



Curso Técnico em Desenvolvimento de Sistemas Online

BANCO DE DADOS II

GEEaD - Grupo de Estudo de Educação a Distância

Centro de Educação Tecnológica Paula Souza

Expediente

GEEaD – CETEC
GOVERNO DO ESTADO DE SÃO PAULO
EIXO TECNOLÓGICO DE INFORMAÇÃO E COMUNICAÇÃO
CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS
FUNDAMENTOS DE INFORMÁTICA

Autor: José Mendes da Silva Neto

Revisão Técnica: Eliana Cristina Nogueira Barion e Lilian Aparecida Bertini

Revisão Gramatical: Juçara Maria Montenegro Simonsen Santos

Editoração e Diagramação: Flávio Biazim

São Paulo – SP, 2019

APRESENTAÇÃO

Este material didático do Curso Técnico em Desenvolvimento de Sistemas modalidade EaD foi elaborado especialmente por professores do Centro Paula Souza para as Escolas Técnicas Estaduais – ETECs.

O material foi elaborado para servir de apoio aos estudos dos discentes para que estes atinjam as competências e as habilidades profissionais necessárias para a sua plena formação como Técnicos em Desenvolvimento de Sistemas.

Esperamos que este livro possa contribuir para uma melhor formação e aperfeiçoamento dos futuros Técnicos.

AGENDA 4

LINGUAGEM DE MANIPULAÇÃO DE DADOS - DML (DATA MANIPULATION LANGUAGE)





MERGULHANDO NO TEMA...

O que você precisa agora é aprender a manipular as informações em um Banco de Dados. Vamos continuar trabalhando com o SGBD MySQL e parte da Linguagem SQL chamada DML, que possui comandos utilizados para inserir, alterar e excluir registros nas estruturas do Banco de Dados.

Assim como na agenda anterior, antes de cada comando será apresentada a sua **sintaxe**. Vale ainda lembrar que em linguagem de programação, quando falamos de sintaxe, nos referimos à **forma de escrever código fonte (palavras reservadas, comandos, recursos diversos)**. Os conteúdos entre os símbolos <> ou [] encontrados na sintaxe significam que os mesmos devem ser substituídos ou são opcionais, respectivamente.

Vamos em frente!!!

Para demonstrar a utilização da Linguagem SQL utilizaremos o Banco de Dados **bd_agenda2**, o mesmo utilizado na agenda anterior. Nele, temos as tabelas **curso** e **professor**, que possuem as seguintes estruturas:

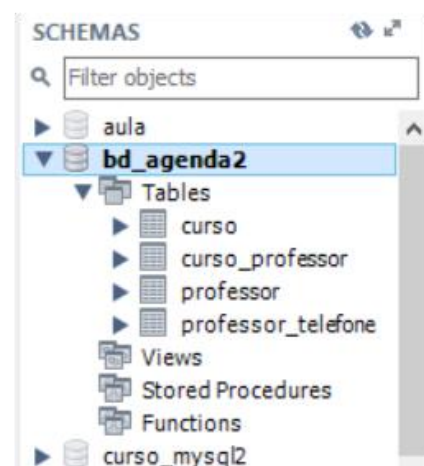
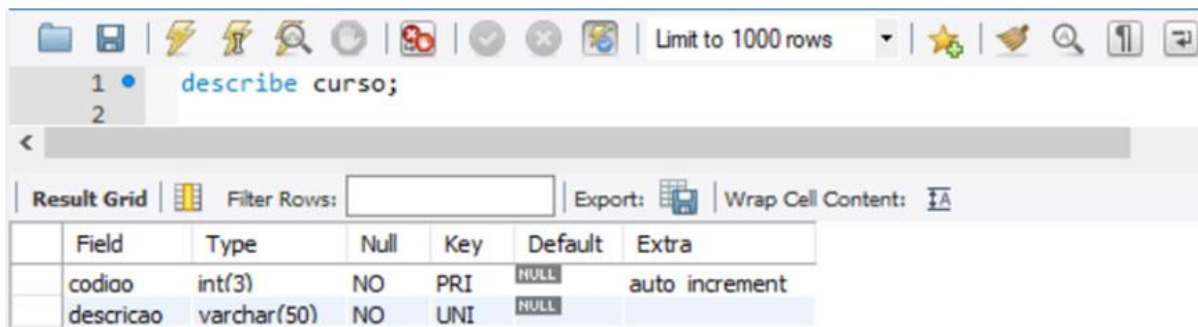


Imagem 3 – Interface Gráfica
Workbench – Objetos da Base de
Dados da agenda 2

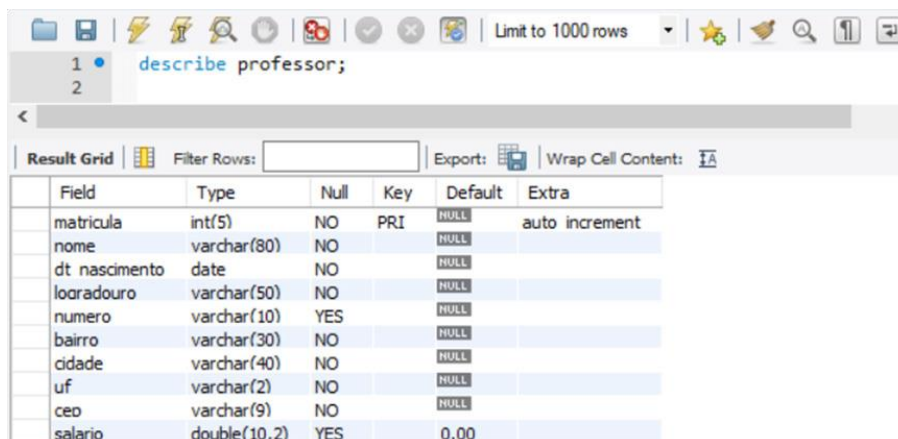


1 • describe curso;
2

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Field	Type	Null	Key	Default	Extra
codiao	int(3)	NO	PRI	NULL	auto increment
descricao	varchar(50)	NO	UNI	NULL	

Imagem 4 – Interface Gráfica Workbench – Descrição da Estrutura da Tabela curso




1 • describe professor;
2

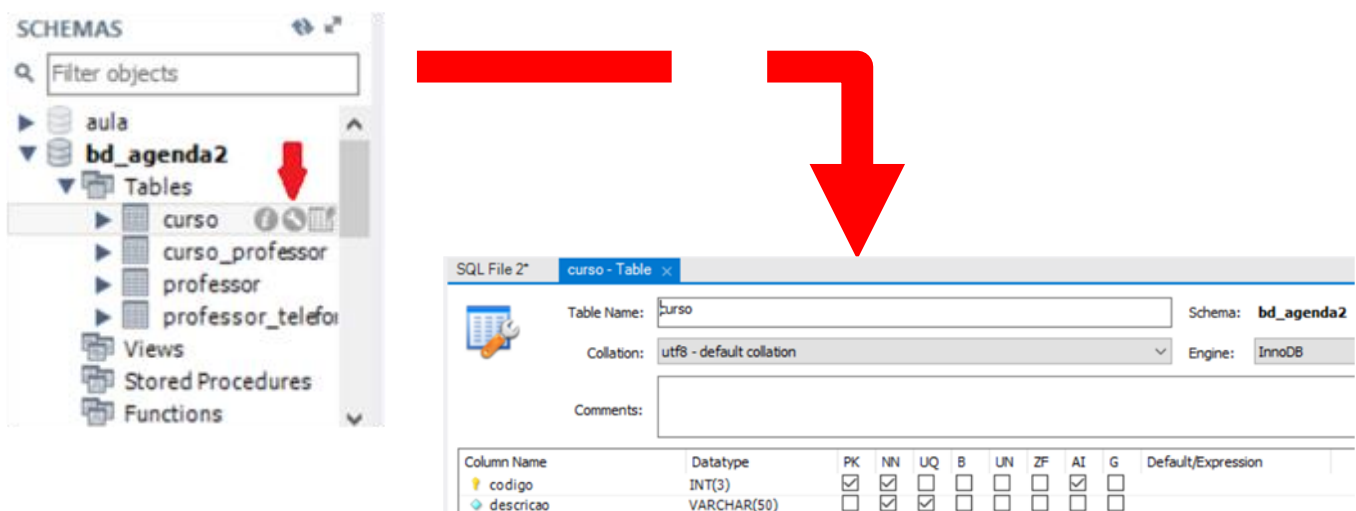
Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Field	Type	Null	Key	Default	Extra
matricula	int(5)	NO	PRI	NULL	auto increment
nome	varchar(80)	NO		NULL	
dt nascimento	date	NO		NULL	
logradouro	varchar(50)	NO		NULL	
numero	varchar(10)	YES		NULL	
bairro	varchar(30)	NO		NULL	
cidade	varchar(40)	NO		NULL	
uf	varchar(2)	NO		NULL	
cep	varchar(9)	NO		NULL	
salario	double(10,2)	YES		0.00	

Imagem 5 – Gráfica Workbench – Descrição da Estrutura da Tabela professor

Obs.: o comando describe apresenta as informações da estrutura da tabela.

Existem ainda várias outras formas para você visualizar a estrutura de uma Tabela pelo quadro SCHEMAS, uma delas é utilizando a opção  que é apresentada quando posicionamos o ponteiro do mouse sobre o nome da Tabela.



SCHEMAS

Filter objects

- aula
- bd_agenda2
 - Tables
 - curso
 - curso_professor
 - professor
 - professor_telefo
 - Views
 - Stored Procedures
 - Functions

curso - Table

Table Name: curso Schema: bd_agenda2

Collation: utf8 - default collation Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
codigo	INT(3)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
descricao	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Imagem 6 – Interface Gráfica Workbench –Estrutura da Tabela curso

Vamos agora inserir registros nessas estruturas. Para inserirmos um registro em uma Tabela utilize o comando **insert into**:

Sintaxe:

```
insert into <nome_da_tabela> (  
campo_1, ..., campo_n)  
values (  
valor_1, ..., valor_n);
```

lista de campos que receberão

lista de valores que serão
atribuídos aos campos

Exemplo:

```
insert into curso  
(descricao)  
values  
('Sistemas');
```

nome da tabela


campo da tabela que receberá o valor

valor que será atribuído ao campo

O valor do campo **codigo** não necessita ser definido porque na criação da estrutura **curso** foi definido, pela cláusula **auto_increment**, que o conteúdo seja gerado automaticamente.



Imagem 7 - Interface Gráfica Workbench – Quadro Output

Obs.: o símbolo  no quadro **OUTPUT** indica que o comando foi executado com sucesso. Para verificar se um registro foi incluído, utilize o comando a seguir:

```
select * from curso;
```

Esse comando listará todos os registros existentes na tabela **curso**.

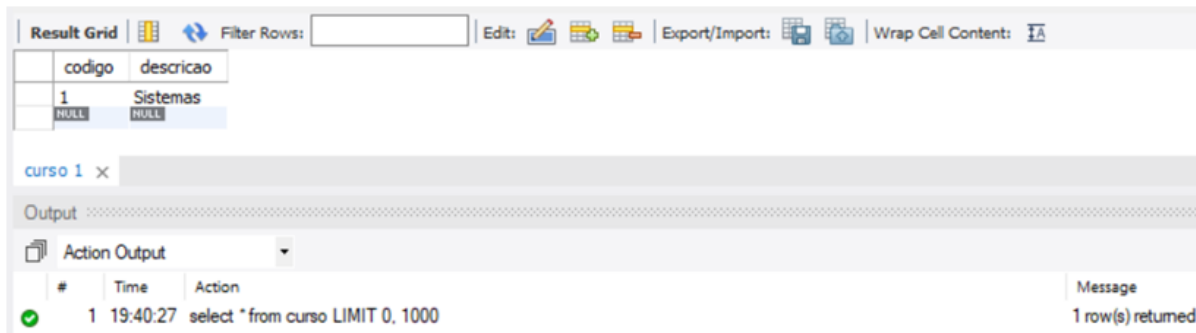


Imagem 8 - Interface Gráfica Workbench – Resultado da execução de um select

Aproveite para inserir os cursos **Designer** e **Redes**, vamos precisar deles mais à frente.

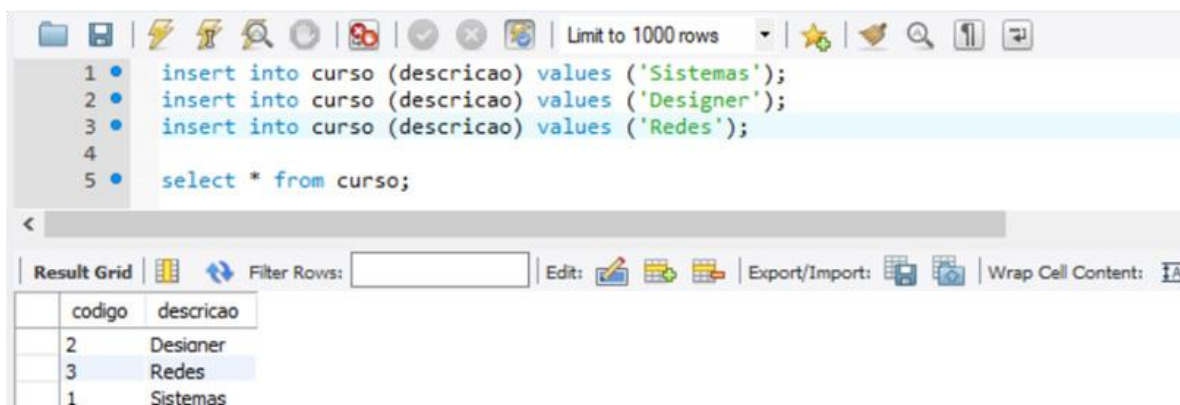


Imagem 9 - Gráfica Workbench – Inserção de registros na Tabela curso e resultado da execução de um select

Agora vamos exemplificar com a Tabela **professor**, inserindo um registro de uma pessoa que leciona no **curso** Sistemas:

```
insert into professor
(nome, dt_nascimento, logradouro, numero, bairro, cidade, uf, cep, salario)
values
('José da Silva', '1977-10-03', 'Alameda 5', 1000, 'Centro', 'São Paulo', 'SP', '08150-640', 1800);
```

campos da tabela que
receberão o valor

valores que serão atribuídos aos campos

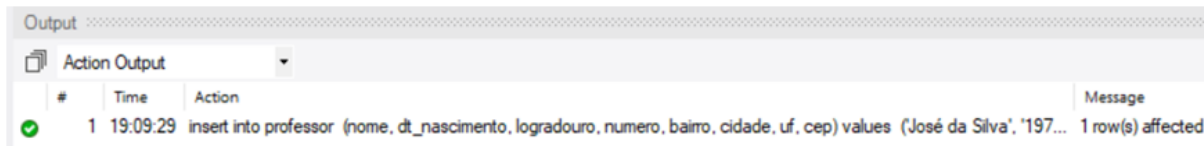


Imagem 10 - Interface Gráfica Workbench – Quadro Output, inserção do registro na Tabela professor executada com sucesso

Exemplo 2:

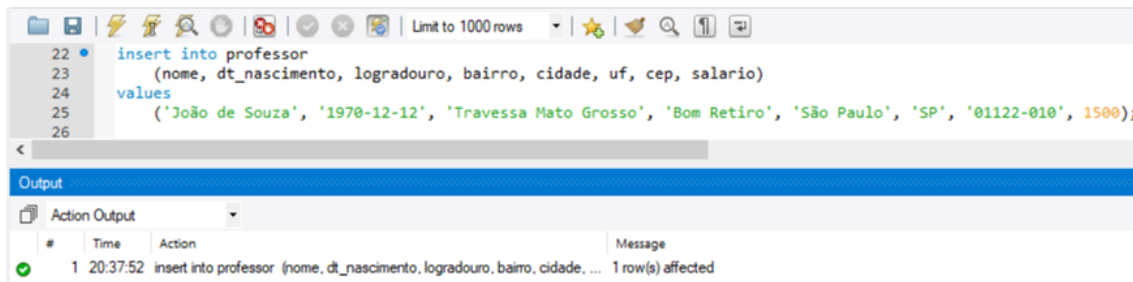


Imagem 11 - Interface Gráfica Workbench – Quadro Output, inserção do registro na Tabela professor sem a definição do campo numero executada com sucesso

Perceba que utilizamos nos dois exemplos a vírgula para separar os campos e seus respectivos valores, que serão atribuídos às colunas de acordo com sua ordem, ou seja, a primeira coluna receberá o conteúdo do primeiro valor e assim sucessivamente.

Obs.: Para atribuir um conteúdo ao um campo do tipo date utilize o formato AAAA-MM-DD.

Onde:

AAAA: corresponde ao ano.

MM: corresponde ao mês.

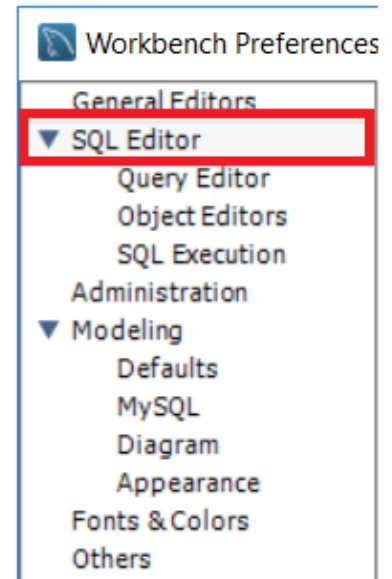
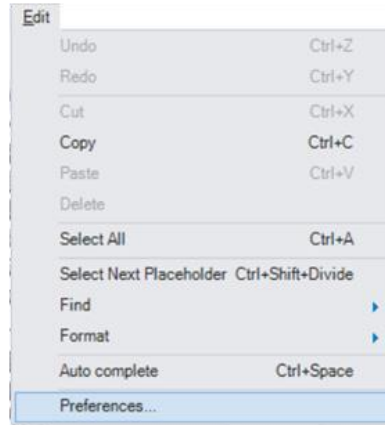
DD: corresponde ao dia.

Result Grid										
	matricula	nome	dt_nascimento	logradouro	numero	bairro	cidade	uf	cep	salario
1		José da Silva	1977-10-03	Alameda 5	1000	Centro	São Paulo	SP	08150-640	1800.00
2		João de Souza	1970-12-12	Travessa Mato Grosso	NULL	Bom Retiro	São Paulo	SP	01122-010	1500.00

Imagem 12 - Interface Gráfica Workbench – Resultado da execução de um select na Tabela professor

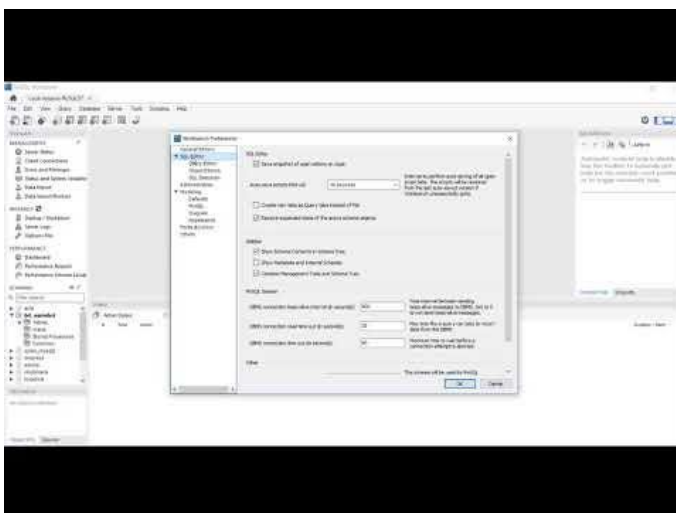
Obs: No Exemplo 2 não foi informado o campo **numero**, neste caso, quando um campo não tem valor definido ou padrão, é apresentado o valor **NULL**. Isso porque na estrutura **professor** o campo **numero** não é obrigatório.

Vamos utilizar instruções para alteração e exclusão de registros, para esses casos a instalação padrão do Workbench disponibiliza um controle de segurança que não permite alteração e exclusão de registros sem que na cláusula where tenha sido informado um campo chave. Como em nossos exemplos vamos utilizar campos que não são chaves, será necessário desabilitar essa configuração de segurança. Para isso, acesse o Menu Edit, opção Preferences, desmarcando a opção , que se encontra em SQL Editor.



Se preferir assista ao vídeo [Agenda 4 - Vídeo.mp4](#)

Imagens 13 e 14 – Interface Gráfica Workbench – Desabilitando configuração



Obs.: você deverá reiniciar o Workbench para que a nova configuração seja aplicada.

Para alterar um ou mais registros da tabela, utilize o comando **update**. Veja:

Sintaxe:

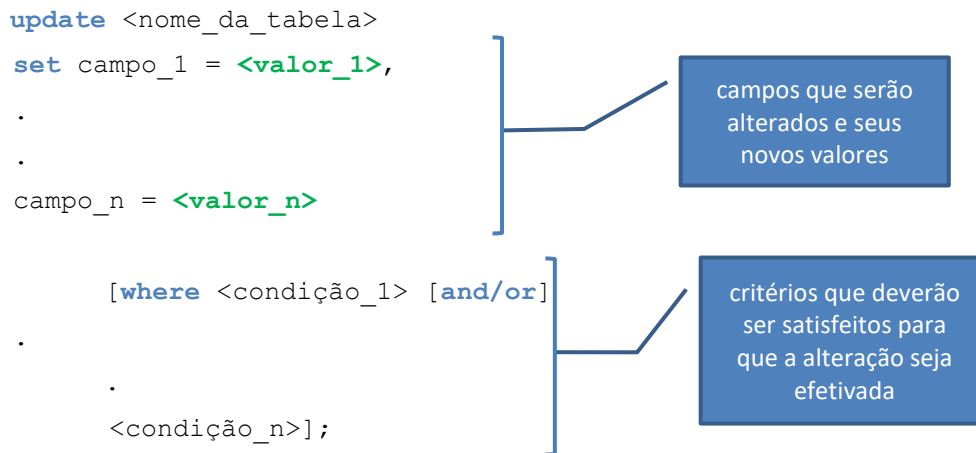


Imagem 15 - Interface Gráfica Workbench – Execução de Update na Tabela curso

Onde:

WHERE: cláusula que especifica que uma instrução deve afetar apenas as linhas que atendem aos critérios especificados.

AND/OR: operadores lógicos utilizados para combinar dois ou mais critérios em uma instrução DML ou DQL da Linguagem SQL.

Além dos operadores lógicos você também pode usar os operadores relacionais na cláusula where. Veja a tabela ao lado:

Operador	Descrição
>	Maior que
<	Menor que
=	Igual
>=	Maior ou igual
<=	Menor ou igual
!= ou <>	Diferente

Exemplo 1:

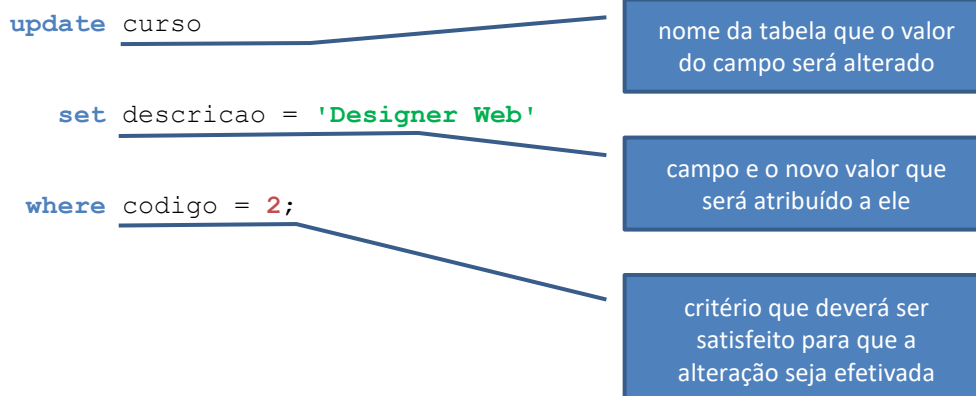




Imagem 16 – Interface Gráfica Workbench – Execução de Update na Tabela curso

O conteúdo do campo **descricao** do registro, onde o curso era **Designer**, foi alterado para **Designer Web**, uma vez que somente ele estava dentro do critério definido na instrução de alteração, **codigo=2**.

Exemplo 2:

```
update professor
set salario = salario * 1.02

where matricula = 2
and salario <= 1500;
```

campo e o novo valor que será atribuído a ele, que pode ser o resultado de um cálculo

critérios que deverão ser satisfeitos para que a alteração seja efetivada

No **Exemplo 2** foi aplicado um aumento de **2%** somente para os **professores** que recebem até **R\$ 1500,00** de salário. Neste caso, somente o **professor João de Souza** se enquadra neste critério, seu salário passou de **R\$ 1500,00 para R\$ 1530,00**.

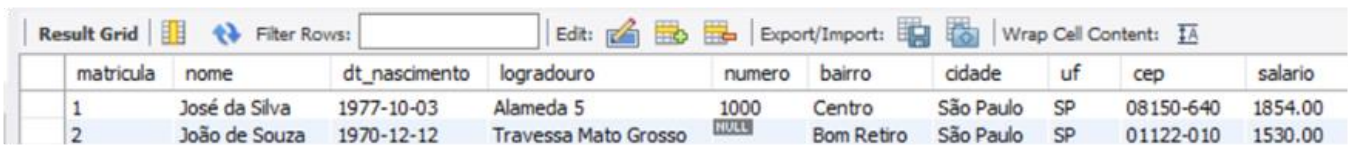
matricula	nome	dt_nascimento	logradouro	numero	bairro	cidade	uf	cep	salario
1	José da Silva	1977-10-03	Alameda 5	1000	Centro	São Paulo	SP	08150-640	1800.00
2	João de Souza	1970-12-12	Travessa Mato Grosso	NULL	Bom Retiro	São Paulo	SP	01122-010	1530.00

Imagem 17 - Interface Gráfica Workbench – Resultado da execução de um select na Tabela professor

Exemplo 3:

```
update professor
set salario = salario * 1.03
where salario < 1400
or salario > 1700;
```

No **Exemplo 3** foi aplicado um aumento de **3%** somente para os professores que recebem menos que **R\$ 1400,00** “OU” mais de **R\$ 1700,00** de salário. Neste caso, somente o professor **José da Silva** se enquadrou nestes critérios.



	matricula	nome	dt_nascimento	logradouro	numero	bairro	cidade	uf	cep	salario
1		José da Silva	1977-10-03	Alameda 5	1000	Centro	São Paulo	SP	08150-640	1854.00
2		João de Souza	1970-12-12	Travessa Mato Grosso	1000	Bom Retiro	São Paulo	SP	01122-010	1530.00

Imagem 18 - Interface Gráfica Workbench – Resultado da execução de um select na Tabela professor

Aqui é muito importante a utilização da cláusula **where**, limitando o efeito da alteração, que neste caso poderia causar uma perda financeira para a escola, caso outros registros fossem alterados.

Para excluir um ou mais registros da tabela, utilize o comando **delete**.

Veja:

Sintaxe:

```
delete from <nome_da_tabela>
```

```
[where <condição_1> [and/or]
```

```
.
```

```
.
```

```
<condição_n>];
```

critérios que deverão ser satisfeitos para que a exclusão seja efetivada

Exemplo:

```
delete from curso
```

nome da tabela que terá o registro excluído

```
where codigo = 3;
```

critério que deverá ser satisfeito para que a exclusão do registro seja efetivada

Para verificar se um registro foi excluído, utilize o comando a seguir:

```
select *
```

```
from curso;
```

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
	codigo	descricao		
2	2	Desianer		
1	1	Sistemas		

Imagem 19 - Interface Gráfica Workbench – Resultado da execução de um select na Tabela curso

Assim como no comando **update**, no **delete** você também pode utilizar os **operadores lógicos** **and** e **or** na cláusula **where**, que por sua vez não é obrigatória, mas com certeza é fundamental para evitar perda de dados. A utilização desses **operadores** sem uma análise criteriosa, pode levar a perda de todos os registros de uma tabela de uma só vez, portanto muito cuidado!

No exemplo anterior, somente o registro do curso **Redes** foi excluído, pois na cláusula where consta o critério **codigo=3**, ou seja, onde o conteúdo do campo **codigo** for igual a **3**.

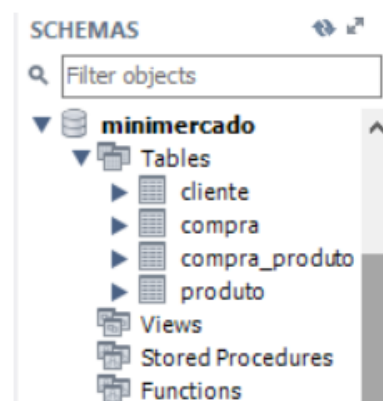
Agora é com você!!!

VOCÊ NO COMANDO

Vamos utilizar o Projeto do Banco de Dados do Sistema do minimercado do Adriano que já está pronto. Inclua pelo menos 5 (cinco) registros de Produtos e Clientes.

Simule a compra desses produtos pelos clientes, além de alterar e a excluir cadastros, como a alteração de um endereço de cliente e a exclusão do cadastro de um produto que tenha sido realizado de maneira equivocada.

O maior desafio aqui é a definir a sequência correta para inclusão dos registros. Utilize o WorkBench para auxiliá-lo nessa tarefa e lembre-se, as Tabelas que possuem **chaves estrangeiras** devem ficar por último.



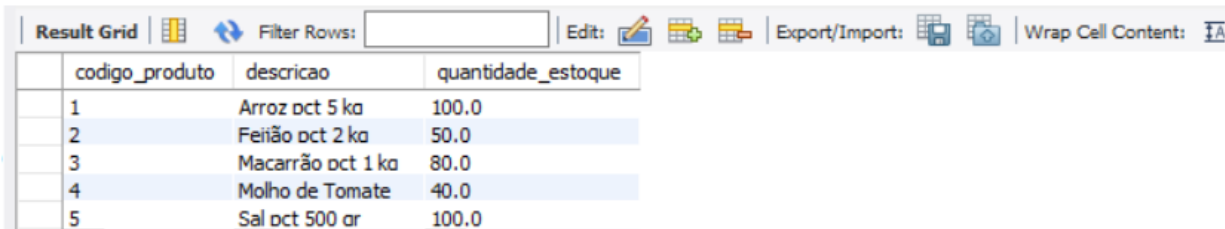
Obs.: Não esqueça de selecionar o Banco de Dados antes de iniciar a execução ⚡ das instruções.

Tudo certo? Você conseguiu? Tenho certeza que sim!!!! Vamos juntos conferir!!!

Como esse contexto possui **chaves estrangeiras**, cujo principal objetivo é manter a **integridade referencial** dos conteúdos atribuídos a esses **campos**, nenhuma compra de **produto** pode ser realizada sem que antes o **cliente** e o **produto** estejam previamente cadastrados, correto!!!

Começaremos então pela inclusão dos Produtos:

```
insert into produto (descricao, quantidade_estoque) values ('Arroz pct 5 kg', 100);
insert into produto (descricao, quantidade_estoque) values ('Feijão pct 2 kg', 50);
insert into produto (descricao, quantidade_estoque) values ('Macarrão pct 1 kg', 80);
insert into produto (descricao, quantidade_estoque) values ('Molho de Tomate', 40);
insert into produto (descricao, quantidade_estoque) values ('Sal pct 500 gr', 100);
```



codigo_produto	descricao	quantidade_estoque
1	Arroz pct 5 kg	100.0
2	Feijão pct 2 kg	50.0
3	Macarrão pct 1 kg	80.0
4	Molho de Tomate	40.0
5	Sal pct 500 gr	100.0

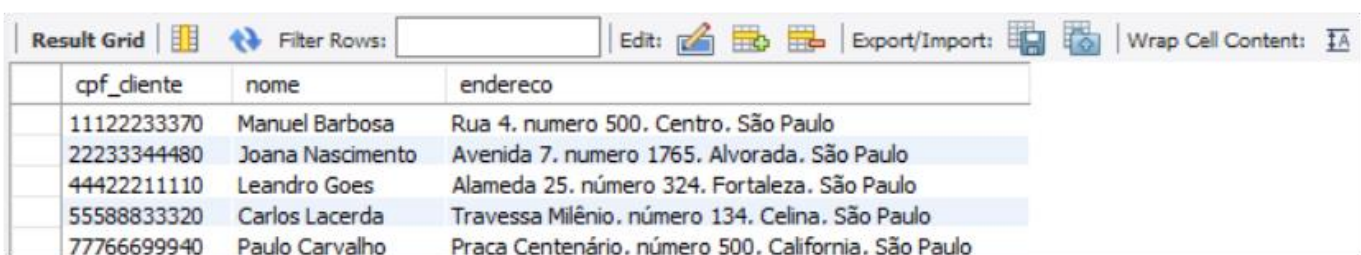
Imagem 18 - Interface Gráfica Workbench – Confirmação da inclusão de registros na Tabela produto

Vamos fazer também a inclusão de um produto que não tem nada a ver com o minimercado. Mais a frente utilizaremos esse registro para simular o processo de exclusão de um produto que foi cadastrado de maneira equivocada.

```
insert into produto (descricao, quantidade_estoque) values ('Cimento sc 50 kg', 1000);
```

Agora vamos incluir os Clientes:

```
insert into cliente (cpf_cliente, nome, endereco) values
    ('11122233370', 'Manuel Barbosa', 'Rua 4, numero 500, Centro, São Paulo');
insert into cliente (cpf_cliente, nome, endereco) values
    ('22233344480', 'Joana Nascimento', 'Avenida 7, numero 1765, Alvorada, São Paulo');
insert into cliente (cpf_cliente, nome, endereco) values
    ('44422211110', 'Leandro Goes', 'Alameda 25, número 324, Fortaleza, São Paulo');
insert into cliente (cpf_cliente, nome, endereco) values
    ('55588833320', 'Carlos Lacerda', 'Travessa Milênio, número 134, Celina, São Paulo');
insert into cliente (cpf_cliente, nome, endereco) values
    ('77766699940', 'Paulo Carvalho', 'Praça Centenário, número 500, California, São Paulo');
```



cpf_cliente	nome	endereco
11122233370	Manuel Barbosa	Rua 4, numero 500, Centro, São Paulo
22233344480	Joana Nascimento	Avenida 7, numero 1765, Alvorada, São Paulo
44422211110	Leandro Goes	Alameda 25, número 324, Fortaleza, São Paulo
55588833320	Carlos Lacerda	Travessa Milênio, número 134, Celina, São Paulo
77766699940	Paulo Carvalho	Praça Centenário, número 500, California, São Paulo

Imagem 19 - Interface Gráfica Workbench – Confirmação da inclusão de registros na Tabela cliente

Com os cadastros prontos de **produtos** e **clientes**, o Banco de Dados está preparado para receber os registros de **compras**. Vamos lançar uma compra para o Cliente Paulo Carvalho e outra para o cliente Leandro Goes.

1. Compra

```
insert into compra (data, cpf_cliente) values ('2019-10-01', 77766699940);
insert into compra_produto (codigo_compra, codigo_produto, quantidade, preco) values
(1, 1, 2, 8.90);
insert into compra_produto (codigo_compra, codigo_produto, quantidade, preco) values
(1, 2, 2, 5.40);
```

2. Compra

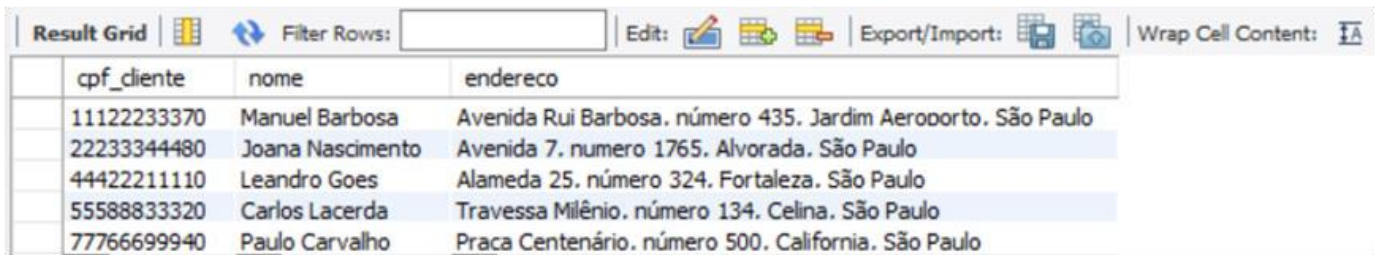
```
insert into compra (data, cpf_cliente) values ('2019-10-10', 44422211110);
insert into compra_produto (codigo_compra, codigo_produto, quantidade, preco) values
(2, 3, 1, 2.45);
insert into compra_produto (codigo_compra, codigo_produto, quantidade, preco) values
(2, 4, 2, 3.30);
insert into compra_produto (codigo_compra, codigo_produto, quantidade, preco) values
(2, 5, 1, 4.70);
```

Observe que para as duas compras foi necessário incluímos primeiramente o cabeçalho da compra e só depois os produtos, isso porque o registro de produtos comprados necessita do código da compra para relacioná-los.



*Vamos agora alterar um dos registros de cliente. Como ficaria a instrução SQL se fosse necessária a alteração do endereço do cliente Manuel Barbosa, que atualmente mora na Avenida Rui Barbosa, número 435, Jardim Aeroporto, São Paulo? Para alterar o endereço do cliente vamos utilizar o comando **update** da seguinte forma:*

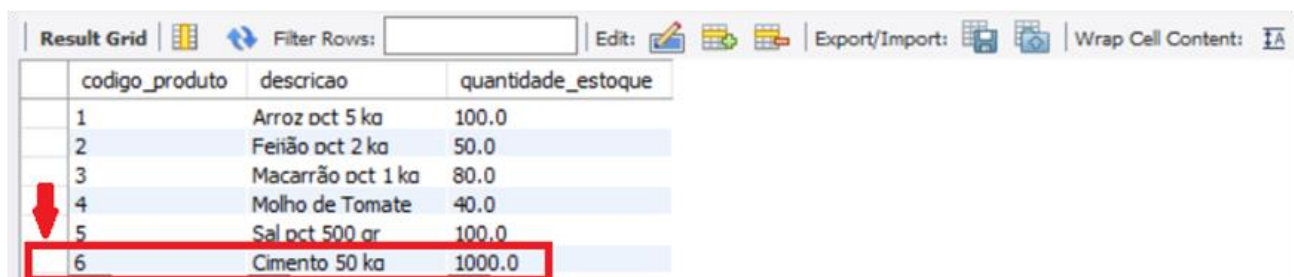
```
update cliente set endereco = 'Avenida Rui Barbosa, número 435, Jardim Aeroporto, São Paulo' where cpf_cliente = '11122233370';
```



cpf_cliente	nome	endereco
11122233370	Manuel Barbosa	Avenida Rui Barbosa, número 435, Jardim Aeroporto, São Paulo
22233344480	Joana Nascimento	Avenida 7, numero 1765, Alvorada, São Paulo
44422211110	Leandro Goes	Alameda 25, número 324, Fortaleza, São Paulo
55588833320	Carlos Lacerda	Travessa Milênio, número 134, Celina, São Paulo
77766699940	Paulo Carvalho	Praca Centenário, número 500, California, São Paulo

Imagem 20 - Gráfica Workbench – Confirmação da alteração de registro na Tabela cliente

Lembra do produto que incluímos para que fosse utilizado posteriormente, vamos agora excluir esse registro.

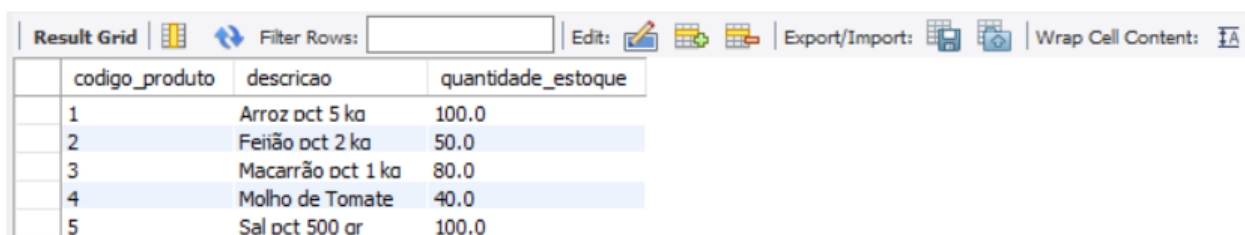


codigo_produto	descricao	quantidade_estoque
1	Arroz oct 5 ka	100.0
2	Feijão oct 2 ka	50.0
3	Macarrão oct 1 ka	80.0
4	Molho de Tomate	40.0
5	Sal oct 500 gr	100.0
6	Cimento 50 ka	1000.0

Imagem 21 - Interface Gráfica Workbench – Registro da Tabela produto que deverá ser excluído

Para excluir esse produto vamos utilizar o comando `delete` da seguinte forma:

```
delete from produto where codigo_produto = 6;
```



codigo_produto	descricao	quantidade_estoque
1	Arroz oct 5 ka	100.0
2	Feijão oct 2 ka	50.0
3	Macarrão oct 1 ka	80.0
4	Molho de Tomate	40.0
5	Sal oct 500 gr	100.0

Imagem 22 - Interface Gráfica Workbench – Confirmação da exclusão do registro da Tabela produto

Importante: utilizamos a **chave primária** como critério na cláusula `where` dos dois exemplos, com isso teremos a certeza que as operações de alteração e exclusão serão aplicadas somente em um registro, ou seja, no cliente cujo conteúdo do campo `cpf` é igual a '11122233370' e o do campo `codigo_produto` é igual a 6.

É isso aí!!! Vamos agora finalizar essa agenda colocando a mão na massa.