

Consultas do tipo Join no MySQL

Iremos as duas tabelas relacionadas abaixo:

```
CREATE TABLE clientes (  
  id int primary key auto_increment,  
  cpf varchar(15),  
  nome varchar(50),  
  email varchar(50),  
  vendedor_id int not null,  
  CONSTRAINT `vendedor_fk` FOREIGN KEY (`vendedor_id`) REFERENCES `vendedores`  
  (`id`) ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
INSERT INTO `clientes` (`cpf`, `nome`, `email`, `vendedor_id`) VALUES  
( '68201072598', 'Sr. José Lucas Marques', 'fabiana.vila@terra.com.br', '3'),  
( '17143346419', 'Carolina Madeira Filho', 'gabriela.sales@ferreira.br', '3'),  
( '55960434068', 'Sr. Jácomo Galhardo Neto', 'matias.uchoa@valentin.br', '3'),  
( '27489452188', 'Sr. Pablo Ian Ferminiano Sobrinho', 'camilo.carmona@sepulveda.com', '4'),  
( '90519735971', 'Sra. Michele Elizabeth Sanches Jr.', 'camilo.sepulveda@maia.com', '4'),  
( '58968205477', 'Pedro Mateus Salazar Filho', 'santiago.faria@uol.com.br', '3'),  
( '70524030889', 'Dr. Franco Fernando Ferreira', 'lucas44@rezende.com.br', '4'),  
( '58912986268', 'Sra. Sophie Nicole Marques Filho', 'irene.colaco@ig.com.br', '4'),  
( '15195433451', 'Sr. Giovane Bezerra Jr.', 'camacho.paulo@yahoo.com', '3'),  
( '78672177704', 'Andres Santacruz Filho', 'rangel.andres@valentin.com', '3');
```

```
CREATE TABLE vendedores (  
  id int primary key auto_increment,  
  cpf varchar(15),  
  nome varchar(50),  
  email varchar(50)  
);
```

```
INSERT INTO `vendedores` (`cpf`, `nome`, `email`) VALUES  
( '40209440753', 'Sr. Sebastião Natal da Cruz Jr.', 'ornela74@aguiar.net'),  
( '17494390959', 'Srta. Nádia Matias Filho', 'vsoares@ortiz.com.br'),  
( '53613318583', 'Joaquin Tamoio Matos Sobrinho', 'zcordeiro@hotmail.com'),  
( '28911319292', 'Amanda Rebeca D'ávila Filho', 'qpadrao@ig.com.br'),  
( '62629323066', 'Dr. Beatriz Luara Vieira', 'joao52@madeira.net.br'),  
( '17995929582', 'Manuel Marinho Campos Filho', 'giovane.pereira@uol.com.br'),  
( '89555531436', 'Dr. Kevin Gabriel Aguiar Jr.', 'carlos.santacruz@padilha.com'),  
( '18489837101', 'Dr. Mateus Queirós Delatorre Sobrinho', 'alan.pedrosa@terra.com.br'),  
( '60711303340', 'Miranda Rangel', 'catarina.torres@uol.com.br'),  
( '61894687477', 'Miranda Fontes Padrão Sobrinho', 'csantana@gmail.com');
```

O que resulta em:

cpf	nome	email
68201072598	Sr. José Lucas Marques	ornela74@aguiar.net
17143346419	Carolina Madeira Filho	vsoares@ortiz.com.br
55960434068	Sr. Jácomo Galhardo Neto	zcordeiro@hotmail.com
27489452188	Sr. Pablo Ian Ferminiano Sobrinho	qpadrao@ig.com.br
90519735971	Sra. Michele Elizabeth Sanches Jr.	joao52@madeira.net.br
58968205477	Pedro Mateus Salazar Filho	giovane.pereira@uol.com.br
70524030889	Dr. Franco Fernando Ferreira	carlos.santacruz@padilha.com
58912986268	Sra. Sophie Nicole Marques Filho	alan.pedrosa@terra.com.br
15195433451	Sr. Giovane Bezerra Jr.	catarina.torres@uol.com.br
78672177704	Andres Santacruz Filho	csantana@gmail.com

Inner Join

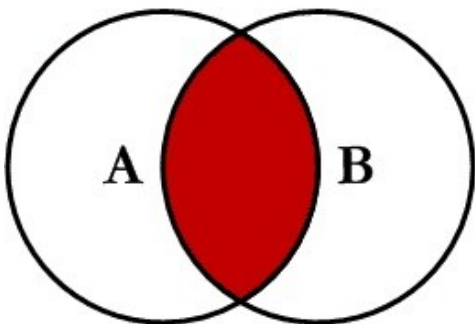
A consulta do tipo Inner Join junta os registros de duas ou mais tabelas de acordo com uma condição, chamada de predicado.

O Inner Join compara cada registro da primeira tabela com cada um da segunda tabela. Caso o valor de ambos os registros torne a condição verdadeira, o Inner Join cria um novo registro, com as colunas de ambas as tabelas. Resumindo o Inner Join inclui somente registros que atendem à condição e estão nas duas tabelas.

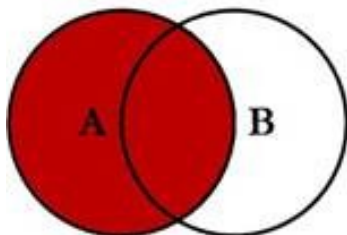
Executar a consulta

```
SELECT  
c.cpf,  
c.nome,  
v.email  
FROM clientes c  
INNER JOIN vendedores v  
ON c.id = v.id;
```

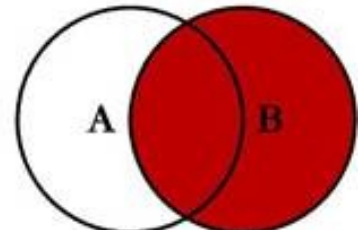
Gráfico



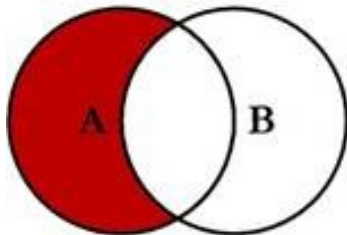
SQL JOINS



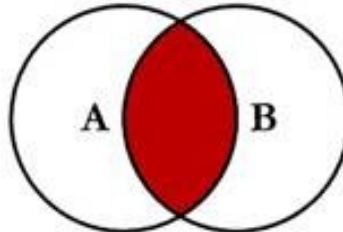
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



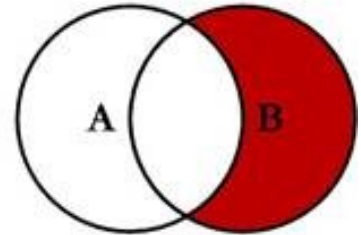
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



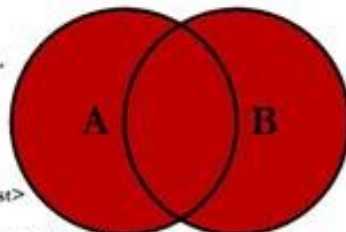
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL.
```



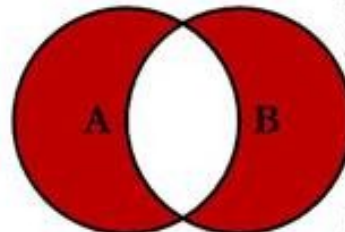
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL.
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



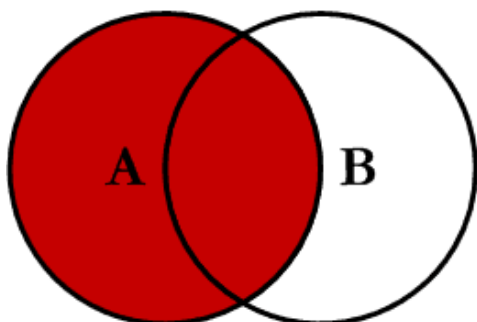
```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL.
```

© C.L. Moffatt, 2008

Consulta do Tipo Left Join

O Left Join, cujo funcionamento é ilustrado na **Figura 4**, tem como resultado todos os registros que estão na tabela A (mesmo que não estejam na tabela B) e os registros da tabela B que são comuns à tabela A.

Gráfico



EXEMPLO

```
SELECT c.nome, v.nome
FROM clientes as c
LEFT JOIN vendedores as v
ON c.nome = v.nome;
```

Resultado

nome	nome
Sr. José Lucas Marques	<i>NULL</i>
Carolina Madeira Filho	<i>NULL</i>
Sr. Jácomo Galhardo Neto	<i>NULL</i>
Sr. Pablo Ian Ferminiano Sobrinho	<i>NULL</i>
Sra. Michele Elizabeth Sanches Jr.	<i>NULL</i>
Pedro Mateus Salazar Filho	<i>NULL</i>
Dr. Franco Fernando Ferreira	<i>NULL</i>
Sra. Sophie Nicole Marques Filho	<i>NULL</i>
Sr. Giovane Bezerra Jr.	<i>NULL</i>
Andres Santacruz Filho	<i>NULL</i>

Mas se mudarmos a consulta para

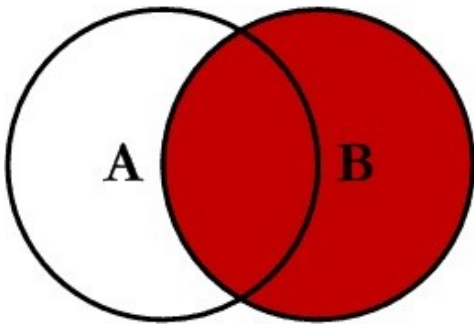
```
SELECT c.nome, v.nome
FROM clientes as c
LEFT JOIN vendedores as v
ON c.id = v.id;
```

Então retorna valores válidos

nome	nome
Sr. José Lucas Marques	Sr. Sebastião Natal da Cruz Jr.
Carolina Madeira Filho	Srta. Nádia Matias Filho
Sr. Jácomo Galhardo Neto	Joaquin Tamoio Matos Sobrinho
Sr. Pablo Ian Ferminiano Sobrinho	Amanda Rebeca D'ávila Filho
Sra. Michele Elizabeth Sanches Jr.	Dr. Beatriz Luara Vieira
Pedro Mateus Salazar Filho	Manuel Marinho Campos Filho
Dr. Franco Fernando Ferreira	Dr. Kevin Gabriel Aguiar Jr.
Sra. Sophie Nicole Marques Filho	Dr. Mateus Queirós Delatorre Sobrinho
Sr. Giovane Bezerra Jr.	Miranda Rangel
Andres Santacruz Filho	Miranda Fontes Padrão Sobrinho

Right Join

Usando o Right Join, teremos como resultado todos os registros que estão na tabela B (mesmo que não estejam na tabela A) e os registros da tabela A que são comuns à tabela B.



Exemplo

```
SELECT c.nome, v.nome
FROM clientes as c
RIGHT JOIN vendedores as v
ON c.id = v.id;
```

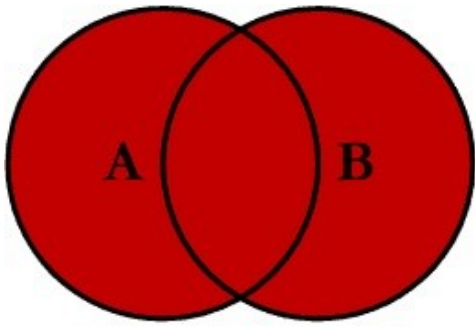
Execute e veja que não houve diferença.

Simulando Full Outer Join

```
SELECT * FROM clientes
LEFT JOIN vendedores ON clientes.id = vendedores.id
UNION
SELECT * FROM clientes
RIGHT JOIN vendedores ON clientes.id = vendedores.id;
```

Veja o que retorna

id	cpf	nome	email	vendedor_id	id	cpf	nome	email
1	68201072598	Sr. José Lucas Marques	fabiana.vila@terra.com.br	3	1	40209440753	Sr. Sebastião Natal da Cruz Jr.	ornela74@ag
2	17143346419	Carolina Madeira Filho	gabriela.sales@ferreira.br	3	2	17494390959	Srta. Nádia Matias Filho	vsoares@ortiz
3	55960434068	Sr. Jácomo Galhardo Neto	matias.uchoa@valentin.br	3	3	53613318583	Joaquin Tamoio Matos Sobrinho	zcordeiro@ho
4	27489452188	Sr. Pablo Ian Ferminiano Sobrinho	camilo.carmona@sepulveda.com	4	4	28911319292	Amanda Rebeca D'ávila Filho	qpadrao@ig.c
5	90519735971	Sra. Michele Elizabeth Sanches Jr.	camilo.sepulveda@maia.com	4	5	62629323066	Dr. Beatriz Luara Vieira	joao52@made
6	58968205477	Pedro Mateus Salazar Filho	santiago.faria@uol.com.br	3	6	17995929582	Manuel Marinho Campos Filho	giovane.pereir
7	70524030889	Dr. Franco Fernando Ferreira	lucas44@rezende.com.br	4	7	89555531436	Dr. Kevin Gabriel Aguiar Jr.	carlos.santacr
8	58912986268	Sra. Sophie Nicole Marques Filho	irene.colaco@ig.com.br	4	8	18489837101	Dr. Mateus Queirós Delatorre Sobrinho	alan.pedrosa@g
9	15195433451	Sr. Giovane Bezerra Jr.	camacho.paulo@yahoo.com	3	9	60711303340	Miranda Rangel	catarina.torres
10	78672177704	Andres Santacruz Filho	rangel.andres@valentin.com	3	10	61894687477	Miranda Fontes Padrão Sobrinho	csantana@gm



O MySQL não suporta o Full Outer Join, mas podemos simulara com a consulta acima.

Exemplos

Consulta nas tabelas posts e users

```
SELECT *,  
posts.id as postId,  
user.id as userId,  
posts.created_at as postCreated,  
user.created_at as userCreated  
FROM posts  
INNER JOIN user  
ON posts.user_id = user.id  
ORDER BY posts.created_at DESC;
```

```
SELECT torneios.nome, torneios.tamanho, refeicoes.data_nascimento  
FROM torneios  
JOIN refeicoes ON torneios.nome=refeicoes.nome;
```