# CIS/CSC384

HomeWork 7: Due Apr 20, 2010 before class
You will turn in the assignment via bb. Turn in a single word/pdf document.

1. Use the following data about a table to answer the questions below. [2 pts]

   Row size = 100 bytes
   Number of rows = 100,000
   Physical Record Size = 4,096 bytes
   Primary Key size = 6 bytes
   Pointer size = 4 bytes

   (a) How many rows (logical records) can be fit into a physical record? Assume that only complete rows are stored in a physical record.

   (b) What is the number of physical records needed for a sequential file?

   (c) If an unordered sequential file is used, calculate the number of physical records that need to be accessed to find a key (find rows with specified key values) that does not exist in the table.

   (d) In a node in the B+ tree we can hold x key values and (x + 1) pointers. Suppose we build a B+ tree index on the primary key (key size = 6 bytes), what is the maximum possible value for x?

2. Consider the following query:
   SELECT * FROM Customer
   WHERE CustBalance > 5000 AND CustState='CO'

   Suppose that the total number of rows in the Customer table is 7240. The CustBalance ranges from 0 – 10000. [1 pt]

   (a) Using the uniform distribution assumption, state how many rows are expected to satisfy the condition CustBalance > 5000 (see that the condition CustState='CO' is not used).

   (b) Use the following histogram to estimate the number of rows expected to satisfy the condition CustBalance > 5000.

   Histogram for CustBalance

   | Range | Number of Rows |
   |---|---|
   | 0 – 100 | 1000 |
   | 101 – 250 | 950 |
   | 251 – 500 | 1050 |
   | 501 – 1000 | 1030 |
   | 1001 – 2000 | 975 |
   | 2001 – 4500 | 1035 |
   | 4501 – 10000 | 1200 |

3. Consider the following query:
   SELECT OrdNo, OrdDate, Vechicle.ModelNo
   FROM Customer, Order, Vehicle
   WHERE CustBalance > 5000 AND Customer.CustNo = Vehicle.CustNo
         AND Vehicle.SerialNo = Order.SerialNo

   Draw the access plan showing the following:

   You access Customer using a Btree on CustNo, you sort vehicle on CustNo
   Now you join Customer and Vehicle using Sort Merge Join (the Btree on CustNo gives you the customers sorted by CustNo, and we sorted vehicles on CustNo)
   You sort order by serialNo, and you sort the result of the above Sort Merge Join on serialNo. Now you join the order table with the result of the first sort merge join using another sort merge join. [1 pt]

4. For the following indexes, and the query conditions, indicate if the index matches the condition. [3 pts]

   (a) B tree Index on TrdDate. Query Condition: TrdDate BETWEEN '1-Oct-2006' AND '31-Oct-2006'

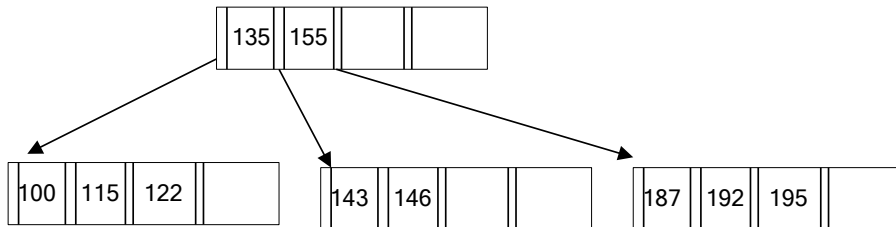   (b) B-tree Index on CustPhone. Query Condition: CustPhone LIKE '(810)%'

   (c) B-tree Index on TrdType. Query Condition: TrdType <> 'BUY'

   (d) BITMAP Index on BondRating. Query Condition: BondRating IN ('AAA', 'AA', 'A')

   (e) B-tree Index on <CustState, CustCity, CustZip>. Query Condition: CustState ='CO' AND CustCity = 'Denver'

   (f) B-tree Index on <CustState, CustCity, CustZip>. Query Condition: CustState IN ('CO', 'CA') AND CustZip LIKE '8%'

5. Consider the B tree shown below.

   

   You are going to insert the following keys. Show the B tree after each insertion. [2 pts]
   142, 111, 134, 170, 175, 127, 137, 108, 140 (Note that first you insert 142, then to the B tree resulting after inserting 142, you insert 111, to the resulting B tree you insert 134 and so on..)

6. Consider the B tree resulting from the problem above. Delete the following keys one after the other and show the B tree after each deletion. Keys to be deleted: 111, 108, 137. [1 pt]