

SQL

Structured Query Language

Linguagem de Consulta Estruturada

INNER JOIN, ORDER BY E GROUP BY

AULA 5

Banco zoologico

v zoologico animais	
🔑	codigo : int
📄	nome : varchar(80)
📄	sexo : varchar(1)
#	codigo_especie : int

v zoologico especies	
🔑	codigo : int
📄	nome : varchar(80)

Tabela especies

```
CREATE TABLE `especies` (  
    `codigo` int NOT NULL,  
    `nome` varchar(80) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;  
ALTER TABLE `especies`  
    ADD PRIMARY KEY (`codigo`);  
ALTER TABLE `especies`  
    MODIFY `codigo` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=1;
```

Tabela animais

```
CREATE TABLE `animais` (  
  `codigo` int NOT NULL,  
  `nome` varchar(80) DEFAULT NULL,  
  `sexo` varchar(1) DEFAULT NULL,  
  `codigo_especie` int DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;  
ALTER TABLE `animais`  
  ADD PRIMARY KEY (`codigo`);  
ALTER TABLE `animais`  
  MODIFY `codigo` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=1;
```

Exemplo de registros da tabela especies

codigo	nome
1	jabuti
2	coelho
3	onça
4	cobra
5	macaco

Exemplo de registros da tabela animais

codigo	nome	sexo	codigo_especie
1	jackson	M	1
3	carlos	M	2
4	felipe	M	3
5	mariana	F	4
6	douglas	M	5
7	miguel	M	1
8	damares	F	2
9	galileu	M	3
10	nestor	M	4
11	rosinha	F	5

INNER JOIN

União de campos de duas tabelas por meio da chave primária de uma e da chave estrangeira da outra.

Sintaxe geral:

```
SELECT [CAMPOS] FROM [TABELA 1] INNER JOIN  
[TABELA 2] ON [CHAVE ESTRANGEIRA DA  
TABELA 1] = [CHAVE PRIMARIA DA TABELA 2];
```

INNER JOIN

Consulta para mostrar o código, nome do animal, sexo e o nome da espécie.

```
SELECT animais.codigo,  
animais.nome, sexo, especies.nome as especie  
FROM animais INNER JOIN especies ON  
animais.codigo_especie = especies.codigo;
```


INNER JOIN

Se as duas tabelas têm campos com o mesmo nome, é necessário colocar o nome da tabela antes do nome do campo.

```
SELECT animais.codigo,  
animais.nome, sexo, especies.nome as especie  
FROM animais INNER JOIN especies ON  
animais.codigo_especie = especies.codigo;
```

INNER JOIN

Podemos dar apelidos para um campo, para diferenciá-lo de seu homônimo, ou apresentar o dado com uma descrição mais compreensível no contexto em que os resultados serão apresentados.

```
SELECT animais.codigo,  
animais.nome, sexo, especies.nome as especie FROM  
animais INNER JOIN especies ON  
animais.codigo_especie = especies.codigo;
```

Exemplo de saída

codigo	nome	sexo	especie
1	jackson	M	jabuti
3	carlos	M	coelho
4	felipe	M	onça
5	mariana	F	cobra
6	douglas	M	macaco
7	miguel	M	jabuti
8	damares	F	coelho
9	galileu	M	onça
10	nestor	M	cobra
11	rosinha	F	macaco

INNER JOIN

Consulta para mostrar o código, nome do animal, sexo e o nome da espécie quando o nome da espécie for 'coelho', com os registros ordenados pelo nome da espécie.

```
SELECT animais.codigo,  
animais.nome,sexo,especies.nome as especie FROM  
animais INNER JOIN especies ON  
animais.codigo_especie = especies.codigo WHERE  
especies.nome = 'coelho' ORDER BY especies.nome;
```

Exemplo de saída

codigo	nome	sexo	especie
3	carlos	M	coelho
8	damares	F	coelho

INNER JOIN

Consulta para mostrar o nome da espécie e a quantidade de animais da espécie com os registros ordenados pelo nome da espécie.

```
SELECT especies.nome as especie,  
count(animais.codigo) as quantidade from animais  
INNER JOIN especies on animais.codigo_especie =  
especies.codigo GROUP BY especie ORDER BY especie;
```

Exemplo de saída

especie ▲ 1	quantidade
cobra	2
coelho	2
jabuti	2
macaco	2
onça	2

REFERÊNCIAS

- Documentação oficial do MySQL:
<https://dev.mysql.com/doc/>
- Tutorial de MySQL:
<https://www.w3schools.com/mysql/default.asp>