

BANCO DE DADOS

Linguagem SQL

Consultas Combinadas

SELECT - JOIN

SQL (STRUCTURED QUERY LANGUAGE)

Esquema para o modelo Relacional do contexto didático: **EMPRESA**.

FUNCIONARIO (ident, nome, sobrenome, endereco, dtnasc, salario, sexo, supident, dnumero)

DEPENDENTE (fident, nome, dt_nasc, sexo, relacionamento)

DEPARTAMENTO (numero, nome, gident, dtinicio)

LOCALIZACOES (dnumero, localizacao)

PROJETO (numero, nome, localizacao, dnumero)

TRABALHA_EM (pnumero, fident, horas)

Algumas simplificações nos nomes de variáveis foram realizadas para facilitar o uso delas nas consultas. A chave primária, em cada relação, está sublinhada.

JOIN

- A cláusula **JOIN** permite que sejam combinados dados de duas ou mais tabelas em apenas um único comando SELECT.
- O **JOIN** permite juntar dados que foram para entidade/tabelas separadas a partir do uso do modelo relacional.
- Dominar o modelo do projeto de Banco de Dados (principalmente as chaves primárias e estrangeiras) é muito importante para realizar consultas diferenciadas e consistentes.

FUNCIONARIO

ident	nome	sobrenome	endereco	dtnasc	salario	sexo	supident	dnumero
1163	Claudia	Morais	Rua A – 1 --SP	12/08/1974	5.558,00	F	NULL	4
1164	Jorge	Vila Verde	Rua M – 25 -- SP	29/01/1986	1.550,00	M	1163	1
1165	Moacir	Junqueira	Rua F – 57 -- SP	08/11/1981	1.550,00	M	1164	1
1166	Patrícia	Sorte	Rua F –22 -- SP	22/06/1979	2.200,00	F	1163	4
1167	Caio	Brotas	Rua I –100 -- SP	15/08/1977	3.400,00	M	1163	3
1170	Antonio	Monde	Rua X - 5 -- SP	23/03/1985	2.300,00	M	1167	NULL

DEPARTAMENTO

<u>numero</u>	nome	gident	dtinicio
1	Informática	1164	01/01/2010
3	Engenharia	1163	10/08/2000
4	Recursos Humanos	1167	07/05/2015
5	Logística	NULL	19/03/2021

NATURAL JOIN

- Natural JOIN, CROSS JOIN ou simplesmente JOIN é o nome dado a um produto cartesiano entre duas ou mais tabelas em um comando SELECT.
- Os comandos a seguir são exatamente iguais:

```
SELECT f.ident, f.nome, d.numero, d.nome  
FROM funcionario f JOIN departamento d
```

```
SELECT f.ident, f.nome, d.numero, d.nome  
FROM funcionario f  
CROSS JOIN departamento d
```

SELECT f.ident, f.nome, f.dnumero, d.numero, d.nome
FROM funcionario f JOIN departamento d

	ident	f.nome	dnumero	numero	d.nome
1	1164	Jorge	1	5	Logística
2	1164	Jorge	1	4	Recursos Humanos
3	1164	Jorge	1	3	Engenharia
4	1164	Jorge	1	1	Informática
5	1165	Moacir	1	5	Logística
6	1165	Moacir	1	4	Recursos Humanos
7	1165	Moacir	1	3	Engenharia
8	1165	Moacir	1	1	Informática
9	1166	Patrícia	4	5	Logística
10	1166	Patrícia	4	4	Recursos Humanos
11	1166	Patrícia	4	3	Engenharia
12	1166	Patrícia	4	1	Informática
13	1167	Caio	3	5	Logística
14	1167	Caio	3	4	Recursos Humanos
15	1167	Caio	3	3	Engenharia
16	1167	Caio	3	1	Informática
17	1163	Claudia	4	5	Logística
18	1163	Claudia	4	4	Recursos Humanos
19	1163	Claudia	4	3	Engenharia
20	1163	Claudia	4	1	Informática
21	1170	Antonio	<null>	5	Logística
22	1170	Antonio	<null>	4	Recursos Humanos
23	1170	Antonio	<null>	3	Engenharia
24	1170	Antonio	<null>	1	Informática

INNER JOIN

- Ao combinar tabelas, espera-se o cruzamento entre elas de forma que haja coincidência entre os valores, por meio de relações entre as chaves. O INNER JOIN executa o cruzamento com essas relações, que devem ser explicitadas pela cláusula ON. A seguir veja como implementar:

```
SELECT f.ident, f.nome, d.numero, d.nome  
FROM funcionario f INNER JOIN departamento d  
ON f.dnumero=d.numero
```

```
SELECT f.ident, f.nome, d.numero, d.nome  
FROM funcionario f JOIN departamento d  
ON f.dnumero=d.numero
```

```
SELECT f.ident, f.nome, f.dnumero, d.numero, d.nome  
FROM funcionario f INNER JOIN departamento d ON  
f.dnumero=d.numero
```

Qual o nome do departamento que cada funcionário trabalha?

	ident	f.nome	dnumero	numero	d.nome
1	1164	Jorge	1	1	Informática
2	1165	Moacir	1	1	Informática
3	1166	Patrícia	4	4	Recursos Humanos
4	1167	Caio	3	3	Engenharia
5	1163	Claudia	4	4	Recursos Humanos

LEFT E RIGHT JOIN (OUTER)

- As cláusulas LEFT e RIGHT, também conhecidas como OUTER (externo), tem como finalidade permitir que registros que tenham **valores nulos nas chaves estrangeiras** possam ser vistos.
- Possibilita listar uma ou mais tabelas do JOIN por completo.
- Há muitas variações para o uso das chamadas junções externas:
 - LEFT JOIN ou LEFT OUTER JOIN
 - RIGHT JOIN ou RIGHT OUTER JOIN
 - FULL OUTER JOIN

```
SELECT f.ident, f.nome, f.dnumero, d.numero, d.nome  
FROM funcionario f LEFT JOIN departamento d ON  
f.dnumero=d.numero
```

Liste TODOS os funcionários indicando qual é o NOME do departamento que atuam.

	ident	f.nome	dnumero	numero	d.nome
1	1164	Jorge	1	1	Informática
2	1165	Moacir	1	1	Informática
3	1166	Patrícia	4	4	Recursos Humanos
4	1167	Caio	3	3	Engenharia
5	1163	Claudia	4	4	Recursos Humanos
6	1170	Antonio	<null>	<null>	<null>

```
SELECT f.ident, f.nome, f.dnumero, d.numero, d.nome  
FROM funcionario f RIGHT JOIN departamento d ON  
f.dnumero=d.numero
```

Liste os funcionários indicando qual é o NOME do departamento que atuam. **Todos departamentos precisam ser listados!**

	ident	f.nome	dnumero	numero	d.nome
1	1165	Moacir	1	1	Informática
2	1164	Jorge	1	1	Informática
3	1167	Caio	3	3	Engenharia
4	1163	Claudia	4	4	Recursos Humanos
5	1166	Patrícia	4	4	Recursos Humanos
6	<null>	<null>	<null>	5	Logística

```
SELECT f.ident, f.nome, f.dnumero, d.numero, d.nome
FROM funcionario f FULL OUTER JOIN departamento d ON
f.dnumero=d.numero
```

Liste TODOS os funcionários indicando qual é o NOME do departamento que atuam. Todos departamentos precisam ser listados!

	ident	f.nome	dnumero	numero	d.nome
1	1164	Jorge	1	1	Informática
2	1165	Moacir	1	1	Informática
3	1166	Patrícia	4	4	Recursos Humanos
4	1167	Caio	3	3	Engenharia
5	1163	Claudia	4	4	Recursos Humanos
6	1170	Antonio	<null>	<null>	<null>
7	<null>	<null>	<null>	5	Logística

MYSQL!

O MySQL não implementa o FULL OUTER JOIN, a solução poderia ser então adaptada:

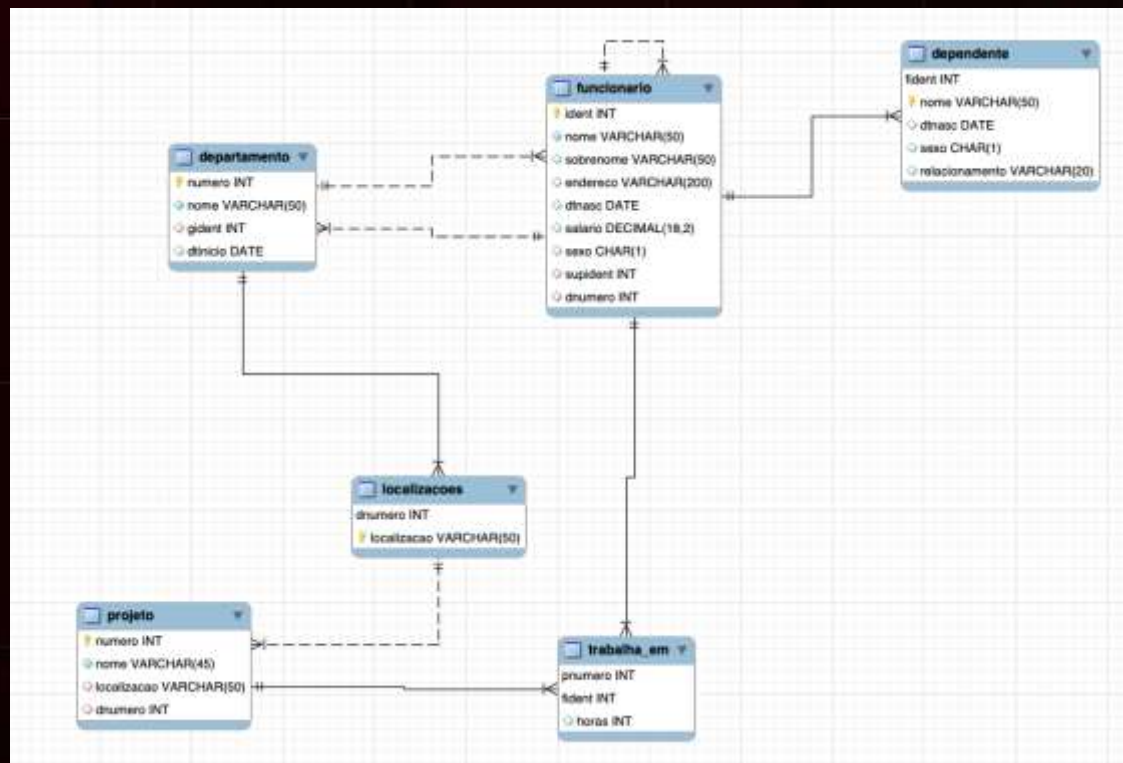
```
SELECT f.ident, f.nome, f.dnumero, d.numero, d.nome FROM  
funcionario f LEFT JOIN departamento d ON f.dnumero=d.numero  
UNION
```

```
SELECT f.ident, f.nome, f.dnumero, d.numero, d.nome  
FROM funcionario f RIGHT JOIN departamento d ON  
f.dnumero=d.numero
```

	ident	f.nome	dnumero	numero	d.nome
1	1164	Jorge	1	1	Informática
2	1165	Moacir	1	1	Informática
3	1166	Patrícia	4	4	Recursos Humanos
4	1167	Caio	3	3	Engenharia
5	1163	Claudia	4	4	Recursos Humanos
6	1170	Antonio	<null>	<null>	<null>
7	<null>	<null>	<null>	5	Logística

JOIN - MAIS DE DUAS TABELAS

- Os bancos de dados relacionais criam ligações que encadeiam tabelas seguidamente. Em várias situações, um conjunto de dados tem origem em mais de uma ou duas tabelas.



PROJETO

numero	nome	localizacao	dnumero
1	Sistema CRM	São Paulo	1
2	Esteira de Big Data	São Paulo	1
3	DRE Financeiro	São Paulo	1
4	Reestrutura do Parque X	São Paulo	3

TRABALHA_EM

pnumero	fident	horas
1	1164	20
1	1165	13
2	1164	10
2	1165	12
4	1167	40


```
SELECT p.nome as Projeto, f.nome as Funcionario, te.horas
FROM trabalha_em te
LEFT JOIN projeto p ON te.pnumero=p.numero
LEFT JOIN funcionario f ON te.fident=f.ident
```

Lista a **quantidade de horas** que cada funcionário trabalha em cada projeto (**mostre o nome do funcionário e o nome do projeto**)

	Projeto	Funcionario	horas
1	Sistema CRM	Jorge	20
2	Sistema CRM	Moacir	13
3	Esteira de Big Data	Jorge	10
4	Esteira de Big Data	Moacir	12
5	Reestrutura do Parque X	Caio	40

OBSERVE QUE PODERIA TER FEITO A ORDENAÇÃO DESEJADA (QUE NÃO FOI APLICADA)

SELF JOIN – MESMA TABELA

```
SELECT f.nome as Funcionario, s.nome as Superior FROM  
funcionario f LEFT JOIN funcionario s ON f.supident=s.ident
```

Apresente o nome dos funcionários da empresa, com o nome do seu superior imediato. Todos funcionários devem ser listados.

	Funcionario	Superior
1	Jorge	Claudia
2	Moacir	Jorge
3	Patrícia	Claudia
4	Caio	Claudia
5	Claudia	<null>
6	Antonio	Claudia

BANCO DE DADOS

Linguagem SQL

Consultas Combinadas

SELECT - JOIN