In this project, you will develop a program that processes a single-block SQL query by applying heuristic optimizations. Your task is to optimize the initial query with heuristic techniques to improve execution efficiency.

A diagram of a diagram

Description automatically generated

Input Requirements

SQL Query Input: The primary input for the program will be a single-block SQL query composed of standard SQL syntax and constructs such as SELECT, FROM, WHERE, GROUP BY, HAVING, and ORDER BY.

Additional Optional Inputs: To enhance the optimization process, you may include additional optional inputs for adjusting the optimization strategy based on more detailed database characteristics or operational constraints. Some recommendations are listed below:

Selectivity Estimates: Selectivity factors for selection predicates and join conditions.

Data Distribution Statistics: Histograms or other statistical information describing the distribution of distinctive attribute data values.

Index Availability: Specify which indexes are available on the tables involved in the query.

Output Requirements

Canonical Query Tree

Output the initial canonical query tree generated from the input SQL query. This tree should represent the logical flow and structure of the query before any optimizations are applied. Its outputting format can be, but not limited to, graphical or a structured textual representation.

Optimized Query Tree

After applying heuristic optimizations, output the optimized query tree reflecting the changes and enhancements made to improve the query's performance.

Refined SQL Query

Convert the optimized query tree back into SQL format. This should be a runnable SQL Query that represents the optimized version of the original input.

Deliverables

Source Code: Well-commented source code in the language of your choice.

Code Description: Detailed documentation describing the flow, logic, and methodology of the code.