

Junções de tabelas – join – retorno de dados – consulta

Para trazer apenas os campos necessários da consulta

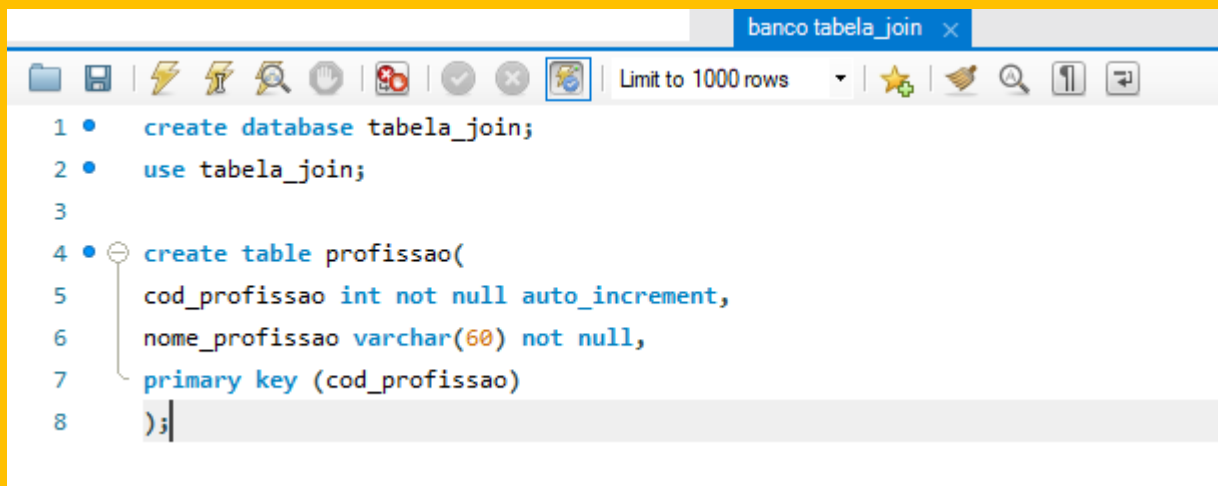
No SELECT utilizamos em vez do nome do campo simples, o nome_tabela.nome_campo;

Em FROM utilizamos os nomes das tabelas que possuem os campos que queremos trazer;

Em WHERE determinamos a dependência das tabelas, lembrando que a relação entre as tabelas é efetuada pela chave estrangeira. Desta forma o WHERE sempre especifica as chaves estrangeiras que ligam as tabelas.

Como exemplo, vamos imaginar que possuímos duas tabelas relacionadas: cliente e profissão.

A tabela profissão contém o código



```
1 • create database tabela_join;
2 • use tabela_join;
3
4 • create table profissao(
5     cod_profissao int not null auto_increment,
6     nome_profissao varchar(60) not null,
7     primary key (cod_profissao)
8 );
```

Tabela cliente armazena os dados pessoais do cliente e tem a tabela profissão como **chave estrangeira foreign key**

```
10 • ○ create table cliente(  
11     cod_cliente int not null auto_increment,  
12     nome_cliente varchar(60) not null,  
13     data_nascimento date,  
14     telefone char (11),  
15     cod_profissao int,  
16     primary key (cod_cliente),  
17     foreign key(cod_profissao)  
18     references profissao (cod_profissao)  
19 );
```

Tabela pedido e tem a tabela cliente como **chave estrangeira foreign key**

```
21 • ○ create table pedido(  
22     num_pedido int not null auto_increment,  
23     cod_cliente int,  
24     total_pedido decimal (10,2),  
25     primary key (num_pedido),  
26     foreign key (cod_cliente)  
27     references cliente (cod_cliente));  
28 |  
29
```

Agora inserir dados nestas tabelas

```
21 • insert into profissao (Cod_profissao, Nome_profissao)
22 values (09, 'programador'),
23         (03, 'analista de banco de dados'),
24         (08, 'suporte'),
25         (02, 'estagiario');
26
27 • insert into cliente (Cod_cliente, Nome_cliente, data_nascimento, telefone)
28 values (2, 'maria barros', '1971-01-22', '3456-7891'),
29         (3, 'jose mendes', '1982-04-29', '3487-9025'),
30         (4, 'rogerio cavalcante', '1990-01-12', '2390-3026');
--

42 • insert into pedido (num_pedido, cod_cliente, total_pedido)
43 values (100, '3', '1200.00'),
44         (204, '2', '900.00'),
45         (203, '4', '800.00');
```

Para visualizarmos todos os dados contidos nas duas tabelas após a inserção de dados podemos utilizar:

Comando select * from cliente

Nesta seção iremos utilizar o **select from** para mencionar a tabela

e **where** para especificarmos as ligações entre as tabelas.

Junção interna tabelas –inner join – retorno de dados

Junção interna é caracterizada por uma seleção que retorna apenas os dados que atendem as **condições de junção** ou seja:

Quais as linhas de uma tabela se relacionam com as linhas de outra tabela

Para isso utilizamos a **clausula ON** que é semelhante a **clausula WHERE**, por isso utilizamos a palavra **inner join**.

Exemplo de junção interna inner join

```
46
47 • select cliente.nome_cliente, pedido.cod_cliente, pedido.num_pedido
48 from cliente inner join pedido
49 on cliente.cod_cliente = pedido.cod_cliente;
50
51
52
53
54
```

```
21 • create table cliente(
22     cod_cliente int not null auto_increment,
23     nome_cliente varchar(60) not null,
24     data_nascimento date,
25     telefone char (11),
26     cod_profissao int,
27     primary key (cod_cliente),
28     foreign key(cod_profissao)
29     references profissao (cod_profissao)
30 );
```

```
21 • create table pedido(
22     num_pedido int not null auto_increment,
23     cod_cliente int,
24     total_pedido decimal (10,2),
25     primary key (num_pedido),
26     foreign key (cod_cliente)
27     references cliente (cod_cliente));
28
```

Result Grid

nome_cliente	cod_cliente	num_pedido
maria barros	2	204
jose mendes	3	100
rogerio cavalcante	4	203

Junção externa – outer join

Uma junção externa é uma seleção que **não requer** que os registros de uma tabela possuam registros equivalentes em outra.

Geralmente os registros não possuem correspondência, exemplo tabela da esquerda, tabela da direita ou ambas.

```
59 • select * from cliente
60 right outer join profissao
61 on cliente.cod_profissao = profissao.cod_profissao;
62
```

	cod_cliente	nome_cliente	data_nascimento	telefone	cod_profissao	cod_profissao	nome_profissao
▶	NULL	NULL	NULL	NULL	NULL	2	estagiario
	NULL	NULL	NULL	NULL	NULL	3	analista de banco de dados
	NULL	NULL	NULL	NULL	NULL	8	suporte
	NULL	NULL	NULL	NULL	NULL	9	programador

Exemplo utilizando o comando left outer join

```
51 • select distinct * from cliente
52 left outer join profissao
53 on cliente.cod_profissao = profissao.cod_profissao;
54
```

	cod_cliente	nome_cliente	data_nascimento	telefone	cod_profissao	cod_profissao	nome_profissao
▼	2	maria barros	1971-01-22	3456-7891	NULL	NULL	NULL
	3	jose mendes	1982-04-29	3487-9025	NULL	NULL	NULL
	4	rogerio cavalcante	1990-01-12	2390-3026	NULL	NULL	NULL

Banco tabela_join

-- nome banco comando create database tabela_join.

create database tabela_join;

-- comando use para se certificar que esta trabalhando no banco criado

use tabela_join;

-- comando create table para criar tabela

-- neste caso iremos criar a tabela profissao, cliente e pedido

create table profissao(

cod_profissao int not null auto_increment,

nome_profissao varchar(60) not null,

primary key (cod_profissao)

);

create table cliente(

cod_cliente int not null auto_increment,

nome_cliente varchar(60) not null,

data_nascimento date,

telefone char (11),

cod_profissao int,

primary key (cod_cliente),

foreign key(cod_profissao)

```
references profissao (cod_profissao)
```

```
);
```

```
create table pedido(
```

```
num_pedido int not null auto_increment,
```

```
cod_cliente int,
```

```
total_pedido decimal (10,2),
```

```
primary key (num_pedido),
```

```
foreign key (cod_cliente)
```

```
references cliente (cod_cliente));
```

```
-- comando insert para inserir dados nas tabelas criadas anteriormente
```

```
-- insert tabela profissao
```

```
insert into profissao (Cod_profissao, Nome_profissao)
```

```
values (09, 'programador'),
```

```
        (03, 'analista de banco de dados'),
```

```
        (08, 'suporte'),
```

```
        (02, 'estagiario');
```

```
-- insert tabela cliente
```

```
insert into cliente (Cod_cliente, Nome_cliente, data_nascimento, telefone)
```

```
values (2, 'maria barros', '1971-01-22', '3456-7891'),
```

```
        (3, 'jose mendes', '1982-04-29', '3487-9025'),
```

```
(4, 'rogerio cavalcante', '1990-01-12', '2390-3026');
```

```
-- insert tabela pedido
```

```
insert into pedido (num_pedido,cod_cliente, total_pedido)
```

```
values (100,'3','1200.00'),
```

```
        (204,'2','900.00'),
```

```
        (203,'4','800.00');
```

```
-- utilizando inner join condição de junção .
```

-- Junção interna é caracterizada por uma seleção que retorna apenas os dados que atendem as condições de junção ou seja:

-- Quais as linhas de uma tabela se relacionam com as linhas de outra tabela

-- Para isso utilizamos a clausula ON que é semelhante a clausula WHERE, por isso utilizamos a palavra inner join.

```
select cliente.nome_cliente, pedido.cod_cliente, pedido.num_pedido
```

```
from cliente inner join pedido
```

```
on cliente.cod_cliente = pedido.cod_cliente;
```

```
-- right outer join a direita
```

-- Uma junção externa é uma seleção que não requer que os registros de uma tabela possuam registros equivalentes em outra.

-- Geralmente os registros não possuem correspondência, exemplo tabela da esquerda, tabela da direita ou ambas.


```
select * from cliente
```

```
right outer join profissao
```

```
on cliente.cod_profissao = profissao.cod_profissao;
```

```
-- left outer join a esquerda
```

```
-- comando que irá mostrar os registro a esquerda
```

```
select distinct * from cliente
```

```
left outer join profissao
```

```
on cliente.cod_profissao = profissao.cod_profissao;
```

Right outer join

Retorna todos os dados da direita da tabela. Exemplo:

```
create table profissao(  
  cod_profissao int not null auto_increment  
  nome_profissao varchar(60) not null,  
  primary key (cod_profissao)  
);
```

```
create table cliente(  
    cod_cliente int not null auto_increment,  
    nome_cliente varchar(60) not null,  
    data_nascimento date,  
    telefone char (11),  
    cod_profissao int,  
    primary key (cod_cliente),  
    foreign key(cod_profissao)  
    references profissao (cod_profissao)  
);
```

```
50
51 • select * from cliente
52 right outer join profissao
53 on cliente.cod_profissao = profissao.cod_profissao;
```

cod_cliente	nome_cliente	data_nascimento	telefone	cod_profissao	cod_profissao	nome_profissao
NUL	NUL	NUL	NUL	NUL	2	estagiario
NUL	NUL	NUL	NUL	NUL	3	analista de banco de dados
NUL	NUL	NUL	NUL	NUL	8	suporte
NUL	NUL	NUL	NUL	NUL	9	programador