Banco de dados 1

Linguagem SQL – DDL e DML
Parte 2

Professor: Victor Hugo L. Lopes

Banco de Dados 1

Agenda:

DML:

- Criando registros
- Alterando e removendo registros
- Selecionando registros

Banco de Dados 1

Criando registros em uma relação

Para criarmos registros em uma referida tabela existente no banco de dados, utilizamos o comando insert...

Sintaxe:

INSERT INTO <nome_tabela> (<lista_de_colunas>) VALUES (<lista_de_valores>);

Onde a lista de colunas não é obrigatória.

Banco de Dados 1

Criando registros em uma relação

ex.:
Tendo em vista a relação:
cliente(idCliente, nomeCliente, cpf, nascimento)

Então:

INSERT INTO cliente VALUES (10, 'Adalberdson Creidson', '000.000.000-00', '1982-07-23');

Banco de Dados 1

Criando registros em uma relação

Avaliando o exemplo:

cliente(idCliente, nomeCliente, cpf, nascimento)

Note que não foi informado a lista de colunas, então a lista de valores deve respeitar a ordem das colunas existentes na tabela.

INSERT INTO cliente VALUES (10, 'Adalberdson Creidson', '000.000.000-00', '1982-07-23');

Banco de Dados 1

Criando registros em uma relação

Suprimindo colunas na inserção:

cliente(idCliente, nomeCliente, cpf, nascimento)

Veja que a coluna idCliente foi suprimida na lista de colunas, e a lista de valores também respeita a ordem das colunas informadas.

```
INSERT INTO cliente (nomeCliente, cpf, nascimento) VALUES ('Adalberdson Creidson', '000.000.000-00', '1982-07-23');
```

Banco de Dados 1

Criando registros em uma relação

Inserindo mais de um registro por vez:

cliente(idCliente, nomeCliente, cpf, nascimento)

```
INSERT INTO cliente (nomeCliente, cpf, nascimento) VALUES
('Adalberdson Creidson', '000.000.000-00', '1982-07-23'),
('Manoelson Valdo', '001.001.001-01', '1975-01-27');
```

Banco de Dados 1

E pra corrigir um valor informado?

Banco de Dados 1

Alterando registros

Para alteração de valores em um registro, utiliza-se o comando update, seguindo a sintaxe:

UPDATE <nome_tabela> SET <coluna> = <valor>;

Exemplo:

Update cliente set cidade = 'Formosa';

Como resultado, todos os registros terão a coluna com o valor 'Formosa' após a execução do comando.

Banco de Dados 1

Alterando registros

Para contornar a alteração global, pode-se recorrer à clausula Where, que especifica uma condição lógica para garantir a alteração.

UPDATE <nome_tabela> SET <coluna> = <valor> WHERE <condição>;

Exemplo:

Update cliente set cidade = 'Formosa' WHERE idCliente = 1;

Banco de Dados 1

Alterando registros

Ex 2:

```
Update conta
        set saldo = saldo * 1.05
        where saldo >= 1000;
```

Ex 3

```
Update conta
        set saldo = case
                        when saldo <= 1000 then saldo * 1.05
                        else saldo * 1.10
                        end;
```

Banco de Dados 1

Alterando registros

Forma geral da instrução case:

```
Case
        when predicado1 then resultado1
        when predicado2 then resultado2
        …
        qhen predicado n  then resultado n
        eles resultado

End
```

Banco de Dados 1

Alterando registros

Exercitando:
Projete e implemente um banco de dados para uma instituição bancária, que deve prover recursos para cadastro das contas dos clientes e suas movimentações financeiras.

Popule este banco de dados implementado com clientes diversos, contas correntes e/ou poupanças, além das movimentações destas contas.

Banco de Dados 1

Passando uma borracha em registros

Banco de Dados 1

Removendo registros

Uma requisição de exclusão é expressa utilizando a instrução delete...

Sintaxe:

DELETE from <tabela> WHERE <condição>;

Banco de Dados 1

Removendo registros

ex.:

```
banco1=# select * from cliente;
 idcliente |  nomecliente    | nascimento
-----------+-----------------+------------
         4 | nome novo       | 1980-01-01
         1 | Astolfo Silva   |
         2 | Carlos Paulada  | 2014-01-01
         3 | Antônio Silva   | 2014-01-01
(4 rows)
```

Delete from cliente where nomecliente = 'nome novo';

```
banco1=# select * from cliente;
 idcliente |  nomecliente    | nascimento
-----------+-----------------+------------
         1 | Astolfo Silva   |
         2 | Carlos Paulada  | 2014-01-01
         3 | Antônio Silva   | 2014-01-01
(3 rows)
```

Banco de Dados 1

Removendo registros

Atenção que caso o comando não contenha a cláusula where, todo o conteúdo da tabela será excluído.

Delete from cliente;

Removendo registros

Outros exemplos:

- Exclua todas as tuplas *conta* na agência Perryridge.

```
delete from conta
where nome_agência = 'Perryridge'
```

- Exclua todos os empréstimos com quantia entre $1.300 e $1.500.

```
delete from empréstimo
where quantia between 1300 and 1500
```

Banco de Dados 1

SELECT

Dentro da visão do Crud, especificamente sobre a capacidade de leitura de dados, utilizamos a estrutura de seleção, o SELECT, para extrairmos dados catalogados em tabelas de um banco de dados padrão SQL.

Sintaxe:

SELECT <lista de colunas separadas por vírgula>
FROM <nome da tabela>

Banco de Dados 1

SELECT

A estrutura SELECT cria uma nova tabela, que não existirá no banco de dados, sendo somente uma projeção dos registros encontrados na tabela, levando-se em consideração a lista de colunas indicada.

O SGBD varre toda a tabela e cria uma “visualização” (projeção) dos registros encontrados.

Banco de Dados 1

SELECT

ex.:

	Tabela1
	Id – Integer serial Nome – varchar(100) Ende- Varchar(100) Cidade Varchar(100) Estado - Varchar(2)

	id integer	nome character varying(100)	ende character varying(100)	cidade character varying(100)	estado character varying(2)
1	0	Apolônio Silva	Rua 01, n 10	Formosa	GO
2	1	Juca Lambuca	Rua 02, n 11	Formoso	TO
3	2	Pedro Timão	Rua 03, n 1	Brasília	DF
4	3	Paulo Paulada	Rua 01, n 100	Formosa	FO

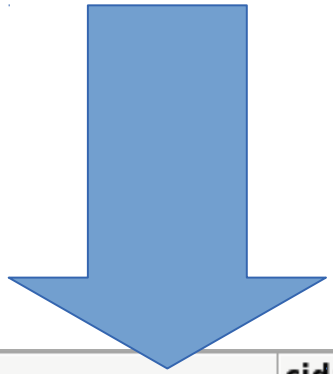
A.SELECT <lista de colunas separadas por vírgula> FROM <nome da tabela>

SELECT nome, ende, cidade, estado FROM Tabela1;

Banco de Dados 1

SELECT

	id integer	nome character varying(100)	ende character varying(100)	cidade character varying(100)	estado character varying(2)
1	0	Apolônio Silva	Rua 01, n 10	Formosa	GO
2	1	Juca Lambuca	Rua 02, n 11	Formoso	TO
3	2	Pedro Timão	Rua 03, n 1	Brasília	DF
4	3	Paulo Paulada	Rua 01, n 100	Formosa	FO



	nome character varying(100)	ende character varying(100)	cidade character varying(100)	estado character varying(2)
1	Apolônio Silva	Rua 01, n 10	Formosa	GO
2	Juca Lambuca	Rua 02, n 11	Formoso	TO
3	Pedro Timão	Rua 03, n 1	Brasília	DF
4	Paulo Paulada	Rua 01, n 100	Formosa	FO

Banco de Dados 1

SELECT

Em uma estrutura SELECT, a lista de colunas não precisa respeitar a ordem de criação das colunas da tabela:

SELECT nome, ende, cidade, estado FROM Tabela1;

	nome character varying(100)	ende character varying(100)	cidade character varying(100)	estado character varying(2)
1	Apolônio Silva	Rua 01, n 10	Formosa	GO
2	Juca Lambuca	Rua 02, n 11	Formoso	TO
3	Pedro Timão	Rua 03, n 1	Brasília	DF
4	Paulo Paulada	Rua 01, n 100	Formosa	FO

SELECT cidade, estado, nome FROM Tabela1;

	cidade character varying(100)	estado character varying(2)	nome character varying(100)
1	Formosa	GO	Apolônio Silva
2	Formoso	TO	Juca Lambuca
3	Brasília	DF	Pedro Timão
4	Formosa	FO	Paulo Paulada

Banco de Dados 1

SELECT

Quando não se sabe todas as colunas existentes da tabela, o há a necessidade de obtermos todas as colunas na projeção, utiliza-se o caractere asterisco (*):

SELECT * FROM Tabela1;

	id integer	nome character varying(100)	ende character varying(100)	cidade character varying(100)	estado character varying(2)
1	0	Apolônio Silva	Rua 01, n 10	Formosa	GO
2	1	Juca Lambuca	Rua 02, n 11	Formoso	TO
3	2	Pedro Timão	Rua 03, n 1	Brasília	DF
4	3	Paulo Paulada	Rua 01, n 100	Formosa	FO

A. Serão extraídos todas as colunas, na ordem de sua criação.

Banco de Dados 1

SELECT

Quando não se sabe todas as colunas existentes da tabela, o há a necessidade de obtermos todas as colunas na projeção, utiliza-se o caractere asterisco (*):

SELECT * FROM Tabela1;

	id integer	nome character varying(100)	ende character varying(100)	cidade character varying(100)	estado character varying(2)
1	0	Apolônio Silva	Rua 01, n 10	Formosa	GO
2	1	Juca Lambuca	Rua 02, n 11	Formoso	TO
3	2	Pedro Timão	Rua 03, n 1	Brasília	DF
4	3	Paulo Paulada	Rua 01, n 100	Formosa	FO

A. Serão extraídos todas as colunas, na ordem de sua criação.
B. Vantagens???
C. Desvantagens???

SELECT

Como podemos notar, a saída da projeção de uma estrutura SELECT utiliza o próprio nome da coluna no cabeçalho da tabela de saída.

Há a possibilidade de sobrescrevermos os nomes das colunas na tabela projetada:

SELECT nome AS “FullName” FROM Tabela1;

	FullName character varying(100)
1	Apolônio Silva
2	Juca Lambuca
3	Pedro Timão
4	Paulo Paulada

Sobrescrita de coluna utilizando *?

Banco de Dados 1

SELECT

Espero que tenhamos observado que a ordem dos registros apresentados na projeção é a mesma da ordem de inserção dos registros:

SELECT * FROM Tabela1;

	id integer	nome character varying(100)	ende character varying(100)	cidade character varying(100)	estado character varying(2)
1	0	Apolônio Silva	Rua 01, n 10	Formosa	GO
2	1	Juca Lambuca	Rua 02, n 11	Formoso	TO
3	2	Pedro Timão	Rua 03, n 1	Brasília	DF
4	3	Paulo Paulada	Rua 01, n 100	Formosa	FO

Mas, isto pode não ser uma verdade absoluta, pois o SGBD é livre para fazer alterações na organização em disco dos registros de uma tabela.

Banco de Dados 1

SELECT

ex.:

Execute o comando SQL:

```
UPDATE Tabela1 set nome='Pedro João' WHERE nome='Pedro Timão';
```

Agora se fizermos novamente a seleção:

```
SELECT * FROM Tabela1;
```

	id integer	nome character varying(100)	ende character varying(100)	cidade character varying(100)	estado character varying(2)
1	0	Apolônio Silva	Rua 01, n 10	Formosa	GO
2	1	Juca Lambuca	Rua 02, n 11	Formoso	TO
3	3	Paulo Pauleira	Rua 01, n 100	Formosa	FO
4	2	Pedro João	Rua 03, n 1	Brasília	DF

Banco de Dados 1

SELECT

Em resumo: se não especificarmos na instrução qual a ordem queremos, o SGBD é quem vai se preocupar com isto.
Mas..., e se precisarmos de uma ordem específica??

Podemos indicar na estrutura SELECT a ordem que queremos, através da cláusula ORDER BY:

Sintaxe:
SELECT <lista de colunas separadas por vírgula> FROM <Nome da Tabela> ORDER BY <Nome da Coluna> [ASC | DESC]

Banco de Dados 1

SELECT

ex.:

SELECT * FROM Tabela1 ORDER BY id;

	id integer	nome character varying(100)	ende character varying(100)	cidade character varying(100)	estado character varying(2)
1	0	Apolônio Silva	Rua 01, n 10	Formosa	GO
2	1	Juca Lambuca	Rua 02, n 11	Formoso	TO
3	2	Pedro João	Rua 03, n 1	Brasília	DF
4	3	Paulo Pauleira	Rua 01, n 100	Formosa	FO

Parâmetro opcional ASC ou DESC: ASC é padrão.

Banco de Dados 1

SELECT

ex.:

SELECT * FROM Tabela1 ORDER BY id DESC;

	id integer	nome character varying(100)	ende character varying(100)	cidade character varying(100)	estado character varying(2)
1	3	Paulo Pauleira	Rua 01, n 100	Formosa	FO
2	2	Pedro João	Rua 03, n 1	Brasília	DF
3	1	Juca Lambuca	Rua 02, n 11	Formoso	TO
4	0	Apolônio Silva	Rua 01, n 10	Formosa	GO

Parâmetro opcional ASC ou DESC: ASC é padrão.

Banco de Dados 1

SELECT

ex.:

Utilizando agora mais de uma coluna como ordenador:
SELECT * FROM Tabela1 ORDER BY cidade DESC,estado;

	id integer	nome character varying(100)	ende character varying(100)	cidade character varying(100)	estado character varying(2)
1	1	Juca Lambuca	Rua 02, n 11	Formoso	TO
2	3	Paulo Pauleira	Rua 01, n 100	Formosa	FO
3	0	Apolônio Silva	Rua 01, n 10	Formosa	GO
4	2	Pedro João	Rua 03, n 1	Brasília	DF

Banco de Dados 1

SELECT

ex.:

A.E se tivermos sobrescrito o nome de uma coluna?

```
SELECT nome, cidade AS "City", estado FROM Tabela1 ORDER BY "City" DESC,estado;
```

```
SELECT nome, cidade AS "City", estado FROM Tabela1 ORDER BY cidade DESC,estado;
```

Banco de Dados 1

SELECT

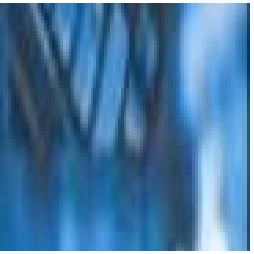
Removendo entradas duplicadas da projeção:

A.Adicionando a clausula DISTINCT à estrutura SELECT, conseguimos eliminar listagens duplicadas:

```
SELECT DISTINCT cidade from Tabela1;
```

	cidade character varying(100)
1	Formosa
2	Formoso
3	Brasília

Banco de Dados 1



SELECT

Realizando cálculos simples:

A.É possível realizar operações básicas e apresentá-las como colunas na projeção:

```
bpsimple=> SELECT description, cost_price FROM item;
  description   |  cost_price
----------------+-------------
 Wood Puzzle    |       15.23
 Rubic Cube     |        7.45
 Linux CD       |        1.99
 Tissues        |        2.11
 Picture Frame  |        7.54
 Fan Small      |        9.23
 Fan Large      |       13.36
 Toothbrush     |        0.75
 Roman Coin     |        2.34
 Carrier Bag    |        0.01
 Speakers       |       19.73
(11 rows)
```

```
bpsimple=> SELECT description, cost_price * 100 FROM item;
  description   |  ?column?
----------------+-------------
 Wood Puzzle    |  1523.00
 Rubic Cube     |   745.00
 Linux CD       |   199.00
 Tissues        |   211.00
 Picture Frame  |   754.00
 Fan Small      |   923.00
 Fan Large      |  1336.00
 Toothbrush     |    75.00
 Roman Coin     |   234.00
 Carrier Bag    |     1.00
 Speakers       |  1973.00
(11 rows)
```

Banco de Dados 1

SELECT

Realizando cálculos simples:

```
CREATE TABLE item (
        descricao varchar(100),
        precoCusto float,
        precoVenda float,
        qtd int
);
```

insert into item values ('Caneta Bic Cristal',0.89, 1.99, 10),
('Borracha macia',0.19, 0.99, 32),
('Lápis Faber-castel',0.69, 1.99, 3)

Banco de Dados 1

SELECT

Realizando cálculos simples:

```
SELECT descricao, precoCusto AS "Custo", precoVenda AS
"Venda", precoCusto * qtd from item;
```

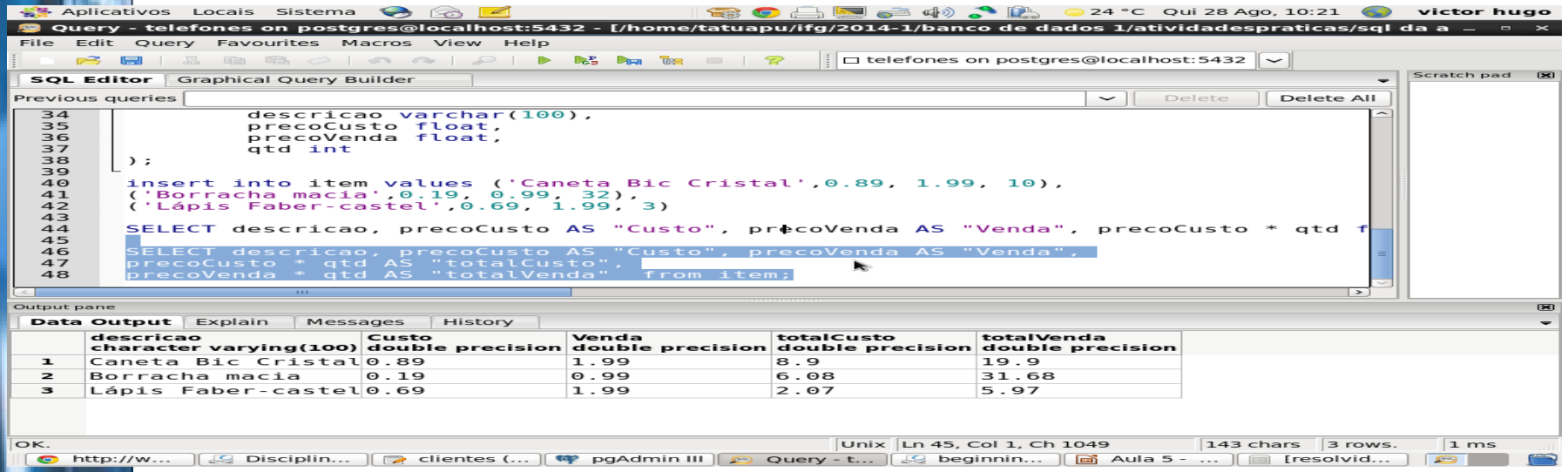
	descricao character varying(100)	Custo double precision	Venda double precision	?column? double precision
1	Caneta Bic Cristal	0.89	1.99	8.9
2	Borracha macia	0.19	0.99	6.08
3	Lápis Faber-castel	0.69	1.99	2.07

Banco de Dados 1

SELECT

Realizando cálculos simples:
Nomeando o cálculo

```
SELECT descricao, precoCusto AS "Custo", precoVenda AS "Venda",
precoCusto * qtd AS "totalCusto",
precoVenda * qtd AS "totalVenda"  from item;
```



Banco de Dados 1

SELECT

Realizando cálculos simples:
Nomeando o cálculo

```
SELECT descricao, precoCusto AS "Custo", precoVenda AS "Venda",
precoCusto * qtd AS "totalCusto",
precoVenda * qtd AS "totalVenda"  from item;
```