

UNIVERSIDADE DE BRASÍLIA
Faculdade do Gama

Sistemas de Banco de Dados 1

Pesquisa (JOIN)

Exercício Extra 2 da Aula 10

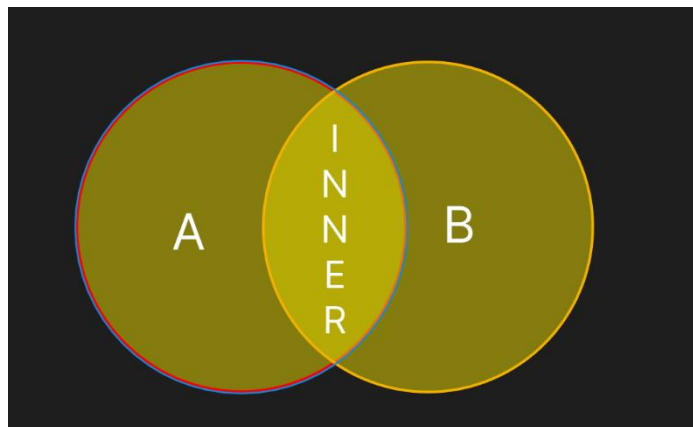
Mylena Angélica Silva Farias- 211029497

Brasília, DF
2024

A operação JOIN em SQL é usada para combinar linhas de duas ou mais tabelas com base em uma coluna relacionada entre elas. O objetivo principal é recuperar dados que estão distribuídos entre diferentes tabelas. Como toda operação possui suas vantagens e desvantagens. As suas vantagens são: Combinar dados de diversas tabelas, reduzir redundância. Já suas desvantagens são: Difícil otimização, Performance prejudicada devido a não indexação correta.

Existem diversos tipos de JOIN em SQL sendo eles:

- Inner Join
 - O que é? É uma operação de junção que retorna apenas as linhas que têm correspondências nas duas tabelas envolvidas.
 - Vantagens: Retorna apenas os dados que têm correspondências, eliminando resultados irrelevantes; Geralmente mais eficiente, pois trabalha apenas com dados que possuem correspondências.
 - Desvantagens: Linhas sem correspondências em ambas as tabelas são descartadas, o que pode resultar na perda de informações úteis.
 - Representação visual:



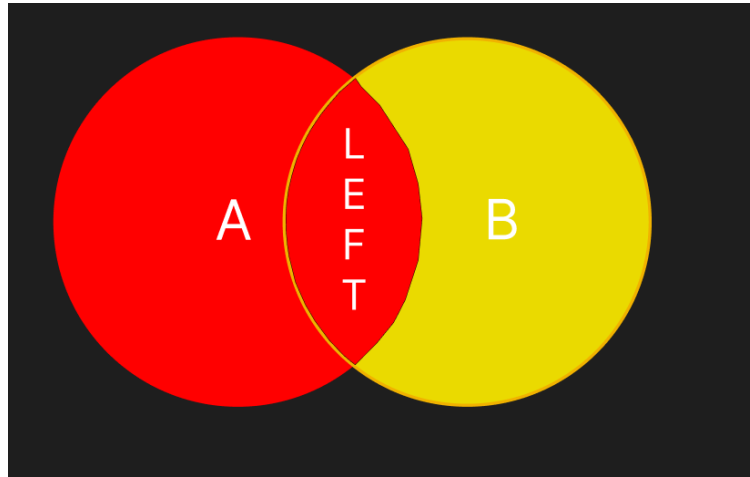
- Código em SQL:

```
SELECT  
FROM  
INNER JOIN ON ;
```

- Left Join
 - O que é? Retorna todas as linhas da tabela à esquerda e as linhas correspondentes da tabela à direita. Se não houver correspondência, os resultados conterão NULL.
 - Vantagens: Mantém todas as linhas da tabela à esquerda, mesmo sem

correspondência na tabela à direita; Útil para relatórios que precisam mostrar todos os itens de uma tabela principal.

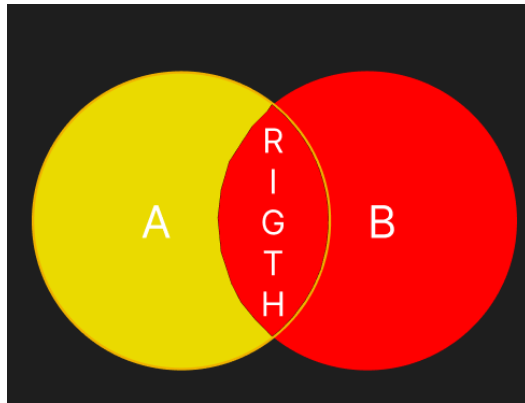
- Desvantagens: As linhas sem correspondências na tabela à direita terão valores NULL, exigindo tratamento adicional; Pode ser mais lento que Inner Join devido à necessidade de verificar todas as linhas da tabela da esquerda.
- Representação visual:



- Código em SQL:
SELECT
FROM
LEFT JOIN ON ;

- Right Join

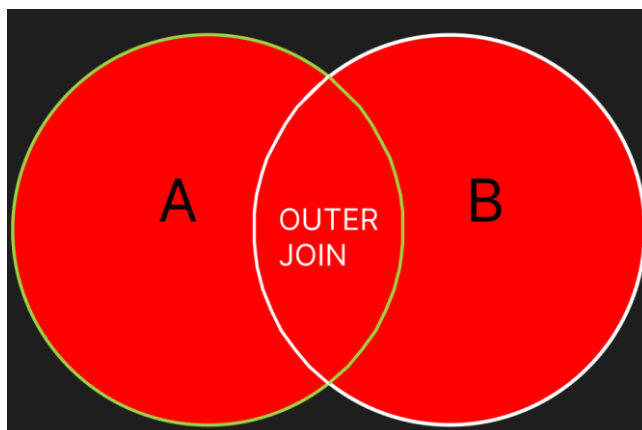
- O que é? Retorna todas as linhas da tabela à direita e as linhas correspondentes da tabela à esquerda. Se não houver correspondência, os resultados conterão NULL.
- Vantagens: Mantém todas as linhas da tabela à direita, mesmo sem correspondência na tabela à esquerda; Útil quando a tabela secundária é a principal fonte de dados.
- Desvantagens: As linhas sem correspondências na tabela à esquerda terão valores NULL, exigindo tratamento adicional; Pode ser mais lento que Inner Join devido à necessidade de verificar todas as linhas da tabela da direita.
- Representação visual:



- Código em SQL:
`SELECT`
`FROM`
`RIGHT JOIN ON ;`

- Outer Join

- O que é? retorna todas as linhas quando há uma correspondência em uma das tabelas. Se não houver correspondência, os resultados conterão NULL.
- Vantagens: Retém todas as linhas de ambas as tabelas, mostrando todas as possíveis combinações; Permite ver todas as discrepâncias e todas as correspondências entre as tabelas.
- Desvantagens: Pode ser pesado em termos de performance e resultar em grandes conjuntos de dados com muitos valores NULL; Exige um tratamento considerável de valores NULL no pós-processamento.
- Representação visual:

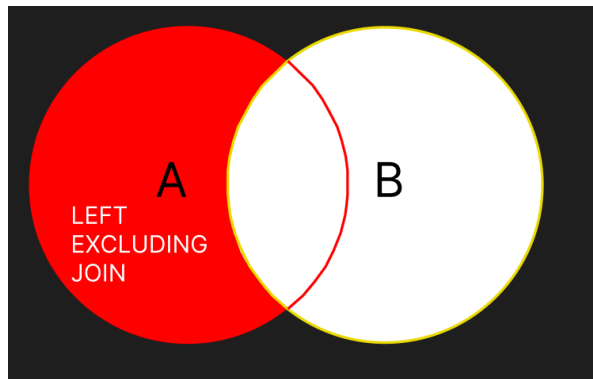


- Código em SQL:
`SELECT`
`FROM`

FULL OUTER JOIN ON ;

- Left Excluding Join

- O que é? retorna todas as linhas da tabela à esquerda que não têm correspondências na tabela à direita.
- Vantagens: Útil para encontrar dados que estão na tabela principal, mas não na secundária; Fácil de entender e implementar para identificar discrepâncias.
- Desvantagens: Só retorna dados da tabela à esquerda, excluindo informações da tabela à direita; Pode ser menos eficiente se a tabela à esquerda for muito grande.
- Representação visual

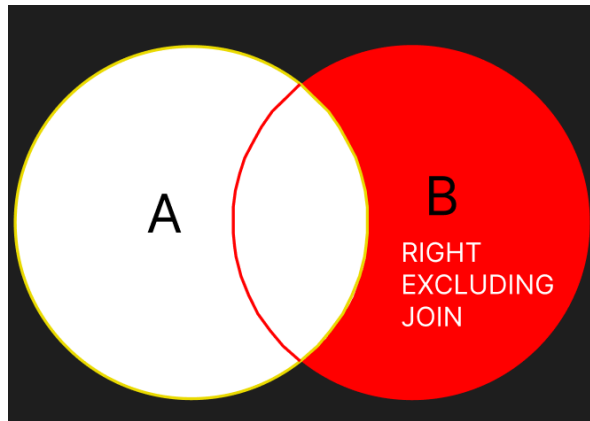


- Código em SQL:
SELECT
FROM
LEFT JOIN ON
WHERE IS NULL;

- Right Excluding Join

- O que é? retorna todas as linhas da tabela à direita que não têm correspondências na tabela à esquerda.
- Vantagens: Útil para encontrar dados que estão na tabela secundária mas não na principal; Fácil de entender e implementar para identificar discrepâncias.
- Desvantagens: Só retorna dados da tabela à direita, excluindo informações da tabela à esquerda; Pode ser menos eficiente se a tabela à direita for muito grande.

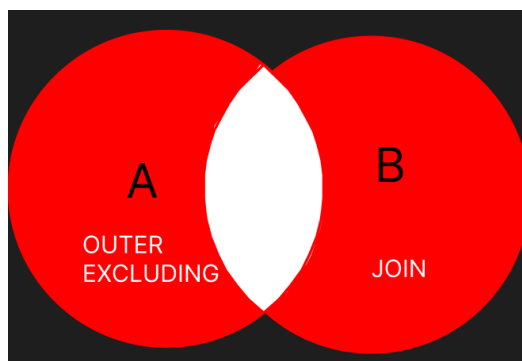
- Representação visual:



- Código em SQL:
SELECT
FROM
RIGHT JOIN ON
WHERE IS NULL;

- Outer Excluding Join

- O que é? retorna todas as linhas das tabelas à esquerda e à direita que não têm correspondências na outra tabela.
- Vantagens: Útil para encontrar discrepâncias em ambas as tabelas; Mostra todas as linhas sem correspondência, proporcionando uma visão completa das discrepâncias.
- Desvantagens: Pode ser mais complexo de entender e implementar; Pode ser menos eficiente, especialmente com grandes tabelas, devido à necessidade de verificar todas as linhas em ambas as tabelas.
- Representação visual:



- Código em SQL:
SELECT
FROM

```
FULL OUTER JOIN ON  
WHERE IS NULL OR IS NULL;
```