

National University of Computer and Emerging Sciences, Lahore Campus



Course:	Advance Database Systems	Course Code:	CS451
Program:	BS(Computer Science)	Semester:	Spring 2018
Out Date:	28-March-2018	Total Marks:	50
Due Date:	Tue 03-April-2018 (start of class)	Weight:	
Section	CS	Page(s):	2
Assignment:	3 (Indexing)		

Instructions:

- Use proper assignment papers for solving your assignment questions. Assignment done on diary pages, register pages, rough pages will not be credited.
- Do not copy the work of your peers. In case cheating is detected, then your case will be referred to DC

Take the following assumptions for the block size and file size to solve the questions:

- Q1.** Block Size **B=4096** bytes and File Records **r=10million**
Q2. Block Size **B=4096** bytes and File Records **r=10billion**
Q3. Block Size **B=8192** bytes and File Records **r=10million**
Q4. Block Size **B=8192** bytes and File Records **r=10billion**

A block pointer is $P = 6$ bytes long and a record pointer is $P_R = 7$ bytes long. A file has above records of fixed length. Each record has the following fields: NAME (30 bytes), SSN (9 bytes), DEPARTMENTCODE (9 bytes), ADDRESS (40 bytes), PHONE (9 bytes), BIRTHDATE (8 bytes), SEX(1 byte), JOBCODE (4 bytes), SALARY (4 bytes, real number). An additional byte is used as a deletion marker.

- Calculate the record size R in bytes.
- Calculate the blocking factor bfr and the number of file blocks b , assuming an unspanned organization.
- Suppose that the file is *ordered* by the key field SSN and we want to construct a *primary* index on SSN. Calculate (i) the index blocking factor $bfri$ (which is also the index fan-out fa); (ii) the number of first-level index entries and the number of first-level index blocks; (iii) the number of levels needed if we make it into a multilevel index; (iv) the total number of blocks required by the multilevel index; and (v) the number of block accesses needed to search for and retrieve a record from the file-given its SSN value-using the primary index.
- Suppose that the file is not *ordered* by the key field SSN and we want to construct a *secondary* index on SSN. Repeat the previous exercise (part c) for the secondary index and compare with the primary index.
- Suppose that the file is not *ordered* by the nonkey field DEPARTMENTCODE and we want to construct a *secondary* index on DEPARTMENTCODE, with an extra level of indirection that stores record pointers. Assume there are 20000 distinct values of DEPARTMENTCODE and that the EMPLOYEE records are evenly distributed among these values. Calculate (i) the index blocking factor bfr , (which is also the index fan-out fa); (ii) the number of blocks needed by the level of indirection that stores record pointers; (iii) the number of firstlevel index entries and the number of first-level index blocks; (iv) the number of levels needed if we make it into a multilevel index; (v) the total number of blocks required by the multilevel index and the blocks used in the extra level of indirection; and (vi) the approximate number of block accesses needed to search for and retrieve all records in the file that have a specific DEPARTMENTCODE value, using the index.
- Suppose that the file is *ordered* by the nonkey field DEPARTMENTCODE and we want to construct a *clustering index* on DEPARTMENTCODE that uses block anchors (every new value of DEPARTMENTCODE starts at the beginning of a new block). Assume there are 20000 distinct values of DEPARTMENTCODE and

that the EMPLOYEE records are evenly distributed among these values. Calculate (i) the index blocking factor bfr, (which is also the index fan-out fa); (ii) the number of first-level index entries and the number of first-level index blocks; (iii) the number of levels needed if we make it into a multilevel index; (iv) the total number of blocks required by the multilevel index; and (v) the number of block accesses needed to search for and retrieve all records in the file that have a specific DEPARTMENTCODE value, using the clustering index (assume that multiple blocks in a cluster are contiguous)

- g.** Suppose the file is not ordered by the key field Ssn and we want to construct a B⁺-tree access structure (index) on SSN. Calculate (i) the orders p and p leaf of the B⁺-tree; (ii) the number of leaf-level blocks needed if blocks are approximately 69% full (rounded up for convenience); (iii) the number of levels needed if internal nodes are also 69% full (rounded up for convenience); (iv) the total number of blocks required by the B⁺-tree; and (v) the number of block accesses needed to search for and retrieve a record from the file--given its SSN value--using the B⁺-tree.
- h.** Repeat part g, but for a B-tree *rather than for a B⁺-tree*. Compare your results for the B-tree and for the B⁺-tree.

Q 5: B⁺-tree and B-tree (10 points)

Consider a DBMS that has the following characteristics:

- 2KB fixed-size blocks
- 12-byte pointers
- 56-byte block headers

We want to build an index on a search key that is 8 bytes long. Calculate the maximum number of records we can index with

- a) a 3-level B⁺ tree (including the root level)
- b) a 3-level B tree (including the root level)