DSAA 5012: ADVANCED Database Management FOR DATA SCIENCE

Lecture 12 Exercises
Indexing: Introduction

A movie database has the following files and sizes of each field.

Film(title: 40 bytes, director: 20 bytes, releaseYear: 4 bytes, company: 20 bytes)
Actor(id: 4 bytes, name: 20 bytes, dateOfBirth: 4 bytes)

There are 30,000 film and 100,000 actor records. Each page is 512 bytes. Each pointer is 6 bytes.

Exercise 1: For the movie database:

a) What is the blocking factor bf_F for the Film file and bf_A for the Actor file?

bf_F = bf_A =

b) Assuming the Film file is ordered on title and there is no index, what is the page I/O cost for:

i. Finding the film with title "Titanic"?

Page I/O cost:

ii. Finding all the films directed by "John Woo"?

Page I/O cost:

Exercise 2: Assume the Actor file is ordered on name and we want to create an ordered index on id (4 bytes) where each index entry has the form <id, *pointer*>.

a) What is bf_{Aindex} if the index is single-level?

bf_{Aindex} =

b) How many index entries are needed? (***Briefly explain your answer.)***

Index entries:

Explanation:

c) How many pages are required for the Actor index entries?

Pages needed:

d) What is the page I/O cost to retrieve a single id value using the Actor index (e.g., "Find the actor with id 100")?

Page I/O cost:

e) If the single-level index is converted into a multi-level index, how many levels are needed (assuming full pages)? (***Briefly explain your answer.***)

Index levels:

Explanation:

f) Using the multi-level index, what is the cost of answering the query "Find the actor with id 100"? (***Briefly explain your answer.***)

Page I/O cost:

Explanation:

Do not upload this exercise worksheet to Canvas.

Name: _____/_____ **Student#:** _____ **Date:** _____
Family/Last (PRINT) Given/First (PRINT)

DSAA 5012: ADVANCED Database Management FOR DATA SCIENCE

Lecture 12 Exercises
Indexing: Introduction

Show your calculations for all questions.

Exercise 3: A company database has the following file and sizes of each field.

Employee(employeeId: 6 bytes, employeeName: 10 bytes, departmentId: 4 bytes)

where departmentId is the id of the department where the employee works.

There are 100,000 employee records. There are 1,000 departments (each department has 100 employees). A page is 1,000 bytes. A pointer is 4 bytes. Assume the file is *ordered on departmentId* and there is *no index*.

a) What is the blocking factor for the Employee file?

$bf_{Employee}$:

b) How many pages are needed to store the Employee file?

Pages needed:

c) What is the page I/O cost for retrieving the records of all employees working in a department with a given departmentId (e.g., departmentId=64)? (***Briefly explain your answer.)***

Page I/O cost:

Explanation:

Exercise 4: For the Employee file of Exercise 3, assume we add a *single-level ordered index on employeeId* (6 bytes) where each entry has the form <employeeId, *pointer*> and *the number of pointers is the same as the number of search keys*.

a) How many index entries are needed? (***Briefly explain your answer.)***

Number of index entries:

Explanation:

b) How many pages are required for these index entries?

Index pages:

c) What is the page I/O cost of retrieving the record of an employee with a given employeeId?

Page I/O cost:

d) If the single-level index is converted into a *multi-level index*, how many index levels are needed (assuming full pages)? (***Briefly explain your answer.***)

Levels needed:

Explanation:

Upload this completed exercise worksheet to Canvas by **11 p.m. on March 12th.**