**CSE 370 – Database Systems**

## **Assignment 04**

## **Spring 2024**

**Full Name (in Block Letter): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ID: \_\_\_\_\_\_\_\_\_\_\_\_\_**

**Question 1 [CO6] : 10 Points**

1. **Insert the following search key values in a B+ tree. The order of the B+ tree, n = 4.**

**2, 5, 8, 11, 18, 22, 32, 26, 20, 21, 29, 43, 73, 66, 53**

**Each time there is a split, a new B+ tree should be drawn.**

1. **Explain the search path in the B+ tree, for the following search keys.**

* **11**
* **20**
* **12**

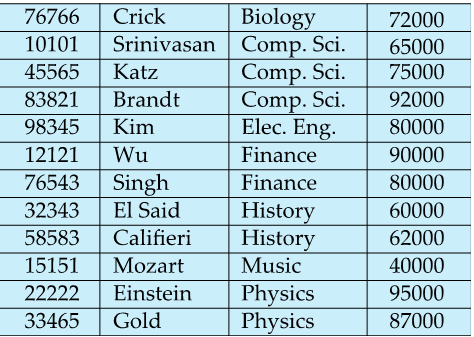
**Question 2 [CO6] : 10 Points**

**Insert the following search key values in a B+ tree. The order of the B+ tree, n = 5.**

**2, 5, 8, 11, 18, 22, 32, 26, 20, 21, 29, 43, 13, 66, 52, 70, 3**

**Each time there is a split, a new B+ tree should be drawn.**

**Question 3 [CO6] : 10 Points**

****

1. **Suppose you have a table titled ‘“Instructor”. Make a hash index on attribute instructor\_id. Suppose the number of buckets in the hash index is 4 and each bucket can house 2 index entries at max.**

**The hash function, h = (sum of all digits in id) % 4**

1. **Explain how to search for the following search keys in your hash index.**

* **45565**
* **22222**
* **11111**

1. **Repeat part (a), but now use a different type of hash function named ‘Mid Square Hashing”.**

The hash function will be generated using Mid-Square hashing. It works as follows:

Square the search key value. Extract the two middle digits from the squared results. If the squared result has n digits then you take the (n // 2)-th and ((n // 2) +1)-th digits.

Then, calculate the sum of the two extracted digits and take the remainder after dividing the sum with the number of buckets in the hash index.

**Consider the example below:**

**For ID = 76766:**

Square of 76766 = (76766 \* 76766) = 5893018756. This result contains n = 10 digits. So, the (10 // 2)-th and ((10 // 2) + 1)-th digits are 0 and 1, respectively.

* Sum of the 5th and 6th digits = 0 + 1 = 1.
* Remainder = 1 % 4 = 1.
* Therefore, with the correct consideration of the 5th and 6th digits, ID 76766 will be stored in bucket number 1 of the hash index.

**Question 4 [CO6] : 8 Points**

**Consider the relation Hospital\_Record with the following attributes:**

**Hospital\_Record** (Patient\_ID, Doctor\_ID, Patient\_Name, Admission\_Date, Discharge\_Date, Doctor\_Name, Doctor\_Specialization, Treatment\_Type, Room\_Number, Hospital\_Name, Hospital\_Location, Total\_Bill, Patient\_Rating, Doctor\_Performance)

The primary key of the relation is underlined. The relation has the following additional functional dependencies (FDs):

* **FD1**: Patient\_ID → Patient\_Name, Admission\_Date, Discharge\_Date, Treatment\_Type, Room\_Number, Hospital\_Name, Hospital\_Location, Total\_Bill
* **FD2**: Doctor\_ID → Doctor\_Name, Doctor\_Specialization, Doctor\_Performance
* **FD3**: Hospital\_Name → Hospital\_Location
* **FD4**: Treatment\_Type, Room\_Number → Total\_Bill

1. **Explain if the above relation is in the first normal form (1NF) or not? If not, apply 1NF normalization.**
2. **Explain if the relation(s) of no (a) is/are in the second normal form (2NF) or not? If not, apply 2NF normalization.**
3. **Explain if the relation(s) of no (b) is/are in the third normal form (3NF) or not? If not, apply 3NF normalization.**