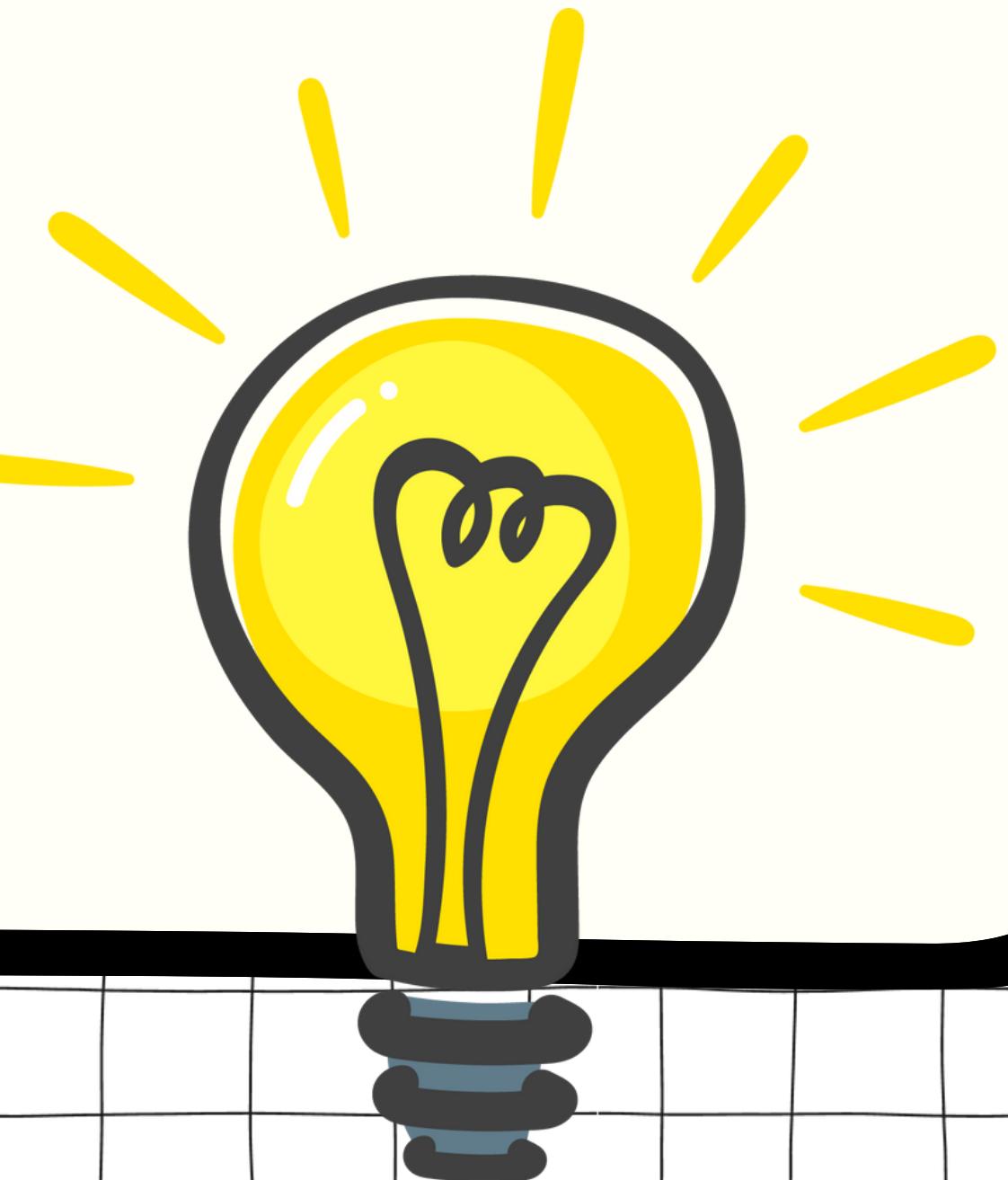


# Joins MySQL

M. M. Omai



# Introdução

Até agora, selecionamos dados de **apenas uma tabela**. Para fazer um **consulta em mais de uma**, nós utilizamos os chamados **JOINS**

...

## curso

codCurso	nomeCurso	valor
1	INI	5000
2	PAV	5000

## aluno

matricula	nome	sexo	idade
SM300213	Helena	F	17
SM300214	Hellen	F	17
SM300215	Luana	F	17
SM300216	Murilo	M	16
SM300217	Henrico	M	15
SM300218	Erick	M	16
SM300219	Larissa	F	15
SM300220	Joana	F	16
SM300221	Lucas	M	15

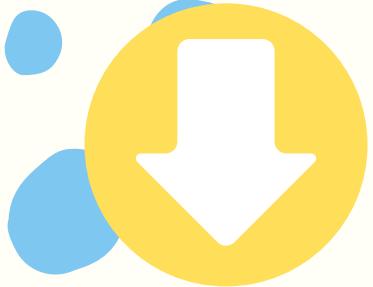
# Como funciona o JOIN

Precisamos usar uma **condição de ligação** para “JUNTAR” as tabelas.

Uma boa opção,  
normalmente é a  
**chave primária** com a  
**chave estrangeira**.  
Mas, não é obrigatório..



Existem vários tipos de  
joins, cabe a você  
analisar em qual  
situação usar cada um..



## matcurso

matricula	codCurso	dataMatricula
SM300213	1	2021-02-01
SM300214	1	2021-02-01
SM300215	1	2020-02-01
SM300216	1	2022-02-01
SM300217	1	2021-02-01
SM300218	1	2021-02-01
SM300219	2	2020-02-01
SM300220	2	2019-02-01
SM300221	2	2020-02-01

## curso

codCurso	nomeCurso	valor
1	INI	5000
	PAV	5000

## aluno

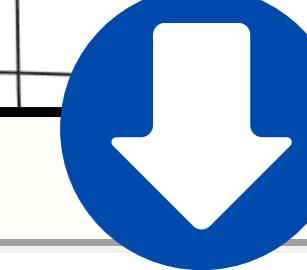
matricula	nome	sexo	idade
SM300213	Helena	F	17
SM300214	Hellen	F	17
SM300215	Luana	F	17
SM300216	Murilo	M	16
SM300217	Henrico	M	15
SM300218	Erick	M	16
SM300219	Larissa	F	15
SM300220	Joana	F	16
SM300221	Lucas	M	15

S

## matcurso



matricula	codCurso	dataMatricula
SM300213	1	2021-02-01
SM300214	1	2021-02-01
SM300215	1	2020-02-01
SM300216	1	2022-02-01
SM300217	1	2021-02-01
SM300218	1	2021-02-01
SM300219	2	2020-02-01
SM300220	2	2019-02-01
SM300221	2	2020-02-01



## curso

codCurso	nomeCurso	valor
1	INI	5000
2	PAV	5000

## aluno

matricula	nome	sexo	idade
SM300213	Helena	F	17
SM300214	Hellen	F	17
SM300215	Luana	F	17
SM300216	Murilo	M	16
SM300217	Henrico	M	15
SM300218	Erick	M	16
SM300219	Larissa	F	15
SM300220	Joana	F	16
SM300221	Lucas	M	15

US

## Tabela criada usando JOIN

matricula	nome	sexo	idade	curso
SM300213	Helena	F	17	INI
SM300214	Hellen	F	17	INI
SM300215	Luana	F	17	INI
SM300216	Murilo	M	16	INI
SM300217	Henrico	M	15	INI
SM300218	Erick	M	16	INI
SM300219	Larissa	F	15	PAV
SM300220	Joana	F	16	PAV
SM300221	Lucas	M	15	PAV

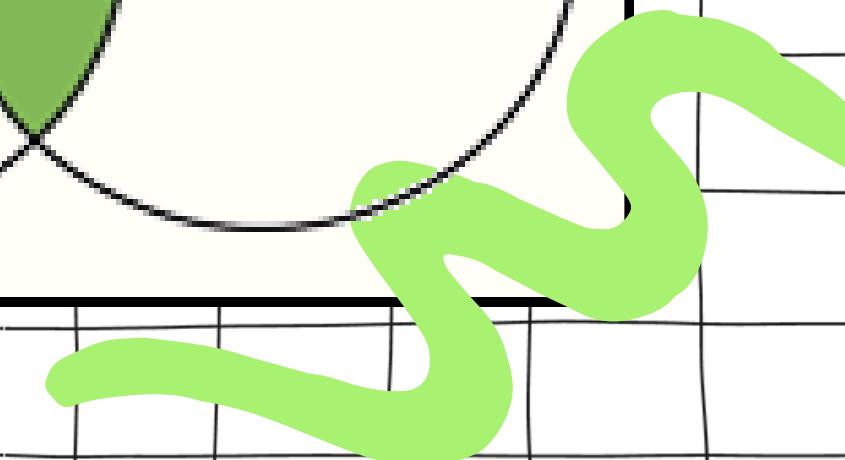
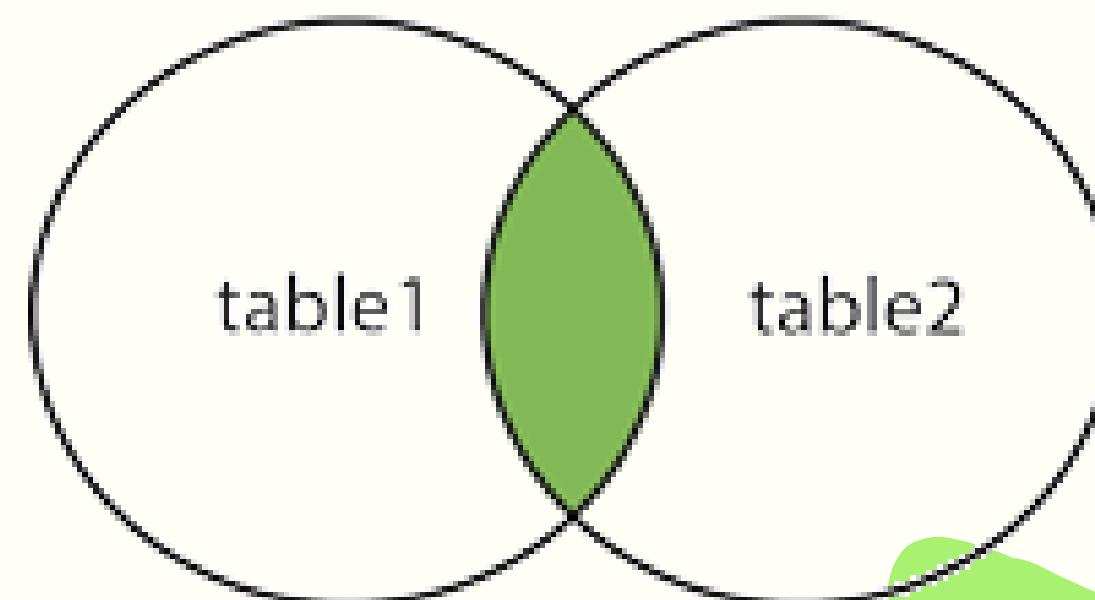
# INNER JOIN

```
SELECT t1.coluna, t2.coluna  
FROM table1 t1  
INNER JOIN table2 t2 ON  
condicao_join
```

Retorna apenas os registros que têm correspondência em ambas as tabelas envolvidas na consulta.

**Descarta os registros que não possuem correspondência.**

INNER JOIN

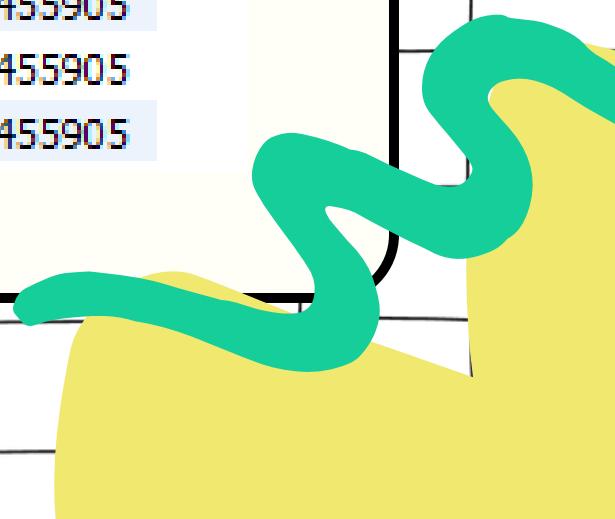


```
1 • use loja;  
2 #aula-join  
3  
4 • select * from cliente inner join endereco;
```



Result Grid | Filter Rows: Export: Wrap Cell Content:

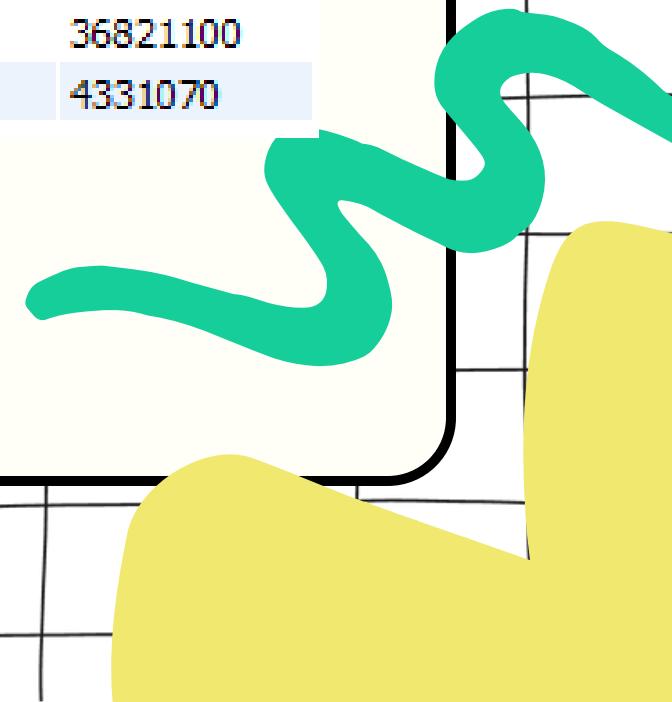
	idCliente	idEndereco	nome	sexo	idade	idEndereco	idTipoEndereco	idBairro	nome	numero	cep
▶	1	100	Kliver Gabriel	M	16	100	104	1	Rua Fancisco Joé Viana	109	5882020
	2	101	Matheus Vitor	M	17	100	104	1	Rua Fancisco Joé Viana	109	5882020
	3	102	Davi Kobayashi	M	15	100	104	1	Rua Fancisco Joé Viana	109	5882020
	4	103	Maria João	F	18	100	104	1	Rua Fancisco Joé Viana	109	5882020
	5	104	José Maria	M	77	100	104	1	Rua Fancisco Joé Viana	109	5882020
	6	105	Antônio Francisco	M	48	100	104	1	Rua Fancisco Joé Viana	109	5882020
	7	106	Maria Maria	F	50	100	104	1	Rua Fancisco Joé Viana	109	5882020
	8	107	Vinícius João	M	17	100	104	1	Rua Fancisco Joé Viana	109	5882020
	9	108	Pedro Diniz	M	18	100	104	1	Rua Fancisco Joé Viana	109	5882020
	10	109	Inácio Bonke	M	15	100	104	1	Rua Fancisco Joé Viana	109	5882020
	1	100	Kliver Gabriel	M	16	101	104	3	Rua Hungria	888	14455905
	2	101	Matheus Vitor	M	17	101	104	3	Rua Hungria	888	14455905
	3	102	Davi Kobayashi	M	15	101	104	3	Rua Hungria	888	14455905
	4	103	Maria João	F	18	101	104	3	Rua Hungria	888	14455905



```
1 • use loja;  
2 #aula-join  
3  
4 • select * from cliente c inner join endereco e on c.idendereco = e.idendereco;
```



	idCliente	idEndereco	nome	sexo	idade	idEndereco	idTipoEndereco	idBairro	nome	numero	cep
▶	1	100	Klever Gabriel	M	16	100	104	1	Rua Fancisco Joé Viana	109	5882020
	2	101	Matheus Vitor	M	17	101	104	3	Rua Hungria	888	14455905
	3	102	Davi Kobayashi	M	15	102	105	2	Sérgio de Carvalho	647	226412200
	4	103	Maria João	F	18	103	108	1	Fernão Dias	666	2283000
	5	104	José Maria	M	77	104	100	6	Avenida Getúlio Vargas	34	699006100
	6	105	Antônio Francisco	M	48	105	107	4	Estrada Atalaia	276	317484000
	7	106	Maria Maria	F	50	106	109	7	Vila Elizeu Batista	88	60843246
	8	107	Vinícius João	M	17	107	103	9	Largo Professor Saul Borges Carneiro	789	222400901
	9	108	Pedro Diniz	M	18	108	101	5	Rua São José	345	36821100
	10	109	Inácio Bonke	M	15	109	106	8	Rua Francisco Scardapane	22	4331070





```
1 • use loja;  
2 #aula-join  
3  
4 • select c.nome, c.idade, e.nome as 'endereço', e.numero, e.cep from cliente c inner join endereco e on c.idendereco = e.idendereco;
```

nome	idade	endereco	numero	cep
Klever Gabriel	16	Rua Francisco José Viana	109	5882020
Matheus Vitor	17	Rua Hungria	888	14455905
Davi Kobayashi	15	Sérgio de Carvalho	647	226412200
Maria João	18	Fernão Dias	666	2283000
José Maria	77	Avenida Getúlio Vargas	34	699006100
Antônio Francisco	48	Estrada Atalaia	276	317484000
Maria Maria	50	Vila Elizeu Batista	88	60843246
Vinícius João	17	Largo Professor Saul Borges Carneiro	789	222400901
Pedro Diniz	18	Rua São José	345	36821100
Inácio Bonke	15	Rua Francisco Scardapane	22	4331070

Quando temos colunas com mesmo nome em tabelas distintas, precisamos identificar a tabela junto com a coluna, como mostra o exemplo.

2 #aula-join

```
4 select c.nome, c.idade, e.nome as 'endereço', e.numero, e.cep, b.nome as 'bairro' from cliente c  
5 inner join endereco e on c.idendereco = e.idendereco  
6 inner join bairro b on e.idbairro = b.idbairro;
```

nome	idade	endereço	numero	cep	bairro
Klever Gabriel	16	Rua Francisco José Viana	109	5882020	Tiradentes
Matheus Vitor	17	Rua Hungria	888	14455905	São Miguel
Davi Kobayashi	15	Sérgio de Carvalho	647	226412200	Copacabana
Maria João	18	Fernão Dias	666	2283000	Tiradentes
José Maria	77	Avenida Getúlio Vargas	34	699006100	Rosa Linda
Antônio Francisco	48	Estrada Atalaia	276	317484000	Caxias do Sul
Maria Maria	50	Vila Elizeu Batista	88	60843246	Barra do Ceará
Vinícius João	17	Largo Professor Saul Borges Carneiro	789	222400901	Ipanema
Pedro Diniz	18	Rua São José	345	36821100	Pimentas
Inácio Bonke	15	Rua Francisco Scardapane	22	4331070	Perdizes

Podemos fazer sucessivos JOINs e eles não precisam ser do mesmo tipo

Observe que dentre os clientes, existe um que não tem endereço associado. Portanto, ele não aparece na tabela resultante.

• • •

<b>idCliente</b>	<b>idEndereco</b>	<b>nome</b>	<b>sexo</b>	<b>idade</b>
1	100	Kliver Gabriel	M	16
2	101	Matheus Vitor	M	17
3	102	Davi Kobayashi	M	15
4	103	Maria João	F	18
5	104	José Maria	M	77
6	105	Antônio Francisco	M	48
7	106	Maria Maria	F	50
8	107	Vinícius João	M	17
9	108	Pedro Diniz	M	18
10	109	Inácio Bonke	M	15
11	NULL	Mayara Omai	F	29

<b>nome</b>	<b>idade</b>	<b>endereco</b>	<b>numero</b>	<b>cep</b>	<b>bairro</b>
Kliver Gabriel	16	Rua Francisco José Viana	109	5882020	Tiradentes
Matheus Vitor	17	Rua Hungria	888	14455905	São Miguel
Davi Kobayashi	15	Sérgio de Carvalho	647	226412200	Copacabana
Maria João	18	Fernão Dias	666	2283000	Tiradentes
José Maria	77	Avenida Getúlio Vargas	34	699006100	Rosa Linda
Antônio Francisco	48	Estrada Atalaia	276	317484000	Caxias do Sul
Maria Maria	50	Vila Elizeu Batista	88	60843246	Barra do Ceará
Vinícius João	17	Largo Professor Saul Borges Carneiro	789	222400901	Ipanema
Pedro Diniz	18	Rua São José	345	36821100	Pimentas
Inácio Bonke	15	Rua Francisco Scardapane	22	4331070	Perdizes

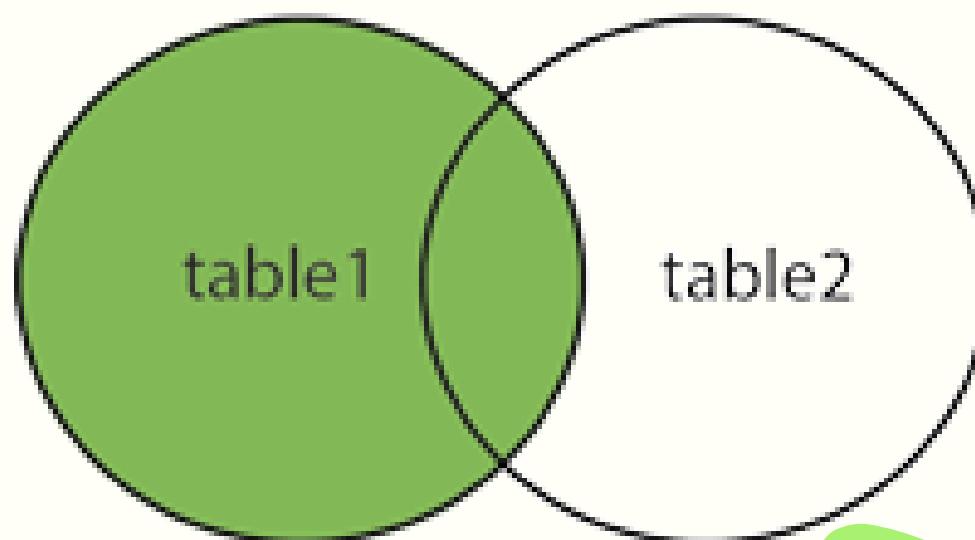
# LEFT JOIN

```
SELECT t1.coluna, t2.coluna  
FROM table1 t1  
LEFT JOIN table2 t2 ON  
condicao_join
```

A junção à esquerda recupera os dados a partir da tabela à esquerda.

**Se não houver correspondência na tabela da direita, são retornados valores NULL para as colunas da tabela da direita.**

LEFT JOIN

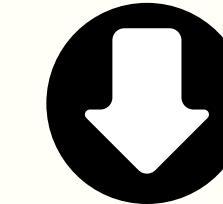


É útil quando se deseja recuperar todos os registros da tabela da esquerda e seus relacionamentos correspondentes da tabela da direita, mesmo que alguns registros na tabela da direita não tenham correspondência.

```
select c.nome, c.idade, e.nome as 'endereço', e.numero, e.cep from cliente c
left join endereco e on c.idendereco = e.idendereco;
```

<b>idCliente</b>	<b>idEndereco</b>	<b>nome</b>	<b>sexo</b>	<b>idade</b>
1	100	Kliver Gabriel	M	16
2	101	Matheus Vitor	M	17
3	102	Davi Kobayashi	M	15
4	103	Maria João	F	18
5	104	José Maria	M	77
6	105	Antônio Francisco	M	48
7	106	Maria Maria	F	50
8	107	Vinícius João	M	17
9	108	Pedro Diniz	M	18
10	109	Inácio Bonke	M	15
11	NULL	Mayara Omai	F	29

<b>nome</b>	<b>idade</b>	<b>endereço</b>	<b>numero</b>	<b>cep</b>
Kliver Gabriel	16	Rua Francisco José Viana	109	5882020
Matheus Vitor	17	Rua Hungria	888	14455905
Davi Kobayashi	15	Sérgio de Carvalho	647	226412200
Maria João	18	Fernão Dias	666	2283000
José Maria	77	Avenida Getúlio Vargas	34	699006100
Antônio Francisco	48	Estrada Atalaia	276	317484000
Maria Maria	50	Vila Elizeu Batista	88	60843246
Vinícius João	17	Largo Professor Saul Borges Carneiro	789	222400901
Pedro Diniz	18	Rua São José	345	36821100
Inácio Bonke	15	Rua Francisco Scardapane	22	4331070
Mayara Omai	29	HULL	HULL	HULL



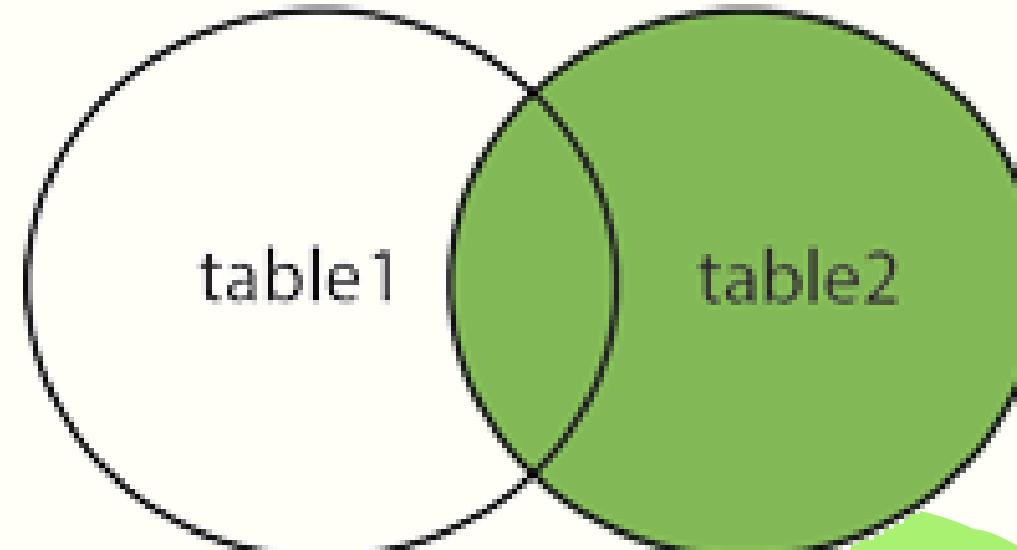
# RIGHT JOIN

```
SELECT t1.coluna, t2.coluna  
FROM table1 t1  
RIGHT JOIN table2 t2 ON  
condicao_join
```

A junção à direita recupera os dados a partir da tabela à direita.

**Se não houver correspondência na tabela da esquerda, são retornados valores NULL para as colunas da tabela da direita.**

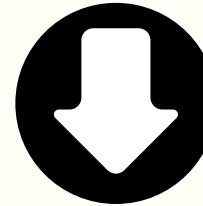
RIGHT JOIN



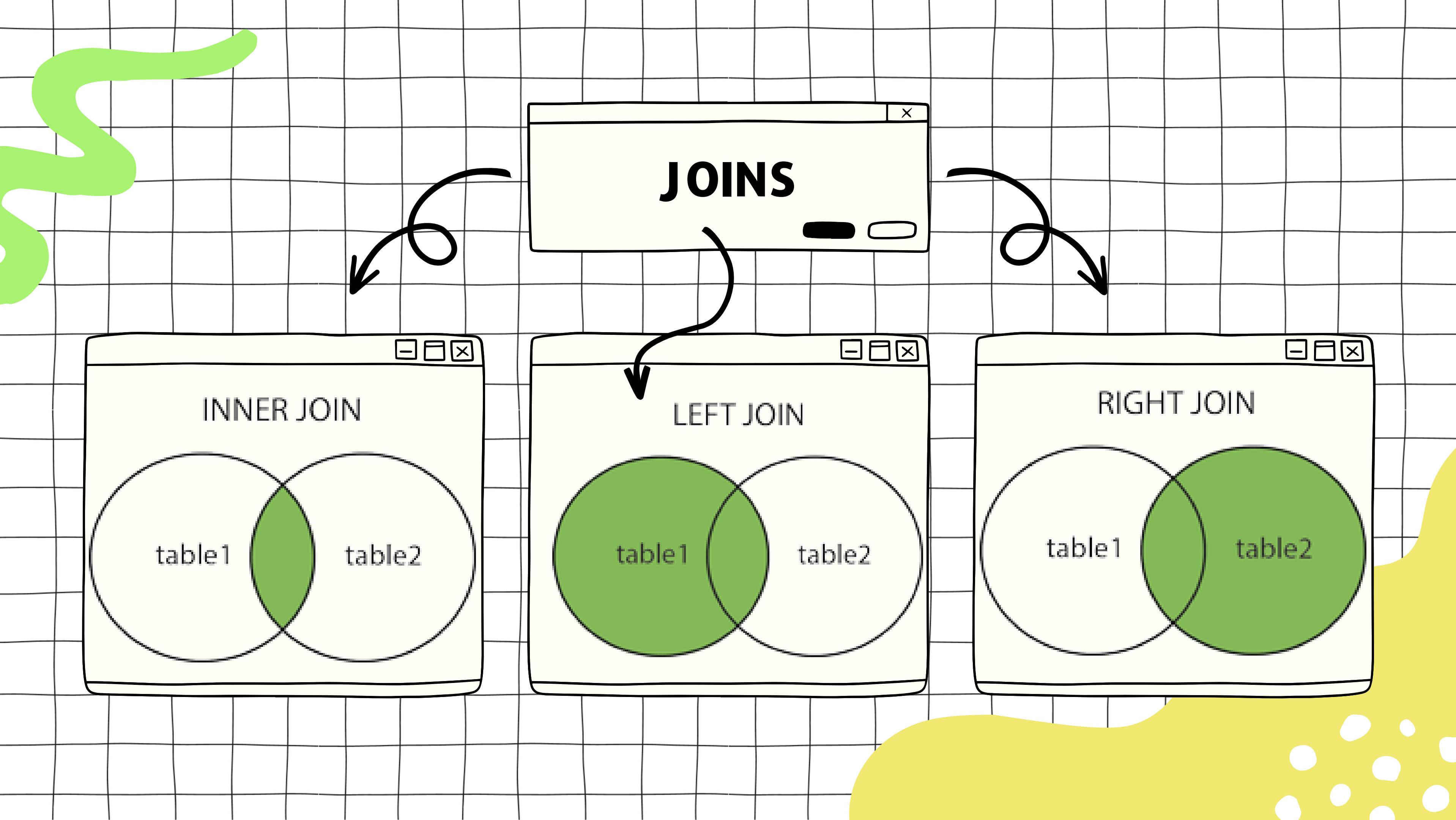
Menos comum que o LEFT JOIN, mas útil quando se deseja recuperar todos os registros da tabela da direita, mesmo que alguns registros na tabela da esquerda não tenham correspondência

```
select c.nome, c.idade, e.nome as 'endereço', e.numero, e.cep from cliente c  
right join endereco e on c.idendereco = e.idendereco;
```

idCliente	idEndereco	nome	sexo	idade
1	100	Kliver Gabriel	M	16
2	101	Matheus Vitor	M	17
3	102	Davi Kobayashi	M	15
4	103	Maria João	F	18
5	104	José Maria	M	77
6	105	Antônio Francisco	M	48
7	106	Maria Maria	F	50
8	107	Vinícius João	M	17
9	108	Pedro Diniz	M	18
10	109	Inácio Bonke	M	15
11	NULL	Mayara Omai	F	29



nome	idade	endereço	numero	cep
Kliver Gabriel	16	Rua Fandisco José Viana	109	5882020
Matheus Vitor	17	Rua Hungria	888	14455905
Davi Kobayashi	15	Sérgio de Carvalho	647	226412200
Maria João	18	Fernão Dias	666	2283000
José Maria	77	Avenida Getúlio Vargas	34	699006100
Antônio Francisco	48	Estrada Atalaia	276	317484000
Maria Maria	50	Vila Elizeu Batista	88	60843246
Vinícius João	17	Largo Professor Saul Borges Carneiro	789	222400901
Pedro Diniz	18	Rua São José	345	36821100
Inácio Bonke	15	Rua Francisco Scardapane	22	4331070



# FULL JOIN



```
SELECT t1.coluna, t2.coluna  
FROM table1 t1  
RIGHT JOIN table2 t2 ON condicao_join  
UNION ALL  
SELECT t1.coluna, t2.coluna  
FROM table1 t1  
LEFT JOIN table2 t2 ON condicao_join
```

Retorna todos os registros de ambas as tabelas, incluindo aqueles que não têm correspondência na outra tabela. O MySQL não suporta FULL JOIN diretamente, mas você pode simulá-lo usando uma combinação de LEFT JOIN e UNION ALL.

...

```

select c.nome, c.idade, e.nome as 'endereço', e.numero, e.cep from cliente c
left join endereco e on c.idendereco = e.idendereco
UNION ALL
select c.nome, c.idade, e.nome as 'endereço', e.numero, e.cep from cliente c
right join endereco e on c.idendereco = e.idendereco;

```

nome	idade	endereco	numero	cep				
Klever Gabriel	16	Rua Francisco José Viana	109	5882020				
Matheus Vitor	17	Rua Hungria	888	14455905				
Davi Kobayashi	15	Sérgio de Carvalho	647	226412200				
Maria João	18	Fernão Dias	666	2283000				
José Maria	77	Avenida Getúlio Vargas	34	699006100				
Antônio Francisco	48	Estrada Atalaia	276	317484000	Maria João	18	Fernão Dias	666
Maria Maria	50	Vila Elizeu Batista	88	60843246	José Maria	77	Avenida Getúlio Vargas	34
Vinícius João	17	Largo Professor Saul Borges Carneiro	789	222400901	Antônio Francisco	48	Estrada Atalaia	276
Pedro Diniz	18	Rua São José	345	36821100	Maria Maria	50	Vila Elizeu Batista	88
Inácio Bonke	15	Rua Francisco Scardapane	22	4331070	Vinícius João	17	Largo Professor Saul Borges Carneiro	789
Mayara Omai	29	NULL	NULL	NULL	Pedro Diniz	18	Rua São José	345
Klever Gabriel	16	Rua Francisco José Viana	109	5882020	Inácio Bonke	15	Rua Francisco Scardapane	22
Matheus Vitor	17	Rua Hungria	888	14455905				4331070
Davi Kobayashi	15	Sérgio de Carvalho	647	226412200				

# CROSS JOIN

```
SELECT t1.coluna, t2.coluna  
FROM table1 t1  
CROSS JOIN table2 t2
```

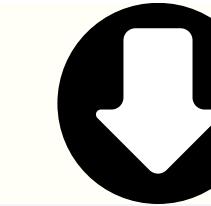
Produz o **produto cartesiano** entre as tabelas envolvidas, combinando cada registro da primeira tabela com cada registro da segunda tabela.

Gera um grande conjunto de resultados, especialmente quando as tabelas são grandes.



Geralmente não é usado para recuperar dados relacionados, mas pode ser útil em situações específicas.

```
select * from cliente c  
cross join endereco e;
```



	idCliente	idEndereco	nome	sexo	idade	idEndereco	idTipoEndereco	idBairro	nome	numero	cep
▶	1	100	Kliver Gabriel	M	16	100	104	1	Rua Fancisco Joé Viana	109	5882020
	1	100	Kliver Gabriel	M	16	101	104	3	Rua Hungria	888	14455905
	1	100	Kliver Gabriel	M	16	102	105	2	Sérgio de Carvalho	647	226412200
	1	100	Kliver Gabriel	M	16	103	108	1	Fernão Dias	666	2283000
	1	100	Kliver Gabriel	M	16	104	100	6	Avenida Getúlio Vargas	34	699006100
	1	100	Kliver Gabriel	M	16	105	107	4	Estrada Atalaia	276	317484000
	1	100	Kliver Gabriel	M	16	106	109	7	Vila Elizeu Batista	88	60843246
	1	100	Kliver Gabriel	M	16	107	103	9	Largo Professor Saul Borges Carneiro	789	222400901
	1	100	Kliver Gabriel	M	16	108	101	5	Rua São José	345	36821100
	1	100	Kliver Gabriel	M	16	109	106	8	Rua Francisco Scardapane	22	4331070
	2	101	Matheus Vitor	M	17	100	104	1	Rua Fancisco Joé Viana	109	5882020
	2	101	Matheus Vitor	M	17	101	104	3	Rua Hungria	888	14455905
	2	101	Matheus Vitor	M	17	102	105	2	Sérgio de Carvalho	647	226412200
	2	101	Matheus Vitor	M	17	103	108	1	Fernão Dias	666	2283000