# Query 1

## Query Details:

| SELECT AVG(r.Rating), c.Description, c.CourseName, c.CourseNumber, c.DeptAbv FROM Reviews r LEFT JOIN Courses c ON r.CRN = c.CRN LEFT JOIN Enrollments e ON c.CRN = e.CRN GROUP BY c.CRN LIMIT 15; |
| --- |

## EXPLAIN ANALYZE results:

| EXPLAIN

| -> Limit: 15 row(s) (actual time=8.779..8.783 rows=15 loops=1)

-> Table scan on <temporary> (actual time=0.002..0.005 rows=15 loops=1)

-> Aggregate using temporary table (actual time=8.777..8.781 rows=15 loops=1)

-> Nested loop left join (cost=801.75 rows=1000) (actual time=0.071..5.774 rows=1177 loops=1)

-> Nested loop left join (cost=451.75 rows=1000) (actual time=0.058..2.536 rows=1000 loops=1)

-> Table scan on r (cost=101.75 rows=1000) (actual time=0.040..0.355 rows=1000 loops=1)

-> Single-row index lookup on c using PRIMARY (CRN=r.CRN) (cost=0.25 rows=1) (actual time=0.002..0.002 rows=1 loops=1000)

-> Index lookup on e using CRN (CRN=c.CRN) (cost=0.25 rows=1) (actual time=0.002..0.003 rows=1 loops=1000)

|

1 row in set (0.21 sec)

## 3 Different Indexing Designs:

**Indexing on c.DeptAbv results:**

| EXPLAIN

| -> Limit: 15 row(s) (actual time=9.361..9.366 rows=15 loops=1)

-> Table scan on <temporary> (actual time=0.002..0.005 rows=15 loops=1)

-> Aggregate using temporary table (actual time=9.360..9.364 rows=15 loops=1)

-> Nested loop left join (cost=825.37 rows=1185) (actual time=0.071..6.242 rows=1177 loops=1)

-> Nested loop left join (cost=451.75 rows=1000) (actual time=0.061..2.926 rows=1000 loops=1)

-> Table scan on r (cost=101.75 rows=1000) (actual time=0.044..0.375 rows=1000 loops=1)

-> Single-row index lookup on c using PRIMARY (CRN=r.CRN) (cost=0.25 rows=1) (actual time=0.002..0.002 rows=1 loops=1000)

-> Index lookup on e using CRN (CRN=c.CRN) (cost=0.26 rows=1) (actual time=0.002..0.003 rows=1 loops=1000)

|

1 row in set (0.22 sec) (this is poorer in performance compared to the raw one, 0.22 > 0.21)

**Indexing on c.CourseNumber:**

| EXPLAIN

| -> Limit: 15 row(s) (actual time=9.357..9.362 rows=15 loops=1)

-> Table scan on <temporary> (actual time=0.002..0.005 rows=15 loops=1)

-> Aggregate using temporary table (actual time=9.357..9.360 rows=15 loops=1)

-> Nested loop left join (cost=825.37 rows=1185) (actual time=0.105..6.218 rows=1177 loops=1)

-> Nested loop left join (cost=451.75 rows=1000) (actual time=0.075..3.035 rows=1000 loops=1)

-> Table scan on r (cost=101.75 rows=1000) (actual time=0.053..0.419 rows=1000 loops=1)

-> Single-row index lookup on c using PRIMARY (CRN=r.CRN) (cost=0.25 rows=1) (actual time=0.002..0.002 rows=1 loops=1000)

-> Index lookup on e using CRN (CRN=c.CRN) (cost=0.26 rows=1) (actual time=0.002..0.003 rows=1 loops=1000)

|

1 row in set (0.22 sec) (this is poorer in performance compared to the raw one, 0.22 > 0.21)

**Indexing on r.Rating:**

| EXPLAIN

| -> Limit: 15 row(s) (actual time=8.617..8.621 rows=15 loops=1)

-> Table scan on <temporary> (actual time=0.002..0.004 rows=15 loops=1)

-> Aggregate using temporary table (actual time=8.616..8.619 rows=15 loops=1)

-> Nested loop left join (cost=825.37 rows=1185) (actual time=0.060..5.598 rows=1177 loops=1)

-> Nested loop left join (cost=451.75 rows=1000) (actual time=0.051..2.767 rows=1000 loops=1)

-> Table scan on r (cost=101.75 rows=1000) (actual time=0.034..0.370 rows=1000 loops=1)

-> Single-row index lookup on c using PRIMARY (CRN=r.CRN) (cost=0.25 rows=1) (actual time=0.002..0.002 rows=1 loops=1000)

-> Index lookup on e using CRN (CRN=c.CRN) (cost=0.26 rows=1) (actual time=0.002..0.003 rows=1 loops=1000)

|

1 row in set (0.22 sec)

**Conclusion**

It is unlikely that indexing will help here since we are finding the average rating, hence the need to go through each and every row. so indexes do not speed up this query at all.

# Query 2

## Query Details:

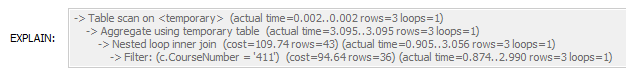
| SELECT c.CourseName, e.CRN, AVG(e.GPA) AS averageGPA FROM Enrollments e LEFT JOIN Courses c ON c.CRN = e.CRN WHERE c.DeptAbv='CS' AND c.CourseNumber='411' GROUP BY e.CRN; |
| --- |

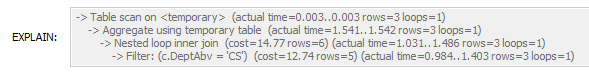
## EXPLAIN ANALYZE results:

## 

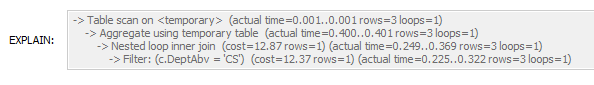
## 3 Different Indexing Designs:

**Indexing on c.DeptAbv:**

**Indexing on c.CourseNumber:**



**Indexing on c.DeptAbv and c.CourseNumber:**



**Conclusion:**

Since both Enrollments and Courses have CRN as a Primary key, they already have CRN as an index. Thus, two other indexes that would make sense are c.DeptAbv and c.CourseNumber, which both halved the aggregation time, while having increased time for the inner join and filtering.

Indexing on both reduced the aggregation time and filter time even more but increased the inner join time. However, since the inner join took only 0.376 s for the initial query while the aggregation time was 6.104, it makes sense that we prioritize reduction in the aggregation time.