**Database Technologies**

**UE19CS344**

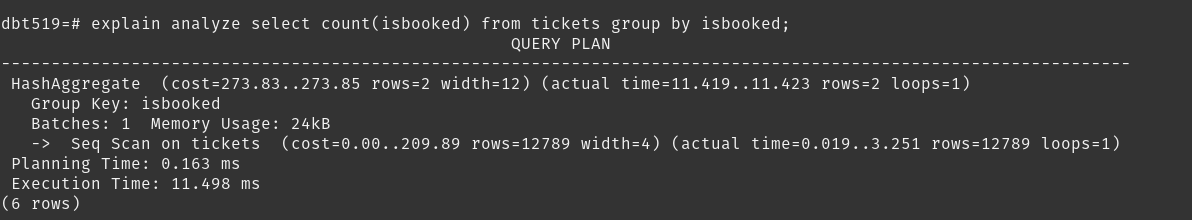
**6th Semester, Academic Year 2021-22**

Week #7: Query Optimization

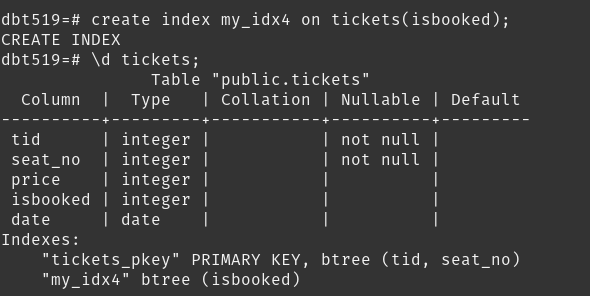
Date: 8/3/2022

|  |  |  |
| --- | --- | --- |
| Name :  SUMUKH RAJU BHAT | SRN :  PES1UG19CS519 | Section :  H |

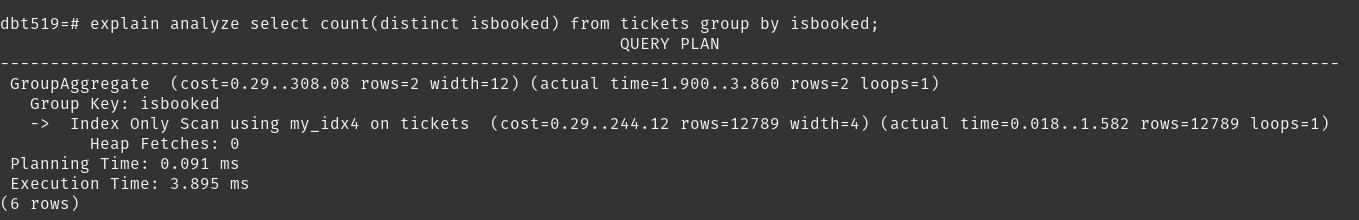
1. Queries containing group by:

We can see that it hashes based on group key and then does a sequential scan on group key which is performance bottleneck for the query.

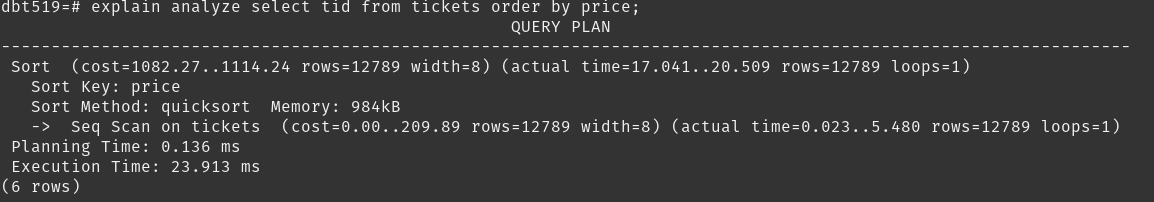
Hence, we add a secondary index on the group key:



We can now see that, index scan is performed using the new secondary index added. It results in approximately 4x faster execution:

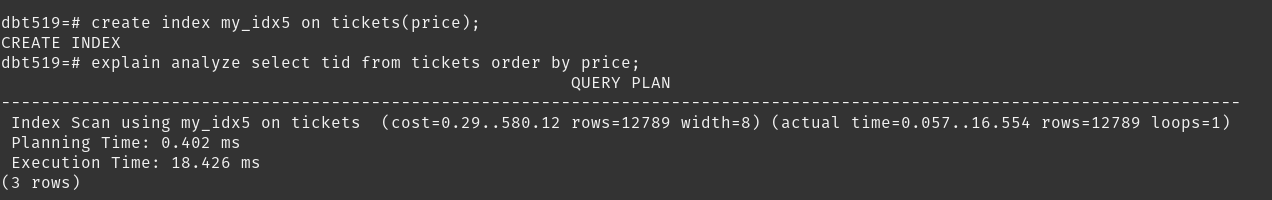


2. Queries containing Order by:

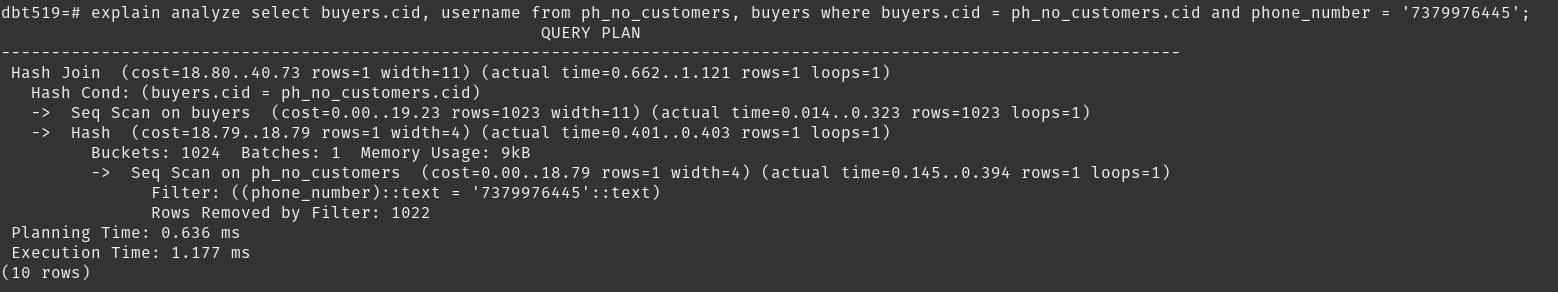
We can see that it does quicksort using sequential scan on the sort key, which is definetely a performance bottleneck for the query.

Hence we add a secondary index on the sort key.

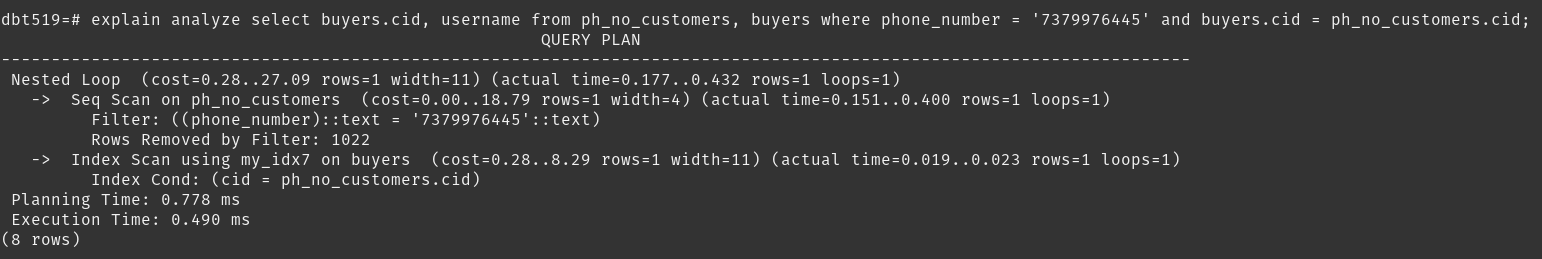
Now we can see index scan being performed and decrease in execution time:

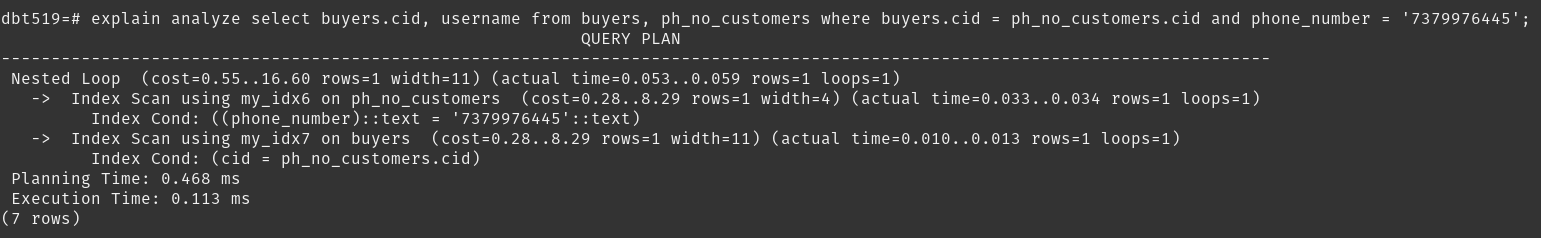


3. Queries containing equijoin:

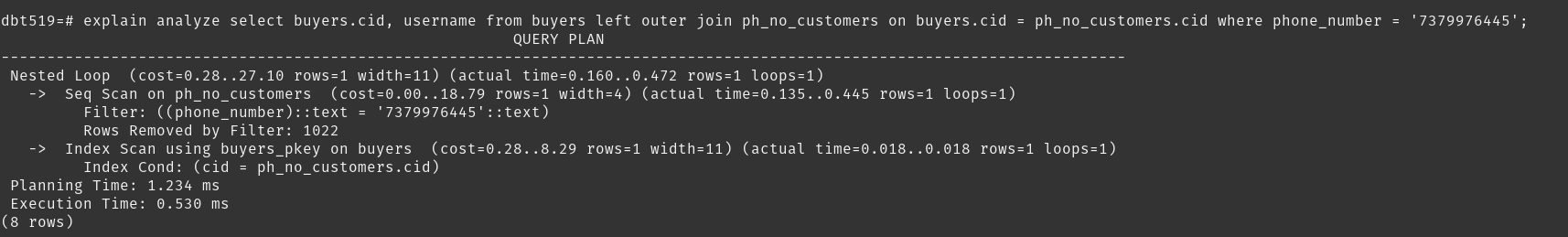
We can see that it does a hash join using sequential scan on the join attribute on both tables, which is a performance bottleneck. (Primary key constraint for cid is removed for both tables for demonstration of secondary indices)

Hence we add secondary indices on join condition and we get 2x improvement in execution speed:

Eventhough index scan is performed on join attribute, there is a sequential scan on phone\_number attribute. We can also add a secondary index on it and we see a 10x improvement in execution speed:



4. Queries containing outer joins:

Similar to inner joins, we add secondary indices on join attribute and get 5x improvement in execution time:

