INFO-H417: Database Systems Architecture Project
Range Types: Statistics, Selectivity and Join Estimations

Mahmoud SAKR, Maxime SCHOEMANS

2021-2022

1 Introduction

Range types in postgres are data types for representing a continuous interval of values. Interestingly range types can be instantiated with a subtype, then they express intervals of this subtype. For instance, ranges of timestamp is a time interval, e.g. the lecture slot. Clearly the subtystrictly left ofpe must have a total order so that it is well-defined whether element values are within, before, or after a range of values.

Postgres comes with a set of built in range types, i.e., already instantiated with subtypes. In addition, one can use CREATE TYPE to define more. The built-in types include int4range, int8range, tsrange (Range of timestamp without time zone), etc. More information about the range types can be found here.

The query optimizer collects statistics of range columns in the form of equidepth histograms: (1) range lower bounds, (2) range upper bounds, (3) range lengths, see range_typanalyze. These statistics are used in selectivity estimation of range predicates such as overlaps, less, greater, etc, see rangetypes_selfuncs.c

In contract, join cardinality estimation is not implemented, e.g., see for instance the function areajoinsel which is used for the estimation of the range overlap join.

The goal of this project is to improve the overall scheme of statistics collection and cardinality estimation for range types. That is, you are required to propose and implement: (1) range typanalyze, (2) selectivity estimation functions for the *overlaps*(&&) and the *strictly left of* (<<) predicates, (3) join cardinality estimation for the *overlaps* (&&) predicate.

2 Deliverables

The project is to be completed in **groups of 4 members**, and the deliverables are listed below.

1. You should deliver a report (as a .pdf) containing the following elements:
    - A cover page with the list of group members, including student ID,

- The theory and the *interesting code snippets* of your statistics collection model, the selectivity, and the join estimation functions.
- The design, implementation, and result of comprehensive benchmark evaluating your implementation, and comparing with the built-in implementation in Postgres

2. A presentation and a demo of your solution.

The deadline for the report is Monday 13 Dec, and the presentations will be organized between Wednesday 15 Dec and Friday 17 Dec.

3 Evaluation

The evaluation jury consists of the two course instructors. You will be asked to present your solution and defend it (item 2 in deliverables). The grading will consider both the theory and the implementation of your solution, for all four of the main tasks.

This project will count for 40% of your course mark.

	Theory	Implementation
Range typanalyze	5 pts	5 pts
Selectivity estimation function (*overlaps* and *less than*)	5 pts	5 pts
Join cardinality estimation function (*overlaps*)	5 pts	5 pts
Benchmark	5 pts	5 pts

For the first three parts, the ideas and the theoretical model used to answer the problem will be grades separately from the practical implementation of the solution. The benchmark will similarly be split in two part: 1. (Theory) Which tests are used to benchmark the performance of the implementation, and why? 2. (Implementation) A description of the results returned by the benchmark and a reflection on these results.