ADVANDB Release Date: Jan 27 / 28, 2016

MCO1: Query Processing and Optimization Final Due Date: Feb 29 / Mar 01, 2016

Project Description

The course project on *Query Processing and Optimization* is a venue for students to conduct a comparative analysis of the impact of query operations to an application's performance. Students will use the CBMS dataset to formulate both poorly and properly written query statements and determine how these affect the response time of an application. They will also learn the correlation between good database design (i.e., normalization) and faster query operations (number of join operations). Finally, the large sample database to be used in this project will also provide an opportunity for students to understand how the dataset itself affect query operations. All these findings will be documented in a technical report that students will be required to prepare and present as part of the output.

Overview of the CBMS Dataset

In 2013, the Angelo King Institute of Economic Business Studies partnered with the College of Computer Studies to develop digital data collection survey forms using mobile devices, and a web portal to manage the distribution of digital survey forms and submission of collected data for CBMS.

CBMS (Community-Based Monitoring System) provides an organized approach for monitoring poverty through the collection of household data at the local government (or barangay) level. Household data cover a wide range, from the bio-profile of the different household members, to their educational attainment, work experience (both local and overseas); household state and events, such as availability of appliances and any death or crime against family members; and services that the household is receiving from the government, which include health services, garbage collection and water supply. These collected data can then be processed and utilized by various organizations and agencies to deliver better services aimed at addressing the needs of the community. Evidence-based decision making can be performed by barangay officials, local government offices (LGO), regional district offices (RDO), NGOs and the national government through the development of strategic plans and proper utilization of limited resources that are focused towards poverty reduction.

For this trimester, the CBMS dataset contains household information collected from the provinces of Bohol, Tarlac and Palawan from 2014 - 2015. The data dictionary is found in the Appendix.

Methodology

Students are to form teams with **3 - 4 members, subject to certain grouping constraints as discussed by your teacher**. Each team will work on the same CBMS dataset that is available for download at …

To proceed with this project, each team should conduct the following:

Step 1. Develop a Simple Query Application

Write a simple application to issue queries to the CBMS database. The application may or may not be web-based, as long as there is an interface for selecting the query to be executed and viewing the query results. Each team can use its preferred programming language from the following choices – Java, C/C++, C#, Visual Basic, or Python.

The following types and number of queries must be formulated:

No. of Tables per Query	No. of Queries	Requirements
1	2	Single-table query must be written for either of the two largest tables (degree > 20) only.
2	2	Query must support conditions to test the effect of the dataset (different input values) on the performance.

3	2	Query must be relevant to the given purpose of the CBMS dataset. Query must be complex and utilize the aggregate functions and group by clause of SQL.
4 - 5	1	

Criteria for grading the Queries:

- Uniqueness. Though duplicates cannot be avoided due to the large number of student groups, there should be evidence of careful thinking and analysis in formulating "less common but relevant and useful" queries to receive full credit.
- Flexibility. Queries should be formulated to provide variances of results to the user, while allowing your team to test the effect of the dataset on the application's performance. That is, a single query should allow different search strings, and your team should use this to benchmark how a query's response time may degrade based on the content of the database.
- Relevance. Formulated queries should not randomly join tables based on the foreign key constraint, but will also be graded based on the relevance of the query to the problem domain as described in the "Overview of the CBMS Dataset" section of this document.
- Complexity. Evidence of clear appreciation of the capabilities of SQL in formulating queries that not only source data from multiple tables, but also provide variances (through filter conditions) and the use of aggregate functions and grouping of rows.

Step 2. Prepare and Execute the Test Script

The query will take some time to execute. Run each query multiple times on different input values. For each run, record the execution time. Identify the cause/s of the delay. Remember that the test script should be efficient and effective (SPSWENG).

Step 3. Optimize the Performance of the Queries

Optimize the performance of the queries in Step 1. Conduct research to identify different approaches to query optimization, which may include (i) redesigning the tables, e.g., splitting tables with > 20 columns to multiple tables, or combining small tables to form larger tables while allowing for possible occurrence of data redundancy; (ii) creating secondary indexes for candidate keys and/or frequently queried column(s); (iii) rewriting query statements based on properties of relational algebra operations, among others. Document the strategies that you employed to optimize the performance of the queries. Redo Step 2. Compare the execution times of the queries based on the original implementation (database design, SQL statements) and the revisions (revisions to the database design, use of indexes, revised SQL statements, and so on). Analyze the causes for the improvement or non-improvement in performance.

NOTE: It is important to highlight that the revised database design and/or query statements may not improve the performance, but instead increase the delay in the query execution time. This is acceptable and will not cause any demerit in grades. The goal is to be able to correctly identify and state the causes for the improvement or non-improvement in performance by conducting a thorough comparative analysis of different strategies.

Step 4. Prepare the Final Report

Using all the data from Steps 1 - 3, write your Technical Report. The outline is provided below.

I. Introduction. Give a brief description of the application that you built. What information did you want to get from the data and for what purpose?

II. Original Queries. Discuss each of the queries. For each query, what is the query title or name? What is the query for? What is the exact SQL statement for the query? What is the expected output? What is the size (in terms of rows and columns) of the query results? How long did the query take to execute under what condition (e.g., what input values did you use)?

III. Query Optimization. Discuss the strategies you employed to try to improve the performance of the queries. Include justification or motivation for the use of these strategies. Citing relevant theories or related works is strongly encouraged to justify your strategy. Use diagrams, tables and examples appropriately to clarify your discussion.

IV. Results and Analysis. Compare the performance of the queries, which include but not limited to different input values, actual SQL statements used, commutative properties of operations (e.g., join), use of grouping functions, database design, availability of indexes, among others. Which performed better? Why? Use tables and figures appropriately to show your comparative analysis. Provide sample query results to clarify your discussion. Remember the NOTE in Step 3.

V. Conclusion. How did the design and structure of the database (normalization, key constraints, indexes) helped in increasing or decreasing the performance? How can queries be formulated to work around the limitations of the database design? How can queries be optimized? How did your findings correlate with theories on query processing and optimization? Cite your references.

VI. References. Reviewing literature on query optimization and database design (e.g., normalization, indexes) should be conducted to help you in writing your paper and doing your analysis.

Criteria for grading the Technical Report (A rubric will be used for grading.):

- Efficiency and Effectiveness of the methodology (Test Process)
- Quality of the formulated queries (based on the criteria specified for the Queries)
- Correctness and Appropriateness of the strategies (e.g., database design, indexes) employed to conduct the comparison of the performance of SQL statements
- Clarity and Thoroughness of the comparative analysis
- Relevance of the discussion and conclusion, with evidence of understanding the impact of the different strategies and formulated SQL statement to query execution time
- Overall document presentation, e.g., format (title page, page numbers, sections, tables and figures), references, and language (spelling, choice of words and grammar)

Final Deliverables

The following final deliverables are required to be submitted:

1. Source Code, specifically the SQL queries
2. Technical Report following ACM publication format (www.acm.org/sigs/publications/pubform.doc)

- Softcopy of the deliverables must be submitted at **12 midnight of February 28** (MW classes) **and 29** (TH classes). Follow the file naming convention: **ADVANDB_<section>_<lastnames of members>.<ext>**
- Printed copies of the deliverables (if required) must be submitted during the first 15 minutes of the class on **February 29** (MW classes) and **March 01** (TH classes).
- Late submissions will receive 10 points deduction per day and 0 points for the presentation. No late submissions will be accepted after March 3, 2016.
- Prepare a Powerpoint presentation to demonstrate your software application and queries, test methodology and results. The actual schedule for class presentation will be provided by your respective teacher.
- **Plagiarized works will automatically be given a grade of 0.0 for the course.**