UNIVERSIDADE DE BRASÍLIA
Faculdade do Gama

Sistemas de Banco de Dados 1

Pesquisa JOINs no SQL

Lucas Felipe Soares - 202016767

Brasília, DF
2023

Ao esturdarmos maneiras de se fazer consultas mais eficientes em SQL, somos apresentados ao conceito de Join em SQL. JOINs no SQL são instruções que permitem a junção de dados de duas ou mais tabelas com as devidas condições entre as mesmas, durante uma consulta no banco de dados. Com isso, as JOINs se tornam essenciais em projetos que necessitem de manipulação da base de dados pois, com ele, é possível realizar pesquisas mais completas e principalmente mais eficientes.

Dentre as principais vantagens do JOIN, podemos destacar o poder de combinar os dados de duas ou mais tabelas, visto que isso nos ajuda a obter resultados mais relevantes para aquilo que se deseja extrair de uma pesquisa de uma determinada base de dados. Além mais, há uma grande flexibilidade na definição dos JOINs, o que nos permite manipular os dados de diversas formas, sem contar que os SGBDs são otimizados para realizar consultas JOIN, quando escrito de uma maneira eficiente.

Por outro lado, ao trabalharmos de maneira geral as desvantagens de utilização do JOIN. Justamente por sua grande capacidade de executat consultas, em projetos mais complexos , as consultas com o JOIN muito provavelmente tendem a se tornar cada vez mais complexas também, o que pode o tornar de difícil compreensão e, consequentemente, dificultando a escalabilidade do projeto. Mais ainda, quando ocorrem de não serem bem projetados, os JOINs podem acabar não sendo eficiente nas consultas, por conta da falta de índices adequados ou até mesmo a recuperação de dados indesejados. Com isso, temos exatamente o efeito contrário em relação a eficiência que podemos encontrar com sua utilização.

Há diversos tipos de JOIN, sendo eles: INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN e o CROSS JOIN. A seguir veremos alguns exemplos encontrados em diversos materiais sobre cada um dos mesmos:

```
SELECT <select_list>
FROM Tabela A
INNER JOIN Tabela B
ON A.Key = B.Key
```

A cláusula **INNER JOIN** compara cada linha da tabela A com as linhas da tabela B para encontrar todos os pares de linhas que satisfazem a condição de junção. Se a condição de junção for avaliado como `TRUE`, os valores da coluna das linhas correspondentes das tabelas A e B serão combinados em uma nova linha e incluídos no conjunto de resultados.

LEFT JOIN

Retorna todos os registros da tabela esquerda e os registros correspondentes da tabela direita.

```
SELECT <select_list>
FROM Tabela A
LEFT JOIN Tabela B
ON A.Key = B.Key
```

Para cada linha da tabela A, a consulta a compara com todas as linhas da tabela B. Se um par de linhas fizer com que a condição de junção seja avaliado como `TRUE`, os valores da coluna dessas linhas serão combinados para formar uma nova linha que será incluída no conjunto de resultados.

Se uma linha da tabela **"esquerda" A** não tiver nenhuma linha correspondente da tabela **"direita" B**, a consulta irá combinar os valores da coluna da linha da tabela **"esquerda" A** com `NULL` para cada valor da coluna da tabela da **"direita" B** que não satisfaça a condição de junto (`FALSE`).

RIGHT JOIN

Retorna todos os registros da tabela direita e os registros correspondentes da tabela esquerda.

```
SELECT <select_list>
FROM Tabela A
RIGHT JOIN Tabela B
ON A.Key = B.Key
```

A `RIGHT JOIN` combina dados de duas ou mais tabelas. A `RIGHT JOIN` começa a selecionar dados da tabela **"direita" B** e a corresponder às linhas da tabela **"esquerda" A**.

A `RIGHT JOIN` retorna um conjunto de resultados que inclui todas as linhas da tabela **"direita" B**, com ou sem linhas correspondentes na tabela **"esquerda" A**. Se uma linha na tabela **direita B** não tiver nenhuma linha correspondente da tabela **"esquerda" A**, a coluna da tabela **"esquerda" A** no conjunto de resultados será nula igualmente ao que acontece no LEFT JOIN.

FULL JOIN

Retorna todos os registros quando houver uma correspondência na tabela esquerda ou direita.

```
SELECT <select_list>
FROM Tabela A
FULL JOIN Tabela B
ON A.Key = B.Key
```

A cláusula `FULL JOIN` retorna todas as linhas das tabelas unidas, correspondidas ou não, ou seja, você pode dizer que a `FULL JOIN` combina as funções da `LEFT JOIN` e da `RIGHT JOIN`. `FULL JOIN` é um tipo de junção externa, por isso também é chamada junção externa completa.

Quando não existem linhas correspondentes para a linha da tabela esquerda, as colunas da tabela direita serão nulas. Da mesma forma, quando não existem linhas correspondentes para a linha da tabela direita, a coluna da tabela esquerda será nula.

Para trazer um segundo exemplo mais prático, utilizaremos as duas tabelas abaixo para entender melhor como funcionaria um inner Join.

Aluno

#	matricula	nome	semestre	dtNascimento
1	12345678	Lucas	2020.1	12/05/1971
2	78563412	Felipe	2020.2	06/03/2006
3	87654321	Roberto	2021.1	22/10/1998

Notas

#	idNota	nota	matricula
1	1	9	12345678
2	2	5	78563412
3	3	8	87654321

INNER JOIN: Retorna apenas os registros que possuem correspondência nas duas tabelas envolvidas na junção.

SELECT Aluno.nome, Notas.nota

FROM Aluno
INNER JOIN Notas **ON** Aluno.matricula = Notas.matricula;

#	nome	nota
1	Lucas	9
2	Felipe	5