

{[INNER] | {LEFT | RIGHT | FULL}[OUTER]} JOIN

Bruno A. N. Travençolo – FACOM.

Sintaxe SELECT

```
SELECT [ ALL | DISTINCT [ ON ( expressão [, ...] ) ] ]  
* | expressão [ AS nome_de_saída ] [, ...]  
[ FROM item_do_from [, ...] ] -- aula de hoje  
[ WHERE condição ]  
[ GROUP BY expressão [, ...] ]  
[ HAVING condição [, ...] ]  
[ { UNION | INTERSECT | EXCEPT } [ ALL ] seleção ]  
[ ORDER BY expressão [ ASC | DESC | USING operador ] [, ...] ]  
[ LIMIT { contador | ALL } ]  
[ OFFSET início ]  
[ FOR UPDATE [ OF nome_da_tabela [, ...] ] ]
```



Itens FROM

onde item_do_from pode ser um entre:

[ONLY] nome_da_tabela [*] [[AS] alias [(alias_de_coluna [, ...])]]

(seleção) [AS] alias [(alias_de_coluna [, ...])]

nome_da_função ([argumento [, ...]]) [AS] alias [(alias_de_coluna [, ...]
| definição_de_coluna [, ...])]

nome_da_função ([argumento [, ...]]) AS (definição_de_coluna [, ...])

item_do_from [NATURAL] tipo_de_junção item_do_from [ON condição_de_junção
| USING (coluna_de_junção [, ...])]

Onde tipo_de_junção pode ser:

[INNER] JOIN

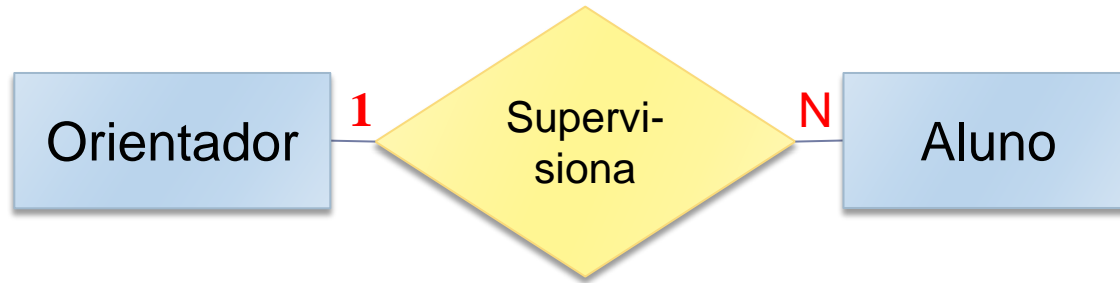
LEFT [OUTER] JOIN

RIGHT [OUTER] JOIN

FULL [OUTER] JOIN

CROSS JOIN



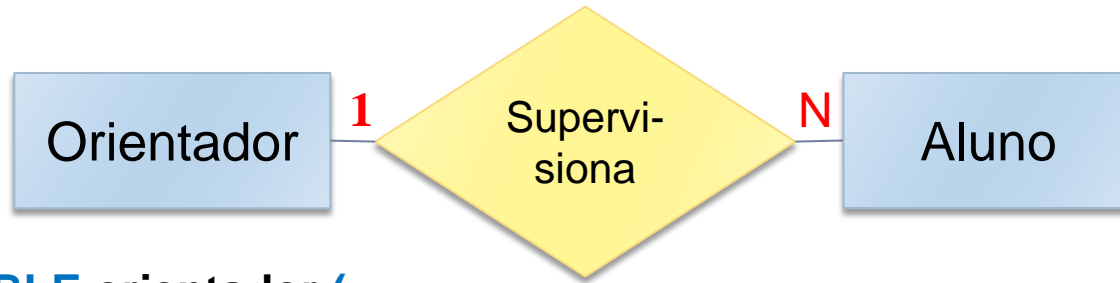


/ Criando as tabelas*/*

```
CREATE TABLE orientador (  
  id INT PRIMARY KEY,  
  nome VARCHAR(255)  
);
```

```
CREATE TABLE aluno (  
  matricula INT PRIMARY KEY,  
  nome VARCHAR(255),  
  orientador_id INT REFERENCES orientador(id)  
);
```





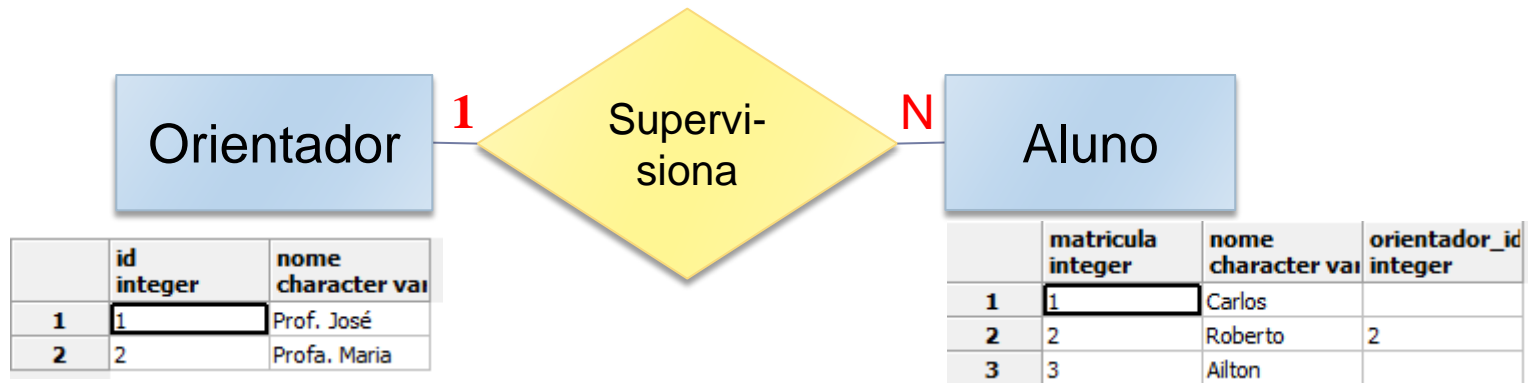
```
CREATE TABLE orientador (  
  id INT PRIMARY KEY,  
  nome VARCHAR(255)  
);
```

```
CREATE TABLE aluno (  
  matricula INT PRIMARY KEY,  
  nome VARCHAR(255),  
  orientador_id INT REFERENCES orientador(id)  
);
```

/ Povoando as tabelas */*

```
INSERT INTO orientador VALUES (1,'Prof. José'), (2,'Profa. Maria');  
INSERT INTO aluno VALUES (1,'Carlos',NULL), (2,'Roberto',2),  
(3,'Ailton',NULL)
```





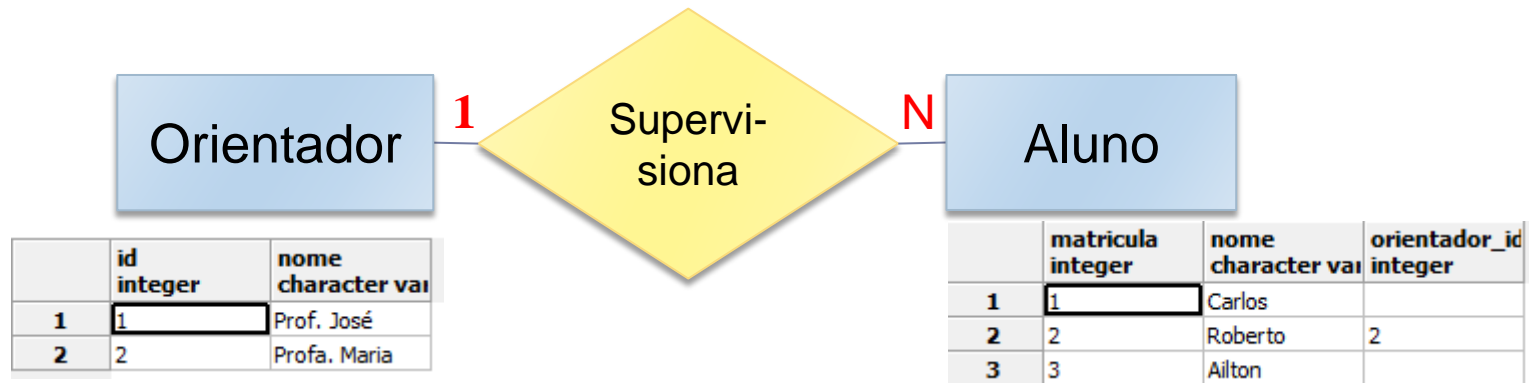
```

SELECT *
FROM orientador INNER JOIN aluno
ON aluno.orientador_id = orientador.id;

```

	id integer	nome character vari	matricula integer	nome character varying	orientador_id integer
1	2	Profª. Maria	2	Roberto	2

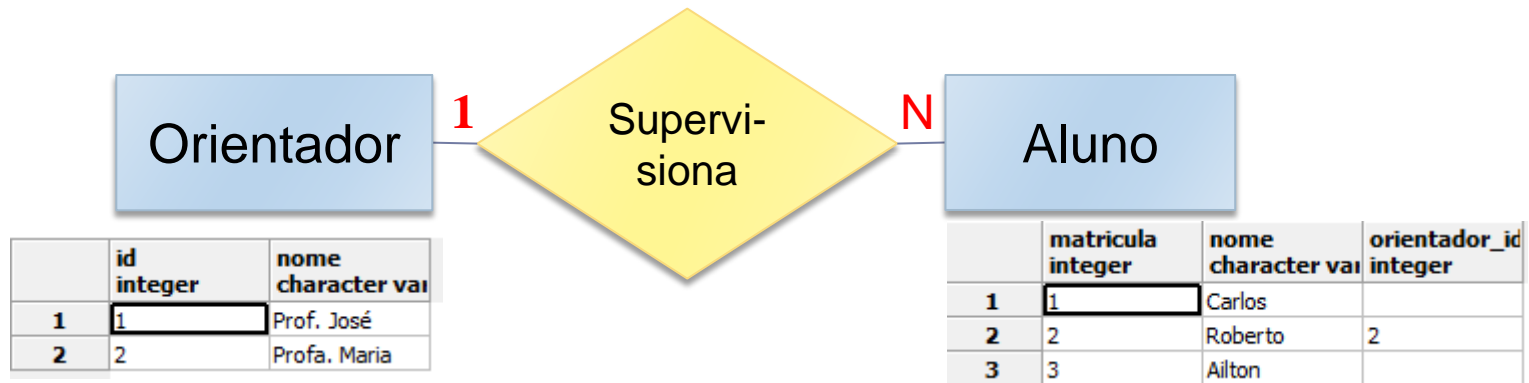




```
SELECT *
FROM orientador LEFT OUTER JOIN aluno
ON aluno.orientador_id = orientador.id;
```

	id integer	nome character varying	matricula integer	nome character varying	orientador_id integer
1	1	Prof. José			
2	2	Prof. Maria	2	Roberto	2





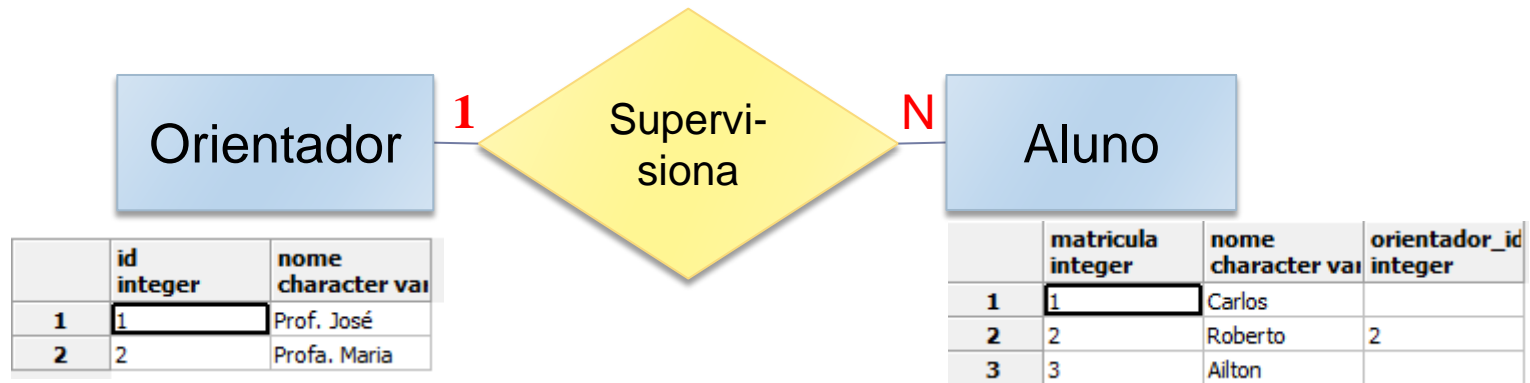
```

SELECT *
FROM orientador RIGHT OUTER JOIN aluno
ON aluno.orientador_id = orientador.id;

```

	id integer	nome character vai	matricula integer	nome character varying	orientador_id integer
1			1	Carlos	
2	2	Profa. Maria	2	Roberto	2
3			3	Ailton	





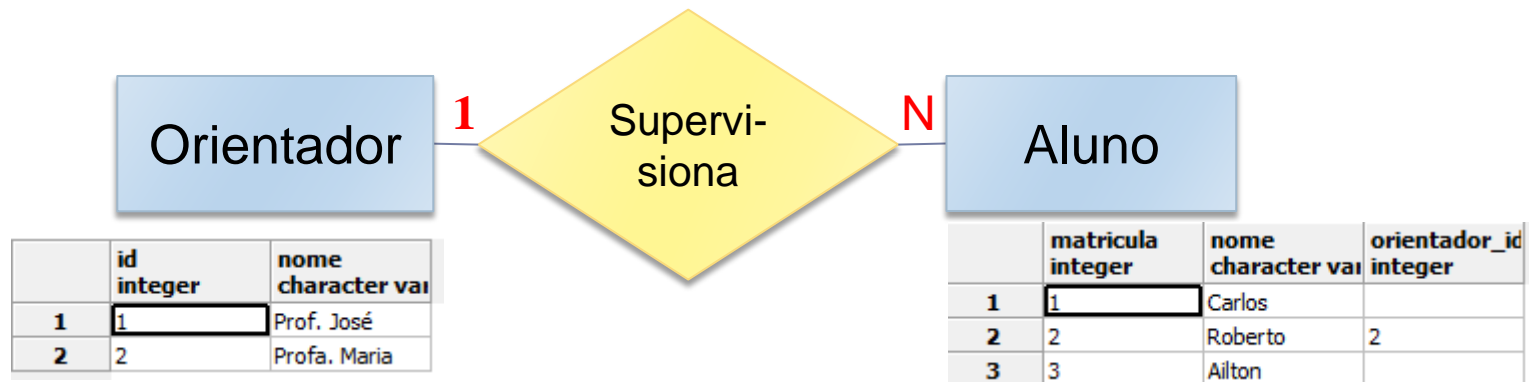
```

SELECT *
FROM orientador FULL OUTER JOIN aluno
ON aluno.orientador_id = orientador.id;

```

	id integer	nome character vai	matricula integer	nome character vai	orientador_id integer
1	1	Prof. José			
2	2	Profa. Maria	2	Roberto	2
3			1	Carlos	
4			3	Ailton	





SELECT *
FROM orientador CROSS JOIN aluno;

	id integer	nome character vai	matricula integer	nome character vai	orientador_id integer
1	1	Prof. José	1	Carlos	
2	1	Prof. José	2	Roberto	2
3	1	Prof. José	3	Ailton	
4	2	Profa. Maria	1	Carlos	
5	2	Profa. Maria	2	Roberto	2
6	2	Profa. Maria	3	Ailton	

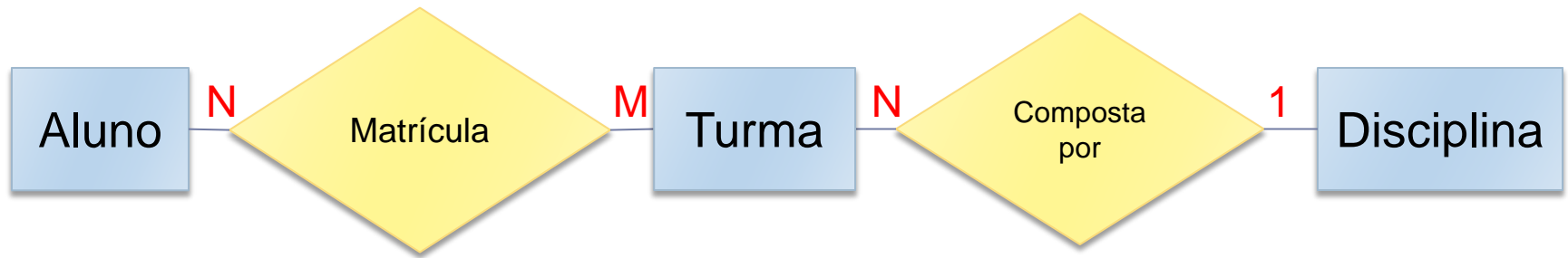
OUTER JOINS

- ▶ Variante da operação de JOIN que baseia-se em valores NULL. O resultado de um OUTER JOIN é igual a de um INNER JOIN mas com a inclusão das tuplas que não satisfazem a condição de JOIN.
- ▶ Três variantes:
 - ▶ LEFT OUTER JOIN
 - ▶ As tuplas da tabela à esquerda que não obedecem a condição do JOIN aparecem na resposta
 - ▶ RIGHT OUTER JOIN
 - ▶ As tuplas da tabela à direita que não obedecem a condição do JOIN aparecem na resposta
 - ▶ FULL OUTER JOIN
 - ▶ As tuplas das duas tabelas que não obedecem a condição do JOIN aparecem na resposta



Mais exemples..

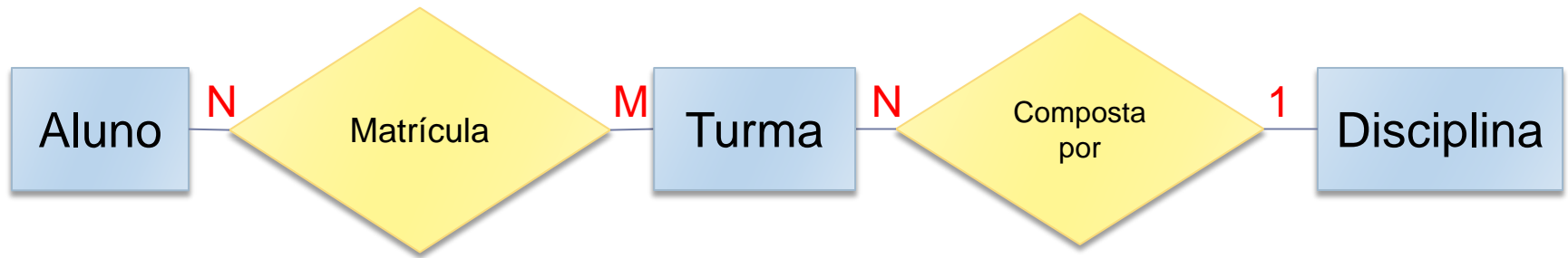




/ Liste as matriculas efetuadas em cada turma de cada disciplina */*

```
SELECT Aluno.NMat, Aluno.NOME,  
Matricula.CODIGOTURMA,  
Turma.SIGLA, Turma.NUMERO, Discip.NOME  
FROM Discip INNER JOIN  
    (Turma INNER JOIN  
        (Aluno INNER JOIN Matricula  
            ON Aluno.NMat = Matricula.NMat)  
        ON Turma.Codigo = Matricula.CODIGOTURMA)  
    ON Discip.SIGLA = Turma.SIGLA;
```

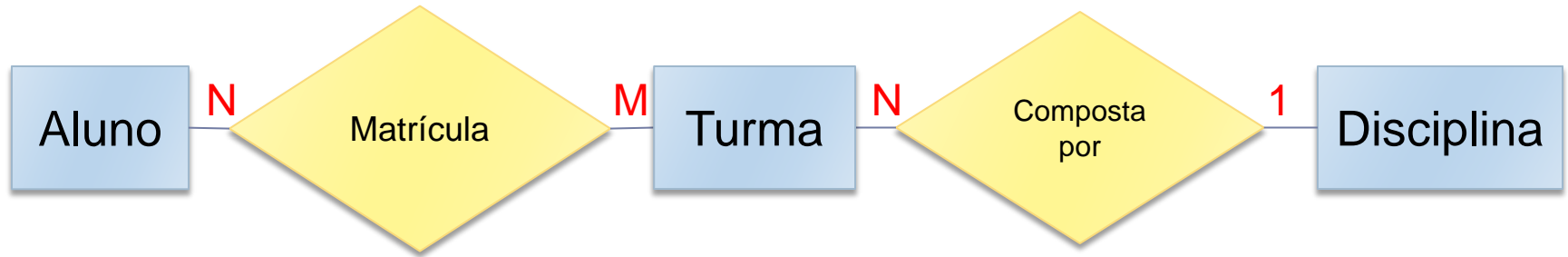




/ Liste as matriculas efetuadas em cada turma de cada disciplina – alterando a ordem das junções*/*

```
SELECT Aluno.NMat, Aluno.NOME,  
Matricula.CODIGOTURMA,  
Turma.SIGLA, Turma.NUMERO, Discip.NOME  
FROM Aluno INNER JOIN  
    (Matricula INNER JOIN  
        (Turma INNER JOIN Discip  
            ON Turma.SIGLA = Discip.SIGLA)  
        ON Matricula.CODIGOTURMA = Turma.Codigo)  
    ON Aluno.NMat = Matricula.NMat;
```





/ Liste as matriculas efetuadas em cada turma de cada disciplina – Comando equivalente usando WHERE*/*

```
SELECT A.NMat, A.NOME, M.CODIGOTURMA, T.SIGLA,  
        T.NUMERO, D.NOME  
FROM Aluno A, Matricula M, Turma T, DiscipD  
WHERE T.SIGLA = D.SIGLA AND  
        M.CODIGOTURMA = T.Codigo AND  
        A.NMat = M.NMat;
```

