

# CRIAR TABELAS: CURSO e CURSO\_FUNCIONARIO

```
CREATE TABLE Curso  
( idCurso INT NOT NULL,  
  Curso VARCHAR(45) NOT NULL,  
  PRIMARY KEY (idCurso))
```

# CRIAR TABELAS: CURSO e CURSO\_FUNCIONARIO

```
CREATE TABLE Curso_Funcionario
( idCursofuncionario INT NOT NULL,
  idCurso INT NOT NULL,
  idFuncionario INT NOT NULL,
  oferta DATE,
  aprovado CHAR,
  PRIMARY KEY (idCursofuncionario),
  CONSTRAINT FOREIGN KEY (idFuncionario) REFERENCES Funcionario (idFuncionario),
  CONSTRAINT FOREIGN KEY (idCurso) REFERENCES Curso (idCurso))
```

# INSERIR DADOS: Curso e Curso\_funcionários

```
INSERT INTO Curso VALUES (1,'SQL'),(2,'PHP'),(3,'Java'),(4,'PFSense');
```

```
INSERT INTO curso_funcionario VALUES
```

```
(1,3,123,'2018-05-30','S'),  
(2,3,152,'2018-05-30','N'),  
(3,3,222,'2018-05-30','S'),  
(4,1,123,'2018-10-20','S'),  
(5,1,152,'2018-10-20','S'),  
(6,4,222,'2018-11-27','');
```

Inserção incorreta:

```
INSERT INTO curso_funcionario VALUES (9,1,222, '30-05-2018','S');
```

# DML – Selecionar Dados (SELECT)

## Cláusulas GROUP BY, HAVING e ORDER BY

```
SELECT coluna1, coluna2, coluna3  
FROM tabela_nome1, tabela_nome2  
WHERE coluna1=valor1  
AND   coluna2=valor2  
OR   coluna2=valor3  
GROUP BY coluna1  
HAVING AVG(coluna1) > 100  
ORDER BY coluna2;
```

# DML – Selecionar Dados (SELECT)

## Agrupando valores GROUP BY

As funções de agregação também podem agrupar os valores de acordo com determinadas colunas.

Listar quantos cursos cada funcionário fez (agrupar por funcionário):

```
SELECT COUNT(*), idfuncionario FROM curso_funcionario GROUP BY (idfuncionario)
```

Listar quantos funcionários fez cada curso (agrupar por curso):

```
SELECT COUNT(*), idcurso FROM curso_funcionario GROUP BY (idcurso)
```

# DML – Selecionar Dados (SELECT)

## Agrupando valores HAVING

Podemos restringir os resultados das funções de agregação.  
Para isso utilizamos a cláusula HAVING

```
SELECT COUNT(*), idFuncionario FROM curso_funcionario  
GROUP BY (idfuncionario) HAVING idfuncionario=123
```

# SELECT LIMIT

Retornando no máximo 5 linhas

**SELECT \* FROM curso\_funcionario limit 5**

Retornando no máximo 8 linhas, a partir da 2

**SELECT \* FROM funcionario limit 2,8**

idFuncionario	Nome	Sobrenome	idDepartamento
123	Julio		1
152	Arnaldo	Coelho	1
222	Carol	Ferreira	2
326	João	Silveira	2
331	George	de la Rocha	3
332	José	Oliveira	1
546	José	Pereira	4
631	David	Luz	3
654	Zacarias	Ferreira	4
745	Eric	Estrada	4
845	Elizabeth	Coelho	1
846	Joaquim	Goveia	1

idFuncionario	Nome	Sobrenome	idDepartamento
222	Carol	Ferreira	2
326	João	Silveira	2
331	George	de la Rocha	3
332	José	Oliveira	1
546	José	Pereira	4
631	David	Luz	3
654	Zacarias	Ferreira	4
745	Eric	Estrada	4

# TIPOS DE DADOS PARA DATA E HORA

O Formato padrão é **YYYY-MM-DD**, para extrair a informação deverá ser tratado na aplicação ou na consulta.

**DATE** - format YYYY-MM-DD

**DATETIME** - format: YYYY-MM-DD HH:MI:SS

**TIMESTAMP** - format: YYYY-MM-DD HH:MI:SS

**YEAR** - format YYYY or YY

**DATETIME:** representa uma data como no calendário e a hora como encontrado no relógio.

**TIMESTAMP:** leva em consideração o fuso horário. Exemplo: Para determinadas consultas, há uma diferença entre estar no Brasil ou Japão.



## MySQL: Funções DATE\_FORMAT e TIME\_FORMAT

Máscara	Descrição	Exemplo
%m	mês	01, ..., 12
%c	mês	1, ..., 12
%M	mês por extenso	'janeiro', ..., 'dezembro'
%d	dia do mês	01, ..., 31
%e	dia do mês	1, ..., 31
%Y	ano - 4 dígitos	2012, 2013, ...
%y	ano - 2 dígitos	12, 13, ...
%h	hora - 12h	01, ..., 12
%H	hora - 24h	00, ..., 23
%i	minuto	0, ..., 59
%s	segundo	0, ..., 59
%W	dia da semana	'domingo', ..., 'sábado'

## MySQL: Funções DATE\_FORMAT e TIME\_FORMAT

```
SELECT DATE_FORMAT('2018-03-31', '%e %M %Y');    -- 31 março 2018
SELECT DATE_FORMAT('2018-03-29', '%d/%m/%Y');    -- 29/03/2018
SELECT TIME_FORMAT('15:40:00', '%Hh %im %ss');    -- 15h 40m 00s
SELECT CURRENT_DATE();                            -- 2018-11-29
SELECT CURRENT_TIME();                            -- hora atual
SELECT NOW();                                     -- 2018-11-29 15:40:00
SELECT DATEDIFF(CURRENT_DATE(), '2018-02-08');    -- valor em dias
```

# SELECT – DATA/HORA

idCursofuncionario	idCurso	idFuncionario	oferta	aprovado
1	3	123	2018-05-30 00:00:00	S
2	3	152	2018-05-30 00:00:00	N
3	3	222	2018-05-30 00:00:00	S
4	1	123	2018-10-20 00:00:00	S
5	1	152	2018-10-20 00:00:00	S
6	4	222	2018-11-27 00:00:00	
8	4	331	2018-11-01 00:00:00	
9	1	222	0000-00-00 00:00:00	S
10	3	222	0000-00-00 00:00:00	S

Primeiro faça um select para verificar se o formato é DATE ou DATETIME.

```
SELECT * FROM curso_funcionario where oferta between '2018-01-01' and '2018-07-31'
```

```
SELECT * FROM curso_funcionario where oferta > '2018-07-31'
```

```
SELECT * FROM curso_funcionario WHERE oferta LIKE '%2018-11%'
```

```
SELECT DATEDIFF(CURRENT_DATE(), '2018-11-26'); diferença entre duas datas
```

```
SELECT TIMESTAMPDIFF(HOUR, hora_inicial, hora_final ) diferença em horas
```

## MySQL: Funções DATE\_ADD e DATE\_SUB

- Função para operações de soma e subtração com datas

**DATE\_ADD( data, INTERVAL expr unidade) ou DATE\_SUB(...)**

-- unidades: MICROSECOND, SECOND, MINUTE, HOUR, DAY, WEEK, MONTH, YEAR

### **EXEMPLOS:**

**SELECT DATE\_ADD('2018-03-31 15:40', INTERVAL 2 HOUR);**      -- 2018-03-31 17:40:00

**SELECT DATE\_SUB('2018-03-31 15:40', INTERVAL 1 YEAR);**      -- 2017-03-31 15:40:00

**SELECT DATE\_ADD('2018-03-31 15:40:00', INTERVAL '1 2' DAY\_HOUR);**

-- 2018-04-01 17:40:00

# SQL JOINS - Introdução

## Consulta sobre múltiplas relações

idFuncionario	Nome	Sobrenome	idDepartamento
123	Julio	Silva	1
326	João	Silveira	2
331	George	de la Rocha	3
332	José	Oliveira	1

idDepartamento	dNome	Orcamento
1	Financeiro	15000
2	TI	60000
3	Gestão de Pessoas	150000

- Como obter o nome e sobrenome de todos os funcionários, juntamente com os nomes dos departamentos onde estão lotados?

# SQL JOINS - Introdução

## Consulta sobre múltiplas relações

Selecionando colunas de mais de uma relação (Produto cartesiano)

```
SELECT Nome, Sobrenome, dNome  
FROM Funcionario, Departamento;
```

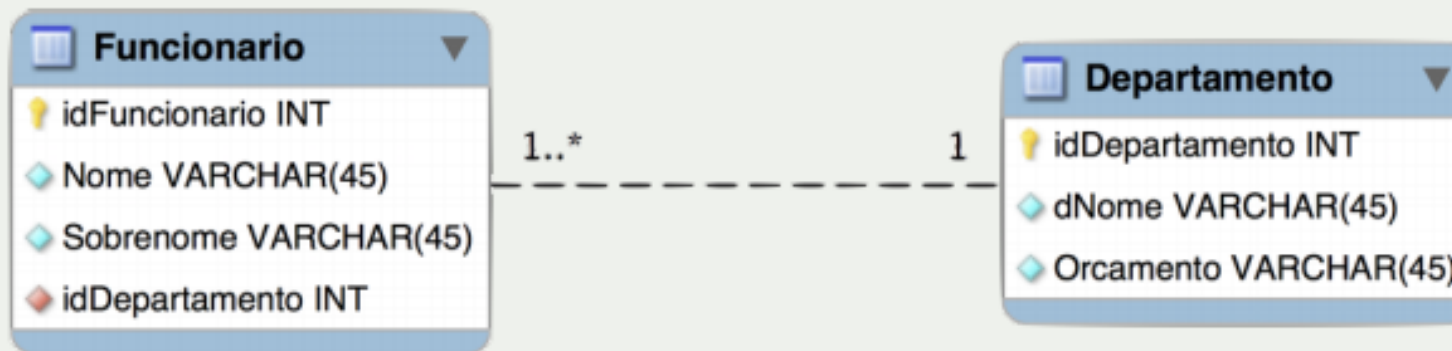
Nome	Sobrenome	dNome
Julio	Silva	Financeiro
Julio	Silva	TI
Julio	Silva	Gestão de Pessoas
Arnaldo	Coelho	Financeiro
.....		

```
SELECT f.nome,f.sobrenome,d.dnome  
FROM funcionario f, departamento d  
WHERE f.idDepartamento=d.idDepartamento
```

# SQL JOINS - Introdução

Uma cláusula JOIN é usada para combinar linhas de duas ou mais tabelas, com base em uma coluna relacionada entre elas.

Esquema usado nos próximos exemplos

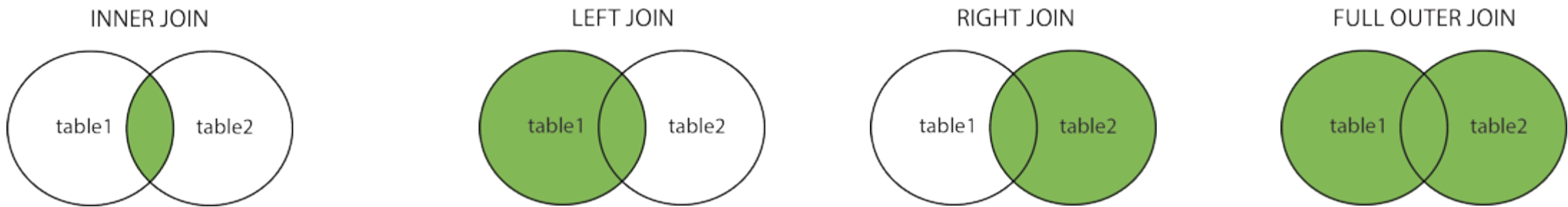




# SQL JOINS - Introdução

idFuncionario	Nome	Sobrenome	idDepartamento		idDepartamento	dNome	Orcamento
123	Julio		1	Financeiro	1	Financeiro	15000
152	Arnaldo	Coelho	1	Financeiro	2	TI	60000
222	Carol	Ferreira	2	TI	3	Gestão de Pessoas	150000
326	João	Silveira	2	TI	4	Pesquisa e Desenvolvimento	7500
331	George	de la Rocha	3	Gestão de Pessoas	5	Jurídico	1000
332	José	Oliveira	1	Financeiro	10	Limpeza	0
546	José	Pereira	4	Pesquisa e Desenvolvimento			
631	David	Luz	3	Gestão de Pessoas			
654	Zacarias	Ferreira	4	Pesquisa e Desenvolvimento			
745	Eric	Estrada	4	Pesquisa e Desenvolvimento			
845	Elizabeth	Coelho	1	Financeiro			
846	Joaquim	Goveia	1	Financeiro			

# Tipos de SQL JOINS



**INNER JOIN:** Retorna registros que possuam valores correspondentes em ambas tabelas.

SELECT colunas FROM tabela1 **INNER JOIN** tabela2 **ON** tabela1.colunaA = tabela2.colunaB;

SELECT f.Nome, d.dNome FROM Funcionario f **INNER JOIN** Departamento d **ON** f.idDepartamento = d.idDepartamento

```
SELECT f.nome,f.sobrenome,d.dnome  
FROM funcionario f, departamento d  
WHERE f.idDepartamento=d.idDepartamento;
```

```
SELECT f.Nome,f.sobrenome, d.dNome  
FROM Funcionario f INNER JOIN Departamento d  
ON f.idDepartamento= d.idDepartamento;
```

- New
- admin
- aula1
- aula2
- aula3
- aula4
  - New
  - curso
  - curso\_funcionario
  - departamento
  - funcionario
- exportacao
- information\_schema
- mysql
- performance\_schema
- phpmyadmin
- taxi
- test

## Mostrar Caixa do query

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available. ⓘ

✓ A mostrar registos de 0 - 11 (12 total, A consulta demorou 0,0000 segundos.)

```
SELECT f.Nome, d.dNome FROM Funcionario f INNER JOIN Departamento d ON f.idDepartamento = d.idDepartamento
```

☐ Perfil [\[Editar em linha\]](#) [\[Edita\]](#) [\[Explicar SQL\]](#) [\[Criar código PHP\]](#) [\[Actualizar\]](#)

☐ Mostrar tudo

Número de registos: 25 ▼

Filtrar registos:

Ordenar por chave:

## + Opções

Nome	dNome
Julio	Financeiro
Arnaldo	Financeiro
Carol	TI
João	TI
George	Gestão de Pessoas
José	Financeiro
José	Pesquisa e Desenvolvimento
David	Gestão de Pessoas
Zacarias	Pesquisa e Desenvolvimento
Eric	Pesquisa e Desenvolvimento
Elizabeth	Financeiro
Joaquim	Financeiro


☐ Mostrar tudo

Número de registos: 25 ▼

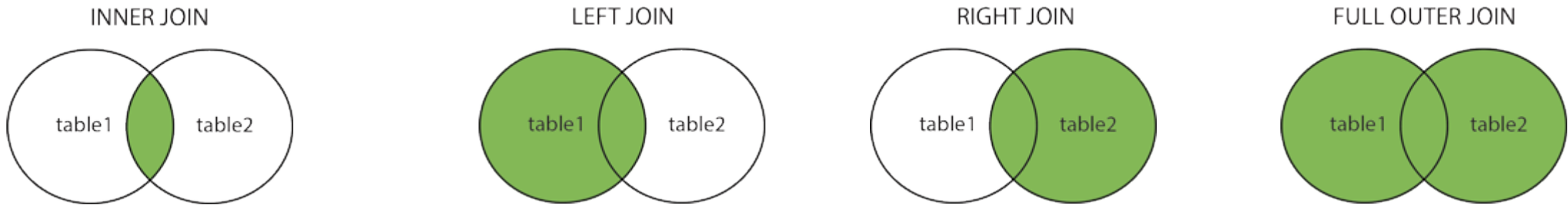
Filtrar registos:

Ordenar por chave:

## Operações resultantes das consultas

 Imprimir  Copiar para área de transferência  Exportar  Mostrar gráfico  Criar visualização

# Tipos de SQL JOINS



**LEFT JOIN:** Retorna todos os registros da relação à esquerda, mesmo aquelas que não possuam correspondentes na tabela à direita.

```
SELECT colunas FROM tabela1 LEFT JOIN tabela2 ON  
tabela1.colunaA = tabela2.colunaB;
```

```
SELECT * FROM Departamento d LEFT JOIN Funcionario f  
ON d.idDepartamento = f.idDepartamento;
```

- New
- admin
- aula1
- aula2
- aula3
- aula4
  - New
  - curso
  - curso\_funcionario
  - departamento
  - funcionario
- exportacao
- information\_schema
- mysql
- performance\_schema
- phpmyadmin
- taxi
- test

Mostrar Caixa do query

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available. ?

✓ A mostrar registos de 0 - 13 (14 total, A consulta demorou 0,0000 segundos.)

`SELECT * FROM Departamento d LEFT JOIN Funcionario f ON d.idDepartamento = f.idDepartamento`

☐ Perfil [Editar em linha] [Edita] [Explicar SQL] [Criar código PHP] [Actualizar]

☐ Mostrar tudo | Número de registos: 25 | Filtrar registos: Pesquisar esta tabela | Ordenar por chave: Nenhum

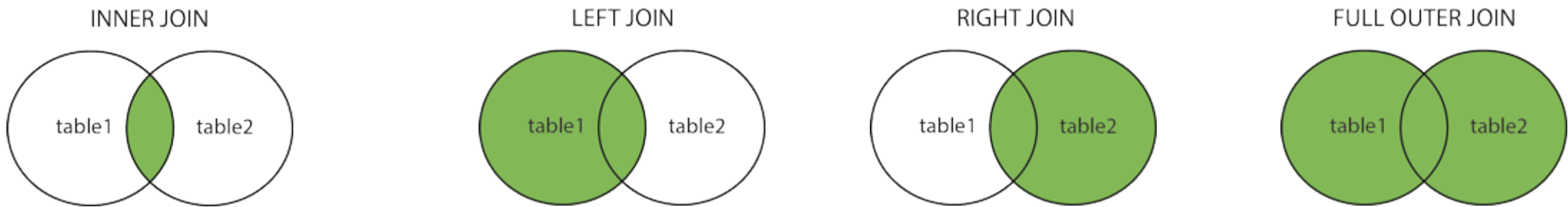
+ Opções

idDepartamento	dNome	Orcamento	idFuncionario	Nome	Sobrenome	idDepartamento
1	Financeiro	15000	123	Julio		1
1	Financeiro	15000	152	Arnaldo	Coelho	1
1	Financeiro	15000	332	José	Oliveira	1
1	Financeiro	15000	845	Elizabeth	Coelho	1
1	Financeiro	15000	846	Joaquim	Goveia	1
2	TI	60000	222	Carol	Ferreira	2
2	TI	60000	326	João	Silveira	2
3	Gestão de Pessoas	150000	331	George	de la Rocha	3
3	Gestão de Pessoas	150000	631	David	Luz	3
4	Pesquisa e Desenvolvimento	7500	546	José	Pereira	4
4	Pesquisa e Desenvolvimento	7500	654	Zacarias	Ferreira	4
4	Pesquisa e Desenvolvimento	7500	745	Eric	Estrada	4
5	Jurídico	1000	NULL	NULL	NULL	NULL
10	Limpeza	0	NULL	NULL	NULL	NULL

☐ Mostrar tudo | Número de registos: 25 | Filtrar registos: Pesquisar esta tabela | Ordenar por chave: Nenhum

Operações resultantes das consultas

# Tipos de SQL JOINS



**RIGHT JOIN:** Retorna todos registros da relação à direita, mesmo aquelas que não possuam correspondentes na tabela à esquerda.

```
SELECT colunas FROM tabela1 RIGHT JOIN tabela2 ON  
tabela1.colunaA = tabela2.colunaB;
```

```
SELECT * FROM Departamento d RIGHT JOIN Funcionario f  
ON d.idDepartamento = f.idDepartamento;
```



Recente Favoritos

- New
- admin
- aula1
- aula2
- aula3
- aula4
  - New
  - curso
  - curso\_funcionario
  - departamento
  - funcionario
- exportacao
- information\_schema
- mysql
- performance\_schema
- phpmyadmin
- taxi
- test

Servidor: 127.0.0.1 » Base de Dados: aula4 » Tabela: Departamento

Procurar

Estrutura

SQL

Pesquisar

Inserir

Exportar

Importar

Mais

Mostrar Caixa do query

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available. ⓘ

✓ A mostrar registos de 0 - 11 (12 total, A consulta demorou 0,0000 segundos.)

```
SELECT * FROM Departamento d RIGHT JOIN Funcionario f ON d.idDepartamento = f.idDepartamento
```

☐ Perfil [Editar em linha] [Edita] [Explicar SQL] [Criar código PHP] [Actualizar]

☐ Mostrar tudo | Número de registos: 25 | Filtrar registos: Pesquisar esta tabela | Ordenar por chave: Nenhum

+ Opções

idDepartamento	dNome	Orcamento	idFuncionario	Nome	Sobrenome	idDepartamento
1	Financeiro	15000	123	Julio		1
1	Financeiro	15000	152	Arnaldo	Coelho	1
2	TI	60000	222	Carol	Ferreira	2
2	TI	60000	326	João	Silveira	2
3	Gestão de Pessoas	150000	331	George	de la Rocha	3
1	Financeiro	15000	332	José	Oliveira	1
4	Pesquisa e Desenvolvimento	7500	546	José	Pereira	4
3	Gestão de Pessoas	150000	631	David	Luz	3
4	Pesquisa e Desenvolvimento	7500	654	Zacarias	Ferreira	4
4	Pesquisa e Desenvolvimento	7500	745	Eric	Estrada	4
1	Financeiro	15000	845	Elizabeth	Coelho	1
1	Financeiro	15000	846	Joaquim	Goveia	1

☐ Mostrar tudo | Número de registos: 25 | Filtrar registos: Pesquisar esta tabela | Ordenar por chave: Nenhum

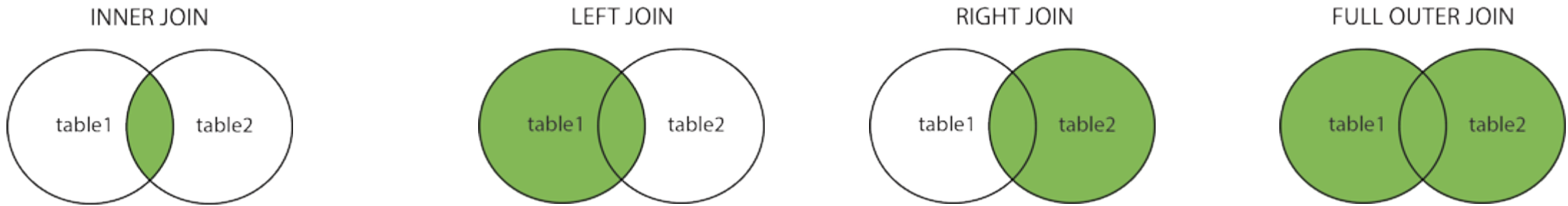
Operações resultantes das consultas

Imprimir Copiar para área de transferência Exportar Mostrar gráfico Criar visualização

Consola



# Tipos de SQL JOINS



**FULL OUTER JOIN:** Retorna todos registros da relação à direita, mesmo aquelas que não possuam correspondentes na tabela à esquerda.

SELECT colunas FROM tabela1 **FULL OUTER JOIN** tabela2  
**ON** tabela1.colunaA = tabela2.colunaB;

SELECT \* FROM Departamento d NATURAL FULL OUTER  
JOIN Funcionario f; **MySQL não implementa FULL JOIN**