# Implementing Query Optimization and Performance Comparison

Query optimization is an essential technique for enhancing database performance by reducing unnecessary operations and improving query execution plans. This section demonstrates the impact of query optimization by comparing execution times for a sample query before and after optimization.

## Query Execution Without Optimization

The query retrieves orders for customers located in the USA, filtered by a specific date range. Initially, the query was executed without optimization. The following metrics were recorded:

- \*\*SQL Server Execution Times:\*\*  
 - CPU time: 8 ms  
 - Elapsed time: 10 ms  
- \*\*Total Execution Time:\*\* 00:00:00.291

The following code was used to execute the query without optimization:

SET STATISTICS IO ON; -- Enables I/O statistics  
SET STATISTICS TIME ON; -- Enables time measurements  
SELECT \*  
FROM Orders o  
JOIN Customers c ON o.CustomerID = c.CustomerID  
WHERE c.Country = 'USA' AND o.OrderDate >= '2023-01-01';  
SET STATISTICS IO OFF;  
SET STATISTICS TIME OFF;

## Query Execution With Optimization

After optimizing the query to select only required columns and applying proper filters, the query was executed again. The following metrics were recorded:

- \*\*SQL Server Execution Times:\*\*  
 - CPU time: 6 ms  
 - Elapsed time: 5 ms  
- \*\*Total Execution Time:\*\* 00:00:00.101

The following code was used to execute the query after optimization:

SET STATISTICS IO ON;  
SET STATISTICS TIME ON;  
SELECT o.OrderID, c.Name, c.Email, o.OrderAmount, o.OrderDate  
FROM Customers c  
INNER JOIN Orders o ON c.CustomerID = o.CustomerID  
WHERE c.Country = 'USA' AND o.OrderDate >= '2023-01-01';  
SET STATISTICS IO OFF;  
SET STATISTICS TIME OFF;

## Performance Comparison

The table below compares the query execution times before and after optimization:

|  |  |  |
| --- | --- | --- |
| Metric | Without Optimization | With Optimization |
| SQL Server Execution Times (CPU) | 8 ms | 6 ms |
| SQL Server Execution Times (Elapsed) | 10 ms | 5 ms |
| Total Execution Time | 00:00:00.291 | 00:00:00.101 |

From the above results, it is evident that query optimization significantly reduces execution time and improves efficiency. By restructuring the query to include only relevant columns and applying appropriate filters, database resources are utilized more effectively, enhancing overall performance.