



Banco de dados 1

Linguagem DML – SQL Select Avançado

Professor: Victor Hugo L. Lopes



Banco de Dados 1

Agenda:

- União de tabelas em consultas;

- Joins.





Banco de Dados 1

Junção de relações

Até este momento, vimos as capacidades da SQL de completa manipulação de dados em relações. Sabe-se construir tabelas e relacionamentos binários que une estas tabelas e cria restrições.

Agora deve-se ter capacidade para projetar tabelas com resultados de registros que unem as informações normalizadas e dispersas em diversas tabelas.



Banco de Dados 1

Junção de relações

cliente(idcliente, nomecliente, nascimento)

```
banco1=# select * from cliente;
 idcliente | nomecliente | nascimento
-----+-----+-----
          2 | Carlos Paulada | 2014-01-01
          3 | Antônio Silva | 2014-01-01
          1 | Astolfo Silva | 1902-10-10
(3 rows)
```

emprestimo(idemprestimo, valor, idcliente)

```
banco1=# select * from emprestimo;
 idemprestimo | valor | idcliente
-----+-----+-----
             1 | 1900 |          1
             2 | 2200 |          1
             3 | 4500 |          2
             4 | 258  |          1
(4 rows)
```



Banco de Dados 1

Junção de relações

cliente(idcliente, nomecliente, nascimento)

```
bancol=# select * from cliente;  
 idcliente | nomecliente | nascimento  
-----+-----+-----  
          2 | Carlos Paulada | 2014-01-01  
          3 | Antônio Silva | 2014-01-01  
          1 | Astolfo Silva | 1902-10-10  
(3 rows)
```

Select * from cliente
Where nomecliente =
'Carlos Paulada'

emprestimo(idemprestimo, valor, idcliente)

```
bancol=# select * from emprestimo;  
 idemprestimo | valor | idcliente  
-----+-----+-----  
            1 | 1900 |          1  
            2 | 2200 |          1  
            3 | 4500 |          2  
            4 | 258  |          1  
(4 rows)
```



Banco de Dados 1

Junção de relações

cliente(idcliente, nomecliente, nascimento)

```
bancol=# select * from cliente;
 idcliente | nomecliente | nascimento
-----+-----+-----
          2 | Carlos Paulada | 2014-01-01
          3 | Antônio Silva | 2014-01-01
          1 | Astolfo Silva | 1902-10-10
(3 rows)
```

Select * from cliente
Where nomecliente =
'Carlos Paulada'

```
 idcliente | nomecliente | nascimento
-----+-----+-----
          2 | Carlos Paulada | 2014-01-01
(1 row)
```

emprestimo(idemprestimo, valor, idcliente)

```
bancol=# select * from emprestimo;
 idemprestimo | valor | idcliente
-----+-----+-----
             1 | 1900 |          1
             2 | 2200 |          1
             3 | 4500 |          2
             4 | 258  |          1
(4 rows)
```



Banco de Dados 1

Junção de relações

cliente(idcliente, nomecliente, nascimento)

```
banco1=# select * from cliente;
 idcliente | nomecliente | nascimento 
-----+-----+-----
          2 | Carlos Paulada | 2014-01-01
          3 | Antônio Silva | 2014-01-01
          1 | Astolfo Silva | 1902-10-10
(3 rows)
```

Select * from cliente
Where nomecliente =
'Carlos Paulada'

```
 idcliente | nomecliente | nascimento 
-----+-----+-----
          2 | Carlos Paulada | 2014-01-01
(1 row)
```

emprestimo(idemprestimo, valor, idcliente)

```
banco1=# select * from emprestimo;
 idemprestimo | valor | idcliente 
-----+-----+-----
             1 | 1900 |          1
             2 | 2200 |          1
             3 | 4500 |          2
             4 | 258  |          1
(4 rows)
```

Select * from emprestimo
Where idcliente =
2

```
 idemprestimo | valor | idcliente 
-----+-----+-----
             3 | 4500 |          2
(1 row)
```



Banco de Dados 1

Junção de relações

Mas... e como trazer todas estas informações de uma única vez, em um único comando?

E... porque temos que nos preocupar com isto, sendo que pode ser atribuição do programador realizar as consultas separadamente?





Banco de Dados 1

Junção de relações

Join foi incorporado ao SQL para que os usuários pudessem especificar uma tabela que fosse resultado da operação de união entre duas ou mais tabelas à partir de características comuns às tabelas.

Basicamente, Join permite unir uma relação A com uma relação B, ou um conjunto maior de relacionamento.

Temos alguns tipos de Joins:

Natural Join, Inner join, Left Join, Right Join, Outer Join e outras formas.



Banco de Dados 1

Junção de relações

Forma de uso básica:

```
SELECT <Lista de campos> FROM (<Rel1> JOIN <Rel2> ON  
<Rel1.Coluna> = <Rel2.Coluna>) <Cláusulas>.
```

Onde Rel1 e Rel2 são as tabelas que serão juntadas, baseado no ponto comum que cada registro das duas tabelas tem entre as colunas de cada tabela, neste caso, “Coluna” da relação Rel1 e “Coluna” da relação Rel2.

Obs.: tanto a escrita com JOIN ou como INNER JOIN estão corretas, mas algumas queries aceitam somente a segunda escrita.



Banco de Dados 1

Junção de relações

Exemplo:

```
banco1=# select * from cliente;
 idcliente | nomecliente | nascimento
-----+-----+-----
          2 | Carlos Paulada | 2014-01-01
          3 | Antônio Silva | 2014-01-01
          1 | Astolfo Silva | 1902-10-10
(3 rows)
```

```
banco1=# select * from emprestimo;
 idemprestimo | valor | idcliente
-----+-----+-----
             1 | 1900 |          1
             2 | 2200 |          1
             3 | 4500 |          2
             4 | 258  |          1
(4 rows)
```

Select * from (emprestimo join cliente on emprestimo.idcliente = cliente.idcliente);

```
 idemprestimo | valor | idcliente | idcliente | nomecliente | nascimento
-----+-----+-----+-----+-----+-----
             1 | 1900 |          1 |          1 | Astolfo Silva | 1902-10-10
             2 | 2200 |          1 |          1 | Astolfo Silva | 1902-10-10
             3 | 4500 |          2 |          2 | Carlos Paulada | 2014-01-01
             4 | 258  |          1 |          1 | Astolfo Silva | 1902-10-10
(4 rows)
```

Idcliente presente nas duas tabelas é chamado de atributos de junção.



Banco de Dados 1

Junção de relações

Nos casos, como no exemplo anterior, em que os atributos de junção de duas tabelas possuem os mesmos nomes e tipos, pode-se utilizar o Natural Join, que dispensa a instrução que indica quais os atributos de junção.

```
bancol=# select * from (emprestimo natural join cliente);
```

idcliente	idemprestimo	valor	nomecliente	nascimento
1	1	1900	Astolfo Silva	1902-10-10
1	2	2200	Astolfo Silva	1902-10-10
2	3	4500	Carlos Paulada	2014-01-01
1	4	258	Astolfo Silva	1902-10-10

(4 rows)



Banco de Dados 1

Junção de relações

Uma outra forma de utilização de Joins utiliza a forma tradicional de utilização do Select em uma única tabela:

Select * from emprestimo **inner join** cliente on cliente.idcliente = emprestimo.idcliente where valor >= 2000;

idemprestimo	valor	idcliente	idcliente	nomecliente	nascimento
2	2200	1	1	Astolfo Silva	1902-10-10
3	4500	2	2	Carlos Paulada	2014-01-01
(2 rows)					

Observe que serão listados todos os empréstimos que possuem valor maior ou igual a 2.000,00, onde todas as informações do cliente tomador são apresentadas.



Banco de Dados 1

Junção de relações

Todos os exemplos anteriores funcionam muito bem se houver um relacionamento tipo chave estrangeira com regras bem definidas, pois sempre só haverá registro na tabela “filho” se ocorrer um registro “pai”.

Mas se isto não for verdade, pode haver registros sem “pai”.

idemprestimo	valor	idcliente
1	1900	1
2	2200	1
3	4500	2
4	258	1
5	1.99	
(5 rows)		



Banco de Dados 1

Junção de relações

Nestes casos, as junções mostradas não serão capazes de listar os registros que não possuírem valores comuns nas propriedades de junção.

```
bancol=# select * from cliente;
```

idcliente	nomecliente	nascimento
2	Carlos Paulada	2014-01-01
3	Antônio Silva	2014-01-01
1	Astolfo Silva	1902-10-10

(3 rows)

idemprestimo	valor	idcliente
1	1900	1
2	2200	1
3	4500	2
4	258	1
5	1.99	

(5 rows)

```
bancol=# select * from emprestimo inner join cliente on emprestimo.idcliente = cliente.idcliente;
```

idemprestimo	valor	idcliente	idcliente	nomecliente	nascimento
1	1900	1	1	Astolfo Silva	1902-10-10
2	2200	1	1	Astolfo Silva	1902-10-10
3	4500	2	2	Carlos Paulada	2014-01-01
4	258	1	1	Astolfo Silva	1902-10-10

(4 rows)



Banco de Dados 1

Junção de relações

Left join ou left outer join:

Considerado uma junção externa, pois visa listar registros que não precisam necessariamente possuir ligações em atributos de junção. A junção à esquerda mostra todos os registros da relação1(emprestimo), mesmo que não haja registro comum na relação2(cliente). Neste exemplo, o empréstimo sem “dono” também é listado.

```
banco1=# select * from cliente;
```

idcliente	nomecliente	nascimento
2	Carlos Paulada	2014-01-01
3	Antônio Silva	2014-01-01
1	Astolfo Silva	1902-10-10

(3 rows)

idemprestimo	valor	idcliente
1	1900	1
2	2200	1
3	4500	2
4	258	1
5	1.99	

(5 rows)

```
banco1=# select * from emprestimo left outer join cliente on emprestimo.idcliente = cliente.idcliente;
```

idemprestimo	valor	idcliente	idcliente	nomecliente	nascimento
1	1900	1	1	Astolfo Silva	1902-10-10
2	2200	1	1	Astolfo Silva	1902-10-10
3	4500	2	2	Carlos Paulada	2014-01-01
4	258	1	1	Astolfo Silva	1902-10-10
5	1.99				

(5 rows)



Banco de Dados 1

Junção de relações

Left join ou left outer join:

Repare que se alterarmos a ordem da relação1 e relação2, o resultado será completamente diferente, onde teremos listado todos os clientes e seus respectivos empréstimos, mas também serão listados clientes que não tenham empréstimos.

```
bancol=# select * from cliente;
```

idcliente	nomecliente	nascimento
2	Carlos Paulada	2014-01-01
3	Antônio Silva	2014-01-01
1	Astolfo Silva	1902-10-10

(3 rows)

idemprestimo	valor	idcliente
1	1900	1
2	2200	1
3	4500	2
4	258	1
5	1.99	

(5 rows)

```
bancol=# select * from cliente left outer join emprestimo on emprestimo.idcliente = cliente.idcli
ente;
```

idcliente	nomecliente	nascimento	idemprestimo	valor	idcliente
1	Astolfo Silva	1902-10-10	1	1900	1
1	Astolfo Silva	1902-10-10	2	2200	1
2	Carlos Paulada	2014-01-01	3	4500	2
1	Astolfo Silva	1902-10-10	4	258	1
3	Antônio Silva	2014-01-01			

(5 rows)



Banco de Dados 1

Junção de relações

Right join ou right outer join:

Esta mesma capacidade de alterar a listagem alterando-se a ordem das relações, pode ser realizada simplesmente trocando a instrução de junção à esquerda (left) para junção à direita (Right)

```
bancol=# select * from emprestimo left outer join cliente on emprestimo.idcliente = cliente.idcliente;
```

idemprestimo	valor	idcliente	idcliente	nomecliente	nascimento
1	1900	1	1	Astolfo Silva	1902-10-10
2	2200	1	1	Astolfo Silva	1902-10-10
3	4500	2	2	Carlos Paulada	2014-01-01
4	258	1	1	Astolfo Silva	1902-10-10
5	1.99				

(5 rows)

```
bancol=# select * from emprestimo right outer join cliente on emprestimo.idcliente = cliente.idcliente;
```

idemprestimo	valor	idcliente	idcliente	nomecliente	nascimento
1	1900	1	1	Astolfo Silva	1902-10-10
2	2200	1	1	Astolfo Silva	1902-10-10
3	4500	2	2	Carlos Paulada	2014-01-01
4	258	1	1	Astolfo Silva	1902-10-10
				Antônio Silva	2014-01-01

(5 rows)



Banco de Dados 1

Junção de relações

Cross Join:

Para cada registro da relação1 serão apresentados todos os registros da relação2:

```
Select * from emprestimo cross join cliente;
```

idemprestimo	valor	idcliente	idcliente	nomecliente	nascimento
1	1900	1	2	Carlos Paulada	2014-01-01
1	1900	1	3	Antônio Silva	2014-01-01
1	1900	1	1	Astolfo Silva	1902-10-10
2	2200	1	2	Carlos Paulada	2014-01-01
2	2200	1	3	Antônio Silva	2014-01-01
2	2200	1	1	Astolfo Silva	1902-10-10
3	4500	2	2	Carlos Paulada	2014-01-01
3	4500	2	3	Antônio Silva	2014-01-01
3	4500	2	1	Astolfo Silva	1902-10-10
4	258	1	2	Carlos Paulada	2014-01-01
4	258	1	3	Antônio Silva	2014-01-01
4	258	1	1	Astolfo Silva	1902-10-10
5	1.99		2	Carlos Paulada	2014-01-01
5	1.99		3	Antônio Silva	2014-01-01
5	1.99		1	Astolfo Silva	1902-10-10

15 rows)



Banco de Dados 1

Junção de relações

Full outer join:

Todo o trabalho de Left e Right outer join feitos em um único comando.

```
bancol=# select * from emprestimo full outer join cliente on emprestimo.idcliente = cliente.idcliente;
```

idemprestimo	valor	idcliente	idcliente	nomecliente	nascimento
1	1900	1	1	Astolfo Silva	1902-10-10
2	2200	1	1	Astolfo Silva	1902-10-10
3	4500	2	2	Carlos Paulada	2014-01-01
4	258	1	1	Astolfo Silva	1902-10-10
5	1.99		3	Antônio Silva	2014-01-01

(6 rows)