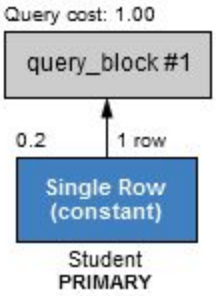
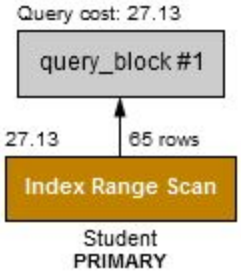
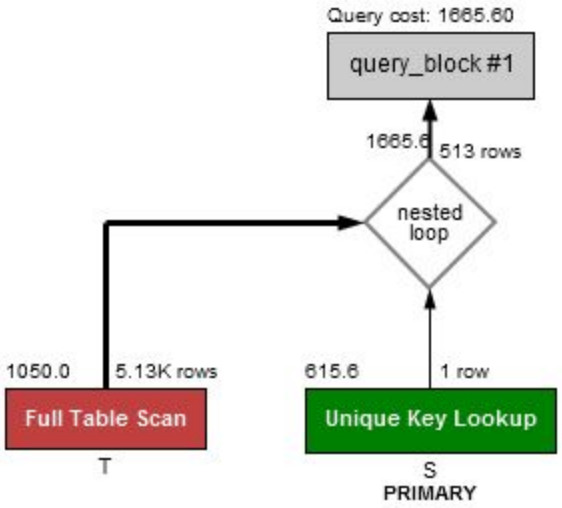
* Screenshots for the explain plans are also available in a separate folder.



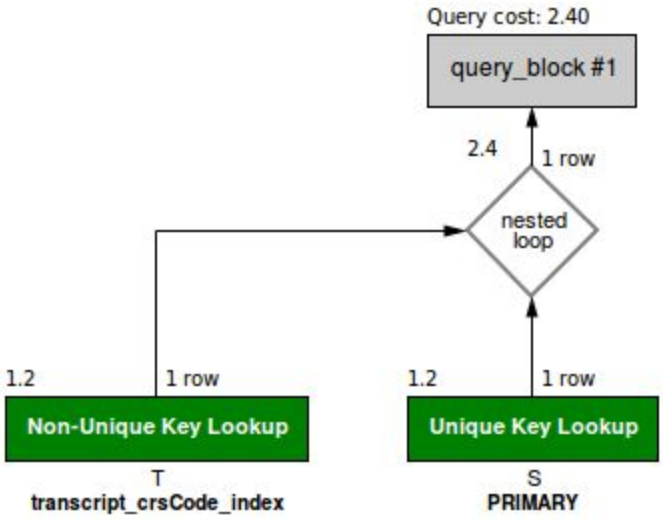
Above is the the visual explain plan for the first query, in our original Schema we created the table with a Primary Key, so there was no need to make any changes here. The running time of this query was 0.001 seconds.



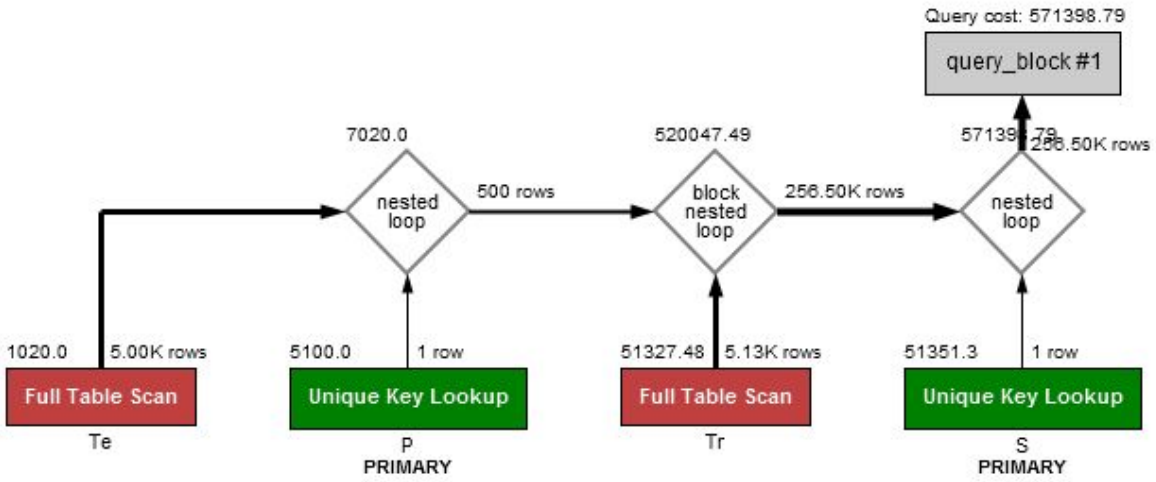
Like with the first query, we initially created the Student table with a primary key, there was no much we could do. The running time was 0.001 seconds.



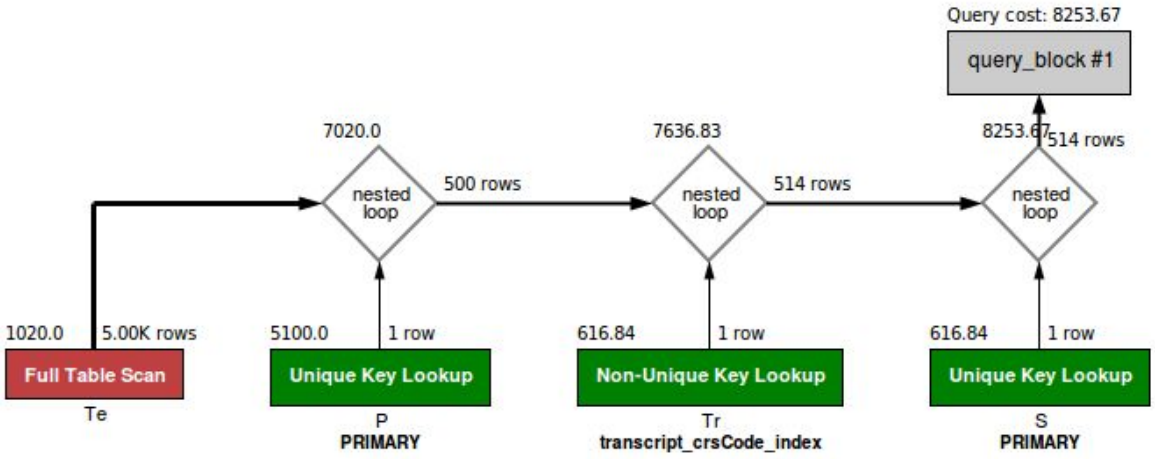
For query 3, we can see that the Transcript table was doing a full table search, which required a lot of rows to be scanned. We added an index to the “crsCode” column to allow for more efficient searching. The resulting explain plan is below.



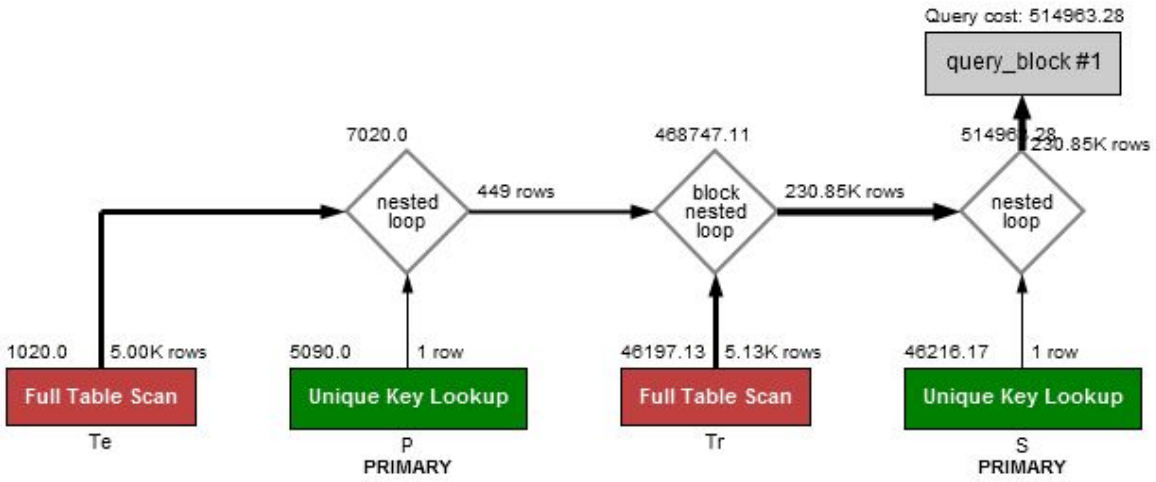
As you can see, adding the index did the job, as we were able to cut down the query cost to 2.4 compared to over 1000 before. Running time of the tuned query was 0.001 seconds compared to the about 0.50 seconds running on MySQL 5.5.4.



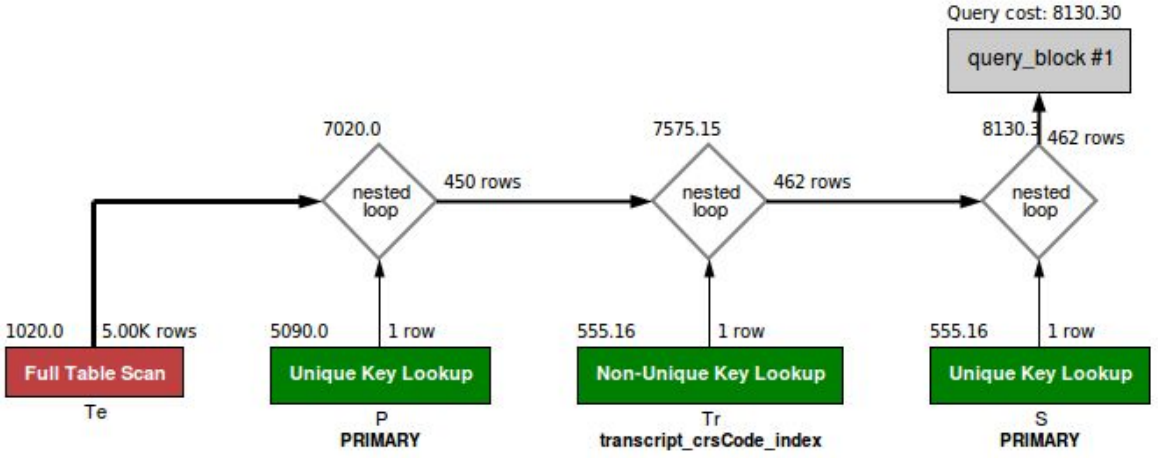
Here we see two full table scans and a block nested loop, we indexed on the Teaching and Transcript tables, and was able to completely get rid of the block loop and one full table scan. The Teaching full table scan seemed to stay and using index hinting had a negative effect so I took it off. The run time before tuning was 4.80 seconds on MySQL 5.5. The new explain plan is below.



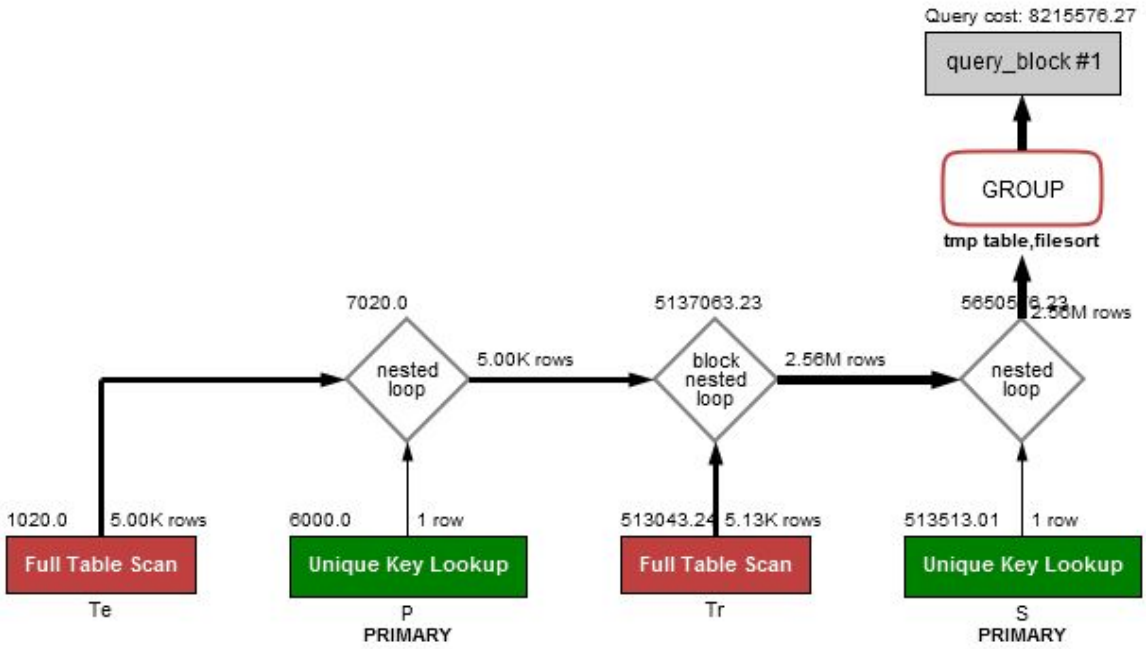
You can see the change in query cost. The new run time is 0.043 seconds.



Query 5, was acting in a very similar way to the previous one, adding the indexes in the previous query also helped solve this one. The time was also about 4.8 seconds for this one before and was down to 0.048 seconds.



The cost was cut from over half a million to a bit over 8000, like the previous ones, using index hints slowed down the query instead of helping so we left it as is.



For the final query we attempted to index on Student name, in conjunction with the previous changes. An optimization on the GROUP BY was attempted but could not be done with neither a loose or tight index since we are doing a multitable query and using non-constant equalities. The result can be seen below, with a run time of 0.041 seconds.