Database Systems
COSC 2406/2407
Assignment 2

Assessment Type	Individual assignment. Submit online via Canvas→Assignments→Assignment 2. Marks awarded for meeting requirements as closely as possible. Clarifications/updates may be made via announcements/relevant discussion forums.
Due Date	Week 12, Tuesday 24 May 2022, 11:59pm
Marks	100 points (45% of the overall assessment)

1. Overview

You will use the AWS Linux instance assigned to you and the dataset provided from a public source to complete the following tasks:

1. Implement in Java a B+-tree index for your heap file;
2. Implement in Java a query that uses the index; and
3. Create an index in an Apache Derby database and query the database using the index.

In this second assignment, you will extend your solution developed in the first assignment and conduct further timing experiments on your AWS Linux instance with indexes.

2. Learning Outcomes

This assessment relates to the following learning outcomes of the course:

- CLO 1: Explain and critique data structures and algorithms used to efficiently store and retrieve information in database systems,
- CLO 2: Evaluate, critically analyse and compare alternative designs for implementation of database systems, including data models, file structures, index schemes, and query evaluation, and
- CLO 4: Design, implement and report on significant software components of a database system (such as file structures and index schemes) according to analysis of requirements and specified constraints.

3. Submission

When you submit work electronically, you agree to the assessment declaration:

https://www.rmit.edu.au/students/student-essentials/assessment-and-exams/assessment/assessment-declaration

Submit on Canvas: Assignments > Assignment 2. You MUST submit:

- a zip file of your code for tasks 1 and 2 (all Java sources files including your git log); and
- your report (a single PDF file) that explains your approach and answers for each task (1, 2 and 3), including description of any scripts for data pre-processing, queries you used, and output.

Progress submission of your Java code in Week 11 You must make a progress submission during a scheduled lab class in week 11. Failing to do this result in a penalty of 10 points.

Late submission:

- After the due time, you will have 7*24 hours to submit your assignment as a late submission. Late submissions follow the same procedure but will be penalised by 10 points for each (up to) 24 hours being late. For assignments that are more than 7*24 hours late, zero points will be awarded.
- Tasks (Section 4) in this assignment require you run database systems and timing experiments in a cloud(AWS) Linux instance. All experiments and database systems must be shut down before you log out of your Linux instance. Otherwise your instance may run out of memory. Other precautions are provided via announcements on Canvas. *Ignoring such precautions will NOT be grounds for extension to due time.*

4. Academic integrity and plagiarism (standard warning)

Academic integrity is about honest presentation of your academic work. It means acknowledging the work of others while developing your own insights, knowledge and ideas. You should take extreme care that you have:

- Acknowledged words, data, diagrams, models, frameworks and/or ideas of others you have quoted (i.e. directly copied), summarised, paraphrased, discussed or mentioned in your assessment through the appropriate referencing methods,
- Provided a reference list of the publication details so your reader can locate the source if necessary. This includes material taken from Internet sites.

If you do not acknowledge the sources of your material, you may be accused of plagiarism because you have passed off the work and ideas of another person without appropriate referencing, as if they were your own.

RMIT University treats plagiarism as a very serious offence constituting misconduct. Plagiarism covers a variety of inappropriate behaviours, including:

- Failure to properly document a source
- Copyright material from the internet or databases
- Collusion between students

For further information on our policies and procedures, please refer to https://www.rmit.edu.au/students/student-essentials/rights-and-responsibilities/academic-integrity

5. Marking Guidelines

- B+-tree implementation 50/100
- Range query implementation 30/100
- Derby indexing 20/100

Assessment details

This assignment builds on Assignment 1 using the same data source (refer to Assignment 1 for details about the data source).

Run database systems and timing experiments

Timing experiments are only reliable when running one program at a time. You are reminded to only run one database system, Derby or MongoDB, at a time and to ensure that you shut down one database system before starting the other. All database systems must be shut down before you run your own program and before you log out of your Linux instance. Otherwise your instance may run 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.

Progress submission of your Java code in Week 11 You must make a progress submission during a scheduled lab class in week 11. Failing to do this result in a penalty of 10 points.

Code and git log: You must use git to track your assignment code. You need to set up your git repository 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. You must submit a text file of your git log. Do not include the git repository in your code submission.

Report: Create a file called report.pdf (various software including Word processors can export as PDF). Use this file to report on the following three 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: B+-tree. *Limit your report to three pages.*

Task 1: Build a B+-tree index in Java (50 points).

Write a program that builds a B+-tree index on the “birthDate” field that indexes the heap file heap.pagesize from Assignment 1. You can assume a fixed length for each index entry and your program should use the same pagesize as used to generate the heap file. Where there are records in the heap file with a NULL value for “birthDate” and these records should not be indexed. You should not assume the records are sorted, and should repeatably call a function that inserts a single entry into the B+-tree (readjusting the tree as required). It is NOT required to write a delete function.
The output file, containing the B+-tree index, should be named index.pagesize. Your program must also output the following to stdout: the number of records indexed (which excludes records with NULL values for “birthDate”); the number of index pages created; the height of the B+-tree; and, the number of milliseconds to create the index file.

The executable name of your program must be **btindex** and should be run via the following command:
```
java btindex -p <pagesize> <heapfile>
```

In your report, explain your b+ tree design and implementation. Give detailed instructions to run and test your program.

Task 2: Implement the range query using the B+-tree index (30 points).

Write a program to perform a range query on the field “birthDate” using the B+-tree index for the heap file from your Assignment 1. Your program may load the entire index file into memory, but should only load pages from the heap file containing answers. If a match is found, print the matching record to stdout. There may be multiple answers. The process should continue until there are no more matching entries in the index to process. In addition, the program must always output the total time taken to do all the search operations in milliseconds to stdout.

The executable name of your program must be **btsearch** and should be run via the following command:
```
java btsearch <heapfile> <indexfile> <start date> <end date>
```

Use the same queries from Assignment 1 for comparison. For example, for the query “List all artists born in 1970”. The command below:
```
java btsearch heapfile.4096 index.4096 19700101 19701231
```

In your report, explain how your range query implementation using the B+-tree index. Give detailed instructions to run and test your program. Report on the performance of the range search using the index compared to the range search without an index (Assignment 1).

Task 3. Derby indexing (20 points).

Relational database systems by default build an index for the primary key of tables. Using the database you built as part of your Assignment 1, create an index on the field "birthDate". Write three range queries that use the index on "birthDate". Run the queries at least twice on two versions of the database (before and after the "birthDate" index), and record and compare the runtime of the queries.

Run the same three queries using your program from Task 2 and record the runtime. Compare the runtime of your program against that of Derby for the three queries.

In your report, you should include

- Your design for selecting the three queries to show what types of queries would benefit from the "birthDate" index and the experiment setting for comparing performance;
- The three queries and description of query output, rather than a long list of output;
- A table of timing results for the performance of Derby and your program; and
- Analysis of the results in the table in terms of the performance for your program and Derby. Explain possible reasons for any performance difference.