

SQL com My SQL – Manutenção dos dados nas tabelas

Inserindo Registros nas Tabelas

Para começar vamos inserir registros na tabela de Produtos. Para inserir um registro na tabela basta escrever o seguinte comando. **INSERT INTO table VALUES()**.

```
/*Insert data into TblProduto*/
INSERT INTO TblProduto VALUES
('1040107', 'Light - 350 ml - Melância', 'Lata', '350 ml', 'Melância', 4.56),
('1037797', 'Clean - 2 Litros - Laranja', 'PET', '2 Litros', 'Laranja', 16.01),
('1000889', 'Sabor da Montanha - 700 ml - Uva', 'Garrafa', '700 ml', 'Uva', 6.31),
('1004327', 'Videira do Campo - 1,5 Litros - Melância', 'PET', '1,5 Litros', 'Melância', 19.51),
('1088126', 'Linha Citros - 1 Litro - Limão', 'PET', '1 Litro', 'Limão', 7.00),
('5449314', 'Frescor do Verão - 350 ml - Limão', 'Lata', '350 ml', 'Limão', 2.46),
('1078680', 'Frescor do Verão - 470 ml - Manga', 'Garrafa', '470 ml', 'Manga', 5.18),
('1042712', 'Linha Citros - 700 ml - Limão', 'Garrafa', '700 ml', 'Limão', 4.90),
('7889756', 'Pedacos de Frutas - 1,5 Litros - Maça', 'PET', '1,5 Litros', 'Maça', 18.01),
('1002767', 'Videira do Campo - 700 ml - Cereja/Maça', 'Garrafa', '700 ml', 'Cereja/Maça', 8.41);
```

Perceba que é possível cadastrar mais de um registro na tabela, melhor do que cadastrar um por um.

É possível cadastrar os dados fora da ordem, porém você precisa especificar as colunas antes de realizar o cadastro.

Tenho uma tabela chamada **TblVendedor** e inseri os seguintes dados nela.

```
/*Insert data into TblVendedor*/
INSERT INTO TblVendedor VALUES
('00233', 'João Geraldo da Fonseca', 0.10),
('00235', 'Márcio Almeida Silva', 0.08),
('00236', 'Cláudia Moraes', 0.08);
```

Ela é composta por três colunas **Matrícula, Nome_Vendedor e Porcentagem de Comissão**.

Porém fui informado que a comissão do vendedor cuja matrícula é **00236** subiu para 11% e o vendedor com a matrícula **00236** disse que o nome dele é **João Geraldo da Fonseca Júnior**, como eu faço a alteração? Você pode fazê-lo usando o comando **UPDATE**.

```

/*Change information in a table*/
UPDATE TblVendedor SET Percentual_Comissao = 0.11
WHERE Matricula = '00236';

UPDATE TblVendedor SET Nome_Vendedor = 'José Geraldo da Fonseca Júnior'
WHERE Matricula = '00233';

```

Surgiu um problema, na tabela **TblVendedor** precisa ser modificada, ela precisa de dois campos chamados **Data_Admissão e De_Férias**. A opção mais adequada aqui é apagar a tabela e cria-la com essas novas colunas.

O comando para apagar tabelas ou bancos é o **DROP TABLE ou DROP DATABASE**.

```

/*Delete a table or a database*/
DROP DATABASE IF EXISTS Sucos;
DROP TABLE TblCliente;
DROP TABLE TblProduto;
DROP TABLE TblVendedor;

```

Agora vamos criar a **TblVendedor** novamente com os campos novos.

```

CREATE TABLE TblVendedor
(
Matricula VARCHAR(5) NOT NULL,
Nome_Vendedor VARCHAR(100) NOT NULL,
Percentual_Comissao FLOAT NOT NULL,
Data_Admissao DATE NOT NULL,
De_Ferias BIT(1) NOT NULL
);

```

E inserir os dados.

```

/*Inset data into a table*/
INSERT INTO TblVendedor VALUES
('00233', 'João Geraldo da Fonseca Júnior', 0.11, 15/08/2014, 0),
('00236', 'Cláudia Moraes', 0.08, 17/09/2013, 1),
('00237', 'Roberta Martins', 0.11, 18/03/2017, 1),
('00238', 'Péricles Alves', 0.11, 21/08/2016, 0);

```

Chave Primária

A **Chave Primária** ou **Primary Key** é a chave que identifica um registro em um banco de dados, ele deve ser único e não pode ser repetido. É um elemento importante nos bancos de dados que garante a restrição de integridade das informações. Agora fica a pergunta, como criar uma **PK** no SQL?

Existe um comando que é assim.

ALTER TABLE tabela ADD PRIMARY KEY (Coluna).

O **ALTER TABLE** é um comando que é utilizado para alterar as tabelas como adicionar uma coluna nova ou criar uma chave primária ou estrangeira.

No My SQL Workbench é possível adicionar uma PK através do assistente.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Codigo_Produto	VARCHAR(20)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Nome_Produto	VARCHAR(150)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Embalagem	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Tamanho	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Sabor	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Preço de Venda	DECIMAL(10,2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name: Codigo_Produto Data Type: VARCHAR(20)

Charset/Collation: Default Charset Default Collation

Default:

Storage: ☐ Virtual ☐ Stored

☒ Primary Key ☒ Not Null ☐ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☐ Generated

Columns Indexes Foreign Keys Triggers Partitioning Options

Apply Revert

Lá na tabela, clique no ícone de chave inglesa e o assistente irá abrir. Em seguida clique na caixinha onde está escrito **PK (Primary Key)** e em seguida clique em **Apply**. Depois irá aparecer o código SQL de criação da PK e em seguida finalize a operação.

Perceba que normalmente a **PK** é criada junto com a tabela, eu estou escrevendo esse material de maneira didática para você que está lendo entenda o que cada comando significa.

Filtros

No SQL podemos criar filtros usando o comando **WHERE** e com ele podemos criar vários filtros, como por exemplo.

```
/*Using Filters*/  
SELECT * FROM TblCliente WHERE Estado = 'SP';
```

Um filtro que traz para mim todos os clientes que moram no estado de SP.

```
SELECT * FROM TblProduto WHERE Embalagem = 'Garrafa';
```

Um filtro que traz todas as embalagens que sejam garrafas.

```
SELECT * FROM TblProduto WHERE Preco_de_Lista >= 10;
```

Outro filtro que traz todos os preços que sejam maiores ou iguais a 10. Observe que é possível utilizar sinais de maior, menor e igual nos filtros.

Veja, outro exemplo.

```
SELECT * FROM TblCliente WHERE YEAR(Data_Nascimento) BETWEEN '1990' AND '2000';
```

Aqui eu filtro todos os clientes que nasceram entre os anos de 1990 e 2000. Para colocar essa faixa de tempo entre os anos eu utilizo a cláusula **BETWEEN** e para que eu possa filtrar somente o ano, eu uso a função **YEAR** antes de especificar a coluna que quero usar para criar um filtro.

Utilizando a cláusula **AND** eu posso criar mais filtros, assim criando um filtro composto.