CS 122A: Introduction to Data Management – Spring 2018

Homework 7: SQL Design and Indexing (100 points)

Due Date: **Tuesday, May 29 (5:00 PM)**

Submission

All HW assignments should contain both your student ID and your name and must be submitted online (e.g., 12345678_John_Doe_HW7.pdf) via the HW7 dropbox on Gradescope. See the table below for the HW 7 submission opportunities. Note that after 5 PM on Wednesday the 30th no further HW 7 submissions will be accepted. (We will be releasing the solution at that time.) Please strive to get all your work in on time! If possible, try to save the one dropped assignment for the end of the term when you are most likely to want/need it.

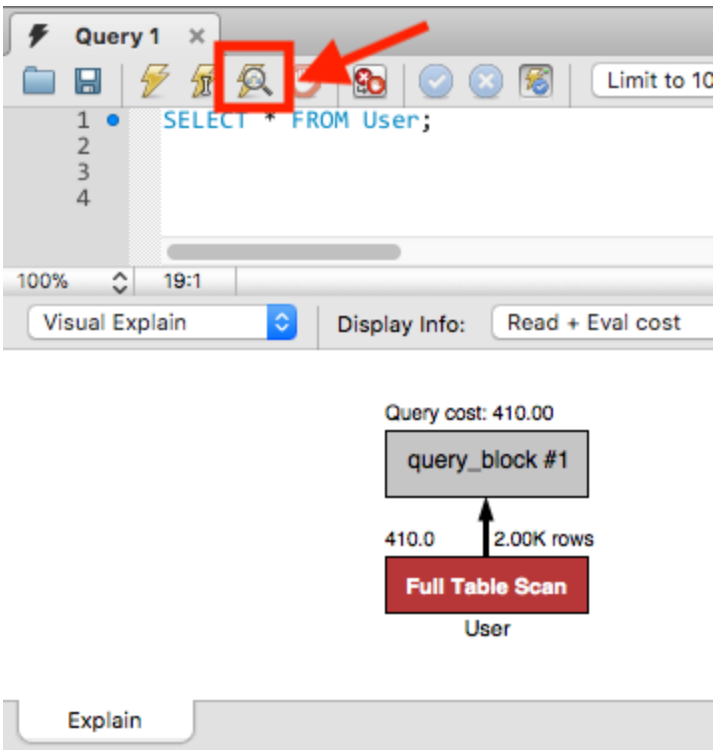
Date / Time	**Grade Implications**
Tuesday, May 29 (5:00 PM)	Full credit will be available
Wednesday, May 30 (5:00 PM)	20 points will be deducted

SQL Design and Indexing (SQL) [100 pts]

A physical query plan is like a routine (or a relational algebra expression) that the DBMS follows to assemble the requested query results from the underlying base tables. Different queries will have different physical plans. in fact, the same query may be translated into different physical plans depending on the physical database design. MySQL provides an 'EXPLAIN' function, as shown in the figure below, to help you to check the query plan of your query. You can use this function to examine how is your query will be executed internally, what indexes are being used, and the total cost of your query.

Since our website is getting larger, simple queries are taking too long to execute. The CTO of Peeeza.com is asking the database design team to compile a report about the performance of the system and provide solutions to improve the performance of Peeeza.com website. Since you have taken CS122A, you are more than capable of taking care of that report and presenting it to the CTO on time.

Note: For this assignment, you can turn in a PDF by cutting/pasting from MySQL into a copy of HW7 template and then **PDF-printing** the results. **Make sure your solution is 6 pages only and all screenshots are properly included.**



1. [12 pts] To start down the path of exploring physical database designs (indexing) and query plans, start by checking the queries plans for each of the following queries without any indexes using the EXPLAIN function (see above). Report the query plan of each query by taking snapshots and pasting them into your copy of HW7 template file. (Just take a snapshot of the query plan, not the whole screen.)

   a) [3 pts] SELECT * FROM Post WHERE popularity > 98 AND popularity <= 100;

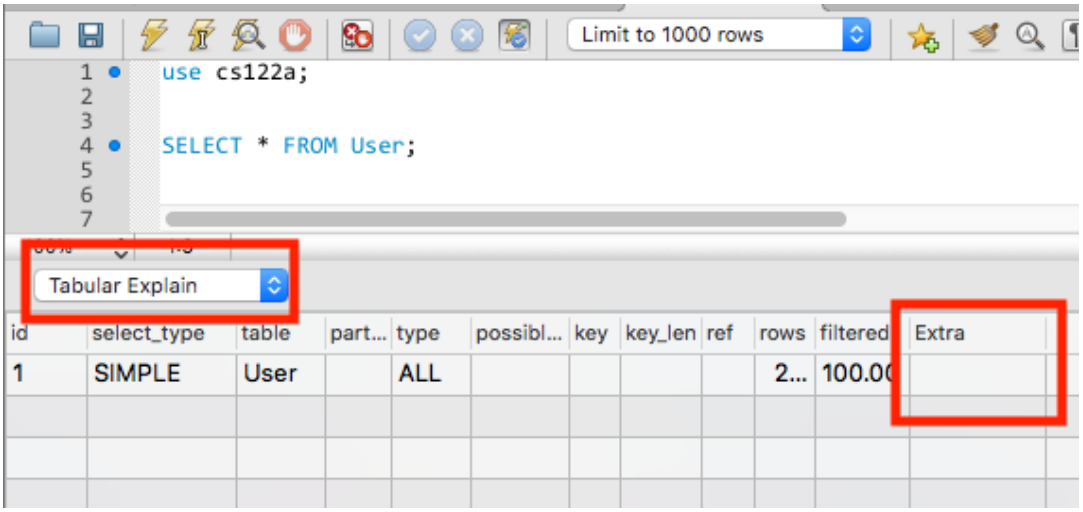
   b) [3 pts] SELECT * FROM User WHERE first_name LIKE '%mil%';

   c) [3 pts] SELECT * FROM User WHERE first_name LIKE 'Jay%';

   d) [3 pts] SELECT COUNT(*) FROM Post WHERE popularity = 93;

2. [10 pts] Now create indexes (which are B+ trees, under the hood of MySQL) on the User.first_name attribute and Post.popularity *separately*. (I.e., create two indexes, one per table.) Paste your CREATE INDEX statements below.

3. [12 pts] Re-explain the queries in Q1 and indicate whether the indexes you created in Q2 are used, and if so, whether the uses are index-only plans or not. Report the query plan after each query, as before. **Note**: you might find it useful to also look at the 'Tabular Explain' result in addition to the 'Visual Explain' one (picture below) to help verify your answers on index-only plan. Look carefully at the 'Extra' field. For more information look for 'Using Index': HERE



a) [3 pts] SELECT * FROM Post WHERE popularity > 98 AND popularity <= 100;

b) [3 pts] SELECT * FROM User WHERE first_name LIKE '%mil%';

c) [3 pts] SELECT * FROM User WHERE first_name LIKE 'Jay%';

d) [3 pts] SELECT COUNT(*) FROM Post WHERE popularity = 93;

4. [21 pts] Examine each of the above queries with and without the use of an index. Please **briefly** answer the following questions.

   a) [7 pts] For range queries (e.g., query a), explain whether an index is useful and why. (Assume the number of result records in the selected range is extremely small compared to the total number of records in the file.)

   b) [7 pts] For each of the LIKE queries (b and c), explain whether an index is useful and why or why not (<= 2 sentences per query).

   c) [7 pts] For equality queries (e.g., query d), explain whether an index is useful and why (assuming the number of selected result records is extremely small compared to the number of records in the file).

5. [21 pts] It's time to go one step further and explore the notion of a "composite Index", which is an index that covers several fields together.

   1) SELECT * FROM Assignment WHERE dept = 'CS' and cno = '122A';
   2) SELECT * FROM Assignment WHERE cno = '222';

   a) [5 pts] Create a composite index on the attributes dept and cno (in that order!) of the Assignment table. Paste your CREATE INDEX statement below.

   b) [6 pts] 'Explain' the queries 1 and 2 above. Report on the query plan of each query, as before. Be sure to look carefully at each plan.

   c) [10 pts] Report for each query whether the composite index is used or not and why.

6. [24 pts] For each of the following queries, indicate whether a use of an index would be helpful or not. If so, specify which **tables** and **attributes** the index should be created on and also what the best choice between a **clustered** or **unclustered** index would be. For the purpose of this question, you can assume that the system provides that choice (even if it doesn't).

a) [8 pts] SELECT * FROM Post WHERE user_id = “998”; (Assume the number of result records is < 1%)

b) [8 pts] SELECT kind, COUNT(*) as cnt FROM Post GROUP BY kind;

c) [8 pts] SELECT * FROM Assignment;