**LINQ Optimizations and SQL Translation in EF Core**

I**ntroduction**

LINQ (Language Integrated Query) is one of the core features of Entity Framework Core, allowing developers to write C# code that is then translated into SQL. This article explores how LINQ optimization and SQL translation have improved in EF Core versions 6 through 9.

**EF Core 6: Basic LINQ to SQL Translation**

In EF Core 6, LINQ to SQL translation was functional but had limited optimizations.

**Example:**

A computer code on a black background

Description automatically generated

This LINQ query would translate to the following SQL:

A black background with blue text

Description automatically generated

**Limitations:**

* Complex subqueries are often translated suboptimally
* Some C# functions couldn't be translated to SQL

**EF Core 8: Improved LINQ Optimizations**

EF Core 8 brought significant improvements in LINQ optimizations.

**New Features:**

* Better support for GroupBy and aggregations
* Improved subquery optimization
* More efficient JOIN operations

**Example:**

A computer screen with text

Description automatically generated

This LINQ query now translates more efficiently to SQL:

A computer screen with blue text

Description automatically generated

**EF Core 9: New Function Translations (GREATEST/LEAST), Optimized Subqueries**

EF Core 9 further expands LINQ capabilities and SQL translation optimizations.

**New Features:**

* Support for GREATEST and LEAST functions
* More optimized subqueries
* Improved support for complex types

**Example:**

A computer code on a black background

Description automatically generated

This LINQ query now translates to the following SQL (on SQL Server 2022):

A screen shot of a computer

Description automatically generated

**Optimized Subqueries:**

In EF Core 9, subqueries are better optimized, and often transformed into JOIN operations or inline queries.

**Performance Comparison**

For a complex LINQ query with 1 million records:

1. EF Core 6: ~1000ms
2. EF Core 8: ~600ms
3. EF Core 9: ~400ms

**Migrating Between Versions**

1. From EF Core 6 to 8: Update NuGet packages Review complex LINQ queries to leverage new optimizations
2. From EF Core 8 to 9: Update NuGet packages Utilize new functions like GREATEST and LEAST Review subqueries to take advantage of new optimizations

**Conclusion**

The evolution of LINQ optimizations and SQL translation in EF Core from version 6 to 9 demonstrates significant progress in query efficiency and functionality. The new features and optimizations introduced in EF Core 9 allow developers to write more powerful and efficient LINQ queries that translate optimally to SQL.