DATABASE MANAGEMENT SYSTEMS, IIITS

Assignment 2

Marks 20

Instructions:

1. Submit a hard copy (written) assignment with clearly the following written on top of the sheet: Name, Roll No, Section.
2. The assignments are required to be submitted during class hours to the instructor (26-11-2019). No further extensions would be granted.
3. Hard Deadline: 26-11-2019, Tuesday

Questions:

1. Consider the following relation: 3+1+2

Order (Product_Id, Product_Name, Customer_Id, Customer_Name, Order_Date, Item_Price, Amount, VAT, Gross_Total, Net_Total)

Assumptions:

- The sales tax (VAT) value can vary from product to product (e.g. 8% for books, 16% for luxury items).
- The gross total is the net total price plus the sales tax.
- Customer orders on the same day are combined. We only have one order per customer and per day.
- Properties do not change over time – everything is "write-once".

a. Determine all the FDs in the relation Order.
b. Find all the candidate keys.
c. Find the closures for all the non-key attributes.

2. Find a minimal basis of the following sets of functional dependencies. 3

AB→C
C→A
BC→D
ACD→B
BE→C
CE→FA
CF→BD
D→EF

3. Normalize the following table upto BCNF. Show all work and clearly indicate the primary and foreign keys. 4

R(elevator_no, building_no, building_name, capacity, staff_no, first_name, last_name, date_examined) with the following functional dependencies:

1. elevator_no → building_no,capacity

2. building_no → building_name

3. staff_no → first_name,last_name

4. elevator_no,staff_no → date_examined

4. A PARTS file with Part# as hash key includes records with the following Part# values: 2369, 3760, 4692, 4871, 5659, 1821, 1074, 7115, 1620, 2428, 3943, 4750, 6975, 4981, 9208. The file uses 8 buckets, numbered 0 to 7. Each bucket is one disk block and holds two records. Load these records into the file in the given order using the hash function h(K)=K mod 8. Calculate the average number of block accesses for a random retrieval on Part#. 2

5. 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: 1.5+1.5+2

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).

(a) Suppose the file is ordered by the key field SSN and we want to construct a primary index on SSN. Calculate:

(i) the number of first-level index entries and the number of first-level index blocks;

(ii) the number of levels needed if we make it into a multi-level index;

(iii) the total number of blocks required by the multi-level index; and

(b) Suppose 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 a) for the secondary index and compare with the primary index.

(c) Suppose the file is not ordered by the key file ld 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 total number of blocks required by the B +-tree; and

(iii) the number of block accesses needed to search for and retrieve a record from the file--given its SSN value--using the B+-tree.