Examination: Final

**Brac University**

**Department of Computer Science and Engineering**

Duration: 1 Hour 40 Minutes

(+10 Mins for Submission)

**CSE 370: Database Systems**

Semester: Fall 2023 Full Marks: 45

**Answer ALL of the following questions. Understanding the question is part of the exam**.

**1. [CO3] Construct** a relational Schema by mapping the following EER diagram for an online Classroom Management System. For the specialization/generalization portion, choose any applicable option except 8A: separate tables for subclasses and superclasses.

**13**

message

Date *Time*

FID

Messages

Name

SID

M Student T

submits

Details

Student

M

marks

Student 2 N

Email

N

Project Name

Projects

Faculty

Email

Enrolled\_in

Room

N

grade

M

M

submitted\_to

Name

Teaches

Counsulta tion\_Hour

Classroom

N

total\_students

Time

Day

Start Time

End

Time

ClassID

CourseCode

Course\_name

Section

Theory

Semester

duration

Lab

**2. [CO4**] Consider the following relation:

**RideService** (**jobID, driverID, clientID, jobDateTime, job PickUpAddress, dName, dContactNo, vehicleID, vehicleModel, maxSeat**, **mileage, cName, cContactNo, paymentLog, payment Method**, **paidAmount**, **reviewID)**

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

**FD1:** jobID → jobDateTime, jobPickUpAddress

**FD2:** driverID → dName, dContactNo, vehicleID, vehicleModel, maxSeat, mileage **FD3:** clientID → cName, cContactNo, paymentLog, paymentMethod, paidAmount **FD4:** paymentLog→ paymentMethod, paidAmount

**FD5:** vehicleID, vehicleModel → maxSeat, mileage

**10**

a. **Explain** if the above relation is in the first normal form (1NF) or not? If not, **apply** 1NF **| 2**

normalization.

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

2NF normalization.

4

c. **Explain** if the relation(s) of no (b) is/are in the third normal form (3NF) or not? If not, **apply** 4

3NF normalization.

**3. [CO5**] Consider the following relational database schema for a library management system.

Book

copyNo ISBN

title

edition year

price

available

Borrower

borrowerNo

borrowerName

borrower Address country

BookLoan

copyNo

dateOut

dateDue

borrower No

**[Foreign Key:** copyNo references Book (copyNo) and borrowerNo references Borrower (borrowerNo)]

The primary keys are underlined and foreign keys are mentioned in bold under each table that has any foreign keys.

Write appropriate SQL statements for the following questions (for each question write a **single query)**:

a. Retrieve all the Book title, ISBN, edition and price sorted by ascending order of price. If the

book has the same price, then sort based on the title in descending order.

b. For each country, list the number of borrowers if there are more than two borrowers for each

country. Print the country and the number of borrowers.

**2**

**2**

**C.**

List all the books that contain the phrase 'Harry Potter' in its title and whose price is more than the average book price of all books of 2023.

**3**

d. Find the maximum price of books that have been borrowed by the borrowers living in the

country 'Bangladesh'.

**3**

فيا

**4. [CO6]**

a.

**Construct** a B+ tree of order n = 4 for the following search key values inserted in the given order: 40, 30, 70, 20, 50, 60, 80, 25, 35, 45, 55, 65. Each time there is a split, **a new B+ tree must** be drawn.

b. **Construct** a hash index on attribute "Product Code" of the "Product" table. The hash index has 5 buckets, each capable of holding a maximum of 2 index entries. Bucket overflow is resolved using forward chaining.

The "Product" table is provided below:

**Product\_Code**

**Name**

**Price**

QY9

Laptop

999.99

MH6

Smartphone

499.99

EG5

Smart TV

799.99

FP3

Wireless Headphones

129.99

RT7

Digital Camera

349.99

U18

Coffee Maker

79.99

TK4

Bluetooth Speaker

59.99

FQ7

Fitness Tracker

149.99

**ASCII Value Chart**

**Character**

**Value**

**Character**

**Value**

**Character**

**Value**

**Character**

**Value**

0

48

9

57

I

73

R

82

1

49

A

65

J

74

S

83

2

50

B

66

K

75

T

84

3

51

C

67

L

76

U

85

4

52

D

68

M

77

V

86

5

53

E

69

N

78

W

87

6

54

F

70

O

79

X

88

7

55

G

71

P

80

Y

89

8

56

H

72

Q

81

Z

90

Steps involving hash function is summarized below:

Find the sum of the ASCII values for each character in the given Product Code string. Square the sum. Extract the two middle digits from the squared result. 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 Product Code = **'GX2'**, the ASCII values of the corresponding characters, 'G' = 71, 'X' = 88 and '2' 50. The sum of the individual ASCII values 718850 = 209. Square of the sum 209 (209\* 209) = 43681. This result contains n = 5 digits. So the (5// 2)-th and ((5 // 2) +1)-th digits of the squared sum are 3 and 6 respectively. Sum of the middle digits = 3+ 6 = 9. Remainder = 9 % 54. So the index entry of **'GX2**' will be stored in bucket 4.

7

5