

Problems for practice in Module-5

1. Write execution plan for the following query after making the optimized tree.
`SELECT E.ENAME , COUNT(P.SSN), D.NAME, COUNT(N.SSN) FROM
EMPLOYEE E, DEPENDENT N, DEPARTMENT D, WORKS_ON P WHERE
E.SSN=N.SSNAND E.DNO=D.DNO AND E.SSN=P.SSN GROUP BY E.ENAME,
D.NAME WHERE D.NUM=9 OR D.NUM=10;`
2. Consider a disk with block size $B = 512$ bytes. A block pointer is $P = 6$ bytes long, and a record pointer is $PR = 7$ bytes long. A file has $r = 30,000$ EMPLOYEE records of *fixed length*. Each record has the following fields: Name (30 bytes), Ssn (9 bytes), Department_code (9 bytes), Address (40 bytes), Phone (10 bytes), Birth_date (8 bytes), Sex (1 byte), Job_code (4 bytes), and Salary (4 bytes, real number). An additional byte is used as a deletion marker.
 - a. Calculate the record size R in bytes.
 - b. Calculate the blocking factor bfr and the number of file blocks b , assuming an unspanned organization.
 - c. 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 fo); (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.
 - d. 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.
 - e. Suppose that the file is *not ordered* by the nonkey field Department_code and we want to construct a *secondary index* on Department_code. Assume there are 1,000 distinct values of Department_code and that the EMPLOYEE records are evenly distributed among these values. Calculate (i) the index blocking factor $bfri$ (which is also the index fan-out fo); (ii) the number of blocks needed by the level of indirection that stores record pointers; (iii) the number of first-level 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 Department_code value, using the index.
 - f. Suppose that the file is *ordered* by the nonkey field Department_code and we want to construct a *clustering index* on Department_code that uses block anchors (every new value of Department_code starts at the beginning of a new block). Assume there are 1,000 distinct values of Department_code and that the EMPLOYEE records are evenly distributed among these values. Calculate (i) the index blocking factor $bfri$ (which is also the index fan-out fo); (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 Department_code value, using the clustering index (assume that multiple blocks in a cluster are contiguous).

3. Rewrite the following queries and write the reasons behind tuning.

- a. **SELECT** Fname, Lname, Salary, Age7
FROM EMPLOYEE
WHERE Age > 45 **OR** Salary < 50000;
- b. **SELECT** Region#, Prod_type, Month, Sales
FROM SALES_STATISTICS
WHERE Region# = 3 **AND** ((Prod_type **BETWEEN** 1 **AND** 3) **OR** (Prod_type **BETWEEN** 8 **AND** 10));
- c. **SELECT** Ssn
FROM EMPLOYEE
WHERE Dno **IN** (**SELECT** Dnumber **FROM** DEPARTMENT
WHERE Mgr_ssn = '333445555');
- d. **SELECT** distinct SSN
FROM EMPLOYEE
WHERE dno between 11 and 30;
- e. **SELECT** NAME
FROM EMPLOYEE E, DEPARTMENT D
WHERE E.SSN=D.MGRSSN;