Logo, company name

Description automatically generated

**Group: Shrey Chaudhary (2803175)**

**Omi Patel (2835823)**

**Subject: CIS 611**

**Supervised by Prof. - SS Chung**

**Project Status Report**

**Building a Simple Query Optimizer with Performance Evaluation Experiment on Query Rewrite Optimization**

**Goals:**

* Create a basic optimizer to analyze and suggest optimized versions of SQL queries..
* Conduct systematic tests to measure the impact of query rewrite on query performance
* Investigate and implement various optimization strategies within the query optimizer.
* Measure performance metrics, compare against baseline, document implementation, and draw conclusions for future improvements.

**Programming language**

JAVA, JAVASCRIPT

**Digital Platform:-**

**Software:-** Visual Studio Code

**Operating System:-** MacOS( Big Sur(version 11.6) )

**Memory:-** 8 GB 2133 MHz

**Processor: -** 3.1 GHz Dual-Core Intel Core i5

**Query Q1:-**

SELECT T1.x1, SUM(T2.x2)

FROM T1,T2 WHERE T2.x2 = ( SELECT SUM(T3.x3)

FROM T3 WHERE T1.x1 = T3.x3) GROUP BY T1.x1;

Here above query retrieves the values of the column x1 from table T1 and the sum of column x2 from table T2 where the condition T2.x2 is equal to the result of a subquery. The subquery calculates the sum of column x3 from table T3 where the values of T1.x1 match those in T3.x3. The result is then grouped by the values of T1.x1. Essentially, it combines data from multiple tables, comparing and aggregating values based on specified conditions, and presents the result in a tabular format with one row for each distinct value of T1.x1, along with the corresponding sum of T2.x2.

**Rebuilding it to RQ1:-**

SELECT T1.x1, SUM(T2.x2) FROM T1,T2, (Select T1.Rowid, SUM(T3.x3)

From T1, T3 Where T1.x1 = T3.x3 Group By T1.Rowid) as Temp1 (rowid, x3) WHERE T2.x2 **=** Temp.x3 and T1.Rowid = Temp.rowid GROUP BY T1.x1;

Above query selects the values of T1.x1 and the sum of T2.x2 from tables T1 and T2, joining with a temporary table (Temp1) that calculates the sum of T3.x3 grouped by T1.Rowid. The result is filtered by conditions linking T2.x2 to Temp1.x3 and T1.Rowid to Temp1.rowid, then grouped by T1.x1.

**A possible Query Execution Steps of Q1:**

1. Join t1 t2
2. Join temp1 t3
3. Project temp2
4. GroupBy temp3

**A possible Query Execution Steps of RQ1:**

1. Join t1 t3
2. GroupBy temp0
3. Join t1 Temp1
4. Join t2 Temp2
5. Project temp3
6. GroupBy temp3

**Given Join Methods:**

- Tuple Nested Loop Join: TNL

- Page Nested Loop Join: PNL

- Block Nested Loop Join: with Buffer memory B =50: BNJM

- Sort Merge Join with buffer memory B=50: SMJM

- Hash Join with Buffer memory requirement B=50 pages for hash table: HJM

- Hash Join with less Buffer memory B= 30 pages : HJL

- Block Nested Loop Join: with less Buffer memory B =30: BNJL

- Sort Merge Join with less buffer memory B=30: SMJL

**We have to calculate join cost for each join methods**

Cost of TNL = LPages + (#Tuples\_Per\_Page\_In\_L \* LPages) \* RPages  
Cost of PNL = LPages + ( LPages \* RPages)  
Cost of BNL = LPages + (LPages / BlockSize) \* RPages

SMJL Join cost = 2 (M + N) (MlogM )+ M + N

If we have Buffer size B2 > Size of bigger Table SMJM or SMJL Join cost = 3 \* (M + N)

HJM cost: 3(M + N)

**Given input:-**

bufferMemoryWithBuff = 50;

bufferMemoryWithlessBuff = 30;

Page Size=4096;

Block Size=100;

Table Size:-

t1s = 20;

t1p = 1000;

t2s = 40;

t2p = 500;

t3s = 100;

t3p = 2000;

Q1t = "Join t1 t2\r\n" +"Join temp1 t3\r\n" +"Project temp2\r\n" +"GroupBy temp3";

RQ1t = "Join t1 t3\r\n" +"GroupBy temp0\r\n" +"Join t1 Temp1\r\n" +"Join t2 Temp2\r\n" +"Project temp3\r\n" + "GroupBy temp3";