



UC 6 - AULA 5

CRIAR E MANTER BANCO DE DADOS

● PROF. CALEBE PEREIRA LEMOS
EMAIL: CALEBE.PEREIRA@UFMS.BR



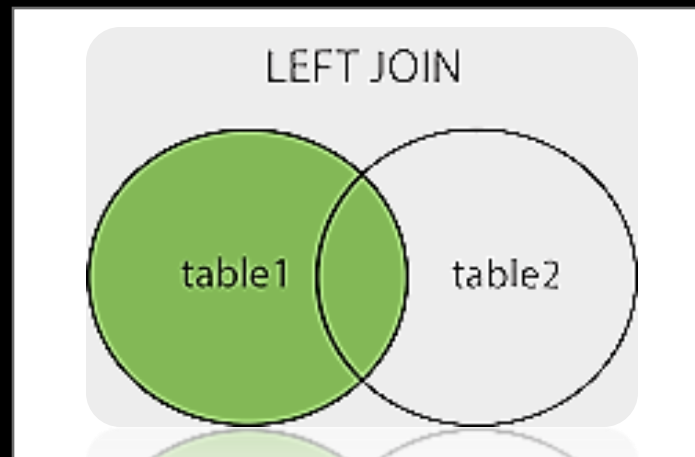


LEFT JOIN
RIGHT JOIN
CROSS JOIN



● Left Join

O comando LEFT JOIN é usado para criar junções que retornam todos os registros da tabela à esquerda (primeira tabela mencionada na consulta) e os registros correspondentes da tabela à direita (segunda tabela mencionada na consulta), se houver correspondência, caso contrário, os campos da tabela à direita são preenchidos como NULL.



● Exemplo - Left Join

Suponha duas tabelas: clientes e pedidos.

Tabela Clientes:

ID	NOME	EMAIL
1	JOÃO	JOAO@ICLOUD.COM
2	MARIA	MARIA@TERRA.COM
3	CARLOS	CARLOS@GMAIL.COM



● Exemplo - Left Join

Tabela Pedidos:

ID	CLIENTE_ID	PRODUTO
1	1	PRODUTO A
2	1	PRODUTO B
3	2	PRODUTO C

```
SELECT Clientes.ID, Clientes.Nome, Pedidos.Produto  
FROM Clientes  
LEFT JOIN Pedidos ON Clientes.ID = Pedidos.Cliente_ID;
```



● Exemplo - Left Join - Resultado

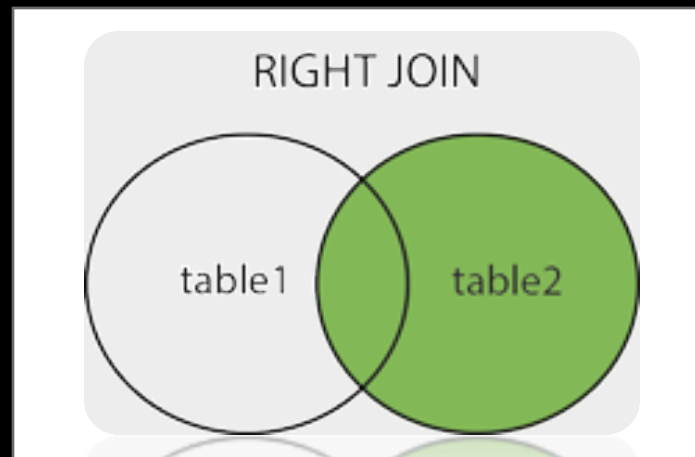
ID	NOME	PRODUTO
1	JOÃO	PRODUTO A
1	JOÃO	PRODUTO B
2	MARIA	PRODUTO C
3	CARLOS	NULL

```
SELECT Clientes.ID, Clientes.Nome, Pedidos.Produto  
FROM Clientes  
LEFT JOIN Pedidos ON Clientes.ID = Pedidos.Cliente_ID;
```



● Right Join

O comando RIGHT JOIN cria a junção que retorna todos os registros da tabela à direita (segunda tabela mencionada na consulta) e os registros correspondentes na tabela à esquerda (primeira tabela mencionada na consulta), preenchendo com NULL quando não há correspondências.



● Exemplo - Right Join

Suponha duas tabelas: clientes e pedidos.

Tabela Clientes:

ID	NOME	EMAIL
1	JOÃO	JOAO@ICLOUD.COM
2	MARIA	MARIA@TERRA.COM
3	CARLOS	CARLOS@GMAIL.COM



● Exemplo - Right Join

Tabela Pedidos:

ID	CLIENTE_ID	PRODUTO
1	1	PRODUTO A
2	1	PRODUTO B
3	2	PRODUTO C

```
SELECT Clientes.ID, Clientes.Nome, Pedidos.Produto  
FROM Clientes  
RIGHT JOIN Pedidos ON Clientes.ID = Pedidos.Cliente_ID;
```



Exemplo - Right Join - Resultado

ID	NOME	PRODUTO
1	JOÃO	PRODUTO A
1	JOÃO	PRODUTO B
2	MARIA	PRODUTO C

```
SELECT Clientes.ID, Clientes.Nome, Pedidos.Produto  
FROM Clientes  
RIGHT JOIN Pedidos ON Clientes.ID = Pedidos.Cliente_ID ;
```



● Exemplo – Consulta 2 : Right Join

ID	NOME	PRODUTO
1	JOÃO	PRODUTO A
1	JOÃO	PRODUTO B
2	MARIA	PRODUTO C
3	CARLOS	NULL

```
SELECT Clientes.ID, Clientes.Nome, Pedidos.Produto  
FROM Pedidos  
RIGHT JOIN Clientes ON Pedidos.Cliente_ID = Clientes.ID ;
```



● CROSS JOIN

O comando CROSS JOIN é usado para produzir o produto cartesiano de duas tabelas. Isso significa que ele combina cada linha de uma tabela com todas as linhas de outra tabela.



Exemplo – Cross Join

```
SELECT Clientes.Nome, Pedidos.Produto  
FROM Clientes  
CROSS JOIN Pedidos;
```

NOME	PRODUTO
JOÃO	PRODUTO A
MARIA	PRODUTO A
CARLOS	PRODUTO A
JOÃO	PRODUTO B
MARIA	PRODUTO B
CARLOS	PRODUTO B
JOÃO	PRODUTO C
MARIA	PRODUTO C
CARLOS	PRODUTO C





EXERCÍCIO



- Bora praticar!

Crie o *database* e as tabelas a seguir. Logo após, realize as inserções e as consultas propostas para prática de Join.






```
CREATE DATABASE ExemploJoins;
```

```
USE ExemploJoins;
```

```
CREATE TABLE Clientes (  
    ClienteID INT PRIMARY KEY,  
    Nome VARCHAR(50)  
);
```

```
CREATE TABLE Pedidos (  
    PedidoID INT PRIMARY KEY,  
    ClienteID INT,  
    DataPedido DATE,  
    FOREIGN KEY (ClienteID) REFERENCES Clientes(ClienteID)  
);
```






```
CREATE TABLE Produtos (  
    ProdutoID INT PRIMARY KEY,  
    NomeProduto VARCHAR(50),  
    CategoriaID INT,  
    Preco DECIMAL(10, 2),  
    FOREIGN KEY (CategoriaID) REFERENCES Categorias(CategoriaID)  
);
```


```
CREATE TABLE Categorias (  
    CategoriaID INT PRIMARY KEY,  
    NomeCategoria VARCHAR(50)  
);
```





```
CREATE TABLE DetalhesPedidos (  
    DetalheID INT PRIMARY KEY,  
    PedidoID INT,  
    ProdutoID INT,  
    Quantidade INT,  
    FOREIGN KEY (PedidoID) REFERENCES Pedidos(PedidoID),  
    FOREIGN KEY (ProdutoID) REFERENCES Produtos(ProdutoID)  
);
```






```
INSERT INTO Clientes (ClienteID, Nome) VALUES
(1, 'João'),
(2, 'Maria'),
(3, 'Carlos'),
(4, 'Ana');
```

```
INSERT INTO Pedidos (PedidoID, ClienteID, DataPedido) VALUES
(101, 1, '2023-01-10'),
(102, 2, '2023-01-15'),
(103, 3, '2023-02-05'),
(104, 2, '2023-02-10');
```






```
INSERT INTO Produtos (ProdutoID, NomeProduto, CategoriaID, Preco) VALUES
(201, 'Camiseta', 1, 29.99),
(202, 'Calça Jeans', 1, 39.99),
(203, 'Tênis', 2, 59.99),
(204, 'Boné', 2, 19.99);
```

```
INSERT INTO Categorias (CategoriaID, NomeCategoria) VALUES
(1, 'Roupas'),
(2, 'Acessórios');
```





```
INSERT INTO DetalhesPedidos (DetalheID, PedidoID,  
ProdutoID, Quantidade) VALUES  
(1, 101, 201, 2),  
(2, 101, 203, 1),  
(3, 102, 202, 1),  
(4, 103, 204, 3);
```



● Consultas

1. Listar os clientes juntamente com os pedidos realizados por eles.
2. Mostrar todos os produtos, incluindo detalhes de pedidos (se houver).
3. Exibir os pedidos feitos, incluindo clientes que fizeram pedidos.
4. Mostrar os produtos com suas respectivas categorias.
5. Exibir todos os pedidos, inclusive aqueles sem detalhes de pedidos.
6. Listar as categorias com produtos associados, incluindo categorias sem produtos.



● Consultas

7. Exibir detalhes de pedidos com produtos correspondentes e informações de pedidos associadas.
8. Listar todos os produtos, inclusive aqueles sem pedidos associados.
9. Relacionar todas categorias à produtos.

