

SQL

EXISTS e operações de conjunto

Carina Friedrich Dorneles
dorneles@inf.ufsc.br

Banco de Dados

Subconsultas com EXISTS

- ▶ O operador EXISTS verifica a existência de linha retornadas por uma subconsulta:

```
SELECT nome
FROM produto p
WHERE EXISTS (SELECT *
               FROM produtoVendido pv
               WHERE p.codigo = pv.codprod)
```

```
SELECT nome
FROM produto p
WHERE NOT EXISTS (SELECT *
                  FROM produtoVendido pv
                  WHERE p.codigo = pv.codprod)
```



Subconsultas com EXISTS

- ▶ O operador EXISTS verifica a existência de linha retornadas por uma subconsulta:

Devolve ***todas*** as tuplas da consulta interna relacionadas a ***uma*** tupla da externa

```
SELECT nome
FROM produto p
WHERE EXISTS (SELECT *
               FROM produtoVendido pv
               WHERE p.codigo = pv.codprod)
```

```
SELECT nome
FROM produto p
WHERE NOT EXISTS (SELECT *
                  FROM produtoVendido pv
                  WHERE p.codigo = pv.codprod)
```

Exemplos de Subconsultas

► Consultas equivalentes:

```
SELECT c.nome
FROM cidade c
WHERE EXISTS (SELECT cl.codcid
               FROM cliente cl
               WHERE cl.codcid=c.codigo)
```

```
SELECT c.nome
FROM cidade c
WHERE c.codigo IN (SELECT codcid
                  FROM cliente)
```

```
SELECT c.nome
FROM cidade c JOIN cliente cl ON cl.codcid=c.codigo
```

São equivalentes
porque são
correlacionadas
através das PKs e
FKs

Exemplos de Subconsultas

- Consultas equivalentes, sem correlação de PK e FK:

```
SELECT f.nome
FROM funcionario f
WHERE EXISTS (SELECT *
               FROM listaContatos lc
               WHERE f.nome = lc.nome);
```

```
SELECT p.prof_codigo, p.prof_nome
FROM professor p
WHERE p.prof_nome IN (SELECT c.nome
                     FROM contatos c);
```



Exemplos de Subconsultas

► Consultas não equivalentes:

```
SELECT nome
FROM produto
WHERE valorcusto IN (SELECT valorcusto
                      FROM produto
                      WHERE categoria = 'vestuário')
```

```
SELECT nome
FROM produto
WHERE EXISTS (SELECT valorcusto
              FROM produto
              WHERE categoria = 'vestuário')
```



SQL

Operações de conjunto

Carina Friedrich Dorneles
dorneles@inf.ufsc.br

Banco de Dados

Operações de conjunto

- ▶ A linguagem SQL dá suporte a quatro operações sobre conjuntos:
 - ▶ União - UNION
 - ▶ Interseção - INTERSECT
 - ▶ Diferença - EXCEPT
 - ▶ Produto cartesiano - ,



Operações de conjunto

- ▶ A utilização de UNION, INTERSECT, EXCEPT só é válida, se as seguintes diretrizes forem cumpridas
 - ▶ as **tabelas** envolvidas devem ser **compatíveis**.
 - ▶ Duas relações são compatíveis se possuem o mesmo número de atributos e se cada par de atributos correspondentes tem o mesmo domínio
 - ▶ a cláusula **ORDER BY** só deve ser usada sobre o **atributo da primeira consulta**
 - ▶ as cláusulas **GROUP BY** e **HAVING** podem ser usadas apenas dentro de consultas individuais,
 - ▶ elas **não** podem ser usadas para afetar o **conjunto de resultados final**

UNION

- ▶ Combina os resultados de duas ou mais consultas em um único conjunto de resultados, que inclui todas as linhas pertencentes a todas as consultas da união.
- ▶ Sintaxe

CONSULTA

UNION [ALL]

CONSULTA



UNION

- ▶ Exemplo: Nomes dos médicos que também sejam pacientes (o nome aparece nas duas tabelas).
- ▶ Sintaxe

```
SELECT nome  
FROM paciente
```

UNION

```
SELECT nome  
FROM medico
```



UNION

- ▶ Exemplo: Nomes dos médicos que também sejam pacientes (o nome aparece nas duas tabelas).

- ▶ Sintaxe

```
SELECT nome  
FROM paciente  
  
UNION  
  
SELECT nome  
FROM medico
```

nome character varyin
João Carlos
Maria Aparecida
Pedro Antonio
Marcos Vinicius
Carolina Pereira
Antonio Carlos
Mariana Faria
Joaquim José Silva
Joana Darq
Pedro Avars

nome character varyin
Paulo Rangel
Ana Maria
José Paulo O
Carla Ana
Nena Lina
Paulina Tirou
Luara dos Santos

UNION

- ▶ Exemplo: Nomes dos médicos que também sejam pacientes (o nome aparece nas duas tabelas).

- ▶ Sintaxe

SELECT nome

FROM paciente 

UNION

SELECT nome

FROM medico

nome
character varyin
Marcos Vinicius
João Carlos
Joaquim José Silva
Ana Maria
Antonio Carlos
Maria Aparecida
Carla Ana
Paulo Rangel
Joana Darq
Luara dos Santos
Paulina Tirou
José Paulo O
Pedro Antonio
Mariana Faria
Nena Lina
Carolina Pereira
Pedro Avars

UNION

- ▶ Exemplo: Nomes de pacientes e médicos também podem ser armazenados em uma única tabela, eliminando as duplicatas.

Elimina
Duplicatas

- ▶ Sintaxe

```
SELECT nome  
FROM paciente
```

UNION

```
SELECT nome  
FROM medico
```

nome
character varyin
Marcos Vinicius
João Carlos
Joaquim José Silva
Ana Maria
Antonio Carlos
Maria Aparecida
Carla Ana
Paulo Rangel
Joana Darq
Luara dos Santos
Paulina Tirou
José Paulo O
Pedro Antonio
Mariana Faria
Nena Lina
Carolina Pereira
Pedro Avars

UNION

- ▶ Exemplo: Nomes dos médicos que também sejam pacientes (o nome aparece nas duas tabelas).
- ▶ Sintaxe

```
SELECT nome  
FROM paciente  
UNION ALL  
SELECT nome  
FROM medico
```



Mantém
Duplicatas

The diagram consists of a light blue oval with a dark blue border. An arrow points from the text 'UNION ALL' in the SQL query above to the oval. The oval contains the text 'Mantém Duplicatas'.

EXCEPT

- ▶ Retorna qualquer valor (distinto) da consulta à esquerda que não esteja na consulta à direita
- ▶ Sintaxe

CONSULTA

EXCEPT [ALL]

CONSULTA



EXCEPT

- ▶ Exemplo: Código dos pacientes que **não** tiveram consultas
- ▶ Sintaxe

```
SELECT codigo  
FROM paciente  
EXCEPT
```

```
SELECT codpac  
FROM consulta
```



EXCEPT

- ▶ Exemplo: Código dos pacientes que **não** tiveram consultas

- ▶ Sintaxe

```
SELECT codigo  
FROM paciente
```

EXCEPT

```
SELECT codpac  
FROM consulta
```

codigo integer
1
2
3
4
5
6
7
8
9
10

codpac integer
1
3
4
5
6
4
7
3
2
6

EXCEPT

- ▶ Exemplo: Código dos pacientes que não tiveram consultas
- ▶ Sintaxe


```
SELECT codigo
```

```
FROM paciente
```

```
EXCEPT
```

```
SELECT codpac
```

```
FROM consulta
```



codigo integer
8
10
9

INTERSECT

- ▶ Retorna qualquer valor (distinto) retornado pela consulta à esquerda **e** à direita
- ▶ Sintaxe

CONSULTA

INTERSECT [ALL]

CONSULTA



EXCEPT

- ▶ Exemplo: Nomes dos pacientes que **não** tiveram consultas

- ▶ Sintaxe

```
SELECT codigo  
FROM paciente  
INTERSECT  
SELECT codpac  
FROM consulta
```

codigo integer
1
2
3
4
5
6
7
8
9
10

codpac integer
1
3
4
5
6
4
7
3
2
6

EXCEPT

- ▶ Exemplo: Nomes dos pacientes que **não** tiveram consultas
- ▶ Sintaxe


```
SELECT codigo
```

```
FROM paciente
```

```
INTERSECT
```

```
SELECT codpac
```

```
FROM consulta
```



codigo integer
4
1
5
3
6
2
7

Operações de conjunto

- ▶ As consultas envolvidas em um UNION, EXCEPT ou INTERSECT podem ser completas (com todas as cláusulas) e complexas
- ▶ Exemplo:

```
SELECT <atributos>, count(*)  
FROM tabela A JOIN tabela B ON ...  
WHERE atributo > ... AND ...  
GROUP BY atributos
```

UNION

```
SELECT <atributos>, count(*)  
FROM tabela C JOIN tabela B ON ...  
WHERE atributo > ... AND ...  
GROUP BY atributos
```



SQL

exemplos de aninhamentos diversos

Carina Friedrich Dorneles
dorneles@inf.ufsc.br

Banco de Dados

Estrutura geral de uma consulta SQL

```
SELECT <atributo (s) e/ou função(ões)>  
FROM <tabela (s)>  
WHERE <condição (ões) sobre atributo(s)>  
GROUP BY <atributo (s)>  
HAVING <condição (ões) sobre função(ões)>  
ORDER BY <atributo (s)>
```



Ordem de execução no SGBD

1ª. FROM <tabela (s)>

2ª. WHERE <condição (ões) sobre atributo(s)>

3ª. GROUP BY <atributo (s)>

4ª. HAVING <condição (ões) sobre função(ões)>

5ª. ORDER BY <atributo (s)>

6ª. SELECT <atributo (s) e/ou função(ões)>



Subconsulta no SELECT

- Consultas semanticamente equivalentes

Q1:

```
SELECT c.nome, count(v.codclie) as QtdComprada  
FROM cliente c LEFT JOIN venda v ON c.codigo =  
v.codclie  
GROUP BY c.nome  
ORDER BY count(v.codclie) d
```

NOME	QTDCOMPRADA
Juliana Paula	5
Monira Rosa	2
Tamires Ana	2
Linda Regina	2
Claudia Regina	1
Paulinha Vera	1
Claudio Rogério	1
Tatiana Clara	0
Marco Antonio	0
Paulo Marcos	0
Há mais de 10 linhas disponíveis. Aumente o seletor de linhas para ver mais linhas.	

Subconsulta no SELECT

- ▶ Consultas semanticamente equivalentes

Q2:

```
SELECT c.nome, (SELECT count(v.codigo)
FROM venda v
WHERE v.codclie = c.codigo) as
```

QtdComprada

FROM cliente c

ORDER BY QtdComprada des

Esta construção é possível se a consulta mais interna retornar uma única linha como resultado

NOME	QTDCOMPRADA
Juliana Paula	5
Monira Rosa	2
Tamires Ana	2
Linda Regina	2
Claudia Regina	1
Paulinha Vera	1
Claudio Rogério	1
Tatiana Clara	0
Marco Antonio	0
Paulo Marcos	0
Há mais de 10 linhas disponíveis. Aumente o seletor de linhas para ver mais linhas.	

Subconsulta no FROM

- Consultas semanticamente equivalentes

Q3:

SELECT distinct c.nome

FROM cliente c JOIN venda v

ON c.codigo = v.codclie

WHERE v.data BETWEEN '01-10-2000' AND '01-07-2006'

NOME
Tamires Ana
Monira Rosa
Paulinha Vera
Claudio Rogério
Juliana Paula
Linda Regina

Subconsulta no FROM

- Consultas semanticamente equivalentes

Q4:

SELECT distinct c.nome

FROM cliente c JOIN (SELECT codclie

FROM venda

WHERE data BETWEEN '01-10-2000'

AND

'01-07-2006')

ON v2.codclie =c.codigo

NOME
Tamires Ana
Monira Rosa
Paulinha Vera
Claudio Rogério
Juliana Paula
Linda Regina

Subconsulta no WHERE

- Consultas semanticamente equivalentes

Q5:

SELECT distinct c.nome, c.telefone

FROM cliente c JOIN venda v ON c.codigo = v.codclie

NOME	TELEFONE
Claudia Regina	55.3131.0000
Claudio Rogério	51.3131.0000
Juliana Paula	51.3131.0000
Linda Regina	54.0909.9898
Monira Rosa	54.0101.9090
Paulinha Vera	-
Tamires Ana	54.3131.0000

Subconsulta no WHERE

- Consultas semanticamente equivalentes

Q6:

SELECT c.nome, c.telefone

FROM cliente c

WHERE EXISTS (SELECT *

FROM venda v

WHERE c.codigo = v.codclie)

NOME	TELEFONE
Claudia Regina	55.3131.0000
Claudio Rogério	51.3131.0000
Juliana Paula	51.3131.0000
Linda Regina	54.0909.9898
Monira Rosa	54.0101.9090
Paulinha Vera	-
Tamires Ana	54.3131.0000

ORDER BY - usando resultado de subconsulta

Q2:

```
SELECT c.nome, (SELECT count(v.codclie)
                FROM venda v
                WHERE v.codclie =c.codigo) as QtdComprada
FROM cliente c
ORDER BY QtdComprada desc
```

Q1:

```
SELECT c.nome, count(v.codclie) as QtdComprada
FROM cliente c LEFT JOIN venda v ON c.codigo = v.codclie
GROUP BY c.nome
ORDER BY count(v.codclie) desc /*... ou ORDER BY
QtdComprada desc */
```

HAVING - testando resultados de subconsulta

Q7: produtos cuja média vendida seja maior que a média geral

```
SELECT codprod, avg(valor) as MediaProduto
FROM produtovendido
GROUP BY codprod
HAVING avg(valor) > (SELECT avg(v
```

CODPROD	MEDIAPRODUTO
1	104,5
5	136,725
8	80,375

```
FROM produtovendido)
```

[AND avg(valor) > 100]

~~HAVING MediaProduto > (...)~~

CODPROD	MEDIAPRODUTO
1	104,5
5	136,725



INSERT + SELECT

Cliente

codigo integer	nome character var	email character var	telefone character var
1	Juliana Paula	juju@abc.de.br	51.3131.0000
2	Paulinha Vera	pv@abc.de.br	
3	Claudia Regina	cr@abc.de.br	55.3131.0000
4	Marco Antonio	mat@abc.de.br	
5	Claudio Rogério	crog@abc.de.br	51.3131.0000
6	Flavio Augusto	flau@abc.de.br	
7	Monira Rosa	moro@abc.de.br	54.0101.9090
8	Paulo Marcos	pama@abc.de.br	
9	Linda Regina	lire@abc.de.br	54.0909.9898
10	Tamires Ana	tana@abc.de.br	54.3131.0000
11	Luciano Paulo	lupa@abc.de.br	
12	Tatiana Clara	tacl@abc.de.br	
13	Robercleberson	robi@abc.de.br	

Funcionario

codigo integer	nome character var	email character var	dtanasc date	salario numeric(18,3)
1	Claudison	clau@empresa.c	1969-10-10	1000.000
2	Leinaura	lei@empresa.co	1975-10-10	750.000
3	Lobervau	lob@empresa.co	1971-11-10	350.000
4	Norlindson	lind@empresa.c	1974-05-10	400.000
5	Berlindo	be@empresa.cc	1978-06-17	300.000
6	fluvialson	flu@empresa.cc	1980-03-01	470.000
7	Robercleberson	rob@empresa.c	1981-04-23	1500.000
8	Aristideles	ari@empresa.cc	1977-04-28	1240.000
9	Jucemar	juce@empresa.i	1978-04-28	1550.000
15	Luizinha	lui@empresa.co	1968-04-28	2550.000
16	Janira	jan@empresa.co	1971-10-18	1550.000

INSERT + SELECT

Funcionario

codigo integer	nome character var	email character var	dtanasc date	salario numeric(18,3)
1	Claudison	clau@empresa.c	1969-10-10	1000.000
2	Leinaura	lei@empresa.co	1975-10-10	750.000
3	Lobervau	lob@empresa.co	1971-11-10	350.000
4	Norlindson	lind@empresa.c	1974-05-10	400.000
5	Berlindo	be@empresa.cc	1978-06-17	300.000
6	Fluvialson	flu@empresa.cc	1980-03-01	470.000
7	Robercleberson	rob@empresa.c	1981-04-23	1500.000
8	Aristideles	ari@empresa.cc	1977-04-28	1240.000
9	Jucemar	juce@empresa.v	1978-04-28	1550.000
15	Luizinha	lui@empresa.co	1968-04-28	2550.000
16	Janira	jan@empresa.co	1971-10-18	1550.000

Cliente

codigo integer	nome character var	email character var	telefone character var
1	Juliana Paula	juju@abc.de.br	51.3131.0000
2	Paulinha Vera	pv@abc.de.br	
3	Claudia Regina	cr@abc.de.br	55.3131.0000
4	Marco Antonio	mat@abc.de.br	
5	Claudio Rogério	crog@abc.de.br	51.3131.0000
6	Flavio Augusto	flau@abc.de.br	
7	Monira Rosa	moro@abc.de.b	54.0101.9090
8	Paulo Marcos	pama@abc.de.t	
9	Linda Regina	lire@abc.de.br	54.0909.9898
10	Tamires Ana	tana@abc.de.br	54.3131.0000
11	Luciano Paulo	lupa@abc.de.br	
12	Tatiana Clara	tacl@abc.de.br	
13	Robercleberson	robi@abc.de.br	

INSERT into cliente (codigo, nome, email)
SELECT codigo, nome, email
FROM funcionario WHERE codigo >=15



INSERT + SELECT

Funcionario

codigo integer	nome character var	email character var	dtanasc date	salario numeric(18,3)
1	Claudison	clau@empresa.c	1969-10-10	1000.000
2	Leinaura	lei@empresa.co	1975-10-10	750.000
3	Lobervau	lob@empresa.co	1971-11-10	350.000
4	Norlindson	lind@empresa.c	1974-05-10	400.000
5	Berlindo	be@empresa.cc	1978-06-17	300.000
6	Fluvialson	flu@empresa.cc	1980-03-01	470.000
7	Robercleberson	rob@empresa.c	1981-04-23	1500.000
8	Aristideles	ari@empresa.cc	1977-04-28	1240.000
9	Jucemar	juce@empresa.v	1978-04-28	1550.000
15	Luizinha	lui@empresa.co	1968-04-28	2550.000
16	Janira	jan@empresa.co	1971-10-18	1550.000

Cliente

codigo integer	nome character var	email character var	telefone character var
1	Juliana Paula	juju@abc.de.br	51.3131.0000
2	Paulinha Vera	pv@abc.de.br	
3	Claudia Regina	cr@abc.de.br	55.3131.0000
4	Marco Antonio	mat@abc.de.br	
5	Claudio Rogério	crog@abc.de.br	51.3131.0000
6	Flavio Augusto	flau@abc.de.br	
7	Monira Rosa	moro@abc.de.b	54.0101.9090
8	Paulo Marcos	pama@abc.de.t	
9	Linda Regina	lire@abc.de.br	54.0909.9898
10	Tamires Ana	tana@abc.de.br	54.3131.0000
11	Luciano Paulo	lupa@abc.de.br	
12	Tatiana Clara	tacl@abc.de.br	
13	Robercleberson	robi@abc.de.br	

INSERT into cliente (codigo, nome, email)
SELECT codigo, nome, email
FROM funcionario WHERE codigo >

Cliente

codigo integer	nome character var	email character var	telefone character var
1	Juliana Paula	juju@abc.de.br	51.3131.0000
2	Paulinha Vera	pv@abc.de.br	
3	Claudia Regina	cr@abc.de.br	55.3131.0000
4	Marco Antonio	mat@abc.de.br	
5	Claudio Rogério	crog@abc.de.br	51.3131.0000
6	Flavio Augusto	flau@abc.de.br	
7	Monira Rosa	moro@abc.de.b	54.0101.9090
8	Paulo Marcos	pama@abc.de.t	
9	Linda Regina	lire@abc.de.br	54.0909.9898
10	Tamires Ana	tana@abc.de.br	54.3131.0000
11	Luciano Paulo	lupa@abc.de.br	
12	Tatiana Clara	tacl@abc.de.br	
13	Robercleberson	robi@abc.de.br	
15	Luizinha	lui@empresa.co	
16	Janira	jan@empresa.co	



UPDATE + SELECT

Cliente

codigo integer	nome character var	email character var	telefone character var
1	Juliana Paula	juju@abc.de.br	51.3131.0000
2	Paulinha Vera	pv@abc.de.br	
3	Claudia Regina	cr@abc.de.br	55.3131.0000
4	Marco Antonio	mat@abc.de.br	
5	Claudio Rogério	crog@abc.de.br	51.3131.0000
6	Flavio Augusto	flau@abc.de.br	
7	Monira Rosa	moro@abc.de.b	54.0101.9090
8	Paulo Marcos	pama@abc.de.t	
9	Linda Regina	lire@abc.de.br	54.0909.9898
10	Tamires Ana	tana@abc.de.bi	54.3131.0000
11	Luciano Paulo	lupa@abc.de.br	
12	Tatiana Clara	tacl@abc.de.br	
13	Robercleberson	robi@abc.de.br	

Contato

codigo integer	nome character var	email character var	fone character var
1	Juliana Paula	juju@abc.de.br	51.3131.0000
3	Claudia Regina	cr@abc.de.br	55.3131.0000
5	Claudio Rogério	crog@abc.de.br	51.3131.0000
7	Monira Rosa	moro@abc.de.b	54.0101.9090
9	Linda Regina	lire@abc.de.br	54.0909.9898
10	Tamires Ana	tana@abc.de.bi	54.3131.0000
2	Paulinha Vera	pv@abc.de.br	48.5656.5656
4	Marco Antonio	mat@abc.de.br	48.5656.5656
6	Flavio Augusto	flau@abc.de.br	48.5656.5656
8	Paulo Marcos	pama@abc.de.t	48.5656.5656
11	Luciano Paulo	lupa@abc.de.br	48.5656.5656
15	Luizinha	lui@empresa.co	48.5656.5656
16	Janira	jan@empresa.c	48.5656.5656
12	Tatiana Clara	tacl@abc.de.br	48.5655.9999
13	Robercleberson	robi@abc.de.br	48.5655.9999

UPDATE + SELECT

Cliente

codigo integer	nome character var	email character var	telefone character var
1	Juliana Paula	juju@abc.de.br	51.3131.0000
2	Paulinha Vera	pv@abc.de.br	
3	Claudia Regina	cr@abc.de.br	55.3131.0000
4	Marco Antonio	mat@abc.de.br	
5	Claudio Rogério	crog@abc.de.br	51.3131.0000
6	Flavio Augusto	flau@abc.de.br	
7	Monira Rosa	moro@abc.de.b	54.0101.9090
8	Paulo Marcos	pama@abc.de.t	
9	Linda Regina	lire@abc.de.br	54.0909.9898
10	Tamires Ana	tana@abc.de.bi	54.3131.0000
11	Luciano Paulo	lupa@abc.de.br	
12	Tatiana Clara	tacl@abc.de.br	
13	Robercleberson	robi@abc.de.br	

Contato

codigo integer	nome character var	email character var	fone character var
1	Juliana Paula	juju@abc.de.br	51.3131.0000
3	Claudia Regina	cr@abc.de.br	55.3131.0000
5	Claudio Rogério	crog@abc.de.br	51.3131.0000
7	Monira Rosa	moro@abc.de.b	54.0101.9090
9	Linda Regina	lire@abc.de.br	54.0909.9898
10	Tamires Ana	tana@abc.de.bi	54.3131.0000
2	Paulinha Vera	pv@abc.de.br	48.5656.5656
4	Marco Antonio	mat@abc.de.br	48.5656.5656
6	Flavio Augusto	flau@abc.de.br	48.5656.5656
8	Paulo Marcos	pama@abc.de.t	48.5656.5656
11	Luciano Paulo	lupa@abc.de.br	48.5656.5656
15	Luizinha	lui@empresa.co	48.5656.5656
16	Janira	jan@empresa.c	48.5656.5656
12	Tatiana Clara	tacl@abc.de.br	48.5655.9999
13	Robercleberson	robi@abc.de.br	48.5655.9999

UPDATE cliente

SET telefone = (SELECT fone

FROM contato

WHERE codigo =4)

WHERE codigo = 4



UPDATE + SELECT

Cliente

codigo integer	nome character var	email character var	telefone character var
1	Juliana Paula	juju@abc.de.br	51.3131.0000
2	Paulinha Vera	pv@abc.de.br	
3	Claudia Regina	cr@abc.de.br	55.3131.0000
4	Marco Antonio	mat@abc.de.br	
5	Claudio Rogério	crog@abc.de.br	51.3131.0000
6	Flavio Augusto	flau@abc.de.br	
7	Monira Rosa	moro@abc.de.b	54.0101.9090
8	Paulo Marcos	pama@abc.de.t	
9	Linda Regina	lire@abc.de.br	54.0909.9898
10	Tamires Ana	tana@abc.de.bi	54.3131.0000
11	Luciano Paulo	lupa@abc.de.br	
12	Tatiana Clara	tacl@abc.de.br	
13	Robercleberson	robi@abc.de.br	

Contato

codigo integer	nome character var	email character var	fone character var
1	Juliana Paula	juju@abc.de.br	51.3131.0000
3	Claudia Regina	cr@abc.de.br	55.3131.0000
5	Claudio Rogério	crog@abc.de.br	51.3131.0000
7	Monira Rosa	moro@abc.de.b	54.0101.9090
9	Linda Regina	lire@abc.de.br	54.0909.9898
10	Tamires Ana	tana@abc.de.bi	54.3131.0000
2	Paulinha Vera	pv@abc.de.br	48.5656.5656
4	Marco Antonio	mat@abc.de.br	48.5656.5656
6	Flavio Augusto	flau@abc.de.br	48.5656.5656
8	Paulo Marcos	pama@abc.de.t	48.5656.5656
11	Luciano Paulo	lupa@abc.de.br	48.5656.5656
15	Luizinha	lui@empresa.co	48.5656.5656
16	Janira	jan@empresa.co	48.5656.5656
12	Tatiana Clara	tacl@abc.de.br	48.5655.9999
13	Robercleberson	robi@abc.de.br	48.5655.9999

Cliente

codigo integer	nome character var	email character var	telefone character var
1	Juliana Paula	juju@abc.de.br	51.3131.0000
2	Paulinha Vera	pv@abc.de.br	
3	Claudia Regina	cr@abc.de.br	55.3131.0000
5	Claudio Rogério	crog@abc.de.br	51.3131.0000
6	Flavio Augusto	flau@abc.de.br	
7	Monira Rosa	moro@abc.de.b	54.0101.9090
8	Paulo Marcos	pama@abc.de.t	
9	Linda Regina	lire@abc.de.br	54.0909.9898
10	Tamires Ana	tana@abc.de.bi	54.3131.0000
11	Luciano Paulo	lupa@abc.de.br	
12	Tatiana Clara	tacl@abc.de.br	
13	Robercleberson	robi@abc.de.br	
15	Luizinha	lui@empresa.co	
16	Janira	jan@empresa.co	
4	Marco Antonio	mat@abc.de.br	48.5656.5656

UPDATE cliente

SET telefone = (SELECT fone
FROM contato
WHERE codigo = 4)

WHERE codigo = 4



DELETE + SELECT

Cliente

codigo integer	nome character var	email character var	telefone character var
1	Juliana Paula	juju@abc.de.br	51.3131.0000
2	Paulinha Vera	pv@abc.de.br	
3	Claudia Regina	cr@abc.de.br	55.3131.0000
4	Marco Antonio	mat@abc.de.br	
5	Claudio Rogério	crog@abc.de.br	51.3131.0000
6	Flavio Augusto	flau@abc.de.br	
7	Monira Rosa	moro@abc.de.br	54.0101.9090
8	Paulo Marcos	pama@abc.de.br	
9	Linda Regina	lire@abc.de.br	54.0909.9898
10	Tamires Ana	tana@abc.de.br	54.3131.0000
11	Luciano Paulo	lupa@abc.de.br	
12	Tatiana Clara	tacl@abc.de.br	
13	Robercleberson	robi@abc.de.br	
15	Luizinha	lui@empresa.co	
16	Janira	jan@empresa.co	

Funcionario

codigo integer	nome character var	email character var	dtanasc date	salario numeric(18,3)
1	Claudison	clau@empresa.co	1969-10-10	1000.000
2	Leinaura	lei@empresa.co	1975-10-10	750.000
3	Lobervau	lob@empresa.co	1971-11-10	350.000
4	Norlindson	lind@empresa.co	1974-05-10	400.000
5	Berlindo	be@empresa.co	1978-06-17	300.000
6	Fluvialson	flu@empresa.co	1980-03-01	470.000
7	Robercleberson	rob@empresa.co	1981-04-23	1500.000
8	Aristideles	ari@empresa.co	1977-04-28	1240.000
9	Jucemar	juce@empresa.co	1978-04-28	1550.000
15	Luizinha	lui@empresa.co	1968-04-28	2550.000
16	Janira	jan@empresa.co	1971-10-18	1550.000

DELETE + SELECT

Funcionario

codigo integer	nome character var	email character var	dtanasc date	salario numeric(18,3)
1	Claudison	clau@empresa.c	1969-10-10	1000.000
2	Leinaura	lei@empresa.co	1975-10-10	750.000
3	Lobervau	lob@empresa.co	1971-11-10	350.000
4	Norlindson	lind@empresa.c	1974-05-10	400.000
5	Berlindo	be@empresa.cc	1978-06-17	300.000
6	fluvialson	flu@empresa.cc	1980-03-01	470.000
7	Robercleberson	rob@empresa.c	1981-04-23	1500.000
8	Aristideles	ari@empresa.cc	1977-04-28	1240.000
9	Jucemar	juce@empresa.v	1978-04-28	1550.000
15	Luizinha	lui@empresa.co	1968-04-28	2550.000
16	Janira	jan@empresa.co	1971-10-18	1550.000

Cliente

codigo integer	nome character var	email character var	telefone character var
1	Juliana Paula	juju@abc.de.br	51.3131.0000
2	Paulinha Vera	pv@abc.de.br	
3	Claudia Regina	cr@abc.de.br	55.3131.0000
4	Marco Antonio	mat@abc.de.br	
5	Claudio Rogério	crog@abc.de.br	51.3131.0000
6	Flavio Augusto	flau@abc.de.br	
7	Monira Rosa	moro@abc.de.b	54.0101.9090
8	Paulo Marcos	pama@abc.de.b	
9	Linda Regina	lire@abc.de.br	54.0909.9898
10	Tamires Ana	tana@abc.de.br	54.3131.0000
11	Luciano Paulo	lupa@abc.de.br	
12	Tatiana Clara	tacl@abc.de.br	
13	Robercleberson	robi@abc.de.br	
15	Luizinha	lui@empresa.co	
16	Janira	jan@empresa.co	

DELETE FROM cliente
WHERE codigo = (SELECT codigo
FROM funcionario
WHERE nome = 'Luizinha')

DELETE + SELECT

Funcionario

codigo integer	nome character var	email character var	dtanasc date	salario numeric(18,3)
1	Claudison	clau@empresa.c	1969-10-10	1000.000
2	Leinaura	lei@empresa.co	1975-10-10	750.000
3	Lobervau	lob@empresa.cc	1971-11-10	350.000
4	Norlindson	lind@empresa.c	1974-05-10	400.000
5	Berlindo	be@empresa.cc	1978-06-17	300.000
6	fluvialson	flu@empresa.cc	1980-03-01	470.000
7	Robercleberson	rob@empresa.c	1981-04-23	1500.000
8	Aristideles	ari@empresa.cc	1977-04-28	1240.000
9	Jucemar	juce@empresa.v	1978-04-28	1550.000
15	Luizinha	lui@empresa.co	1968-04-28	2550.000
16	Janira	jan@empresa.co	1971-10-18	1550.000

Cliente

codigo integer	nome character var	email character var	telefone character var
1	Juliana Paula	juju@abc.de.br	51.3131.0000
2	Paulinha Vera	pv@abc.de.br	
3	Claudia Regina	cr@abc.de.br	55.3131.0000
4	Marco Antonio	mat@abc.de.br	
5	Claudio Rogério	crog@abc.de.br	51.3131.0000
6	Flavio Augusto	flau@abc.de.br	
7	Monira Rosa	moro@abc.de.b	54.0101.9090
8	Paulo Marcos	pama@abc.de.b	
9	Linda Regina	lire@abc.de.br	54.0909.9898
10	Tamires Ana	tana@abc.de.br	54.3131.0000
11	Luciano Paulo	lupa@abc.de.br	
12	Tatiana Clara	tacl@abc.de.br	
13	Robercleberson	robi@abc.de.br	
15	Luizinha	lui@empresa.co	
16	Janira	jan@empresa.co	

Cliente

codigo integer	nome character var	email character var	telefone character var
1	Juliana Paula	juju@abc.de.br	51.3131.0000
2	Paulinha Vera	pv@abc.de.br	
3	Claudia Regina	cr@abc.de.br	55.3131.0000
5	Claudio Rogério	crog@abc.de.br	51.3131.0000
6	Flavio Augusto	flau@abc.de.br	
7	Monira Rosa	moro@abc.de.b	54.0101.9090
8	Paulo Marcos	pama@abc.de.b	
9	Linda Regina	lire@abc.de.br	54.0909.9898
10	Tamires Ana	tana@abc.de.br	54.3131.0000
11	Luciano Paulo	lupa@abc.de.br	
12	Tatiana Clara	tacl@abc.de.br	
13	Robercleberson	robi@abc.de.br	
16	Janira	jan@empresa.co	
4	Marco Antonio	mat@abc.de.br	48.5656.5656

DELETE FROM cliente

WHERE codigo = (SELECT codigo

FROM funcionario

WHERE nome = 'Luizinha')