RMIT University
COSC2406/2407 – Database Systems
Assignment #1

Creating Derby and MongoDB databases and implementing a heap file in Java

Due: 11.59pm on Sunday 4 April 2021

Marks: This **individual** assignment is worth **20%** of your overall mark

1 Introduction

The goal of the assignment work in this course is to gain practical experience that helps you to:

- explain data structures and algorithms used to efficiently store and retrieve information in database systems;
- investigate alternative approaches for design of database systems (both relational and non-relational); and
- design and implement (using Java) file structures and indexing schemes.

The aim of the first assignment is to start using the **AWS linux instance** assigned to you and the **open data** provided from a public source to complete the following tasks:

1. store and retrie at you create,
2. store and retrie

3. store and retrieve data in a heap file that you implement usi .

In the second assignment, you will extend your solution dev t and conduct further timing experiments on your AWS linux insta these approaches.

Please read ALL the following requirements carefully before you start. What you submit should be of a standard that would be acceptable in a workplace context and poorer submissions may attract deductions.

2 Data

The data that you are going to use in this assignment is open data from the City of Melbourne about pedestrian traffic in the Melbourne CBD (download from https://data.melbourne.vic.gov.au/Transport/Pedestrian-Counting-System-Monthly-counts-per-hour/b2ak-trbp). please refer to the help file available that describes the data set from the provided link.

3 Academic Integrity

This is an **individual assignment**, which means 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.

4 Tasks and Analysis

In this section, you will be asked to carry out several tasks using your AWS linux instance to create and query a database (using both MongoDB and Derby), with the data in the provided file and to analyse your results.

Create a file called `report.pdf` (various software including word processors can export as PDF). Use this file to report on the following tasks. Each task should be reported under a separate heading with the task name and description, for example for the first task use the heading: **Task 1: Derby**.

Task 1: Derby

You are required to l

- explain how ive reasons.

- provide details of the time to load the data into Derby. Y consider appropriate ways to structure the data and t gramming or other tools to format the data accordingly.

- **Postgraduate students only:** *What alternative way or ways could you have organised the data when storing in Derby, and what advantages or disadvantages would these alternative designs have?*

Task 2: MongoDB

You are required to load the data into MongoDB. In your report:

- explain how have you chosen to structure the data inserted in MongoDB

- provide details of the time taken to load the data (The `mongoimport` is one utility will provide such information). Please note that a naive import into a flat structure in Mongodb will not accrue you a great mark. You need to analyse the data and consider appropriate ways to structure the data and then using any scripting, programming or other tools to format the data accordingly.

- **Postgraduate students only:** *What alternative way or ways could you have organised the data when storing in MongoDB, and what advantages or disadvantages would these alternative designs have?*

Task 3: Implement Heap File in Java

Set up a git repository for your code, and complete the following programming tasks using Java on the AWS linux instance assigned to you.

4.0.1 Explain your design

This should include justification for using fixed length vs variable length fields and how they were implemented (eg delimiters, headers etc). For postgraduates, you should discuss alternatives and their advantages and disadvantages like you do for task 1 and 2.

A program to load a database relation writing a heap file

The source records are variable-length. Your heap file may hold fixed-length records (you will need to choose appropriate maximum lengths for each field). However, you may choose to implement variable lengths for some fields, especially if you run out of disc space or secondary memory!

All attributes with *Int* type must be stored in 4 bytes of binary, e.g. if the value of ID is equal to 70, it must be stored as 70 (in decimal) or 46 (in hexadecimal; in Java: `0x46`). It must not be stored as the string "`70`", occupying two bytes. Your heap file is therefore a binary file.

For simplicity, the heap file does not need a header (containing things like the number of records in the file or a free space list), though you might need to keep a count of records in each page. The file should be packed, i.e. there is no gap between records, but there will need to be gaps at the end of each page.

The executab[illegible] `dbload` and should be executed using the c[illegible]

```
java dbload -p pagesize datafile
```

The output file will be `heap.pagesize` where you [illegible] ten as a heap.

Your program should write out one "page" of the file at a time. For example, with a `pagesize` of 4096, you would write out a page of 4096 bytes possibly containing multiple records of data to disk at a time. You are not required to implement spanning of records across multiple pages.

Your `dbload` program must also output the following to `stdout`, the number of records loaded, number of pages used and the number of milliseconds to create the heap file.

You are also suggested the use of utilities like xxd for examining the output heap file to see if their code is producing the expected format. ie

```
xxd heap.pagesize | less
```

A program that performs a text search using your heap file

Write a program to perform text query search operations on the field "SDT_NAME" `heap` file (without an index) produced by your `dbload` program in Section 4.0.1. Note that SDT_NAME is a new field you created by considering Sensor_ID and Date_Time as strings and connecting them together.

The executable name of your program to build a heap file must be `dbquery` and should be executed using the command:

```
java dbquery text pagesize
```

Your program should read in the file, one “page” at a time. For example, if the pagesize parameter is 4096, your program should read in the records in the first page in heap.4096 from disk. These can then be scanned, in-memory, for a match (the string in text parameter is contained in the field “SDT_NAME”). If a match is found, print the matching record to stdout, there may be multiple answers. Then read in the next page of records from the file. The process should continue until there are no more records in the file to process.

In addition, the program must always output the total time taken to do all the search operations in milliseconds to stdout.

5 General Requirements and Getting Help

This section contains information about the general requirements that your assignment must meet and how to get help.

1. Your database and Java programs must be set up and run on the AWS linux machine assigned to you for this course.

2. Your database must be set up on your AWS linux instance (as set up following the instructions in the initial practical classes in the laboratories).

3. You must implement your program in Java 1.8. 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 styl

4. If your marker c r the coding componen

5. Your Java program may be developed on any machine, b and run your AWS linux instance. You need to test your solutions on will otherwise be deductions for problems that arise during that testing for markers.

6. 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.

7. Paths must not be hard-coded.

8. Diagnostic messages must be output to stderr.

9. Parts of this assignment will ask you to analyse your results, and to write your conclusions in a report. The report MUST be a PDF file. Submissions that do not meet this requirement will NOT be marked.

10. Your report must be **well-written**. 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.

11. 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.

12. 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://my.rmit.edu.au.

13. 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.

6 Submission

Before you submit anything, read through the assignment specifications again carefully, **especially** Section 5. Check that you have followed ALL instructions. Also check that you have attempted all parts of all tasks in Section 4.

When

The assignment is d

What

You MUST submit:

1. your report (a single PDF file) that explains your approach and answers for each task (1, 2 and 3) and includes any scripts, queries you used, and output; and

2. a zip file of your code for task 3(all Java sources files including your git log)

Note: Please do not submit your scripts in the pdf as that's useless. You must submit all scripts in the zip file with your code as that will make it much easier for the markers to test your scripts. For the git log, you need to set up your git repo so that each commit identifies you with your full name as per course enrolment and your student email address. It sets an expectation of professionalism.

How

You need to submit your report in one PDF file using the link under "Assesments" on the course blackboard through myRMIT by 11.59pm on Sunday 4 April 2021.

Late submissions should be submitted using the same Blackboard procedure, but 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.

You should ensure that your score from the turnitin similarity checker is in the **green** range. Any greater similarity (**yellow** , **orange** or **red**) will be flagged for closer inspection.

7 Marking Criteria and Weighting

Marking criteria will include: (i) appropriate design of databases, (ii) correctness of scripts, queries, and explanations (iii) completeness of results, (iv) clarity and quality of justifications and explanations (v) depth of critical analysis.

Task 1: Derby 25 points

- scripts for shaping data
- justification for scripts and explanation of chosen design
- explanation and analysis of alternative designs
- queries for Derby

Task 2: MongoDB 25 points

- scripts for shaping data
- justification for scripts and explanation of chosen design
- explanation and analysis of alternative designs

Task 3: Heap file in Java 50 points

- implementation of heap file and text search
- queries for Java