Aula 4 - Usando Operadores para categorizar dados

Declaração CREATE TABLE – criando tabelas no MySQL

Para criar tabelas em um banco de dados, usamos a declaração **CREATE TABLE**. Veja a sintaxe abaixo:

```
CREATE TABLE [IF NOT EXISTS] nome_tabela
 (
      coluna tipo_dados constraints
      coluna tipo_dados constraints
      coluna tipo_dados constraints
      ...
);
```

Note que precisamos especificar, além do nome da tabela, os nomes das colunas que a comporão e também seus respectivos tipos de dados, além das eventuais *constraints*.

Tipos de Dados no MySQL

A tabela abaixo resume os tipos de dados mais comuns no MySQL, que podem ser usados na criação de tabelas, para estabelecer o tipo de cada coluna:

Tipo	Descrição
INT	Inteiros entre -2,147,483,648 e 2,147,483,647
TINYINT	Números inteiros de -128 a 127
SMALLINT	Números inteiros de -32768 a 32767
MEDIUMINT	Números inteiros de -8388608 a 8388607
BIGINT	Números entre -9,223,372,036,854,775,808 e 9,223,372,036,854,775,807
DECIMAL(M,D)	Ponto decimal com M dígitos no total (precisão) e D casas decimais (escala); o padrão é 10,0; M vai até 65 e D até 30.
FLOAT(M,D)	Ponto flutuante com precisão M e escala D; o padrão é 10,2; D vai até 24.
CHAR(M)	String que ocupa tamanho fixo entre 0 e 255 caracteres
BOOL / BOOLEAN	Valores binários 0 / 1; Na verdade, é um alias para o tipo TINYINT(1)
VARCHAR(M)	String de tamanho variável, com até 65535 M caracteres.
BLOB / MEDIUMBLOB/ TINYBLOB	Campo com tamanho máximo de 65535 caracteres binários; 'Binary Large Objects', são usados para armazenar grandes quantidades de dados, como imagens.
MEDIUMTEXT	Permite armazenar até 16.777.215 caracteres.
LONGTEXT	Permite armazenar até 4.294.967.295 caracteres.
DATE	Uma data de 01/01/1000 a 31/12/9999, no formato YYYY-MM-DD
DATETIME	Uma combinação de data e hora de 01/01/1000 00:00:00 a 31/12/9999 23:59:59, no formato YYYY-MM-DD HH:MM:SS
TIME	Hora apenas, no formato HH:MM:SS
YEAR(M)	Ano nos formatos de 2 ou 4 dígitos; Se forem 2 (YEAR(2)), ano vai de 1970 a 2069; para 4 (YEAR(4)), vai de 1901 a 2155. O padrão é 4.

OPERADORES

Os operadores são usados junto com o comando SELECT para especificar critérios mais amplos para os dados que são retornados por uma consulta. Existem muitos operadores disponíveis para a SQL que suportam as exigências de consultas de dados.

O que é um operador

Operador é uma palavra ou um caractere reservado que é usado principalmente em uma cláusula WHERE de uma instrução SQL para realizar uma ou mais operações, como comparações e cálculos aritméticos. Os operadores são usados para especificar condições em uma instrução SQL e para servir de conjunções para várias condições em uma instrução.

Os operadores em questão são do tipo:

a) **Operadores de comparação;**
b) **Operadores lógicos**
c) **Operadores usados para negar condições**
d) **Operadores aritméticos**

a) Operadores de comparação

Os operadores de comparação são usados para testar valores individuais em uma instrução SQL. Veja a seguir os operadores de comparação.

- ✔ **> (Maior que).**
- ✔ **< (Menor que).**
- ✔ **= (igual).**
- ✔ **<= (menor quer ou igual a).**
- ✔ **>= (maior que ou igual a).**
- ✔ **<> (diferente).**
- ✔ **!< (não menor que).**
- ✔ **!> (não maior que).**

b)Operadores Lógicos

Os operadores lógicos são aqueles que usam palavras-chaves SQL ao invés de símbolos, para estabelecer comparações. Podemos formular a consulta invertendo o resultado, para isto, adicionamos a palavra **NOT** antes do operador, o que negará a condição proposta.

Os operadores lógicos que veremos são:

b.1) IS NULL ou IS NOT NULL
b.2) BETWEEN ou NOT BETWEEN
b.3) IN ou NOT IN
b.4) LIKE ou NOT LIKE
b.5) EXISTS ou NOT EXISTS

Para os seguintes exemplos considere db_empresa01

```
CREATE TABLE IF NOT EXISTS tb_funcionario (
  cd_funcionario int(11) NOT NULL AUTO_INCREMENT,
  nm_funcionario varchar(45) NOT NULL,
  cpf_funcionario varchar(11) NOT NULL,
  nro_depto varchar(45) NOT NULL,
  tel_funcionario varchar(10) DEFAULT NULL,
```

```
  salario float NOT NULL,
  cd_supervisor int(11) DEFAULT NULL,
  PRIMARY KEY (cd_funcionario));
```

Inserindo valores na tabela (Popular as tabelas) Via script

```
INSERT INTO `tb_funcionario` VALUES
   (1111,'Carlos','1122334455','3','1333334455',800.5,4444),
  (1112,'Ana','3333444555','2','1332325544',950,4444),
  (1113,'Eduardo','4446677888','1','1331317744',1500,NULL),
  (1114,'Andre','5553334448','2','1334343778',1200,4444),
  (1115, 'Vagner', '12343212', '4',NULL, '1150.00', ' 2222 '),
(1116, 'Gustavo', '85678768', '4',NULL, '897.00', ' 2222 ');
```

A nova tabela ficaria assim

cd_funcionario	nm_funcionario	cpf_funcionario	nro_depto	tel_funcionario	salario	cd_supervisor
1111	Carlos	1122334455	3	1333334455	800.5	4444
1112	Ana	3333444555	2	1332325544	950	4444
1113	Eduardo	4446677888	1	1331317744	1500	NULL
1114	Andre	5553334448	2	1334343778	1200	4444
1115	Vagner	12343212	4	NULL	1150	2222
1116	Gustavo	85678768	4	NULL	897	2222
NULL	NULL	NULL	NULL	NULL	NULL	NULL

b.1) Operador IS NULL ou IS NOT NULL

Estes operadores são utilizados para testar se um valor é nulo ou não, isto é, em uma consulta, podemos querer selecionar valores nulos ou valores que não sejam nulos.
Como exemplo, podemos formular uma consulta onde queremos saber quais os funcionários que não possuem número de telefone cadastrado.

Veja a consulta(consulta16):

```
SELECT nm_funcionario AS 'Funcionario sem telefone'
        FROM tb_funcionario AS F
        WHERE tel_funcionario IS NULL;
```

Funcionario sem telefone
Vagner
Gustavo

Tabela 17: Após a consulta16

b.2) Operador BETWEEN ou NOT BETWEEN

O operado BETWEEN é usado para procurar valores que estejam dentro de um conjunto de valores, especificados os valores mínimos e máximos, que são incluídos como parte do conjunto condicional. Veja uma consulta(consulta17) de exemplo:

```
SELECT nm_funcionario, salario
        FROM tb_funcionario
        WHERE salario BETWEEN 1000.00 AND 1500.00;
```

E o resultado:

	nm_funcionario	salario
▶	Eduardo	1500
	Andre	1200
	Vagner	1150

Tabela 18: Após a consulta17

Agora a mesma consulta(consulta18), mas usando o **NOT BETWEEN**

```
SELECT nm_funcionario, salario
        FROM tb_funcionario
        WHERE salario NOT BETWEEN 1000.00 AND 1500.00;
```

E o resultado:

	nm_funcionario	salario
▶	Carlos	800.5
	Ana	950
	Gustavo	897

Tabela 19: Após a consulta18

Perceba que neste resultado, a consulta retornou os salários que estavam fora da faixa de R$ 1.000,00 à R$ 1.500,00.

b.3) Operador IN ou NOT IN

O operador IN é usado para comparar um valor a uma lista de valores literais que foram especificados. **Para que TRUE seja retornado, o valor comparado deve corresponder a, pelo menos, um dos valores na lista**.
Continuando com o salário ainda, veja a consulta(consulta19) exemplo a seguir:

```
SELECT nm_funcionario, salario
        FROM tb_funcionario
        WHERE salario IN (1000.00, 1200.00, 1500.00);
```

E o resultado:

	nm_funcionario	salario
▶	Eduardo	1500
	Andre	1200

Tabela 20: Após a consulta19

Agora a mesma consulta, usando **NOT IN**:

```
SELECT nm_funcionario, salario
        FROM tb_funcionario
        WHERE salario  NOT IN (1000.00, 1200.00, 1500.00);
```

E o resultado:

	nm_funcionario	salario
▶	Carlos	800.5
	Ana	950
	Vagner	1150
	Gustavo	897

Tabela 21: Após a consulta20

Este operador (IN) produz os mesmos resultados que o uso do operador OR, mas o seu processamento é mais rápido.

Views no MySQL

Uma view é uma tabela virtual criada a partir de uma consulta a uma tabela ou mais tabelas (ou ate é mesmo de outras views) no BD

A ideia de uma view é armazenar as informações de uma consulta em SQL. Ou seja, quando utilizamos um filtro ou fazemos qualquer consulta dentro do banco de dados, geramos uma visualização temporária.

Para que essa visualização seja salva e possamos utilizá-la posteriormente, é importante salvar essa consulta dentro de uma view.

Isso é extremamente útil porque nem sempre fazemos consultas simples de apenas selecionar e visualizar uma tabela do banco de dados. Às vezes, as consultas são mais complexas e levam tempo.

Fazendo uma consulta para visualizar o resultado

Ações envolvidas numa View

(1) Criação de uma view (CREATE VIEW)

(2) Alteração de uma view (ALTER VIEW)

(3) **Exclusão de uma view (DROP VIEW**)

(1) Criando uma view

Agora vamos partir para a criação da view, lembrando que vamos começar com o comando **CREATE VIEW** e depois o nome dessa view.

```
CREATE VIEW vw_consulta1  as

SELECT cd_funcionario, nm_funcionario, nro_depto

   FROM tb_funcionario;
```

```
select *

from vw_consulta1;
```

cd_funcionario	nm_funcionario	nro_depto
1111	Carlos	3
1112	Ana	2
1113	Eduardo	1
1114	Andre	2
1115	Vagner	4
1116	Gustavo	4

(2) Alterando uma view

Para o segundo exemplo temos um código de **CREATE** com **ALTER**, utilizando o OR, pois assim se a view não existir ela será criada, e se ela existir vamos apenas alterá-la.

```
ALTER VIEW vw_consulta1  as

SELECT cd_funcionario, nm_funcionario, nro_depto

   FROM tb_funcionario;
```

```
select *

from vw_consulta1;
```

cd_funcionario	nm_funcionario	nro_depto
1111	Carlos	3
1112	Ana	2
1113	Eduardo	1
1114	Andre	2
1115	Vagner	4
1116	Gustavo	4

(3) Excluir uma view

Quando precisar excluir uma view é muito simples, basta utilizar o comando **DROP VIEW**.
Excluindo uma view

```
drop view vw_consulta1;
```

Vantagens

Reutilização	✔Sempre que necessário, possível utilizar uma consulta VIEW, pois ela fica armazenada no BD
Segurança	✔ Estamos ocultando linhas ou colunas da tabela original do BD. ✔ Desta forma, apenas alguns dados relevantes serão visualizados na VIEW
Simplificação	✔ Estamos poupando tempo para recriar vários SELECTs, o que aumenta a produtividade

Bibliografia

(a) **Fundamentals of Database Systems; Ramez Elmasri, Shamkant Navathe; The Benjamin CummingsPublishing Company; 1989;**

(b) **Sistema de Banco de Dados; Henry F. Korth, Abraham Silberschatz; Makro Books; 1995;**

(c) **MySQL 5 - Interativo - Guia Básico de Orientação e Desenvolvimento, José Augusto N. G. Manzano, Editora Erica 2007**

(d) **Projeto de Banco de Dados - Uma Visão Prática, Maurício Pereira de Abreu e Felipe Nery R. Machado. Editora Eric**

(e) **https://www.bosontreinamentos.com.br/mysql/mysql-criacao-de-tabelas-create-table-07/**