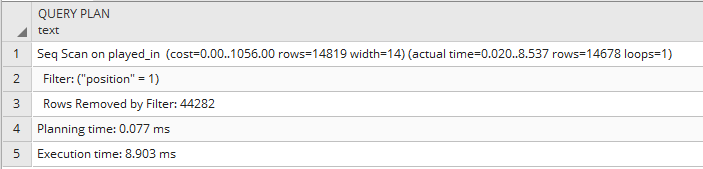
# **Part A**

## Q1.

*EXPLAIN ANALYZE SELECT \* from played\_in WHERE position=1;*

Execution Plan: Sequential Scan



## Q2.

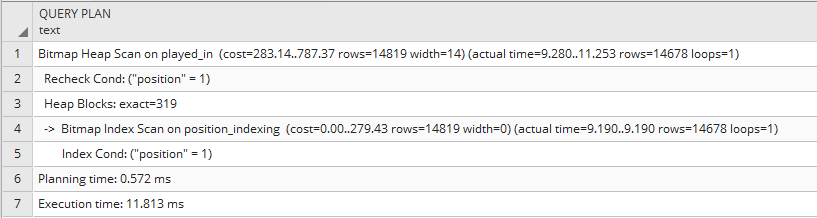
Estimated Cost for this query with the above execution plan was 0.00 + 1056.00.

## Q3.

*CREATE INDEX* name\_indexing *ON* played\_in (name);

*EXPLAIN ANALYZE SELECT \* from played\_in WHERE position=1;*

Execution Plan: Bitmap Heap Scan



## Q4.

Estimated Cost for this query with the above execution plan was 283.14 + 787.37.

## Q5.

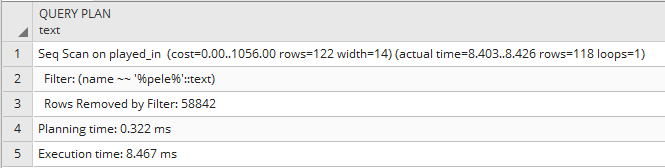
Yes, the cost changed from 0.00 + 1056.00 to 283.14 + 787.37 because the bitmap index scan was used. Instead of brute-force iterations over the data.

# **Part B**

## Q1.

*EXPLAIN ANALYZE SELECT \* from played\_in WHERE name like '%pele%';*

Execution Plan: Sequential Scan



## Q2.

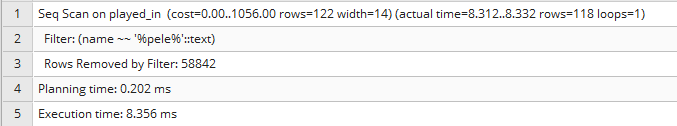
Estimated Cost for this query with the above execution plan was 0.00 + 1056.00.

## Q3.

*CREATE INDEX* name\_indexing *ON* played\_in (name);

*EXPLAIN ANALYZE SELECT \* from played\_in WHERE name like '%pele%';*

Execution Plan: Sequential Scan



## Q4.

Estimated Cost for this query with the above execution plan was 0.00 + 1056.00.

## Q5.

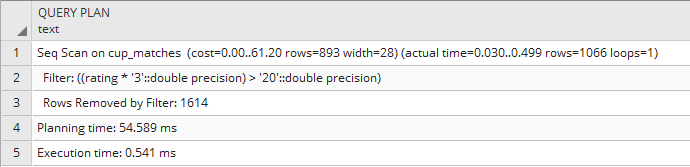
No, the cost did not change as no indexing was needed for this query because unification over strings was performed with all possible prefixes and suffixes, therefore no indexing was needed and brute-force iteration over the data is used.

# **Part C**

## Q1.

*EXPLAIN ANALYZE SELECT \* from cup\_matches WHERE rating\*3 > 20;*

Execution Plan: Sequential Scan



## Q2.

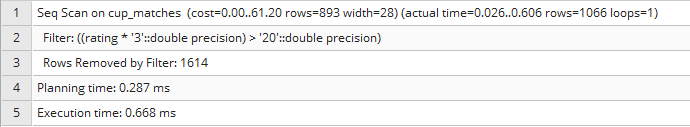
Estimated Cost for this query with the above execution plan was 0.00 + 61.20.

## Q3.

*CREATE INDEX rating\_indexing ON cup\_matches (rating);*

*EXPLAIN ANALYZE SELECT \* from cup\_matches WHERE rating\*3 > 20;*

Execution Plan: Sequential Scan



## Q4.

Estimated Cost for this query with the above execution plan was 0.00 + 61.20.

## Q5.

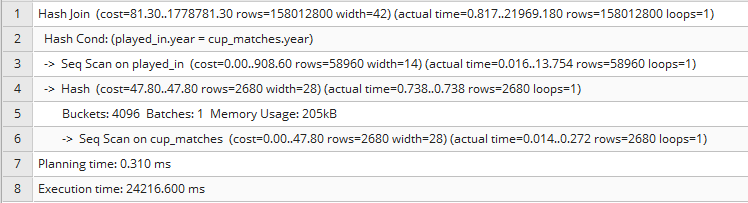
No, the cost did not change as no indexing was needed for this query because index is only available for direct comparisons, not post-evaluation comparisons.

# **Part D**

## Q1.

*EXPLAIN ANALYZE SELECT \* from cup\_matches, played\_in WHERE cup\_matches.year=played\_in.year;*

Execution Plan: Sequential Scan



## Q2.

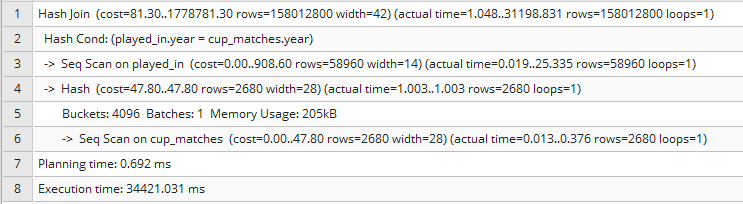
Estimated Cost for this query with the above execution plan was 81.30 + 1778781.30.

## Q3.

*CREATE INDEX year\_indexing ON cup\_matches (year);*

*EXPLAIN ANALYZE SELECT \* from cup\_matches, played\_in WHERE cup\_matches.year=played\_in.year;*

Execution Plan: Sequential Scan



## Q4.

Estimated Cost for this query with the above execution plan was 81.30 + 1778781.30.

## Q5.

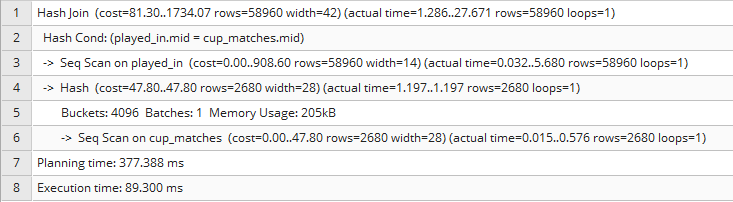
# **Part E**

## Q1.

*EXPLAIN ANALYZE SELECT \* from cup\_matches, played\_in WHERE cup\_matches.mid=played\_in.mid;*

Execution Plan: Sequential Scan

Join Algorithm: Hash Join



Estimated Cost for this query with the above execution plan was 81.30 + 1734.07.

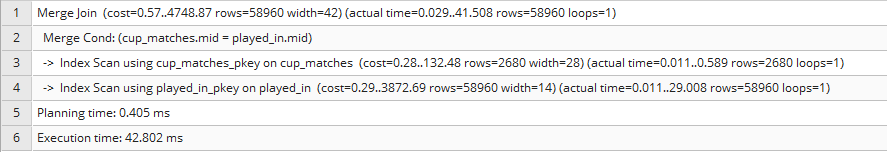
## Q2.

*SET enable\_hashjoin=false;*

*EXPLAIN ANALYZE SELECT \* from cup\_matches, played\_in WHERE cup\_matches.mid=played\_in.mid;*

Execution Plan: Index Scan

Join Algorithm: Merge Join



Estimated Cost for this query with the above execution plan was 0.57 + 4748.87.

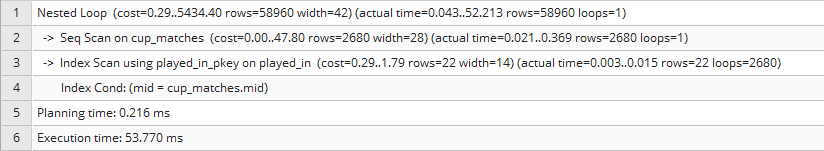
## Q3.

*SET enable\_mergejoin=false;*

*EXPLAIN ANALYZE SELECT \* from cup\_matches, played\_in WHERE cup\_matches.mid=played\_in.mid;*

Execution Plan: Sequential Scan

Join Algorithm: Nested Loop



Estimated Cost for this query with the above execution plan was 81.30 + 1778781.30.