**Introduction to INNER JOIN vs. WHERE Clause in SQL**

SQL queries can use either the INNER JOIN clause or the WHERE clause to combine data from multiple tables. Understanding when to use each method is crucial for optimizing query performance. This report compares the performance of INNER JOIN and WHERE clause approaches using SQL Server's SET STATISTICS IO ON and SET STATISTICS TIME ON commands to measure I/O and execution time.

**Query Optimization Using INNER JOIN**

Objective is the optimization of SQL queries using INNER JOIN and WHERE clause approaches in the AdventureWorksLT database.

**Initial Query with WHERE Clause**

The initial query retrieves Name, Color, and ListPrice from the SalesLT.Product table, filtering by ProductCategoryID using the WHERE clause to join with SalesLT.ProductCategory.

SET STATISTICS IO ON;

SET STATISTICS TIME ON;

SELECT p.Name, Color, ListPrice

FROM SalesLT.Product p, SalesLT.ProductCategory pc

WHERE p.ProductCategoryID = pc.ProductCategoryID;

**Analysis of the Query**

The query uses the WHERE clause for joining, which is a traditional approach but can be less clear and less optimized compared to INNER JOIN.

**Optimized Query with INNER JOIN**

The optimized query uses INNER JOIN to join SalesLT.Product and SalesLT.ProductCategory on ProductCategoryID.

SET STATISTICS IO ON;

SET STATISTICS TIME ON;

SELECT p.Name, Color, ListPrice

FROM SalesLT.Product p

INNER JOIN SalesLT.ProductCategory pc

ON p.ProductCategoryID = pc.ProductCategoryID;

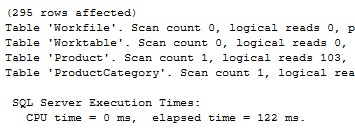
**Explanation of the Optimized Query**

1. **INNER JOIN**: INNER JOIN explicitly states how tables are related (ON p.ProductCategoryID = pc.ProductCategoryID), making it clearer and potentially more optimized by the query optimizer.
2. **WHERE Clause**: Using the WHERE clause for joining is an older approach and can be less optimized, especially in more complex queries.

**Comparison of Query Performance**

To compare the performance, we will execute both queries with SET STATISTICS IO ON and SET STATISTICS TIME ON to measure I/O and execution time.

**Initial Query Execution with WHERE Clause**



**Optimized Query Execution with INNER JOIN**

A white text with black text

Description automatically generated

**Benefits of Using INNER JOIN**

* **Clarity**: INNER JOIN provides clearer syntax, explicitly stating how tables are joined.
* **Optimization**: INNER JOIN can be more optimized by the query optimizer, potentially improving query performance.
* **Modern Approach**: INNER JOIN is a more modern and preferred approach for SQL queries.

**Conclusion**

Using INNER JOIN instead of the WHERE clause for joining tables can lead to improved query performance and clarity. This report demonstrates the importance of using efficient SQL syntax and understanding when to use different join methods to optimize database queries.