamente cem megabytes por tabela.

- Nas versões mais recentes o MySQL ultrapassa (e muito) esses limites e capacidades das primeiras versões.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Introdução ao MySQL

- MySQL é conhecido por ser de fácil utilização, e usado por empresas que trabalham com grandes volumes de dados, tais como, Google, NASA, Bradesco, HP, Sony entre outras grandes empresas de renome.

- Possui uma interface extremamente simples e é compatível com grande parte dos sistemas operacionais.

- Podemos dizer que essas são duas das grandes características que fazem o MySQL ser tão utilizado atualmente e estar em constante crescimento.

Compatibilidade

- MySQL é compatível com a maioria dos sistemas operacionais existentes atualmente no mercado.

- Pelo fato de ser desenvolvido em C e C++, isso faz com que seja extremamente fácil sua acessibilidade entre os diversos sistemas operacionais.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Compatibilidade

- Vejamos alguns desses sistemas.
  - Windows: Compatível com todas as versões
  - Linux: Sendo compatível com as principais versões, como Fedora, Core, Debian, SuSE e RedHat.
  - Unix: Sendo compatível com as versões Solaris, HP-UX, AIX, SCO.
  - FreeBSD.
  - Mac OS X Server.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Licença

- O MySQL é de Código Aberto (Open-Source), desenvolvido e distribuído sob as licenças GNU/GLP (General Public Licence, ou traduzindo, Licença Publica Geral), a qual determina o que se pode ou não fazer à ferramenta e demais recursos.

- Além do programa, o seu código-fonte também é disponibilizado para que qualquer usuário possa edita-lo de forma que atenda suas necessidades.

Licença

- Os princípios básicos da licença GNU/GLP são:
  - Utilização: Permite que o usuário faça uso do software para qualquer finalidade.
  - Distribuição: Livre distribuição do software entre quaisquer pessoas.
  - Didática: Permite que seu funcionamento seja estudado através de seu código-fonte
  - Colaboração: Possibilita que seu código-fonte seja modificado para evoluir a ferramenta. Como regra seu novo código-fonte tem que permanecer sendo livre segundo essa licença.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Características

- Portabilidade

Devido o MySQL ter sido desenvolvido em C e C++, tornou-se extremamente fácil a portabilidade entre os diferentes sistemas, plataformas e compiladores. Possui também módulos de interface para múltiplas linguagens, tais como Delphi, Java, Python, PHP, ASP, Ruby e entre outras linguagens mais.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Características

- Formas de Armazenamento

O MySQL possibilita diversos tipos de tabela para o armazenamento dos dados, tendo em conta que cada tipo tem suas próprias características. Dessa maneira temos a possibilidade de escolhermos o tipo de acordo com cada situação diferente. Enquanto um tipo tem como prioridade a velocidade, outro da prioridade ao volume de dados, entre outras características.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Características

- Velocidade

Alta velocidade no acesso dos dados em razão de diversos motivos em seu desenvolvimento com tabelas ISAM, que foi substituído pelo novo sistema MyISAM na versão 5 do MySQL, além de utilização de caches em consultas, utilização de indexação, algoritmos de busca, entre outros recursos.

Características

Capacidade

O MySQL possui um alto poder de execução e de armazenamento. De acordo com a plataforma em que seja usado, suas tabelas poderão armazenar grandes volumes de dados, o limite ficará por conta somente do tamanho máximo de arquivos que a plataforma que estiver sendo utilizada puder manipular.

Características

Já no caso de tabelas do tipo InooDB, onde o armazenamento pode ser realizado em um ou vários arquivos separados, fica possível armazenar volumes de dados equivalentes a TB (Terabytes) de tamanho. E referente a expressões SQL, o MySQL suporta execuções de script SQL com até 61 milhões de tabelas “joins”.

O MySQL, por ser um banco de dados poderoso, tem a capacidade de realizar bilhões de consultas em um único dia em um site e também fazer o processamento de milhões de transações por minuto.

Características

- Como já sabemos, o MySQL trabalha com a linguagem SQL, sendo extremamente rápido. E isso foi possível devido a SQL ter sido implementada no MySQL através de códigos e funções altamente customizadas pelos seus desenvolvedores. Isso gerou a grande vantagem de velocidade no processamento dos códigos SQL.

- Porém, ao mesmo tempo trouxe um ponto negativo, sendo ele o fato de que com essa customização, nem todos os padrões das versões mais atuais do SQL tenham sido trazidos para o MySQL, porque poderiam prejudicar consideravelmente a velocidade do banco de dados. Entretanto, essa desvantagem influencia muito pouco na aplicação desenvolvida.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Conceito de Banco de Dados

- Uma base de dados, nada mais é do que estruturas complexas de dados. Estes dados são gravados em forma de registros em tabelas.

- Imagine um arquivo de fichas, numa empresa onde há várias caixas, cada uma contendo os dados dos funcionários de um certo setor.

- Cada caixa possui várias fichas, que são os cadastros dos funcionários – cada ficha contém os dados de apenas um funcionário.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Conceito de Banco de Dados

- Indo mais longe, podemos concluir que cada ficha contém diversas informação sobre o funcionário em questão.
- Portanto, cada caixa é uma tabela, contendo diversas fichas, que são os registros, e cada ficha possui várias informações sobre o funcionário, que são os campos.
- Como foi dito, há várias caixas, uma para cada departamento, a soma de todas as caixas forma a base de dados.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Conceito de Banco de Dados

- Observando tudo isto de fora, podemos formar o seguinte esquema:

- Cenário exemplo de fichas em caixas:
  - Base de dados > Tabela > Registro > Coluna

- Em banco de dados relacional:
  - Banco de dados > Tabela > Linha > Campo

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Conceito de Banco de Dados

- Banco de dados > Tabela > Linha > Campo
- A linha acima mostra os termos normalmente usados para o que acabamos de aprender.
- Os campos podem ser de diferentes tipos e tamanhos, permitindo ao programador criar tabelas que satisfaçam ao escopo do projeto.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Conceito de Banco de Dados

- A decisão de quais campos usar e quais não usar é muito importante, pois influi drasticamente na performance da base de dados que estamos desenvolvendo, portanto, é de bom grado um conhecimento sólido destes conceitos.

- A etapa de montagem das tabelas, é senão a mais importante, uma das mais importantes etapas da montagem de uma base de dados, pois um bom projeto pode facilitar muito o trabalho de programação.

Campos de Banco de Dados

- Como já sabemos, os campos são a parte fundamental de uma base de dados. É nos campos que as informações ficam armazenadas.

- Para um otimização da base de dados, antes e utilizar, devemos definir os campos que desejamos usar, e especificar o que cada um pode conter.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Campos de Banco de Dados

- O MySQL oferece campos bem comuns, que até mesmo um programador novato já deve ter visto.

- Alguns deles estão aqui listados:
  - CHAR
  - VARCHAR
  - INT
  - FLOAT
  - DATE
  - TEXT/BLOB

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Atividade 1

- O que é MySQL?
- Quais as vantagens de usar MySQL?
- O que é Banco de Dados?
- Qual o conceito de banco de dados?
- O que é uma tabela em banco de dados?
- O que é um registro em banco de dados?
- O que é um campo em banco de dados?
- O que são 'tipos de dados' em banco de dados?

TIPOS DE CAMPOS SQL

Tipos de campos de Banco de Dados

- CHAR (M)

- Os campos CHAR são usados para caracteres alfanuméricos, como endereços e nomes. Seu tamanho é fixo e instaurado ao ser criado. Um campo do tipo CHAR pode ter de 1 a 255 caracteres.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Tipos de campos de Banco de Dados

- Exemplo:
  - bairro CHAR(30);

- Define um campo chamado ’bairro', que pode conter até trinta letras. Observe que não há acentos no nome do campo, pois muitos servidores não acentuam, e sua tabela teria difícil acesso.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Tipos de campos de Banco de Dados

- VARCHAR(M)

- Sua estrutura e funcionamento é idêntico ao campo anterior, salvo que no tipo CHAR, o tamanho definido é fixo, e mesmo que não usado, aquele espaço em disco é alocado. Já o campo VARCHAR, aloca apenas o espaço necessário para gravação. Contudo, vale lembrar que trocamos espaço por velocidade, pois este campo é 50% mais lento que o anterior.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Tipos de campos de Banco de Dados

- Exemplo:
  - bairro VARCHAR(30);

- Define um campo chamado bairro que pode conter até trinta letras. Se você preencher apenas duas, o campo não ocupara todos os trinta bytes, mas apenas dois.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Tipos de campos de Banco de Dados

			MIN	MAX
CHAR	String de tamanho fixo. Sempre é completada com espaços a direita até o tamanho definido		1	255 caracteres
		OBS	Espaços excessivos são removidos quando o valor é trazido.Os valores são ordenados e comparados ignorando caixas altas e baixas de acordo com a codificação padrão, a menos que seja fornecido uma chave binária.	
VARCHAR	String de tamanho variável		1	255 caracteres
		OBS	Os valores são ordenados e comparados ignorando caixas altas e baixas de acordo com a codificação padrão, a menos que seja fornecido uma chave binária.Nota: Espaços execessivos são removidos quando o valor é inserido.	
TINYTEXT			0	255 (2^8 – 1) caracteres
TEXT			0	65535 (2^16 – 1) caracteres
MEDIUMTEXT			0	16777215 (2^24 – 1) caracteres
LONGTEXT			0	4294967295 (2^32 – 1) caracteres

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Tipos de campos de Banco de Dados

- INT(M) [Unsigned]

- O campo INT, que como o próprio número diz, suporta o conjunto dos números inteiros, originalmente numa variação de −2147483648 a 2147483647.

- O parâmetro Unsigned pode ser passado, excluindo os números negativos,
- proporcionando um intervalo de 0 até 4294967295.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Tipos de campos de Banco de Dados

- Exemplo 01:
  - temperatura INT;
- Campo com nome temperatura que pode ter valores positivos e negativos.
- Exemplo 02:
  - temperatura INT unsigned;
- Campo com nome temperatura que pode ter apenas valores positivos.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Tipos de campos de Banco de Dados

Type	Storage	Minimum Value	Maximum Value
	(Bytes)	(Signed/Unsigned)	Signed/Unsigned)
TINYINT	1	-128	127
		0	255
SMALLINT	2	-32768	32767
		0	65535
MEDIUMINT	3	-8388608	8388607
		0	16777215
INT	4	-2147483648	2147483647
		0	4294967295
BIGINT	8	-9223372036854775808	9223372036854775807
		0	18446744073709551615

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Tipos de campos de Banco de Dados

- FLOAT[(M,D)]
- Os pontos flutuantes (FLOAT) representam pequenos números decimais, e são usados para representar números com maior precisão.
- Exemplo:
  - voltagem_cadeira_eletrica FLOAT(4,2);
- Float válido: 1324.50

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Tipos de campos de Banco de Dados

- DATE

- Campo usado para armazenar informações referentes a data. A forma padrão , é 'YYYY-MM-DD', onde YYYY corresponde ao ano, MM ao mês, e DD ao dia. Ele pode variar de 1000-01-01 a 9999-12-31.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Tipos de campos de Banco de Dados

- Exemplo:
  - data_de_nascimento DATE;

- Data válida: 1999-12-25
- Data inválida: 1999-25-12

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Tipos de campos de Banco de Dados

- TIME

- Campo usado para armazenar informações referentes a hora. A forma padrão , é HH:MM:SS, onde HH é a hora, MM ao mês e SS os segundos. Ele pode variar de
- -838:59:59 até 839:59:59.

- Exemplo:
  - inicio_aula TIME;

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Tipos de campos de Banco de Dados

- DATETIME

- Campo usado para armazenar informações referentes a data e hora. A forma padrão , é YYYY-MM-DD HH:MM:SS, onde YYYY corresponde ao ano, MM ao mês, DD ao dia, HH a hora, MM ao minuto e SS os segundos. Ele pode variar de 1000–01–01 00:00:00 até 9999–12–31 23:59:59.

Marcio Michelluzzi – marciomichelluzzi@gmail.com

Tipos de campos de Banco de Dados

- Exemplo:
  - ocorrencia DATETIME;

- Data válida: 2016-02-26 12:15:00

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Tipos de campos de Banco de Dados

- TEXT/BLOB

- Os campos texto e blob são usados para guardar grandes quantidades de caracteres. Podendo conter de 0 a 65535 bytes, os blobs e texts são úteis para armazenar documentos completos, como este que você está lendo.

- A única diferença entre os campos BLOB e TEXT está no fato de um campo TEXT não ser sensível a letras maiúsculas e minúscula quando uma comparação é realizada, e os BLOBs sim.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Tipos de campos de Banco de Dados

- Exemplo:
  - relatorio BLOB;
- BLOB válido: 'Minha terra tem palmeiras onde canta o...'

- relatorio TEXT;
- TEXT válido: 'A que saudades que eu sinto...'

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Tipos de campos de Banco de Dados

- BIT OU BOOLEAN

- Um número inteiro que pode ser 0 ou 1
- Usado para representar false/ true

- Exemplo:
  - ativo bool;
- Campo para dizer se o usuário esta ativo

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Tabelas de Banco de Dados

- Um conjunto de registros, forma uma tabela. As tabelas portanto, armazenam grande quantidade de dados.

- Como no exemplo anterior, poderíamos ter centenas de nomes diferentes cadastrados em nossa tabela de pessoas. Cada conjunto de dados corresponde a um registro.

Atividade 2

- Imagine uma ficha de cadastro de uma pessoa. Para cada dado abaixo dê um nome para o campo, inserindo o tamanho do mesmo quando necessário.

- Exemplo:
- Nome completo - nome_completo VARCHAR(100)

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Atividade 2

- Casos para fazer:

- Código de cadastro
- Nome completo
- Idade
- Data de nascimento
- CPF
- CNPJ

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Atividade 2

- Rua
- Número
- Complemento
- CEP
- Cidade
- Estado
- Código de cadastro
- Data de cadastro
- Última atualização do cadastro
- Sexo
- Religião
- Cor
- QI
- Tipo sanguíneo
- Se é doador de órgãos
- Se tem passagem pela polícia
- Qual seu grau de ensino
- Se o usuário esta ou não ativo

Marcio Michelluzzi - marciomichelluzzi@gmail.com

OPERADORES

Chave primária

- Usado para que não seja permitido o cadastro de dois registros com chaves primárias iguais.

- Isto é claramente útil, quando não é desejado que seja digitado um segundo registro igual ao primeiro por engano.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Chave primária

- Para se definir uma chave primária, basta adicionar 'PRIMARY KEY' a definição do campo que se deseja a não duplicidade.

- Exemplo de uso para chave primária:
  - Um cadastro de pessoas com entrada VIP em uma festa, cadastradas com base no seu CPF. Não podem haver dois nomes com o mesmo CPF.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

AUTO INCREMENT

- Este recurso, faz com que conforme novos registros são criados, automaticamente estes obtém valores que correspondem ao valor deste mesmo campo no registro anterior, somado a 1.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

AUTO INCREMENT

- Exemplo:
  - codigo INT AUTO_INCREMENT;
- Automaticamente soma um a cada registro neste campo. Começando de 1, com inserção subsequente.
- Exemplo de uso para auto incremento:
  - O número de um pedido, ou até mesmo o número de uma nota fiscal, que sempre vai ser o número anterior + 1;

Marcio Michelluzzi - marciomichelluzzi@gmail.com

UNSIGNED

- Parâmetro Unsigned pode ser passado, excluindo os números negativos, proporcionando um intervalo de 0 até o tamanho máximo multiplicado por dois.

- Exemplo:
  - idade TINYINT
- Faz a faixa de números ir de -128 até 127

- idade TINYINT unsigned
- Faz a faixa de números ir de 0 até 255

Marcio Michelluzzi - marciomichelluzzi@gmail.com

UNIQUE

- Parâmetro Unique pode ser passado para que um campo seja único dentro de uma tabela.

- Exemplo:
  - idade CPF
- Permite o cadastro de dois ou mais CPFs iguais
  - idade CPF UNIQUE
- Não permite o cadastro duplicado.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

NULL e NOT NULL

- O parâmetro NULL indica se o campo pode ou não estar vazio.
- Quando dizemos que um campo pode ser NULL, significa que aquela informação, de certa forma, é opcional.
- Ao indicar um campo como NOT NULL nós estamos dizendo que o preenchimento daquele campo é obrigatório.

MySQL

NULL e NOT NULL

- Exemplo:
  - cpf CHAR(11) NOT NULL
- Faz com que o campo CPF seja obrigatório.
  - celular CHAR(11) NULL
- Faz com que o campo CELULAR seja de preenchimento opcional

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Atividade 3

- Complemente a atividade anterior da seguinte forma:
- Verifique quais campos criados necessitam ser do tipo chave primária. Ou seja, quais campos não podem repetir registro dentro do banco de dados. Nestes campos, adicione o PRIMARY KEY (PK)
- Analise e coloque NULL para os campos opcionais e NOT NULL para os campos obrigatórios

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Atividade 3

- Verifique também quais campos criados necessitam ser do tipo auto incremento. Ou seja, quais campos precisam ter o valor automaticamente inserido e incrementado. Nestes campos, adicione o AUTO INCREMENT (AI) e também veja quais devem ser Unsigned. Ou seja, campos que não precisam ter números negativos. Nestes campos, adicione o UNSIGNED (U).

SQL

- Para todas as operações que precisarmos fazer nós vamos executar um comando SQL.
- Existem comandos dos mais variados tipos: Para criar bancos, tabelas, campos, exibir informações, retornar consulta, atualizar, excluir, etc.
- Temos que ter em mente que vamos utilizar um gerenciador de banco de dados chamado phpMyAdmin, que vai agilizar alguns destes comandos para nós.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

SQL

- Voltamos pro USBWEBSERVER

- phpMyAdmin: http://localhost:8080/phpmyadmin
- Usuário: root
- Senha: usbw

Marcio Michelluzzi - marciomichelluzzi@gmail.com

DDL - Data Definition Language

CREATE DATABASE

- O comando CREATE nos permite criar um novo banco de dados.
- Para criar um novo banco de dados a sintaxe é:
  - CREATE DATABASE nome_do_banco_de_dados
- Exemplo:
  - CREATE DATABASE revenda

Marcio Michelluzzi - marciomichelluzzi@gmail.com

USE

- O comando USE seleciona o banco de dados que vamos usar. Depois de criar um banco de dados novo é preciso selecionar o banco para podermos criar as tabelas e fazer uso do mesmo.

- Para selecionar um banco de dados a sintaxe é:
  - USE nome_do_banco_de_dados

- Exemplo:
  - USE revenda

Marcio Michelluzzi - marciomichelluzzi@gmail.com

CREATE TABLE

- O comando CREATE nos permite criar tabelas em um banco de dados selecionado. Nós podemos criar uma tabela vazia, ou criar a mesma já informando quais campos ela deve possuir.

- Para criar uma tabela vazia a sintaxe é:
  - CREATE TABLE nome_da_tabela

- Exemplo:
  - CREATE TABLE fornecedor

Marcio Michelluzzi - marciomichelluzzi@gmail.com

CREATE TABLE

- O comando CREATE nos permite criar tabelas em um banco de dados selecionado. Nesta continuação vamos ver como criar uma tabela com os seus devidos campos.

- Para criar uma tabela e seus campos a sintaxe é:

```
CREATE TABLE nome_da_tabela (
    `campo1` tipo(tamanho) opcionais,
    `campo2` tipo(tamanho) opcionais
)
```


Marcio Michelluzzi - marciomichelluzzi@gmail.com

CREATE TABLE

```
CREATE TABLE ‘testes’ (
        ‘campo1’ tinyint(2) UNSIGNED NOT NULL AUTO_INCREMENT,
        ‘campo2’ smallint(5) UNSIGNED DEFAULT NULL,
        ‘campo3’ mediumint(9) UNSIGNED DEFAULT NULL,
        ‘campo4’ int(15) UNSIGNED DEFAULT NULL,
        ‘campo5’ bigint(25) UNSIGNED DEFAULT NULL,
        ‘campo6’ varchar(255) DEFAULT NULL,
        ‘campo7’ char(10) DEFAULT NULL,
        ‘campo8’ text,
        ‘campo9’ bit(1) NOT NULL DEFAULT '1', -- o campo9 será NOT NULL e terá o valor 1 como padrão
        ‘campo10’ float(4,2) NOT NULL,
        PRIMARY KEY (‘campo1’), -- nesta linha estou dizendo para que o campo1 seja do tipo PRIMARY KEY
        UNIQUE KEY ‘campo7’ (‘campo7’) -- nesta linha estou dizendo para que o campo7 seja do tipo UNIQUE
)
```


Marcio Michelluzzi - marciomichelluzzi@gmail.com

ALTER TABLE

- - O comando ALTER nos permite editar um banco de dados ou tabela existente.
- Exemplos:
- Inserindo coluna na tabela:
  - ALTER TABLE endereco ADD pais varchar(25);
- Remove chave primaria da tabela:
  - ALTER TABLE endereco DROP primary key;

Marcio Michelluzzi - marciomichelluzzi@gmail.com

ALTER TABLE

- Inserindo chave primaria na tabela
  - ALTER TABLE endereco ADD PRIMARY KEY(id_endereco);
- Modificar definições de uma coluna
  - ALTER TABLE endereco MODIFY bairro varchar(50);
- Excluir coluna da tabela:
  - ALTER TABLE endereco DROP cidade;
- Renomear tabela:
  - ALTER TABLE endereco RENAME localizacao;

Marcio Michelluzzi - marciomichelluzzi@gmail.com

SHOW TABLES

- O comando SHOW mostra todas as tabelas de um banco de dados selecionado.
- A sintaxe para exibir todas as tabelas de um banco é:
- SHOW TABLES

Marcio Michelluzzi - marciomichelluzzi@gmail.com

DESC

- O comando DESC exibe os campos de uma tabela.
- Para ver os campos a sintaxe é:

- DESC nome_da_tabela

- Exemplo:
  - DESC fornecedor

Marcio Michelluzzi - marciomichelluzzi@gmail.com

DROP DATABASE

- O comando DROP nos permite excluir um banco de dados.
- Para excluir um novo banco de dados a sintaxe é:

- DROP DATABASE nome_do_banco_de_dados

- Exemplo:
  - DROP DATABASE revenda

Marcio Michelluzzi - marciomichelluzzi@gmail.com

DROP TABLE

- - O comando DROP nos permite excluir tabelas em um banco de dados selecionado.
- Para excluir uma tabela a sintaxe é:

- DROP TABLE nome_da_tabela

- Exemplo:
  - DROP TABLE fornecedor

Marcio Michelluzzi - marciomichelluzzi@gmail.com

TRUNCATE TABLE

- O comando TRUNCATE limpa uma tabela. Ou seja, exclui todos os dados dela.
- Para limpar uma tabela usamos a sintaxe:

- TRUNCATE TABLE tabela

- Exemplo:
  - TRUNCATE TABLE cliente

Marcio Michelluzzi - marciomichelluzzi@gmail.com

DML - Data Manipulation Language

Comando SELECT

- O comando SELECT serve pra exibir dados de uma tabela.
- Por exemplo, preciso consultar todas as pessoas de SC que estão cadastradas na tabela cliente. Farei o comando SQL abaixo:

- `SELECT * FROM cliente WHERE uf = 'SC';`

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Comando SELECT

- O * significa que deve-se retornar todos os campos da tabela.
- cliente é o nome da tabela
- uf = 'SC' é o nosso argumento.
- Argumento = condição.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Comando SELECT

- Eu poderia também retornar apenas alguns campos, ao invés de selecionar todos.
- Ficaria:
  - SELECT nome, marca, veiculo FROM cliente WHERE uf = 'SC';
- No lugar do * eu disse que os campos de nome, marca e veiculo devem retornar da tabela.

Operadores Condicionais

- = IGUAL
- > MAIOR QUE
- < MENOR QUE
- <= MENOR IGUAL
- >= MAIOR IGUAL A
- != ou < > DIFERENTE
- AND operador E
- OR operador OU

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Operadores Condicionais

- Exemplo:
  - SELECT nome, marca, veiculo FROM cliente WHERE idade >= 18;

- Pedindo retorno dos dados onde a pessoa tem 18 anos ou mais.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Atividade 4

- Abra seu usbwebserver. Abra o phpMyAdmin. Crie um banco chamado lavacao
- Importe o arquivo de banco de dados enviado por e-mail no seu phpMyAdmin
- Crie um arquivo .txt (no bloco de notas) e coloque a consulta SQL e o resultado da consulta para cada item abaixo.

- Exemplo: Consulte todos os clientes que tem carro Civic
  - SQL: SELECT * FROM cliente WHERE veiculo = 'Civic'
  - Resultado: 30, Marcelo Oliveira...

Atividade 4

- Responda os itens abaixo:

- Consulte todos os dados clientes que tem carro da marca Volkswagen
- Consulte todos os nomes clientes que tem carro da marca chevrolet
- Consulte nomes e celular de clientes que tem veiculo modelo fusca
- Consulte o nome e o celular do cliente com celular 48908989899

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Atividade 4

- Consulte o modelo do veiculo do carro do cliente Derpina Derp
- Consulte o telefone do cliente Otavio Otto
- Consulte os veículos da marca citroen ou Renault
- Consulte as cidades que há clientes na região sul do Brasil

Pesquisando com LIKE

- A função LIKE, do MySQL, faz uma busca sofisticada por uma substring dentro de uma string informada. Traduzindo, ele faz uma pesquisa.

- % - Busca zero ou mais caracteres

- SELECT * FROM tabela WHERE campo LIKE critério;

Pesquisando com LIKE

- Exemplo 1: Quero pesquisar os nomes que começam com A:
  - SELECT * FROM cliente WHERE nome LIKE 'A%';
- Exemplo 2: Quero pesquisar os modelos de carros que terminam com i
  - SELECT * FROM cliente WHERE modelo LIKE '%i';
- Exemplo 3: Quero pesquisar todos os telefones que tenham 30 em algum lugar do número
  - SELECT * FROM cliente WHERE telefone LIKE '%30%';

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Atividade 5

- Execute comandos de consulta usando o LIKE:

- Todos os celulares que começam com ‘47’;
- Todos os modelos de veículos que pertencem a alguém chamado ‘Jose’;
- Todos os modelos de veículos que clientes que tem e-mail do vwcharmoso;
- Todos os nomes de pessoas que tem ‘de’ como separação do nome;
- Todos que possuem e-mail @vwcharmoso e que moram em SC;
- 6 – Todos que possuem e-mail argentino (.com.ar);
- 7 – Todos os celulares que tem 4788 como prefixo;

Pesquisando com LIKE

- A função LIKE, do MySQL, faz uma busca sofisticada por uma substring dentro de uma string informada na posição informada. Traduzindo, ele faz uma pesquisa na posição da string que queremos.

- _ - Busca somente um caractere.

- SELECT tabela WHERE campo LIKE critério;

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Pesquisando com LIKE

- Exemplo 1: Quero pesquisar as marcas de carro onde a segunda letra é O:
  - SELECT * FROM cliente WHERE marca LIKE '_o%';
- Exemplo 2: Quero pesquisar os modelos de carros que a terceira letra é 0:
  - SELECT * FROM cliente WHERE modelo LIKE '__0%';

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Atividade 6

- Execute comandos de consulta usando o LIKE:

1. Nomes e telefones dos registros onde o telefone tem os quatro primeiros dígitos (tirando o DDD) iniciando com 8884.
2. Nome, modelo e marca de todos os registros onde a cidade tem a segunda letra com A.
3. Nome e celular de todos os registros que tem o modelo terminando em a
4. Todos os nomes e celulares de registros onde o celular tem o numero 8 na segunda e na penúltima posição (desconsiderando o DDD)
5. Selecionar todos os nomes dos registros que tenham marca igual a honda ou volkswagen, e que o celular inicie em 90 (tirando o DDD).
6. Selecionar todos os nomes dos registros que tenham a cadastro inativo, que o modelo do carro comece com ‘c’ e que o penúltimo caractere do nome do cliente seja ‘d’.
7. Selecionar todos os celulares de registros que o primeiro caractere for 4 e o terceiro for 8, e ainda que estejam com o cadastro ativo.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Funções e Operadores

Função DISTINCT

- A função DISTINCT é utilizada para não mostrar valores repetidos.

- Se pesquisarmos:
  - SELECT marca FROM cliente;
- O resultado seria: Volkswagen, Volkswagen , Fiat, Volkswagen, Fiat... e muito mais.

- Porém, eu não quero que apareça a Volkswagen três vezes. Quero apenas saber quais marcas estão na tabela

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Função DISTINCT

- Podemos resolver isto com a função DISTINCT:
  - SELECT DISTINCT(marca) FROM cliente;
- O resultado será: Volkswagen, Fiat, Chevrolet, Seat, Renault, Citroen, Toyota, Honda, Ferrari

- Perceba que neste exemplo eu não estou usando qualquer critério na cláusula WHERE, porém eu poderia usar em qualquer momento. Exemplo:
  - SELECT DISTINCT(marca) FROM cliente WHERE ativo = 0

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Função COUNT

- A função COUNT é utilizada para saber quantos registros existem na consulta.

- Exemplo:
- Quero saber quantos registros a tabela cliente tem:
  - SELECT COUNT(*) FROM cliente;
- O resultado será: 31

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Função COUNT

- Note que neste exemplo eu também não estou usando qualquer critério na cláusula WHERE, porém eu poderia usar em qualquer momento.

- Exemplo:
  - SELECT COUNT(*) FROM cliente WHERE celular LIKE '47%';
- O resultado será: 10

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Atividade 7

- Fazer as consultas SQL:

1. Listagem de veículos diferentes que a tabela cliente tem;
2. Listagem de marcas diferentes que a tabela cliente tem;
3. Quantos carros estão com o cadastro ativo.
4. Quantos carros estão com o cadastro inativo.
5. Total de carros de marca japonesa (Honda e Toyota) tem na tabela.
6. Quantos carros ativos a marca Hyundai tem na tabela.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Função BETWEEN

- A função BETWEEN é utilizada para consultar valores entre um intervalo de dados previamente definido. Sempre será pedido entre um valor inicial e um valor final.
- Por exemplo, todos os registros de pessoas que tenham entre 18 e 40 anos.
- Ou então, todos os registros que tenham carro fabricado entre 2000 e 2008.

- A sintaxe ficaria da seguinte forma:
  - SELECT * FROM cliente WHERE ano BETWEEN 2000 AND 2008;

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Operador ORDER BY

- O operador lógico ORDER BY, ou ORDENAR POR, simplesmente lista os registros, colocando-os em ordem de acordo com o campo solicitado.

- Exemplo:
  - SELECT * FROM cliente WHERE estado = “sc” ORDER BY nome;

- Este select retornará todos os dados da tabela cliente ordenando os resultados pelo nome.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Operador ORDER BY

- A ordenação pode ser em ordem ascendente (ASC) ou descendente (DESC)
- Exemplo:
  - SELECT * FROM cliente WHERE estado = “sc” ORDER BY nome DESC;
- Este select retornará todos os dados da tabela cliente ordenando os resultados pelo nome em ordem descendente.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Operador LIMIT

- O LIMIT fará o papel de exibir uma parte dos resultados.
- Ele indica qual trecho da consulta será retornado.
- Suponha que você quer retornar os 10 primeiros resultados de uma consulta. Você colocará LIMIT 0,10. Ou seja, exibir a partir do registro zero, e exibir 10 resultados.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Operador LIMIT

- Ou então, somente do registro 5 ao 10. Ficaria: LIMIT 5,10
- A sintaxe é Sempre é LIMIT inicio quantidade

- Exemplo:
  - SELECT * FROM cliente WHERE marca = 'ford' LIMIT 0,5;
- Este select retornará os dados da tabela cliente a partir do primeiro resultado, exibindo 5 registros.

- SELECT * FROM cliente WHERE marca = 'ford' ORDER BY ano ASC LIMIT 0,5;

Marcio Michelluzzi - marciomichelluzzi@gmail.com

MAX, MIN, AVG

- A função MAX(campo) retorna o maior valor encontrado para aquele campo na tabela.

- Exemplo:
  - SELECT max(ano) FROM cliente;

Marcio Michelluzzi - marciomichelluzzi@gmail.com

MAX, MIN, AVG

- A função MIN(campo) retorna o menor valor encontrado para aquele campo na tabela.

- Exemplo:
  - SELECT min(ano) FROM cliente;

Marcio Michelluzzi - marciomichelluzzi@gmail.com

MAX, MIN, AVG

- A função AVG(campo) retorna a media dos valores encontrados para aquele campo na tabela.

- Exemplo:
  - SELECT avg(ano) FROM cliente;

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Atividade 8

- Fazer as consultas SQL.
- Apresentar o SQL e o resultado.

1. Qual a menor idade do cadastro de clientes?
2. Qual a maior idade do cadastro de clientes?
3. Qual a media de idade do cadastro de clientes?
4. Quais são as três primeiras marcas de carro em ordem ascendente?
5. Quais são os 10 primeiros clientes da lavação?

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Atividade 8

6. Quantos clientes tem carro com 5 anos ou menos de uso?
7. Quantos clientes tem carro da década de 70?
8. Quantos clientes tem idade entre 23 e 29 anos?
9. Qual a media de ano dos carros da honda?
10. Qual a media de ano dos carros inativos?
11. Qual a media de idade das pessoas que o nome inicia com as 10 primeiras letras do alfabeto.
12. Quais as marcas distintas dos carros onde a idade do motorista é maior que 40 anos.

Comando UPDATE

- UPDATE é útil quando você quer alterar registros de uma tabela. Você pode alterar um ou mais campos ao mesmo tempo.

- UPDATE não gera um conjunto de resultados. Se você quiser saber quais resultados serão alterados, examine primeiro os resultados da consulta que use os mesmos critérios e então execute a atualização.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Comando UPDATE

- Exemplo:
  - UPDATE cliente SET nome = 'Anderson Novo' WHERE id = ‘1';

- Estou pedindo para que todos os registros da tabela cliente que possuam o campo id preenchido com 1 tenham o campo nome atualizado para ‘Anderson Novo’.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

SQL – Comando UPDATE

- Sintaxe:

```
UPDATE tabela
SET campo1 = valornovo, ...
WHERE critério;
```

- Onde:
  - Tabela - O nome da tabela cujos os dados você quer modificar.
  - Valornovo - Uma expressão que determina o valor a ser inserido em um campo específico nos registros atualizados.
  - Critério - Uma expressão que determina quais registros devem ser atualizados. Só os registros que satisfazem a expressão são atualizados.

Marcio Michelluzzi – marciomichelluzzi@gmail.com

Atividade 9

- Execute comandos de atualização, colocando o SQL e o resultado conforme cada item abaixo.

1. Parece que Mario Armario ganhou na mega sena. Atualize a marca do carro dela para Ferrari e o veiculo para F50.
2. O coração do Hermanoteu Godá petrificou, e ele perdeu seu carro para a Receita Federal. Faça com que seu cadastro mude de ativo para inativo.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

Atividade 9

3. A Micalateia Santa perdeu uma causa na justiça e não conseguiu pagar seu licenciamento. Ela trocou seu carro por um Chevrolet Chevette. Atualize no banco de dados.

4. Você recebeu um e-mail de feiao@chines.com.br reclamando que seu carro foi riscado durante a lavação. Pesquise o telefone e nome do cliente para poder entrar em contato.

5. Quem diria que Molusco Um trocaria sua terrinha por Laguna, SC. Atualize sua cidade e estado.

COMANDO DELETE

- - O comando DELETE nos permite excluir um ou mais registros de uma tabela. A exclusão se aplica a todos os registros onde a nossa cláusula WHERE for verdade.

- Para excluir um dado usamos a sintaxe:
  - DELETE FROM nome_da_tabela WHERE condição

- Exemplo:
  - DELETE FROM cliente WHERE ativo = 0
- Obs: Estou pedindo para que a consulta exclua todos os cadastros de clientes onde o estado dele é inativo.

Marcio Michelluzzi - marciomichelluzzi@gmail.com

COMANDO INSERT

- - O comando INSERT nos permite cadastrar dados em uma tabela existente.

- Para cadastrar um dado usamos a sintaxe:
  - INSERT INTO nome_da_tabela (coluna1, coluna2, ...) VALUES (valor1, valor2, ...)

- Exemplo:
  - INSERT INTO cliente (nome) VALUES ('Anderson')

- Obs: Note que ao invés de eu informar um valor para o ID eu coloquei NULL. O banco de dados vai adicionar o valor deste registro automaticamente, pois cadastramos este campo como AUTO INCREMENTO

Marcio Michelluzzi - marciomichelluzzi@gmail.com

ATIVIDADE 10

1. O dono da nossa lavação andou fazendo trabalho mal feito e perdeu todos os clientes de São Paulo e Rio de Janeiro. Faça um comando SQL que delete os registros destes estados.

2. No final das contas ele é um bom vendedor, tagarela, e um tanto quanto malvado. Com essas qualidades ele conseguiu clientes em outros estados. Faça um comando SQL para cada novo cliente a ser cadastrado, conforme abaixo.

3. Terezinha Ranzinha, Belo Horizonte, MG, Ford KA ano 2010, sem e-mail, sem celular, sexo Feminino, 89 anos.

4. Astrogildo Estelar Souza, Salvador, BA, Hyundai Veloster ano 2015, rapidinho@uol.com.br, sem celular, sexo Masculino, 22 anos.