# JOINs:

Chave simples

Chave múltipla

- com funções

[INNER] JOIN

LEFT JOIN

RIGHT JOIN

FULL JOIN

CROSS JOIN

JOIN com NULLs

SELECT \* FROM e1 JOIN e2 ON e1.C1=e2.C1

## Tipos de JOINs

### INNER JOIN e CROSS JOIN

INNER JOIN equivale a um CROSS JOIN com WHERE

SELECT \* FROM pessoa INNER JOIN fone ON Pessoa.ID\_Pessoa = Fone.ID\_Pessoa

SELECT \* FROM pessoa CROSS JOIN fone WHERE Pessoa.ID\_Pessoa = Fone.ID\_Pessoa

SELECT \* FROM pessoa, fone WHERE Pessoa.ID\_Pessoa = Fone.ID\_Pessoa

### LEFT JOIN (e RIGHT JOIN)

Lista todos os registros à esquerda, e apenas os que existem na cláusula da tabela à direita

SELECT \* FROM pessoa JOIN fone ON Pessoa.ID\_Pessoa = Fone.ID\_Pessoa

 JOIN Fone\_Tipo ft ON Fone.ID\_Fone\_Tipo = ft.ID\_Fone\_Tipo

SELECT \* FROM pessoa JOIN fone ON Pessoa.ID\_Pessoa = Fone.ID\_Pessoa

 LEFT JOIN Fone\_Tipo ft ON Fone.ID\_Fone\_Tipo = ft.ID\_Fone\_Tipo

**Não tem equivalência com um CROSS JOIN diretamente.**

SELECT \* FROM pessoa JOIN fone ON Pessoa.ID\_Pessoa = Fone.ID\_Pessoa

 LEFT JOIN Fone\_Tipo ft ON Fone.ID\_Fone\_Tipo = ft.ID\_Fone\_Tipo

SELECT \* FROM pessoa JOIN fone ON Pessoa.ID\_Pessoa = Fone.ID\_Pessoa

 CROSS JOIN Fone\_Tipo ft

 WHERE Fone.ID\_Fone\_Tipo = ft.ID\_Fone\_Tipo

Registros são filtrados ANTES do cruzamento:

SELECT \* FROM pessoa INNER JOIN fone ON Pessoa.ID\_Pessoa = Fone.ID\_Pessoa OR Fone.ID\_Pessoa=2

SELECT \* FROM pessoa LEFT JOIN fone ON Pessoa.ID\_Pessoa = Fone.ID\_Pessoa AND ID\_Fone=4

SELECT \* FROM pessoa LEFT JOIN fone ON Pessoa.ID\_Pessoa = Fone.ID\_Pessoa

 WHERE ID\_Fone=4 OR ID\_Fone IS NULL

## NULL em JOINs

Não existe restrição quanto a NULL em JOINs, porém, as comparações com NULL devem ser feitas corretamente.

Existe restrições para NULL em chaves primárias.

## SELECT com VALUES e CASE (‘de’-‘para’)

SELECT \*

 ,CASE ID\_Fone\_Tipo

 WHEN 1 THEN 'Celular'

 WHEN 2 THEN 'Fixo'

 WHEN 3 THEN 'Fax'

 ELSE NULL

 END

 FROM Pessoa

 JOIN Fone ON Pessoa.ID\_Pessoa = Fone.ID\_Pessoa

ou

;

WITH

 Tipo AS (SELECT \* FROM (VALUES

 (1, 'Celular'),

 (2, 'Fixo'),

 (3, 'Fax')

 ) Tipo (k, v))

SELECT \*

 FROM Pessoa

 JOIN Fone ON Pessoa.ID\_Pessoa = Fone.ID\_Pessoa

 LEFT JOIN Tipo ft ON Fone.ID\_Fone\_Tipo = ft.k

ou

SELECT \*

 FROM Pessoa

 JOIN Fone ON Pessoa.ID\_Pessoa = Fone.ID\_Pessoa

 LEFT JOIN (VALUES

 (1, 'Celular'),

 (2, 'Fixo'),

 (3, 'Fax')

 ) Tipo (k, v) ON Fone.ID\_Fone\_Tipo = k

## Criar valores com CROSS JOIN / VALUES:

### cria números de 0 a 99999

with d as (select v from (VALUES (0),(1),(2),(3),(4),(5),(6),(7),(8),(9)) AS X(v))

select d1.v + d2.v\*10 + d3.v\*100 + d4.v\*1000 + d5.v\*10000

 from d d1, d d2, d d3, d d4, d d5 order by 1

### AnoMes de 2000/01 a 2029/12

select ((D.v\*10+U.v)+2000)\*100+Mes as AnoMes

 from (VALUES (1),(2),(3),(4),(5),(6),(7),(8),(9),(10),(11),(12)) AS X(Mes)

 join (VALUES (0),(1),(2),(3),(4),(5),(6),(7),(8),(9)) AS U(v) ON 1=1

 join (VALUES (0),(1),(2)) AS D(v) ON 1=1