Database Management System

Rule Based Query Optimization

OUR TEAM

Sourabh Soumyakanta Das

Shiladitya De

Ashwani Kumar Kamal

Archit Mangrulkar

Hardik Soni

PROJECT SCOPE AND OBJECTIVES

- Specification of fixed rules for query optimization.
- Design and implementation of a query rewriter tool.
- Testing the query rewriter on a suitable testset.

METHODOLOGY

01 Specifying syntax for relational algebra query

02 Developing Heuristic based rules for query optimization

03 Specifying the Grammar for the input Query

04 Parsing the Input Query using Flex

05 Bottom-Up Parsing using bison

06 Optimizing the Parse Tree using our Rules

SPECIFYING SYNTAX FOR RELATIONAL ALGEBRA QUERY

The first task is to define a proper syntax to give relational algebra query input by the user.

The syntax finalized by us is as follows:

- σ : SELECT
- $\prod$: PROJECT
- $\bowtie$: JOIN
- $\times$: PROD
- $\bigcup$: UNION
- $-$: DIFF
- $\bigcap$: INTERSECT
- $\wedge$: AND
- $\vee$: OR

Actual Query: $\sigma_{(a \wedge b)}(T)$
Query according to the rules given above: `SELECT [a AND b] (T)`

Actual Query: $\prod_{(a)}(\prod_{(b)}(\sigma_{(c \wedge d)}(T))$
Query according to the rules given above: `PROJECT [ a ] (PROJECT [ b ] (SELECT [ c AND d ] (T)))` etc.

RULES FOR QUERY OPTIMIZATION

- $\sigma_{(a \wedge b)}(T) = \sigma_{(a)}(\sigma_{(b)}(T))$
- $\prod_{(a_1)}(\prod_{(a_2)}(\ldots(\prod_{(a_n)}(T)\ldots)) = \prod_{(a_1)}(T)$
- $\sigma_{\theta}(T_1 \times T_2) = T_1 \bowtie_{\theta} T_2$
- $\sigma_{\theta_1}(T_1 \bowtie_{\theta_2} T_2) = T_1 \bowtie_{(\theta_1 \wedge \theta_2)} T_2$
- $\sigma_{\theta}(T_1 - T_2) = \sigma_{\theta}(T_1) - \sigma_{\theta}(T_2)$
- $\prod_{\theta}(T_1 \cup T_2) = \prod_{\theta}(T_1) \cup \prod_{\theta}(T_2)$

We are considering for now only those rules which are heuristic based and not based on the actual table statistics (cost based ones). If time permits then we will try to incorporate those as well.


CODE

RESULTS

Query: PROJECT [B1] (PROJECT [A2] (SELECT [A1 AND B5] (T)))
Result: PROJECT [B1] (SELECT [A1] (SELECT [B5] (T)))

Query: SELECT [A3] (T1 DIFF T2)
Result: SELECT [A3] (T1) DIFF SELECT [A3] (T2)

Query: PROJECT [A5] (SELECT [A3] (T1 DIFF T2) UNION T2)
Result: PROJECT [A5] (SELECT [A3] (T1) DIFF SELECT [A3] (T2)) UNION PROJECT [A5] (T2)

Query: SELECT [A3] (SELECT [A1 AND B5] (T) DIFF T2)
Result: SELECT [A3] (SELECT [A1] (SELECT [B5] (T))) DIFF SELECT [A3] (T2)

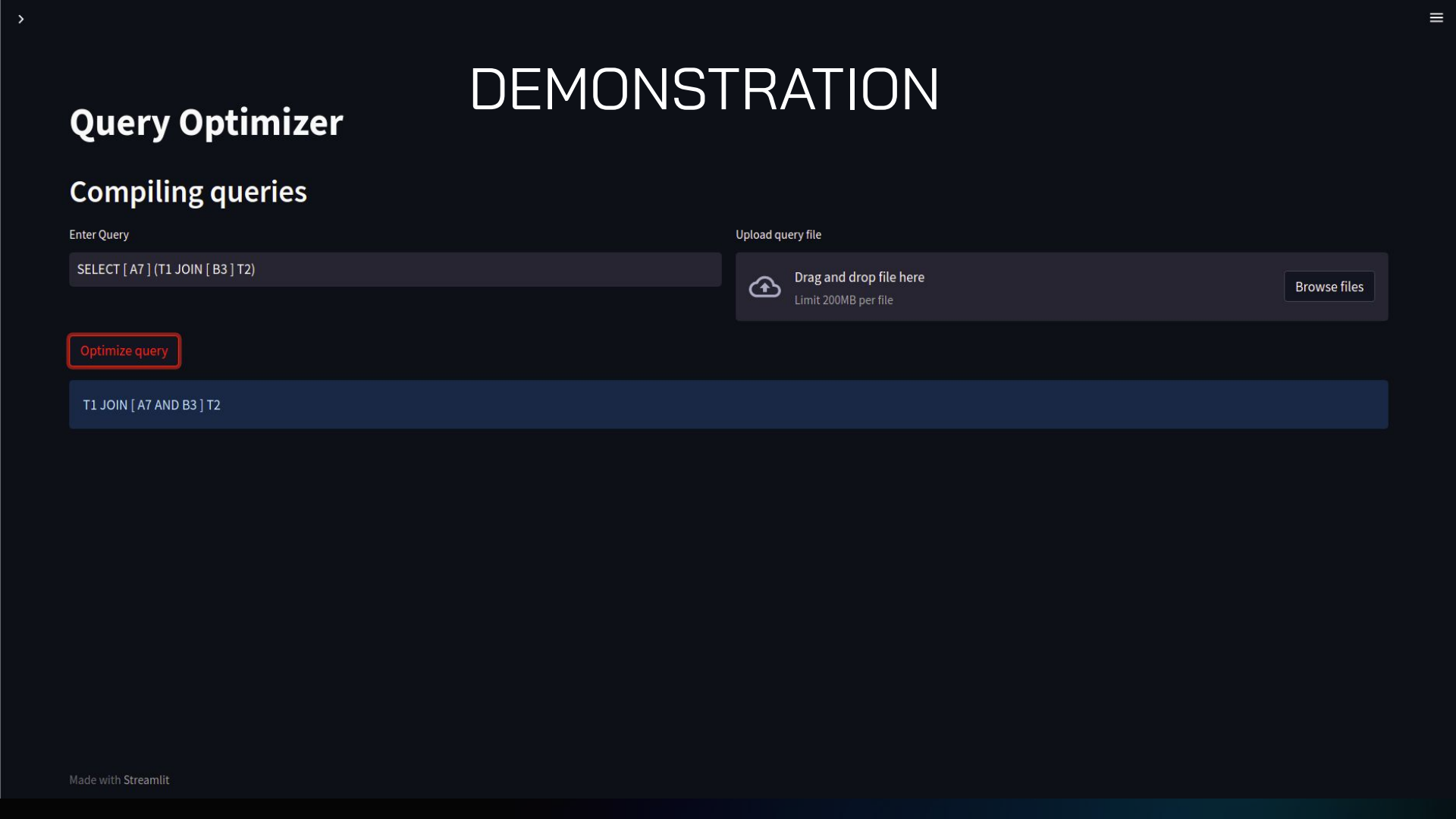
REFERENCES

1. Chaudhuri, S. (1998). An overview of query optimization in relational systems. ACM SIGMOD Record, 27(2), 34-49.

2. Graefe, G. (1995). Volcano-an extensible and parallel query evaluation system. IEEE Transactions on Knowledge and Data Engineering, 7(4), 563-575.

3. Selinger, P. G., Astrahan, M. M., Chamberlin, D. D., Lorie, R. A., & Price, T. G. (1979). Access path selection in a relational database management system. In Proceedings

DEMONSTRATION



THANK YOU