COMP 3311: Database Management Systems

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 exist 30,000 films in the database and 100,000 actors.
Each page is 512 bytes and 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?

b) Assuming that the Film file is ordered on title and there is no index, what is the cost (in terms of page I/Os) for:

i. Finding the film with title "Titanic”? (Briefly explain your answer.)

ii. Finding all the films directed by “John Woo”? (Briefly explain your answer.)

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

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

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

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

d) What is the cost of retrieval based on a single id value using the Actor index (e.g., “Find the actor with id 100”)?

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

f) Using the multi-level index, what is the cost of answering the query “Find the actor with id 100”?

Name: ______________________/________________ **Student#:** ___________________ **Date:** __________

Family/Last (PRINT) Given/First (PRINT)

COMP 3311: Database Management Systems

Lecture 12 Exercises
Indexing: Introduction

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

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

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

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

Exercise 4: For the Employee file of Exercise 3, assume that we add a *single-level ordered index on employeeId* 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.)

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

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

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

You must upload this completed exercise page to Canvas by **2:30 p.m. today.**