

transforme ■ se



JAVA

AULA 8 - Parte 3 - Relacionamentos



Relacionamentos no SQL



Relacionamentos no SQL

É possível, como já foi falado, existem relacionamentos entre tabelas num banco de dados relacional.

Os relacionamentos de banco de dados são associações entre tabelas que são criadas usando instruções de junção para recuperar dados.

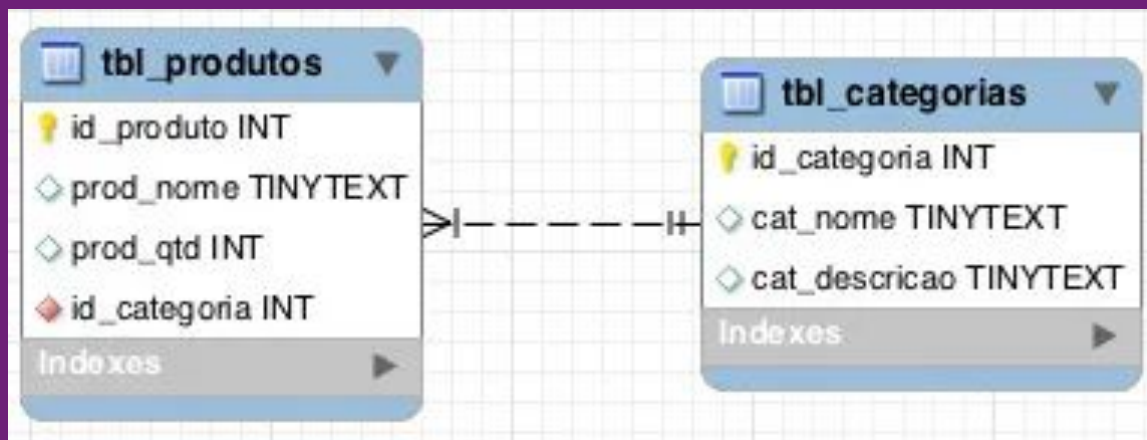
Existem três tipos de relacionamentos e cada um deles tem o seu nome de acordo com os registros dos relacionamentos que estão acontecendo.

1-para-1

Ocorre quando um registro de uma tabela tem uma, e apenas uma contraparte em uma outra tabela. Este tipo de relacionamento é muito raro de acontecer, já que é muito mais simples e eficiente agrupar os registros é uma única tabela.

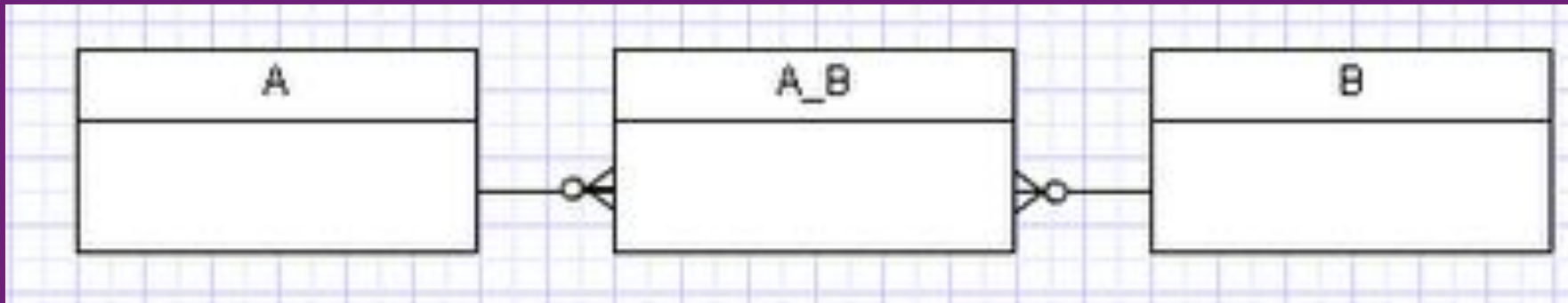
1-para-N

É o tipo de relacionamento mais comum de acontecer. Eles ocorrem quando um registro de uma tabela se relaciona com vários registros de uma outra tabela, mas que cada registro da segunda tabela só corresponde com um registro da primeira tabela. Uma tabela de categorias tem os seus registros e a tabela de produtos também, onde cada produto pertence a uma única categoria, mas nunca um produto pertencerá a várias categorias. Veja o exemplo da figura abaixo:



N-para-M

Este ocorre quando um registro de uma tabela corresponde a um ou mais registros da segunda tabela e cada registro da segunda tabela corresponde a um ou mais registros da primeira. Um exemplo simples, seria de uma tabela de professores e cursos, porque cada professor pode ensinar em mais de um curso e cada curso pode ter mais de um professor. Neste caso, devemos criar uma nova tabela, onde irá reduzir para um relacionamento 1-para-N entre as duas tabelas iniciais.



Pegando dados de relacionamentos

Para fazer um SELECT para buscar dados de relacionamentos, como estes ocorrem em diferentes tabelas, precisamos fazer uso do parâmetro JOIN e suas variações.

A sintaxe ficaria assim:

```
SELECT ... FROM T1 JOIN T2 ON T1.FK = T2.PK
```


Inner join

É conhecido como o JOIN padrão. Quando não é especificado nenhuma outra versão de JOIN é o INNER JOIN que é executado.

Gera um produto cartesiano entre as tabelas, ou seja, combina todas as linhas da primeira tabela com todas as linhas da segunda, que satisfaçam as condições das chaves.

Um exemplo:



```
1 SELECT * FROM pessoas p INNER JOIN veiculos v ON p.cpf = v.cpf;
```

	cpf	nome	id	veiculo	placa	cpf
▶	00667791035	p2	1	monza	kqx1155	00667791035
	27783525032	p4	2	chevet	nks1245	27783525032
	27783525032	p4	3	ford ka	rte7845	27783525032

Equi join

É uma maneira mais simplificada de se escrever o Inner Join. Pode ser usado no exemplo dado uma vez que a coluna que define a relação entre as tabelas tem o mesmo nome das duas, que é cpf.

Um ponto interessante do Equi Join é que ele ao identificar o campo usado para a relação, não o traz de forma repetida, como é o caso do Inner Join, veja o comando e o resultado:



```
1  SELECT * FROM pessoas JOIN veiculos USING (cpf);
```

	cpf	nome	id	veiculo	placa
►	00667791035	p2	1	monza	kqx1155
	27783525032	p4	2	chevet	nks1245
	27783525032	p4	3	ford ka	rte7845

Equi join

O Equi Join pode ser realizado também quando não temos campos com o nome em comum, mas nesse caso, precisamos passar os campos que serão usados na relação.



```
1 SELECT F.NOME, F.SALARIO, S.FAIXA FROM FUNCIONARIOS F INNER JOIN SALARIOS S ON F.SALARIO BETWEEN S.INICIO AND S.FIM;
```

	id	nome	salario	departamento	cpf
▶	1	Madalena	2750	vendas	11111111111
	2	Ana	1650	vendas	22222222222
	4	Maria Souza	4500	diretoria	33333333333

	faixa	inicio	fim
	Analisa Pl	2001	3000
	Analista JR	1000	2000

	NOME	SALARIO	FAIXA
▶	Madalena	2750	Analisa Pl
	Ana	1650	Analista JR

Outer join, Left join, Left outer join

A cláusula LEFT JOIN ou LEFT OUTER JOIN permite obter não apenas os dados relacionados de duas tabelas, mas também os dados não relacionados encontrados na tabela à esquerda da cláusula JOIN. Caso não existam dados relacionados entre as tabelas à esquerda e à direita do JOIN, os valores resultantes de todas as colunas da lista de seleção da tabela à direita serão nulos. Vamos a uma exemplo:



```
1 SELECT * FROM PESSOAS LEFT JOIN VEICULOS ON PESSOAS.CPF = VEICULOS.CPF;
```

Pessoas

	cpf	nome
▶	00667791035	p2
	27783525032	p4
	44701230057	p3
	81489134042	p1

Veículos

	id	veiculo	placa	cpf
▶	1	monza	kqx1155	00667791035
	2	chevet	nks1245	27783525032
	3	ford ka	rte7845	27783525032

cpf	nome	id	veiculo	placa	cpf
00667791035	p2	1	monza	kqx1155	00667791035
27783525032	p4	3	ford ka	rte7845	27783525032
27783525032	p4	2	chevet	nks1245	27783525032
44701230057	p3	NULL	NULL	NULL	NULL
81489134042	p1	NULL	NULL	NULL	NULL

Right join, Right outer join

Similar ao Left join. Ao contrário do LEFT JOIN, a cláusula RIGHT JOIN ou RIGHT OUTER JOIN retorna todos os dados encontrados na tabela à direita de JOIN. Caso não existam dados associados entre as tabelas à esquerda e à direita de JOIN, serão retornados valores nulos.

```
1 SELECT * FROM VEICULOS RIGHT JOIN PESSOAS ON PESSOAS.CPF = VEICULOS.CPF;
```

Pessoas

	cpf	nome
▶	00667791035	p2
	27783525032	p4
	44701230057	p3
	81489134042	p1

Veículos

	id	veiculo	placa	cpf
▶	1	monza	kqx1155	00667791035
	2	chevet	nks1245	27783525032
	3	ford ka	rte7845	27783525032

id	veiculo	placa	cpf	cpf	nome
1	monza	kqx1155	00667791035	00667791035	p2
3	ford ka	rte7845	27783525032	27783525032	p4
2	chevet	nks1245	27783525032	27783525032	p4
NULL	NULL	NULL	NULL	44701230057	p3
NULL	NULL	NULL	NULL	81489134042	p1

Self join

SELF JOIN vai funcionar exatamente como um INNER JOIN, porém, como o próprio nome diz, a junção é feita na mesma tabela.



```
1 SELECT A.NOME, B.NOME AS INDICADO_POR FROM PESSOAS A JOIN PESSOAS B ON A.INDICADO = B.CPF;
```

Pessoas

cpf	nome	indicado	estado
00667791035	p2	27783525032	RJ
27783525032	p4	NULL	SP
44701230057	p3	81489134042	RS
81489134042	p1	NULL	RN

NOME	INDICADO_POR
p2	p4
p3	p1

transforme ■ se

O conhecimento é o poder
de transformar o seu futuro.