

1




2

# Modificando dados com UPDATE

► Sintaxe:

```
UPDATE nome_da_tabela
SET nome_coluna = expressão
[WHERE condição];
```



3

# Modificando um atributo

```
SELECT nomeVendedor
FROM vendedor
WHERE matriculaVendedor = 30;
```

nomevendedor
FERNANDO TAVARES SABINO


1 row(s)

```
UPDATE vendedor
SET nomeVendedor = 'Joao da Silva'
WHERE matriculaVendedor = 30;
```

```
SELECT nomeVendedor
FROM vendedor
WHERE matriculaVendedor = 30;
```

nomevendedor
Joao da Silva

1 row(s)



Exercício Prático

4

## Modificando mais de um atributo

```
SELECT matriculaVendedor, nomeVendedor, CPFVendedor
FROM vendedor
WHERE matriculaVendedor = 30;
```

matriculavendedor	nomevendedor	cpf vendedor
30	Joao da Silva	17507050772

1 row(s)

```
UPDATE vendedor
SET nomeVendedor = 'Pedro Albuquerque',
    CPFVendedor = '28618161883'
WHERE matriculaVendedor = 30;
```

```
SELECT matriculaVendedor, nomeVendedor, CPFVendedor
FROM vendedor
WHERE matriculaVendedor = 30;
```

matriculavendedor	nomevendedor	cpf vendedor
30	Pedro Albuquerque	28618161883

1 row(s)

Exercício Prático

5

## Usando subconsulta com UPDATE

```
UPDATE estoque
SET qtdEstocadaLoja = 0
WHERE qtdEstocadaLoja <
(SELECT avg(qtdEstocadaLoja)
FROM estoque);
```

num	romarca	codigomodelo	numero	loja	qtdestocada	loja	valormedio	loja
1	GOL	1	8		10.00			
1	GOL	2	5		5.00			
1	GOL	9	8		10.00			
1	JETTA	9	7		10.00			
1	UP	1	8		10.00			
1	UP	2	9		10.00			
1	UP	7	13		10.00			
1	UP	10	9		10.00			
1	VOYAGE	7	5		5.00			
1	VOYAGE	8	9		10.00			
2	CELTA	3	7		10.00			
2	CORSA	3	5		10.00			
3	KA	4	8		5.00			
5	RIT	5	5		5.00			
6	205	6	2		10.00			
1	GOL	7	0		3.00			
1	GOL	8	0		5.00			
1	GOL	10	0		5.00			
1	JETTA	1	0		5.00			
1	JETTA	2	0		5.00			
1	JETTA	7	0		5.00			
1	JETTA	8	0		5.00			
1	JETTA	10	0		5.00			
1	UP	8	0		5.00			
1	UP	9	0		5.00			
1	TIGUAN	1	0		5.00			
1	TIGUAN	2	0		5.00			
1	TIGUAN	7	0		5.00			
1	TIGUAN	8	0		5.00			
1	TIGUAN	9	0		5.00			
1	TIGUAN	10	0		5.00			
1	VOYAGE	1	0		5.00			
1	VOYAGE	2	0		5.00			
1	VOYAGE	9	0		5.00			
1	VOYAGE	10	0		5.00			
2	CIVIC	3	0		5.00			
3	FIESTA	4	0		5.00			
3	FOCUS	4	0		5.00			
5	CIVIC	5	0		5.00			
6	307	6	0		5.00			

40 row(s)

Exercício Prático

6

## Excluindo linhas com DELETE

► Sintaxe:

```
DELETE FROM nome_da_tabela
[WHERE condição];
```

```
DELETE FROM estoque;
```

Não executar!!!

7

## Usando subconsulta com DELETE

```
DELETE FROM estoque
WHERE qtdEstocadaLoja <
(SELECT avg(qtdEstocadaLoja)
FROM estoque);
```

```
SELECT * from estoque;
```

num	marca	codigo	modelo	num	loja	qtd	estocada	loja	val	med	loja
1	GOL			1		8			10.00		
1	GOL			2		5			5.00		
1	GOL			9		8			10.00		
1	JETTA			9		7			10.00		
1	UP			1		8			10.00		
1	UP			2		9			10.00		
1	UP			7		13			10.00		
1	UP			10		9			10.00		
1	VOYAGE			7		5			5.00		
1	VOYAGE			8		9			10.00		
2	CELTA			3		7			10.00		
2	CORSA			3		5			10.00		
3	KA			4		8			5.00		
5	FIT			5		5			5.00		
6	206			6		8			10.00		

15 row(s)

Exercício Prático

8



## Removendo todas as linhas – Truncate

**TRUNCATE table nome\_da\_tabela;**

**TRUNCATE table estoque;**

**SELECT \* from estoque;**

No rows found.  
Total runtime: 0.610 ms  
SQL executed.

Exercício Prático

9

## DELETE X TRUNCATE

- ▶ Ambos removem os registros (dados).
- ▶ Delete
  - Comando DML (*Data Manipulation Language*)
  - Mais lento
    - Registra no *log* cada linha excluída
  - Aceita cláusula WHERE (condição de seleção)
- ▶ Truncate
  - Comando DDL (*Data Definition Language*)
  - Mais rápido
    - Registra no *log* apenas a remoção de todos os registros na tabela e não cada linha
  - Não aceita cláusula WHERE
    - Remove todas as linhas
  - Truncate X Drop table
    - Truncate elimina somente os dados, deixando a tabela vazia
    - Drop table: elimina dados e tabela

10



11

## Junção

- ▶ A junção (*join*) é uma consulta que recupera linhas de uma ou mais tabelas ou visões.
- ▶ A operação de *join* permite que uma ou mais tabelas se relacionem.
- ▶ É executada sempre que várias tabelas aparecem na cláusula FROM.
- ▶ Se as tabelas têm colunas com nomes em comum e que estão sendo referenciados na consulta, as colunas têm que ser qualificadas através do nome da tabela para evitar ambiguidade.
- ▶ Alguns tipos:
  - Equijoin
  - Inner join
  - Self join
  - Produto Cartesiano
  - Outer join

12

## Equijoin

- É uma junção que possui uma condição na cláusula WHERE que compara colunas das tabelas envolvidas e combina pares de linhas, cada uma pertencente a uma tabela, para as quais a condição avaliada seja verdadeira (TRUE).
- Sintaxe:

```
SELECT <lista_de_colunas>
FROM tabela1, tabela2, ..., tabelan
WHERE tabela1.coluna1 = tabela2.coluna2
  and ...
      Tabela2.coluna3 = tabelan.colunan
```

13

## Junção de duas tabelas

```
SELECT nomeVendedor, V.matriculaVendedor
FROM Carro C, Vendedor V
WHERE C.matriculaVendedor = V.matriculaVendedor
```

Equação de Junção

84 linhas

nomevendedor	matriculavendedor
JOÃO DA SILVA	1
JOSE RUBEM FONSECA	23
BERNARDO ELIS FLEURY DE CAMPOS CURADO	14
JOÃO SIMÕES LOPES NETO	7
AFONSO HENRIQUES DE LIMA BARRETO	5
JOAQUIM MARIA MACHADO DE ASSIS	3
ANTÔNIO CASTILHO DE ALCÂNTARA MACHADO	9
OSMAN JOSÉ LINS	16
JÚLIA VALENTIM DA SILVEIRA LOPES DE ALMEIDA	6
ANÍBAL MONTEIRO MACHADO	12
DINAH SILVEIRA DE QUEIROZ	24
GRACILIANO RAMOS DE OLIVEIRA	10
LYGIA FAGUNDES TELLES	28
SAMUEL RAWET	18
PEDRO DA SILVA	2
RUBEM BRAGA	19
JOÃO PAULO EMÍLIO CRISTÓVÃO DOS SANTOS	4
EDDY JOSÉ DA CRUZ	11
JOAQUIM MARIA MACHADO DE ASSIS	3
ERICO LOPES VERÍSSIMO	21
JOSE RUBEM FONSECA	23
FERNANDO TAVARES SABINO	29
RACHEL DE QUEIROZ	20
ANTÔNIO CASTILHO DE ALCÂNTARA MACHADO	9
OSMAN JOSÉ LINS	16
JÚLIA VALENTIM DA SILVEIRA LOPES DE ALMEIDA	6
JOÃO DA SILVA	1
LYGIA FAGUNDES TELLES	28
BERNARDO ELIS FLEURY DE CAMPOS CURADO	14
PEDRO DA SILVA	2
JOSE J. VEIGA	1
JOÃO PAULO EMÍLIO CRISTÓVÃO DOS SANTOS	4
JOAQUIM MARIA MACHADO DE ASSIS	3
GRACILIANO RAMOS DE OLIVEIRA	10
JOSE RUBEM FONSECA	23
SAMUEL RAWET	18
ORIGENES LESSA	27

Exercício Prático

14

## Junção de três tabelas

```

SELECT nomeFantasiaLoja, nomeVendedor, placaCarro
FROM loja l, carro c, vendedor v
WHERE l.numeroLoja = v.numeroLoja and
v.matriculavendedor = c.matriculavendedor;

SELECT nomeFantasiaLoja, nomeVendedor, placaCarro
FROM loja l
INNER JOIN vendedor v
ON l.numeroLoja = v.numeroLoja
INNER JOIN carro c
ON v.matriculavendedor = c.matriculavendedor;
        
```

nomeFantasiaLoja	nomevendedor	placacarro
DISTAC	JOÃO DA SILVA	ABC0001
DISTAC	JOSE RUBEM FONSECA	ABC0002
Reauto	BERNARDO ELIS FLEURY DE CAMPOS CURADO	ABC0003
Brasileira	JOÃO SIMÕES LOPES NETO	ABC0004
Green Automóveis	AFONSO HENRIQUES DE LIMA BARRETO	ABC0005
SIMCAUTO	JOAQUIM MARIA MACHADO DE ASSIS	ABC0006
SIMCAUTO	ANTÔNIO CASTILHO DE ALCÂNTARA MACHADO	ABC0007
SIMCAUTO	OSMAN JOSE LINS	ABC0008
BARRAFOR	JÚLIA VALENTIM DA SILVEIRA LOPES DE ALMEIDA	ABC0009
BARRAFOR	ANÍBAL MONTEIRO MACHADO	ABC0010
BARRAFOR	DINAH SILVEIRA DE QUEIROZ	DEF0001
DISTAC	GRACILIANO RAMOS DE OLIVEIRA	DEF0002
DISTAC	LYGIA FAGUNDES TELLES	DEF0003
Reauto	SAMUEL RAWET	DEF0004
Rio Tokio	PEDRO DA SILVA	DEF0005
Rio Tokio	RUBEM BRAGA	DEF0006
TRIOMPHE	JOÃO PAULO EMÍLIO CRISTÓVÃO DOS SANTOS	DEF0007
TRIOMPHE	EDDY JOSE DA CRUZ	DEF0008
SIMCAUTO	JOAQUIM MARIA MACHADO DE ASSIS	DEF0009
DISTAC	ERICO LOPES VERÍSSIMO	DEF0010
DISTAC	JOSE RUBEM FONSECA	GHI0101
Reauto	FERNANDO TAVARES SABINO	GHI0202
Brasileira	RACHEL DE QUEIROZ	GHI0303
SIMCAUTO	ANTÔNIO CASTILHO DE ALCÂNTARA MACHADO	GHI0505
SIMCAUTO	OSMAN JOSE LINS	GHI0606
BARRAFOR	JÚLIA VALENTIM DA SILVEIRA LOPES DE ALMEIDA	GHI0808
DISTAC	JOÃO DA SILVA	GHI1111
DISTAC	LYGIA FAGUNDES TELLES	GHI1212
Reauto	BERNARDO ELIS FLEURY DE CAMPOS CURADO	JKL1010
Rio Tokio	PEDRO DA SILVA	JKL2020
TRIOMPHE	JOSE J. VEIGA	JKL3030
TRIOMPHE	JOÃO PAULO EMÍLIO CRISTÓVÃO DOS SANTOS	JKL4040
SIMCAUTO	JOAQUIM MARIA MACHADO DE ASSIS	JKL5050
DISTAC	GRACILIANO RAMOS DE OLIVEIRA	JKL6060
DISTAC	JOSE RUBEM FONSECA	JKL6060
Reauto	SAMUEL RAWET	JKL3032
Brasileira	ORIGENES LESSA	JKL7070
Green Automóveis	MÁRIO RAUL DE MORAIS ANDRADE	JKL8080

84 linhas

Exercício Prático

15

## Inner join

```

SELECT nomeVendedor, V.matriculaVendedor
FROM Carro C INNER JOIN Vendedor V
ON (C.matriculaVendedor = V.matriculaVendedor)
        
```

nomevendedor	matriculavendedor
JOÃO DA SILVA	1
JOSE RUBEM FONSECA	23
BERNARDO ELIS FLEURY DE CAMPOS CURADO	14
JOÃO SIMÕES LOPES NETO	7
AFONSO HENRIQUES DE LIMA BARRETO	5
JOAQUIM MARIA MACHADO DE ASSIS	3
ANTÔNIO CASTILHO DE ALCÂNTARA MACHADO	9
OSMAN JOSE LINS	16
JÚLIA VALENTIM DA SILVEIRA LOPES DE ALMEIDA	6
ANÍBAL MONTEIRO MACHADO	12
DINAH SILVEIRA DE QUEIROZ	24
GRACILIANO RAMOS DE OLIVEIRA	10
LYGIA FAGUNDES TELLES	28
SAMUEL RAWET	18
PEDRO DA SILVA	2
RUBEM BRAGA	19
JOÃO PAULO EMÍLIO CRISTÓVÃO DOS SANTOS	4
EDDY JOSE DA CRUZ	11
JOAQUIM MARIA MACHADO DE ASSIS	3
ERICO LOPES VERÍSSIMO	21
JOSE RUBEM FONSECA	23
FERNANDO TAVARES SABINO	29
RACHEL DE QUEIROZ	20
ANTÔNIO CASTILHO DE ALCÂNTARA MACHADO	9
OSMAN JOSE LINS	16
JÚLIA VALENTIM DA SILVEIRA LOPES DE ALMEIDA	6
JOÃO DA SILVA	1
LYGIA FAGUNDES TELLES	28
BERNARDO ELIS FLEURY DE CAMPOS CURADO	14
PEDRO DA SILVA	2
JOSE J. VEIGA	30
JOÃO PAULO EMÍLIO CRISTÓVÃO DOS SANTOS	4
JOAQUIM MARIA MACHADO DE ASSIS	3
GRACILIANO RAMOS DE OLIVEIRA	10
JOSE RUBEM FONSECA	23
SAMUEL RAWET	18
ORIGENES LESSA	27

84 linhas

Exercício Prático

16



## Junção em auto-relacionamento (Self Join)

- ▶ É uma junção de uma tabela com ela própria.
- ▶ Essa tabela aparece duas vezes na cláusula FROM e é seguida por *alias* a fim de que as colunas possam ser referenciadas sem ambiguidades.

Disciplina (0,1)  
(0,N)

Cada disciplina possui apenas uma disciplina pré-requisito

17

## Junção em auto-relacionamento (Self Join)

```
SELECT * FROM Disciplina;
```

```
SELECT D.Codigo AS Disciplina,
       D.PRE_REQUIS,
       Pre.Nome AS Nome_do_Pre_Req
FROM Disciplina D Disciplina Pre
WHERE D.PRE_REQUIS = Pre.CODIGO;
```

CODIGO	NOME	PRE_REQUIS
Calc_0	Cálculo 0	
Calc_1	Cálculo 1	Calc_0
Fis_1	Física 1	Calc_0
Alg_1	Álgebra 1	Calc_0
Calc_2	Cálculo 2	Calc_1
Calc_3	Cálculo 3	Calc_2
Fis_2	Física 2	Fis_1
Alg_2	Álgebra 2	Alg_1

➔

DISCIPLINA	PRE_REQUIS	NOME_DO_PRE_REQ
Calc_1	Calc_0	Cálculo 0
Fis_1	Calc_0	Cálculo 0
Alg_1	Calc_0	Cálculo 0
Calc_2	Calc_1	Cálculo 1
Calc_3	Calc_2	Cálculo 2
Fis_2	Fis_1	Física 1
Alg_2	Alg_1	Álgebra 1

18

## Intersect X Junção

INTERSECT

↔

INNER JOIN  
+  
DISTINCT

```
SELECT matriculaVendedor
FROM Carro
INTERSECT
SELECT matriculaVendedor
FROM Vendedor
```

```
SELECT distinct V.matriculaVendedor
FROM Carro C, Vendedor V
WHERE C.matriculaVendedor = V.matriculaVendedor
```

```
SELECT distinct V.matriculaVendedor
FROM Carro C
INNER JOIN Vendedor V
ON (C.matriculaVendedor = V.matriculaVendedor)
```

matriculaVendedor
10
3
13
11
16
23
25
15
29
14
7
19
17
4
2
20
27
12
24
21
5
1
18
22
6
28
9

27 linhas

19

## Produto Cartesiano

- ▶ É uma junção em que não há uma condição que relacione as tabelas
  - Junção que não possui a cláusula WHERE
- ▶ Combina cada linha de uma tabela com todas as linhas da outra tabela
- ▶ Sempre gera um resultado com muitas linhas e que raramente é útil.
  - No exemplo a seguir, não faz sentido dizer que um vendedor pertence a todas as lojas. O próprio modelo restringe essa afirmação.

```
SELECT L.numeroLoja as Loja, nomeVendedor as Vendedor
FROM Loja L, Vendedor V;
```

Exercício Prático

20

## Outer Join – Definição

- ▶ Estende o resultado do *join* simples
- ▶ Retorna:
  - Todas as linhas que satisfaçam à condição de join e
  - Mais todas as outras linhas de uma tabela para as quais não existam linhas correlacionadas (através da condição de *join*) na outra tabela.
- ▶ Para um *outer join* entre as tabelas A e B que retorne TODAS as linhas de A, aplica-se o operador de *outer join* (+) nas colunas de B, na condição de *join*.

21

## Exemplo – Outer Join

### LEFT OUTER JOIN

```
SELECT M.codigoModelo AS Modelo,
       qtdEstocadaLoja QTD
FROM estoque E
left outer join Modelo M on
(E.numeroMarca = M.numeroMarca and
E.codigoModelo = M.codigoModelo);
```

modelo	qtd
GOL	10
JETTA	5

2 row(s)

### RIGHT OUTER JOIN

```
SELECT M.codigoModelo AS Modelo,
       qtdEstocadaLoja QTD
FROM estoque E
right outer join Modelo M on
(E.numeroMarca = M.numeroMarca and
E.codigoModelo = M.codigoModelo);
```

modelo	qtd
GOL	10
JETTA	5
UP	NULL
TIGUAN	NULL
VOYAGE	NULL
CELTA	NULL
CORSA	NULL
ONIX	NULL
FIESTA	NULL
FOCUS	NULL
KA	NULL
UNO	NULL
PALIO	NULL
PUNTO	NULL
CIVIC	NULL
FIT	NULL
206	NULL
307	NULL
T6	NULL

19 row(s)

Exercício Prático

22

## Outer Join – Características

- ▶ O operador (+) só pode:
  - Aparecer na cláusula WHERE.
  - Ser aplicado a colunas de tabela ou visão e, não, a expressões.
- ▶ Se várias condições de junção são usadas para juntar 2 tabelas, o operador (+) tem que ser usado em todas estas condições.
- ▶ Uma condição contendo o operador (+) não pode:
  - Ser combinada com outra condição usando o operador OR.
  - Usar o operador IN para comparar a coluna marcada com o operador (+) com outra expressão.
  - Comparar a coluna marcada com o operador (+) com uma *subquery*.

23

## Consulta entre esquemas

- ▶ Criar uma nova tabela no novo esquema (meuesquema1 ou meuesquema2)

Servidor?: PostgreSQL (localhost:5432:allow) Banco de dados?: automovel

Diretório de pesquisa do esquema? meuesquema1

```
create table aluno
(
    nome text not null
)
```


- ▶ Inserir 5 linhas

Diretório de pesquisa do esquema?: meuesquema1

```
insert into aluno values ('Carolina');
insert into aluno values ('Dennis');
insert into aluno values ('Igor');
insert into aluno values ('Karran');
insert into aluno values ('Raphael');
```

24





## Consulta entre esquemas

- Referenciando uma tabela no novo esquema
 

schema.table


database.schema.table
- Consulta cruzando esquemas
 

Diretório de pesquisa do esquema: public  


```
select nomecomprador from public.carro
union
select nome from meuesquema1.aluno
```

nomecomprador
NULL
Raphael
Bille August
Whoopi Goldberg
Jane Seymour
Dennis
Igor
Karran
Nicole Kidman
Carolina

25



INSTITUTO MATEMÁTICA E ESTATÍSTICA



# Algumas Funções

Ana Carolina Almeida

Material adaptado das notas de aula do prof. Sérgio Côrtes (Puc-Rio)

26

## Funções RTRIM e LTRIM

**SELECT LTRIM(' exemplo');**

Caracter não informado = Padrão é em branco. Retira os "brancos" à esquerda

**SELECT LTRIM('xyxyXxy exemplo', 'xy');**

Ao achar o primeiro caracter que não existe à esquerda ("y"), ele pára.

**SELECT RTRIM('meu exemplo ');**

**SELECT RTRIM('exemplo xyxXxyxy', 'xy');**

**Exercício Prático**

27

## Funções LPAD e RPAD

**SELECT LPAD('Meu exemplo', 25);**

**SELECT LPAD('Meu exemplo', 25, '\*:');**

**SELECT RPAD('Meu exemplo', 25);**

**SELECT RPAD('Meu exemplo', 25, ':-');**

**Exercício Prático**

28

## Funções UPPER, LOWER e INITCAP

```
SELECT LOWER('Sr. Pedro MALLAN');
```

```
SELECT UPPER('Sr. Pedro MALLAN');
```

```
SELECT INITCAP('Sr. pedro mALLAN');
```

lower
sr. pedro mallan

1 row(s)

upper
SR. PEDRO MALLAN

1 row(s)

initcap
Sr. Pedro Mallan

1 row(s)

Exercício Prático

29

## Função REPLACE

```
SELECT REPLACE('Gordo, gorro, amargor e vigora', 'gor', 'ra');
```

replace
Gordo, raro, amarra e viraa

1 row(s)

Exercício Prático

30

## Função CASE (PostgreSQL) e DECODE (Oracle e SQL Server)

SELECT estadoLoja, CASE estadoLoja WHEN 'RJ' THEN 'Rio de Janeiro'  
WHEN 'SP' THEN 'Sao Paulo'  
ELSE 'Outros Estados'

END

FROM loja;

estadoLoja	case
RJ	Rio de Janeiro
RJ	Rio de Janeiro
RJ	Rio de Janeiro
RJ	Rio de Janeiro
RJ	Rio de Janeiro
ES	Outros Estados
MG	Outros Estados
SP	Sao Paulo
SP	Sao Paulo
SP	Sao Paulo
RJ	Rio de Janeiro
RJ	Rio de Janeiro

12 row(s)

Exercício Prático

31

## Funções LENGTH e USER

SELECT LENGTH('Meu exemplo');

length
11

1 row(s)

SELECT USER;

current_user
postgres

1 row(s)

Exercício Prático

32