

Structured Query Language
(Linguagem de busca Estruturada)

SQL

CRIAÇÃO DE TABELA

```
CREATE TABLE Empregado (  
    id SERIAL,  
    matricula VARCHAR(10) NOT NULL,  
    nome TEXT NOT NULL,  
    sobrenome TEXT NOT NULL,  
    PRIMARY KEY(id));
```

```
CREATE TABLE Produto (  
    id SERIAL,  
    nome TEXT NOT NULL,  
    valor MONEY NOT NULL DEFAULT 10.00,  
    PRIMARY KEY(id));
```

01

CRIAÇÃO DE DADOS

Inserir dados em uma tabela

Cuidado com os
valores NOT NULL

INSERT INTO

INSERT INTO tabela [(colunas)] VALUES (valores)

```
INSERT INTO Empregado (matricula, nome) VALUES (9491, 'Ana')
```

```
INSERT INTO Empregado VALUES (9491, 'Ana')
```

Tabela
EMPREGADO com 2
atributos:
Matrícula e Nome

Se as colunas não forem definidas, segue a ordem utilizada na declaração da tabela

```
INSERT INTO Empregado (id, matricula, nome) VALUES (4, 9491, 'Ana')
```

```
INSERT INTO Empregado VALUES (4, 9491, 'Ana')
```

```
INSERT INTO Empregado (matricula, nome) VALUES (9491, 'Ana')
```

Tabela
EMPREGADO com 3
atributos:
ID, Matrícula e
Nome

No caso de um atributo auto incrementável (id), ele pode ser definido automaticamente desde que isso seja declarado.

INSERÇÃO DE DADOS

```
INSERT INTO Empregado (matricula, nome, sobrenome) VALUES ('BAC247', 'Carla', 'Fernandes');  
INSERT INTO Empregado (matricula, nome, sobrenome) VALUES ('DJD778', 'Danilo', 'Curvelo');  
INSERT INTO Empregado (matricula, nome, sobrenome) VALUES ('BAPQIS', 'Daniel', 'Curvelo');  
INSERT INTO Empregado (matricula, nome, sobrenome) VALUES ('38JSSH', 'Alice', 'Curvelo');  
INSERT INTO Empregado (matricula, nome, sobrenome) VALUES ('2JJII9', 'Carla', 'Medeiros');
```

```
INSERT INTO Produto (nome, valor) VALUES ('Nesquik', 2.25);  
INSERT INTO Produto (nome, valor) VALUES ('Todinho', 1.90);  
INSERT INTO Produto (nome, valor) VALUES ('Bisnaguinha', 10.35);  
INSERT INTO Produto (nome, valor) VALUES ('Queijo mussarela', 16.60);  
INSERT INTO Produto (nome, valor) VALUES ('Mortadela', 10.90);
```

02

SELEÇÃO DE DADOS

Ler dados de uma tabela

- *todos os dados

- *dados específicos

SELECT

```
SELECT [DISTINCT] <lista de atributos>  
FROM <lista de tabelas>  
[WHERE <condicao>]  
[ORDER BY <ordenacao>]  
[GROUP BY <grupos>]  
[HAVING <condicao>]
```

FROM

Realiza um produto cartesiano entre as tabelas indicadas

DISTINCT

SQL permite duplicata na tabela e nos resultados das queries
Use DISTINCT para removê-las

WHERE

Indica as condições para limitar os elementos dessa tabela resultante

SELECT

SELECT *
FROM Empregado;

```
mysql> SELECT *  
-> FROM Empregado;  
+-----+  
| id | matricula | nome | sobrenome |  
+-----+  
| 1 | BAC247 | Carla | Fernandes |  
| 2 | DJD778 | Danilo | Curvelo |  
| 3 | BAPQIS | Daniel | Curvelo |  
| 4 | 38JSSH | Alice | Curvelo |  
| 5 | 2JJII9 | Carla | Medeiros |  
+-----+  
5 rows in set (0.00 sec)
```

SELECT nome
FROM Empregado;

```
mysql> SELECT nome  
-> FROM Empregado;  
+-----+  
| nome |  
+-----+  
| Carla |  
| Danilo |  
| Daniel |  
| Alice |  
| Carla |  
+-----+  
5 rows in set (0.00 sec)
```

SELECT DISTINCT nome
FROM Empregado;

```
mysql> SELECT DISTINCT nome  
-> FROM Empregado;  
+-----+  
| nome |  
+-----+  
| Carla |  
| Danilo |  
| Daniel |  
| Alice |  
+-----+  
4 rows in set (0.01 sec)
```


SELECT

```
SELECT nome, sobrenome  
FROM Empregado;
```

```
mysql> SELECT nome, sobrenome  
-> FROM Empregado;  
+-----+-----+  
| nome  | sobrenome |  
+-----+-----+  
| Carla | Fernandes |  
| Danilo | Curvelo  |  
| Daniel | Curvelo  |  
| Alice  | Curvelo  |  
| Carla  | Medeiros |  
+-----+-----+  
5 rows in set (0.00 sec)
```

WHERE

Pode conter AND, OR,
NOT, >, <, =, != entre outros
operadores

```
SELECT nome  
FROM Empregado  
WHERE sobrenome="Curvelo";
```

```
mysql> SELECT nome  
-> FROM Empregado  
-> WHERE sobrenome="Curvelo";  
+-----+  
| nome  |  
+-----+  
| Danilo |  
| Daniel |  
| Alice  |  
+-----+  
3 rows in set (0.00 sec)
```

```
SELECT nome  
FROM Empregado  
WHERE sobrenome="Curvelo" OR sobrenome="Fernandes";
```

```
mysql> SELECT nome  
-> FROM Empregado  
-> WHERE sobrenome="Curvelo" OR sobrenome="Fernandes";  
+-----+  
| nome  |  
+-----+  
| Carla |  
| Danilo |  
| Daniel |  
| Alice  |  
+-----+  
4 rows in set (0.00 sec)
```

SELECT

```
SELECT nome
FROM Produto
WHERE valor > 5.50;
```

```
mysql> SELECT nome
-> FROM Produto
-> WHERE valor > 5.50;
+-----+
| nome          |
+-----+
| Bisnaguinha   |
| Queijo mussarela |
| Mortadela     |
+-----+
3 rows in set (0.00 sec)
```

```
SELECT nome
FROM Produto
WHERE valor::numeric > 5.50 AND valor::numeric < 11.00;
```

```
mysql> SELECT nome
-> FROM Produto
-> WHERE valor > 5.50 AND valor < 11.00;
+-----+
| nome          |
+-----+
| Bisnaguinha   |
| Mortadela     |
+-----+
2 rows in set (0.00 sec)
```

WHERE

Pode conter AND, OR,
NOT, >, <, =, != entre outros
operadores

SELECT

```
SELECT nome, valor, valor*0.9
FROM Produto;
```

```
SELECT nome, valor, round(valor::numeric*0.9,2)
FROM Produto;
```

```
SELECT nome, valor, round(valor::numeric*0.9,2)
      AS valor_desconto
FROM Produto;
```

```
mysql> SELECT nome, valor, valor*0.9
-> FROM Produto;
```

nome	valor	valor*0.9
Nesquik	2.25	2.025
Todinho	1.90	1.710
Bisnaguinha	10.35	9.315
Queijo mussarela	16.60	14.940
Mortadela	10.90	9.810

```
5 rows in set (0.00 sec)
```

SELECT

Pode conter +, -, *, / entre outros operadores

```
mysql> SELECT nome, valor, round(valor*0.9,2)
-> FROM Produto;
```

nome	valor	round(valor*0.9,2)
Nesquik	2.25	2.03
Todinho	1.90	1.71
Bisnaguinha	10.35	9.32
Queijo mussarela	16.60	14.94
Mortadela	10.90	9.81

```
5 rows in set (0.00 sec)
```

```
mysql> SELECT nome, valor, round(valor*0.9,2) AS valor_desconto
-> FROM Produto;
```

nome	valor	valor_desconto
Nesquik	2.25	2.03
Todinho	1.90	1.71
Bisnaguinha	10.35	9.32
Queijo mussarela	16.60	14.94
Mortadela	10.90	9.81

```
5 rows in set (0.00 sec)
```

JUNÇÃO ENTRE TABELAS



```
CREATE TABLE professor (  
  id SERIAL,  
  nome TEXT,  
  CPF CHAR(14),  
  PRIMARY KEY(id));
```

```
CREATE TABLE turma (  
  codigo INT,  
  numero INT,  
  duracao_dias INT NOT NULL,  
  id_professor INT,  
  PRIMARY KEY(codigo, numero),  
  FOREIGN KEY(id_professor) REFERENCES professor(id));
```

Como fazer uma busca
que envolva dados de 2
tabelas?

**Qual o nome do
professor que ensina na
turma com código 1,
número 2?**

INSERÇÃO DE DADOS

```
INSERT INTO Professor (nome, CPF) VALUES ('Carla', '035.641.688-22');  
INSERT INTO Professor (nome, CPF) VALUES ('Danilo', '003.421.445-77');  
INSERT INTO Professor (nome, CPF) VALUES ('Daniel', '344.568.222-54');  
INSERT INTO Professor (nome, CPF) VALUES ('Alice', '098.765.432-11');
```

```
INSERT INTO Turma (codigo, numero, duracao_dias, id_professor) VALUES (1, 1, 60, 1);  
INSERT INTO Turma (codigo, numero, duracao_dias, id_professor) VALUES (1, 2, 70, 1);  
INSERT INTO Turma (codigo, numero, duracao_dias, id_professor) VALUES (1, 3, 80, 2);  
INSERT INTO Turma (codigo, numero, duracao_dias, id_professor) VALUES (2, 1, 10, 2);
```

JUNÇÃO ENTRE TABELAS

Professor		
id	Nome	CPF
1	Carla	035.641.688-22
2	Danilo	003.421.445-77
3	Daniel	344.568.222-54
4	Alice	098.765.432-11

Como eu vou saber o nome do professor que leciona na turma de Código = 1 e Número = 2?

Nome do professor -> Tabela Professor

Código e Número da turma -> Tabela Turma

Turma			
Codigo	Número	Duração_dias	id_professor
1	1	60	1
1	2	70	1
1	3	80	2
2	1	10	2

```
SELECT *  
FROM Professor, Turma;
```

JUNÇÃO ENTRE TABELAS

Professor		
id	Nome	CPF
1	Carla	035.641.688-22
2	Danilo	003.421.445-77
3	Daniel	344.568.222-54
4	Alice	098.765.432-11

Turma			
Codigo	Número	Duração_dias	id_professor
1	1	60	1
1	2	70	1
1	3	80	2
2	1	10	2

Professor, Turma						
id	Nome	CPF	Codigo	Numero	Duração_dias	id_professor
1	Carla	035.641.688-22	1	1	60	1
			1	2	70	1
			1	3	80	2
			2	1	10	2
2	Danilo	003.421.445-77	1	1	60	1
			1	2	70	1
			1	3	80	2
			2	1	10	2
3	Daniel	344.568.222-54	1	1	60	1
			1	2	70	1
			1	3	80	2
			2	1	10	2
4	Alice	098.765.432-11	1	1	60	1
			1	2	70	1
			1	3	80	2
			2	1	10	2

```
SELECT *
FROM Professor, Turma
WHERE id = id_professor;
```

JUNÇÃO ENTRE TABELAS

Professor, Turma						
id	Nome	CPF	Codigo	Numero	Duração_dias	id_professor
1	Carla	035.641.688-22	1	1	60	1
			1	2	70	1
			1	3	80	2
			2	1	10	2
2	Danilo	003.421.445-77	1	1	60	1
			1	2	70	1
			1	3	80	2
			2	1	10	2
3	Daniel	344.568.222-54	1	1	60	1
			1	2	70	1
			1	3	80	2
			2	1	10	2
4	Alice	098.765.432-11	1	1	60	1
			1	2	70	1
			1	3	80	2
			2	1	10	2

Professor, Turma						
id	Nome	CPF	Codigo	Numero	Duração_dias	id_professor
1	Carla	035.641.688-22	1	1	60	1
1	Carla	035.641.688-22	1	2	70	1
2	Danilo	003.421.445-77	1	3	80	2
2	Danilo	003.421.445-77	2	1	10	2

Identificar todos os professores e suas respectivas turmas

Como eu vou saber o nome do professor que leciona na turma de Código = 1 e Número = 2?


```
SELECT *
FROM Professor, Turma
WHERE id = id_professor;
```

JUNÇÃO ENTRE TABELAS

Professor, Turma						
id	Nome	CPF	Codigo	Numero	Duração_dias	id_professor
1	Carla	035.641.688-22	1	1	60	1
			1	2	70	1
			1	3	80	2
			2	1	10	2
2	Danilo	003.421.445-77	1	1	60	1
			1	2	70	1
			1	3	80	2
			2	1	10	2
3	Daniel	344.568.222-54	1	1	60	1
			1	2	70	1
			1	3	80	2
			2	1	10	2
4	Alice	098.765.432-11	1	1	60	1
			1	2	70	1
			1	3	80	2
			2	1	10	2

Professor, Turma						
id	Nome	CPF	Codigo	Numero	Duração_dias	id_professor
1	Carla	035.641.688-22	1	1	60	1
1	Carla	035.641.688-22	1	2	70	1
2	Danilo	003.421.445-77	1	3	80	2
2	Danilo	003.421.445-77	2	1	10	2

```
SELECT nome
FROM Professor, Turma
WHERE id = id_professor AND
       codigo=1 AND numero=2;
```

```
SELECT *  
FROM Professor, Turma  
WHERE id = id_professor;
```

JUNÇÃO ENTRE TABELAS

Professor, Turma						
id	Nome	CPF	Codigo	Numero	Duração_dias	id_professor
1	Carla	035.641.688-22	1	1	60	1
1	Carla	035.641.688-22	1	2	70	1
2	Danilo	003.421.445-77	1	3	80	2
2	Danilo	003.421.445-77	2	1	10	2

```
SELECT nome  
FROM Professor, Turma  
WHERE id = id_professor AND  
       codigo=1 AND numero=2;
```

Professor, Turma						
id	Nome	CPF	Codigo	Numero	Duração_dias	id_professor
1	Carla	035.641.688-22	1	1	60	1
1	Carla	035.641.688-22	1	2	70	1
2	Danilo	003.421.445-77	1	3	80	2
2	Danilo	003.421.445-77	2	1	10	2

nome

Carla
(1 row)

JOIN

Professor		
id	Nome	CPF
1	Carla	035.641.688-22
2	Danilo	003.421.445-77
3	Daniel	344.568.222-54
4	Alice	098.765.432-11

Turma			
Codigo	Número	Duração_dias	id_professor
1	1	60	1
1	2	70	1
1	3	80	2
2	1	10	2

Professor, Turma						
id	Nome	CPF	Codigo	Numero	Duração_dias	id_professor
1	Carla	035.641.688-22	1	1	60	1
1	Carla	035.641.688-22	1	2	70	1
2	Danilo	003.421.445-77	1	3	80	2
2	Danilo	003.421.445-77	2	1	10	2

```
SELECT nome
FROM Professor JOIN Turma ON
      id=id_professor;
```

```
mysql> SELECT nome
-> FROM Professor JOIN Turma ON
-> id=id_professor;
```

```
+-----+
| nome  |
+-----+
| Carla |
| Carla |
| Danilo|
| Danilo|
+-----+
```

```
4 rows in set (0.00 sec)
```

JOIN

Professor		
id	Nome	CPF
1	Carla	035.641.688-22
2	Danilo	003.421.445-77
3	Daniel	344.568.222-54
4	Alice	098.765.432-11

Turma			
Codigo	Número	Duração_dias	id_professor
1	1	60	1
1	2	70	1
1	3	80	2
2	1	10	2

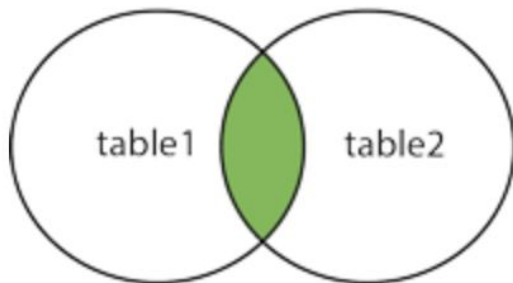
Professor, Turma						
id	Nome	CPF	Codigo	Numero	Duração_dias	id_professor
1	Carla	035.641.688-22	1	1	60	1
1	Carla	035.641.688-22	1	2	70	1
2	Danilo	003.421.445-77	1	3	80	2
2	Danilo	003.421.445-77	2	1	10	2

```
SELECT nome
FROM Professor JOIN Turma ON
      id=id_professor
WHERE codigo=1 AND numero=2;
```

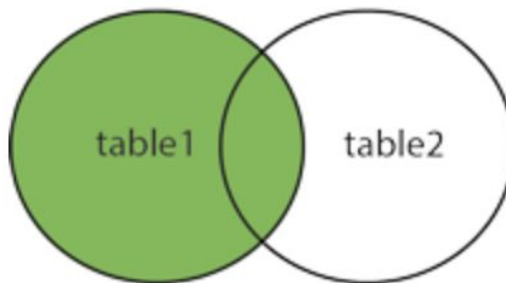
```
mysql> SELECT nome
      -> FROM Professor JOIN Turma ON
      -> id=id_professor
      -> WHERE codigo=1 AND numero=2;
+-----+
| nome  |
+-----+
| Carla |
+-----+
1 row in set (0.00 sec)
```

TIPOS DE JOIN

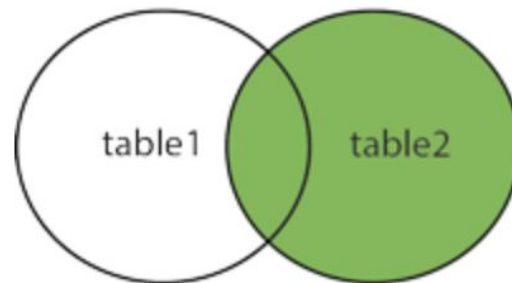
INNER JOIN



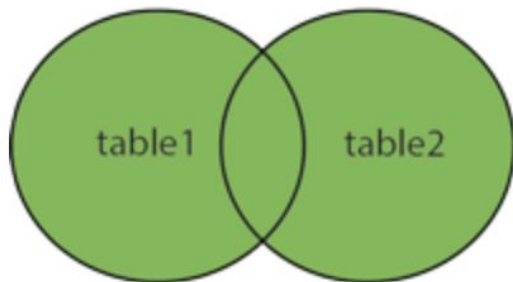
LEFT JOIN



RIGHT JOIN



FULL OUTER JOIN



```
INSERT INTO Turma (codigo, numero, duracao_dias) VALUES  
(3, 4, 120);
```

INNER JOIN

Professor		
id	Nome	CPF
1	Carla	035.641.688-22
2	Danilo	003.421.445-77
3	Daniel	344.568.222-54
4	Alice	098.765.432-11

Turma			
Codigo	Número	Duração_dias	id_professor
1	1	60	1
1	2	70	1
1	3	80	2
2	1	10	2

```
SELECT id, nome, CPF, codigo, numero, id_professor
FROM Professor INNER JOIN Turma ON
      id=id_professor;
```

Professor, Turma						
id	Nome	CPF	Codigo	Numero	Duração_dias	id_professor
1	Carla	035.641.688-22	1	1	60	1
1	Carla	035.641.688-22	1	2	70	1
2	Danilo	003.421.445-77	1	3	80	2
2	Danilo	003.421.445-77	2	1	10	2

id	nome	cpf	codigo	numero	id_professor
1	Carla	035.641.688-22	1	1	1
1	Carla	035.641.688-22	1	2	1
2	Danilo	003.421.445-77	1	3	2
2	Danilo	003.421.445-77	2	1	2

(4 rows)

LEFT
JOIN

Professor		
id	Nome	CPF
1	Carla	035.641.688-22
2	Danilo	003.421.445-77
3	Daniel	344.568.222-54
4	Alice	098.765.432-11

Turma			
Codigo	Número	Duração_dias	id_professor
1	1	60	1
1	2	70	1
1	3	80	2
2	1	10	2

```
SELECT id, nome, CPF, codigo, numero, id_professor
FROM Professor LEFT JOIN Turma ON
      id=id_professor;
```

Professor, Turma						
id	Nome	CPF	Codigo	Numero	Duração_dias	id_professor
1	Carla	035.641.688-22	1	1	60	1
1	Carla	035.641.688-22	1	2	70	1
2	Danilo	003.421.445-77	1	3	80	2
2	Danilo	003.421.445-77	2	1	10	2

id	nome	cpf	codigo	numero	id_professor
1	Carla	035.641.688-22	1	1	1
1	Carla	035.641.688-22	1	2	1
2	Danilo	003.421.445-77	1	3	2
2	Danilo	003.421.445-77	2	1	2
4	Alice	098.765.432-11			
3	Daniel	344.568.222-54			

(6 rows)

RIGHT JOIN

Professor		
id	Nome	CPF
1	Carla	035.641.688-22
2	Danilo	003.421.445-77
3	Daniel	344.568.222-54
4	Alice	098.765.432-11

Turma			
Codigo	Número	Duração_dias	id_professor
1	1	60	1
1	2	70	1
1	3	80	2
2	1	10	2

```
SELECT id, nome, CPF, codigo, numero, id_professor
FROM Professor RIGHT JOIN Turma ON
      id=id_professor;
```

Professor, Turma						
id	Nome	CPF	Codigo	Numero	Duração_dias	id_professor
1	Carla	035.641.688-22	1	1	60	1
1	Carla	035.641.688-22	1	2	70	1
2	Danilo	003.421.445-77	1	3	80	2
2	Danilo	003.421.445-77	2	1	10	2

id	nome	cpf	codigo	numero	id_professor
1	Carla	035.641.688-22	1	1	1
1	Carla	035.641.688-22	1	2	1
2	Danilo	003.421.445-77	1	3	2
2	Danilo	003.421.445-77	2	1	2
			3	1	

(5 rows)

OUTTER JOIN

Professor		
id	Nome	CPF
1	Carla	035.641.688-22
2	Danilo	003.421.445-77
3	Daniel	344.568.222-54
4	Alice	098.765.432-11

Turma			
Codigo	Número	Duração_dias	id_professor
1	1	60	1
1	2	70	1
1	3	80	2
2	1	10	2

```
SELECT id, nome, CPF, codigo, numero, id_professor
FROM Professor FULL OUTER JOIN Turma ON
      id=id_professor;
```

Professor, Turma						
id	Nome	CPF	Codigo	Numero	Duração_dias	id_professor
1	Carla	035.641.688-22	1	1	60	1
1	Carla	035.641.688-22	1	2	70	1
2	Danilo	003.421.445-77	1	3	80	2
2	Danilo	003.421.445-77	2	1	10	2

id	nome	cpf	codigo	numero	id_professor
1	Carla	035.641.688-22	1	1	1
1	Carla	035.641.688-22	1	2	1
2	Danilo	003.421.445-77	1	3	2
2	Danilo	003.421.445-77	2	1	2
			3	1	
4	Alice	098.765.432-11			
3	Daniel	344.568.222-54			

(7 rows)

JUNÇÃO ENTRE TABELAS



E se duas tabelas tiverem
atributos com o mesmo
nome?

```
CREATE TABLE usuario (  
  id SERIAL,  
  nome TEXT,  
  CPF CHAR(14),  
  PRIMARY KEY(id));
```

```
CREATE TABLE entrada (  
  id SERIAL,  
  data DATE DEFAULT CURRENT_TIMESTAMP,  
  id_usuario INTEGER,  
  PRIMARY KEY(id),  
  FOREIGN KEY(id_usuario) REFERENCES usuario(id));
```

INSERÇÃO DE DADOS

```
INSERT INTO Usuario (nome, CPF) VALUES ('Carla', '035.641.688-22');  
INSERT INTO Usuario (nome, CPF) VALUES ('Danilo', '003.421.445-77');  
INSERT INTO Usuario (nome, CPF) VALUES ('Daniel', '344.568.222-54');  
INSERT INTO Usuario (nome, CPF) VALUES ('Alice', '098.765.432-11');
```

```
INSERT INTO Entrada (id_usuario) VALUES (1);  
INSERT INTO Entrada (data, id_usuario) VALUES ('2021-12-18', 1);  
INSERT INTO Entrada (id_usuario) VALUES (2);  
INSERT INTO Entrada (data, id_usuario) VALUES ('2021-12-10', 2);  
INSERT INTO Entrada (id_usuario) VALUES (3);
```

SITUAÇÃO PROBLEMA

Identificar todas datas de entrada de Carla.

Problema:

Nome do usuário -> Tabela **usuário**

Data de entrada -> Tabela **entrada**

```
SELECT data
FROM Usuario as u JOIN Entrada as e ON u.id=id_usuario
WHERE nome='Carla';
```

Usuário		
id	Nome	CPF
1	Carla	035.641.688-22
2	Danilo	003.421.445-77
3	Daniel	344.568.222-54
4	Alice	098.765.432-11

Entrada		
id	Data	id_usuario
1	current_timestamp	1
2	18/12/21 13:17	1
3	current_timestamp	2
4	10/12/21 09:11	2
5	current_timestamp	3

SITUAÇÃO PROBLEMA

```
SELECT data
FROM Usuario as u JOIN Entrada as e ON u.id=id_usuario
WHERE nome='Carla';
```

data

2022-08-25
2021-12-18
(2 rows)

Usuário		
id	Nome	CPF
1	Carla	035.641.688-22
2	Danilo	003.421.445-77
3	Daniel	344.568.222-54
4	Alice	098.765.432-11

Entrada		
id	Data	id_usuario
1	current_timestamp	1
2	18/12/21 13:17	1
3	current_timestamp	2
4	10/12/21 09:11	2
5	current_timestamp	3

SITUAÇÃO PROBLEMA

```
SELECT data  
FROM Usuario u, Entrada e  
WHERE u.id=id_usuario AND nome='Carla';
```

data

2022-08-25
2021-12-18
(2 rows)

Usuário		
id	Nome	CPF
1	Carla	035.641.688-22
2	Danilo	003.421.445-77
3	Daniel	344.568.222-54
4	Alice	098.765.432-11

Entrada		
id	Data	id_usuario
1	current_timestamp	1
2	18/12/21 13:17	1
3	current_timestamp	2
4	10/12/21 09:11	2
5	current_timestamp	3

JUNÇÃO ENTRE TABELAS

E se eu precisar acessar
3 tabelas?

**Quero saber as datas
que foram vendidas
Bisnaguinhas**



```
CREATE TABLE Venda (  
  id SERIAL,  
  data DATE,  
  PRIMARY KEY(id));
```

```
CREATE TABLE Venda_Produto (  
  id_venda INT,  
  id_produto INT,  
  quantidade INT NOT NULL,  
  PRIMARY KEY(id_venda, id_produto),  
  FOREIGN KEY (id_venda) REFERENCES Venda(id),  
  FOREIGN KEY (id_produto) REFERENCES Produto(id));
```

```
CREATE TABLE Produto (  
  id SERIAL,  
  nome TEXT NOT NULL,  
  valor MONEY NOT NULL DEFAULT 10.00,  
  PRIMARY KEY(id));
```

INSERÇÃO DE DADOS

```
INSERT INTO Venda (data) VALUES ('2021-10-30');  
INSERT INTO Venda (data) VALUES ('2021-05-01');  
INSERT INTO Venda (data) VALUES ('2021-12-01');  
INSERT INTO Venda (data) VALUES ('2021-01-01');  
INSERT INTO Venda (data) VALUES ('2021-11-09');
```

```
INSERT INTO Produto (nome, valor) VALUES ('Nesquik', 2.25);  
INSERT INTO Produto (nome, valor) VALUES ('Todinho', 1.90);  
INSERT INTO Produto (nome, valor) VALUES ('Bisnaguinha', 10.35);  
INSERT INTO Produto (nome, valor) VALUES ('Queijo mussarela', 16.60);  
INSERT INTO Produto (nome, valor) VALUES ('Mortadela', 10.90);
```

```
INSERT INTO Venda_Produto (id_venda, id_produto, quantidade) VALUES (1,1,5);  
INSERT INTO Venda_Produto (id_venda, id_produto, quantidade) VALUES (1,2,2);  
INSERT INTO Venda_Produto (id_venda, id_produto, quantidade) VALUES (2,3,8);  
INSERT INTO Venda_Produto (id_venda, id_produto, quantidade) VALUES (3,3,10);  
INSERT INTO Venda_Produto (id_venda, id_produto, quantidade) VALUES (3,1,5);  
INSERT INTO Venda_Produto (id_venda, id_produto, quantidade) VALUES (4,1,5);  
INSERT INTO Venda_Produto (id_venda, id_produto, quantidade) VALUES (5,3,2);
```


SITUAÇÃO PROBLEMA

Quero saber as datas que foram vendidas
Bisnaguinhas

```
SELECT data
FROM Venda v, Produto p, Venda_Produto vp
WHERE id_venda = v.id AND id_produto = p.id AND
      p.nome='Bisnaguinha';
```

Venda		Produto		
id	Data	id	Nome	Valor
1	30/10/21	1	Nesquik	2.25
2	01/05/21	2	Todinho	1.90
3	01/12/21	3	Bisnaguinha	10.35
4	01/01/21	4	Queijo mussarela	16.60
5	09/11/21	5	Mortadela	10.90

Venda_Produto		
id_venda	id_produto	Quantidade
1	1	5
1	2	2
2	3	8
3	3	10
3	1	5
4	1	5
5	3	2

SITUAÇÃO PROBLEMA

Venda		Produto			Venda_Produto		
id	Data	id	Nome	Valor	id_venda	id_produto	Quantidade
1	30/10/21	1	Nesquik	2.25	1	1	5
2	01/05/21	2	Todinho	1.90	1	2	2
3	01/12/21	3	Bisnaguinha	10.35	2	3	8
4	01/01/21	4	Queijo mussarela	16.60	3	3	10
5	09/11/21	5	Mortadela	10.90	3	1	5
					4	1	5
					5	3	2

```

SELECT data
FROM Venda v, Produto p, Venda_Produto vp
WHERE id_venda = v.id AND id_produto = p.id AND
      p.nome='Bisnaguinha';

```

```

mysql> SELECT data
      -> FROM Venda v, Produto p, Venda_Produto vp
      -> WHERE id_venda = v.id AND id_produto = p.id AND
      -> p.nome="Bisnaguinha";
+-----+
| data |
+-----+
| 2021-05-01 |
| 2021-12-01 |
| 2021-11-09 |
+-----+
3 rows in set (0.00 sec)

```

03

EXEMPLOS DE SELEÇÃO

TABELAS

```
CREATE TABLE Departamento (  
    id SERIAL,  
    nome TEXT NOT NULL,  
    data_inicio DATE,  
    PRIMARY KEY(id));
```

```
CREATE TABLE Empregado (  
    matricula INT,  
    nome TEXT NOT NULL,  
    endereco TEXT,  
    salario DECIMAL(10,2),  
    supervisor INT,  
    id_depto INT,  
    PRIMARY KEY (matricula),  
    FOREIGN KEY (supervisor) REFERENCES Empregado(matricula),  
    FOREIGN KEY (id_depto) REFERENCES Departamento(id));
```

TABELAS

```
CREATE TABLE Gerencia (  
    id_depto INT,  
    matricula_emp INT,  
    PRIMARY KEY(id_depto),  
    FOREIGN KEY (matricula_emp) REFERENCES Empregado(matricula),  
    FOREIGN KEY (id_depto) REFERENCES Departamento(id));
```

```
CREATE TABLE Projeto (  
    id SERIAL  
    nome TEXT NOT NULL,  
    local TEXT NOT NULL,  
    id_depto INT,  
    PRIMARY KEY(id),  
    FOREIGN KEY (id_depto) REFERENCES Departamento(id));
```

TABELAS

```
CREATE TABLE Alocacao (  
    id_projeto INT,  
    matricula_emp INT,  
    horas INT,  
    PRIMARY KEY(id_projeto, matricula_emp),  
    FOREIGN KEY (matricula_emp) REFERENCES Empregado(matricula),  
    FOREIGN KEY (id_projeto) REFERENCES Projeto(id));  
  
CREATE TABLE Dependente (  
    id SERIAL,  
    nome TEXT NOT NULL,  
    sexo CHAR(1) NOT NULL,  
    matricula_responsavel INT,  
    PRIMARY KEY(id),  
    FOREIGN KEY (matricula_responsavel) REFERENCES Empregado(matricula));
```

DADOS

```
INSERT INTO Departamento (nome, data_inicio) VALUES ('Pesquisa', "2019-11-09");
INSERT INTO Departamento (nome, data_inicio) VALUES ('Cozinha', "2009-08-05");
INSERT INTO Departamento (nome, data_inicio) VALUES ('Estudo', "2020-03-17");
INSERT INTO Departamento (nome, data_inicio) VALUES ('Ciencia', "2010-03-09");
INSERT INTO Departamento (nome, data_inicio) VALUES ('Compras', "2012-09-09");
```

```
INSERT INTO Empregado (matricula, nome, endereco, salario, id_depto) VALUES
(273729, 'Carla', 'Rua A', 1000.20, 3);
INSERT INTO Empregado VALUES (777746, 'Danilo', 'Rua B', 2000.20, 273729, 3);
INSERT INTO Empregado VALUES (988716, 'Daniel', 'Rua C', 100.20, 273729, 2);
INSERT INTO Empregado VALUES (483292, 'Alice', 'Rua D', 2000.20, 777746, 2);
INSERT INTO Empregado VALUES (553211, 'Picachu', 'Rua E', 20.20, 777746, 2);
```

```
INSERT INTO Gerencia VALUES (1, 273729);
INSERT INTO Gerencia VALUES (2, 777746);
```

DADOS

```
INSERT INTO Projeto (nome, local, id_depto) VALUES ('Projeto Alpha', 'Natal', 1);  
INSERT INTO Projeto (nome, local, id_depto) VALUES ('Projeto Beta', 'Recife', 1);  
INSERT INTO Projeto (nome, local, id_depto) VALUES ('Projeto Delta', 'Natal', 1);  
INSERT INTO Projeto (nome, local, id_depto) VALUES ('Projeto Gama', 'Natal', 2);
```

```
INSERT INTO Alocacao VALUES (1, 273729, 40);  
INSERT INTO Alocacao VALUES (2, 273729, 20);  
INSERT INTO Alocacao VALUES (3, 777746, 60);  
INSERT INTO Alocacao VALUES (3, 988716, 40);  
INSERT INTO Alocacao VALUES (3, 483292, 40);
```

```
INSERT INTO Dependente (nome, sexo, matricula_responsavel) VALUES ('Filho 1', 'M', 273729);  
INSERT INTO Dependente (nome, sexo, matricula_responsavel) VALUES ('Filha 2', 'F', 273729);
```


EXERCÍCIOS

Obtenha o salário do empregado Carla

Selecione todos os atributos dos empregados do departamento id=2

Selecione nome e endereço dos empregados do departamento "Cozinha"

Para cada projeto em Natal, liste o ID do projeto e o nome do departamento que o controla

Para cada projeto em Natal, liste o ID do projeto, o nome do departamento que o controla com o nome de seu gerente, seu endereço e salário

Para cada empregado, liste seu nome e o nome de seu supervisor

Selecione os salários distintos pagos



Obtenha o salário do empregado Carla

```
SELECT salario  
FROM Empregado  
WHERE nome='Carla';
```

Selecione todos os atributos dos empregados do departamento id=2

```
SELECT *  
FROM Empregado  
WHERE id_depto=2;
```

Selecione nome e endereço dos empregados do departamento "Cozinha"

```
SELECT e.nome, e.endereco  
FROM Empregado e, Departamento d  
WHERE e.id_depto = d.id AND  
      d.nome = 'Cozinha';
```

Para cada projeto em Natal, liste o ID do projeto e o nome do departamento que o controla

```
SELECT p.id, d.nome  
FROM Projeto p, Departamento d  
WHERE p.id_depto = d.id AND  
      p.local = 'Natal';
```

Para cada projeto em Natal, liste o ID do projeto, o nome do departamento que o controla com o nome de seu gerente, seu endereço e salário

```
SELECT p.id, d.nome, e.nome, e.endereco, e.salario
FROM Projeto p, Departamento d, Gerencia g, Empregado e
WHERE p.id_depto = d.id AND
      d.id = g.id_depto AND
      G.matricula_emp = e.matricula AND
      p.local = 'Natal';
```

Para cada empregado, liste seu nome e o nome de seu supervisor

```
SELECT e.nome, s.nome
FROM Empregado e, Empregado s
WHERE e.supervisor = s.matricula;
```

Selecione os salários distintos pagos

```
SELECT DISTINCT salario
FROM Empregado;
```

04

DELETE, UPDATE

Deleta elementos de uma tabela

Atualiza elementos de uma tabela

CRUD - Create Read Update Delete

DELETE

```
DELETE FROM tabela  
[WHERE condição]
```

```
DELETE FROM Empregado  
WHERE salario > 1500;
```

```
DELETE FROM Empregado;
```

UPDATE

```
UPDATE tabela  
SET lista_atributos com atribuições de valores  
[WHERE condição]
```

```
UPDATE Departamento  
SET nome = 'DPUFRN'  
WHERE nome = 'Pesquisa';
```


```
UPDATE Empregado  
SET salario = salario * 1.1;
```

```
UPDATE Empregado  
SET salario = salario * 1.1  
WHERE id_depto IN (  
    SELECT id  
    FROM Departamento  
    WHERE nome = 'Cozinha');
```


05

LIKE

Análises mais específicas com strings



```
INSERT INTO Empregado VALUES (282736, "Carla Fernandes", "Avenida Brasil", 500,  
273729, 3);  
INSERT INTO Empregado VALUES (281738, "Alice Fernandes Curvelo", "Avenida dos  
Tororós", 100.20, 273729, 2);  
INSERT INTO Empregado VALUES (191999, "Camila F. Curvelo", "Rua Ble", 2000.20,  
777746, 2);  
INSERT INTO Empregado VALUES (996335, "Danilo Curvelo", "Rua Bla", 20.20, 777746,  
2);  
INSERT INTO Empregado VALUES (282877, "Ana Carla Curvelo Souza", "Avinha X", 20.20,  
777746, 2);
```



LIKE

```
atributo LIKE '%texto%'
atributo LIKE '_texto_'
atributo LIKE '%texto_'
atributo LIKE '_texto%'
```

Caracteres para comparação

%: caracter coringa

_: apenas um caracter

```
SELECT nome
FROM Empregado
WHERE nome LIKE 'Carla%';
```

nome
Carla
Carla Fernandes

```
SELECT nome
FROM Empregado
WHERE nome LIKE '%Carla%';
```

nome
Carla
Carla Fernandes
Ana Carla Curvelo Souza

```
SELECT nome
FROM Empregado
WHERE nome LIKE '%Curvelo%';
```

nome
Camila F. Curvelo
Alice Fernandes Curvelo
Ana Carla Curvelo Souza
Danilo Curvelo

LIKE

```
atributo LIKE '%texto%'
atributo LIKE '_texto_'
atributo LIKE '%texto_'
atributo LIKE '_texto%'
```

Caracteres para comparação

%: caracter curinga

_: apenas um caracter

```
SELECT nome
FROM Empregado
WHERE nome LIKE 'C%';
```

nome
Camila F. Curvelo
Carla
Carla Fernandes

```
SELECT nome
FROM Empregado
WHERE nome LIKE '%Fernandes%' OR nome LIKE '%F.%';
```

nome
Camila F. Curvelo
Alice Fernandes Curvelo
Carla Fernandes

LIKE

```
atributo LIKE '%texto%'
atributo LIKE '_texto_'
atributo LIKE '%texto_'
atributo LIKE '_texto%'
```

Caracteres para comparação

%: caracter curinga

_: apenas um caracter

```
SELECT endereco
FROM Empregado
WHERE endereco LIKE 'Av%';
```

endereco
Avenida dos Tororós
Avenida Brasil
Avinha X

```
SELECT nome
FROM Empregado
WHERE nome LIKE '%Fernandes%Curvelo%';
```

nome
Alice Fernandes Curvelo

LIKE

```
atributo LIKE '%texto%'
atributo LIKE '_texto_'
atributo LIKE '%texto_'
atributo LIKE '_texto%'
```

Caracteres para comparação

%: caracter curinga

_: apenas um caracter

```
SELECT nome
FROM Empregado
WHERE nome LIKE '_arla';
```

nome
Carla

```
SELECT nome
FROM Empregado
WHERE nome LIKE '%arla%';
```

nome
Carla
Carla Fernandes

```
SELECT nome
FROM Empregado
WHERE nome LIKE '%carla%';
```

nome
Carla
Carla Fernandes
Ana Carla Curvelo Souza

Como implementar buscas?

carla curvelo

```
SELECT nome
FROM Empregado
WHERE nome LIKE '%carla%curvelo%';
```

nome
Ana Carla Curvelo Souza

```
SELECT nome
FROM Empregado
WHERE nome LIKE '%carla%' OR nome LIKE '%curvelo%';
```

nome
Camila F. Curvelo
Carla
Alice Fernandes Curvelo
Carla Fernandes
Ana Carla Curvelo Souza
Danilo Curvelo

DÚVIDAS?

Qualquer dúvida entrar em
contato

Profª. Carla Fernandes Curvelo

carla.fernandes@ufrn.br