



**INSTITUTO
FEDERAL**
Pernambuco

Consulta SQL

- JOIN -

Banco de Dados 1

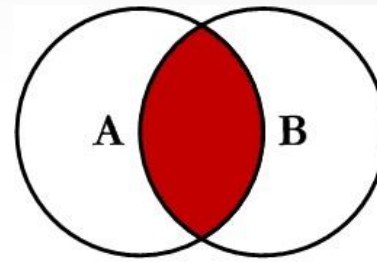
Profa. Carolina Torres

Join

- Exemplo esquema

```
CREATE TABLE TabelaA( Nome varchar(50) NULL );  
CREATE TABLE TabelaB( Nome varchar(50) NULL );  
INSERT INTO TabelaA VALUES  
('Fernanda'),('Luiz'),('Josefa'),('Fernando');  
INSERT INTO TabelaB VALUES  
('Fernando'),('Carlos'),('Manoel'),('Luiz');
```

Join



- **Natural Join:** retorna os registros que são comuns às duas tabelas.
 - Tabelas listadas na cláusula FROM e definindo o cruzamento na WHERE.

```
select A.nome, B.nome  
from TabelaA A, TabelaB B WHERE A.Nome = B.nome
```

- **Inner JOIN:** retorna os registros que são comuns às duas tabelas

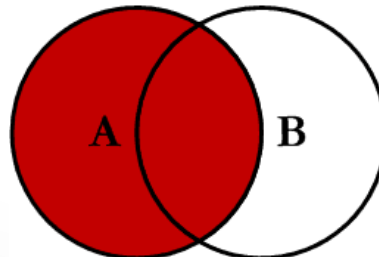
```
select A.nome, B.nome  
FROM TabelaA A INNER JOIN TabelaB B  
ON A.Nome = B.nome
```



Join

- **Left Join:** tem como resultado todos os registros que estão na tabela A (mesmo que não estejam na tabela B) e os registros da tabela B que são comuns à tabela A.

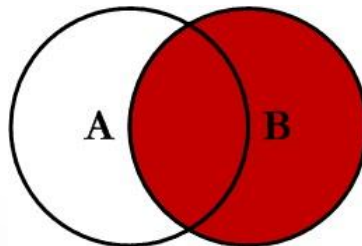
```
SELECT A.nome, B.nome  
FROM TabelaA A LEFT OUTER JOIN TabelaB B  
ON A.Nome = B.nome
```



Join

- **Right Join:** tem como resultado todos os registros que estão na tabela B (mesmo que não estejam na tabela A) e os registros da tabela A que são comuns à tabela B.

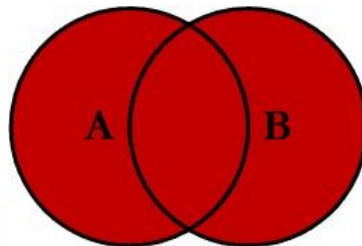
```
SELECT A.nome, B.nome  
FROM TabelaA A RIGHT JOIN TabelaB B  
ON A.Nome = B.nome
```



Join

- **Outer Join ou Full join:** todos os registros que estão na tabela A e todos os registros da tabela B.

```
SELECT A.nome, B.nome  
FROM TabelaA A FULL JOIN TabelaB B  
ON A.Nome = B.nome
```





**INSTITUTO
FEDERAL**
Pernambuco

Exemplo prático

...

Esquema adotado

```
CREATE TABLE area (  
    codigo int primary key ,  
    descricao varchar (30),  
    predio varchar (40)  
);  
CREATE TABLE curso (  
    codigo int primary key ,  
    nome varchar (40) not null,  
    cod_area int,  
    vagas integer,  
    nome_coordenador varchar(50),  
    foreign key (cod_area)  
        references area (codigo)  
);  
CREATE TABLE aluno (  
    matricula int primary key ,  
    nome varchar (50) not null,  
    cidade_endereco varchar (30),  
    telefone int,  
    mensalidade decimal (6,2),  
    media_notas decimal (3,1) not null,  
    data_nascimento date,  
    cod_curso int not null,  
    foreign key (cod_curso)  
        references curso (codigo)  
); ●
```

```
insert into area values  
(1,"Exatas", "Bloco C"), (2,"Saúde", "Bloco B"),  
(3,"Humanas", "Bloco A");
```

```
insert into curso values  
(1,"Informatica para Internet", 1, 20, "Francisco"),  
(2,"Nutrição", 2, 25, null),  
(3,"Enfermagem", 2, 30, "Maria"),  
(4,"Ciências da Computação", 1, 30, null),  
(5,"Redes de Computadores", 1, 20, "Zilmara");
```

```
INSERT INTO aluno values  
(1,"Mariana Torres", "Recife", null, 815.78, 8.9, '1998-10-19', 1),  
(2,"Carolina Pereira", "Olinda", 982736410, 726.37, 7.5, null, 1),  
(3,"Adriano Freire", "Palmares", NULL, 982.62, 5.2, '1994-07-05', 5),  
(4,"Elaine Villas", "Olinda", 902816253, 856.01, 5.8, '2000-04-29', 3),  
(5,"Paulo Veras", "Olinda", 976253123, 582.71, 6.3, '1988-03-30', 3),  
(6,"Talita Veiga", "Jaboatão", 952434172, 837.29, 8.7, '1990-11-23', 4),  
(7,"Katia Garcia", "Palmares", 962534122, 526.62, 9.7, '1991-10-19', 5),  
(8,"Júlio Mercedes", "Palmares", null, 837.73, 7.6, null, 3),  
(9,"Fátima Silva", "Jaboatão", 981722639, 549.91, 9.4, '1986-09-04', 5);
```


Consultas

1. Exibir a quantidade de cursos que cada área tem. Áreas sem cursos relacionados devem aparecer com 0.
2. Exibir a relação de nomes da área, do curso e dos alunos, além das médias, ordenados decrescentemente pela média
3. Apresentar a lista de nomes dos cursos e de seus alunos ordenados pelo nome do curso e depois pelo nome dos alunos. Os cursos que não tiverem alunos matriculados devem aparecer acompanhados de null.
4. Para cada área, deve exibir o nome e a quantidade de alunos matriculados em cursos dessa área.

1

- `select area.descricao as areas,
count(curso.nome) as QTDE_CURSOS`
- `from curso right JOIN area on
area.codigo = curso.cod_area`
- `group by area.descricao`

2

```
select area.descricao "ÁREA",  
curso.nome as curso, aluno.nome as  
ALUNO, media_notas as "MÉDIA"  
from curso LEFT JOIN aluno ON  
aluno.cod_curso = curso.codigo  
right JOIN area on area.codigo =  
curso.cod_area  
ORDER BY media_notas DESC;
```

3

- `select curso.nome as curso,
aluno.nome as ALUNO`
- `from curso LEFT JOIN aluno ON
aluno.cod_curso = curso.codigo`
- `ORDER BY 1, 2;`

4

- `select area.descricao as areas,
count(aluno.nome) as ALUNO`
- `from curso JOIN aluno ON
aluno.cod_curso = curso.codigo`
- `right JOIN area on area.codigo
= curso.cod_area`
- `group by area.descricao`



**INSTITUTO
FEDERAL**
Pernambuco

Consulta SQL

- JOIN -

Banco de Dados 1

Profa. Carolina Torres