

Utilizando JOINS

Utilizar a cláusula WHERE para fazer seus JOINS (relacionamentos), limita os relacionamentos a apenas um tipo deles, o INNER JOIN.

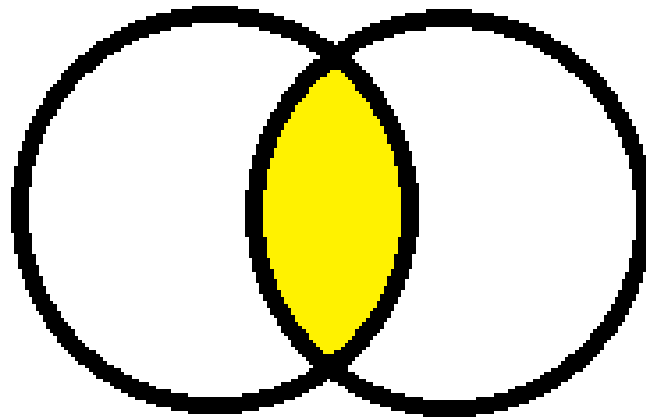
Vamos iniciar com três tipos de Joins:

- INNER JOIN
- LEFT JOIN
- RIGHT JOIN

INNER JOIN

Retorna apenas as linhas das tabelas que sejam comuns entre si, ou seja, as linhas em ambas as tabelas que possuam o campo de relacionamento com o mesmo valor.

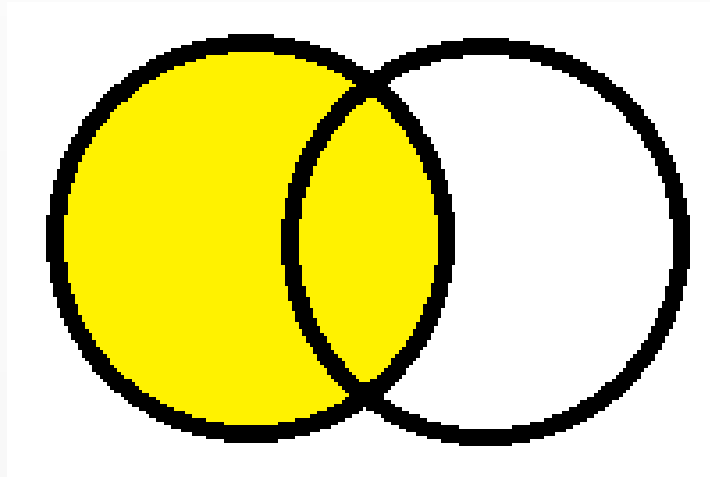
No exemplo criado em aulas anteriores com as tabelas cliente e conta relacionadas, somente as pessoas que possuem contas são exibidas.



LEFT JOIN

Irá listar todas as linhas da primeira tabela relacionada no join, logo após a cláusula from.

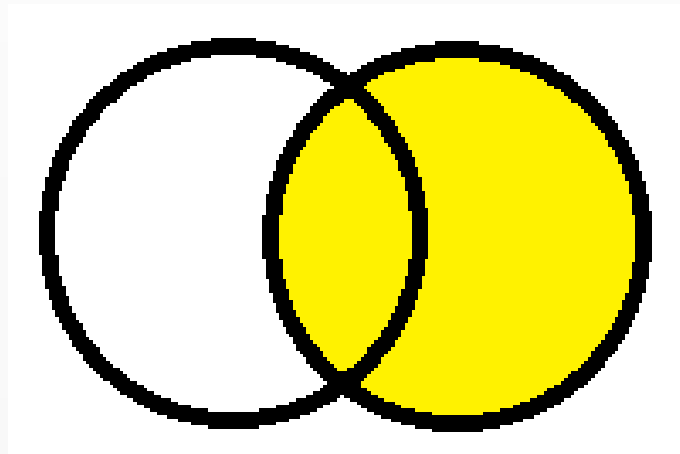
Quando a linha listada não possuir equivalência na tabela destino , as colunas da tabela destino aparecerão com valores nulos



RIGHT JOIN

Irá listar todas as linhas referentes à segunda tabela relacionada no join

Neste caso também , quando a linha listada não possuir equivalência na tabela destino , as colunas da tabela destino aparecerão com valores nulos



Exemplos

Crie as seguintes tabelas:

Cli

Com os campos :

Codigo - inteiro - auto numeração – chave primária

Nome – char (30)

Lance os seguintes registros:

Codigo	Nome
1	José
2	Elisio
3	Roberto
4	Guilherme

Pedido

com os campos:

nr - inteiro – chave primária

cliente – inteiro- chave estrangeira

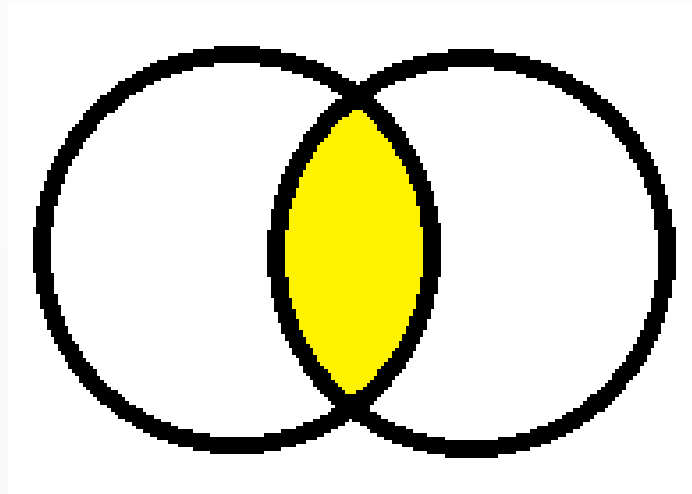
valor – float(5,2)

Lance os seguintes registros:

nr	Cliente	valor
1	2	100.50
2	2	120.00
3	1	20.00
4	3	60.00
5	3	110.00

INNER JOIN

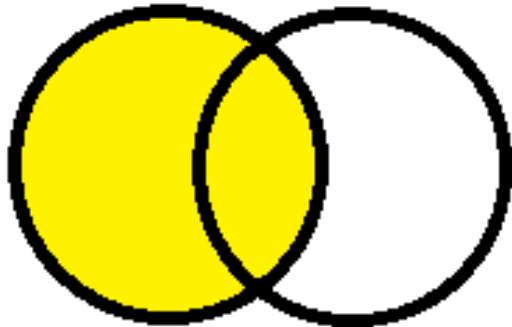
```
Select cli.nome, pedido.nr, pedido.valor  
From pedido inner join cli  
On (pedido.cliente = cli.codigo);
```



Observe que o cliente Guilherme não fez nenhum pedido. Se você quiser um relatório que mostre também os clientes que não fizeram nenhum pedido, terá que usar um left join:

```
mysql> select pedido.nr,cli.nome,pedido.valor  
-> from cli left join pedido  
-> on cli.codigo=pedido.cliente;
```

nr	nome	valor
3	José	20.00
1	Elisio	100.50
2	Elisio	120.00
4	Roberto	60.00
5	Roberto	110.00
NULL	Guilherme	NULL



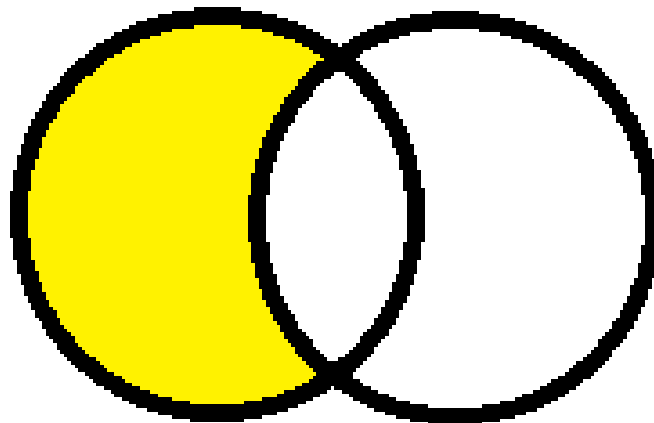
Veja que agora o cliente que não fez nenhum pedido, no caso o Guilherme, também foi exibido. Observe que com left join você deve usar on no lugar de where.

Se quiser exibir apenas os clientes que não fizeram nenhum pedido, use algo como:

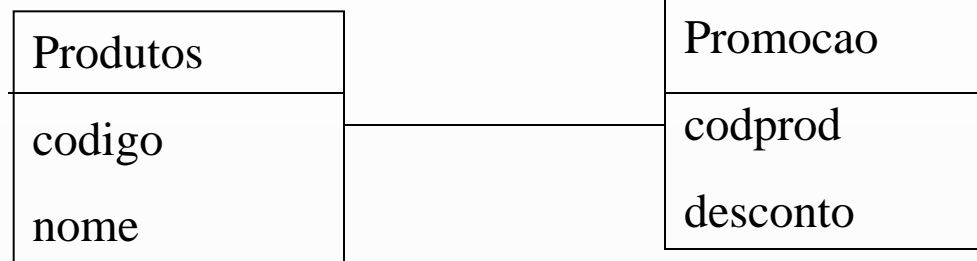
```
mysql> select pedido.nr,cli.nome,pedido.valor  
-> from cli left join pedido  
-> on cli.codigo=pedido.cliente  
-> where pedido.nr is null;
```

nr	nome	valor
NULL	Guilherme	NULL

1 row in set (0.00 sec)



Exercício



codigo	nome
1	caderno
2	caneta
3	estojo
4	lapiseira

codprod	desconto
1	5
3	7

Exercício

1. Quero saber o nome dos produtos que estão em promoção.
2. Quero uma lista de nome dos produtos e, quando houver, mostrar também o desconto.

Correção

1. Quero saber o nome dos produtos que estão em promoção.

```
mysql> select produtos.nome  
      -> from produtos inner join promocao  
      -> on produtos.codigo=promocao.codprod;  
+-----+  
| nome  |  
+-----+  
| caderno |  
| estojo  |  
+-----+
```

2. Quero uma lista de nome dos produtos e mostrar também o desconto.

```
mysql> select produtos.nome, promocao.desconto  
      -> from produtos inner join promocao  
      -> on produtos.codigo=promocao.codprod;  
+-----+-----+  
| nome  | desconto |  
+-----+-----+  
| caderno |        5 |  
| estojo  |        7 |  
+-----+-----+
```

Se quiser exibir todos os produtos devo usar nesse caso o left join pois com inner join só lista os que tem desconto.

Aliases para Tabelas

- Quando usamos Join, o nome da tabela é citado para diferenciar a qual campo se está fazendo referência. Quando a consulta é complexa e envolve várias tabelas, referenciar o nome da tabela pode aumentar muito o tamanho da consulta e em algumas ferramentas como Delphi, há um limite de 255 caracteres para a consulta.
- Para criar uma alias para uma tabela, basta acrescentar um identificador à frente do nome da tabela. A partir de então, basta utilizar este alias para se referenciar a tabela.
- `Select nome,nr,valor from pedido p inner join cli c on (p.cliente=c.codigo);`



Exercícios

Criar a tabela Proprietários e a tabela Carros:

- create table proprietarios (rg char (16) primary key, nome char(40));
- create table carros (renavam char (12) primary key, modelo char(20), marca char(20), cor char(10), rg char(16));
- Alter table carros add foreign key(rg) references proprietarios(rg);

Insira os seguintes valores para proprietários:

rg	nome
-----	-----
123456789	João da Silva
654123987	Maria de Oliveira
987654321	José de Souza

Insira os seguintes valores para carros:

renavam	modelo	marca	cor	rg
-----+-----+-----+-----+				
123456789123	Fiesta	Ford	Prata	123456789
123456789124	Palio	Fiat	Vermelho	123456789
123456789125	Corsa	Chevrolet	Amarelo	987654321
123456789126	Gol	Volkswagen	Branco	987654321

Listar o Renavam, modelo, marca, cor e nome do proprietário de todos os carros.

Select renavam,modelo,marca,cor,nome as
proprietario from carros,proprietários where
carros.rg=proprietarios.rg

ou

Select renavam,modelo,marca,cor,nome as
proprietario from carros inner join proprietários on
(carros.rg=proprietarios.rg);

renavam	modelo	marca	cor	Proprietario
123456789123	Fiesta	Ford	Prata	João da Silva
123456789124	Palio	Fiat	Vermelho	João da Silva
123456789125	Corsa	Chevrolet	Amarelo	José de Souza
123456789126	Gol	Volkswagen	Branco	José de Souza

- Quando queremos recuperar todas as linhas de uma tabela, inclusive aquelas que não possuem linhas equivalentes na outra, trabalhamos com Right / Left Join

```
mysql> select renavam,modelo,marca,cor,nome as Proprietario
-> from carros c right join proprietarios p
-> on (c.rg=p.rg);
```

renavam	modelo	marca	cor	Proprietario
123456789123	Fiesta	Ford	Prata	João da Silva
123456789124	Palio	Fiat	Vermelho	João da Silva
				Maria de Oliveira
123456789125	Corsa	Chevrolet	Amarelo	José de Souza
123456789126	Gol	Volkswagen	Branco	José de Souza

- Observe que foram listados todos os proprietários (pois a tabela à direita (right) do join na linha de comando é proprietários), independente de terem um carro relacionado a ele ou não.

- Observe que nesse exemplo específico, se usarmos o left join, como a tabela de carros é a da esquerda(left) e todos os carros tem um proprietário, a resposta é igual à do inner

```
mysql> select renavam,modelo,marca,cor,nome as Proprietario
-> from carros c left join proprietarios p
-> on (c.rg=p.rg);
```

renavam	modelo	marca	cor	Proprietario
123456789123	Fiesta	Ford	Prata	João da Silva
123456789124	Palio	Fiat	Vermelho	João da Silva
123456789125	Corsa	Chevrolet	Amarelo	José de Souza
123456789126	Gol	Volkswagen	Branco	José de Souza

- Se houvesse carros cadastrados sem proprietário, eles apareceriam listados com o campo proprietário vazio.

Exercícios

- Exiba quantos carros tem cada proprietário.
- Observe essa realidade de dados para montar a consulta.

renavam	modelo	marca	cor	Proprietario
123456789123	Fiesta	Ford	Prata	João da Silva
123456789124	Palio	Fiat	Vermelho	João da Silva
123456789125	Corsa	Chevrolet	Amarelo	Maria de Oliveira
123456789126	Gol	Volkswagen	Branco	José de Souza

Exercícios

- Exiba quantos carros tem cada proprietário.

```
| proprietario | carros |
+-----+-----+
| José de Souza | 2      |
| João da Silva  | 2      |
| Maria de Oliveira | 0      |
+-----+-----+
3 rows in set (0.00 sec)
```

Exercícios

- Exiba quantos carros tem cada proprietário.

*Select p.nome as proprietario,
count(renavam) as carros from
proprietarios p left join carros c on
(p.rg=c.rg) group by p.nome;*

proprietario	carros
José de Souza	2
João da Silva	2
Maria de Oliveira	0

3 rows in set (0.00 sec)

Exercícios

- Agora quero exibir quantos carros tem cada proprietário que possui carros, ou seja, quem não possui nenhum carro não deve ser exibido.

Exercícios

Select p.nome as proprietario, count() as
carros from proprietarios p inner join
carros c on (p.rg=c.rg) group by p.nome;*

proprietario	carros
José de Souza	2
João da Silva	2

- Voltando agora a trabalhar com as tabelas já feitas anteriormente:

C

codigo	nome
1	José
2	Elisio
3	Roberto
4	Guilherme
5	Elaine
6	Maria
7	Thais

nr	cliente	valor
1	2	100.50
2	2	120.00
3	1	20.00
4	3	60.00
5	3	110.00

Para se certificar de ter essas tabelas execute o comando `show tables`; e depois os comandos `select *` para cada uma delas. Acrescente os registros que ainda faltam na sua tabela

Execute o comando:

```
mysql> select cli.nome,pedido.nr,pedido.valor  
-> from pedido inner join cli  
-> on pedido.cliente=cli.codigo;
```

nome	nr	valor
Elisio	1	100.50
Elisio	2	120.00
José	3	20.00
Roberto	4	60.00
Roberto	5	110.00

Exercícios:

Exibir o total dos pedidos por cliente:

Para o exercício de cliente e conta :

- Exiba o nome do cliente que possui a menor conta
- Exiba a cidade dos cliente que possuem conta acima do valor médio das contas

Para exibir o nome do cliente, que possui a menor conta:

```
Select nome from cliente inner join conta on  
(cliente.codigo= conta.codcli) where (valor=(select  
min(valor) from conta ));
```

Exiba a cidade dos cliente que possuem conta acima do valor médio das contas

*Select distinct(cidade) from cliente inner join conta
on (cliente.codigo= conta.codcli) where
(valor>(select avg(valor) from conta));*

Múltiplas Tabelas

- Primeiro vamos criar três tabelas no banco de dados: funcionarios, pagamentos e descontos.

Funcionarios com os atributos
codigo_funcionario e nome

Pagamentos com os atributos
codigo_pagto, codigo_funcionario e
valor

Descontos com os atributos
codigo_desconto, codigo_funcionario e
valor

Não se esqueça de definir as chaves primárias
e estrangeiras.

Inserindo dados...

- Funcionarios

- 1 Luis

- 2 Marina

- 3 Letícia

- 4 Gustavo

- 5 Mateus

Inserindo dados...

- Pagamentos

	codPag	codFunc	Valor
--	--------	---------	-------

1	1	100
---	---	-----

2	1	200
---	---	-----

3	3	300
---	---	-----

4	5	400
---	---	-----

5	5	500
---	---	-----

Inserindo dados...

- Descontos

codDesc	codFunc	Valor
---------	---------	-------

1	1	50
---	---	----

2	2	20
---	---	----

3	5	30
---	---	----

Exemplo de Inner Join

```
select f.nome, p.valor as pagamento  
from funcionarios f INNER JOIN pagamentos p  
ON f.codigo_funcionario = p.codigo_funcionario;
```

	nome	pagamento
1	Luis	100.00
2	Luis	200.00
3	Letícia	300.00
4	Mateus	400.00
5	Mateus	500.00

- Apesar de termos cinco funcionários na tabela, ele mostrou apenas três, o motivo é que apenas estes três tem pagamentos. Veja que o inner join fez uma junção entre funcionarios e pagamentos e desconsiderou os funcionários sem pagamentos.

Inner join com três tabelas

- select f.nome, p.valor as pagamento,
d.valor as desconto
from funcionarios f

INNER JOIN pagamentos p

ON f.codigo_funcionario = p.codigo_funcionario

INNER JOIN descontos d

ON f.codigo_funcionario = d.codigo_funcionario

	nome	pagamento	desconto
1	Luis	100.00	50.00
2	Luis	200.00	50.00
3	Mateus	400.00	30.00
4	Mateus	500.00	30.00

- Neste caso apenas dois funcionários foram mostrados já que incluímos na consulta os descontos, ou seja, a leitura que esta consulta fez é: mostrar funcionários que tem pagamentos e descontos.

Exemplo de Left join

- ```
select f.nome, p.valor as pagamento
from funcionarios f LEFT JOIN
pagamentos p
ON f.codigo_funcionario= p.codigo_funcionario;
```

|   | nome    | pagamento |
|---|---------|-----------|
| 1 | Luis    | 100.00    |
| 2 | Luis    | 200.00    |
| 3 | Marina  | NULL      |
| 4 | Letícia | 300.00    |
| 5 | Gustavo | NULL      |
| 6 | Mateus  | 400.00    |
| 7 | Mateus  | 500.00    |

- Os funcionários 3 e 5 não tem pagamentos, mas ainda assim eles apareceram na consulta, já que a função Left Join considera apenas a coluna da esquerda e retorna Null (nulo) quando a coluna da direita não tiver um valor correspondente.

# Incluindo o desconto...

- select f.nome, p.valor as pagamento,  
d.valor as desconto  
from funcionarios f LEFT JOIN  
pagamentos p  
ON f.codigo\_funcionario= p.codigo\_  
funcionario  
LEFT JOIN descontos d  
ON f.codigo\_funcionario= d.codigo\_  
funcionario



|   | nome    | pagamento | desconto |
|---|---------|-----------|----------|
| 1 | Luis    | 100.00    | 50.00    |
| 2 | Luis    | 200.00    | 50.00    |
| 3 | Marina  | NULL      | 20.00    |
| 4 | Letícia | 300.00    | NULL     |
| 5 | Gustavo | NULL      | NULL     |
| 6 | Mateus  | 400.00    | 30.00    |
| 7 | Mateus  | 500.00    | 30.00    |

- O que fizemos foi uma espécie de left join em cascata e é útil quando queremos partir de uma base (funcionarios) e listar todas as correspondências ou não das tabelas (pagamentos e descontos) a ela relacionadas

# Extras

- Podemos usar mysql como calculadora!

Digite na linha de comando:

- `Select 2+2;`
- `Select pi();`
- `Select cos(pi());`
- `Select cos(pi()/2);`
- `Select cos(pi())/2;`

- `Select round(pi(),2);` Definindo o número de casa decimais
- `Select round(pi(),10);`
- `Select sqrt(9);`
- `Select abs(-2);`
- `Select mod(5,2);`
- `Select power(3,2);`

# Funções String

- `SELECT CONCAT('My', 'S', 'QL');`
- `SELECT CONCAT_WS(",","First name","Second name","Last Name");`  
-> `'First name,Second name,Last Name'`
- `SELECT REPEAT('MySQL', 3);`  
-> `'MySQLMySQLMySQL'`
- `SELECT CHAR_LENGTH('Elaine');`

# Funções String

- `SELECT SUBSTRING('Unicamp',5);`  
-> 'amp'
- `SELECT SUBSTRING('Unicamp' FROM 4);`  
-> 'camp'
- `SELECT SUBSTRING('Unicamp',5,2);`  
-> 'am'

# Funções String

- `SELECT SUBSTRING_INDEX('www.mysql.com', '.', 2);`  
-> 'www.mysql'
- `SELECT SUBSTRING_INDEX('www.mysql.com', '.', -2);`  
-> 'mysql.com'

Retorna a substring da string 'www.mysql.com' por exemplo antes de 2 ocorrências do delimitador .

Se cont é positivo, tudo a esquerda do delimitador final (contando a partir da esquerda) é retornado. Se cont é negativo, tudo a direita do delimitador final (contando a partir da direita) é retornado.

# Funções String

- `SELECT REVERSE('abc');`  
    -> 'cba'
- `SELECT UCASE('Elaine');`  
    -> 'ELAINE'
- `SELECT Upper('Elaine');`  
    -> 'ELAINE'
- `SELECT LCASE('MYSQL');`  
    -> 'mysql'
- `SELECT Lower('MYSQL');`  
    -> 'mysql'

# Funções Data

- `SELECT DAYOFWEEK ('2007-09-15');`  
-> 7
- `SELECT DAYOFMONTH ('2007-09-15');`  
-> 15
- `SELECT MONTH ('2007-09-15');`  
-> 9



# Funções Data

- `SELECT DAYNAME('2007-09-15');`  
-> 'Saturday'
- `SELECT MONTHNAME ('2007-09-15');`  
-> 'September'