

1



2

Calculando mínimo e máximo

```
SELECT min(valorVenda),
       max(valorVenda)
FROM carro;
```


min	max
14641.00	113475.00


1 row(s)

```
SELECT min(valorVenda) as Preco_Minimo,
       max(valorVenda) as Preco_Maximo
FROM carro;
```

preco_minimo	preco_maximo
14641.00	113475.00

1 row(s)





3

Calculando a média

```
SELECT avg(valorVenda)
FROM carro;
```

avg
42916.500000000000

1 row(s)

```
SELECT avg(valorVenda) as Media_Val_Venda
FROM carro;
```


media_val_venda
42916.500000000000


1 row(s)

```
SELECT round(avg(valorVenda), 2) as Media_Val_Venda
FROM carro;
SELECT to_char(avg(valorVenda), '99999.99') as Media_Val_Venda
FROM carro;
```

media_val_venda
42916.50

1 row(s)





4

Calculando a soma

```
SELECT sum (valorVenda)
FROM carro
WHERE dtVenda is not null;
```

sum
2918322.00

1 row(s)

```
SELECT sum (valorVenda) as Faturamento_Total
FROM carro
WHERE dtVenda is not null;
```

faturamento_total
2918322.00

1 row(s)

Exercício Prático

5

Conjunto distintos de valores

```
SELECT estadoLoja
FROM loja;
```

estadoLoja
RJ
RJ
RJ
RJ
ES
MG
SP
SP
SP

10 row(s)

```
SELECT distinct(estadoLoja)
FROM loja;
```

estadoLoja
RJ
SP
ES
MG

4 row(s)

Exercício Prático

6

A cláusula GROUP BY

```
SELECT <nome das colunas ou *>  
FROM <nome da tabela>  
[into <nome da nova tabela>]  
[WHERE <restrições>]  
[GROUP BY <nome das colunas>]  
[HAVING <nome das colunas>]  
[ORDER BY <nome das colunas ou número da coluna>]
```

7

O uso de GROUP BY

```
SELECT estadoLoja  
FROM loja  
GROUP BY estadoLoja;
```


estadoloja
RJ
SP
ES
MG

4 row(s)

Exercício Prático

8

GROUP BY e COUNT



```
SELECT estadoLoja, count(*) as Numero_linhas
FROM loja
GROUP BY estadoLoja;
```

estadoLoja	numero_linhas
RJ	5
SP	3
ES	1
MG	1


4 row(s)

Possibilita o uso de função de agregação

Exercício Prático

9

GROUP BY e COUNT



```
SELECT codigoModelo, count(numeroLoja)
FROM carro
GROUP BY codigoModelo;
```

codigoModelo	count
CORSA	5
ONIX	6
CELTA	9
206	5
FIESTA	5
KA	5
CIVIC	5
VOYAGE	13
JETTA	6
307	5
FIT	5
FOCUS	5
GOL	8
UP	6
TIGUAN	11

15 row(s)

Não se esqueça que as lojas se repetem na relação 'Carro'

Exercício Prático

10

GROUP BY e COUNT

```
SELECT codigoModelo, count(nomeComprador)
FROM carro
GROUP BY codigoModelo;
```

codigoModelo	count
CORSA	3
ONIX	3
CELTA	7
206	4
FIESTA	3
KA	2
CIVIC	4
VOYAGE	10
JETTA	4
307	4
FIT	3
FOCUS	2
GOL	6
UP	5
TIGUAN	8

15 row(s)

Exercício Prático

11

GROUP BY e HAVING

```
SELECT codigoModelo, count(nomeComprador)
FROM carro
GROUP BY codigoModelo
HAVING count(nomeComprador) > 5;
```

codigoModelo	count
CELTA	7
VOYAGE	10
GOL	6
TIGUAN	8

4 row(s)

FUNÇÕES DE AGREGAÇÃO NÃO PODEM SER USADAS NA CLÁUSULA WHERE, EXCETO SE FOREM EM SUBCONSULTAS!

Exercício Prático

12

Qualificador de nome

- ▶ Consiste no nome da tabela seguido de um ponto e o nome da coluna na tabela

- Exemplo:

- Carro.matriculaVendedor

- ▶ Os qualificadores de nomes são utilizados para efetivar junções (joins)

- Exemplo:

- Juntar a tabela VENDEDOR com CARRO.

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

Exercício Prático

13

Uso de alias (sinônimos)

- ▶ Para não escrevermos todo o nome da tabela nos qualificadores, podemos usar “alias”. A definição do mesmo ocorre na cláusula FROM e pode ser usado nas cláusulas WHERE, ORDER BY, GROUP BY, HAVING, SELECT etc.

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

Exercício Prático

Obs.: Oracle não reconhece alias na cláusula GROUP BY.

14

Como usar subconsultas (*subquery*)

- ▶ Subquery é definida como o comando SELECT que aparece em outro comando de manipulação de dados como o próprio SELECT, UPDATE, INSERT ou DELETE.
- ▶ Na maioria dos casos, no comando SELECT, a *subquery* é parte da condição WHERE.
- ▶ Subquery
 - Deve estar entre parênteses
 - O número de linhas de retorno depende da lógica da consulta
 - O número de colunas de retorno também depende da lógica da consulta
 - O comando ORDER BY não é usado em *subqueries*

15

Exemplo de subconsulta

- ▶ Quais os carros que foram vendidos abaixo da média dos preços de vendas?

```
SELECT avg(valorVenda)
FROM carro;
```

avg
42916.500000000000

1 row(s)

```
SELECT placaCarro, valorVenda
FROM carro
WHERE valorVenda <
      (SELECT avg(valorVenda) FROM carro);
```

placaCarro	valorVenda
ABC0001	18525.00
ABC0002	14641.00
ABC0003	24587.00
ABC0005	15000.00
ABC0006	20689.00
ABC0007	28475.00
ABC0008	28475.00
ABC0009	31248.00
ABC0010	42475.00
DEF0001	15018.00
DEF0002	15000.00
DEF0004	15000.00
DEF0007	34758.00
DEF0008	33704.00
DEF0009	26457.00
DEF0010	39171.00
GHI0202	24587.00
GHI0505	22143.00
GHI0606	35233.00
GHI0808	41478.00
GHI1111	15000.00
JKL1010	15000.00
JKL3030	38760.00
JHL3031	26457.00
JKL5050	35187.00
JHL3032	24587.00
JKL8080	15000.00
JHL3033	20689.00
JHL3035	19034.00
MNO1414	28475.00
MNO1616	38760.00
MNO1717	24758.00
MNO1818	20000.00
MNO1919	27315.00
JHL3039	36901.00
MNO2121	33106.00
MNO2222	33106.00
JHL3040	30148.00
JHL3041	21457.00
PQR0111	39171.00
PQR0333	15000.00
PQR0666	22426.00
PQR0888	28014.00
PQR1221	27315.00
PQR2220	18247.00

45 row(s)

Exercício Prático

16

Usando o operador EXISTS

▶ Testa se o resultado de uma determinada consulta é ou não vazio.

```
SELECT numeroMarca, codigoModelo
FROM Modelo Mo
WHERE exists
(SELECT *
 FROM Estoque E
 WHERE Mo.numeroMarca = E.NumeroMarca
 and Mo.codigoModelo = E.codigoModelo);
```

```
SELECT numeroMarca, codigoModelo
FROM Modelo Mo
WHERE not exists
(SELECT *
 FROM Estoque E
 WHERE Mo.numeroMarca = E.NumeroMarca
 and Mo.codigoModelo = E.codigoModelo);
```

numeromarca	codigomodelo
1	GOL
1	JETTA
1	UP
1	TIGUAN
1	VOYAGE
2	CELTA
2	CORSA
2	ONIX
3	FIESTA
3	FOCUS
3	KA
5	CIVIC
5	FIT
6	206
6	307

15 row(s)

numeromarca	codigomodelo
4	UNO
4	PALIO
4	PUNTO
7	T6

4 row(s)

17

EXISTS x IN

```
SELECT estadoloja, nomefantasialoja
FROM loja
WHERE estadoloja in ('RJ', 'SP');
```

10
linhas

```
SELECT estadoloja, nomefantasialoja
FROM loja I1
WHERE exists
(SELECT *
 FROM loja I2
 WHERE (estadoloja = 'RJ' or estadoloja = 'SP');
```

12 linhas
(TODAS)

Testa se o resultado de uma determinada consulta é ou não vazio.

```
SELECT estadoloja, nomefantasialoja
FROM loja I1
WHERE exists
(SELECT *
 FROM loja I2
 WHERE (estadoloja = 'RJ' or estadoloja = 'SP')
 and I1.numeroloja = I2.numeroloja;
```

10
linhas

18

Operadores de Conjunto – INTERSECT

- ▶ Retorna as linhas que são comuns entre dois conjuntos de linhas
- ▶ Sintaxe:

```
SELECT ...  
INTERSECT  
SELECT ...  
[ORDER BY ...]
```

19

Exemplo – INTERSECT

```
SELECT numeroMarca, codigoModelo  
FROM estoque  
INTERSECT  
SELECT numeroMarca, codigoModelo  
FROM modelo;
```

numeromarca	codigomodelo
1	VOYAGE
5	FIT
2	ONIX
3	KA
1	GOL
2	CORSA
5	CIVIC
6	307
1	JETTA
3	FIESTA
6	206
1	UP
1	TIGUAN
2	CELTA
3	FOCUS

15 row(s)

Exercício Prático

20

Operadores de Conjunto – UNION

- ▶ Combina linhas de diferentes tabelas
- ▶ O número de colunas de cada cláusula SELECT devem ser iguais
- ▶ Os tipos de dados de cada atributo nas cláusulas SELECT devem ser iguais
- ▶ A cláusula ORDER BY deve usar o número da coluna
- ▶ Sintaxe:

```
SELECT ...  
UNION  
SELECT ...  
[ORDER BY ...]
```

21

Exemplo – UNION

```
SELECT numeroMarca, codigoModelo  
FROM estoque  
UNION  
SELECT numeroMarca, codigoModelo  
FROM modelo;
```

numerosmarca	codigoModelo
3	FIESTA
3	FOCUS
2	CORSA
4	PUNTO
3	KA
6	206
5	FIT
5	CIVIC
1	TIGUAN
2	CELTA
4	UNO
1	JETTA
1	VOYAGE
7	T6
1	UP
2	ONIX
4	PALIO
1	GOL
6	307

19 row(s)

Exercício Prático

22

Exemplo – UNION

```
SELECT nomeFantasiaLoja
FROM loja
UNION
SELECT nomeVendedor
FROM vendedor;
```

nomeFantasiaLoja
Rio Tokio
Green Automóveis
PEDRO DA SILVA
JOAQUIM MARIA MACHADO DE ASSIS
JOSE RUBEM FONSECA
OTTO DE OLIVEIRA LARA RESENDE
DISTAC
AFONSO HENRIQUES DE LIMA BARRETO
ERICO LOPES VERISSIMO
BERNARDO ELIS FLEURY DE CAMPOS CURADO
DINAH SILVEIRA DE QUEIROZ
SIMCAUTO
JÚLIA VALENTIM DA SILVEIRA LOPES DE ALMEIDA
RUBEM BRAGA
Alta VW
BrasWagen
ORIGENES LESSA
JOÃO PAULO EMÍLIO CRISTÓVÃO DOS SANTOS
JOÃO DA SILVA
TRIOMPHE
JOSE BENTO RENATO MONTEIRO LOBATO
JOSE J. VEIGA
SAMUEL RAWET
OSMAN JOSE LINS
EDDY JOSE DA CRUZ
CLARICE LISPECTOR
FERNANDO TAVARES SABINO
ANIBAL MONTEIRO MACHADO
MURILO EUGÊNIO RUBIÃO
BARRAFOR
MÁRIO RAUL DE MORAIS ANDRADE
CARLOS DRUMMOND DE ANDRADE
LYGIA FAGUNDES TELLES
JOÃO SIMÕES LOPES NETO
Reauto
ANTÔNIO CASTILHO DE ALCÂNTARA MACHADO
GRACILIANO RAMOS DE OLIVEIRA
RACHEL DE QUEIROZ

38 row(s)

Exercício Prático

23

Operadores de Conjunto – EXCEPT

```
SELECT numeroMarca, codigoModelo
FROM modelo
EXCEPT
SELECT numeroMarca, codigoModelo
FROM estoque;
```

numeroMarca	codigoModelo
4	PUNTO
4	PALIO
7	T6
4	UNO

4 row(s)

```
SELECT numeroMarca, codigoModelo
FROM estoque
EXCEPT
SELECT numeroMarca, codigoModelo
FROM modelo;
```

No rows found.
Total runtime: 1.437 ms
SQL executed.

Exercício Prático

24

Operadores de Conjunto – Produto Cartesiano

SELECT I.numeroLoja as Loja, nomeVendedor as Vendedor
FROM Loja I, vendedor V;

loja	vendedor
1 JOÃO DA SILVA	
2 JOÃO DA SILVA	
3 JOÃO DA SILVA	
4 JOÃO DA SILVA	
5 JOÃO DA SILVA	
6 JOÃO DA SILVA	
7 JOÃO DA SILVA	
8 JOÃO DA SILVA	
9 JOÃO DA SILVA	
10 JOÃO DA SILVA	
1 PEDRO DA SILVA	
2 PEDRO DA SILVA	
3 PEDRO DA SILVA	
4 PEDRO DA SILVA	
5 PEDRO DA SILVA	
6 PEDRO DA SILVA	
7 PEDRO DA SILVA	
8 PEDRO DA SILVA	
9 PEDRO DA SILVA	
10 PEDRO DA SILVA	
1 JOAQUIM MARIA MACHADO DE ASSIS	
2 JOAQUIM MARIA MACHADO DE ASSIS	
3 JOAQUIM MARIA MACHADO DE ASSIS	
4 JOAQUIM MARIA MACHADO DE ASSIS	
5 JOAQUIM MARIA MACHADO DE ASSIS	
6 JOAQUIM MARIA MACHADO DE ASSIS	
7 JOAQUIM MARIA MACHADO DE ASSIS	
8 JOAQUIM MARIA MACHADO DE ASSIS	
9 JOAQUIM MARIA MACHADO DE ASSIS	
10 JOAQUIM MARIA MACHADO DE ASSIS	
1 JOÃO PAULO EMILIO CRISTÓVÃO DOS SANTOS	
2 JOÃO PAULO EMILIO CRISTÓVÃO DOS SANTOS	
3 JOÃO PAULO EMILIO CRISTÓVÃO DOS SANTOS	
4 JOÃO PAULO EMILIO CRISTÓVÃO DOS SANTOS	
5 JOÃO PAULO EMILIO CRISTÓVÃO DOS SANTOS	
6 JOÃO PAULO EMILIO CRISTÓVÃO DOS SANTOS	
7 JOÃO PAULO EMILIO CRISTÓVÃO DOS SANTOS	
8 JOÃO PAULO EMILIO CRISTÓVÃO DOS SANTOS	
9 JOÃO PAULO EMILIO CRISTÓVÃO DOS SANTOS	
10 JOÃO PAULO EMILIO CRISTÓVÃO DOS SANTOS	

290 linhas

Exercício Prático

25

Operações Relacionais – Seleção

SELECT *
FROM loja
WHERE estadoLoja = 'RJ';

numeroLoja	nomefantasiaLoja	endereçoLoja	bairroLoja	cidadeLoja	estadoLoja	cnpiLoja
1	DISTAC	RUA DAS LARANJEIRAS, 291	LARANJEIRAS	RIO DE JANEIRO	RJ	10101010000101
2	DISTAC	AV. AUTOMÓVEL CLUBE, 1995	VILAR DOS TELES	SÃO JOÃO DE MERITI	RJ	100111010000100
3	SIMCAUTO	R. VOLUNTÁRIOS DA PÁTRIA, 54	BOTAFOGO	RIO DE JANEIRO	RJ	102022220000102
4	BARRAFOR	AV. AYRTON SENNA, 2541	BARRA DA TIJUCA	RIO DE JANEIRO	RJ	102099990000102
5	Rio Tokio	R. Real Grandeza, 400	BOTAFOGO	RIO DE JANEIRO	RJ	132322223000132

5 row(s)

Exercício Prático

26

Operações Relacionais – Projeção

`SELECT nomeFantasiaLoja, estadoLoja
FROM loja;`

nomefantasiaLoja	estadoLoja
DISTAC	RJ
DISTAC	RJ
SIMCAUTO	RJ
BARRAFOR	RJ
Rio Tokio	RJ
TRIOMPHE	ES
Reauto	MG
Brasilwagem	SP
Green Automóveis	SP
Alta VW	SP

10 row(s)

Exercício Prático

27

Operações Relacionais – Subtração

`SELECT numeroMarca, codigoModelo
FROM Modelo Mo
WHERE not exists
(SELECT *
FROM Estoque E
WHERE Mo.numeroMarca = E.NumeroMarca
and Mo.codigoModelo = E.codigoModelo);`

`SELECT numeroMarca, codigoModelo
FROM Modelo
WHERE (numeroMarca, codigoModelo) not in
(SELECT numeroMarca, codigoModelo
FROM Estoque);`


numeromarca	codigomodelo
4	UNO
4	PALIO
4	PUNTO
7	T6

4 row(s)

Exercício Prático

28

Operações Relacionais – Divisão



```

SELECT nomeVendedor
FROM vendedor V
WHERE NOT EXISTS
  (SELECT *
   FROM Modelo M
   WHERE NOT EXISTS
    (SELECT *
     FROM Carro C
     WHERE V.matriculaVendedor = C.matriculaVendedor and
           M.numeroMarca = C.numeroMarca and
           M.codigoModelo = C.codigoModelo));

```

Selecione os
vendedores, onde não
exista modelo que ele
não tenha vendido


Ao final da
aula, teremos
um exemplo!

Exercício Prático

No rows found.
Total runtime: 88.515 ms
SQL executed.

29

Operações Relacionais – Divisão



```

SELECT nomeVendedor
FROM vendedor
WHERE matriculaVendedor IN
  (SELECT matriculaVendedor
   FROM carro
   GROUP BY matriculaVendedor
   HAVING count(distinct(numeroMarca, codigoModelo)) =
    (SELECT count(*) FROM Modelo));

```

Ao final da
aula, teremos
um exemplo!

Exercício Prático

No rows found.
Total runtime: 88.515 ms
SQL executed.

30

Adicionando linhas com o INSERT

► Sintaxe

```
INSERT INTO nome_da_tabela
[(nome_coluna1, ..., nome_colunan)
VALUES
(valor_coluna1, ..., valor_colunan);
```

31

Especificando o nome das colunas

```
INSERT into Loja
(numeroLoja, nomeFantasiaLoja, enderecoLoja, bairroLoja,
cidadeLoja, estadoLoja, CNPJLoja)
VALUES
(11, 'Dive Ford', 'Av. Brasil, 15148', 'Vigario Geral', 'Rio de
Janeiro', 'RJ', 98761578264932);
```

numeroLoja	nomeFantasiaLoja	enderecoLoja	bairroLoja	cidadeLoja	estadoLoja	cnpjLoja
1	DISTAC	RUA DAS LARANJEIRAS, 291	LARANJEIRAS	RIO DE JANEIRO	RJ	10101010000101
2	DISTAC	AV. AUTOMÓVEL CLUBE, 1995	VILAR DOS TELES	SÃO JOÃO DE MERITI	RJ	10011101000100
3	SIMCAUTO	R. VOLUNTÁRIOS DA PÁTRIA, 54	BOTAFOGO	RIO DE JANEIRO	RJ	10202222000102
4	BARRAFOR	AV. AYRTON SENNA, 2541	BARRA DA TIJUCA	RIO DE JANEIRO	RJ	10209999000102
5	Rio Tokio	R. Real Grandeza, 400	BOTAFOGO	RIO DE JANEIRO	RJ	13232222300132
6	TRIOMPHE	Av. Vitória, 2.415	Consolação	VITÓRIA	ES	13442244300142
7	Reauto	Av. Babita Camargos, 1243	Cidade Industrial	Contagem	MG	15252222500152
8	Brasilwagem	Rua Fiandeiras, 375	Vila Olímpia	São Paulo	SP	13636666300136
9	Green Automóveis	Av. Giovanni Gronchi, 4313	Vila Andrade	São Paulo	SP	13737777300172
10	Alta VW	Av. Prof. Abraão de Moraes, 900	Saúde	São Paulo	SP	18282222800182
11	Dive Ford	Av. Brasil, 15148	Vigário Geral	Rio de Janeiro	RJ	98761578264932

11 row(s)

Exercício Prático

32

Não especificando o nome das colunas

INSERT into Loja
VALUES
(12, 'Topfor Ford', 'Av. Americas, 1110', 'Barra da Tijuca', 'Rio de Janeiro', 'RJ', 98461878364932);

numero	loja	nomefantasia	loja	endereco	loja	bairro	loja	cidade	loja	estado	loja	cnpj	loja
1	DISTAC	RUA DAS LARANJEIRAS, 291	LARANJEIRAS	RIO DE JANEIRO	RJ	101010100000101							
2	DISTAC	AV. AUTOMOVEIL CLUBE, 1995	VILAR DOS TELES	SÃO JOÃO DE MERITI	RJ	100111010000100							
3	SIMCAUTO	R. VOLUNTARIOS DA PATRIA, 54	BOTAFOGO	RIO DE JANEIRO	RJ	102022220000102							
4	BARRAFOR	AV. AYRTON SENNA, 2541	BARRA DA TIJUCA	RIO DE JANEIRO	RJ	102099990000102							
5	Rio Tokio	R. Real Grandeza, 400	BOTAFOGO	RIO DE JANEIRO	RJ	132322223000132							
6	TRIOMPHE	Av. Vitória, 2.415	Consolação	VITÓRIA	ES	134422443000142							
7	Reauto	Av. Babita Camargos, 1243	Cidade Industrial	Contagem	MG	152522225000152							
8	Brasílwagen	Rua Fiandeiras, 375	Vila Olimpia	São Paulo	SP	136366663000136							
9	Green Automóveis	Av. Giovanni Gronchi, 4313	Vila Andrade	São Paulo	SP	137377773000172							
10	Alta VW	Av. Prof. Abraão de Moraes, 900	Saúde	São Paulo	SP	162822228000182							
11	Dive Ford	Av. Brasil, 15148	Vigário Geral	Rio de Janeiro	RJ	98761578264932							
12	Topfor Ford	Av. Americas, 1110	Barra da Tijuca	Rio de Janeiro	RJ	98461878364932							

12 row(s)

Exercício Prático

33

Incluindo datas com o formato padrão

INSERT into carro
VALUES
('BBB1234', '08/07/2015', 1574268950, 'GASOLINA', 2015, 2015, 'PRATA', 'PASSAGEIRO', 1, 'GOL', 3, 'UERJ', 'Rua Sao Francisco Xavier, 524', '08/07/2015', 35000, 9);

Exercício Prático

34

Utilizando pseudocolunas do usuário e data do sistema

```
INSERT into carro
VALUES
('BBB5678', '08/07/2015', 1874268950, 'GASOLINA', 2015, 2015,
'PRATA', 'PASSAGEIRO', 1, 'JETTA', 3, current_user, Rua Sao Francisco
Xavier, 524', current_date, 75000, 9);
```

Exercício Prático

35

Usando expressões

```
INSERT into carro
VALUES
('BBB9012', '08/07/2015', 1874279950, 'GASOLINA', 2015, 2015,
'PRATA', 'PASSAGEIRO', 1, 'UP', 3, 'UERJ', 'Rua Sao Francisco Xavier,
524', '08/07/2015', 2 * 15000 + 100, 9);
```

Exercício Prático

36

Usando subconsulta com INSERT

```

INSERT INTO <nome_da_tabela>
(<nome_das_colunas>)
SELECT <nome das colunas>
FROM <nome_da_tabela>
WHERE <condição>;

```

matriculaVendedor	nomeVendedor	cpfVendedor	numeroLoja
1	SILVA	10507040201	1
2	A SILVA	10508050701	5
3	MARIA MACHADO DE ASSIS	30508050703	3
4	JOÃO EMÍLIO CRISTOVÃO DOS SANTOS	60508050706	6
5	HENRIQUES DE LIMA BARRETO	90508050709	9
6	LENTIM DA SILVEIRA LOPES DE ALMEIDA	40508050704	4
7	MÔES LOPES NETO	80508050708	8
8	VÍTO RENATO MONTEIRO LOBATO	11518150701	10
9	CASTILHO DE ALCÂNTARA MACHADO	10503050301	3
10	GRACIANO RAMOS DE OLIVEIRA	10108010701	1
11	EDDY JOSÉ DA CRUZ	10608060701	6
12	ANÍBAL MONTEIRO MACHADO	10408040701	4
13	MÁRIO RAUL DE MORAIS ANDRADE	10908090701	9
14	JOÃO ELIS FLEURY DE CAMPOS CURADO	10708070701	7
15	DRUMMOND DE ANDRADE	10508051711	10
16	OSÉ LINS	10308030701	3
17	SUGÊNIO RUBIÃO	10509050901	9
18	RAWET	70508050707	7
19	FRAGA	15508055701	5
20	DE QUEIROZ	10808080701	8
21	JPES VERÍSSIMO	10501050101	1
22	TEIGA	10506050601	6
23	HEM FONSECA	12508250721	2
24	OLIVEIRA DE QUEIROZ	10504050401	4
25	CLARICE LISPECTOR	11508051701	10
26	OTTO DE OLIVEIRA LARA RESENDE	12502050701	2
27	ORIGENES LESSA	10588058801	8
28	LIVIA FACUNDES TELLES	10508250721	2
29	FERNANDO TAVARES SABINO	17507050771	7
30	FERNANDO TAVARES SABINO	17507050772	7

```

INSERT into vendedor (matriculaVendedor,
nomeVendedor, CPFVendedor, numeroLoja)
SELECT matriculaVendedor + 1, nomeVendedor,
CPFVendedor + 1, numeroLoja
FROM vendedor
WHERE matriculaVendedor = 29;

```

30 row(s)

37

Retornando para nossa divisão...

► Inserção de demais linhas

```

INSERT into carro
VALUES
('BBB5679', '08/07/2015', 1874268959, 'GASOLINA', 2015, 2015, 'PRATA', 'PASSAGEIRO', 1, 'TIGUAN', 3, 'UERJ', 'Rua Sao Francisco Xavier, 524', '08/07/2015', 175000, 9);
INSERT into carro
VALUES
('BBB5688', '08/07/2015', 1874268958, 'GASOLINA', 2015, 2015, 'PRATA', 'PASSAGEIRO', 1, 'VOYAGE', 3, 'UERJ', 'Rua Sao Francisco Xavier, 524', '08/07/2015', 55000, 9);
INSERT into carro
VALUES
('BBB5698', '08/07/2015', 1874268957, 'GASOLINA', 2015, 2015, 'PRATA', 'PASSAGEIRO', 3, 'FIESTA', 3, 'UERJ', 'Rua Sao Francisco Xavier, 524', '08/07/2015', 33000, 9);
INSERT into carro
VALUES
('BBB5608', '08/07/2015', 1874268956, 'GASOLINA', 2015, 2015, 'PRATA', 'PASSAGEIRO', 3, 'FOCUS', 3, 'UERJ', 'Rua Sao Francisco Xavier, 524', '08/07/2015', 73000, 9);
INSERT into carro
VALUES
('BBB5978', '08/07/2015', 1874268955, 'GASOLINA', 2015, 2015, 'PRATA', 'PASSAGEIRO', 3, 'KA', 3, 'UERJ', 'Rua Sao Francisco Xavier, 524', '08/07/2015', 28000, 9);
INSERT into carro
VALUES
('BBB5878', '08/07/2015', 1874268954, 'GASOLINA', 2015, 2015, 'PRATA', 'PASSAGEIRO', 4, 'UNO', 3, 'UERJ', 'Rua Sao Francisco Xavier, 524', '08/07/2015', 26000, 9);
INSERT into carro
VALUES
('BBB5778', '08/07/2015', 1874268953, 'GASOLINA', 2015, 2015, 'PRATA', 'PASSAGEIRO', 4, 'PALIO', 3, 'UERJ', 'Rua Sao Francisco Xavier, 524', '08/07/2015', 38000, 9);
INSERT into carro
VALUES
('BBB5078', '08/07/2015', 1874268952, 'GASOLINA', 2015, 2015, 'PRATA', 'PASSAGEIRO', 4, 'PUNTO', 3, 'UERJ', 'Rua Sao Francisco Xavier, 524', '08/07/2015', 85000, 9);
INSERT into carro
VALUES
('BBB6678', '08/07/2015', 1874268951, 'GASOLINA', 2015, 2015, 'PRATA', 'PASSAGEIRO', 5, 'CIVIC', 3, 'UERJ', 'Rua Sao Francisco Xavier, 524', '08/07/2015', 155000, 9);
INSERT into carro
VALUES
('BBB7678', '08/07/2015', 1874268960, 'GASOLINA', 2015, 2015, 'PRATA', 'PASSAGEIRO', 5, 'FIT', 3, 'UERJ', 'Rua Sao Francisco Xavier, 524', '08/07/2015', 45000, 9);
INSERT into carro
VALUES
('BBB8678', '08/07/2015', 1874268970, 'GASOLINA', 2015, 2015, 'PRATA', 'PASSAGEIRO', 6, '206', 3, 'UERJ', 'Rua Sao Francisco Xavier, 524', '08/07/2015', 34000, 9);
INSERT into carro
VALUES
('BBB9678', '08/07/2015', 1874268980, 'GASOLINA', 2015, 2015, 'PRATA', 'PASSAGEIRO', 6, '307', 3, 'UERJ', 'Rua Sao Francisco Xavier, 524', '08/07/2015', 68000, 9);
INSERT into carro
VALUES
('BBB4678', '08/07/2015', 1874268990, 'GASOLINA', 2015, 2015, 'PRATA', 'PASSAGEIRO', 7, 'T6', 3, 'UERJ', 'Rua Sao Francisco Xavier, 524', '08/07/2015', 69000, 9);

```

matriculaVendedor	nomeVendedor	cpfVendedor	numeroLoja
1	SILVA	10507040201	1
2	A SILVA	10508050701	5
3	MARIA MACHADO DE ASSIS	30508050703	3
4	JOÃO EMÍLIO CRISTOVÃO DOS SANTOS	60508050706	6
5	HENRIQUES DE LIMA BARRETO	90508050709	9
6	LENTIM DA SILVEIRA LOPES DE ALMEIDA	40508050704	4
7	MÔES LOPES NETO	80508050708	8
8	VÍTO RENATO MONTEIRO LOBATO	11518150701	10
9	CASTILHO DE ALCÂNTARA MACHADO	10503050301	3
10	GRACIANO RAMOS DE OLIVEIRA	10108010701	1
11	EDDY JOSÉ DA CRUZ	10608060701	6
12	ANÍBAL MONTEIRO MACHADO	10408040701	4
13	MÁRIO RAUL DE MORAIS ANDRADE	10908090701	9
14	JOÃO ELIS FLEURY DE CAMPOS CURADO	10708070701	7
15	DRUMMOND DE ANDRADE	10508051711	10
16	OSÉ LINS	10308030701	3
17	SUGÊNIO RUBIÃO	10509050901	9
18	RAWET	70508050707	7
19	FRAGA	15508055701	5
20	DE QUEIROZ	10808080701	8
21	JPES VERÍSSIMO	10501050101	1
22	TEIGA	10506050601	6
23	HEM FONSECA	12508250721	2
24	OLIVEIRA DE QUEIROZ	10504050401	4
25	CLARICE LISPECTOR	11508051701	10
26	OTTO DE OLIVEIRA LARA RESENDE	12502050701	2
27	ORIGENES LESSA	10588058801	8
28	LIVIA FACUNDES TELLES	10508250721	2
29	FERNANDO TAVARES SABINO	17507050771	7
30	FERNANDO TAVARES SABINO	17507050772	7

38

Retornando para nossa divisão...

► Novo resultado da consulta

```
SELECT nomeVendedor
FROM vendedor V
WHERE NOT EXISTS
  (SELECT *
   FROM Modelo M
   WHERE NOT EXISTS
     (SELECT *
      FROM Carro C
      WHERE V.matriculaVendedor = C.matriculaVendedor and
            M.numeroMarca = C.numeroMarca and
            M.codigoModelo = C.codigoModelo));
```

nomevendedor
ANTÔNIO CASTILHO DE ALCÂNTARA MACHADO

1 row(s)

Exercício Prático