ADBMS

Name: Meet Patel SAP-ID: 60004200104

Division/Batch: B/B1 Branch: Computer Engineering

Experiment 3

AIM: To simulate query optimization by performing SQL query on database.

ADBMS

Meet Patel 60004200104
B - 1

Experiment - 3

Aim: To simulate query optimisation by performing SQL query on database.

Theory: Query optimization is of great importance for performance of a relational database, especially for execution of complexed SQL statements. There are 2 ways:-
i) Heuristic Based
ii) Cost Based.

Heuristic Based :

A query tree is data structure that correspond to a relational algebra expression. The same query could be correspond to many different relational expressions and hence many different query trees. The task of heuristic tree that is efficient to execute. The main heuristic is to apply first operations that reduce size of intermediate results.

Cost Based :

Estimate and compare costs of executing query using different execution strategies and chose strategy with lowest cost estimate The cost of any query includes access cost to secondary storage, storage cost, memory usage cost No of memory buffer at time of execution, communication etc.

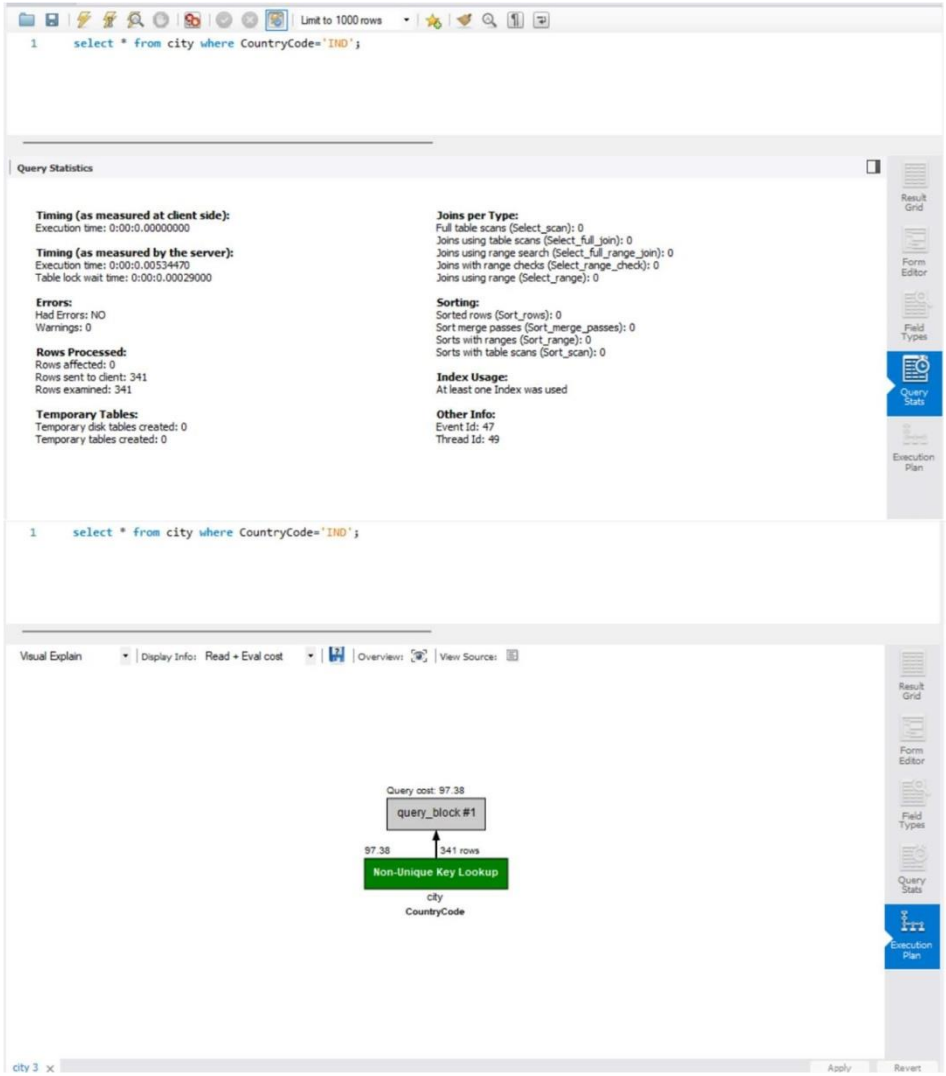
Conclusion :

Thus we performed table level and index level optimisation and compared it to results of no optimisation.

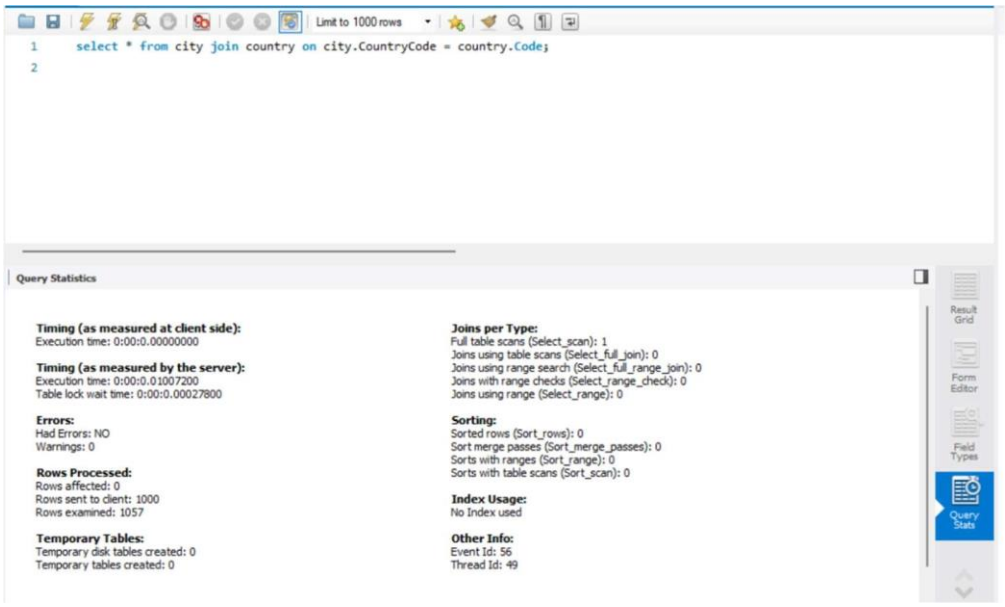
EXECUTION

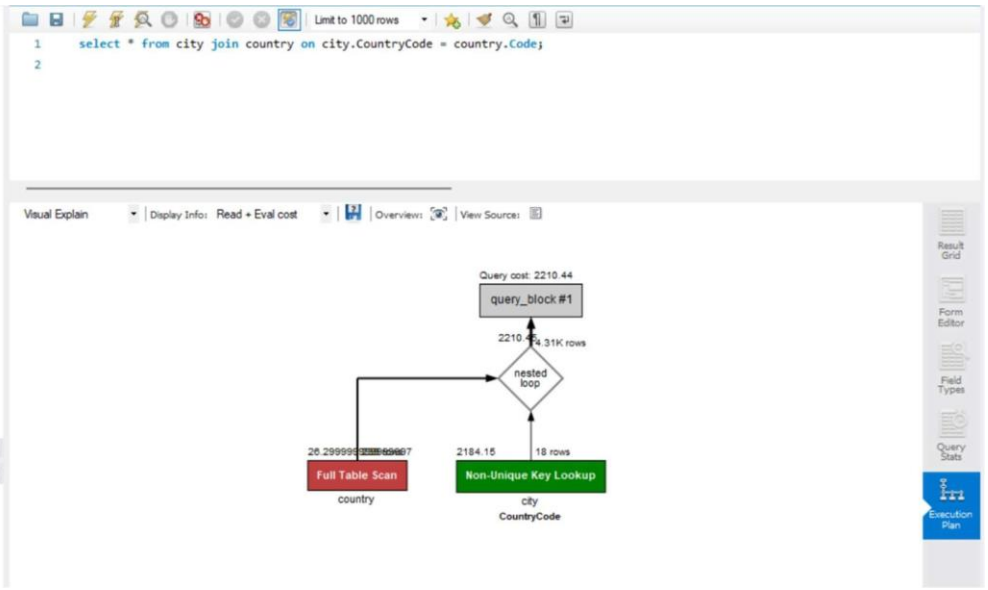
1) Unoptimized queries –

1) SELECT WHERE

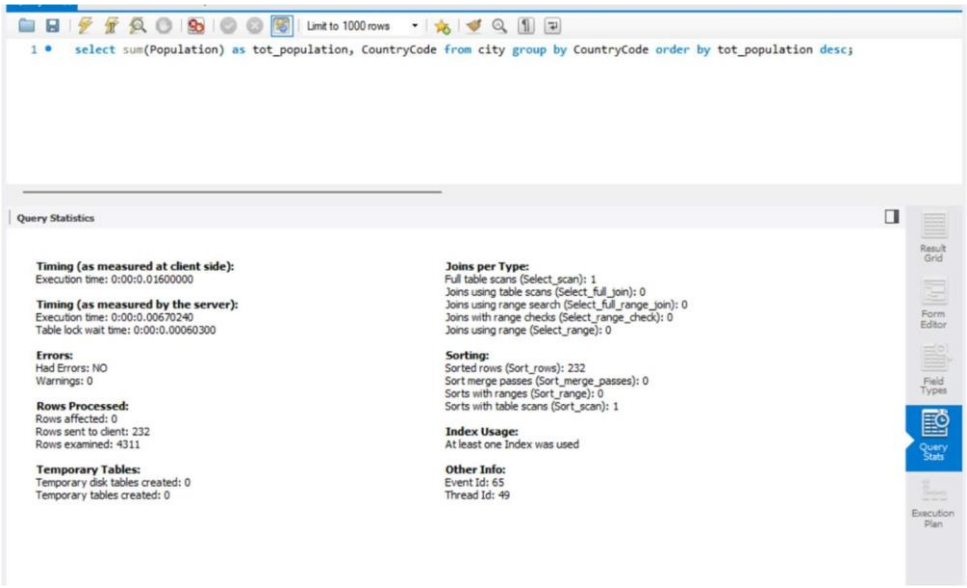


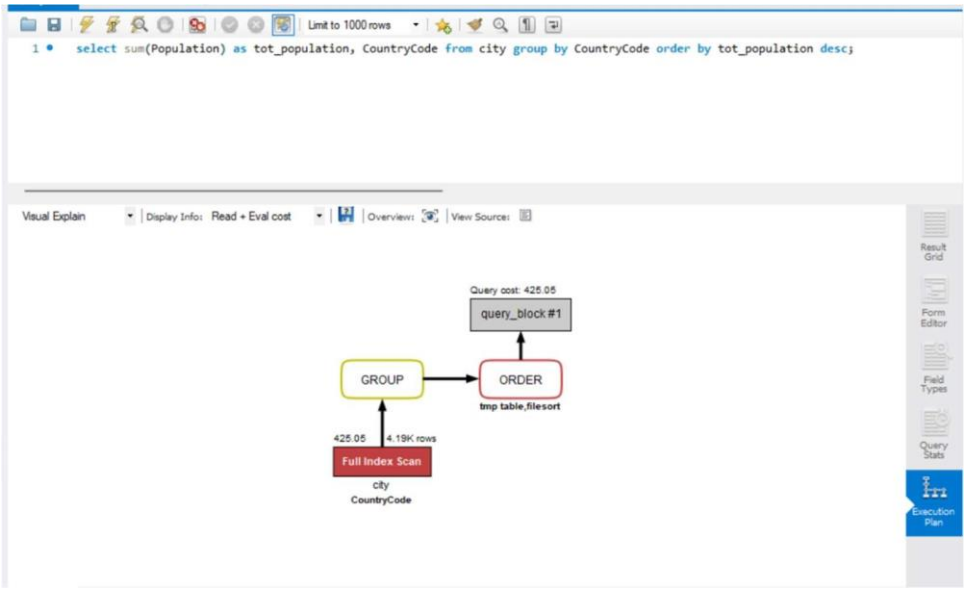
2) JOIN


Limit to 1000 rows
1 select * from city join country on city.CountryCode = country.Code;
2
Query Statistics
Timing (as measured at client side):
Execution time: 0:00:0.00000000
Timing (as measured by the server):
Execution time: 0:00:0.01007200
Table lock wait time: 0:00:0.00027800
Errors:
Had Errors: NO
Warnings: 0
Rows Processed:
Rows affected: 0
Rows sent to client: 1000
Rows examined: 1057
Temporary Tables:
Temporary disk tables created: 0
Temporary tables created: 0
Joins per Type:
Full table scans (Select_scan): 1
Joins using table scans (Select_full_join): 0
Joins using range search (Select_full_range_join): 0
Joins with range checks (Select_range_check): 0
Joins using range (Select_range): 0
Sorting:
Sorted rows (Sort_rows): 0
Sort merge passes (Sort_merge_passes): 0
Sorts with ranges (Sort_range): 0
Sorts with table scans (Sort_scan): 0
Index Usage:
No Index used
Other Info:
Event Id: 56
Thread Id: 49
Result Grid
Form Editor
Field Types
Query Stats


Limit to 1000 rows
1 select * from city join country on city.CountryCode = country.Code;
2
Visual Explain
Display Info: Read + Eval cost
Overview
View Source
Query cost: 2210.44
query_block #1
2210.44
4.31K rows
nested loop
2184.15
18 rows
Full Table Scan
country
Non-Unique Key Lookup
city
CountryCode
Result Grid
Form Editor
Field Types
Query Stats
Execution Plan

3) AGGREGATE

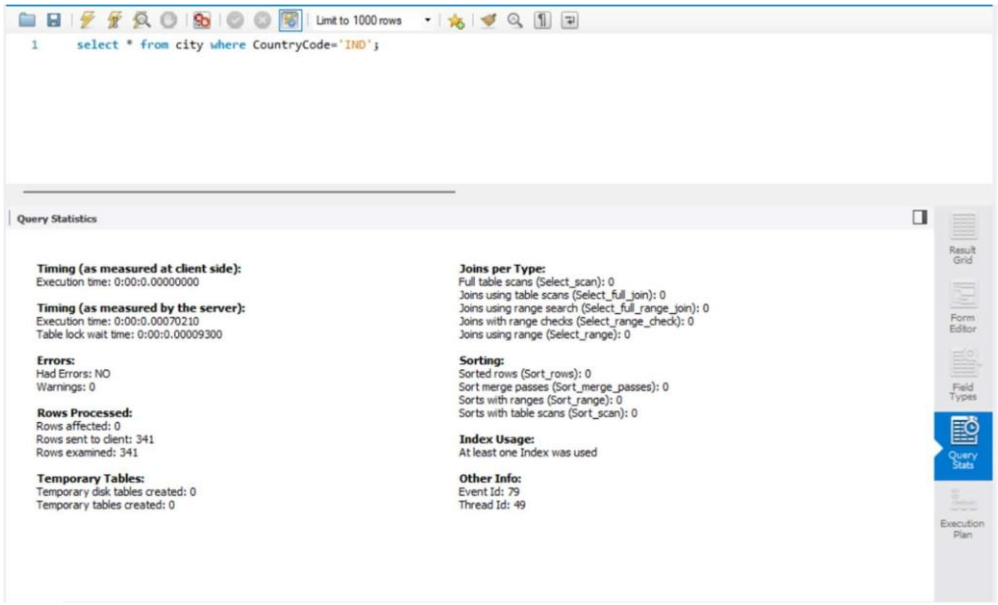

Limit to 1000 rows
1 • select sum(Population) as tot_population, CountryCode from city group by CountryCode order by tot_population desc;
Query Statistics
Timing (as measured at client side):
Execution time: 0:00:0.01600000
Timing (as measured by the server):
Execution time: 0:00:0.00670240
Table lock wait time: 0:00:0.00060300
Errors:
Had Errors: NO
Warnings: 0
Rows Processed:
Rows affected: 0
Rows sent to client: 232
Rows examined: 4311
Temporary Tables:
Temporary disk tables created: 0
Temporary tables created: 0
Joins per Type:
Full table scans (Select_scan): 1
Joins using table scans (Select_full_join): 0
Joins using range search (Select_full_range_join): 0
Joins with range checks (Select_range_check): 0
Joins using range (Select_range): 0
Sorting:
Sorted rows (Sort_rows): 232
Sort merge passes (Sort_merge_passes): 0
Sorts with ranges (Sort_range): 0
Sorts with table scans (Sort_scan): 1
Index Usage:
At least one Index was used
Other Info:
Event Id: 65
Thread Id: 49
Result Grid
Form Editor
Field Types
Query Stats
Execution Plan

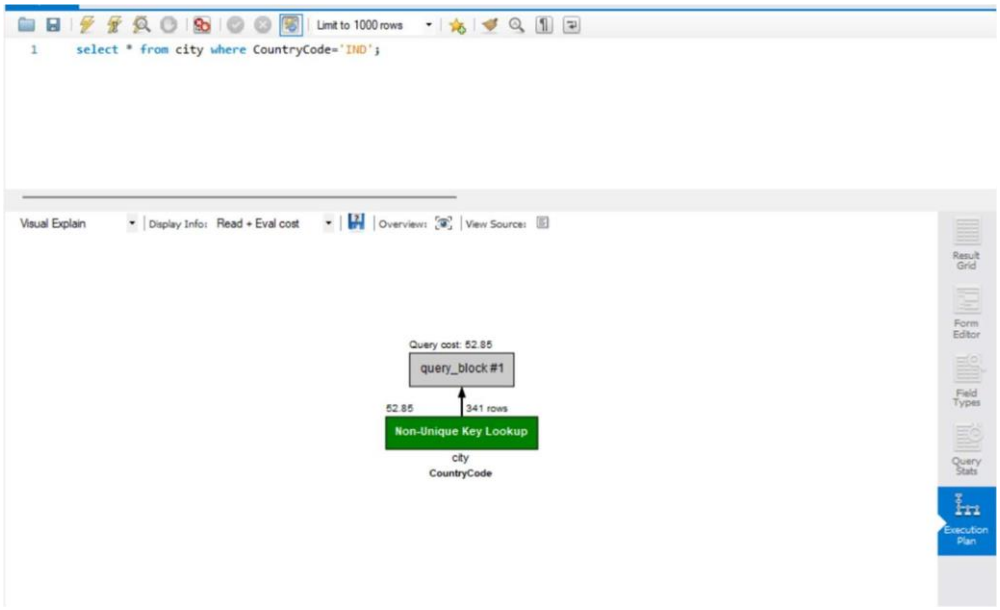

Limit to 1000 rows
1 • select sum(Population) as tot_population, CountryCode from city group by CountryCode order by tot_population desc;
Visual Explain
Display Info: Read + Eval cost
Overview:
View Source:
Query cost: 425.05
query_block #1
GROUP
ORDER
tmp table,filesort
425.05
4.19K rows
Full Index Scan
city
CountryCode
Result Grid
Form Editor
Field Types
Query Stats
Execution Plan

2) After Indexing –

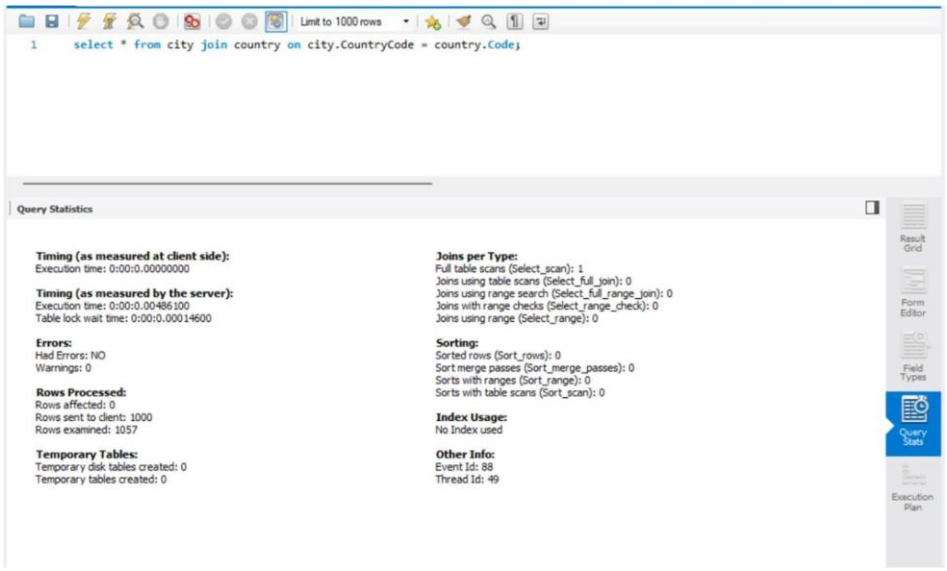
create index idx_countrycode on city (CountryCode);

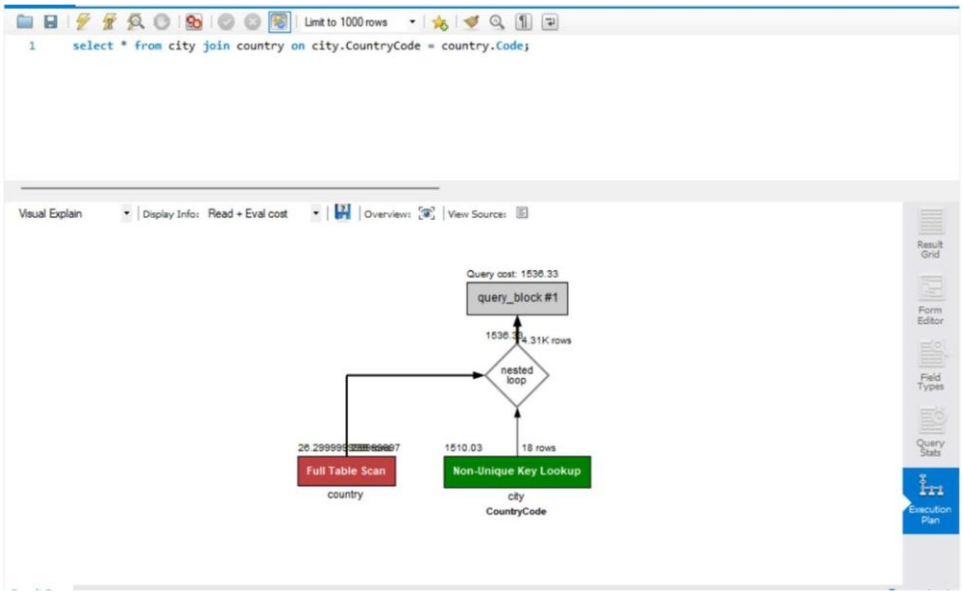
1) SELECT WHERE



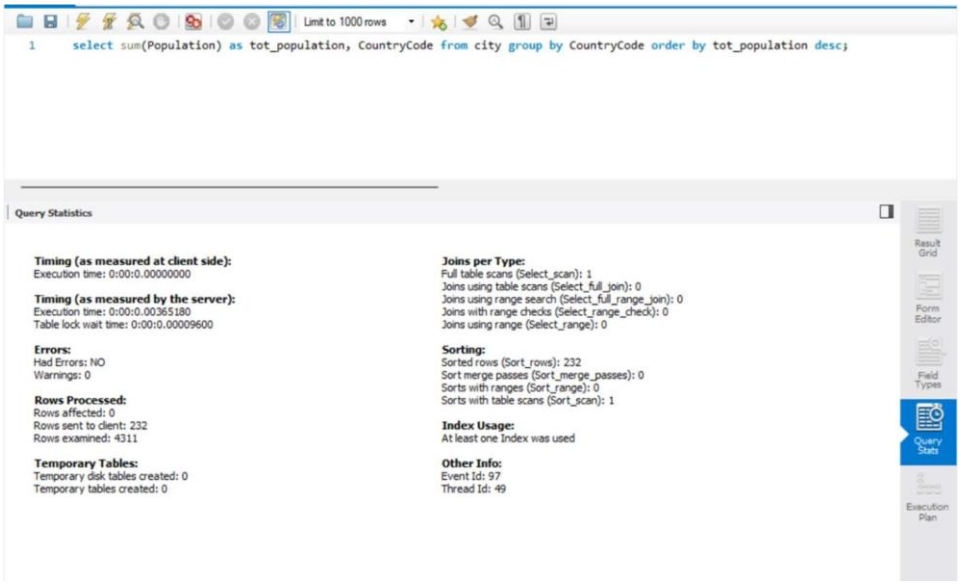


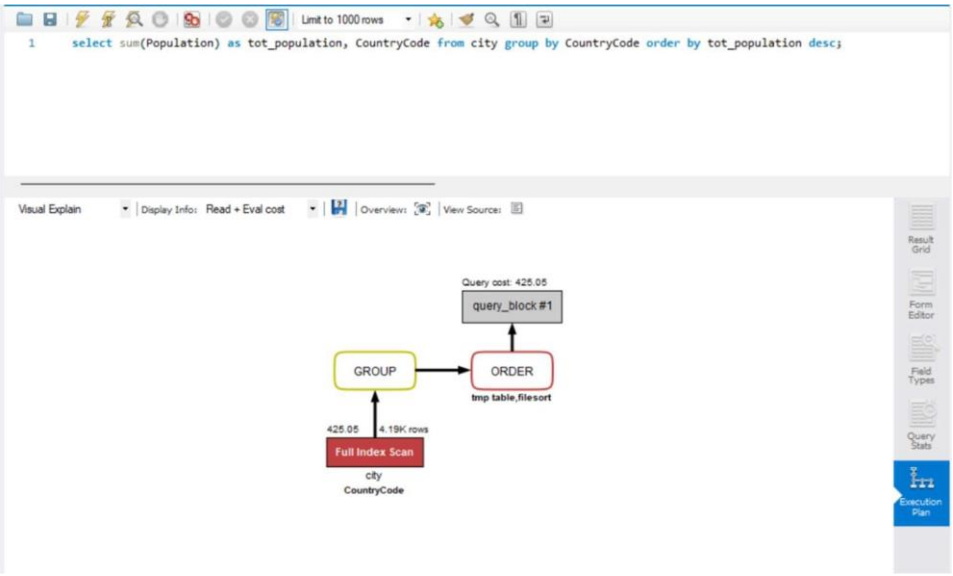
2) JOIN


Limit to 1000 rows
1 select * from city join country on city.CountryCode = country.Code;
Query Statistics
Timing (as measured at client side):
Execution time: 0:00:0.00000000
Timing (as measured by the server):
Execution time: 0:00:0.00486100
Table lock wait time: 0:00:0.00014600
Errors:
Had Errors: NO
Warnings: 0
Rows Processed:
Rows affected: 0
Rows sent to client: 1000
Rows examined: 1057
Temporary Tables:
Temporary disk tables created: 0
Temporary tables created: 0
Joins per Type:
Full table scans (Select_scan): 1
Joins using table scans (Select_full_join): 0
Joins using range search (Select_full_range_join): 0
Joins with range checks (Select_range_check): 0
Joins using range (Select_range): 0
Sorting:
Sorted rows (Sort_rows): 0
Sort merge passes (Sort_merge_passes): 0
Sorts with ranges (Sort_range): 0
Sorts with table scans (Sort_scan): 0
Index Usage:
No Index used
Other Info:
Event Id: 88
Thread Id: 49
Result Grid
Form Editor
Field Types
Query Stats
Execution Plan


Limit to 1000 rows
1 select * from city join country on city.CountryCode = country.Code;
Visual Explain
Display Info: Read + Eval cost
Overview:
View Source:
Query cost: 1536.33
query_block #1
4.31K rows
nested loop
1510.03
18 rows
Full Table Scan
country
Non-Unique Key Lookup
city
CountryCode
Result Grid
Form Editor
Field Types
Query Stats
Execution Plan

3) AGGREGATE

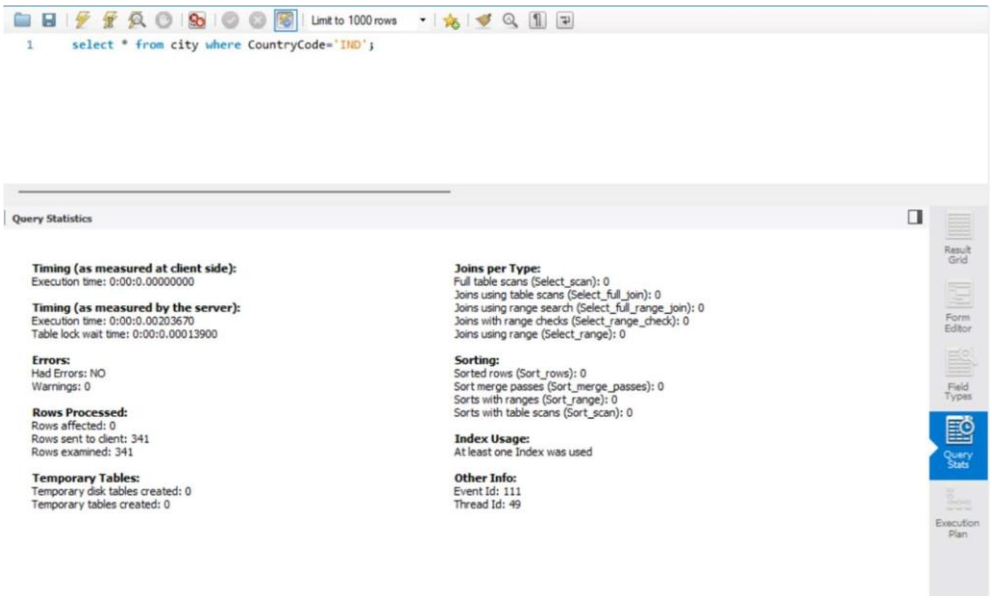

Limit to 1000 rows
1 select sum(Population) as tot_population, CountryCode from city group by CountryCode order by tot_population desc;
Query Statistics
Timing (as measured at client side):
Execution time: 0:00:0.00000000
Timing (as measured by the server):
Execution time: 0:00:0.00365180
Table lock wait time: 0:00:0.00009600
Errors:
Had Errors: NO
Warnings: 0
Rows Processed:
Rows affected: 0
Rows sent to client: 232
Rows examined: 4311
Temporary Tables:
Temporary disk tables created: 0
Temporary tables created: 0
Joins per Type:
Full table scans (Select_scan): 1
Joins using table scans (Select_full_join): 0
Joins using range search (Select_full_range_join): 0
Joins with range checks (Select_range_check): 0
Joins using range (Select_range): 0
Sorting:
Sorted rows (Sort_rows): 232
Sort merge passes (Sort_merge_passes): 0
Sorts with ranges (Sort_range): 0
Sorts with table scans (Sort_scan): 1
Index Usage:
At least one Index was used
Other Info:
Event Id: 97
Thread Id: 49
Result Grid
Form Editor
Field Types
Query Stats
Execution Plan

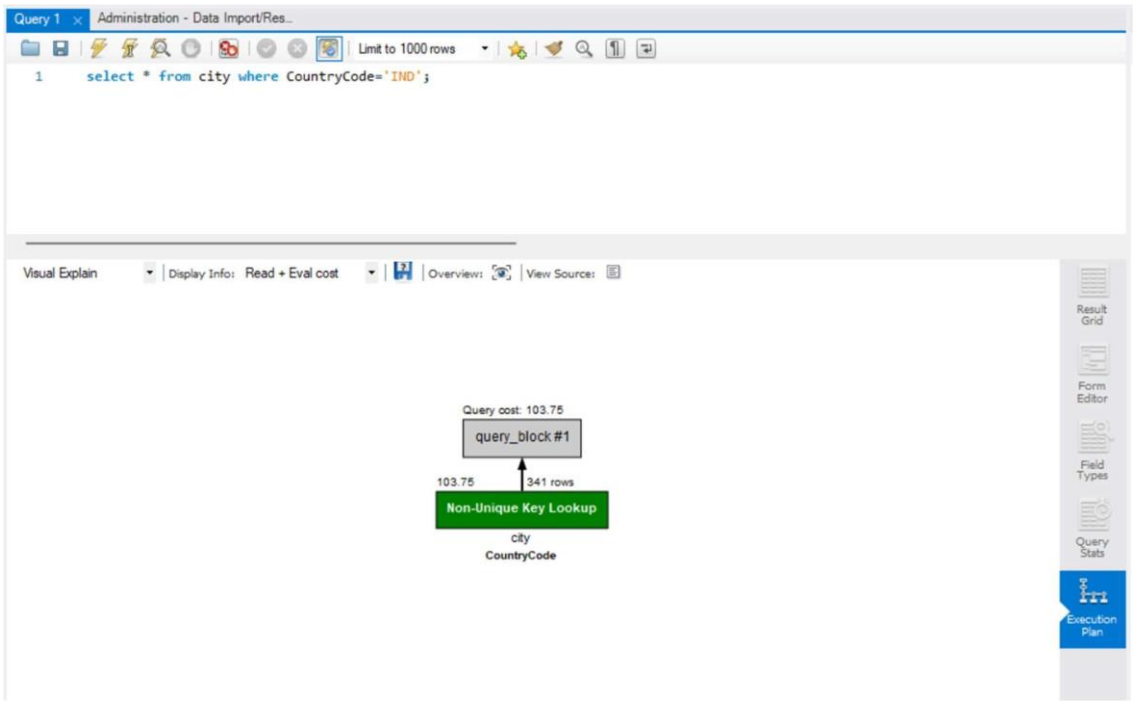

Limit to 1000 rows
1 select sum(Population) as tot_population, CountryCode from city group by CountryCode order by tot_population desc;
Visual Explain
Display Info: Read + Eval cost
Overview:
View Source:
Query cost: 425.05
query_block #1
GROUP
ORDER
tmp table,filesort
425.05
4.19K rows
Full Index Scan
city
CountryCode
Result Grid
Form Editor
Field Types
Query Stats
Execution Plan

3) Using Optimize Table –

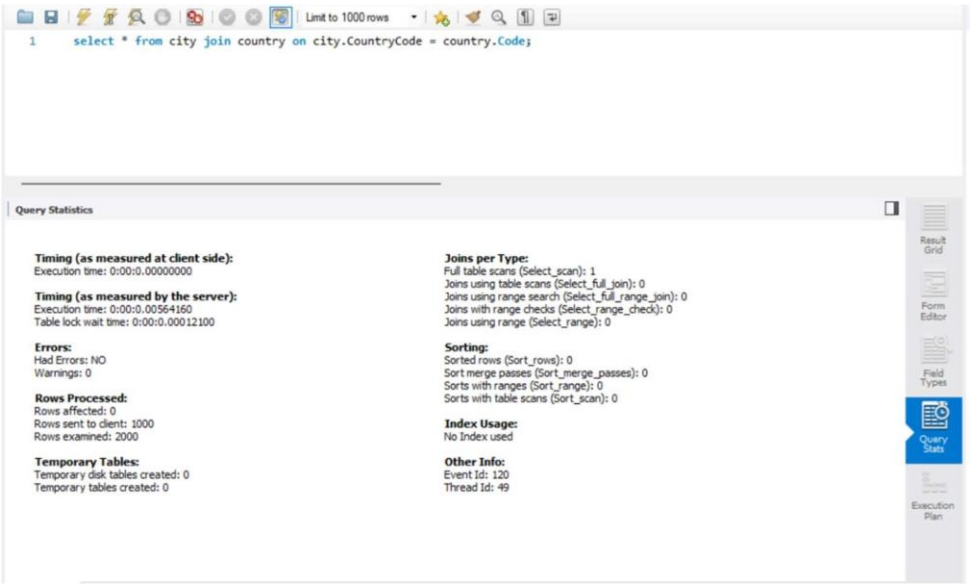
OPTIMIZE TABLE city;

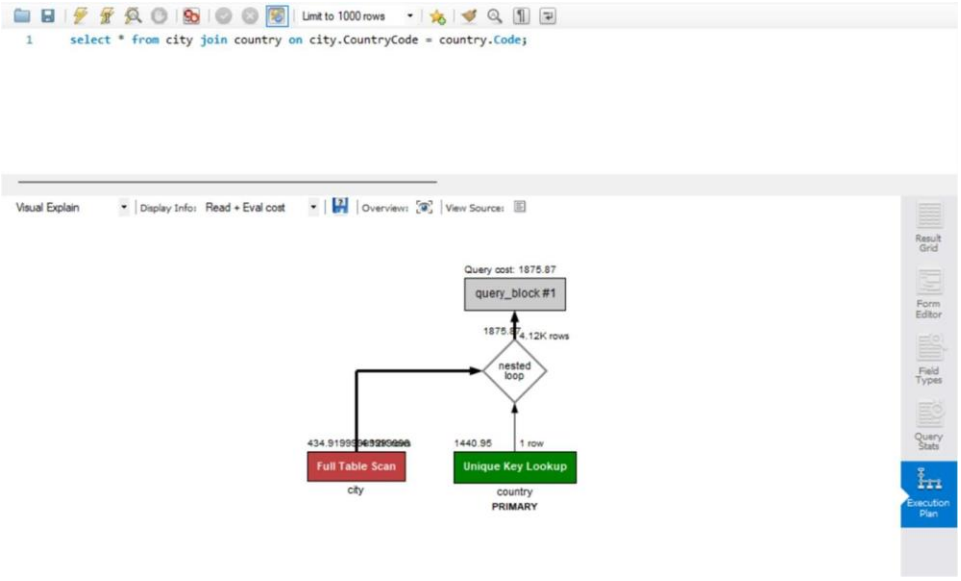
1) SELECT WHERE



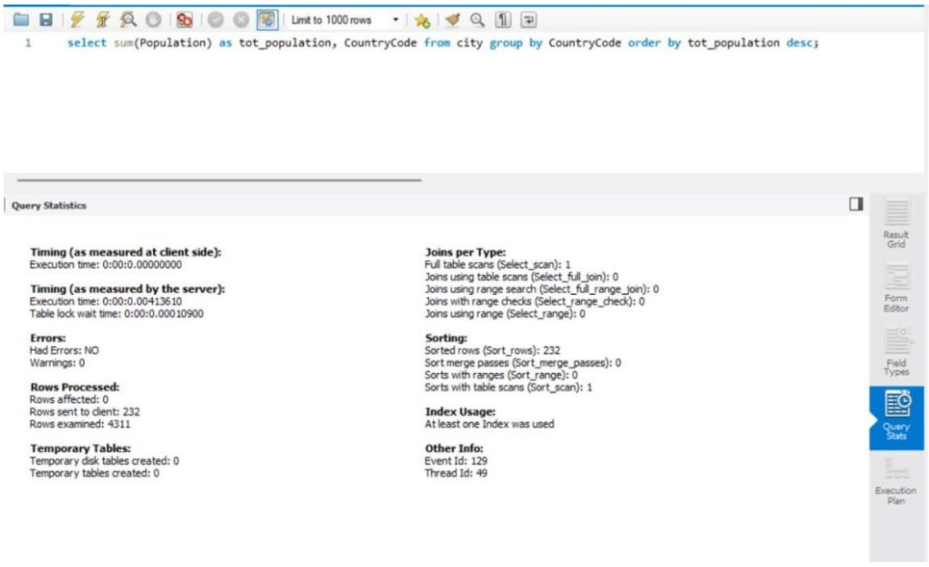


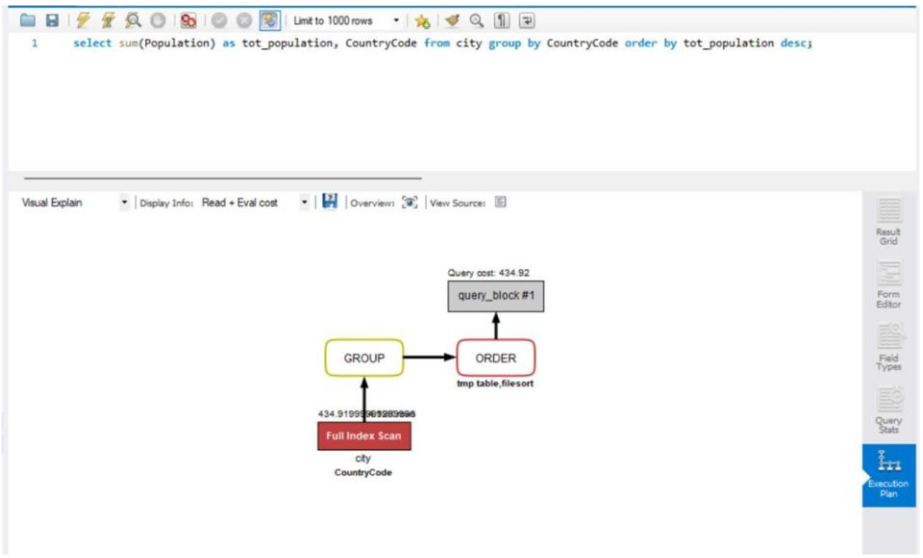
2) JOIN


Limit to 1000 rows
1 select * from city join country on city.CountryCode = country.Code;
Query Statistics
Timing (as measured at client side):
Execution time: 0:00:0.00000000
Timing (as measured by the server):
Execution time: 0:00:0.00564160
Table lock wait time: 0:00:0.00012100
Errors:
Had Errors: NO
Warnings: 0
Rows Processed:
Rows affected: 0
Rows sent to client: 1000
Rows examined: 2000
Temporary Tables:
Temporary disk tables created: 0
Temporary tables created: 0
Joins per Type:
Full table scans (Select_scan): 1
Joins using table scans (Select_full_join): 0
Joins using range search (Select_full_range_join): 0
Joins with range checks (Select_range_check): 0
Joins using range (Select_range): 0
Sorting:
Sorted rows (Sort_rows): 0
Sort merge passes (Sort_merge_passes): 0
Sorts with ranges (Sort_range): 0
Sorts with table scans (Sort_scan): 0
Index Usage:
No Index used
Other Info:
Event Id: 120
Thread Id: 49
Result Grid
Form Editor
Field Types
Query Stats
Execution Plan


Limit to 1000 rows
1 select * from city join country on city.CountryCode = country.Code;
Visual Explain
Display Info: Read + Eval cost
Overview:
View Source:
Query cost: 1875.87
query_block #1
4.12K rows
nested loop
1440.95
1 row
Full Table Scan
city
Unique Key Lookup
country
PRIMARY
Result Grid
Form Editor
Field Types
Query Stats
Execution Plan

3) AGGREGATE





Conclusion:

After indexing and optimization time taken to execute the query is less as compared to query without index and optimization