The STMS database is read-heavy database, hence indices will generally result in improved performance as long as the indexed columns are used. During this project we have aimed to create balance between optimizing read performance and avoiding excessive overhead on write operations. We have indexed columns that were used the most by the select queries which helped to save time. However these indices have caused slight delays with queries where in addition to the indexed columns, other columns had to be retrieved too.

**Tuning tables, queries and Choice of Indices:**

We haveadded indices to columns that are frequently used in filtering, joining, or sorting operations. For example, indexing the team\_id, sport\_id, and date\_of\_birth as these columns are commonly used in queries with WHERE, JOIN, or GROUP BY clauses were an initial decision we have concluded on. We have also indexed coach names and surnames because they were also very widely used.

For example, queries like SELECT team\_id, name FROM Teams WHERE team\_id NOT IN (SELECT team\_id FROM Team\_Tournament\_Participation WHERE tournament\_id = 5); suffered from slight performance degradation, as the index has to be maintained and accessed for each subquery condition and not indexed name attribute from teams was used in addition to the indexed team\_id in teams.

**Impact of changes and how the tests are concluded:**

30 common queries are presented. All of these queries are tested on the database populated with the same data on indexed and not indexed versions. The results reveal that most of the queries are positively affected by the indices. However there were certain occurrences, where the created overhead was seen to be more costly, hence not every query’s runtime increased.

The python file “performance\_test.py”, runs all of the common queries one by one on the database, and records each query’s runtime in a file. This file is run 2 times, once with the database initialization script with no indices and once with the database script with indices. Each individual query result is presented in “query\_times\_no\_indices.txt” and “query\_times\_with\_indices.txt”.

Then another python script “performance\_test\_print.py”, calculates the time improvement between each query in the indiced and non- indiced format. These results are held in the file “time\_imporvement\_with\_indices.txt”. In this file, negative percentage increase’s correlate that the query was indeed ran faster. First 5 queries are also presented below in order to better showcase the logic:

By performing these experiments, we can assess which indices provide the most significant improvements and which types of queries benefit the most. For instance, queries that involve filtering, sorting, or joining on indexed columns often show substantial performance gains.

Query: SELECT \* FROM Players;

Time in query\_times\_no\_indices.txt: 0.076545 seconds

Time in query\_times\_with\_indices.txt: 0.067461 seconds

Time Difference: -0.009084 seconds

Percentage Increase: -11.87%

Query: SELECT DISTINCT name FROM Sports;

Time in query\_times\_no\_indices.txt: 0.001313 seconds

Time in query\_times\_with\_indices.txt: 0.000702 seconds

Time Difference: -0.000611 seconds

Percentage Increase: -46.53%

Query: SELECT name FROM Tournaments WHERE sport\_id = (SELECT sport\_id FROM Sports WHERE name = 'basketball');

Time in query\_times\_no\_indices.txt: 0.000847 seconds

Time in query\_times\_with\_indices.txt: 0.000580 seconds

Time Difference: -0.000267 seconds

Percentage Increase: -31.52%

Query: SELECT \* FROM Coaches;

Time in query\_times\_no\_indices.txt: 0.000872 seconds

Time in query\_times\_with\_indices.txt: 0.000705 seconds

Time Difference: -0.000167 seconds

Percentage Increase: -19.15%

Query: SELECT \* FROM Sports WHERE name = 'basketball';

Time in query\_times\_no\_indices.txt: 0.000317 seconds

Time in query\_times\_with\_indices.txt: 0.000282 seconds

Time Difference: -0.000035 seconds

Percentage Increase: -11.04%

Query: SELECT team\_id, COUNT(\*) AS num\_players FROM Players GROUP BY team\_id;

Time in query\_times\_no\_indices.txt: 0.002228 seconds

Time in query\_times\_with\_indices.txt: 0.001822 seconds

Time Difference: -0.000406 seconds

Percentage Increase: -18.22%