Banco de Dados II

Professora Eliane Marion

ETEC PROF. HORÁCIO AUGUSTO DA SILVEIRA

SQL : JOINS

Pesquisa em várias tabelas

Podemos utilizar:

- Inner Join
- Outer Joins
  - Left Join
  - Right Join
  - Full Join

LEFT JOIN

Junção por diferença

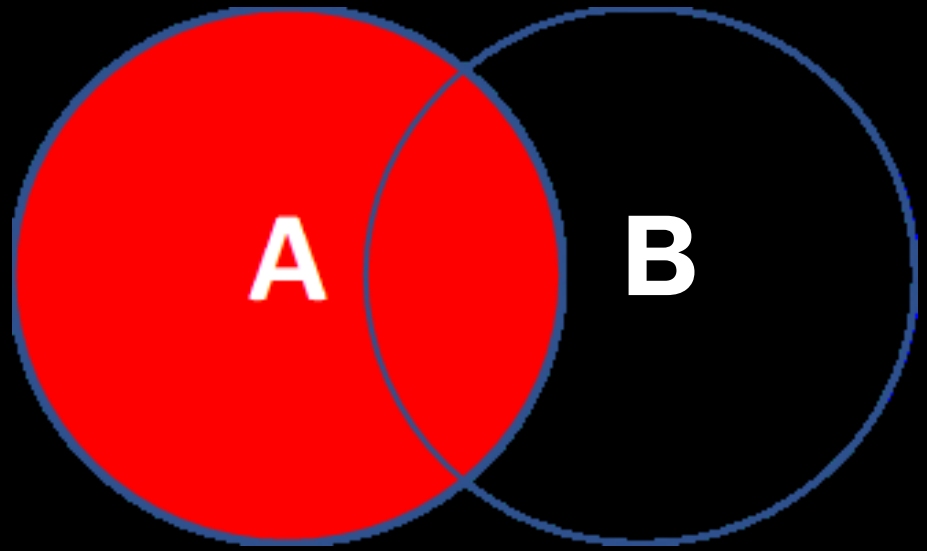
Professora Eliane Marion

ETEC PROF. HORÁCIO AUGUSTO DA SILVEIRA

SQL : LEFT JOIN

Conceito:

Retorna todas as linhas de tabela à esquerda, mesmo se não houver nenhuma correspondência na tabela da direita



SQL : LEFT JOIN

Sintaxe:

```
SELECT coluna FROM tabela_esquerda LEFT_JOIN tabela_direita
 ON tabela_esquerda_coluna = tabela_direita_coluna
```

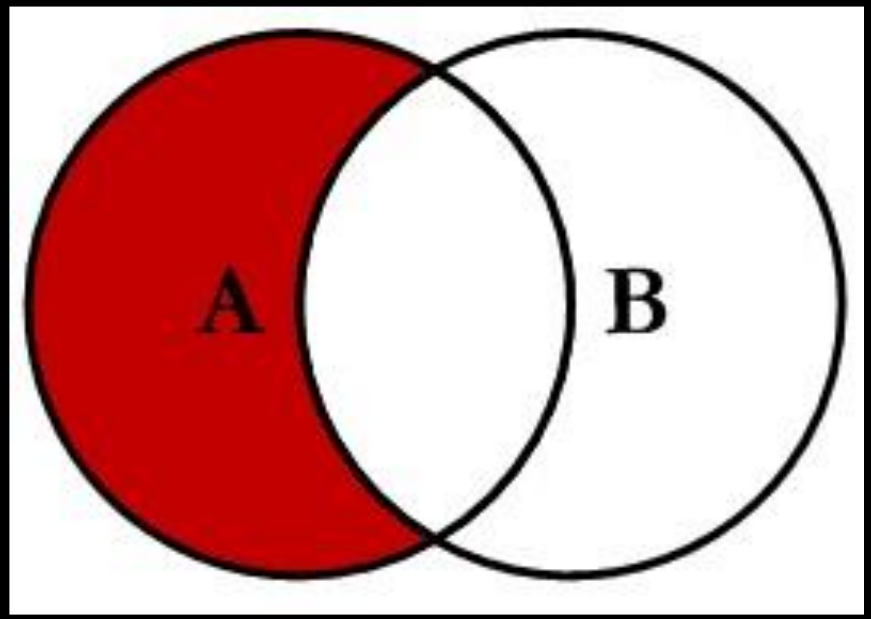
Exemplo:

```
SELECT * FROM livro LEFT_JOIN autor
 ON livro.id_autor = autor.id_autor
```

SQL : LEFT JOIN – Excluindo correspondência

Conceito:

Retorna todas as linhas de tabela à esquerda, excluindo o que pertencer à tabela da direita



SQL : LEFT JOIN

Sintaxe:

```
SELECT coluna FROM tabela_esquerda LEFT_JOIN tabela_direita
 ON tabela_esquerda_coluna = tabela_direita_coluna
 WHERE tabela_direita.coluna IS NULL
```

Exemplo:

```
SELECT * FROM livro LEFT_JOIN autor
 ON livro.id_autor = autor.id_autor
 WHERE autor.id_autor IS NULL;
```

RIGHT JOIN

Junção por diferença

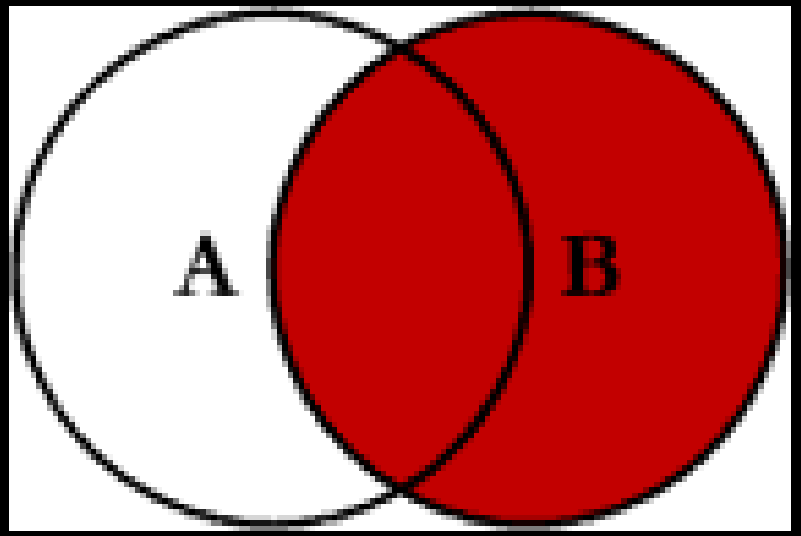
Professora Eliane Marion

ETEC PROF. HORÁCIO AUGUSTO DA SILVEIRA

SQL : RIGHT JOIN

Conceito:

Retorna todas as linhas de tabela à direita, mesmo se não houver nenhuma correspondência na tabela da esquerda



SQL : RIGHT JOIN

Sintaxe:

```
SELECT coluna FROM tabela_esquerda RIGHT_JOIN tabela_direita
 ON tabela_esquerda_coluna = tabela_direita_coluna
```

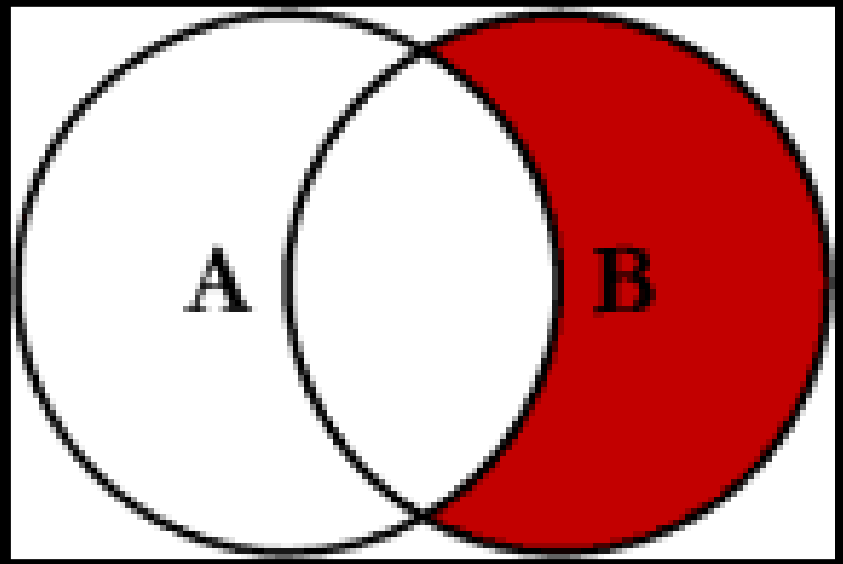
Exemplo:

```
SELECT * FROM livro RIGHT_JOIN editora
 ON livro.id_editora = editora.id_editora
```

SQL : RIGHT JOIN – Excluindo correspondência

Conceito:

Retorna todas as linhas de tabela à direita, excluindo o que pertencer à tabela da esquerda



SQL : LEFT JOIN

Sintaxe:

```
SELECT coluna FROM tabela_esquerda RIGHT_JOIN tabela_direita
 ON tabela_esquerda_coluna = tabela_direita_coluna
 WHERE tabela_esquerda.coluna IS NULL
```

Exemplo:

```
SELECT * FROM livro RIGHT JOIN editora
 ON livro.id_editora = editora.id_editora
 WHERE livro.id_editora IS NULL;
```

EXERCÍCIOS

1. Crie uma base de dados escola.

2. Crie as tabelas abaixo:

```
CREATE TABLE Componente (
cd_compo numeric(3) PRIMARY KEY,
nome_compo varchar(20)
);
```

```
CREATE TABLE Aluno (
Rm numeric(6) PRIMARY KEY,
nome_aluno varchar(20)
);
```

```
CREATE TABLE Avaliacao (
nota1 char(2),
nota2 char(2),
media char(2),
situacao varchar(30),
cd_compo numeric(3),
Rm numeric(6) references aluno,
FOREIGN KEY(cd_compo) REFERENCES Componente
(cd_compo)
);
```

EXERCÍCIOS

3 - Inserir os dados abaixo:

```
insert into componente values(1, 'BDII');
insert into componente values(2, 'PWII');
insert into componente values(3, 'PAMI');
insert into componente values(4, 'MATEMATICA');
insert into aluno values(20, 'Marcel');
insert into aluno values(21, 'Simone');
insert into aluno values(22, 'Carla');
insert into aluno values(23, 'José');
insert into avaliacao values('R', 'B', 'B', 'Aprovado', 1, 20);
insert into avaliacao values('B', 'B', 'B', 'Aprovado', 2, 20);
insert into avaliacao values('I', 'I', 'I', 'Reprovado', 1, 21);
insert into avaliacao values('R', 'I', 'I', 'Reprovado', 2, 21);
insert into avaliacao values('R', 'B', 'B', 'Aprovado', 3, 21);
insert into avaliacao values('MB', 'B', 'B', 'Aprovado', 1, 22);
insert into avaliacao values('R', 'MB', 'B', 'Aprovado', 2, 22);
insert into avaliacao values(null, null, null, null, 2, 23);
```

EXERCÍCIOS

4 – Fazer as consultas:

a. Criar um relatório que mostre quais os componentes (nome) que os alunos (nome) estão cursando.
b. Mostrar o nome dos alunos, seus componentes (nome) e suas notas.
c. Exibir quais são os componentes (nome) que não possuem nenhum aluno (nome) cursando.
d. Mostre os alunos (nome) que não tenham nota em algum componente (nome)