

Group 13 Project Progress Report

Members: Daniel Lindsey, Chase Keller, Long Nguyen

Submission Date: October 13th

Introduction

PostgreSQL features a cost-based query optimizer that transforms SQL queries into more efficient permutations to minimize execution time. This optimizer utilizes a variety of efficiency improving techniques such as index utilization, query flattening, and query unnesting/decorrelation among others.

This project covers the topic of SQL query unnesting, and we seek to provide clear real-world backed evidence on the actual efficiency improvements unnesting provides on average compared to their nested counterparts. We also hope to identify edge cases which are impossible to unnest using current methods and attempt to provide a methodology on how to unnest such queries.

Problem(s) Addressed

Little real-world evidence is available detailing the actual efficiency improvements query unnesting/decorrelation provides.

Current unnesting algorithms only address specific permutations. While the current rules are mostly comprehensive, there still exist edge cases that are not supported.

Project Goal

At the end of our project, we want to have clear graphical data showing the differences between query durations when executing unnested and nested queries. From this, we can confirm the purported efficiency claims referenced in the featured research articles.

We also want to identify a list of exceptions/edge cases of queries which are unable to be unnested in their current states. From this list, we will then provide methodology on how to unnest such queries by making use of substitution and interchangeability. Ultimately, we hope to further reduce execution times.

Current Status & Results

We have compiled multiple large real-world datasets for use in query duration testing. Including daily climate data from the National Centers for Environmental Information (NCEI) covering Ohio over the past 5 years, as well as criminal data from Los Angeles, CA over the past 5 years. We are currently looking into ways to disable PostgreSQL's native optimization protocols so we can properly compare correlated and decorrelated queries to get accurate results. As it stands currently, any queries that are executed are being optimized, resulting in skewed results.

We have identified the below operations as being edge cases/exceptions to the unnesting algorithms. Due to their order dependent nature, decorrelation will either be challenging or impossible as imprecise reordering will result in non-equivalent query plans.

Group 13 Project Progress Report

Edge Case Operations:

1. Views
2. Authorization Methods
3. Set Returning Functions (SRFs)

Evaluation Plan

Data to be used: Ohio Climate Data (2020-2025), LA Crime Data (2020-2025)

Phase 1:

- **Goal of Experiment:** Run correlated and decorrelated versions of queries on the above datasets to graph the difference between execution times. From this, we can confirm from our small sample size that decorrelation is more efficient.
- **Evaluation Metrics:** Query execution time.
- **Outcome:** Graphical data depicting execution duration trends.

Phase 2:

- **Goal of Experiment:** Develop methodology to algorithmically decorrelate edge case operations. If this proves impossible, we will instead succinctly show why each case is unable to be decorrelated.
- **Evaluation Metrics:** Edge case decorrelation feasibility and query execution time (if feasible).
- **Outcome:** Annotated decorrelation algorithms for each edge case. Or a series of counter examples of why each case is unable to be decorrelated.

Remaining Schedule

Date	Milestone Event
October 13 th	Progress Report Due
October 17 th	Setup PostgreSQL testing environment
October 20 th	Create unnested and nested test queries
October 24 th	Execute test queries and record results
November 1 st	Develop decorrelation algorithms for views, authorization methods, and SRFs
November 8 th	Test execution times for decorrelated queries and compare results
November 10 th	Start Presentation
November 17 th	Presentation Due / Start Final Report
December 7 th	Complete Final Report

Member Contribution

Each member provided equal contributions to the creation of this project report.