

BANCO DE DADOS II

prof. Rodrigo Barbosa
prof2300@iesp.edu.br
(83)991330759

Até agora...

- Normalização banco de dados relacional
- Sublinguagens SQL (DML, DDL, DCL, DQL)
- Exemplos com comandos básicos SQL
- Exercícios sobre os comandos básicos

Junção de Tabelas

- É uma operação poderosa que permite combinar dados de duas ou mais tabelas em um único resultado, com base em uma relação entre elas.
- Fundamental para recuperar informações de bancos de dados relacionais complexos

Como identificar?

- Identificar as tabelas que deseja combinar.
- Tabelas geralmente têm uma relação entre elas por meio de chaves primárias e chaves estrangeiras.
- EX: "Clientes" e "Pedidos". A tabela "Clientes" pode ter uma coluna "ID" que é a chave primária, e a tabela "Pedidos" pode ter uma coluna "ClienteID" que é uma chave estrangeira que se relaciona com a tabela "Clientes"

Tipos de Junção

- INNER JOIN: Retorna apenas as linhas que têm correspondência em ambas as tabelas. Em outras palavras, retorna apenas os registros onde a condição de junção é satisfeita em ambas as tabelas.
- LEFT JOIN (ou LEFT OUTER JOIN): Retorna todas as linhas da tabela à esquerda e as linhas correspondentes da tabela à direita. Se não houver correspondência na tabela à direita, os valores NULL são retornados.
- RIGHT JOIN (ou RIGHT OUTER JOIN): Oposto ao LEFT JOIN, retorna todas as linhas da tabela à direita e as linhas correspondentes da tabela à esquerda. Se não houver correspondência na tabela à esquerda, os valores NULL são retornados.

Tipos de Junção

- FULL JOIN (ou FULL OUTER JOIN): Retorna todas as linhas de ambas as tabelas, combinando-as quando houver correspondência. Se não houver correspondência em uma das tabelas, os valores NULL são retornados.
- CROSS JOIN (ou PRODUCT JOIN): Realiza o produto cartesiano de todas as linhas de ambas as tabelas, gerando todas as combinações possíveis. Geralmente é usado quando você deseja combinar todas as linhas de uma tabela com todas as linhas de outra.

Condição da Junção

- Para definir a relação entre as tabelas, você deve especificar uma condição de junção. Essa condição geralmente envolve a igualdade entre as colunas relacionadas nas tabelas envolvidas.
- A condição de junção é especificada na cláusula ON (para INNER JOIN) ou LEFT ON (para LEFT JOIN) ou equivalente nas outras junções. Por exemplo, `Clientes.ID = Pedidos.ClienteID` estabelece a relação entre as tabelas "Clientes" e "Pedidos".

Como conseguimos o resultado dessa junção?

- Seleccionamos as colunas que deseja recuperar na consulta resultante, normalmente usando a cláusula SELECT. Pode ser qualquer combinação de colunas de ambas as tabelas envolvidas.
- Quando a consulta é executada, o sistema de gerenciamento de banco de dados (SGBD) realiza a junção das tabelas com base na condição especificada. Ele verifica as linhas de ambas as tabelas para determinar as correspondências e gera um conjunto de resultados que combina as colunas selecionadas de ambas as tabelas.

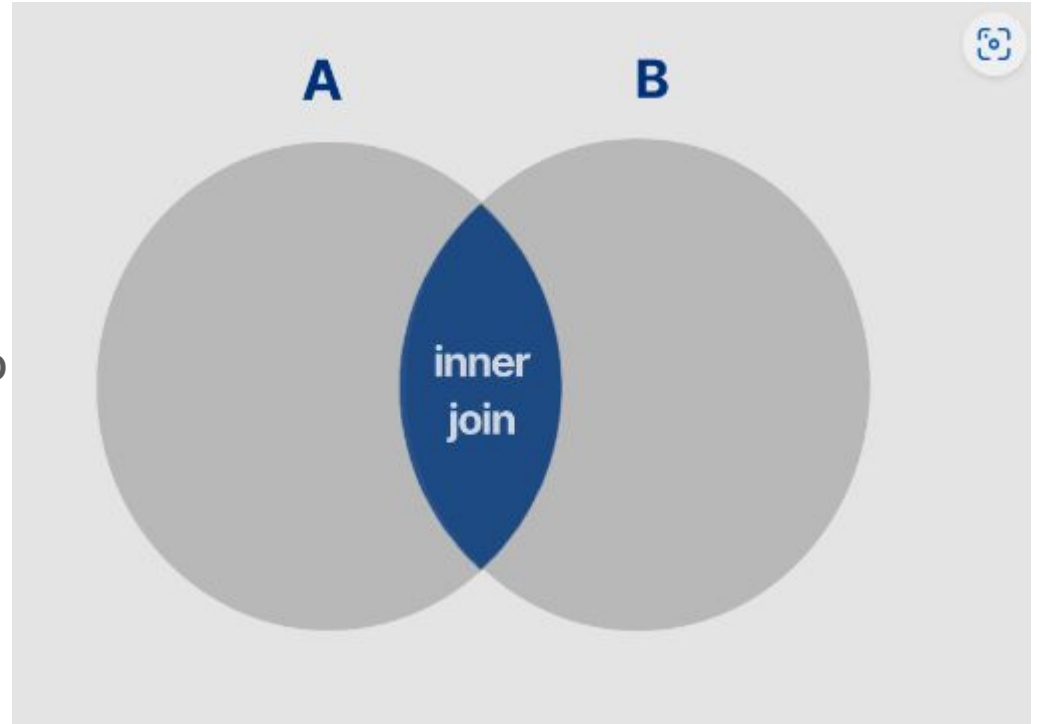
Como conseguimos o resultado dessa junção?

- O resultado da junção é uma única tabela temporária que contém todas as colunas selecionadas de ambas as tabelas. Cada linha representa uma combinação de linhas de ambas as tabelas onde a condição de junção é satisfeita.

INNER JOIN

- Essa condição é definida na cláusula ON, onde você estabelece a relação entre as colunas das tabelas. A condição de junção geralmente envolve a igualdade entre as colunas relacionadas. Por exemplo, `Tabela1.Chave = Tabela2.ChaveEstrangeira` estabelece a relação entre as tabelas "Tabela1" e "Tabela2".
- O resultado do INNER JOIN é uma única tabela temporária que contém as colunas selecionadas de ambas as tabelas, mas apenas as linhas onde há correspondência. Ou seja, ele retorna apenas as linhas que têm uma correspondência em ambas as tabelas. As linhas que não atendem à condição de junção são excluídas do resultado.

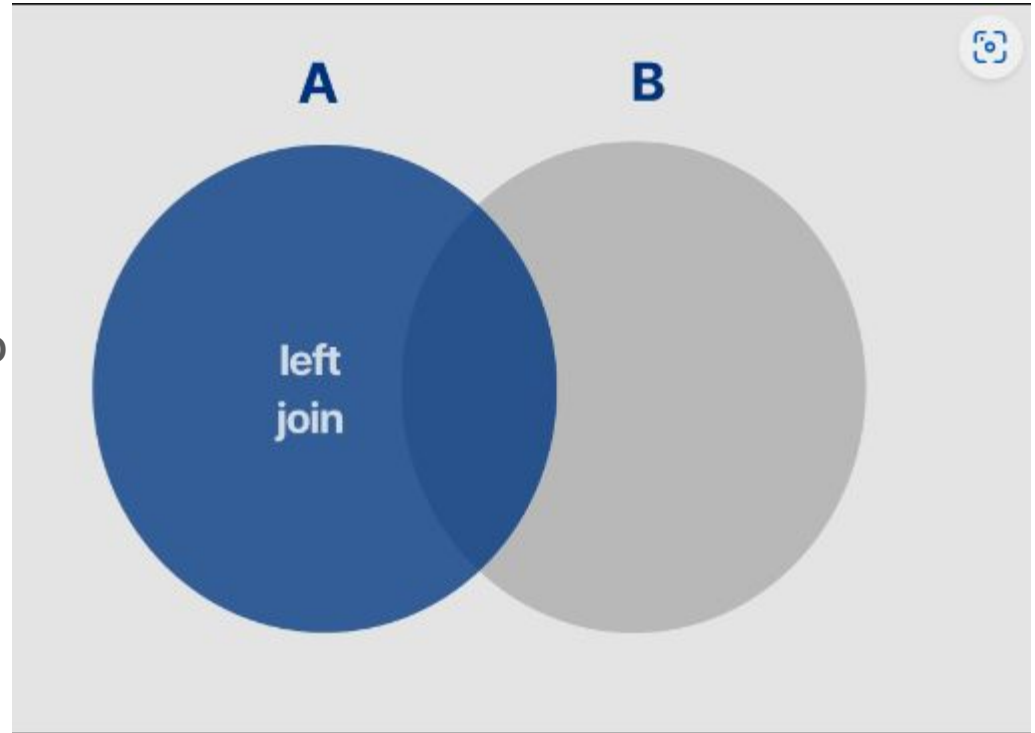
```
SELECT <CAMPOS>  
FROM TABELA_A  
INNER JOIN TABELA_B  
ON TABELA_A.CAMPO = TABELA_B.CAMPO
```



LEFT JOIN

- A condição de junção também é definida na cláusula ON, onde você estabelece a relação entre as colunas das tabelas. A condição geralmente envolve a igualdade entre as colunas relacionadas. Por exemplo, `Tabela1.Chave = Tabela2.ChaveEstrangeira` estabelece a relação entre as tabelas "Tabela1" e "Tabela2".
- O resultado do LEFT JOIN é uma única tabela temporária que contém as colunas selecionadas de ambas as tabelas. Todas as linhas da tabela à esquerda estão presentes no resultado, enquanto as linhas da tabela à direita são incluídas apenas se houver uma correspondência na condição de junção. Se não houver correspondência, os valores NULL são atribuídos às colunas da tabela à direita.

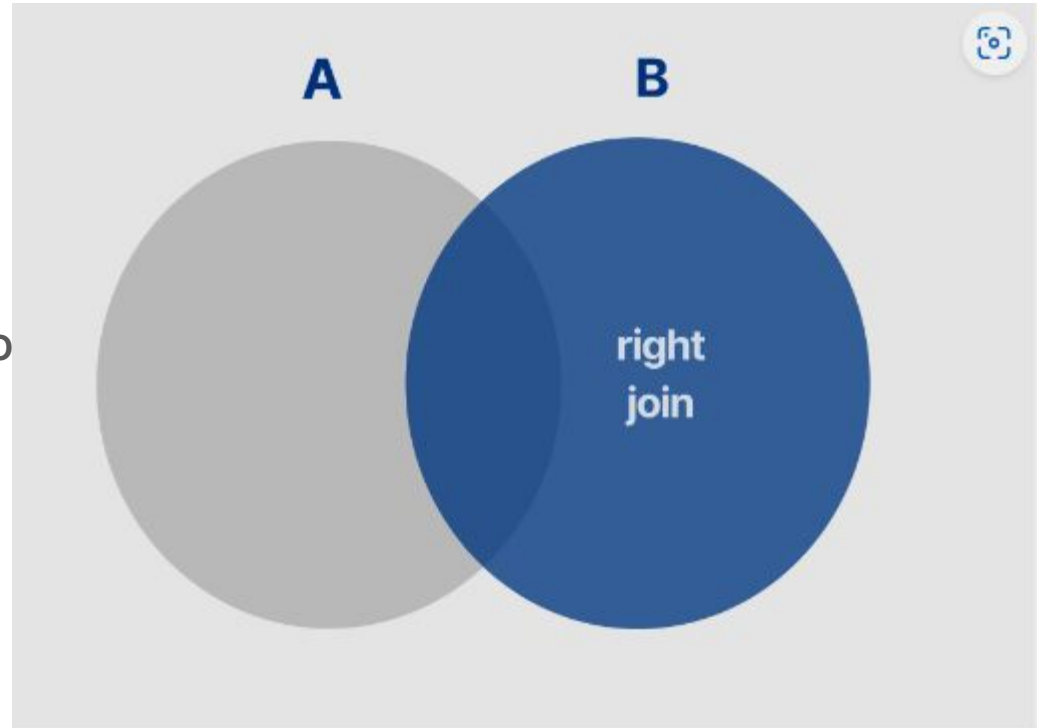
```
SELECT <CAMPOS>  
FROM TABELA_A  
LEFT JOIN TABELA_B  
ON TABELA_A.CAMPO = TABELA_B.CAMPO  
where TABELA_A.CAMPO IS not NULL  
OR TABELA_B.CAMPO IS NOT NULL
```



RIGHT JOIN

- A condição de junção é definida na cláusula ON, onde você estabelece a relação entre as colunas das tabelas. A condição geralmente envolve a igualdade entre as colunas relacionadas. Por exemplo, Tabela1.Chave = Tabela2.ChaveEstrangeira estabelece a relação entre as tabelas "Tabela1" e "Tabela2".
- O resultado do RIGHT JOIN é uma única tabela temporária que contém as colunas selecionadas de ambas as tabelas. Todas as linhas da tabela à direita estão presentes no resultado, enquanto as linhas da tabela à esquerda são incluídas apenas se houver uma correspondência na condição de junção. Se não houver correspondência, os valores NULL são atribuídos às colunas da tabela à esquerda.

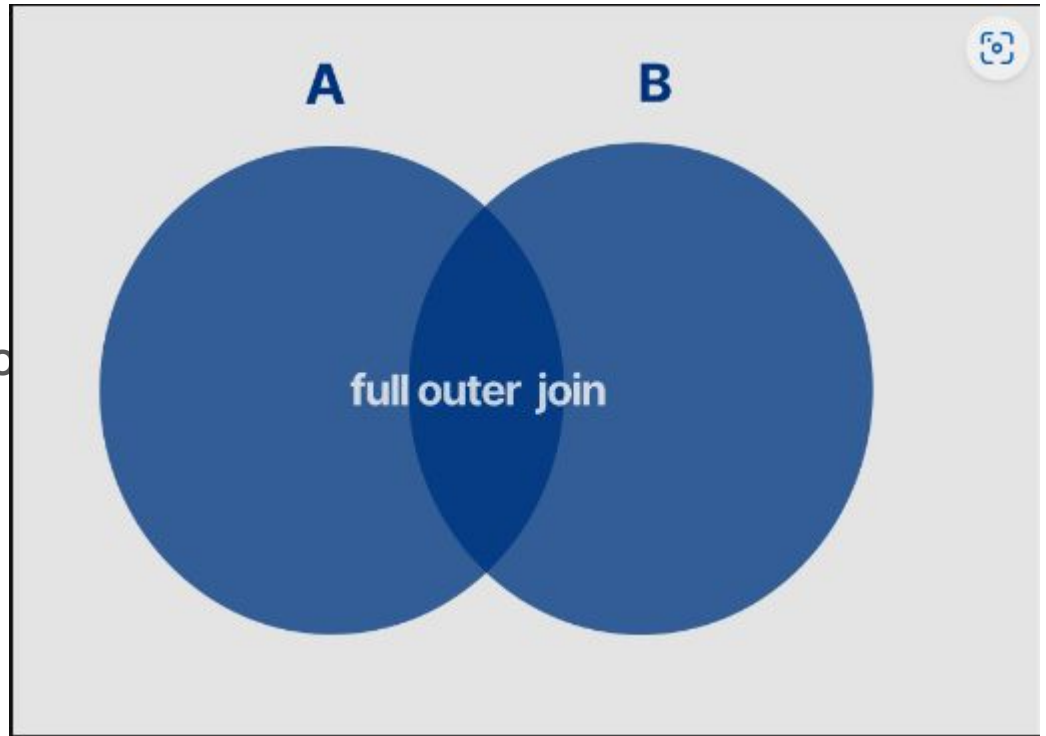
```
SELECT <CAMPOS>  
FROM TABELA_A  
RIGHT JOIN TABELA_B  
ON TABELA_A.CAMPO = TABELA_B.CAMPO
```



FULL JOIN

- A condição de junção é definida na cláusula ON, onde você estabelece a relação entre as colunas das tabelas. A condição geralmente envolve a igualdade entre as colunas relacionadas. Por exemplo, Tabela1.Chave = Tabela2.ChaveEstrangeira estabelece a relação entre as tabelas "Tabela1" e "Tabela2".
- O resultado do FULL JOIN é uma única tabela temporária que contém as colunas selecionadas de ambas as tabelas. Todas as linhas de ambas as tabelas estão presentes no resultado. As colunas das tabelas onde não há correspondência contêm valores NULL.


```
SELECT <CAMPOS>  
FROM TABELA_A  
FULL OUTER JOIN TABELA_B  
ON TABELA_A.CAMPO = TABELA_B.CAMPO
```



```
SELECT <CAMPOS>
```

```
FROM TABELA_A
```

```
LEFT JOIN TABELA_B
```

```
ON TABELA_A.CAMPO = TABELA_B.CAMPO
```

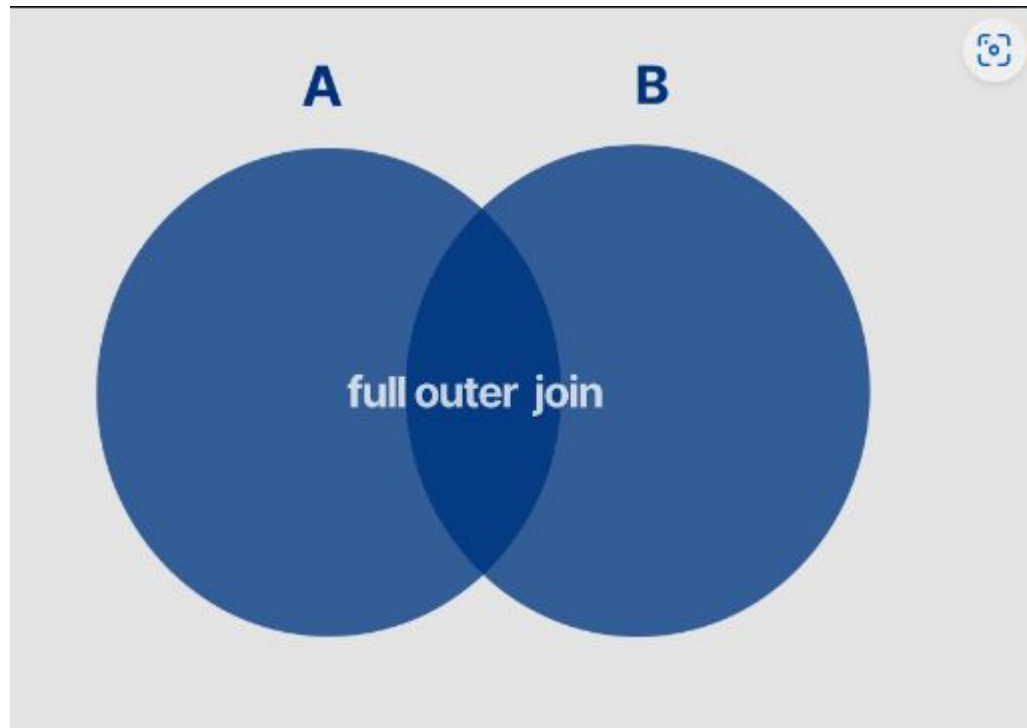
```
UNION
```

```
SELECT <CAMPOS>
```

```
FROM TABELA_A
```

```
RIGHT JOIN TABELA_B
```

```
ON TABELA_A.CAMPO = TABELA_B.CAMPO
```



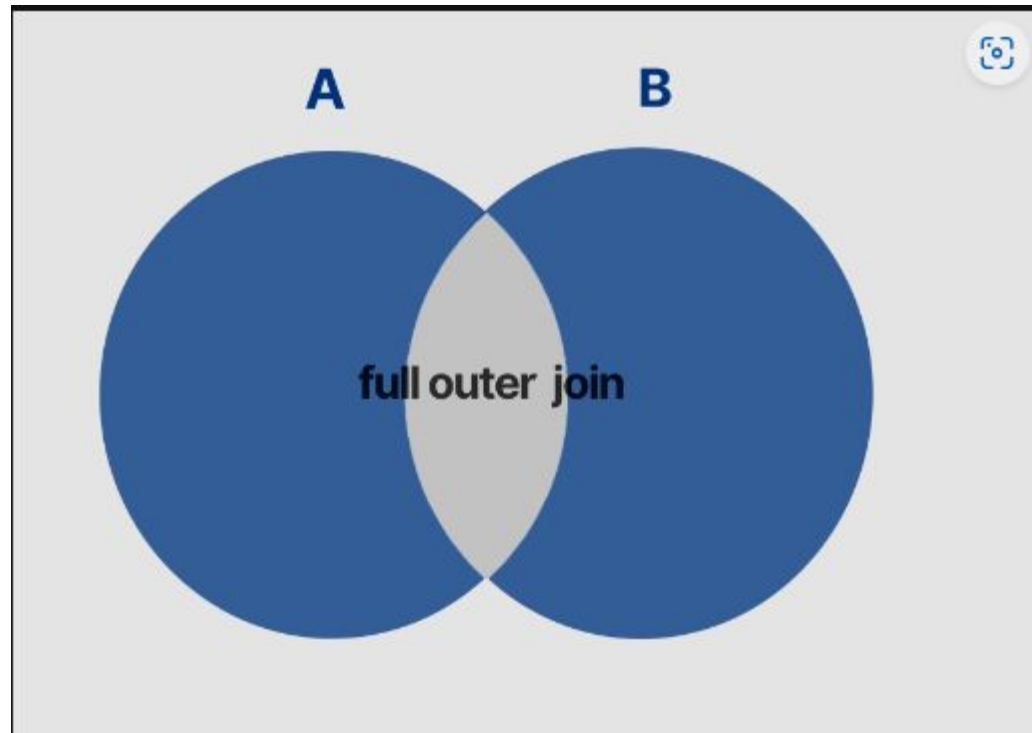
```
SELECT <CAMPOS>
```

```
FROM TABELA_A
```

```
INNER JOIN TABELA_B
```

```
ON TABELA_A.CAMPO = TABELA_B.CAMPO
```

```
WHERE TABELA_A.CAMPO IS NULL OR TABELA_B.CAMPO IS NULL
```



CROSS JOIN

- O CROSS JOIN não usa uma condição de junção, e ele é usado quando você deseja combinar cada linha de uma tabela com cada linha de outra tabela.
- Não se baseia em uma condição de correspondência. Ele combina cada linha de uma tabela com todas as linhas da outra tabela, produzindo uma tabela resultante com o produto cartesiano de todas as combinações possíveis.
- O número de linhas no resultado de um CROSS JOIN é o produto do número de linhas nas tabelas envolvidas. Se uma tabela tem m linhas e a outra tem n linhas, o resultado terá $m * n$ linhas.

CROSS JOIN

- O CROSS JOIN é usado com moderação porque pode gerar um grande número de resultados, especialmente se as tabelas envolvidas forem grandes. Normalmente, é usado em cenários específicos, como quando você deseja criar uma lista completa de todas as combinações possíveis de valores.
- Um exemplo comum de uso do CROSS JOIN é quando você deseja gerar uma tabela que combine todos os produtos com todos os clientes, criando um registro de todas as possíveis associações entre produtos e clientes.

```
SELECT <CAMPOS>  
FROM TABELA_A  
CROSS JOIN TABELA_B
```

