Entendendo os Joins no MySQL

hcode

MySQL 5.7

Joins

Categorias

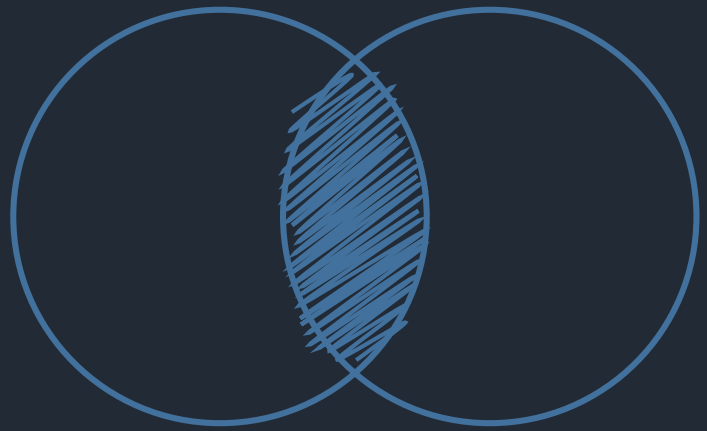
Inner join | Left outer join | Right outer join

Na linguagem SQL, algumas das operações de junções entres tabelas podem ser facilmente demonstradas utilizando a teoria dos conjuntos.

hcode

Intersecção

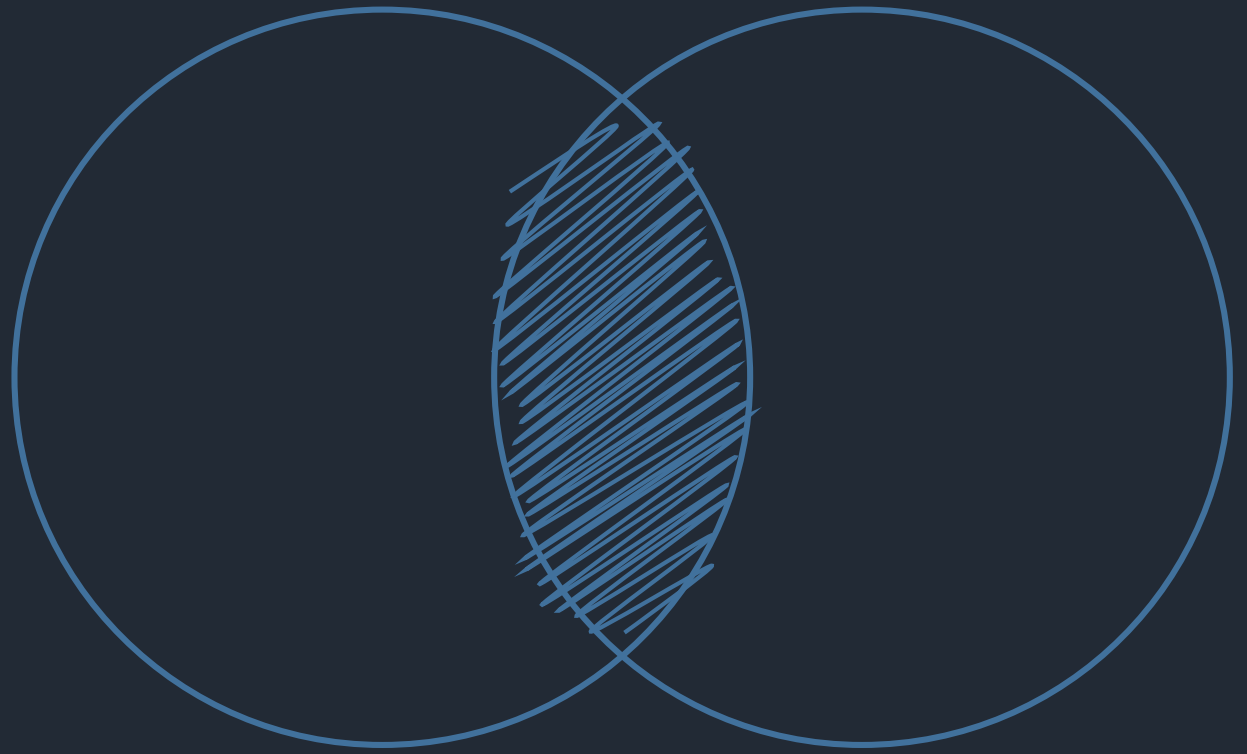
INNER JOIN



hcode

Intersecção – Somente dados que contenha nas duas tabelas

INNER JOIN



hcode

Exemplo:

```
create table tb_funcionario (
    id_func         int,
    nome_func       varchar(200),
    email_func      varchar(250),
    vlr_sal_func    decimal(10,2)
);
```

Exemplo:

```
create table tb_dependentes (
      id_dep          int,
      nome_dep  varchar(200),
      sexo_dep  enum('f','m'),
      id_func   int
constraint fk_dep foreign key (id_func)
references tb_funcionario(id_func)
);
```

No exemplo anterior as tabelas estão relacionadas pela chave id_func, portanto basta realizar uma consulta comparando os dois campos id_func das respectivas tabelas para saber qual funcionário está amarrado a qual dependente, assim, no caso do Inner Join *somente funcionários que possuam* dependentes serão retornados.

```
SELECT a.nome_func, b.nome_dep FROM
tb_funcionario a INNER JOIN tb_dependente b
USING(id_func)
```

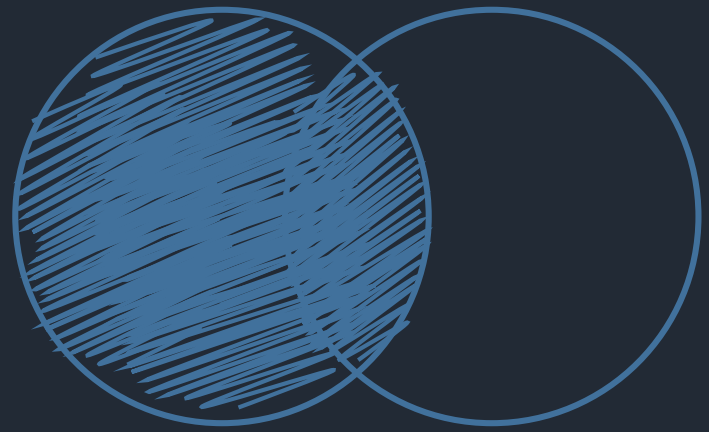
O comando USING é utilizado no MySQL e no MariaDB, nem todos os bancos de dados aceitam esse padrão. Somente quando as duas chaves utilizam o mesmo nome podemos utilizar o comando USING.

```
SELECT a.nome_func, b.nome_dep FROM
tb_funcionario a INNER JOIN tb_dependente b
ON a.id_func = b.id_func
```

As letras a e b são apelidos para a tabela tb_funcionario e tb_dependente respectivamente.

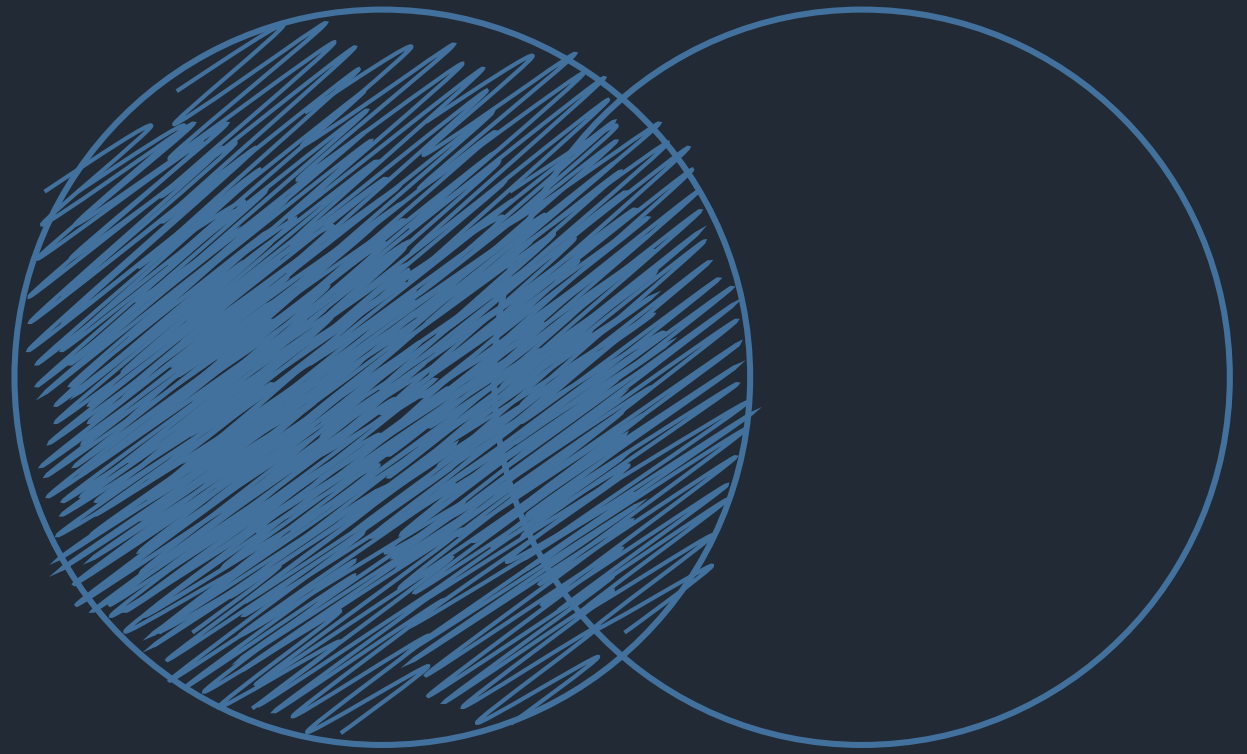
Conheça o

LEFT OUTER JOIN



Todos os dados da tabela localizada ao lado esquerdo da cláusula join. Da tabela do lado direito, somente os dados relacionados.

LEFT OUTER JOIN



hcode

Exemplo:

```
create table tb_funcionario (
	id_func		int,
	nome_func	varchar(200),
	email_func	varchar(250),
	vlr_sal_func	decimal(10,2)
);
```

Exemplo:

```
create table tb_dependentes (
        id_dep          int,
        nome_dep  varchar(200),
        sexo_dep  enum('f','m'),
        id_func   int
constraint fk_dep foreign key (id_func)
references tb_funcionario(id_func)
);
```

No exemplo anterior as tabelas estão relacionadas pela chave id_func, portanto basta realizar uma consulta comparando os dois campos id_func das respectivas tabelas para saber qual funcionário está amarrado a qual dependente, porém no caso do left outer join, todos os funcionários seriam retornados, mesmo que os que não possuem dependentes. Na tabela dependentes somente os dependentes associados seriam retornados, se por alguma razão existisse algum dependente sem associação à tabela funcionário este não seria retornado.

```
SELECT a.nome_func, b.nome_dep FROM
tb_funcionario a LEFT OUTER JOIN tb_dependente b
USING(id_func)
```

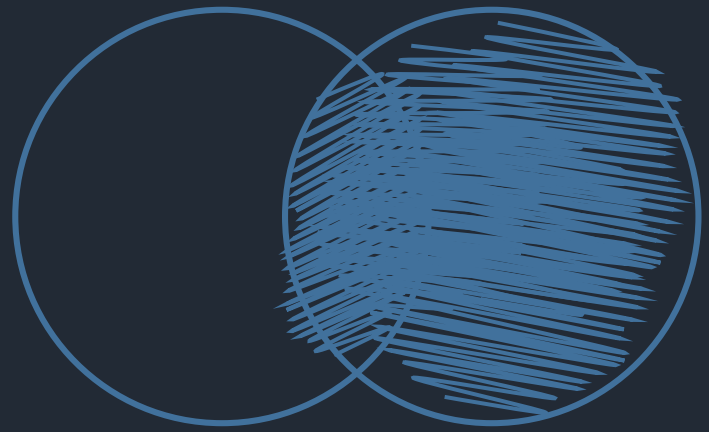
O comando USING é utilizado no MySQL e no MariaDB, nem todos os bancos de dados aceitam esse padrão. Somente quando as duas chaves utilizam o mesmo nome podemos utilizar o comando USING.

```
SELECT a.nome_func, b.nome_dep FROM
tb_funcionario a LEFT OUTER JOIN tb_dependente b
ON a.id_func = b.id_func
```

As letras a e b são apelidos para a tabela tb_funcionario e tb_dependente respectivamente.

Conheça o

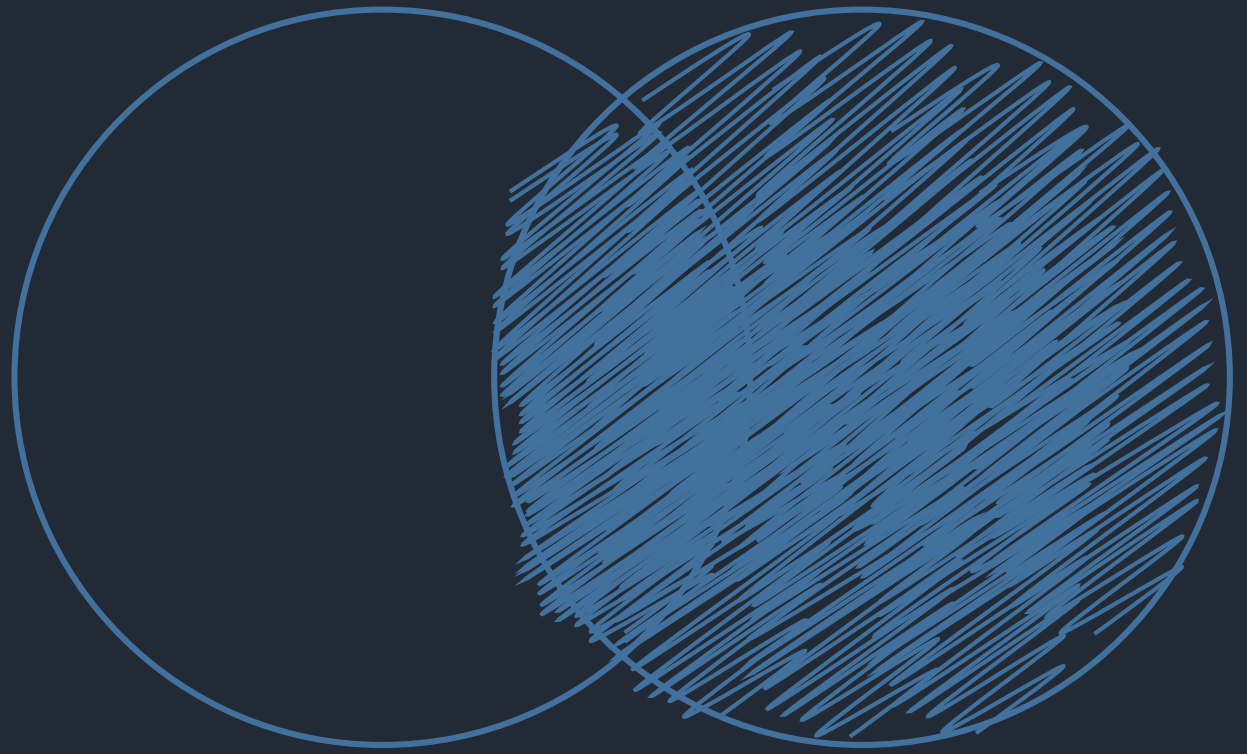
RIGHT OUTER JOIN



hcode

Todos os dados da tabela localizada ao lado direito da cláusula join. Da tabela do lado direito, somente os dados relacionados.

RIGHT OUTER JOIN



hcode

Exemplo:

```
create table tb_funcionario (
	id_func		int,
	nome_func	varchar(200),
	email_func	varchar(250),
	vlr_sal_func	decimal(10,2)
);
```

Exemplo:

```
create table tb_dependentes (
      id_dep          int,
      nome_dep  varchar(200),
      sexo_dep  enum('f','m'),
      id_func   int
constraint fk_dep foreign key (id_func)
references tb_funcionario(id_func)
);
```

No exemplo anterior as tabelas estão relacionadas pela chave id_func, portanto basta realizar uma consulta comparando os dois campos id_func das respectivas tabelas para saber qual funcionário está amarrado a qual dependente, porém no caso do right outer join, todos os dependentes seriam retornados, mesmo que por alguma razão algum dependente não possuísse associação a um funcionário. Left e Right Outer Join possuem a mesma ideia em lados diferentes, sua diferença entretanto será notada em consultas que envolvem mais de uma cláusula join.

```
SELECT a.nome_func, b.nome_dep FROM
tb_funcionario a RIGHT OUTER JOIN tb_dependente b
USING(id_func)
```

O comando USING é utilizado no MySQL e no MariaDB, nem todos os bancos de dados aceitam esse padrão. Somente quando as duas chaves utilizam o mesmo nome podemos utilizar o comando USING.

```
SELECT a.nome_func, b.nome_dep FROM
tb_funcionario a RIGHT OUTER JOIN tb_dependente b
ON a.id_func = b.id_func
```

As letras a e b são apelidos para a tabela tb_funcionario e tb_dependente respectivamente.

hcode

MySQL 5.7

Joins

Categorias

Inner join | Left outer join | Right outer join