Revision 11/28/2022

ComS 363 Fall 2022
Homework 4 Q3
solution

Q3 (65 points) Optimize the following queries in MySQL by creating an appropriate index such that the DBMS uses the created index when executing the same query. The use of the index must reduce the disk I/O cost, which in many cases results in less query execution time. One index may be useful for more than one query.

```
-- ID7
set @hashtag = "GOPDebate";
set @state_name = "North Carolina";
SELECT count(t.tid) as tweet_count, u.screen_name, u.category
FROM tweets as t, users as u, hashtags as h
WHERE t.posting_user = u.screen_name AND t.tid = h.tid and
h.name = @hashtag and u.state = @state_name and t.post_month = 2 and t.post_year = 2016
GROUP BY u.screen_name, u.category
ORDER BY tweet_count DESC
LIMIT 5;
```

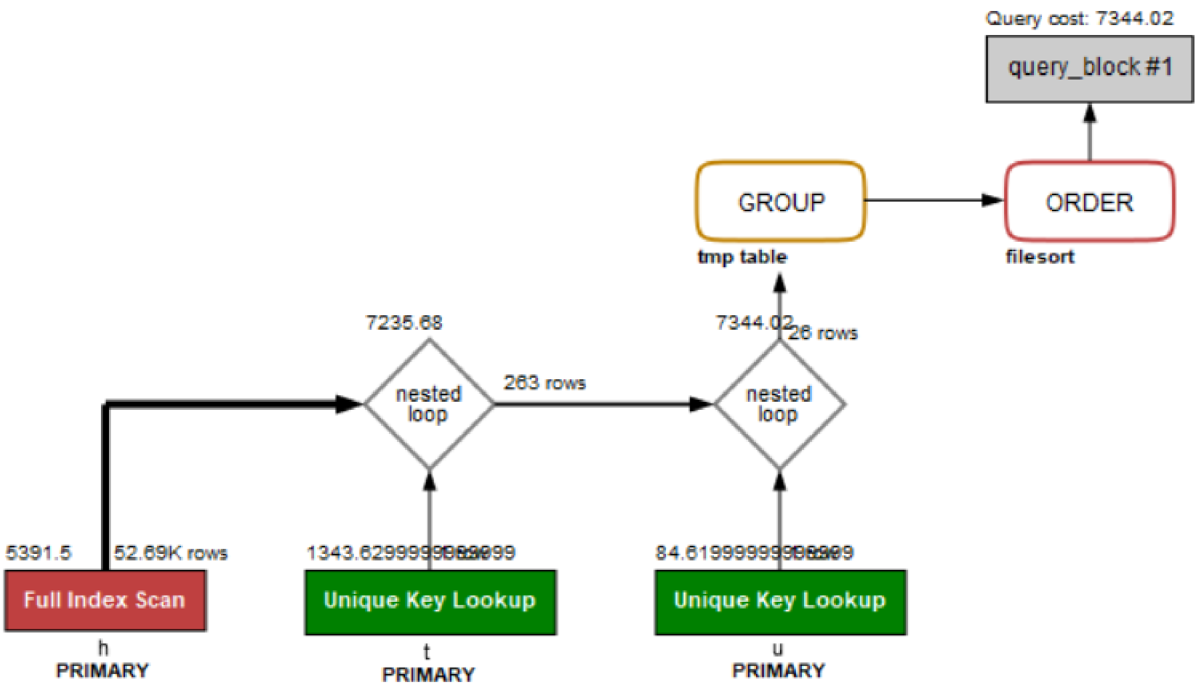
```
-- ID23
SELECT h.name, COUNT(h.tid) as num_uses
FROM Tweets t, Users u, HashTags h
WHERE u.subcategory = 'GOP' AND FIND_IN_SET(t.post_month, '1,2,3')
    AND t.post_year = 2016  AND u.screen_name = t.posting_user AND h.tid = t.tid
GROUP BY h.name
ORDER BY num_uses DESC
LIMIT 5;
```

See Q3.sql for the create statement indexes.

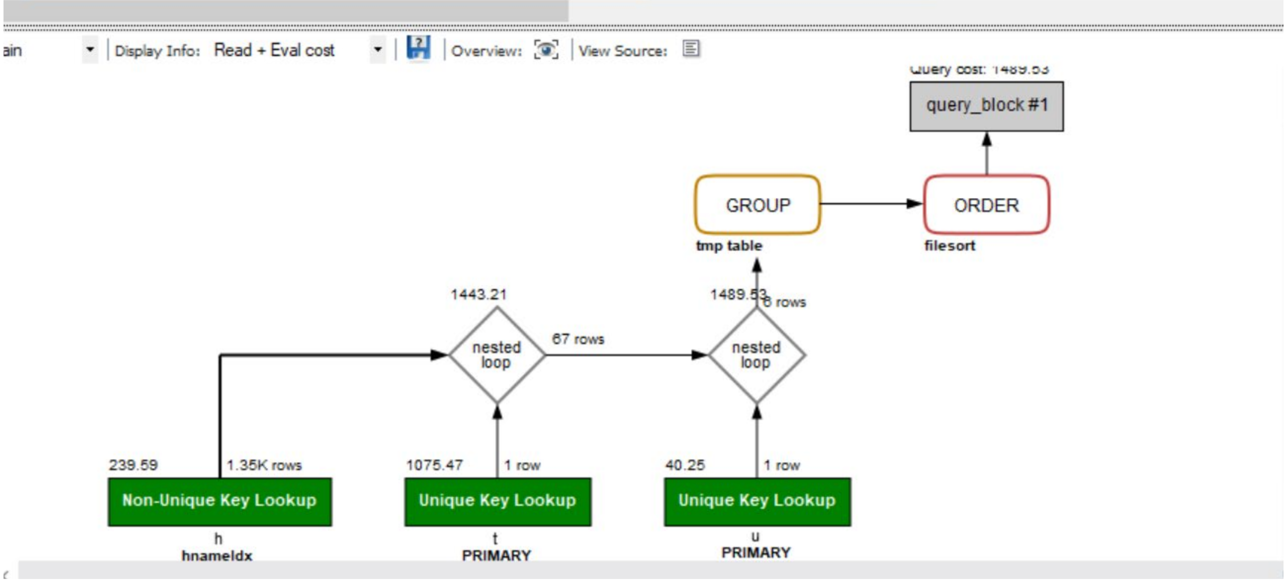
Students may have a different execution plan, but the plan must show the reduced I/O cost for the whole plan and the created index got used. The database buffer pool size and disk I/O costs can be different.

Before:

Revision 11/28/2022



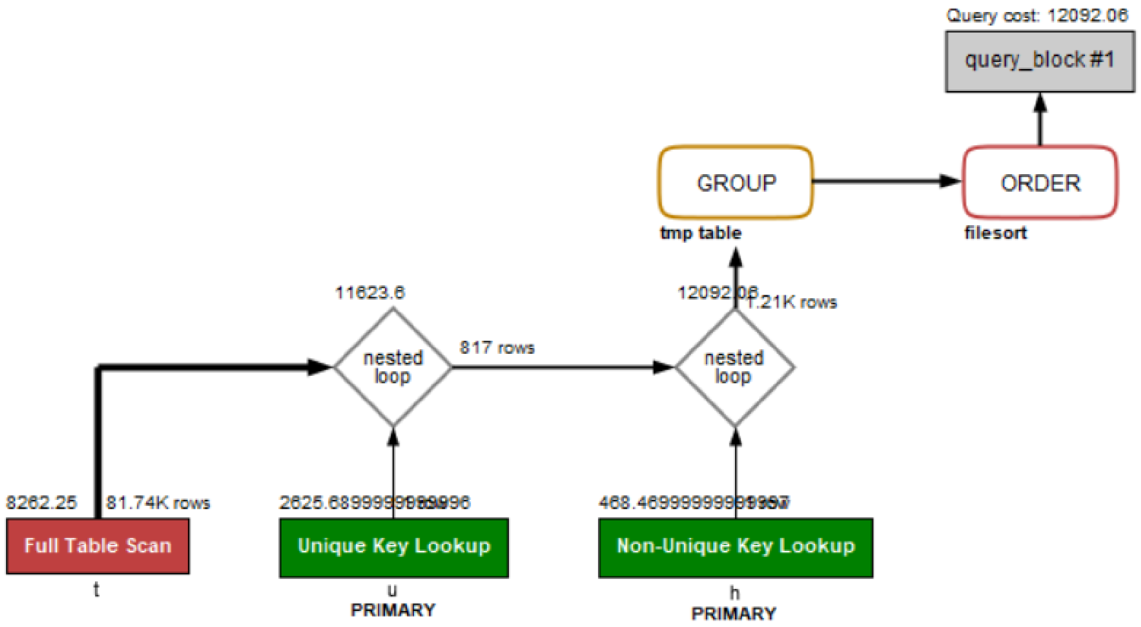
```
create index hnameIdx on hashtags(name);
set @hashtag = "GOPDebate";
set @state_name = "North Carolina";
SELECT count(t.tid) as tweet_count, u.screen_name, u.category
FROM tweets as t, users as u, hashtags as h
WHERE t.posting_user = u.screen_name AND t.tid = h.tid and
```

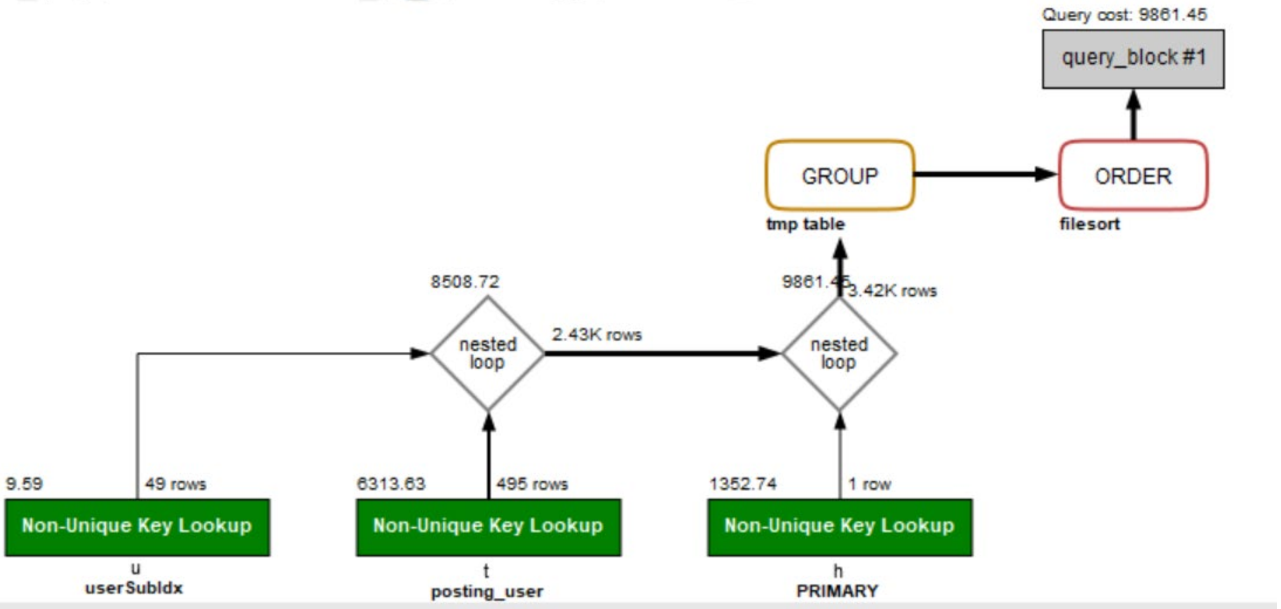


ID7: After

ID23: Before

Revision 11/28/2022





ID27: After

Table 1: MySQL Innodb buffer pool size: __8______ Mbytes

Query ID	Create index statement for each query; your execution plan needs to use the created index and reduce disk I/O cost.	Before optimization (disk I/O cost)		After optimization (disk I/O cost)	
		Run1	Run2	Run1	Run2
ID7	create index hnameIdx on hashtags(name); or create index userState on users (state);	8442.79	7344.02	1489.53 471.43	1489.53 468.39
ID23	create index userSubIdx on users (subcategory);	12092.06	11958.18	8961.45	9861.45