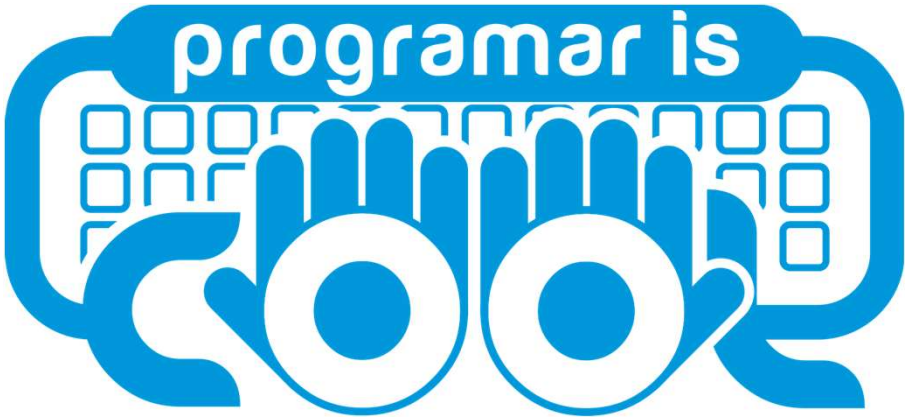
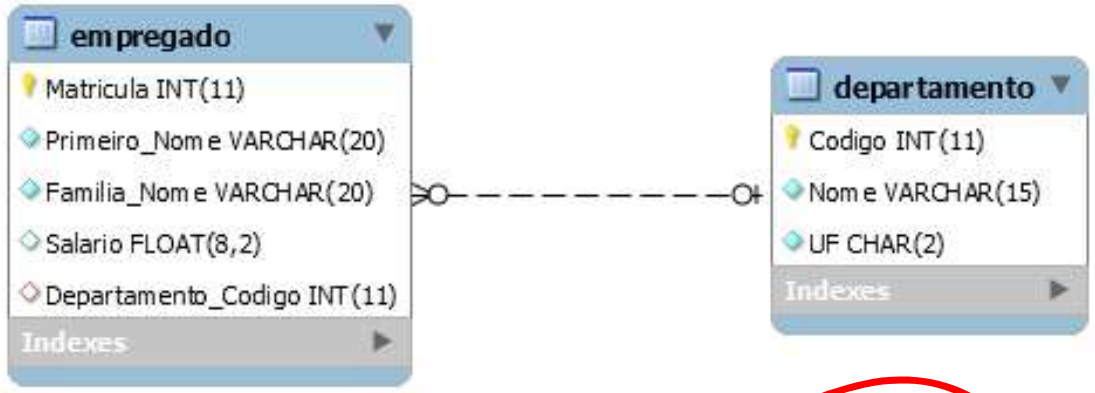


Modelagem de Dados e Linguagem SQL Descomplicadas usando o MySQL

https://www.youtube.com/c/ProgramarIsCool

Linguagem de Manipulação de Dados DML

Estudo de Caso Empresa



Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo
123	Igor	Pereira	100.00	10
159	Denise	Moreno	440.00	40
369	Marcelo	Neiva	900.00	40
456	Ana	Oliveira	200.00	20
789	Clara	Silva	300.00	30
963	Deyse	Silva	330.00	NULL

Codigo	Nome	UF
10	RH	RJ
20	TI	RJ
30	Logística	DF
40	Financeiro	DF
50	Venda	RS
60	Pesquisa	RJ

Estudo de Caso Empresa

```
CREATE TABLE Departamento (
            Codigo      INT NOT NULL,
             Nome       VARCHAR(15) NOT NULL,
            UF          CHAR(2) NOT NULL,
            PRIMARY KEY (Codigo));
```

```
CREATE TABLE IF NOT EXISTS Empregado (
  Matricula                 INT NOT NULL,
  Primeiro_Nome             VARCHAR(20) NOT NULL,
  Familia_Nome              VARCHAR(20) NOT NULL,
  Salario                   FLOAT NULL DEFAULT NULL,
  Departamento_Codigo                 INT NULL,
            PRIMARY KEY (Matricula),
            CONSTRAINT fk_empregado_departamento
            FOREIGN KEY (Departamento_Codigo)
            REFERENCES departamento (Codigo));
```

Inserir Registros (Insert)

Inserir Registros

```
INSERT INTO Departamento (Codigo, Nome, UF)
VALUES (10, 'RH', 'RJ'),
       (20, 'TI', 'RJ'),
       (30, 'Logística', 'DF'),
       (40, 'Financeiro', 'DF'),
       (50, 'Venda', 'RS'),
       (60, 'Pesquisa', 'RJ');
```

```
INSERT INTO Empregado(Matricula, Primeiro_Nome, Familia_Nome, Salario, Departamento_Codigo)
VALUES (123, 'Igor', 'Pereira', 100.00, 10),
(159, 'Denise', 'Moreno', 440.00, 40),
(369, 'Marcelo', 'Neiva', 900.00, 40),
(456, 'Ana', 'Oliveira', 200.00, 20),
(789, 'Clara', 'Silva', 300.00, 30),
(963, 'Deyse', 'Silva', 330.00, null);
```

Inserir Registros

```
CREATE TABLE Dep_Novo
AS SELECT Codigo, Nome, UF
FROM Departamento;
```

Inserir Registros

```
CREATE TABLE Dep ( Codigo      INT NOT NULL,
                   Nome        VARCHAR(15) NOT NULL,
                   UF          CHAR(2) NOT NULL,
                   PRIMARY KEY (Codigo));
```

```
INSERT INTO Dep (Codigo, Nome, UF)
SELECT Codigo, Nome, UF
FROM Departamento;
```

Modificar Registros (Update)

Modificar Registros

- Atualização:

```
UPDATE NOME_TABELA
SET COL1=VAL1, COL2=VAL2,...
WHERE (expressão lógica);
```

- Exemplo:

```
UPDATE Dep
SET UF = 'MG', Nome = 'Finanças'
WHERE Codigo = 40;
```

Excluir Registros (Delete)

Excluir Registros

- Exclusão:

```
DELETE FROM NOME_TABELA
WHERE (expressão lógica);
```

- Exemplo:

```
DELETE FROM Dep
WHERE Codigo = 30;
```

Comandos COMMIT e ROLLBACK

Comandos COMMIT e ROLLBACK

```
START TRANSACTION;
INSERT INTO dep (Departamento_Codigo, Nome, UF)
VALUES (70, 'Teste', 'UF');
```

- ROLLBACK: desfaz a transação.
- COMMIT: torna permanente os efeitos das transações.

Consultar Registros (Select)

Consultar Registros

- Consulta Simples:

```
SELECT COL1, COL2,...,COLN
FROM NOME_TABELA
WHERE (expressão lógica) ;
```

- Exemplo :

```
SELECT Codigo, Nome, UF
FROM Departamento;
```

Consultar Registros com Campos Nulos

- Exemplo:

```
SELECT Primeiro_Nome, Departamento_Codigo
FROM Empregado
WHERE Departamento_Codigo is null;
```

- Exemplo:

```
SELECT Primeiro_Nome, Departamento_Codigo
FROM Empregado
WHERE Departamento_Codigo is not null;
```

Consultar Registros usando o Distinct

- Elimina duplicatas:

```
SELECT DISTINCT(Departamento_Codigo)
FROM Empregado;
```

Operadores Lógicos

- Usados para validar condições:
  - AND;
  - OR;
  - IN;
  - NOT;
  - BETWEEN;
  - LIKE;
  - ALL;
  - ANY/SOME.

Operadores Lógicos AND e OR

- Condição 1: Departamento_Codigo = 40.
- Condição 2: Salário > 500,00.

Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo
123	Igor	Pereira	100.00	10
159	Denise	Moreno	440.00	40
369	Marcelo	Neiva	900.00	40
456	Ana	Oliveira	200.00	20
789	Clara	Silva	300.00	30
963	Deyse	Silva	330.00	NULL

C 1	C 2	AND
V	V	V
V	F	F
F	V	F
F	F	F

Operadores Lógicos AND e OR

- Condição 1: Departamento_Codigo = 40.
- Condição 2: Salário > 500,00.

Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo
123	Igor	Pereira	100.00	10
159	Denise	Moreno	440.00	40
369	Marcelo	Neiva	900.00	40
456	Ana	Oliveira	200.00	20
789	Clara	Silva	300.00	30
963	Deyse	Silva	330.00	NULL

C 1	C 2	OR
V	V	V
V	F	V
F	V	V
F	F	F

Operadores Lógicos IN e NOT IN

- Quais empregados que trabalham nos departamentos 10 ou 30?

Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo
123	Igor	Pereira	100.00	10
159	Denise	Moreno	440.00	40
369	Marcelo	Neiva	900.00	40
456	Ana	Oliveira	200.00	20
789	Clara	Silva	300.00	30
963	Deyse	Silva	330.00	NULL

Operadores Lógicos IN e NOT IN

- Quais empregados que não trabalham nos departamentos 10 ou 30?

Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo
123	Igor	Pereira	100.00	10
159	Denise	Moreno	440.00	40
369	Marcelo	Neiva	900.00	40
456	Ana	Oliveira	200.00	20
789	Clara	Silva	300.00	30
963	Deyse	Silva	330.00	NULL

Operadores Lógicos BETWEEN

- Quais empregados possuem salário entre R$ 250,00 e R$ 800,00?

Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo
123	Igor	Pereira	100.00	10
159	Denise	Moreno	440.00	40
369	Marcelo	Neiva	900.00	40
456	Ana	Oliveira	200.00	20
789	Clara	Silva	300.00	30
963	Deyse	Silva	330.00	NULL

Operadores Lógicos

Expressão	Explicação
LIKE 'A%'	Todas as palavras que iniciem com a letra A.
LIKE '%A'	Todas que terminem com a letra A.
LIKE '% A%'	Todas que tenham a letra A em qualquer posição.
LIKE 'A_'	String de dois caracteres que tenham a primeira letra A e o segundo caractere seja qualquer outro.
LIKE '_A'	String de dois caracteres cujo primeiro caractere seja qualquer um e a última letra seja A.
LIKE '_A_'	String de três caracteres cuja segunda letra seja A, independentemente do primeiro ou do último caractere.
LIKE '%A_'	Todos que tenham a letra A na panúltima posição e a última seja qualquer outro caractere.
LIKE '_A%'	Todos que tanham a letra A na segunda posição e o primeiro caractere seja qualquer um.

Operadores Lógicos ALL e ANY/SOME

- Quais empregados do departamento 40 possuem salário maior que todos os empregados do departamento 30?

Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo
123	Igor	Pereira	100.00	10
159	Denise	Moreno	440.00	40
369	Marcelo	Neiva	900.00	40
456	Ana	Oliveira	200.00	20
789	Clara	Silva	300.00	30
855	Edu	Snatos	600.00	30
963	Deyse	Silva	330.00	NULL

Operadores Lógicos ALL e ANY/SOME

- Quais empregados do departamento 40 possuem salário maior que pelo menos um dos empregados do departamento 30?

Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo
123	Igor	Pereira	100.00	10
159	Denise	Moreno	440.00	40
369	Marcelo	Neiva	900.00	40
456	Ana	Oliveira	200.00	20
789	Clara	Silva	300.00	30
855	Edu	Snatos	600.00	30
963	Deyse	Silva	330.00	NULL

Operadores Aritméticos

- Operadores Aritméticos:
  - +;
  - -;
  - *;
  - /;
  - %.

- Operador % (módulo):
  - Retorna o resto inteiro de uma divisão.
  - Por exemplo, 13% 5 = 3 porque o resto de 13 dividido por 5 é 3.

Operador Concatenação

- Concatena Strings:
  - CONCAT(COL1, COL2,...,COLN).

```
SELECT Concat(Primeiro_Nome, ` ` , Familia_Nome)
FROM Empregado;
```

Funções Agregadas

- Funções Agregadas:
  - COUNT();
  - SUM();
  - AVG();
  - MAX();
  - MIN().

Order By

- Ordenação de Resultados:

```
SELECT Matricula, Primeiro_Nome, Salario
FROM Empregado
ORDER BY Primeiro_Nome ASC;

SELECT Matricula, Primeiro_Nome, Salario
FROM Empregado
ORDER BY Primeiro_Nome DESC;
```

Group By

```
SELECT Departamento_Codigo, MAX(Salario)
FROM Empregado
GROUP BY Departamento_Codigo;
```

Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo
123	Igor	Pereira	100.00	10
159	Denise	Moreno	440.00	40
369	Marcelo	Neiva	900.00	40
456	Ana	Oliveira	200.00	20
789	Clara	Silva	300.00	30
855	Edu	Snatos	600.00	30
963	Deyse	Silva	330.00	NULL

Group By

```
SELECT Departamento_Codigo,  AVG(Salario)
FROM Empregado
GROUP BY Departamento_Codigo;
```

Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo
123	Igor	Pereira	100.00	10
159	Denise	Moreno	440.00	40
369	Marcelo	Neiva	900.00	40
456	Ana	Oliveira	200.00	20
789	Clara	Silva	300.00	30
855	Edu	Snatos	600.00	30
963	Deyse	Silva	330.00	NULL

Group By

```
SELECT Departamento_Codigo, COUNT(*)
FROM Empregado
GROUP BY Departamento_Codigo;
```

Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo
123	Igor	Pereira	100.00	10
159	Denise	Moreno	440.00	40
369	Marcelo	Neiva	900.00	40
456	Ana	Oliveira	200.00	20
789	Clara	Silva	300.00	30
855	Edu	Snatos	600.00	30
963	Deyse	Silva	330.00	NULL

Having

```
SELECT Departamento_Codigo, COUNT(*)
FROM Empregado
GROUP BY Departamento_Codigo
HAVING COUNT(*) > 1;
```

Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo
123	Igor	Pereira	100.00	10
159	Denise	Moreno	440.00	40
369	Marcelo	Neiva	900.00	40
456	Ana	Oliveira	200.00	20
789	Clara	Silva	300.00	30
855	Edu	Snatos	600.00	30
963	Deyse	Silva	330.00	NULL

Having

```
SELECT Departamento_Codigo, AVG(Salario)
FROM Empregado
GROUP BY Departamento_Codigo
HAVING AVG(Salario) > 500.00;
```

Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo
123	Igor	Pereira	100.00	10
159	Denise	Moreno	440.00	40
369	Marcelo	Neiva	900.00	40
456	Ana	Oliveira	200.00	20
789	Clara	Silva	300.00	30
855	Edu	Snatos	600.00	30
963	Deyse	Silva	330.00	NULL

Having

```
SELECT Departamento_Codigo, AVG(Salario)
FROM Empregado
WHERE Salario > 350.00
GROUP BY Departamento_Codigo
HAVING AVG(Salario) > 500.00;
```

Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo
159	Denise	Moreno	440.00	40
369	Marcelo	Neiva	900.00	40
855	Edu	Snatos	600.00	30
				NULL

Consulta Completa

SELECT COL1, COL2,...,COLN
FROM NOME_TABELA
WHERE (expressão lógica)
GROUP BY (atributos de agrupamento)
HAVING (condição de agrupamento)
ORDER BY (lista de atributos)

Consultas Aninhadas

Consultas Aninhadas

- Nome dos departamentos que possuem empregados.

Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo
123	Igor	Pereira	100.00	10
159	Denise	Moreno	440.00	40
369	Marcelo	Neiva	900.00	40
456	Ana	Oliveira	200.00	20
789	Clara	Silva	300.00	30
855	Edu	Snatos	600.00	30
963	Deyse	Silva	330.00	NULL

Codigo	Nome	UF
10	RH	RJ
20	TI	RJ
30	Logística	DF
40	Financeiro	DF
50	Venda	RS
60	Pesquisa	RJ

Subconsultas não-correlacionadas

```
SELECT Nome
FROM Departamento
WHERE Codigo IN (SELECT Departamento_Codigo
                 FROM Empregado
                 WHERE Departamento_Codigo IS NOT NULL);
```

Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo
123	Igor	Pereira	100.00	10
159	Denise	Moreno	440.00	40
369	Marcelo	Neiva	900.00	40
456	Ana	Oliveira	200.00	20
789	Clara	Silva	300.00	30
855	Edu	Snatos	600.00	30
963	Deyse	Silva	330.00	NULL

Codigo	Nome	UF
10	RH	RJ
20	TI	RJ
30	Logística	DF
40	Financeiro	DF
50	Venda	RS
60	Pesquisa	RJ

Subconsultas correlacionadas

```
SELECT Nome
FROM Departamento
WHERE EXISTS (SELECT Departamento_Codigo
                     FROM Empregado
                     WHERE Departamento_Codigo IS NOT NULL
                     AND Codigo = Departamento_Codigo);
```

Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo
123	Igor	Pereira	100.00	10
159	Denise	Moreno	440.00	40
369	Marcelo	Neiva	900.00	40
456	Ana	Oliveira	200.00	20
789	Clara	Silva	300.00	30
855	Edu	Snatos	600.00	30
963	Deyse	Silva	330.00	NULL

Codigo	Nome	UF
10	RH	RJ
20	TI	RJ
30	Logística	DF
40	Financeiro	DF
50	Venda	RS
60	Pesquisa	RJ

Consultas Aninhadas

- Nome dos departamentos que não possuem empregados.

Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo
123	Igor	Pereira	100.00	10
159	Denise	Moreno	440.00	40
369	Marcelo	Neiva	900.00	40
456	Ana	Oliveira	200.00	20
789	Clara	Silva	300.00	30
855	Edu	Snatos	600.00	30
963	Deyse	Silva	330.00	NULL

Codigo	Nome	UF
10	RH	RJ
20	TI	RJ
30	Logística	DF
40	Financeiro	DF
50	Venda	RS
60	Pesquisa	RJ

Consultas Aninhadas

```
SELECT Nome
FROM Departamento
WHERE Codigo NOT IN (SELECT Departamento_Codigo
                     FROM Empregado
                     WHERE Departamento_Codigo IS NOT NULL);
```

Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo
123	Igor	Pereira	100.00	10
159	Denise	Moreno	440.00	40
369	Marcelo	Neiva	900.00	40
456	Ana	Oliveira	200.00	20
789	Clara	Silva	300.00	30
855	Edu	Snatos	600.00	30
963	Deyse	Silva	330.00	NULL

Codigo	Nome	UF
10	RH	RJ
20	TI	RJ
30	Logística	DF
40	Financeiro	DF
50	Venda	RS
60	Pesquisa	RJ

Subconsultas correlacionadas

```
SELECT Nome
FROM Departamento
WHERE NOT EXISTS (SELECT Departamento_Codigo
                  FROM Empregado
                  WHERE Departamento_Codigo IS NOT NULL
                  AND Codigo = Departamento_Codigo);
```

Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo
123	Igor	Pereira	100.00	10
159	Denise	Moreno	440.00	40
369	Marcelo	Neiva	900.00	40
456	Ana	Oliveira	200.00	20
789	Clara	Silva	300.00	30
855	Edu	Snatos	600.00	30
963	Deyse	Silva	330.00	NULL

Codigo	Nome	UF
10	RH	RJ
20	TI	RJ
30	Logística	DF
40	Financeiro	DF
50	Venda	RS
60	Pesquisa	RJ

Cross Join

Cross Join

```
Select *
From Empregado  CROSS JOIN Departamento;



Select *
From Empregado, Departamento;
```

Cross Join

42 Registros

Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo	Codigo	Nome	UF
123	Igor	Pereira	100.00	10	10	RH	RJ
123	Igor	Pereira	100.00	10	20	TI	RJ
123	Igor	Pereira	100.00	10	30	Logística	DF
123	Igor	Pereira	100.00	10	40	Financeiro	DF
123	Igor	Pereira	100.00	10	50	Venda	RS
123	Igor	Pereira	100.00	10	60	Pesquisa	RJ
159	Denise	Moreno	440.00	40	10	RH	RJ
159	Denise	Moreno	440.00	40	20	TI	RJ
159	Denise	Moreno	440.00	40	30	Logística	DF
159	Denise	Moreno	440.00	40	40	Financeiro	DF
159	Denise	Moreno	440.00	40	50	Venda	RS
159	Denise	Moreno	440.00	40	60	Pesquisa	RJ
369	Marcelo	Neiva	900.00	40	10	RH	RJ
369	Marcelo	Neiva	900.00	40	20	TI	RJ
369	Marcelo	Neiva	900.00	40	30	Logística	DF
...	...	...	...	...	...	...	...

Inner Join

Inner Join

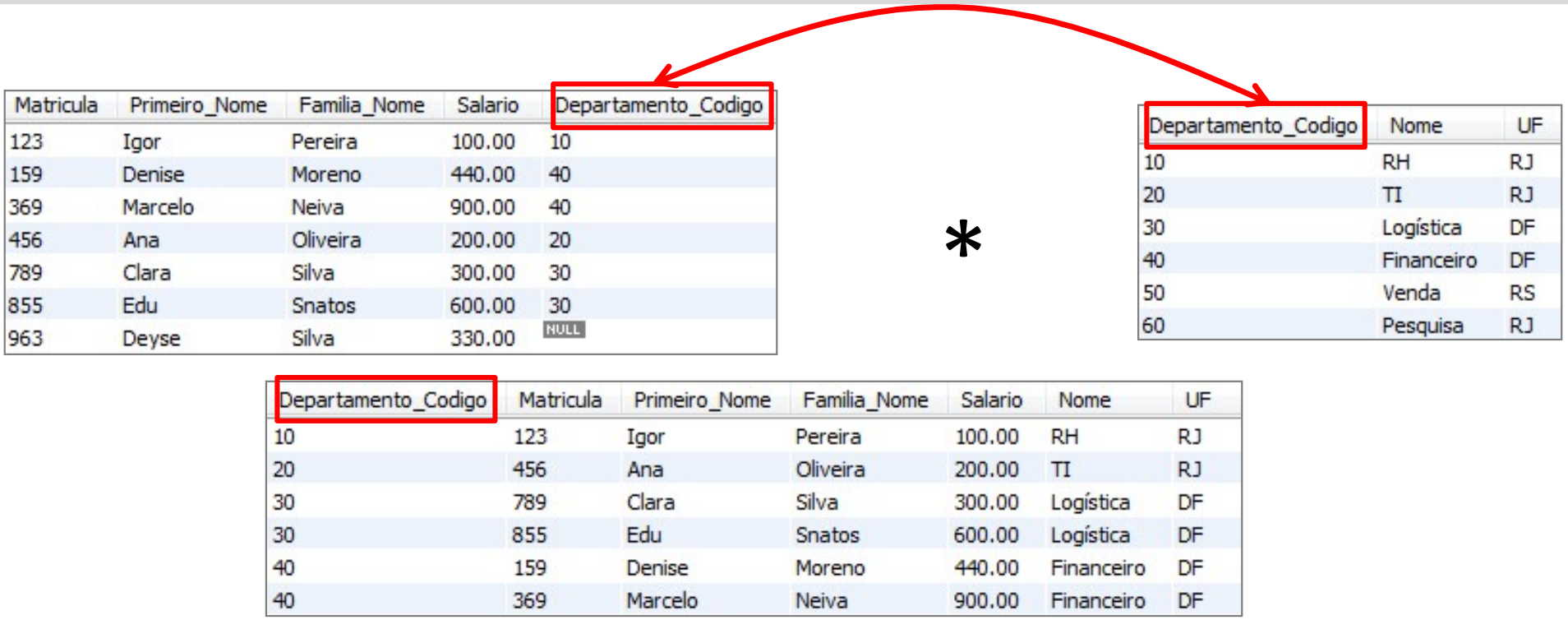
```
Select *
From Empregado  INNER JOIN Departamento
ON Departamento_Codigo = Codigo;

Select *
From Empregado, Departamento
WHERE Departamento_Codigo = Codigo;
```

Inner Join

Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo	Codigo	Nome	UF
123	Igor	Pereira	100.00	10	10	RH	RJ
159	Denise	Moreno	440.00	40	40	Financeiro	DF
369	Marcelo	Neiva	900.00	40	40	Financeiro	DF
456	Ana	Oliveira	200.00	20	20	TI	RJ
789	Clara	Silva	300.00	30	30	Logística	DF
855	Edu	Snatos	600.00	30	30	Logística	DF

Natural Join

```
Select *
From Empregado  NATURAL JOIN Dep;
```



Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo
123	Igor	Pereira	100.00	10
159	Denise	Moreno	440.00	40
369	Marcelo	Neiva	900.00	40
456	Ana	Oliveira	200.00	20
789	Clara	Silva	300.00	30
855	Edu	Snatos	600.00	30
963	Deyse	Silva	330.00	NULL

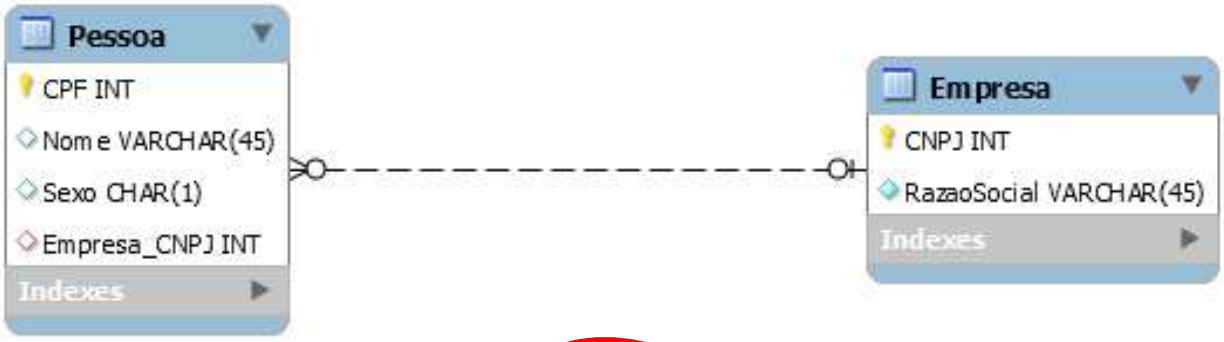
*

Departamento_Codigo	Nome	UF
10	RH	RJ
20	TI	RJ
30	Logística	DF
40	Financeiro	DF
50	Venda	RS
60	Pesquisa	RJ

Departamento_Codigo	Matricula	Primeiro_Nome	Familia_Nome	Salario	Nome	UF
10	123	Igor	Pereira	100.00	RH	RJ
20	456	Ana	Oliveira	200.00	TI	RJ
30	789	Clara	Silva	300.00	Logística	DF
30	855	Edu	Snatos	600.00	Logística	DF
40	159	Denise	Moreno	440.00	Financeiro	DF
40	369	Marcelo	Neiva	900.00	Financeiro	DF

Cross Join x Inner Join

Cross Join x Inner Join



CPF	Nome	Sexo	Empresa_CNPJ
111	Edu	M	
222	Ana	F	123
333	Igor	M	123
555	Lia	F	789

CNPJ	RazaoSocial
123	IBM
456	Dell
789	Apple

Cross Join x Inner Join

```
SELECT*
FROM Pessoa CROSS JOIN Empresa;
```

CPF	Nome	Sexo	Empresa_CNPJ	CNPJ	RazaoSocial
111	Edu	M	NULL	123	IBM
111	Edu	M	NULL	456	Dell
111	Edu	M	NULL	789	Apple
222	Ana	F	123	123	IBM
222	Ana	F	123	456	Dell
222	Ana	F	123	789	Apple
333	Igor	M	123	123	IBM
333	Igor	M	123	456	Dell
333	Igor	M	123	789	Apple
555	Lia	F	789	123	IBM
555	Lia	F	789	456	Dell
555	Lia	F	789	789	Apple

Cross Join x Inner Join

```
SELECT *
FROM Pessoa INNER JOIN Empresa
ON Empresa_CNPJ = CNPJ;
```

CPF	Nome	Sexo	Empresa_CNPJ	CNPJ	RazaoSocial
222	Ana	F	123	123	IBM
333	Igor	M	123	123	IBM
555	Lia	F	789	789	Apple

Cross Join x Inner Join

```
SELECT *
FROM Pessoa , Empresa;
```

```
SELECT *
FROM Pessoa, Empresa
WHERE Empresa_CNPJ = CNPJ;
```

Outer Join

Left Outer Join

```
Select *
From Empregado  LEFT OUTER JOIN Departamento
ON Departamento_Codigo = Codigo;
```

Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo	Codigo	Nome	UF
123	Igor	Pereira	100.00	10	10	RH	RJ
159	Denise	Moreno	440.00	40	40	Financeiro	DF
369	Marcelo	Neiva	900.00	40	40	Financeiro	DF
456	Ana	Oliveira	200.00	20	20	TI	RJ
789	Clara	Silva	300.00	30	30	Logística	DF
855	Edu	Snatos	600.00	30	30	Logística	DF
963	Deyse	Silva	330.00	NULL	NULL	NULL	NULL

Right Outer Join

```
Select *
From Empregado  RIGHT OUTER JOIN Departamento
ON Departamento_Codigo = Codigo;
```

Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo	Codigo	Nome	UF
123	Igor	Pereira	100.00	10	10	RH	RJ
456	Ana	Oliveira	200.00	20	20	TI	RJ
789	Clara	Silva	300.00	30	30	Logística	DF
855	Edu	Snatos	600.00	30	30	Logística	DF
159	Denise	Moreno	440.00	40	40	Financeiro	DF
369	Marcelo	Neiva	900.00	40	40	Financeiro	DF
NULL	NULL	NULL	NULL	NULL	50	Venda	RS
NULL	NULL	NULL	NULL	NULL	60	Pesquisa	RJ

FULL Outer Join

```
Select *
From Empregado  FULL OUTER JOIN Departamento
ON Departamento_Codigo = Codigo;
```

```
Select * From
Empregado  LEFT OUTER JOIN Departamento
ON Departamento_Codigo = Codigo
UNION
Select *
From Empregado  RIGHT OUTER JOIN Departamento
ON Departamento_Codigo = Codigo;
```

FULL Outer Join

Matricula	Primeiro_Nome	Familia_Nome	Salario	Departamento_Codigo	Codigo	Nome	UF
123	Igor	Pereira	100.00	10	10	RH	RJ
159	Denise	Moreno	440.00	40	40	Financeiro	DF
369	Marcelo	Neiva	900.00	40	40	Financeiro	DF
456	Ana	Oliveira	200.00	20	20	TI	RJ
789	Clara	Silva	300.00	30	30	Logística	DF
855	Edu	Snatos	600.00	30	30	Logística	DF

Fim