

RMIT University
COSC2406/2407 – Database Systems
Assignment #2: MongoDB, Apache Derby, Java
Due: 11.59pm on Sunday 23 May 2021

Marks: This individual assignment is worth 45 (260 points) of your overall mark

Introduction

This assignment builds on assignment 1 using the same open data from the City of Melbourne about pedestrian traffic in the Melbourne CBD: <https://data.melbourne.vic.gov.au/Transport/Pedestrian-Counting-System-Monthly-counts-per-hour/b2ak-trbp>.

You are reminded to only run one database system at a time and to ensure all systems are shut down when you log out of the system. This is to avoid problems arising from running out of memory. Other precautions have been provided via announcements on Canvas. Please note that ignoring taking such precautions will not be grounds for extensions.

Assignment Project Exam Help

Task 1: Experiment with and without using a secondary index in Derby

<https://powcoder.com>

In this part of the assignment, using the Derby database you built as part of assignment 1 (or a variation of it), add a secondary index on one (or more) fields in the database. Design four new queries (different from the queries used in assignment 1). Two of these queries should be queries that use the secondary index and two should be queries that do not use the secondary index. Running the queries multiple times (at least twice, once immediately after a reboot, and then again) on two versions of the database (before and after adding the secondary index), and compare the performance of the database.

Add WeChat powcoder

You need to show through the selection of your queries and analysis an understanding of the different performance requirements. For example, the amount of time it takes to compare different types of fields takes different amounts of time, different queries may require a full file scan or an optimised approach may be possible. You should also consider in your choice of queries the performance properties we would expect for the database systems under consideration due to their representation of the data.

Task 2: Compare with MongoDB with Derby

Repeat the four queries from Task 1 (above) using MongoDB instead of Derby, and compare the results.

Task 3: Implement a B+-tree Index in Java

Implement a B+-tree Index structure in Java for your heap file from Assignment 1 and conduct experiments.

General Requirements

This section contains information about the general requirements that your assignment must meet.

Please read all requirements carefully before you start.

1. The “Database Systems” canvas shell contains further announcements and a list of frequently asked questions. You are expected to check the discussion board on daily basis. Login through <https://rmit.instructure.com>.
2. Your database and Java programs must be set up and run on your AWS linux machine using the same data as in assignment 1.
3. As some tasks require timing you should use the same AWS linux machine for all tasks.
4. You must implement your program in Java. Your program must be well written, using good coding style and including appropriate use of comments (that clearly identify the changes you are making to the code). Your markers will look at your source code. Coding style will form part of the assessment of this assignment.
5. If your marker cannot compile your programs, you risk yielding zero marks for the coding component of your assignment.
6. Your program may be developed on any machine, but must compile and run on your AWS linux instance. In particular your code must comply with the java 1.8 standard as that's the version of java that we have gotten you to install there.
7. You must use git as you develop your code (wherever you do the development). As you work on the assignment you should commit your changes to git regularly (for example, hourly or each time you rebuild) as the log may be used as evidence of your progress.
8. Paths must not be hard coded. That is, the program should not require the input files to be in a specific directory - your marker may load the data from any directory and your program should work correctly.
9. Diagnostic messages must be output to `stderr`.
10. Parts of this assignment will ask you to analyse your results, and to write about your conclusions in a report. Your report must be a PDF file, called REPORTyyyyyyy.pdf where yyyyyyy is your student number. Files that do not meet this requirement may not be marked.
11. Your report must be well-written and properly formatted. Poorly written or hard to read reports will receive substantially lower marks. Your report should be appropriate to submit in a professional environment (such as including in a portfolio of your work for a prospective employer). The RMIT Study & Learning Centre employs advisors to help you improve your writing. For details, see <http://www.rmit.edu.au/studyandlearningcentre>.
12. All sections of this assignment are expected to show that you have thought about the problem. The most basic structuring of data and analysis will get the most basic mark.
13. Take care to repeat timings in a consistent way, so that you can make fair comparisons.

14. Depending on your implementation, you may wish to provide additional information about your code (for example, how it is to be compiled and run). If so, put this information into a plain text file called `readme.txt`.
15. Important: You must run all your experiments on your AWS linux instance.
16. Canvas for COSC2406/COSC2407 Database Systems contains a discussion board for this assignment allowing a forum for students to ask questions (see below) and contribute to discussion about aspects of the assignment. If there are announcements about the assignment (including if there are any revisions to the assignment specification) these will also be made via announcements on Canvas. You are expected to check these on a daily basis. Login through <https://rmit.instructure.com>.
17. If you have any questions about the assignment (for example to clarify requirements):
 - (a) Please first check this assignment specification, as well the announcements and the discussion board on canvas to see if it has already been answered.
 - (b) If it has NOT already been answered and does NOT include your own code (including database queries), please post your question on the discussion board.
 - (c) Otherwise, if your question involves your own code (or is about your personal situation) then discuss it in your practical class with the lab instructor or contact the lecturer (or your tutor) via email.
18. You must include the final code from the assignment 1 heap implementation that you used for the assignment 2 heap code in a subfolder, so that we can recreate the heap file.

Assignment Project Exam Help

<https://powcoder.com>

Academic Integrity Add WeChat powcoder

This is individual assignment, which means you can complete it by yourself, and what you submit MUST be your own original work.

So make sure you reference any sources you use (including all web resources) as all assignments will be checked with plagiarism-detection software.

Any student found to have plagiarised will be subject to disciplinary action in accordance with RMIT policy and procedures. Plagiarism includes submitting code that is not your own or submitting text that is not your own. Submitting a comment from someone else in your code or a sentence from someone else's report is plagiarism, and plagiarism includes submitting work from previous years. Allowing others to copy your work is also plagiarism. All plagiarism will be penalised; there are no exceptions and no excuses. For further information, please see: <https://www.rmit.edu.au/students/student-essentials/rights-and-responsibilities/academic-integrity>.

Assessment tasks, weightings and marking criteria

Task 1: Experiment with and without using a secondary index in Derby

Report on your Experiment 1 (60 points)

You are required to write a report on the experiments undertaken using your new queries and discuss the output and timings of queries using Derby with and without using a secondary index.

Task 2: Compare with MongoDB with Derby

Report on your Experiment 2 (60 points)

You are required to write a report on the experiments undertaken using your new queries and discuss the output and timings of queries using Derby and MongoDB.

Task 3: Implement a B+-tree Index in Java

Implement a B+-tree Index in Java for your heap file from Assignment 1 and conduct experiments querying (equality query and range query) with and without the index.

Submission of code (80 points)

You must submit all files that you have modified, including your git log. In your report (in no more than one or two pages), you should explain how you implemented a B+-tree index for your heap file. In particular, for each file make sure you explain any choices you made in your implementation. Also identify any known limitations of your implementation.

Results of your Experiment 3 (60 points)

Undertake experiments using your program and report on the output and timings. In no more than one or two pages, discuss your results and critically analyse the effectiveness of using your B+-tree Index on your heap file. Are the results as you expected?

Important: Your report will be marked on the quality of your written explanations and analysis, and not on the length of the report (the page limits are meant as guidelines only). After writing your report you should carefully revise it checking for clarity of expression and quality of writing.

What to Submit, When, and How

What

You need to submit your source code of any files modified, including git log, and a report. Before you submit anything, read through the assignment specifications again carefully. Check that you have followed all instructions in the general requirements. Also check that you have attempted all parts of all questions. In particular you must submit:

1. your report (a single PDF file) that explains queries used, how your code implements a B+-tree index, output of the queries, and a discussion of your results in the experiments; and
2. a zip file of your code (all files that you have modified and including your git log).
3. You must set up git so that all commits are listed with your student email address and your name as listed on the student roll.

When

The assignment is due at 11.59pm on Sunday 23 May 2024.

Late submissions should be submitted using the same procedure. If you unable submit by the due date you must have an extension approved (follow the process at <http://www1.rmit.edu.au/students/assessment/extension>) otherwise you will be penalised by 10% of total possible marks per day for assignments that are late 1 to 5 days late. For assignments that are more than 5 days late, a penalty of 100% will apply. See the course guide for further information. The onus is on you to check that your submission has been received.

Add WeChat powcoder

How

You need to separately submit two files under assessment tasks on canvas via MyRMIT

1. ONE zip file that contains the Java source files you have modified, and your git log, this should be submitted using the link to Assignment 2 Code Submission, and
2. ONE PDF file containing your report, this should be submitted using the link to Assignment 2 Report Submission (it is a turnitin submission).